# 10

# Programmable Keyboard and Display Interface: 8279

The entering of program or data in the microprocessor base systems is most commonly done by a keyboard. Hence, keyboard is the most versatile input device. The keyboards with hexadecimal or ASCII characters are basically a combination of switches placed in a matrix with rows and columns. Similarly, display devices are to be connected to the system to output the data or the result, and the most common output device is seven segment display. The keyboard has to be constantly scanned to detect a key-press. The display, however, has to be supplied with the data to hold it ready. If the CPU is required to do all these operations itself, it will be heavily burdened and will have lesser time for other processes. These operations are, therefore, carried out by another device so that CPU is relieved from all these burdens. For this purpose, Intel 8279 keyboard/display interface is designed to directly connect to 8085 microprocessor. It performs both keyboard scan and output display operations repetitively and very short time of CPU is utilized for the data transfer between the device and CPU. This chapter will entirely deal with the details of this programmable keyboard and display device 8279.

## 10.1 INTEL PROGRAMMABLE KEYBOARD/DISPLAY INTERFACE 8279

Intel 8279 Programmable Key Board/Display Interface is available in the form of 40 pin IC in plastic dual in line package (DIP). It has been designed to interface the key board (an input device) and display device (an output device) with microprocessor. The 8279 constantly scans to detect a key press and transmit the information of characteristics of the key press to the CPU. It also displays or outputs the data received from CPU to the display devices. These two operations keyboard scan and display are performed repetitively and independently without utilizing the time of CPU except for relatively short time when the data is actually transferred to and from the burden of scanning the keyboard or refreshing the display repetitively.

There are three input modes:

Scanned keyboard mode
Scanned Sensor matrix mode
Strobed input mode.

The keyboard can provide a scanned interface to 64 contact key matrix or array of sensors or a srobed interface keyboard. Key depressions can be 2 key lock out or N-key rollover. Keyboard entries are debounced and strobed in an 8-character FIFO (First In First Out) and set the interrupt lines. If more than 8 characters are entered, overrun interrupt status is set. The display provides a scanned display interface for LED, incandescent or other popular display technologies.

## 10.2 BLOCK DIAGRAM OF 8279

The pin diagram, logic diagram and functional block diagram of this 40 pin 8279 IC are depicted in figures 10.1, 10.2 and 10.3 respectively.

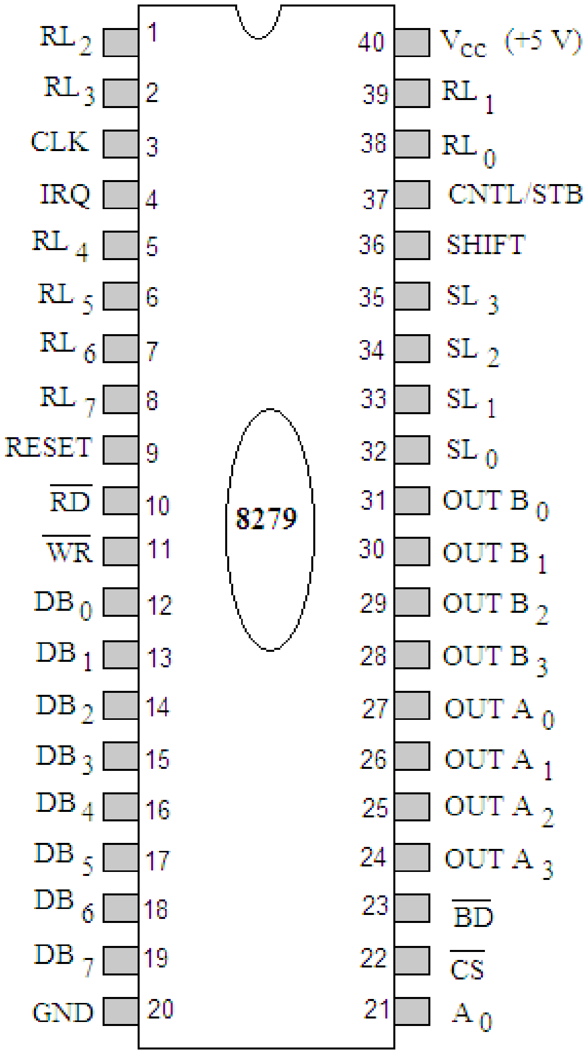


**Fig. 10.1**

The description of pins of 8279 Programmable keyboard/display interface is given as follows:

$DB_0$-$DB_7$: These pins form the bidirectional 8-bit data bus. The commands are also transcribed on this data bus.

CLK: It is a clock terminal to be connected to the external clock terminal of the system clock. It is used to generate internal timing. Maximum frequency to be used is 3 MHz.

$\overline{CS}$ : This is a chip select terminal. A low signal to this terminal enables the chip for programming, reading the keyboard, etc.

$A_0$: It is buffer address pin. A low signal on this pin indicates data and a high on this pin indicates the command.

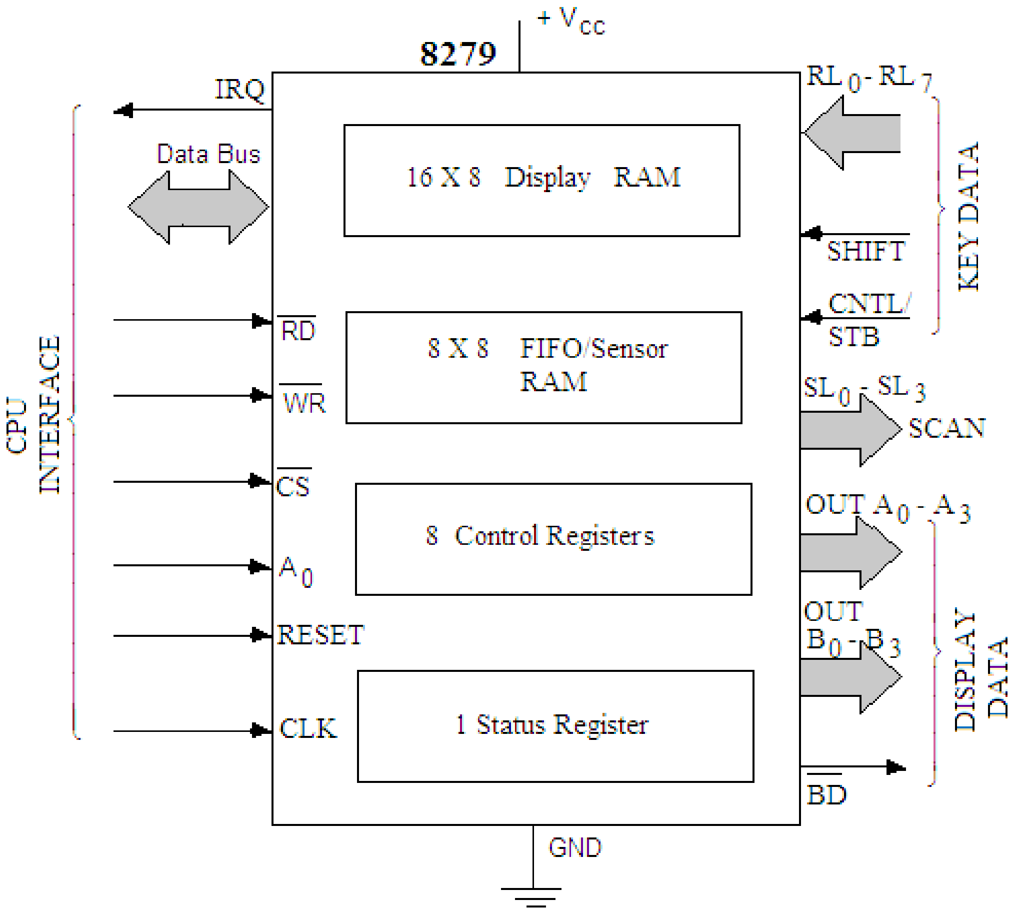


**Fig. 10.2**

IRQ: It is an interrupt request output terminal. Interrupt request, becomes 1 when a key is pressed, data is available.

$SL_0$-$SL_3$ : These four scan lines are used to scan the key switch or sensor matrix and the display digits. Scan line outputs scan both the keyboard and displays.

$RL_0$-$RL_7$: These are 8 Return line inputs which are connected to the scan lines through the key or sensor switches. These lines have internal pull-ups to keep high until a switch closure pulls one of the lines low.

SHIFT: This is an input terminal used in the scanned matrix keyboard modes. The SHIFT input status is stored along with the key position on key closure. SHIFT connects to shift key on keyboard.

CNTL/STB: This is control and strobe line, connected to the control key on the keyboard. It is high until a switch closure pulls it low.

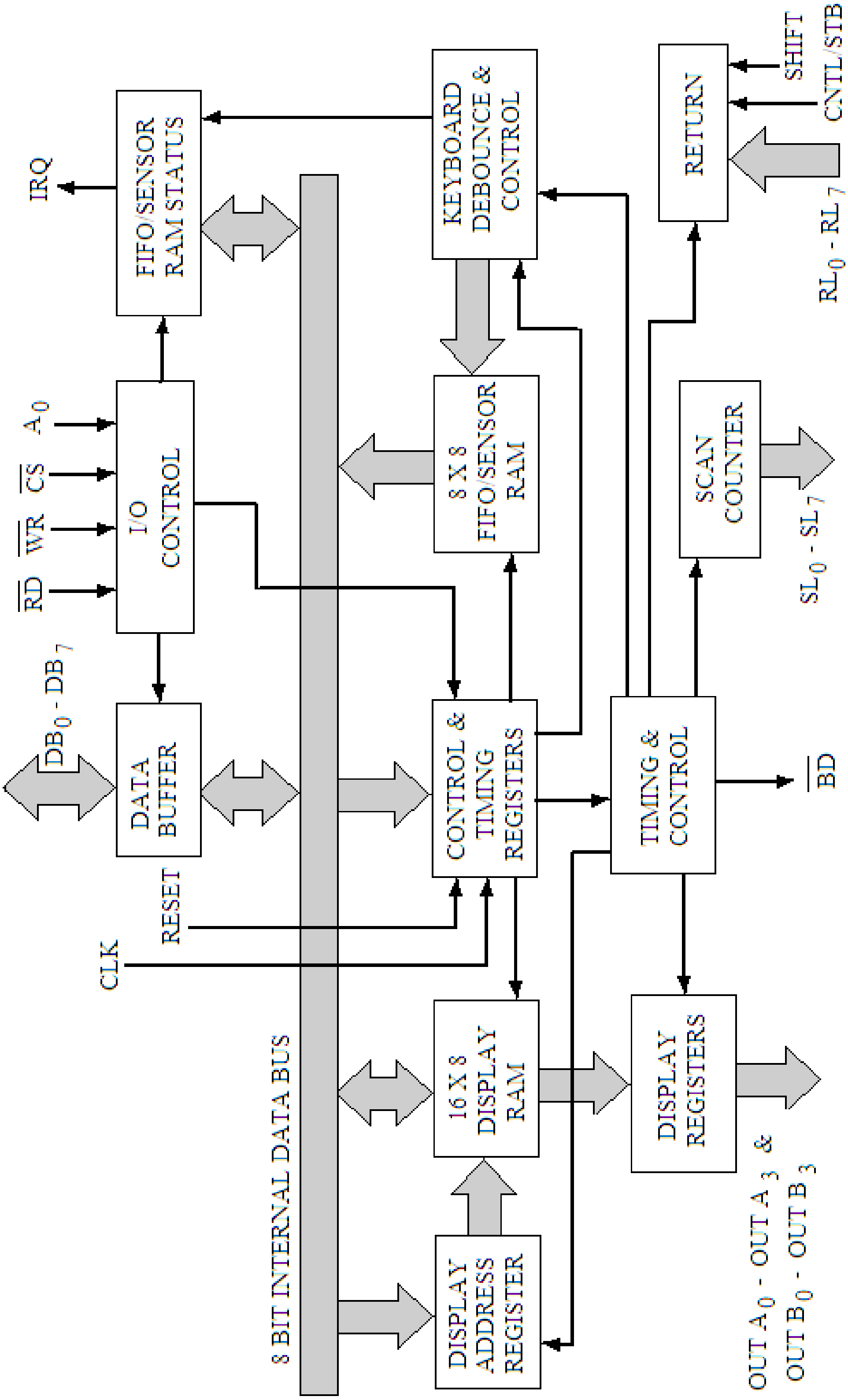

IRQ
FIFO/SENSOR RAM STATUS
$A_0$
$\overline{CS}$
$\overline{WR}$
$\overline{RD}$
I/O CONTROL
$DB_0$ - $DB_7$
DATA BUFFER
RESET
CLK
8 BIT INTERNAL DATA BUS
KEYBOARD DEBOUNCE & CONTROL
8 X 8 FIFO/SENSOR RAM
CONTROL & TIMING REGISTERS
16 X 8 DISPLAY RAM
DISPLAY ADDRESS REGISTER
RETURN
SHIFT
CNTL/STB
$RL_0$ - $RL_7$
SCAN COUNTER
$SL_0$ - $SL_7$
TIMING & CONTROL
$\overline{BD}$
DISPLAY REGISTERS
OUT $A_0$ - OUT $A_3$ &
OUT $B_0$ - OUT $B_3$

**Fig. 10.3**

SHIFT: This is an input terminal used in the scanned matrix keyboard modes. The SHIFT input status is stored along with the key position on key closure.

$\overline{RD}$: It is read input terminal and is connected to microprocessor's RD signal. It reads data/status registers.

$\overline{WR}$: It is write input terminal and is connected to microprocessor's write terminal.

OUT $A_0$ –OUT $A_3$ &
OUT $B_0$-OUT $B_3$: These are two 4-bit ports. The display output is through two 4-bit ports (OUT $A_0$ –OUT $A_3$ and OUT $B_0$-OUT $B_3$). These two ports can be combined to form an 8 bit port.

$\overline{BD}$: This is a blanking display terminal, used to blank the display during the digit switching or by a display blanking command.

## 10.3 FUNCTIONAL DESCRIPTION OF 8279

The functional description of programmable keyboard and display interface 8279 will now be discussed with reference to figures 10.2 and 10.3. The 8279 has mainly been divided into four sections:

- Keyboard Section
- Scan Section
- Display Section
- Microprocessor Interface Section

### Keyboard Section

The keyboard section includes Return buffer, keyboard debounce and control. Return buffers are to buffer the 8-input lines ($RL_0$-$RL_7$). In the keyboard mode, these lines are scanned, looking for the key closure in that row. If the debounce circuit detects a closed switch, it waits about 10 msec to check whether the switch remains closed. If it does, the address of the switch in the matrix in addition to other information is transferred to the FIFO.

The block FIFO/Sensor RAM and Status contains dual function 8 x 8 RAM. It will act as a FIFO in keyboard or strobed input modes. Each new entry is written into successive RAM positions and each is then read in the order of entry. FIFO status keeps track of the number of characters in the FIFO and whether it is full or empty. Too many reads or writes will be recognized as an error. The status can be read by $\overline{RD}$ and $\overline{CS}$ low and $A_0$ high. The status logic also provides an IRQ signal when the FIFO is not empty.

### Scan Section

The scan section has a scan counter and four scan lines $SL_0$-$SL_7$ which are used o scan the key switch or sensor matrix and display digits. This counter can be operated in two ways: Encoded mode and Decoded mode. In the encoded mode, the counter provides a binary count which can be decoded to provide the scan lines for the keyboard or display. However, in case of decoded mode, the scan counter itself is a decoder providing 1 of 4 scan.

### Display Section

This section contains the display address registers and display RAM. The display address registers hold the address of the word currently being written or read by the CPU and the two nibbles being displayed. The read/write addresses are programmed by CPU command. They can also be set to auto increment after each read or write. The display RAM can be directly read by the CPU after the correct mode and address is set. These are two 4-bit ports. The display output is through two 4-bit ports (OUT $A_0$-OUT $A_3$ and OUT $B_0$-OUT $B_3$). These two ports can be combined to form an 8 bit port. The data from these lines are synchronized to the scan lines ($SL_0$-$SL_3$) for the multiplexed digit displays. The two ports may be blanked independently by the blanking display terminal $\overline{BD}$. This section also includes 16 x 8 display RAM and the processor can read from or write into any of these registers.

**Microprocessor Interface Section**

The microprocessor interface sections contains 8-bit bidirectional data lines ($DB_0$-$DB_7$), one interrupt request line (IRQ), one address buffer line $A_0$, and five lines ($\overline{RD}, \overline{WR}, \overline{CS}$, RESET, CLK) for interfacing. When a high signal appears on $A_0$ terminal, the command or status registers inside the 8279 can be accessed i.e. it works as command word or status. A low, however, on this line indicates that the data register, e.g. display RAM or the FIFO/Sensor RAM can be accessed. The interrupt request line IRQ becomes high whenever data entries are stored in the FIFO. This signal is used to interrupt the microprocessor to indicate the availability of the data. The data and commands are communicated between the microprocessor and the 8279 through the data bus ($DB_0$-$DB_7$). The RESET pin resets the 8279, when a high signal appears on this pin. The two signals $\overline{RD}$ and $\overline{WR}$ enable the data bus to either send data to external bus or receive it from the external bus.

## 10.4 KEYBOARD SCAN

As already discussed, the 8279 provides 4 scan lines ($SL_0$-$SL_3$) and 8 return lines ($RL_0$-$RL_7$). The scan lines can be operated either in encoded mode or in decoded mode.

**Encoded Mode:**

In this encoded mode, four scan lines ($SL_0$-$SL_3$) can be used to generate 16 lines with the help of external decoder. Usually 3 to 8 line decoder is used with the scan lines. The most significant scan line $SL_3$ is not recommended to use in keyboard decoder. So with three scan lines ($SL_0$-$SL_2$) and a 3 to 8 line decoder, 8 decoded scan lines are generated. These 8 decoded scan lines in conjunction with the 8 return lines can form an 8 x 8 keyboard matrix as shown in figure 10.4. Two more extra lines SHIFT and CNTL (control) can also provide four more different combinations as shown in table 10.1.

**Table 10.1**

| SHIFT | CNTL |
|---|---|
| 0 | 0 |
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |

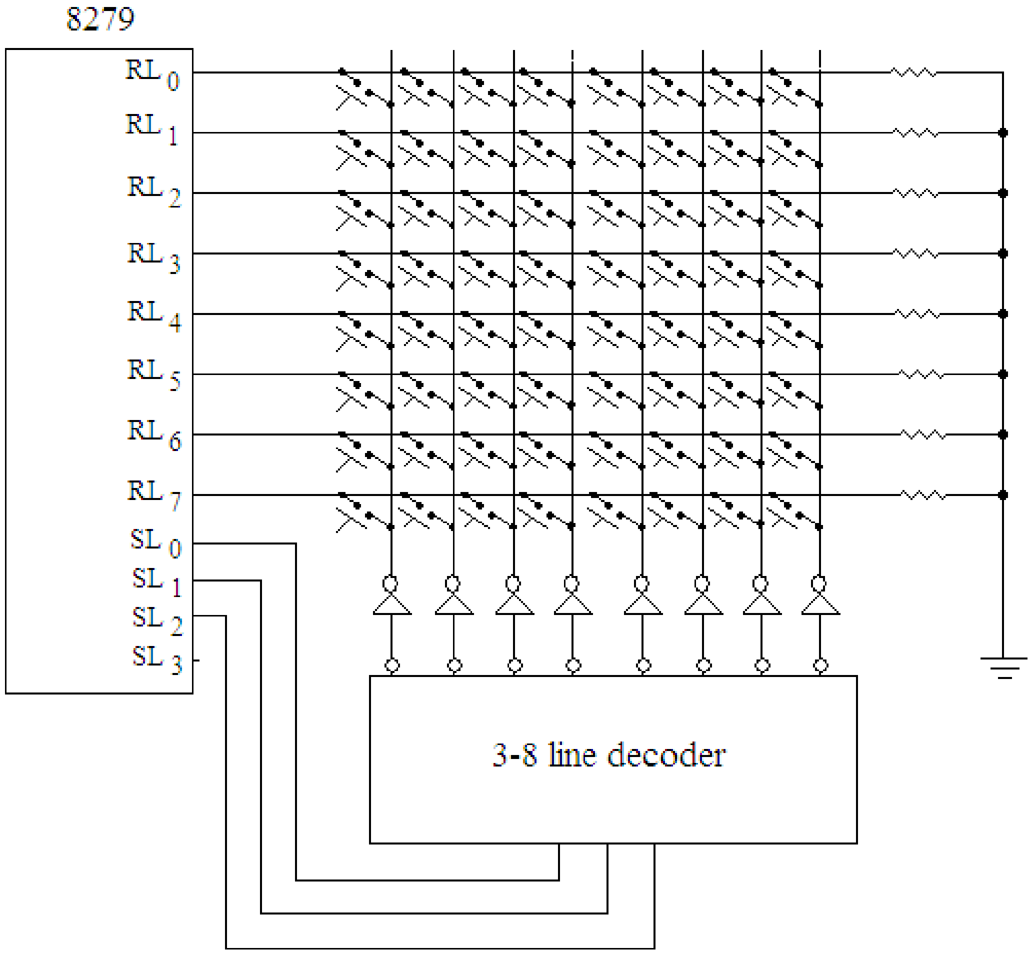


**Fig. 10.4**

With these four combinations and 8 x 8 matrix 4 x 8 x 8 = 256 character definitions are possible.

**Decoded Mode:**

In the decoded mode, an internal decoder is used to provide all the four scan lines (SL0-SL3) as the decoded lines as shown in table 10.2.

**Table 10.2**

| $SL_3$ | $SL_2$ | $SL_1$ | $SL_0$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |

These four decoded lines with 8 return lines form the keyboard matrix as shown in figure 10.5. The SHIFT and CNTL lines in combination with the keyboard matrix can provide 4 x 4 x 8 128 character definitions.

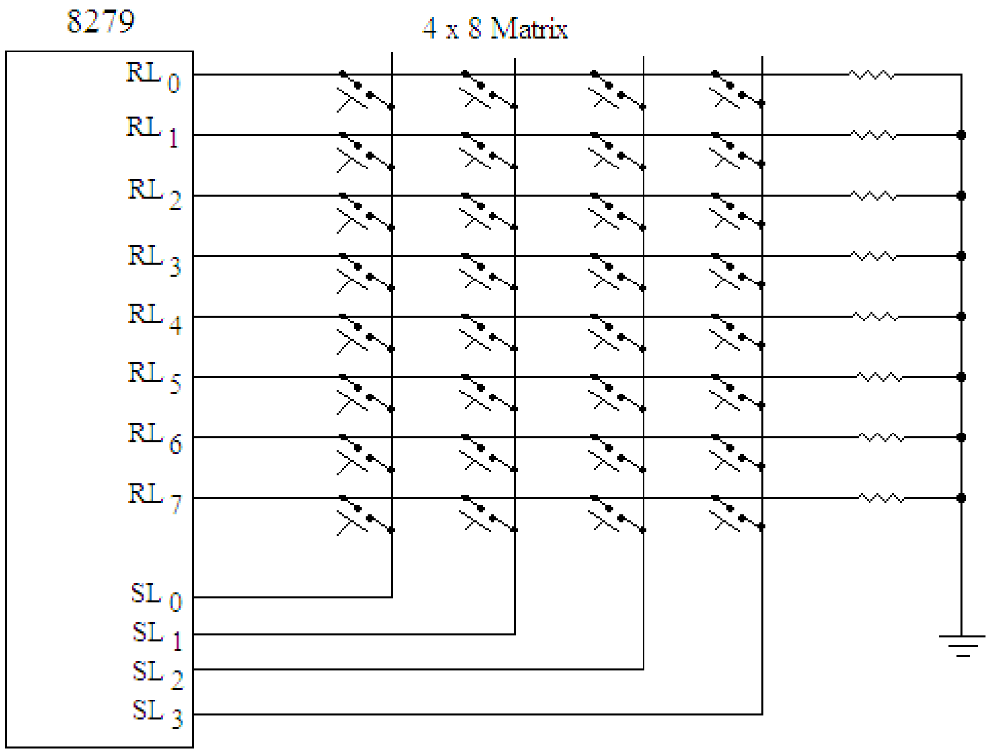
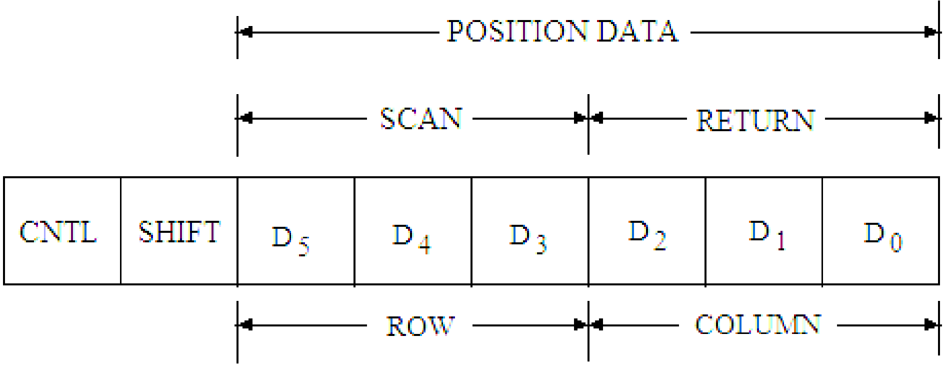


**Fig. 10.5**

The disadvantage of the decoded scan is that the number of combinations of the scan lines as given in table is only four. Hence, only four rows can be used in case of keyboard and only four digits can be used for the display.

## 10.5 SCANNED KEYBOARD

In this mode, when a key is pressed, a unique 6 bit data is generated characteristic of the key position. An 8-bit word is formed, with the 6-bit position data for the key pressed and two bits for control (CNTL) and SHIFT lines. The format for the scanned keyboard mode is shown in figure 10.6. The scan counter has three scan bits ($D_5$-$D_3$) as



DATA FORMAT FOR SCANNED KEYBOARD MODE

**Fig. 10.6**

000 to 111 for the row on which the pressed key is located. The column counter also has three bits ($D_2$-$D_0$) as 000 to 111 for column on which the pressed key is located. Figure

10.7 shows the 8 x 8 keyboard matrix that has the 8 return lines and 8 scan lines. It has 64 key positions (0 to 63) which generate 8 bit data word for the key pressed. The 8 bit data word generated for a key pressed may be understood if we consider CNTL and SHIFT lines are 00. Suppose a key number 15 is pressed. The row line corresponding to the key number 15 is 1 (three bit binary is 001) and the column line for this pressed key is 7 (three bit binary is 111) so the 8 bit data formed for this pressed key becomes:

00 001 111

which is the binary equivalent of 15 (0F H). Similarly, the 8 bit data word for the key pressed 32 is 00 10 000 (20 H).

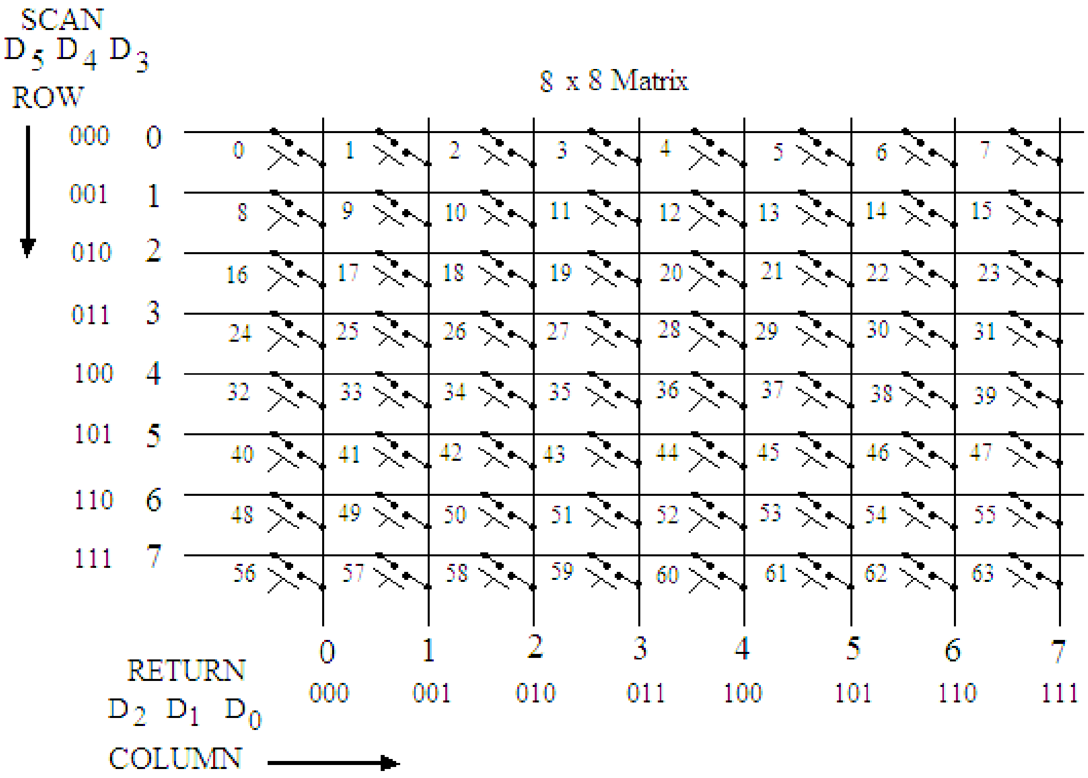


**Fig. 10.7**

The 8-bit data thus formed corresponding to the key pressed is stored in FIFO (first in first out) RAM inside the 8279. As soon as the 8-bit word is stored in FIFO RAM, IRQ terminal of 8279 goes high. This terminal is connected to one of the hardware interrupt line of CPU. With the high IRQ, the interrupt line of the CPU will be activated and the service subroutine program of the hardware interrupt will read the data from the FIFO RAM. As the data is read from the FIFO RAM, the IRQ terminal goes low; and it becomes high again if FIFO RAM contains further data.

The scanned keyboard mode has further two alternative ways of operation:

- Two-key lockout
- N-key rollover

### 10.5.1 Two-key Lockout

The mechanical keys used in the keyboard have a problem. When a key is pressed the contacts bounce back and forth and finally settle down after a small time. Due to this contact bounce problem multiple entries may be made for the same key. This can be avoided either by using the debouncing circuit using flip-flops etc. or by making the device to wait for few milliseconds after the key is pressed. In the 8279, second method is used for debouncing which is the built in feature of this IC.

In the two-key lockout operation, if two keys are depressed within the bounce cycle, it is called a simultaneous depression and neither key will be recognized until one of the keys is released. The last key released will be recognized and its corresponding 8-bit code will be entered in FIFO RAM.

If after the first key is depressed, and no other key is found to be depressed within next two scans, then it is taken as a single key depression and the code for the depressed key will be entered in FIFO RAM.

If after the first key depression, one or more additional key depression is detected within the next two scans, then there will be following two possibilities:

If all keys are released before the first pressed key, then the first key data will be entered in FIFO RAM.

If first key pressed is released before others, then the press key will be entirely ignored.

### 10.5.2 N-key Rollover

In this mode, each key depression is treated independent. If simultaneous key depression occurs then keys are recognized and entered in FIFO RAM according to the order of the key pressed. In fact when a key is pressed the debounce circuit inside the 8279 waits for two scans then checks if this key is still pressed. If this key is still pressed, the code for the key depressed is entered into FIFO RAM.

## 10.6 SCANNED SENSOR MATRIX

As discussed earlier, the keyboard matrix size is 8 x 8 in encoded scan lines and 4 x 8 in decoded scan lines. In this mode, the keys are placed in the form of matrix either in 8 x 8 encoded scan lines or 4 x 8 decoded scan lines, the scan lines form the columns and return lines form the rows of the keyboard matrix. The key status (open or closed) is stored in RAM which can be addressed by the CPU. The data on each of the eight lines enter directly in eight columns of sensor RAM; and each switch position maps to specific sensor RAM positions. The SHIFT and CNTL lines are not considered as inputs. The format for each row of the sensor RAM is shown in figure 10.8. The logic circuits can also be connected to the return lines which will be triggered by the scan lines. The debouncing circuit is not provided in this mode. It therefore has the advantage that the CPU knows how long the sensor was closed. The IRQ line goes high if a sensor value is found to have changed at the end of sensor matrix scan. The IRQ line is cleared by the first data read operation if the auto increment flag is set to zero or by the End Interrupt Command if the auto increment flag is set to one.

| MSB | | | | | | | LSB |
|---|---|---|---|---|---|---|---|
| $RL_7$ | $RL_6$ | $RL_5$ | $RL_4$ | $RL_3$ | $RL_2$ | $RL_1$ | $RL_0$ |

**Fig. 10.8**

## 10.7 STROBED INPUT

In this mode, the data is accepted from the return lines and go to FIFO RAM and entered at the rising edge of CNTL/STB line pulse. The data placed on the return lines can come from any source. The data is stored in the same format as shown in figure 10.8. Each scan line would lead to an 8-bit word.

## 10.8 DISPLAY INTERFACE

The interfacing of keyboard with the 8279 has been discussed in the preceding sections. The interfacing of display devices with the 8279 will now be discussed. Generally seven segment display devices are connected with 8279 using the multiplexing technique. In the multiplexing technique the seven segment code is sent to all the displays simultaneously, but the particular segment to be illuminated is only grounded (in case of common cathode displays).

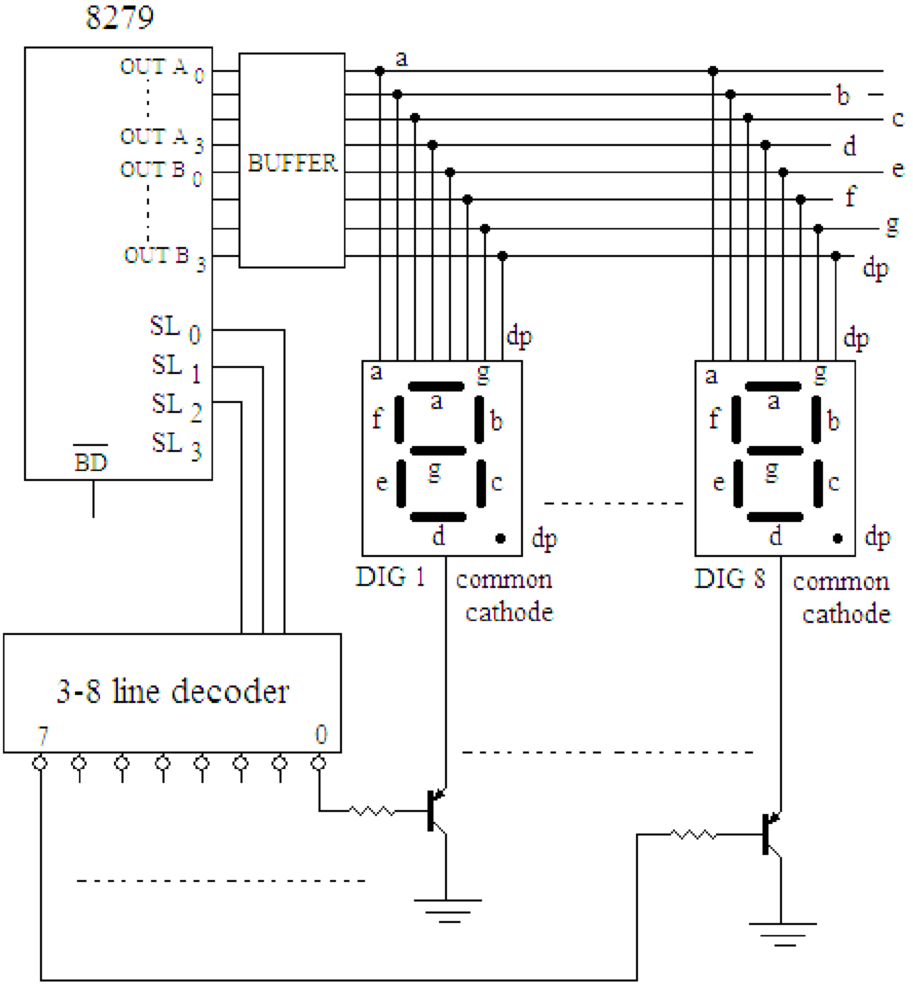
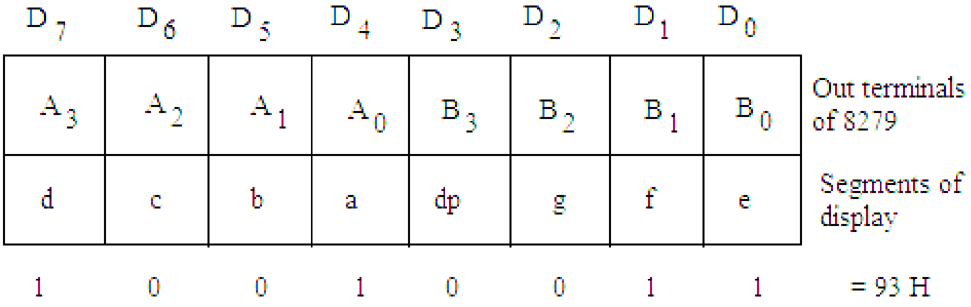


**Fig. 10.9**

In the 8279, eight output terminals, OUT $A_0$-$A_3$ and OUT $B_0$-$B_3$ are provided for the purpose of interfacing the seven segment displays. If a 4-to-16 line decoder is used with the scan lines ($SL_0$-$SL_3$), then a maximum of 16 display devices may be connected to the 8279. The internal FIFO RAM of 8279 can hold 8-bit data for 16 digits. If a 3-to-8 line decoder is used with three scan lines ($SL_0$-$SL_2$), then a maximum of 8 displays may be connected to this IC. All the segments of display devices are connected in parallel (i.e. a's segment of all the displays are tight together, similarly for b's to g's and dp's segments). These segments are then connected to the output terminals OUT $A_0$-$A_3$ and OUT $B_0$-$B_3$, as shown in figure 10.9. The OUT $A_0$, OUT $A_1$, OUT $A_2$ and OUT $A_3$ lines of the 8279 are connected to segments a, b, c and d respectively. The OUT $B_0$, OUT $B_1$, OUT $B_2$ and OUT $B_3$ lines of the 8279 are connected to segments e, f, g and dp respectively. The decoded outputs of scan lines provide 0 to 7 outputs in a periodic fashion, to select one digit of the display devices at a time. The $\overline{BD}$ line is used to blank all display digits.

When a given bit is 1, the corresponding segment is switched ON. For example the key code for the alphabet 'C' is obtained if the segments a, d, e and f are high. The data code for the alphabet 'C' is 93 as shown in figure 9.10.



| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | |
|---|---|---|---|---|---|---|---|---|
| $A_3$ | $A_2$ | $A_1$ | $A_0$ | $B_3$ | $B_2$ | $B_1$ | $B_0$ | Out terminals of 8279 |
| d | c | b | a | dp | g | f | e | Segments of display |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | = 93 H |

**Fig. 10.10**

## 10.9 DISPLAY MODES

The interfacing of display devices with 8279 has been discussed in the earlier section. Further there are following two options for the display formats in the display modes of 8279:

- Left Entry Mode (Type Writer Mode)
- Right Entry Mode (Calculator Mode)

### 10.9.1 Left Entry Mode (Type Writer Mode)

In the left entry mode, the first location of the display RAM data is treated as the segment for the left most digit and second location of the display is treated as the segment data for the second digit from the left and so on. This is similar to typing the paper with the type writer. In this mode there is auto-increment facility as in the type writer; the

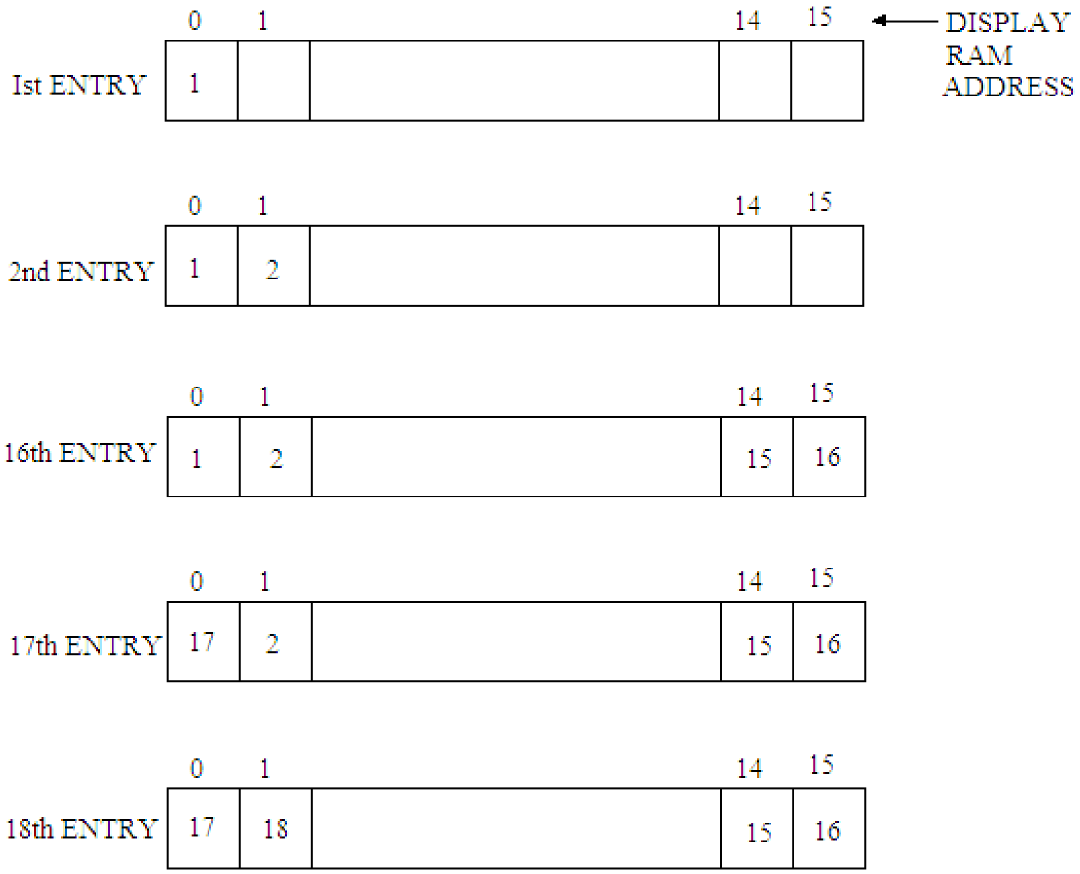
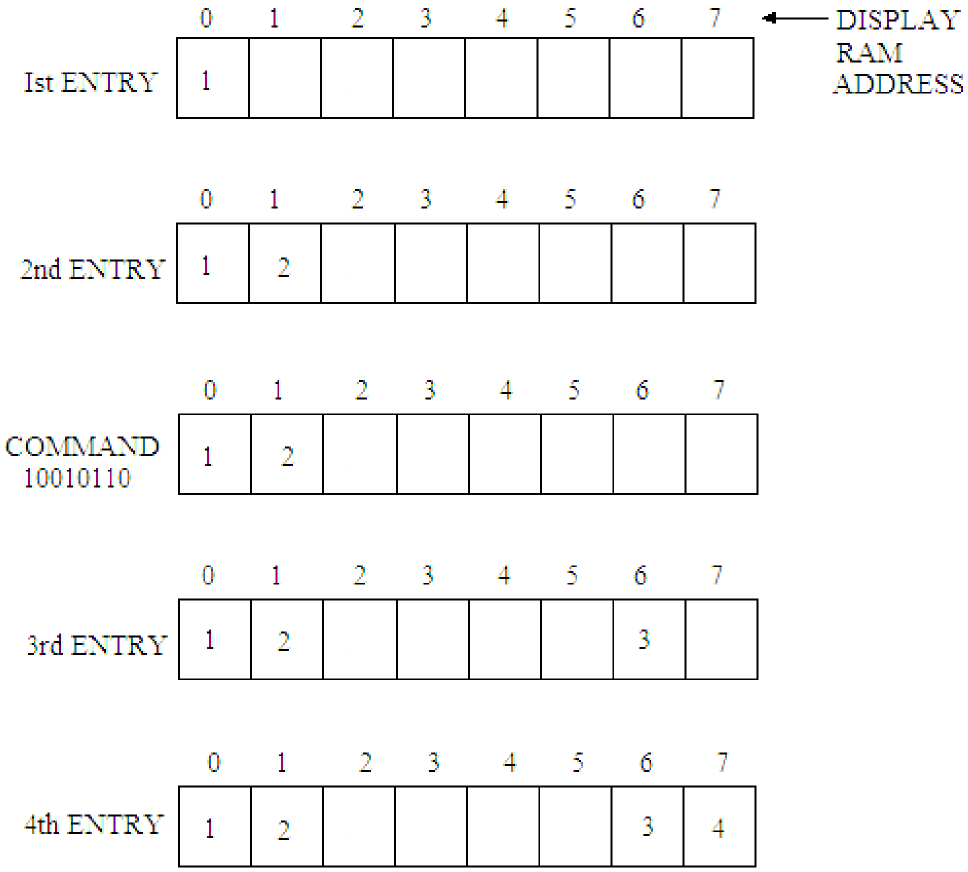


**Fig. 10.11**

carriage advances one step during typing. The left entry mode with auto increment facility is illustrated in figure 10.11 for 16 digits to be displayed on the display devices. From this figure it is clear that the first entry goes to the address 0 (first one of sixteen) for one word RAM and to the left most display position. The second entry goes to address 1 and the second display position and so on. The 16th entry goes to the address 15 of the display RAM and position 16 of the display. The 17th entry fill the left most position again.

There is a command (the details of which will be discussed in the next section) which allows entering data at an arbitrary address location of the display RAM. Figure 10.12 illustrates the result of a command that is used to display the next data to the 6th position using 8-digit display. The command word given here is 10010110 which contains 8 bits, the three most significant bits 100 represents the "write display RAM": the next bit 1 is for auto increment and the next four bits 0110 (6th) are for the position at which it should start filling. This command does not lead to any undesirable result.



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Ist ENTRY | 1 | | | | | | | |
| 2nd ENTRY | 1 | 2 | | | | | | |
| COMMAND<br>10010110 | 1 | 2 | | | | | | |
| 3rd ENTRY | 1 | 2 | | | | | 3 | |
| 4th ENTRY | 1 | 2 | | | | | 3 | 4 |

**Fig. 10.12**

### 10.9.2 Right Entry Mode (Calculator Mode)

The right entry mode is also known as calculator mode. In this mode the first location of the display RAM indicates the display data for the right most digit. Therefore, on the display, the data appears to start from the right and shift towards left as the digits move left in the calculator when the data is entered in to it. Figure 10.13 shows how the data are entered in this mode for 16 digits display with auto increment. The first character appears at the right display position. When a second character enters, the first character shifts towards the left by one place on the display. Similarly at the third entry, both the characters move by one place left each and the third character again takes the original right most position. It has been observed that a given character entered at a certain display position continues to remain there; but in case of right entry mode at every new entry each existing character moves one place to the left and finally the left most character shifts off the end, and is lost.

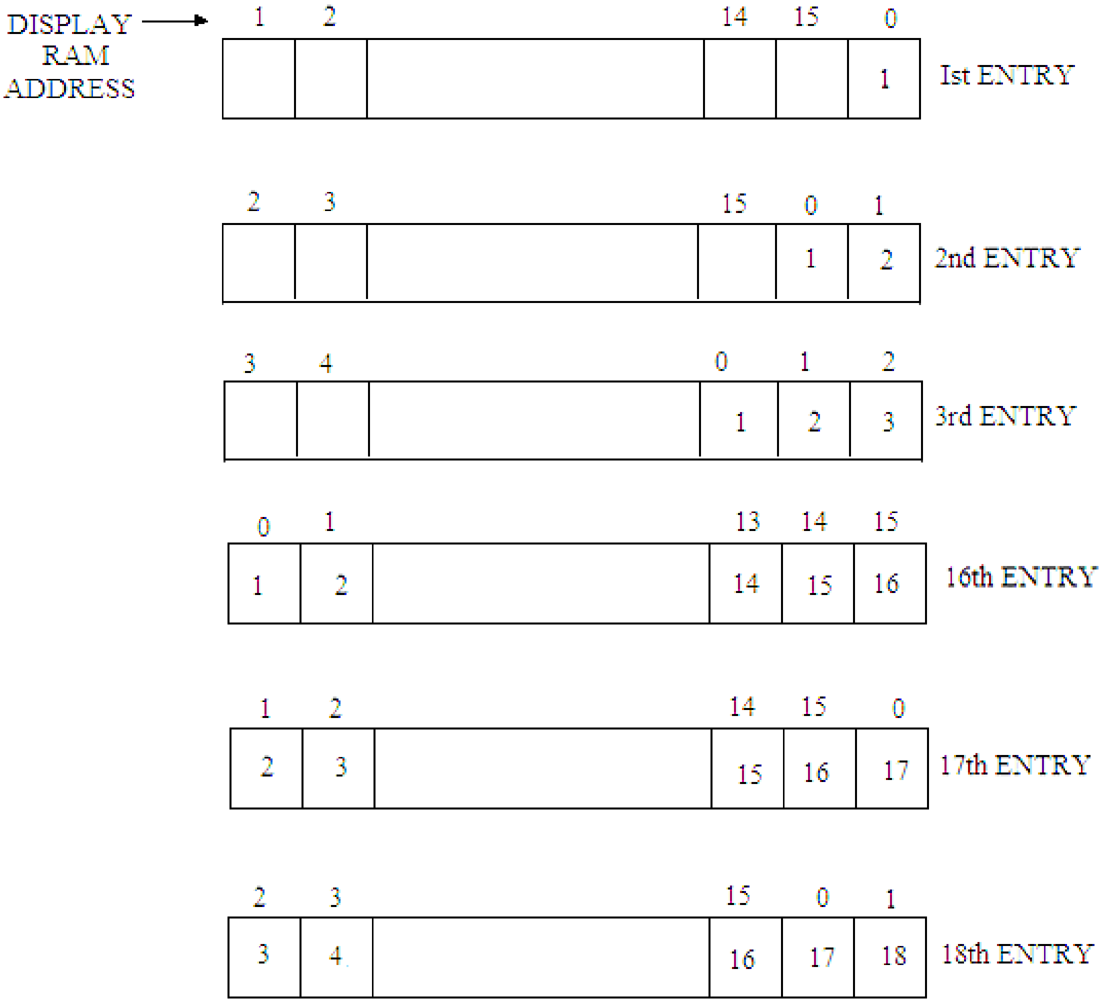


**Fig. 10.13**

Figure 10.14 shows the entering of the data in right entry mode with auto increment for 8 digit display using the command word for entering data at an arbitrary address location of the display RAM. The results obtained for entering data at an arbitrary address location in this mode are unexpected. Therefore, a starting display RAM address 0 and sequential entry are recommended in this right entry mode.

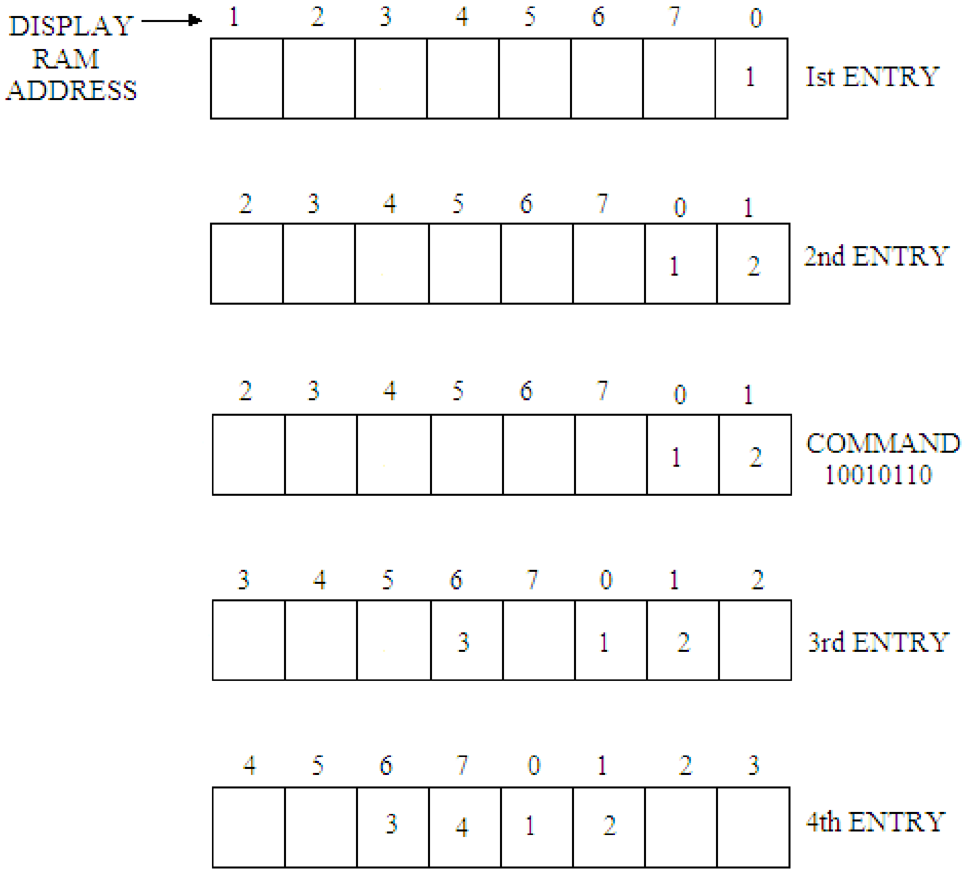


**Fig. 10.14**

## 10.10 PROGRAMMING OF 8279

The 8279 is a programmable keyboard and display interface device so it may be programmed for the desired operation. There are following 8 commands that can be used in the 8279:

- Keyboard/Display Mode Set
- Program Clock
- Read FIFO/Sensor RAM
- Read Display RAM
- Write Display RAM
- Display Write Inhibit/Blanking
- Clear
- End Interrupt/Error Mode Set

The command word, whose format is shown in figure 10.15, is sent on the data bus with $\overline{CS}$ low and $A_0$ high. The word is loaded to the 8279 using the write operation.

# 13

# Interfacing Data converters: A/D and D/A Converters

Sometimes the information available for processing in microprocessor based system is in digital form while in most of the cases it is available in analog form. For example, the outputs of digital voltmeter, digital frequency meter, digital clock and calculators etc. are available in digital form but most physical quantities such as temperature, pressure, light, voltage and current etc. gives information in analog form. It is often necessary to convert information in one form to another form for the purpose of interfacing with the system. For example, to design the microprocessor base temperature controller, the temperature of the device obtained from the transducer such as thermocouple or thermister is first converted to the digital form using D/A converter then interfaced with the microprocessor. Similarly, for plotting the output of a system on a curve plotter or X-Y recorder, the digital output is first converted to analog output with the help of digital to analog converter, the output of which drives a servomotor. So analog to digital (A/D) converters or digital to analog (D/A) converters are the interfacing devices with the system. In this chapter various types of A/D and D/A converters and their interfacing with the microprocessor will be discussed.

## 13.1 DIGITAL TO ANALOG CONVERTER

Digital to Analog (D/A) converter converts the digital information into analog form. The input may be of *n*-bit long having different voltage levels. So in the D/A converters, some method is to be used which can convert this voltage level of *n*-bits to its equivalent analog form. This can be accomplished by using different resistive networks. Following two types of resistive networks are basically used for this purpose:

1. Resistive Divider Network or weighted resistor network
2. Binary Ladder Network or R-2R network

The converter which comprises the resistive divider network is known as Resistive Divider D/A converter and the D/A converter which comprises the binary ladder network is known as Binary Ladder D/A converter. These converters will now be discussed.

### 13.1.1 Resistive Divider D/A converter

As discussed above, the resistive divider D/A converter consists of a resistive divider network, so before discussing the complete circuit diagram of a resistive divider D/A converter, it is better to understand the working of resistive divider network. The resistive divider network changes each of the *n*-bit digital level into its equivalent analog output. The discussion will now be made for the method of converting the *n*-bit digital input to its equivalent analog signal. A weight is assigned to each bit of *n*-bit digital input in such a way that the sum of weight must be equal to 1. In general, the binary weight assigned to LSB in an *n*-bit digital input is $\frac{1}{2^n - 1}$. The weights assigned to 2nd LSB, 3rd LSB, 4th LSB and so on are obtained by multiplying the weights of LSB to $2^1$(=2), $2^2$ (=4), $2^3$ (=8)…. respectively. For instance, weights assigned to different bits of 4-bit binary input $b_3$ $b_2$ $b_1$ $b_0$ are:

Weight assigned to LSB ($b_0$ bit) is $\frac{2^0}{2^4-1} = \frac{1}{15}$

Weight assigned to 2nd LSB ($b_1$ bit) is $\frac{2^1}{2^4-1} = \frac{2}{15}$

Weight assigned to 3rd LSB ($b_2$ bit) is $\frac{2^2}{2^4-1} = \frac{4}{15}$

Weight assigned to MSB ($b_3$ bit) is $\frac{2^3}{2^4-1} = \frac{8}{15}$

The sum of weights assigned to each bit of 4-bit digital input is 1 as $\frac{1}{15} + \frac{2}{15} + \frac{4}{15} + \frac{8}{15} = 1$.

In a four bit binary system there will be 16 different possible input combinations, corresponding to which the analog signal will be obtained if it is assumed that a certain reference voltage ($V_{REF}$) is applied whenever there is a 1 in binary bit. In a 4 bit digital system if $V_{REF}$ =15 volts, the analog voltage available for each combination of binary input should be as given in table 13.1.

**Table 13.1**

| **$b_3$** | **$b_2$** | **$b_1$** | **$b_0$** | **Weight** | **Analog Voltage** |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0/15 | (0/15)$V_{REF}$ = 0 Volt |
| 0 | 0 | 0 | 1 | 1/15 | (1/15) $V_{REF}$ = 1 Volt |
| 0 | 0 | 1 | 0 | 2/15 | (2/15) $V_{REF}$ = 2 Volt |
| 0 | 0 | 1 | 1 | 3/15 | (3/15) $V_{REF}$ = 3 Volt |
| 0 | 1 | 0 | 0 | 4/15 | (4/15) $V_{REF}$ = 4 Volt |
| 0 | 1 | 0 | 1 | 5/15 | (5/15) $V_{REF}$ = 5 Volt |
| 0 | 1 | 1 | 0 | 6/15 | (6/15) $V_{REF}$ = 6 Volt |
| 0 | 1 | 1 | 1 | 7/15 | (7/15) $V_{REF}$ = 7 Volt |
| 1 | 0 | 0 | 0 | 8/15 | (8/15) $V_{REF}$ = 8 Volt |
| 1 | 0 | 0 | 1 | 9/15 | (9/15) $V_{REF}$ = 9 Volt |
| 1 | 0 | 1 | 0 | 10/15 | (10/15) $V_{REF}$ = 10 Volt |
| 1 | 0 | 1 | 1 | 11/15 | (11/15) $V_{REF}$ = 11 Volt |
| 1 | 1 | 0 | 0 | 12/15 | (12/15) $V_{REF}$ = 12 Volt |
| 1 | 1 | 0 | 1 | 13/15 | (13/15) $V_{REF}$ = 13 Volt |
| 1 | 1 | 1 | 0 | 14/15 | (14/15) $V_{REF}$ = 14 Volt |
| 1 | 1 | 1 | 1 | 15/15 | (15/15) $V_{REF}$ = 15 Volt |

So the analog voltage for binary word = (weight of the binary word) x $V_{REF}$

It may be noted from this table 13.1 that the analog voltage corresponding to binary equivalent is discrete step value as given in figure 13.1. The discrete step is of 1 volt if $V_{REF}$ is assumed to be 15 volts in a four bit digital input. The step voltage (analog) will be dependent on the reference voltage. There will, however, be $2^n$ steps in *n*-bit digital system.

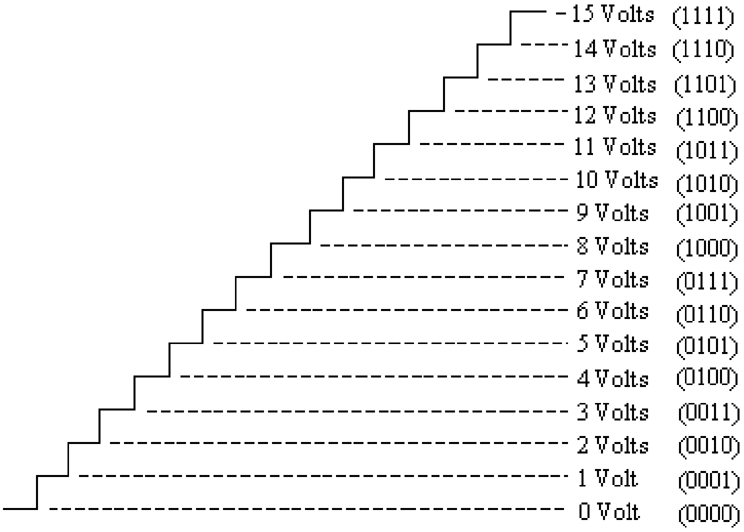


**Fig. 13.1**

Resistive divider network is used for converting digital inputs to analog outputs. The network for 6 bit binary system shown in figure 13.2 is known as the weighted network, as the resistors are weighted inversely with their current values. The input binary bits are $b_5$ $b_4$ $b_3$ $b_2$ $b_1$ $b_0$ where $b_0$ is the LSB and $b_5$ is MSB. These binary bits may be logic 0 or 1. Logic 0 may further be assumed as 0 volt and logic 1 as $V_{REF}$. So $V_0$, $V_1$, $V_2$, $V_3$, $V_4$ and $V_5$ are the input voltage levels which may be 0 volt or $V_{REF}$ depending on the binary bits. The resistors $R_0$, $R_1$, $R_2$, $R_3$, $R_4$ and $R_5$ are connected to bits $b_{0,}$ $b_{1,}$ $b_2$ , $b_3$, $b_{4,}$ $b_5$ respectively. It may be noted from this network that the resistor connected to the binary bit is half the value of resistor connected to the previous (lower) bit. Hence this network also called as the resistive divider network. Let $R_L$ is the load resistance which is supposed to very high i.e. very much higher than the resistor $R_0$.

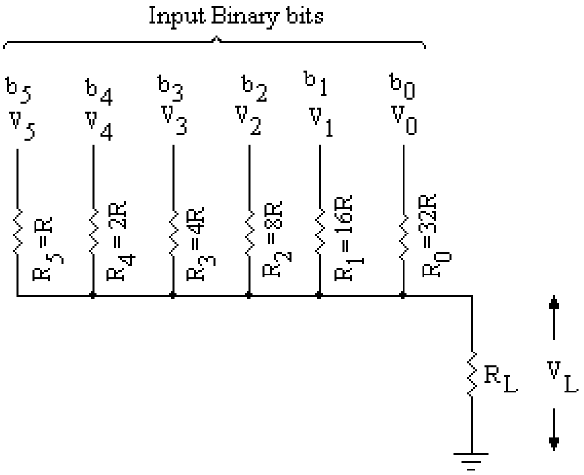


**Fig. 13.2**

Now the voltage $V_L$ across the load resistance $R_L$ can be obtained by using Millman's theorem. This theorem states that the voltage appearing at any node in a resistive network is equal to the sum of all the currents that would enter to the node divided by the sum of conductances connected to that node.

Thus
$$V_L = \frac{\frac{V_5}{R_5}+\frac{V_4}{R_4}+\frac{V_3}{R_3}+\frac{V_2}{R_2}+\frac{V_1}{R_1}+\frac{V_0}{R_0}}{\frac{1}{R_5}+\frac{1}{R_4}+\frac{1}{R_3}+\frac{1}{R_2}+\frac{1}{R_1}+\frac{1}{R_0}}$$

$$= \frac{\frac{V_5}{R}+\frac{V_4}{2R}+\frac{V_3}{4R}+\frac{V_2}{8R}+\frac{V_1}{16R}+\frac{V_0}{32R}}{\frac{1}{R}+\frac{1}{2R}+\frac{1}{4R}+\frac{1}{8R}+\frac{1}{16R}+\frac{1}{32R}}$$

$$= \frac{\frac{[32V_5+16V_4+8V_3+4V_2+2V_1+V_0]}{32}}{\frac{[32+16+8+4+2+1]}{32}}$$

$$= \frac{V_0+2V_1+4V_2+8V_3+16V_4+32V_5}{63}$$

$$= \frac{1}{(2^6-1)}(2^0V_0+2^1V_1+2^2V_2+2^3V_3+2^4V_4+2^5V_5) \quad \ldots(13.1)$$

In this equation (13.1), the load resistance $R_L$ is not considered as it is assumed to be large enough offering low (almost zero) conductance. From this equation it is clear that if the input binary bits are all 1 (in a six bit system) and reference voltage $V_{REF}$ = 6.4 volts (say), the $V_L$ is given by:

$$V_L = \frac{1}{63}x63V_{REF} = 6.4volts$$

In general, the equation (13.1) for output voltage of *n*-bit binary digits is given as:

$$V_L = \frac{1}{(2^n - 1)}(2^0 V_0 + 2^1 V_1 + 2^2 V_2 + 2^3 V_3 + 2^4 V_4 ......... + 2^{n-1} V_{n-1})$$

…(13.2)

The output of this network is as per our requirement, and is proportional to the input binary data.

Using the network discussed above, a D/A converter (called binary weighted D/A converter or Resistive divider D/A converter) can be designed as given below. The schematic diagram of 6-bit D/A converter is shown in figure 13.3. It consists of the following major parts:

(i) n switches, one for each bit applied to the input,

(ii) A binary weighted resistive network which changes each of the digital level into equivalent binary weighted voltage or current.

(iii) A reference voltage source $V_{REF}$.

(iv) A summing amplifier that adds the currents flowing in the resistors of the network to develop a signal that is proportional to the digital input.

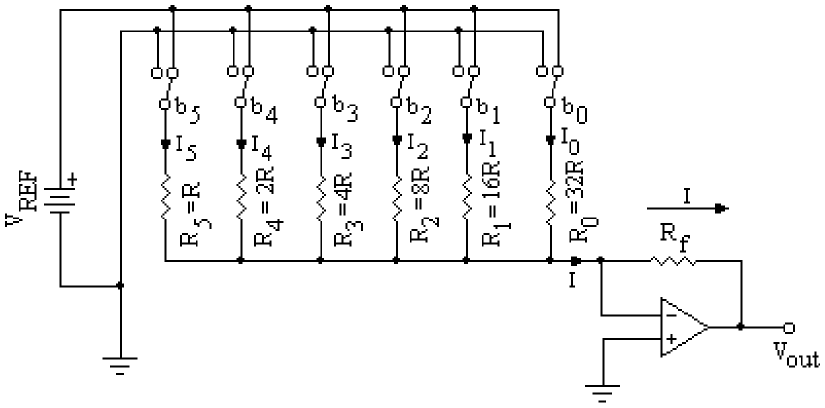


**Fig. 13.3**

In this circuit, one switch is connected to each binary bit. In fact these switches are such that when the binary bit is 0, the corresponding resistor of the network gets connected to the ground potential and when the binary bit is 1, the corresponding resistor of the network gets connected to the $V_{REF}$ volt. The current flowing through any branch of the network will be the logical voltage (0volt or $V_{REF}$ volts) divided by the corresponding resistor.

So the total current I will be given by (ref. fig. 11.3):

$$I = \frac{V_5}{R_5} + \frac{V_4}{R_4} + \frac{V_3}{R_3} + \frac{V_2}{R_2} + \frac{V_1}{R_1} + \frac{V_0}{R_0}$$

$$= \frac{V_5}{R} + \frac{V_4}{2R} + \frac{V_3}{4R} + \frac{V_2}{8R} + \frac{V_1}{16R} + \frac{V_0}{32R}$$

Since the voltages $V_5$ through $V_0$ are either 0 or $V_{REF}$ volts depending upon the bit value, so it is customary to take common voltage $V_{REF}$ and bits are kept in place of voltages. So $V_5$ is replaced by $V_{REF}.b_5$, $V_4$ by $V_{REF}.b_4$ and so on; the bits $b_5$, b4, b3 etc will be 0 or 1. The current I may, therefore, be represented as follows:

$$I = \frac{V_{REF}}{32R}\left[32b_5 + 16b_4 + 8b_3 + 4b_2 + 2b_1 + b_0\right]$$
$$= \frac{V_{REF}}{2^5.R}\left[2^0 b_0 + 2^1 b_1 + 2^2 b_2 + 2^3 b_3 + 2^4 b_4 + 2^5 b_5\right]$$

This is the equation of current *I* for 6 input bits. The general equation of current *I* for *n* input bits is given by:

$$I = \frac{V_{REF}}{2^{n-1}.R}\left[2^0 b_0 + 2^1 b_1 + 2^2 b_2 + 2^3 b_3 + 2^4 b_4 ..... + 2^{n-1} b_{n-1}\right] \qquad ...(13.3)$$

The voltage at the output of operational amplifier will be given by:

$$V_{out} = -R_f .I$$

The resistor $R_f$ is the feed back resistance in the operational amplifier. The output voltage of the operational amplifier is proportional to input binary data.

The switches connected in figure 13.3 can be replaced by the electronic switches (transistorized) as shown in figure 13.4. When the bit is at logic 1, the corresponding transistor conducts and the current flows through the collector resistor as required; and when the bit is at logic 0 the transistor goes into cutoff and no collector current flows.

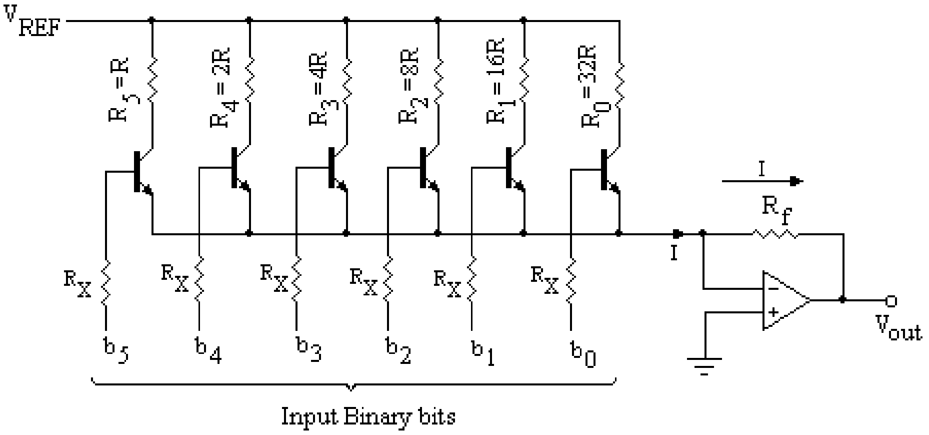


**Fig. 13.4**

This D/A converter is economical and simple method to design but suffers the following serious drawbacks:

1. The network in this D/A converter is constructed using the precession resistors and resistors have different values. So it is difficult in practice to choose the resistors with accuracy and stability.
2. When the number of bits in the network is large, then the current from the source will be large enough. The current in the MSB branch (resistor) will be much larger than LSB branch. In a 10 bit D/A converter, the current in MSB branch will be 512 times larger than the MSB branch.

### 13.1.2 Binary Ladder D/A Converter

A more commonly used D/A converter is a binary ladder D/A converter, which removes the drawbacks discussed in resistive divider D/A converter. This type of D/A converter contains an R-2R ladder network. The R-2R resistive ladder network will now be discussed, which gives the output a weighted sum of digital inputs. Such a ladder network for 4-bit input is shown in figure 13.5. This network is constructed having only two resistor values i.e. R and 2R. In this network $b_0$, $b_1$, $b_2$ and $b_3$ are the input binary bits and $b_0$ is the LSB and $b_3$ is MSB. Any of these bits will be at the ground potential when the corresponding bit is at logic 0 or at the reference potential ($V_{REF}$) when the input bit is at logic 1.

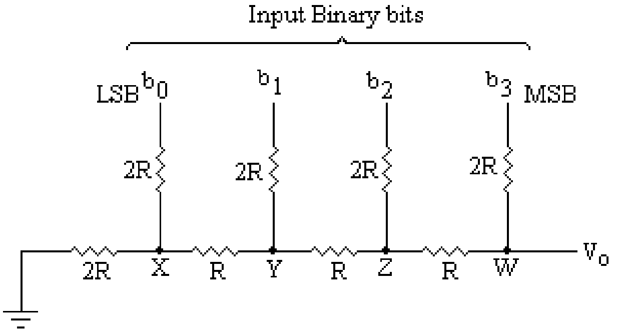
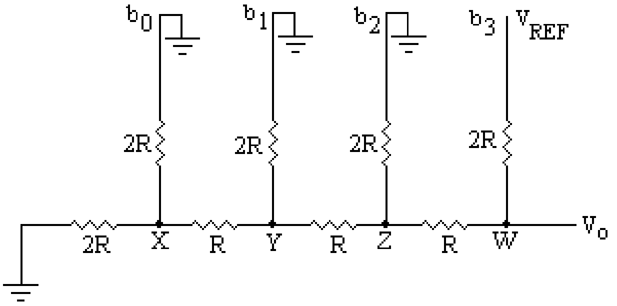
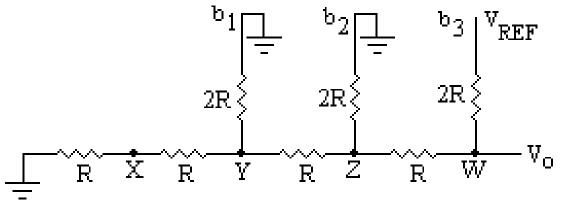


**Fig. 13.5**

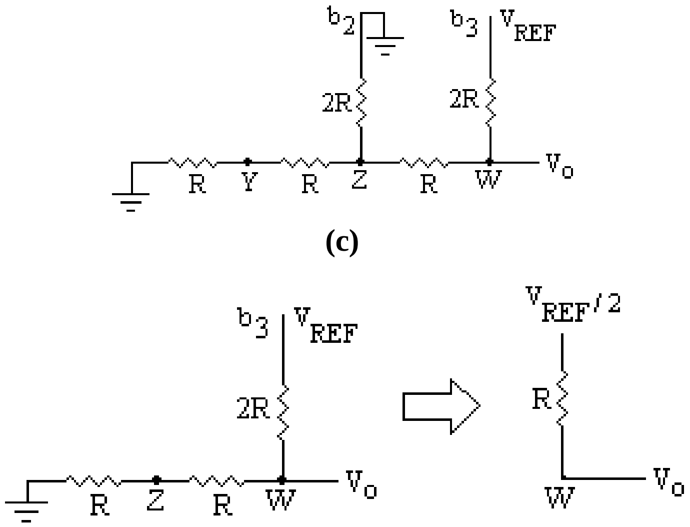
To examine the behaviour of this network, it is assumed that the bit $b_3$ is at logic 1 (or $V_{REF}$ potential) as shown in figure 13.6(a). The output voltage corresponding to MSB may be calculated as follows. The equivalent resistance at the point X is the parallel combination of two resistances each having the value of 2R. So the equivalent resistance looking at point X and ground is R as shown in figure 13.6(b). At the point Y again there is a parallel combination of two 2R resistances; the equivalent resistance looking at the point Y and ground is R as shown in figure 13.6(c). Similarly, one can find the equivalent resistance looking at the point Z and ground is R as shown in figure 13.6(d).



**(a)**



**(b)**



**(c)**

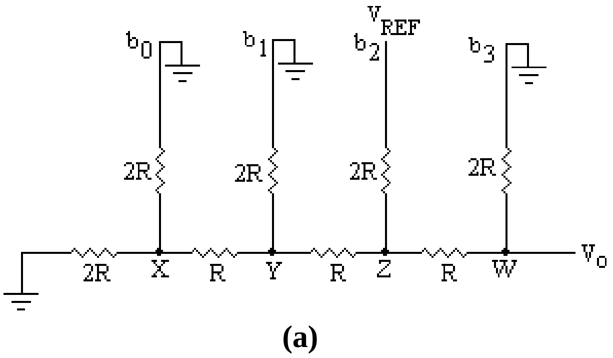
**(d)**

**Fig. 13.6**

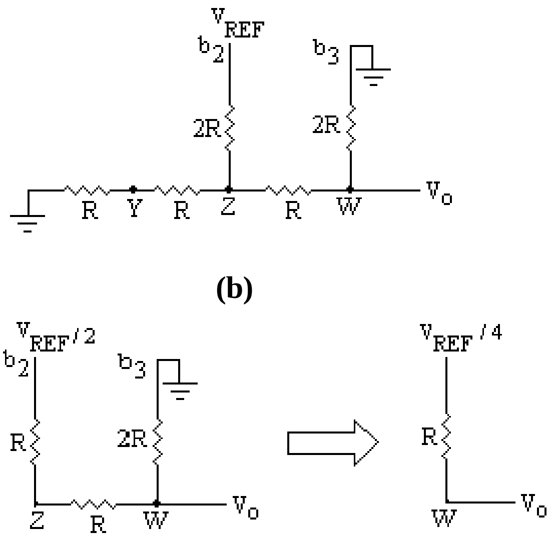
From figure 13.6(d) it is clear that the resistance looking at the point W and ground is 2R, and the resistance looking towards the bit $b_3$ is also 2R. Thus the output voltage at the point W due to bit $b_3$ (MSB) assumed at $V_{REF}$ potential is given by:

$$V_0 = \frac{V_{REF}x2R}{(2R+2R)} = \frac{V_{REF}}{2}$$

The output voltage $V_0$ due to the binary input 1000 (only MSB is high) is half of the reference voltage having Thevenis's resistance R in series with it. Similarly one can calculate the output voltage due to the binary input 0100 (i.e. second MSB); the network for this case is shown in figure 13.7(a). The resistance looking at the point Y and ground is R as shown in figure 13.7(b). The resistance between the point Z and ground is 2R. The voltage at point Z and ground is ($V_{REF}$/ 2) have a Thevenin's resistance R, as shown in figure 13.7(c).



**(a)**



**(b)**

**(c)**

**Fig. 13.7**

From this figure the output voltage $V_0$ at the point W is given by:

$$V_0 = \frac{(V_{REF}/2)x2R}{(2R+2R)} = \frac{V_{REF}}{4}$$

So the output voltage due to second MSB (or for binary input 0100) is $\frac{V_{REF}}{4}$ with Thevenin's resistance R in series with it.

It can further be shown that the output due to third MSB (for binary input 0010) is $\frac{V_{REF}}{8}$. And for LSB (0001 binary input) the output is $\frac{V_{REF}}{16}$. Each voltage source will have Thevenin's resistance R in series with the source. The total output voltage in analog form, due to all the inputs as 1 (for 1111) can easily be found by adding the outputs obtained for each bit as given below:

$$V_0 = \frac{V_{REF}}{2} + \frac{V_{REF}}{4} + \frac{V_{REF}}{8} + \frac{V_{REF}}{16}$$

It may be noted that $\frac{V_{REF}}{2}$ is the voltage due to MSB, $\frac{V_{REF}}{4}$ due to second MSB, $\frac{V_{REF}}{8}$ for third MSB and $\frac{V_{REF}}{16}$ for LSB. So to distinguish these voltages it is useful to write the bit positions along with $V_{REF}$ as given below. So if the bit is 0 the voltage corresponding to that bit will be zero otherwise the voltage as discussed above.

$$V_0 = \frac{V_{REF}xb_3}{2} + \frac{V_{REF}xb_2}{4} + \frac{V_{REF}xb_1}{8} + \frac{V_{REF}xb_0}{16}$$

$$= \frac{V_{REF}}{16}\left[8xb_3 + 4xb_2 + 2xb_1 + 1xb_0\right]$$

$$= \frac{V_{REF}}{2^4}\left[2^0 xb_0 + 2^1 xb_1 + 2^2 xb_2 + 2^3 xb_3\right] \qquad \ldots(13.4)$$

The equation (13.4) is the equation for voltage at the output of 4 bit binary ladder network. A general equation for the output of *n*-bit binary data can be given as follows:

$$V_0 = \frac{V_{REF}}{2^n}\left[2^0 xb_0 + 2^1 xb_1 + 2^2 xb_2 + 2^3 xb_3 + \ldots\ldots + 2^{n-1} xb_{n-1}\right] \qquad \ldots(13.5)$$

The output of this network is proportional to the input binary data. So using this R-2R ladder network, a D/A converter (called binary ladder D/A converter) can be designed as given below. The schematic diagram of 4-bit D/A converter is shown in figure 13.8. It consists of the following major parts.

(i) n switches, one for each bit applied to the input,

(ii) A binary ladder network which changes each of the digital level into equivalent binary weighted voltage or current.

(iii) A reference voltage source $V_{REF}$.

(iv) A summing amplifier that adds the currents flowing in the resistors of the network to develop a signal that is proportional to the digital input.

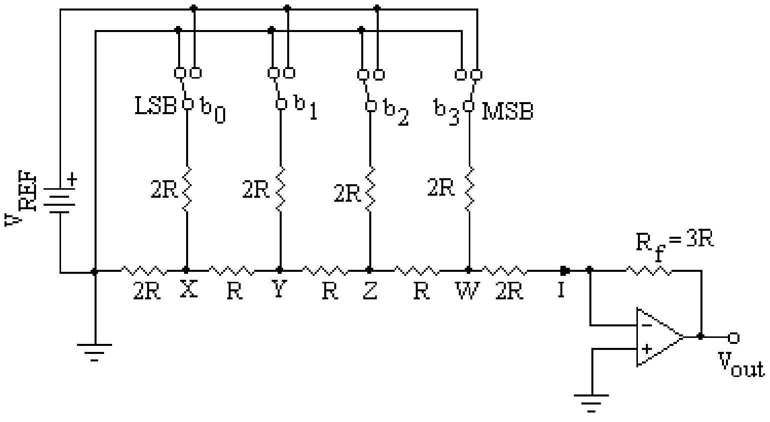


**Fig. 13.8**

The output voltage $V_{out}$ of this D/A converter due to MSB (1000 binary input) will be calculated as given below:

The voltage at the point W due to MSB is $V_{REF}$ / 2 having a Thevenin's resistance R in series with it as discussed above and is shown in figure 13.9

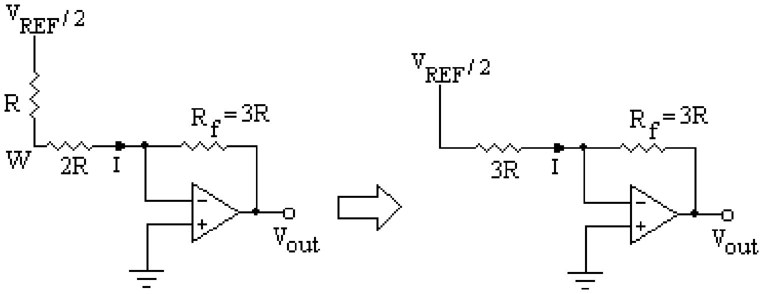


**Fig. 13.9**

From this figure, the current I is given by:

$$I = \frac{V_{REF}}{2}\left(\frac{1}{3R}\right)$$

and the output voltage V$_{out}$ is given by:

$$Vout = -I.R_f$$

$$= -\frac{V_{REF}}{2}\left(\frac{1}{3R}\right).3R = -\frac{V_{REF}}{2}$$

The output voltage is the same as calculated in equation 13.5, with the difference that it has a negative value because the operational amplifier is used in inverting configuration.

Note that the resistors in the ladder network are either R or 2R. It is the ratio of resistances matters rather than the absolute value of resistances. Further the resistors do not cover a wide range of magnitude; it is, therefore, practically possible to get the precision in the ratio of their magnitudes. The temperature coefficients of these resistances can easily match. Because of these advantages, the ladder network is widely used in D/A converters.

## 13.2 PERFORMANCE CRITERIA FOR D/A CONVERTER

The D/A converters are available in the form of ICs with different specifications for their performances. So before discussing D/A converter ICs it will be better to discuss first the characteristics of the converters specified by the manufacturers. These specifications include:

1. Resolution
2. Accuracy
3. Monotonicity
4. Settling time

1. **Resolution:** As discussed above, the analog output of D/A converter is proportional to the digital input (binary data), so a perfect staircase is obtained if there is an LSB increment. The resolution is, therefore, a measure of quality of D/A converter, which is defined as the ratio of the LSB increment to the maximum output. For an *n*-bit D/A converter the resolution is given by:

The change in output due to LSB increment for n-bit digital input (Step size)

= Full scale output / No. of steps

$$= \frac{\text{Full scale output}}{2^n - 1}$$

where $(2^n - 1)$ is the number of steps for n-bit D/A converter.

$$\text{Percentage Resolution} = \frac{\text{Full scale output} / (2^n - 1)}{\text{Full scale output}} \text{ x } 100\%$$

$$= \frac{1}{2^n - 1} \text{ x } 100\%$$

The step size for a 10 bit D/A converter, having full scale output voltage as 10 volts, is given by $= \frac{10}{2^{10} - 1} = \frac{10}{1023} = 9.8mV$

And % Resolution = 0.0978%

**2. Accuracy**: Accuracy of a D/A converter is the closeness of the output analog voltage to the expected theoretical output. In a linear variation of analog output with digital input, the relative accuracy is the maximum deviation of the D/A output compared with the linear behaviour. It is expressed as a percent of a full-scale or maximum output voltage. For example, if a converter has a full scale output of 10 V and the accuracy is $\pm 0.1\%$, then the maximum error for any output voltage is (10V)(0.001) = 10 mV. Ideally, the accuracy should be at most ± ½ of an LSB.

For an 8 bit D/A converter, one LSB is $\frac{1}{256} = 0.0039 = 0.39\%$ of full scale. The accuracy should be approximately $\pm 0.2\%$.

**3. Monotonicity**: A D/A converter is said to be monotonic if it gives an analog output voltage which increases regularly and linearly with increase in input digital signal. Such a quality of the converter is called as monotonicity. In order to demonstrate monotonicity of a D/A converter, a counter output is given as digital input to a D/A converter and the analog output is displayed on the CRO. Monotonicity then requires that the output waveform should be a perfect staircase waveform with steps equally spaced and of same magnitude. If the steps are missing or have varying magnitude, the D/A converter is defective.

**4. Settling Time**: After the application of digital input to a D/A converter, it takes about few nanoseconds to microseconds to produce the correct output. So the settling time is defined as the time the converter takes to give an output to settle within ± ½ LSB of its final value. For example, if a D/A converter has a resolution of 10mV, the settling time is the measure of the time the converter takes to settle with in $\pm 5$mV of its final value. Figure 13.10 illustrates the settling time in a D/A converter. The settling time is important because it places a limit on how fast one can change the digital input. The settling time depends on the stray capacitance, saturation delay time, and other factors.

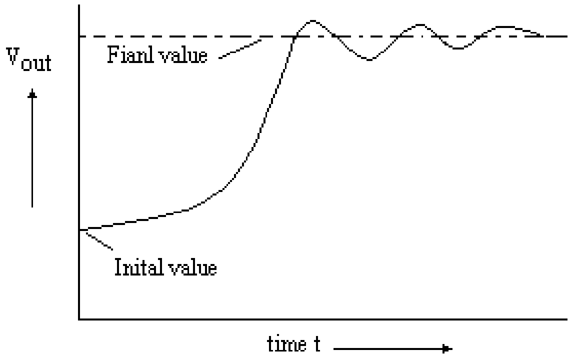


**Fig. 13.10**

### 13.3 D/A CONVERTER IC 0808

There are many commercially available D/A converter ICs. The IC 0808 is the most popular, inexpensive and widely used 8 bit D/A converter. It contains a reference current source, an R-2R binary ladder network and 8 transistor switches to steer the binary currents to the network. Figure 13.11 shows the pin configuration of this D/A converter IC 0808.

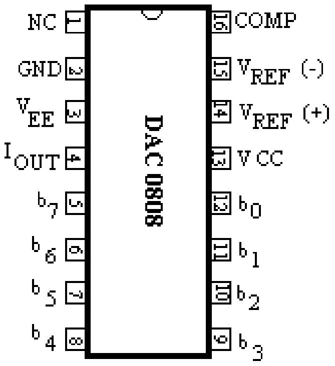


**Fig. 13.11**

In this IC, pins 5 through 12 are the 8 bit input data, so should be connected to input data bits. Pin 15 is to be connected to ground through a resistance. Pin 13 is to be connected to +5 volt supply. Pin 3 ($V_{EE}$) is to be connected to – 15 volts. Pin 4 is the output current of the ladder network should be connected to the operational amplifier. Pin 2 is the ground pin. The pin 16 is the frequency compensation pin, a capacitor between pin 16 and 3 is to be connected for this purpose.

A circuit diagram to get the analog output voltage corresponding to 8 bit digital input is shown in figure 13.12. A +5 V supply sets up a reference current of 2mA for the ladder. The output current $I_{out}$ drives the operational amplifier to give final output between 0 and 2 volts (approximately) for the 8 bit digital input.

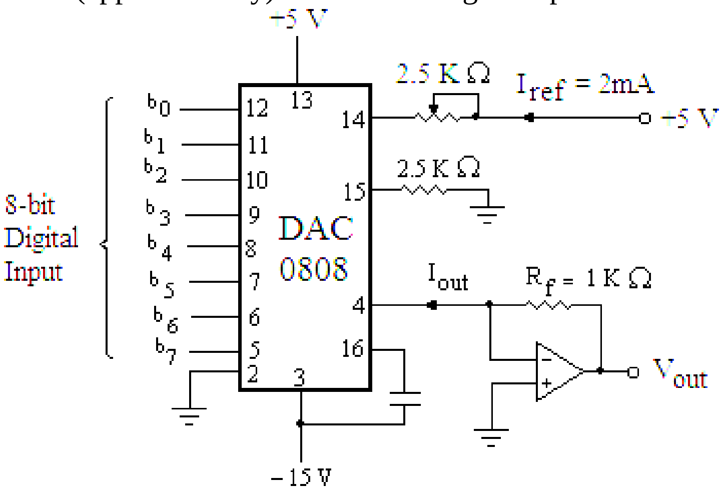


**Fig. 13.12**

There are many other commercially available D/A converter as given below:

DAC 0800 – A monolithic 8-bit high speed current output DAC.

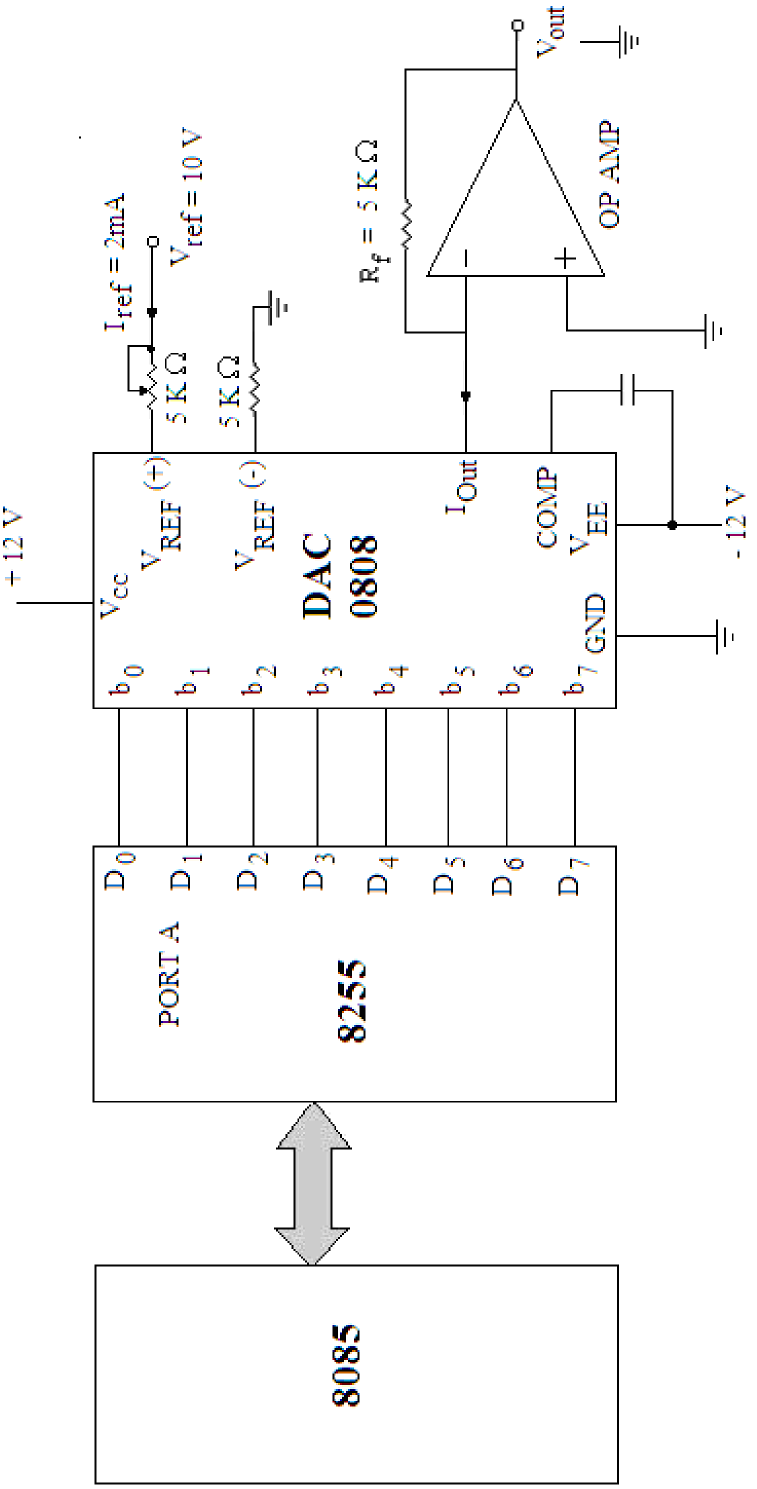
DAC 0806 and DAC 0807 – 8 bit monolithic D/A converters.

DAC 1000 and DAC 1008 – 10 bit microprocessor compatible advanced CMOS D/A converters.

DAC 1202 and DAC 1203 – Three-digit (BCD) D/A converter.

## 13.4 INTERFACING OF D/A CONVERTER

For the interfacing of 0808 D/A converter with the microprocessor 8085A, the circuit shown in figure 13.12 may be connected to the system through the 8255 PPI (Programmable Peripheral Interface as shown in figure 13.13.


I_ref = 2mA
V_ref = 10 V
5 K Ω
5 K Ω
R_f = 5 K Ω
V_out
OP AMP
-
+
+ 12 V
V_CC
V_REF (+)
V_REF (-)
DAC
0808
I_Out
COMP
V_EE
- 12 V
GND
b_0
b_1
b_2
b_3
b_4
b_5
b_6
b_7
D_0
D_1
D_2
D_3
D_4
D_5
D_6
D_7
PORT A
8255
8085

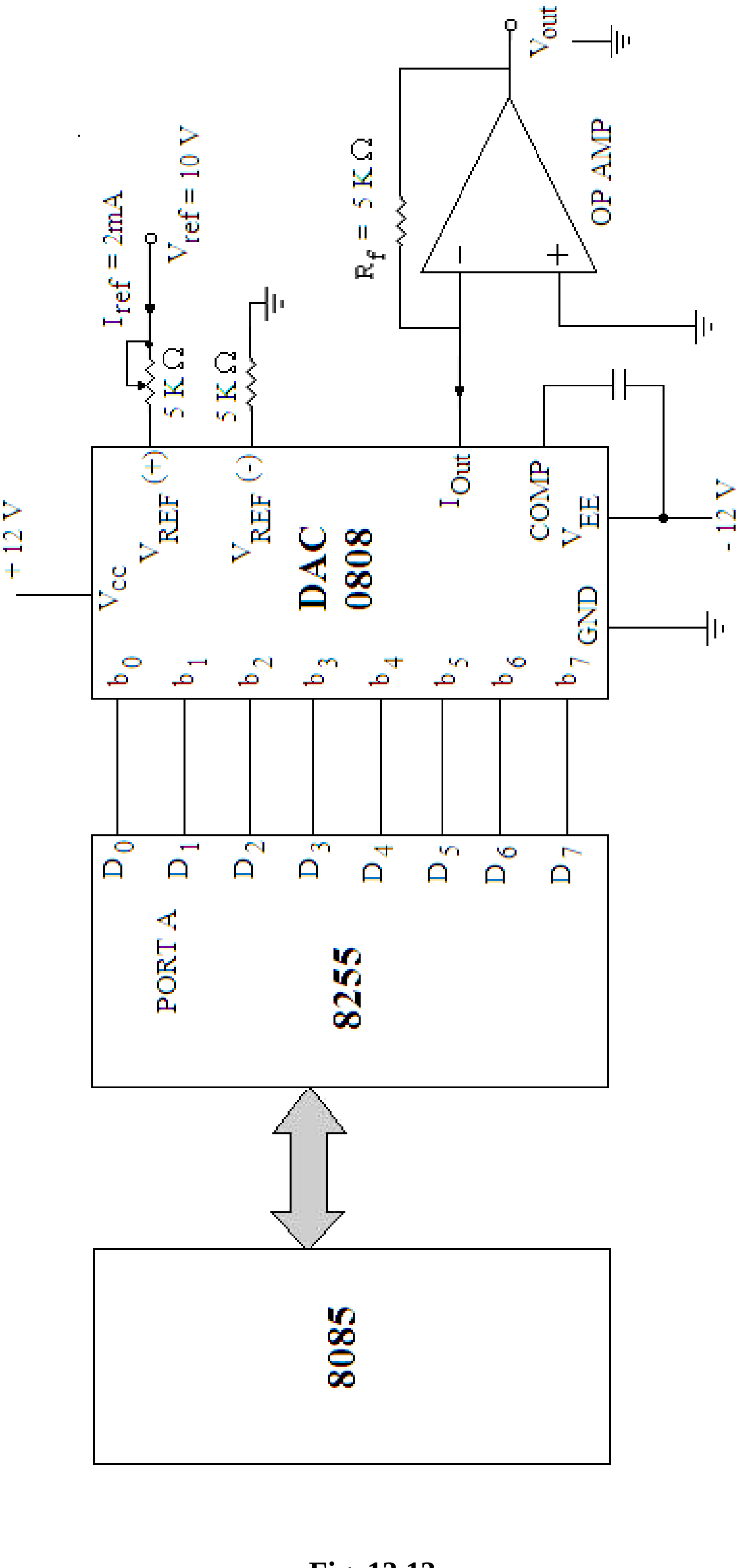

V_out
OP AMP
R_f = 5 KΩ
I_ref = 2mA
V_ref = 10 V
5 KΩ
5 KΩ
+ 12 V
V_cc
V_REF (+)
V_REF (-)
DAC
0808
I_Out
COMP
V_EE
- 12 V
GND
b_0
b_1
b_2
b_3
b_4
b_5
b_6
b_7
D_0
D_1
D_2
D_3
D_4
D_5
D_6
D_7
PORT A
8255
8085

**Fig. 13.13**

In this circuit feedback resistor $R_f$ is considered as 5 KΩ and the reference voltage is taken as $V_{REF}$ = 10 V, so that we get the output current $I_{Out}$ as:

$$I_{Out} = \frac{I_{REF}}{2^n}\left[2^0 xb_0 + 2^1 xb_1 + 2^2 xb_2 + 2^3 xb_3 + ..... + 2^{n-1} xb_{n-1}\right]$$

Here the value of *n* = 8 as it is 8 bit D/A converter.

So $$I_{Out} = \frac{I_{REF}}{256}\left[2^0 xb_0 + 2^1 xb_1 + 2^2 xb_2 + 2^3 xb_3 + ..... + 2^{n-1} xb_{n-1}\right]$$

and $$I_{REF} = \frac{V_{REF}}{R_{REF}} = \frac{10V}{5K\Omega} = 2mA$$

and $$V_{OUT} = \frac{2mA}{256}\left[2^0 xb_0 + 2^1 xb_1 + 2^2 xb_2 + 2^3 xb_3 + ..... + 2^{n-1} xb_{n-1}\right] xR_f$$

$$= \frac{10}{256}\left[2^0 xb_0 + 2^1 xb_1 + 2^2 xb_2 + 2^3 xb_3 + ..... + 2^{n-1} xb_{n-1}\right]$$

If all the input bits ($b_0 - b_7$) are 1 (FF H), then the output voltage (known as full scale output voltage) is given by:

$$V_{OUT} = \frac{255}{256} x10 \approx 10V$$

The output voltage corresponding to the input 1000 0000 (80 H) is given by:

$$V_{OUT} = \frac{128}{256} x10 = 5V$$

So we have the linear output corresponding to the binary inputs i.e.

| Digital Input | Output Voltage |
|---|---|
| 00 H | 0 V |
| 80 H | 5 V |
| FF H | 10 V |

These input output levels may be verified, if the following three programs are executed and the voltages at the outputs in the three cases are measured.

Program 1:

```
MVI A, 80 H        ; 8255 is initialized with all the ports as
                     output port.
OUT  03 H          ; Control word is written in control word
                     register.
MVI A, 00 H        ; Get A = 00 H, so that 00 H is applied to the
                     input of D/A converter.
OUT 00 H           ; 00 H is available at the Port A of 8255 so
                     that it is applied to the input of D/A
                     converter.
HLT                ; Stop processing.
```

After execution of this program, if the voltage at the output of the circuit of D/A converter is measured we get 0 V. It verifies that 00 H is applied at the input of the converter, we get 0 V.

Program 2:

```
MVI A, 80 H        ; 8255 is initialized with all the ports as
                     output port.
```

```
OUT  03 H           ; Control word is written in control word
                      register.
MVI A, 80 H         ; Get A = 80 H, so that 80 H is applied to the
                      input of D/A converter.
OUT 00 H            ; 80 H is available at the Port A of 8255 so
                      that it is applied to the input of D/A
                      converter.
HLT                 ; Stop processing.
```

After execution of this program, if the voltage at the output of the circuit of D/A converter is measured we get 5 V. It verifies that 80 H is applied at the input of the converter, we get 5 V.

Program 3:

```
MVI A, 80 H         ; 8255 is initialized with all the ports as
                      output port.
OUT  03 H           ; Control word is written in control word
                      register.
MVI A, FF H           ; Get A = FF H, so that FF H is applied to
                      the input of D/A converter.
OUT 00 H            ; FF H is available at the Port A of 8255 so
                      that it is applied to the input of D/A
                      converter.
HLT                 ; Stop processing.
```

After execution of this program, if the voltage at the output of the circuit of D/A converter is measured we get 10 V. It verifies that FF H is applied at the input of the converter, we get 10 V.

**Example 13.1.** *The input bits of the D/A converter is connected to the output pins of Port A of 8255, which is already connected with the microprocessor (ref. fig.13.13). Write a program to generate stair case voltage with ten steps. It should have the constant pulse duration.*

**Solution.** There should be 10 steps for the stair case voltage to be generated with the D/A converter. So the height of the stair case should be approximately 1 volt. Since FF H gives 10 volts, so 19 H should be decreased each time in 10 go.

When 19 H is subtracted from FF H in the first go we get E6 H. The output voltage corresponding to E6 H is given by:

$$V_{Out} = \frac{230}{256} x10 = 8.984 \text{ volts}$$

Therefore, the subtraction of 19 H from FF H gives a decrement of 1 volt at the output. The program for the generation of stair voltage is given as:

**PROGRAM**

```
Label       Mnemonics   Operand      Comments
            MVI  A,     80 H         ; Initialize 8255-I to work all the
                                       ports as output ports.
            OUT         03 H         ; Write the control word (80 H) in
                                       the control word register of 8255-I.
START       MVI  A,     FF H         ; Load FF H to accumulator.
```

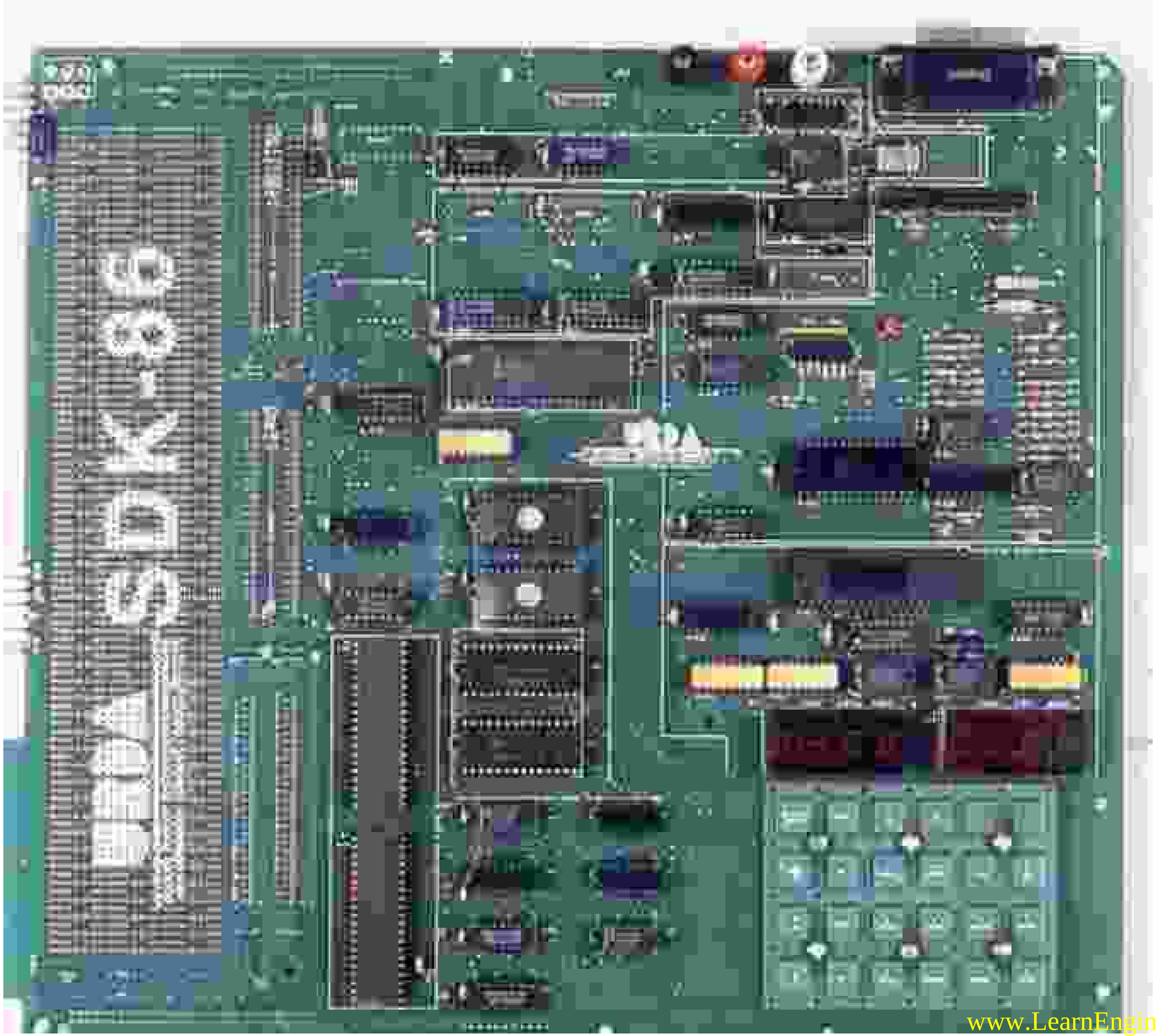
```
          OUT         00 H        ; Send FF H (10 V) to PA0.
          CALL        DELAY       ; Jump to delay subroutine to
                                    introduce a delay of constant pulse
                                    width.
          MVI  B,     00 H        ; Use B-register as counter for 10
                                    steps.
REPEAT    SBI         19 H        ; Subtract 19 H to calculate next
                                    weight for the output of D/A
                                    converter.
          OUT         00 H        ; Send the data to the output.
          INR  B                  ; Increment B-register.
          CPI         0A H        ; If 10 steps complete then
          JZ          END         ; Jump to repeat the process.
          PUSH  PSW               ; Save PSW
          PUSH  B                 ; Save B-C register pair.
          CALL        DELAY       ; Jump to delay subroutine to
                                    introduce a delay of constant pulse
                                    width.
          POP  B                  ; Restore the contents of B-C pair.
          POP  PSW                ; Restore the PSW.
          JUMP        REPEAT      ; Jump to repeat to output for the
                                    next weight.
END       MVI  A,     00 H        ; Store 00 H to the accumulator.
          OUT         00 H        ; Outputs for 00 H.
          CALL        DELAY       ; Jump to delay subroutine to
                                    introduce a delay of constant pulse
                                    width.
          JMP         START       ; Repeat for next cycle.
```

**SUBROUTINE PROGRAM:**

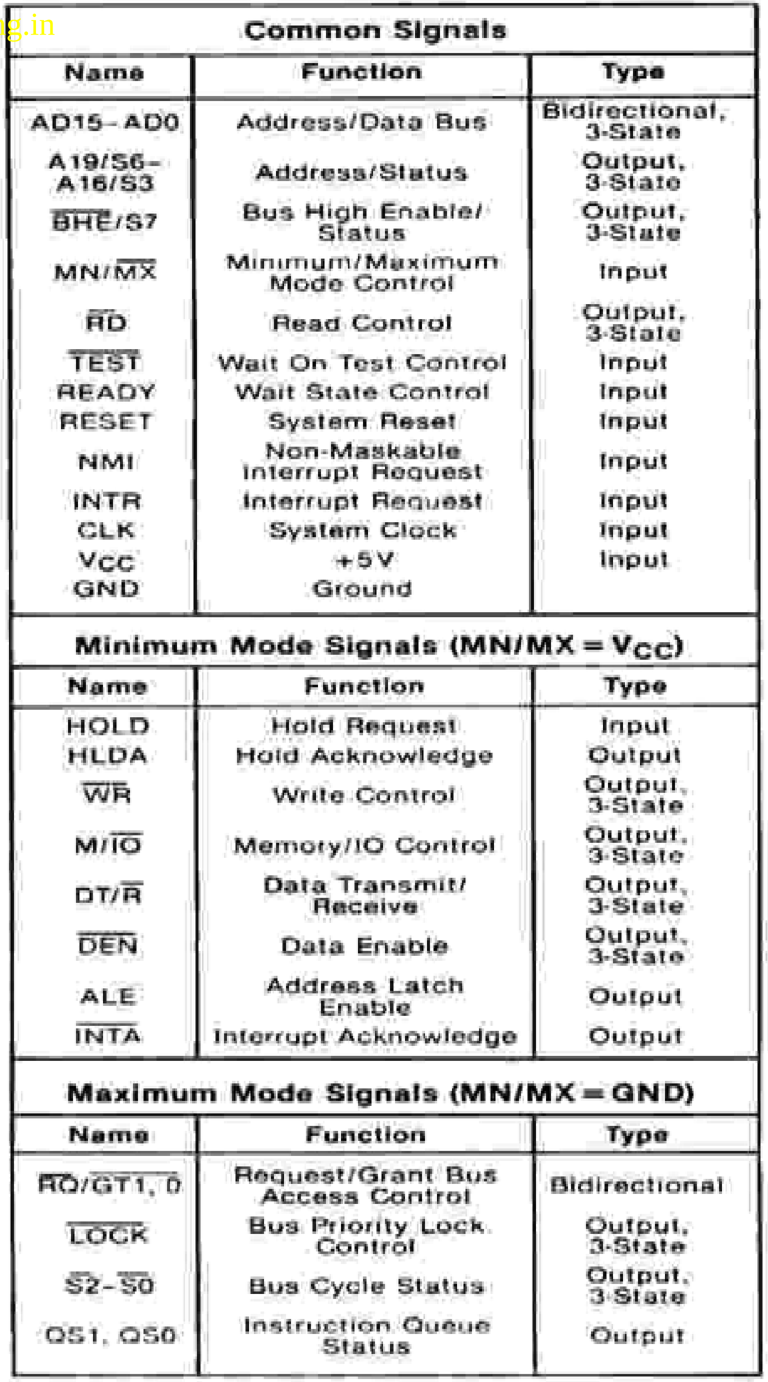
```
   Label   Mnemonics   Operand     Comments
   DELAY   LXI   D,    0020 H      ; Loads DE register pair with a 16-
                                     bit number.
   LOOP1   DCX  D                  ; Decrements DE register pair by 1.
           MOV  A,     E           ; Moves the contents of E register to
                                     accumulator.
           ORA  D                  ; ORing of the contents of D and E
                                     registers are performed to set the
                                     zero flag.
           JNZ         LOOP1       ; If result is not zero than jump to
                                     LOOP1.
           RET                     ; Go back to main program.
```

In the subroutine program, DE register pair may be loaded with any other 16-bit number to change the pulse width. The exact time of the pulse width may be calculated as discussed in the delay programs. After the execution of this program, the output may be seen on the CRO. The output will be stair case wave.

# 8086 MICROPROCESSOR



# Pinouts



## Common Signals

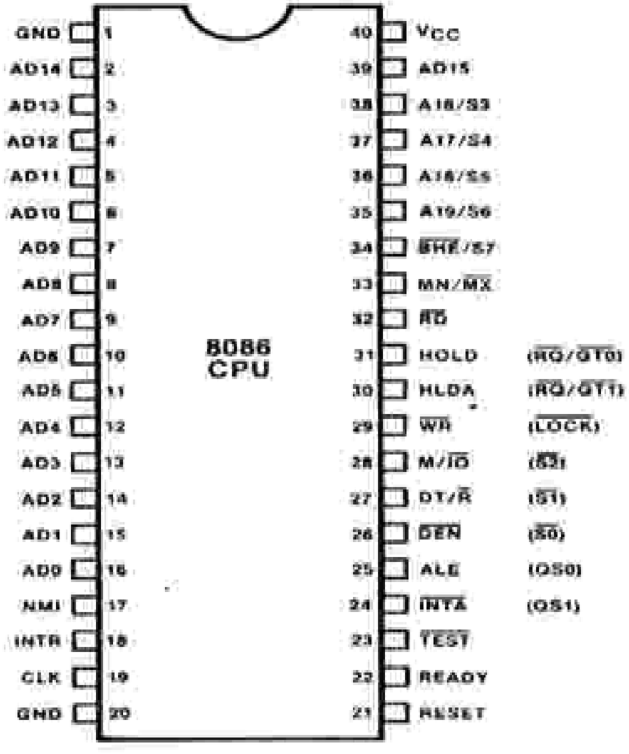
| Name | Function | Type |
|---|---|---|
| AD15–AD0 | Address/Data Bus | Bidirectional, 3-State |
| A19/S6–A16/S3 | Address/Status | Output, 3-State |
| $\overline{\text{BHE}}$/S7 | Bus High Enable/ Status | Output, 3-State |
| MN/$\overline{\text{MX}}$ | Minimum/Maximum Mode Control | Input |
| $\overline{\text{RD}}$ | Read Control | Output, 3-State |
| $\overline{\text{TEST}}$ | Wait On Test Control | Input |
| READY | Wait State Control | Input |
| RESET | System Reset | Input |
| NMI | Non-Maskable Interrupt Request | Input |
| INTR | Interrupt Request | Input |
| CLK | System Clock | Input |
| $V_{CC}$ | +5 V | Input |
| GND | Ground | |

## Minimum Mode Signals (MN/$\overline{\text{MX}}$ = $V_{CC}$)

| Name | Function | Type |
|---|---|---|
| HOLD | Hold Request | Input |
| HLDA | Hold Acknowledge | Output |
| $\overline{\text{WR}}$ | Write Control | Output, 3-State |
| M/$\overline{\text{IO}}$ | Memory/IO Control | Output, 3-State |
| DT/$\overline{\text{R}}$ | Data Transmit/ Receive | Output, 3-State |
| $\overline{\text{DEN}}$ | Data Enable | Output, 3-State |
| ALE | Address Latch Enable | Output |
| $\overline{\text{INTA}}$ | Interrupt Acknowledge | Output |

## Maximum Mode Signals (MN/$\overline{\text{MX}}$ = GND)

| Name | Function | Type |
|---|---|---|
| $\overline{\text{RQ}}/\overline{\text{GT1}}$, 0 | Request/Grant Bus Access Control | Bidirectional |
| $\overline{\text{LOCK}}$ | Bus Priority Lock Control | Output, 3-State |
| $\overline{\text{S2}}$–$\overline{\text{S0}}$ | Bus Cycle Status | Output, 3-State |
| QS1, QS0 | Instruction Queue Status | Output |



MAXIMUM MODE PIN FUNCTIONS (e.g., $\overline{\text{LOCK}}$) ARE SHOWN IN PARENTHESES

# 8086 Pins

**The 8086 comes in a 40 pin package which means that some pins have more than one use or are multiplexed. The packaging technology of time limited the number of pin that could be used.**

**In particular, the address lines 0 - 15 are multiplexed with data lines 0-15, address lines 16-19 are multiplexed with status lines. These pins are**

**AD0 - AD15, A16/S3 - A19/S6**

***The 8086 has one other pin that is multiplexed and this is BHE'/S7. BHE stands for Byte High Enable. This is an active low signal that is asserted when there is data on the upper half of the data bus.***

**The 8086 has two modes of operation that changes the function of some pins. The SDK-86 uses the 8086 in the minimum mode with the MN/MX' pin tied to 5 volts. This is a simple single processor mode. The IBM PC uses an 8088 in the maximum mode with the MN/MX" pin tied to ground. This is the mode required for a coprocessor like the 8087.**

# 8086 Pins

In the minimum mode the following pins are available.

HOLD When this pin is high, another master is requesting control of the local bus, e.g., a DMA controller.

HLDA HOLD Acknowledge: the 8086 signals that it is going to float the local bus.

WR’ Write: the processor is performing a write memory or I/O operation.

M/IO’ Memory or I/O operation.

DT/R’ Data Transmit or Receive.

DEN’ Data Enable: data is on the multiplexed address/data pins.

ALE Address Latch Enable: the address is on the address/data pins. This signal is used to capture the address in latches to establish the address bus.

INTA’ Interrupt acknowledge: acknowledges external interrupt requests.

# 8086 Pins

The following are pins are available in both minimum and maximum modes.

VCC + 5 volt power supply pin.

GND Ground

RD’ READ: the processor is performing a read memory or I/O operation.

READY Acknowledgement from wait-state logic that the data transfer will be completed.

RESET Stops processor and restarts execution from FFFF:0. Must be high for 4 clocks. CS = 0FFFFH, IP = DS = SS = ES = Flags = 0000H, no other registers are affected.

TEST’ The WAIT instruction waits for this pin to go low. Used with 8087.

NMI Non Maskable Interrupt: transition from low to high causes an interrupt. Used for emergencies such as power failure.

INTR Interrupt request: masked by the IF bit in FLAG register.

CLK Clock: 33% duty cycle, i.e., high 1/3 the time.

# 8086 Features

- **16-bit Arithmetic Logic Unit**
- **16-bit data bus (8088 has 8-bit data bus)**
- **20-bit address bus - $2^{20}$ = 1,048,576 = 1 meg**

**The address refers to a byte in memory. In the 8088, these bytes come in on the 8-bit data bus. In the 8086, bytes at even addresses come in on the low half of the data bus (bits 0-7) and bytes at odd addresses come in on the upper half of the data bus (bits 8-15).**

**The 8086 can read a 16-bit word at an even address in one operation and at an odd address in two operations. The 8088 needs two operations in either case.**

**The least significant byte of a word on an 8086 family microprocessor is at the lower address.**

# 8086 Architecture

- The 8086 has two parts, the Bus Interface Unit (BIU) and the Execution Unit (EU).

- The BIU fetches instructions, reads and writes data, and computes the 20-bit address.

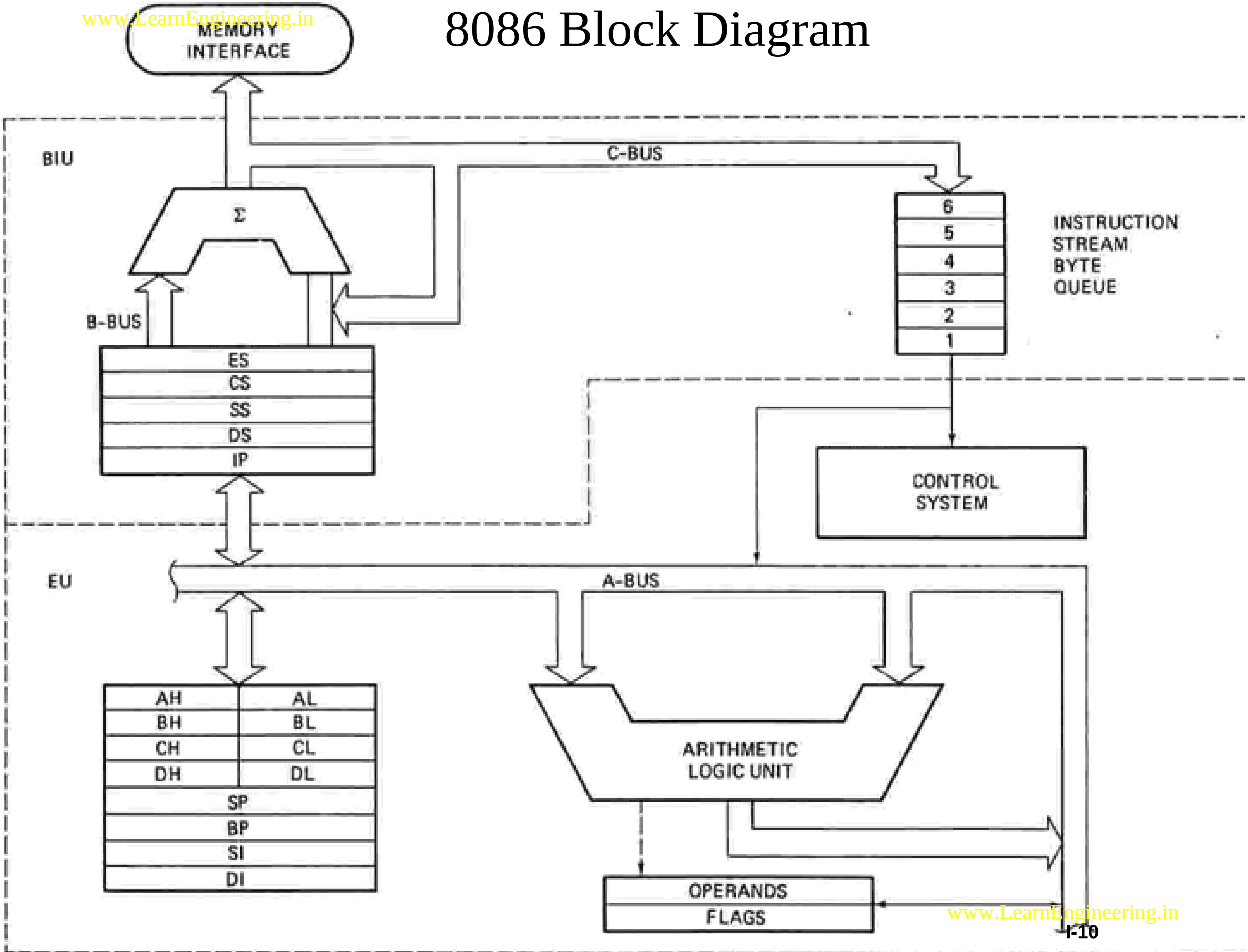
- The EU decodes and executes the instructions using the 16-bit ALU.

- The BIU contains the following registers:

IP - the Instruction Pointer
CS - the Code Segment Register
DS - the Data Segment Register
SS - the Stack Segment Register
ES - the Extra Segment Register

The BIU fetches instructions using the CS and IP, written CS:IP, to contruct the 20-bit address. Data is fetched using a segment register (usually the DS) and an effective address (EA) computed by the EU depending on the addressing mode.

# 8086 Block Diagram



# 8086 Architecture

**The EU contains the following 16-bit registers:**

**AX - the Accumulator**
**BX - the Base Register**
**CX - the Count Register**
**DX - the Data Register**

**SP - the Stack Pointer \ defaults to stack segment**
**BP - the Base Pointer /**
**SI - the Source Index Register**
**DI - the Destination Register**

**These are referred to as general-purpose registers, although, as seen by their names, they often have a special-purpose use for some instructions.**

**The AX, BX, CX, and DX registers can be considers as two 8-bit registers, a High byte and a Low byte. This allows byte operations and compatibility with the previous generation of 8-bit processors, the 8080 and 8085. 8085 source code could be translated in 8086 code and assembled. The 8-bit registers are:**

**AX --> AH,AL**
**BX --> BH,BL**
**CX --> CH,CL**
**DX --> DH,DL**

# Flag Register

- ❑ Flag register contains information reflecting the current status of a microprocessor. It also contains information which controls the operation of the microprocessor.

| 15 | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| — | NT | IOPL | OF | DF | IF | TF | SF | ZF | — | AF | — | PF | — | CF |

- ➢ Control Flags

| | |
|---|---|
| IF: | Interrupt enable flag |
| DF: | Direction flag |
| TF: | Trap flag |

- ➢ Status Flags

| | |
|---|---|
| CF: | Carry flag |
| PF: | Parity flag |
| AF: | Auxiliary carry flag |
| ZF: | Zero flag |
| SF: | Sign flag |
| OF: | Overflow flag |
| NT: | Nested task flag |
| IOPL: | Input/output privilege level |

# Flags Commonly Tested During the Execution of Instructions

❑ There are five flag bits that are commonly tested during the execution of instructions

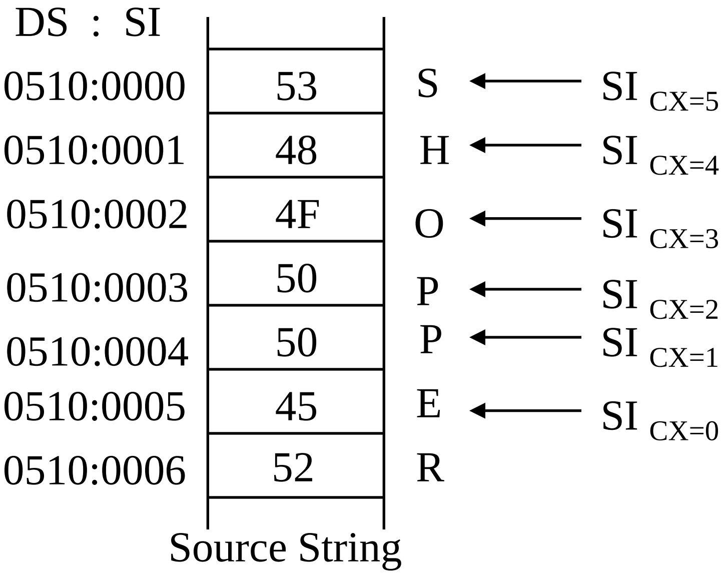
- Sign Flag (Bit 7), SF: 0 for positive number and 1 for negative number
- Zero Flag (Bit 6), ZF: If the ALU output is 0, this bit is set (1); otherwise, it is 0
- Carry Flag (Bit 0), CF: It contains the carry generated during the execution
- Auxiliary Carry, AF: (Bit 4) Depending on the width of ALU inputs, this flag bit contains the carry generated at bit 3 (or, 7, 15) of the 8088 ALU
- Parity Flag (bit2), PF: It is set (1) if the output of the ALU has even number of ones; otherwise it is zero

# Direction Flag

- Direction Flag (DF) is used to control the way SI and DI are adjusted during the execution of a string instruction
  - DF=0, SI and DI will auto-increment during the execution; otherwise, SI and DI auto-decrement
  - Instruction to set DF: **STD**; Instruction to clear DF: **CLD**
  - Example:

```
CLD
MOV CX, 5
REP MOVSB
```

At the beginning of execution,
DS=0510H and SI=0000H



| DS : SI | | | |
|---|---|---|---|
| 0510:0000 | 53 | S | $\text{SI}_{\text{CX=5}}$ |
| 0510:0001 | 48 | H | $\text{SI}_{\text{CX=4}}$ |
| 0510:0002 | 4F | O | $\text{SI}_{\text{CX=3}}$ |
| 0510:0003 | 50 | P | $\text{SI}_{\text{CX=2}}$ |
| 0510:0004 | 50 | P | $\text{SI}_{\text{CX=1}}$ |
| 0510:0005 | 45 | E | $\text{SI}_{\text{CX=0}}$ |
| 0510:0006 | 52 | R | |

Source String

# 8086 Programmer’s Model

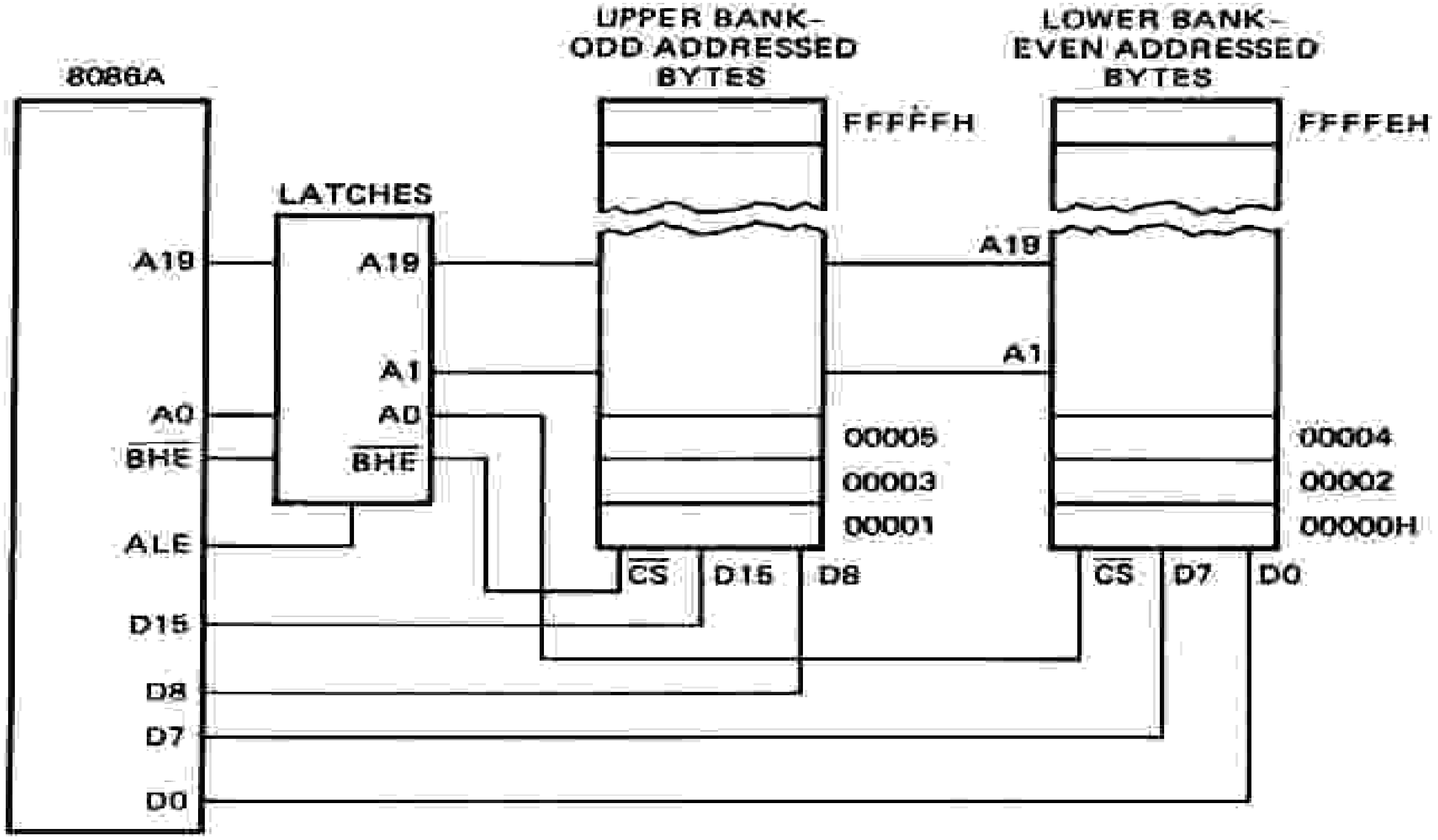
BIU registers
(20 bit adder)

| | |
|---|---|
| ES | Extra Segment |
| CS | Code Segment |
| SS | Stack Segment |
| DS | Data Segment |
| IP | Instruction Pointer |

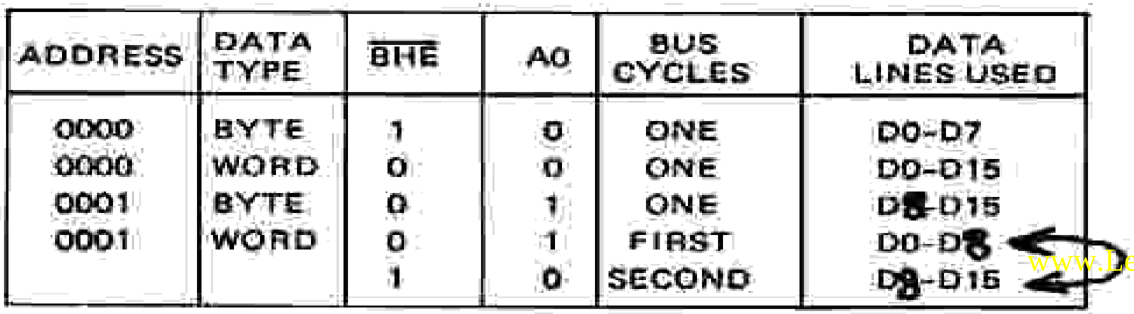
EU registers
16 bit arithmetic

| | | | |
|---|---|---|---|
| AX | AH | AL | Accumulator |
| BX | BH | BL | Base Register |
| CX | CH | CL | Count Register |
| DX | DH | DL | Data Register |
| | SP | | Stack Pointer |
| | BP | | Base Pointer |
| | SI | | Source Index Register |
| | DI | | Destination Index Register |
| | FLAGS | | |

# 8086 memory Organization



(a)



| ADDRESS | DATA TYPE | $\overline{BHE}$ | A0 | BUS CYCLES | DATA LINES USED |
|---|---|---|---|---|---|
| 0000 | BYTE | 1 | 0 | ONE | D0-D7 |
| 0000 | WORD | 0 | 0 | ONE | D0-D15 |
| 0001 | BYTE | 0 | 1 | ONE | D8-D15 |
| 0001 | WORD | 0 | 1 | FIRST | D0-D8 |
| | | 1 | 0 | SECOND | D8-D15 |

Even addresses are on the low half of the data bus (D0-D7).

Odd addresses are on the upper half of the data bus (D8-D15).

A0 = 0 when data is on the low half of the data bus.

BHE’ = 0 when data is on the upper half of the data bus.