UNIT III - INTERFACING MEMORY & IO AND APPLICATIONS OF 8086 MICROPROCESSOR

Static Memory Interfacing

Introduction : Memory is simply a device that can be used to store the information . • The semiconductor memories are extensively used because of their small size, low cost, high speed, high reliability & ease of expansion of the memory size. • It consist of mainly flip-flop & some additional circuitry such as buffers, one flip flop can hold one bit of data.

Memory fundamentals : Memory capacity . The no. of bits that a semiconductor memory chip can store is called its chip capacity. • Memory Organization: . Each memory chip contains 2N locations, where N is the no. of address pins on the chip. Each location contains M bits, where M is the no. of data pins on the chip. The entire chip will contain 2N x M bits. E.g. for 4K x 4, 212 =4096 locations, each location holding 4 bits, so N=12 & M=4.

The general procedure of static memory interfacing with 8086 as follows:

1. Arrange the available memory chips so as to obtain 16-bit data bus width. The upper 8-bit bank is called 'odd address memory bank' and the lower 8-bit bank is called 'even address memory bank'.

2. Connect available memory address lines of memory chips with those of the microprocessor and also connect the memory RD and WR inputs to the corresponding processor control signals. Connect the 16-bit data bus of the memory bank with that of the microprocessor 8086.

3. The remaining address lines of the microprocessor, BHE and Ao are used for decoding the required chip select signals for the odd and even memory banks. The CS of memory is derived from the output of the decoding circuit.

4. As a good and efficient interfacing practice, the address map of the system should be continuous as far as possible

Problem 5.1

Interface two 4K × 8 EPROMS and two 4K × 8 RAM chips with 8086. Select suitable maps.

Solution We know that, after reset, the IP and CS are initialised to form address FFFF0H. Hence, this address must lie in the EPROM. The address of RAM may be selected any where in the 1MB address space of 8086, but we will select the RAM address such that the address map of the system is continuous, as shown in Table 5.1.

Address	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_{09}	A_{08}	A_{07}	A_{06}	A_{05}	A_{04}	A_{03}	A_{02}	A_{01}	A_{00}
FFFFFH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
						EPROM						8K × 8								
FE000H	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
FDFFFH	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
						RAM						8K × 8								
FC000H	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.2 *Memory Chip Selection for Problem 5.1*

Decoder I/P → / Address/$\overline{BHE}$ →	A_2 / A_{13}	A_1 / A_0	A_0 / $\overline{BHE}$	Selection/ Comment
Word transfer on $D_0 - D_{15}$	0	0	0	Even and odd addresses in RAM
Byte transfer on $D_7 - D_0$	0	0	1	Only even address in RAM
Byte transfer on $D_8 - D_{15}$	0	1	0	Only odd address in RAM
Word transfer on $D_0 - D_{15}$	1	0	0	Even and odd addresses in ROM
Byte transfer on $D_0 - D_7$	1	0	1	Only even address in ROM
Byte transfer on $D_8 - D_{15}$	1	1	0	Only odd address in ROM

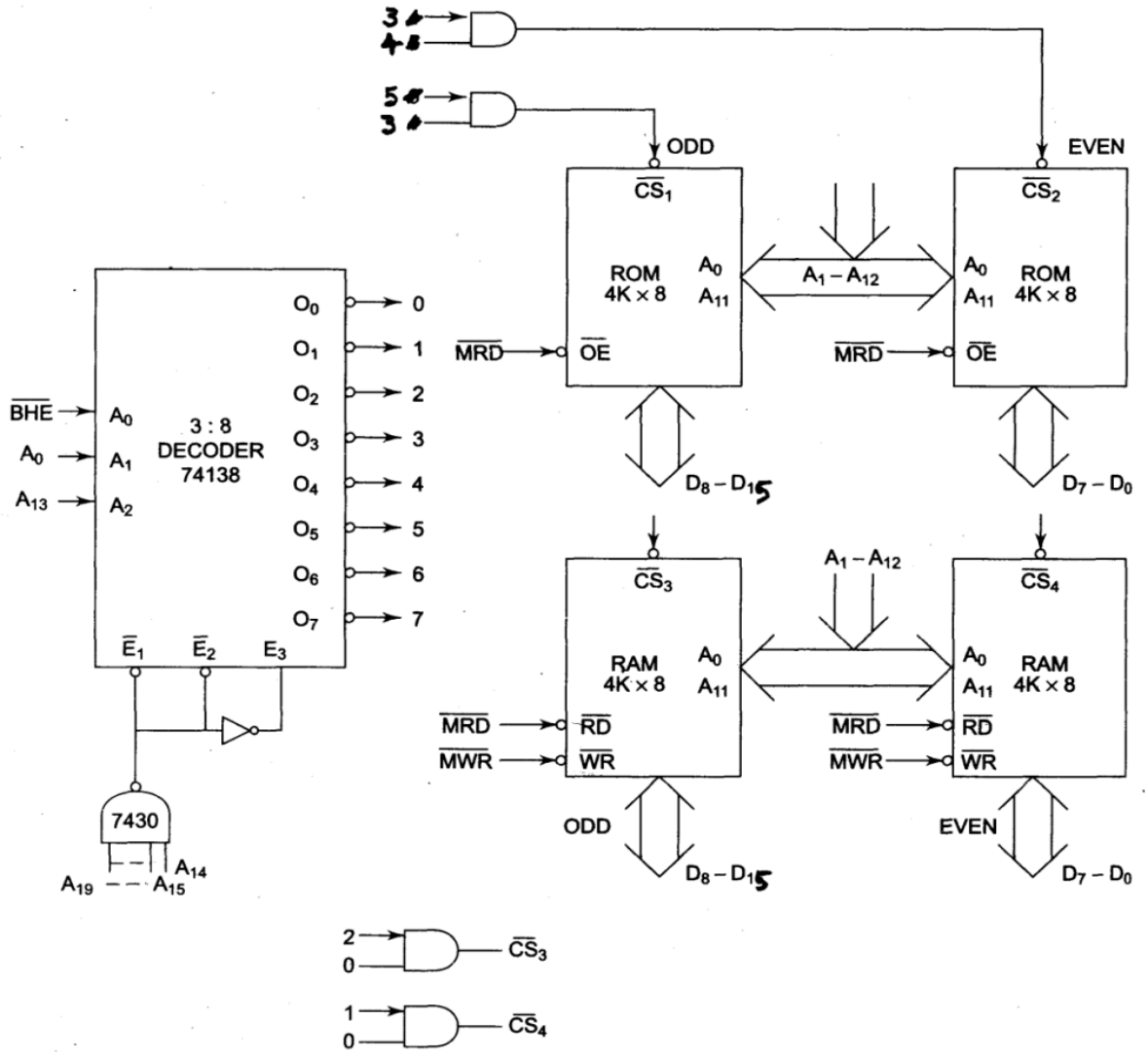


Fig. 5.1 *Interfacing Problem 5.1*

Fig shows the interfacing diagram for the memory system

The memory system in this example contains in total four 4Kx8 memory chip. The two 4Kx8 chips of RAM and ROM are arranged in parallel to obtain 16-bit data bus width. A**o** is 0, i.e. the address is even and is in RAM, then the lower RAM chip is selected indicating 8-bit transfer at an even address. If Ao is 1, i.e. the address is odd and is in RAM, the BHE goes low, the upper RAM chip is selected, further indicating that the 8-bit transfer is at an odd address. The selection of chips here takes place as shown in table 2.

Problem 5.2

Design an interface between 8086 CPU and two chips of 16K × 8 EPROM and two chips of 32K × 8 RAM. Select the starting address of EPROM suitably. The RAM address must start at 00000H.

Solution: The last address in the map of 8086 is FFFFFH. After resetting, the processor starts from FFFF0H. Hence this address must lie in the address range of EPROM. Figure 5.2 shows the interfacing diagram, and Table 5.3 shows complete map of the system.

Table 5.3 *Address Map for Problem 5.2*

Addresses	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_{09}	A_{08}	A_{07}	A_{06}	A_{05}	A_{04}	A_{03}	A_{02}	A_{01}	A_{00}
FFFFFH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
						32KB EPROM														
F8000H	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0FFFFH	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
						64KB RAM														
00000H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

It is better not to use a decoder to implement the above map because it is not continuous, i.e. there is some unused address space between the last RAM address (0FFFFH) and the first EPROM address F8000H). Hence the logic is implemented using logic gates, as shown in Fig. 5.2.

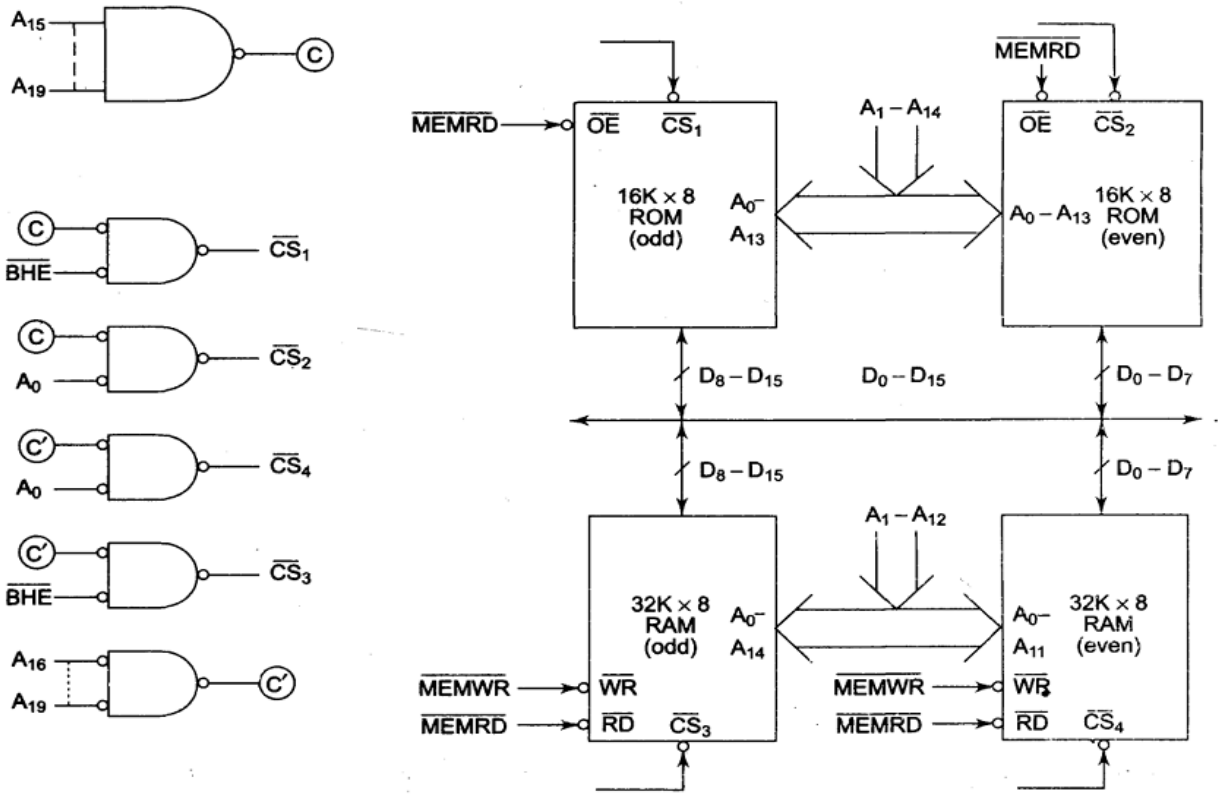


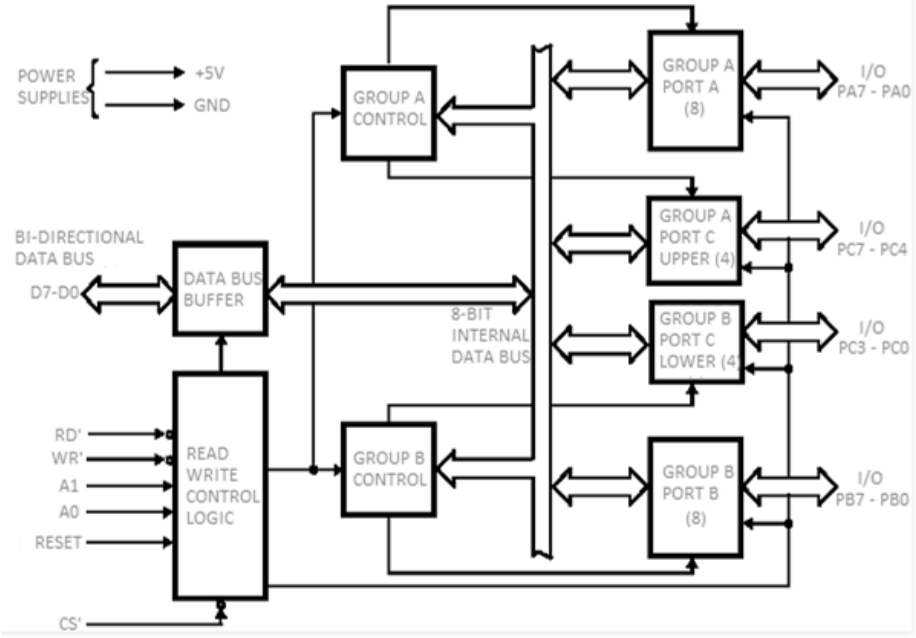
Fig. 5.2 *Interfacing Problem 5.2*

PROGRAMMABLE PERIPHERAL INTERFACE 8255

PPI 8255 is a general purpose programmable I/O device designed to interface the CPU with its outside world such as ADC, DAC, keyboard etc. We can program it according to the given condition. It can be used with almost any microprocessor.

It consists of three 8-bit bidirectional I/O ports i.e. PORT A, PORT B and PORT C. We can assign different ports as input or output functions.

Internal architecture –



It consists of 40 pins and operates in +5V regulated power supply. Port C is further divided into two 4-bit ports i.e. port C lower and port C upper and port C can work in either BSR (bit set rest) mode or in mode 0 of input-output mode of 8255. Port B can work in either mode or in mode 1 of input-output mode. Port A can work either in mode 0, mode 1 or mode 2 of input-output mode.

It has two control groups, control group A and control group B. Control group A consist of port A and port C upper. Control group B consists of port C lower and port B.

Depending upon the value if CS', A1 and A0 we can select different ports in different modes as input-output function or BSR. This is done by writing a suitable word in control register (control word D0-D7).

Pin diagram –

PA0 – PA7 – Port A contains one 8-bit output latch/buffer and one 8-bit input buffer.

PB0 – PB7 – is similar to PORT A.

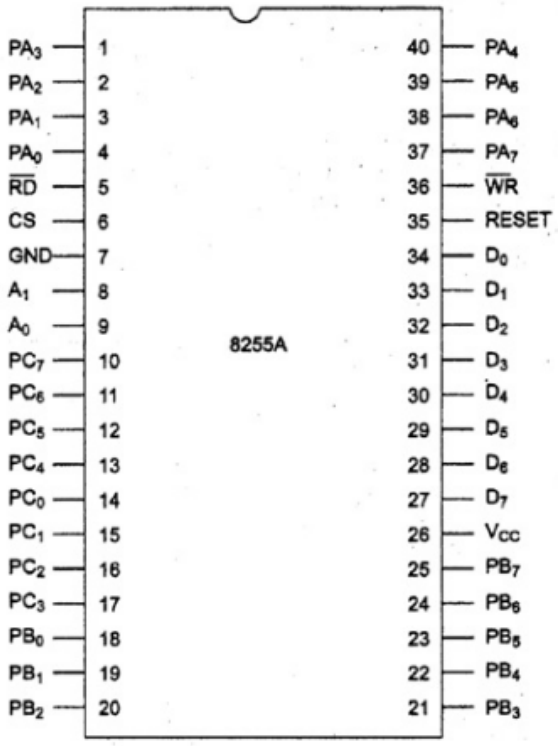


Fig. 5.17(b) *8255A Pin Configuration*

PC0 – PC7 – can be split into two parts, i.e. PORT C lower (PC0-PC3) and PORT C upper (PC7-PC4) by the control word.

- ***Data bus buffer(D0 – D7)***: It is used to connect the internal bus of 8255 with the system bus so as to establish proper interfacing between the two. The data bus buffer allows the read/write operation to be performed from/to the CPU.
- The buffer allows the passing of data from ports or control register to CPU in case of write operation and from CPU to ports or status register in case of read operation.

- **Read/ Write control logic**: This unit manages the internal operations of the system. This unit holds the ability to control the transfer of data and control or status words both internally and externally.
- Whenever there exists a need for data fetch then it accepts the address provided by the processor through the bus and immediately generates command to the 2 control groups for the particular operation.

- ***Group A and Group B control***: These two groups are handled by the CPU and functions according to the command generated by the CPU. The CPU sends control words to the group A and group B control and they in turn sends the appropriate command to their respective port.
- As we have discussed that group A has the access of the port A and higher order bits of port C. While group B controls port B with the lower order bits of port C.

- ***CS***: It stands for chip select. A low signal at this pin shows the enabling of communication between the 8255 and the processor. More specifically we can say that the data transfer operation gets enabled by an active low signal at this pin.
- ***RD*** – It is the signal used for read operation. A low signal at this pin shows that CPU is performing read operation at the ports or status word. Or we can say that 8255 is providing data or information to the CPU through data buffer.
- ***WR*** – It shows write operation. A low signal at this pin allows the CPU to perform write operation over the ports or control register of 8255 using the data bus buffer.
- ***A_0 and A_1***: These are basically used to select the desired port among all the ports of the 8255 and it do so by forming conjunction with RD and WR. It forms connection with the LSB of the address bus.

$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	A_1	A_0	Input (Read) cycle
0	1	0	0	0	Port A to Data bus
0	1	0	0	1	Port B to Data bus
0	1	0	1	0	Port C to Data bus
0	1	0	1	1	CWR to Data bus

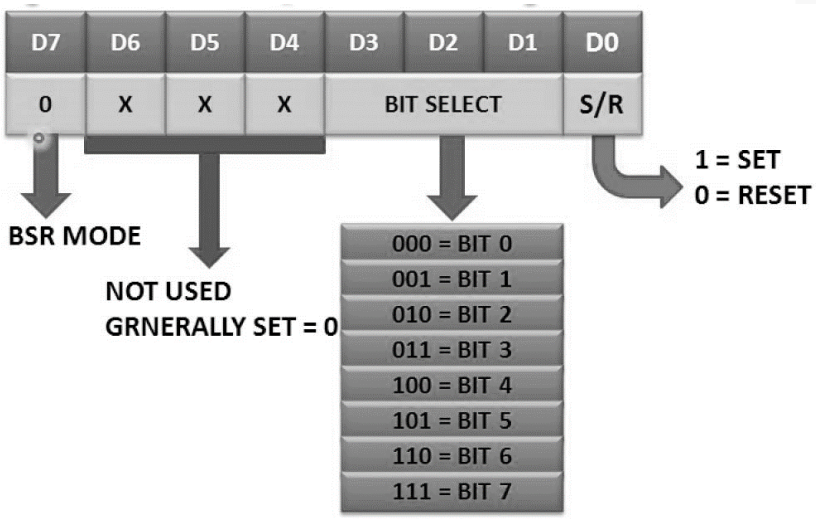
$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	A_1	A_0	Output (Write) cycle
1	0	0	0	0	Data bus to Port A
1	0	0	0	1	Data bus to Port B
1	0	0	1	0	Data bus to Port C
1	0	0	1	1	Data bus to CWR

$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	A_1	A_0	Function
X	X	1	X	X	Data bus tristated
1	1	0	X	X	Data bus tristated

MODES OF OPERATION

These are two basic modes of operation of 8255. I/O mode and Bit Set Reset mode (BSR).
• In I/O mode, the 8255 ports work as programmable I/O ports, while in BSR mode only port C (PC0-PC7) can be used to set or reset its individual port bits.
• Under the I/O mode of operation, further there are three modes of operation of 8255, so as to support different types of applications, mode 0, mode 1 and mode 2.
• ***BSR Mode***: In this mode any of the 8-bits of port C can be set or reset depending on D0 of the control word. The bit to be set or reset is selected by bit select flags D3, D2 and D1 of the CWR as given in table.

1. **Bit set reset (BSR) mode –**
   If MSB of control word (D7) is 0, PPI works in BSR mode. In this mode only port C bits are used for set or reset.

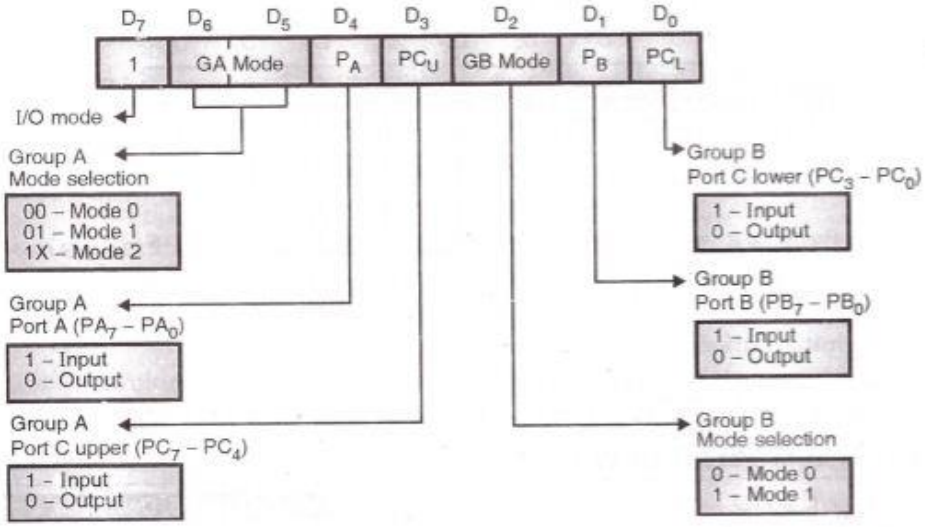


2. **Input-Outpt mode –**
   If MSB of control word (D7) is 1, PPI works in input-output mode. This is further divided into three modes:
   - **Mode 0 –** This mode is also called as basic input/output mode. This mode provides simple input and output capabilities using each of the three ports. Data can be simply read from and written to the input and output ports respectively, after appropriate initialisation.

   The salient features of this mode are as listed below:

1. Two 8-bit ports (port A and port B)and two 4-bit ports (port C upper and lower) are available. The two 4-bit ports can be combinedly used as a third 8-bit port.
2. Any port can be used as an input or output port.
3. Output ports are latched. Input ports are not latched.
4. A maximum of four ports are available so that overall 16 I/O configurations are possible.

• All these modes can be selected by programming a register internal to 8255 known as CWR.
• The control word register has two formats. The first format is valid for I/O modes of operation, i.e. modes 0, mode 1 and mode 2 while the second format is valid for bit set/reset (BSR) mode of operation. These formats are shown in following fig.



I/O modes control word format

b) Mode 1: *(Strobed input/output mode)* In this mode the handshaking control the input and output action of the specified port. Port C lines PC0-PC2, provide strobe or handshake lines for port B. This group which includes port B and PC0-PC2 is called as group B for Strobed data input/output. Port C lines PC3-PC5 provide strobe lines for port A. This group including port A and PC3-PC5 from group A. Thus port C is utilized for generating handshake signals. The salient features of mode 1 are listed as follows:

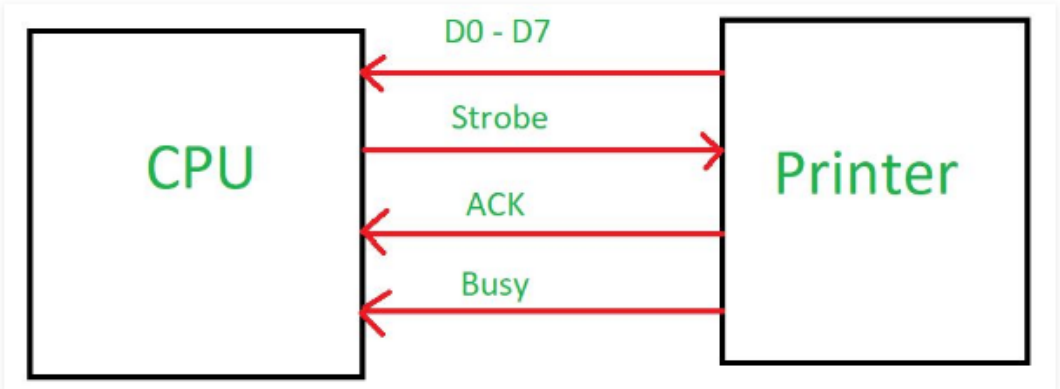
• Two groups – group A and group B are available for strobed data transfer.

• Each group contains one 8-bit data I/O port and one 4-bit control/data port.

• The 8-bit data port can be either used as input and output port. The inputs and outputs both are latched.

• Out of 8-bit port C, PC0-PC2 are used to generate control signals for port B and PC3-PC5 are used to generate control signals for port A. the lines PC6, PC7 may be used as independent data lines.

Example: When CPU wants to send data to slow peripheral device like printer, it will send handshaking signal to printer to tell whether it is ready or not to transfer the data. When printer will be ready it will send one acknowledgement to CPU then there will be transfer of data through data bus.



• The control signals for both the groups in input and output modes are explained as follows:

Input control signal definitions (mode 1):

• STB' (Strobe input) – If this lines falls to logic low level, the data available at 8-bit input port is loaded into input latches.

• IBF (Input buffer full) – If this signal rises to logic 1, it indicates that data has been loaded into latches, i.e. it works as an acknowledgement. IBF is set by a low on STB and is reset by the rising edge of RD input.

• INTR (Interrupt request) – This active high output signal can be used to interrupt the CPU whenever an input device requests the service. INTR is set by a high STB pin and a high at IBF pin. INTE is an internal flag that can be controlled by the bit set/reset mode of either PC4(INTEA) or PC2(INTEB) as shown in fig.

• INTR is reset by a falling edge of RD input. Thus an external input device can be request the service of the processor by putting the data on the bus and sending the strobe signal.

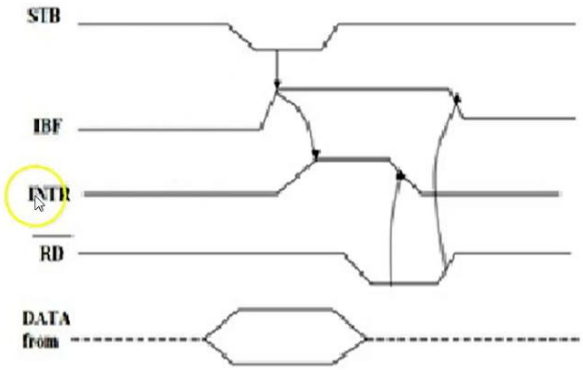
Output control signal definitions (mode 1) :

• OBF (Output buffer full) – This status signal, whenever falls to low, indicates that CPU has written data to the specified output port. The OBF flip-flop will be set by a rising edge of WR signal and reset by a low going edge at the ACK input.

• ACK (Acknowledge input) – ACK signal acts as an acknowledgement to be given by an output device. ACK signal, whenever low, informs the CPU that the data transferred by the CPU to the output device through the port is received by the output device.

• INTR (Interrupt request) – Thus an output signal that can be used to interrupt the CPU when an output device acknowledges the data received from the CPU. INTR is set when ACK, OBF and INTE are 1. It is reset by a falling edge on WR input. The INTEA and INTEB flags are controlled by the bit set-reset mode of PC6 and PC2 respectively.

8255 MODE 1 STROBED I/P DATA TRANSFER



Mode 2: (Strobed Bi-directional Bus I/O):

This mode of operation provides 8255 with an additional feature for the communicating with a peripheral device on an 8-bit data bus. Handshaking signal are provided to maintain proper data flow and synchronization between data transmitter and receiver. The interrupt generation and other functions are similar to mode 1.

• In this mode, 8255 is a bidirectional 8-bit port with handshake signals. The Rd and WR signals decide whether the 8255 is going to operate as an input port or output port.

• The Salient features of Mode 2 of 8255 are listed as follows:

• The single 8-bit port in group A is available.

• The 8-bit port is bidirectional and additionally a 5-bit control port is available.

• Three I/O lines are available at port C.(PC2 – PC0)

• Inputs and outputs are both latched.

• The 5-bit control port C (PC3-PC7) is used for generating / accepting handshake signals for the 8-bit data transfer on port A.

• ***Control signal definitions in mode 2*:**

• INTR – (Interrupt request) As in mode 1, this control signal is active high and is used to interrupt the microprocessor to ask for transfer of the next data byte to/from it. This signal is used for input (read) as well as output (write) operations.

• ***Control Signals for Output operations*:**

• OBF (Output buffer full) – This signal, when falls to low level, indicates that the CPU has written data to port A.

• ACK (Acknowledge) This control input, when falls to logic low level, acknowledges that the previous data byte is received by the destination and next byte may be sent by the processor. This signal enables the internal tristate buffers to send the next data byte on port A.

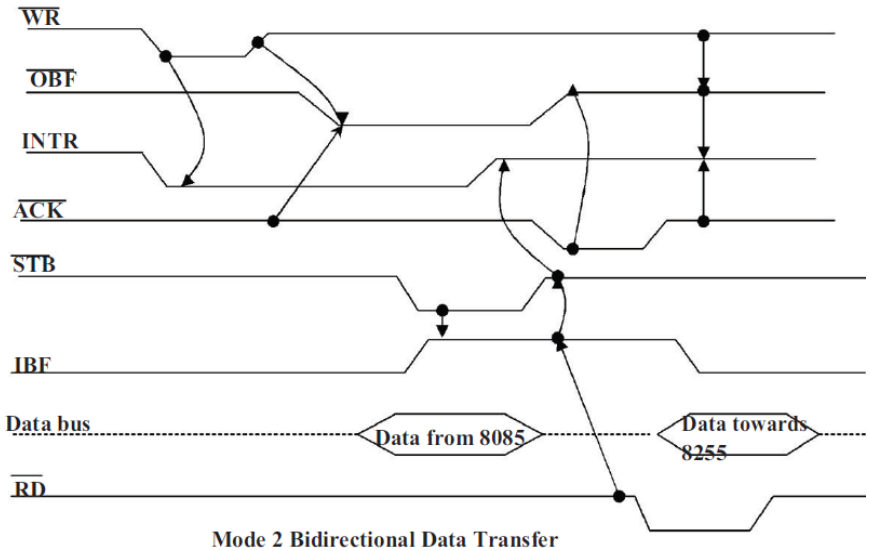
• INTE1 (A flag associated with OBF) This can be controlled by bit set/reset mode with PC6.

• ***Control signals for input operations :***

• STB (Strobe input) A low on this line is used to strobe in the data into the input latches of 8255.

• IBF (Input buffer full) When the data is loaded into input buffer, this signal rises to logic '1'. This can be used as an acknowledge that the data has been received by the receiver.

• The waveforms in fig show the operation in Mode 2 for output as well as input port.

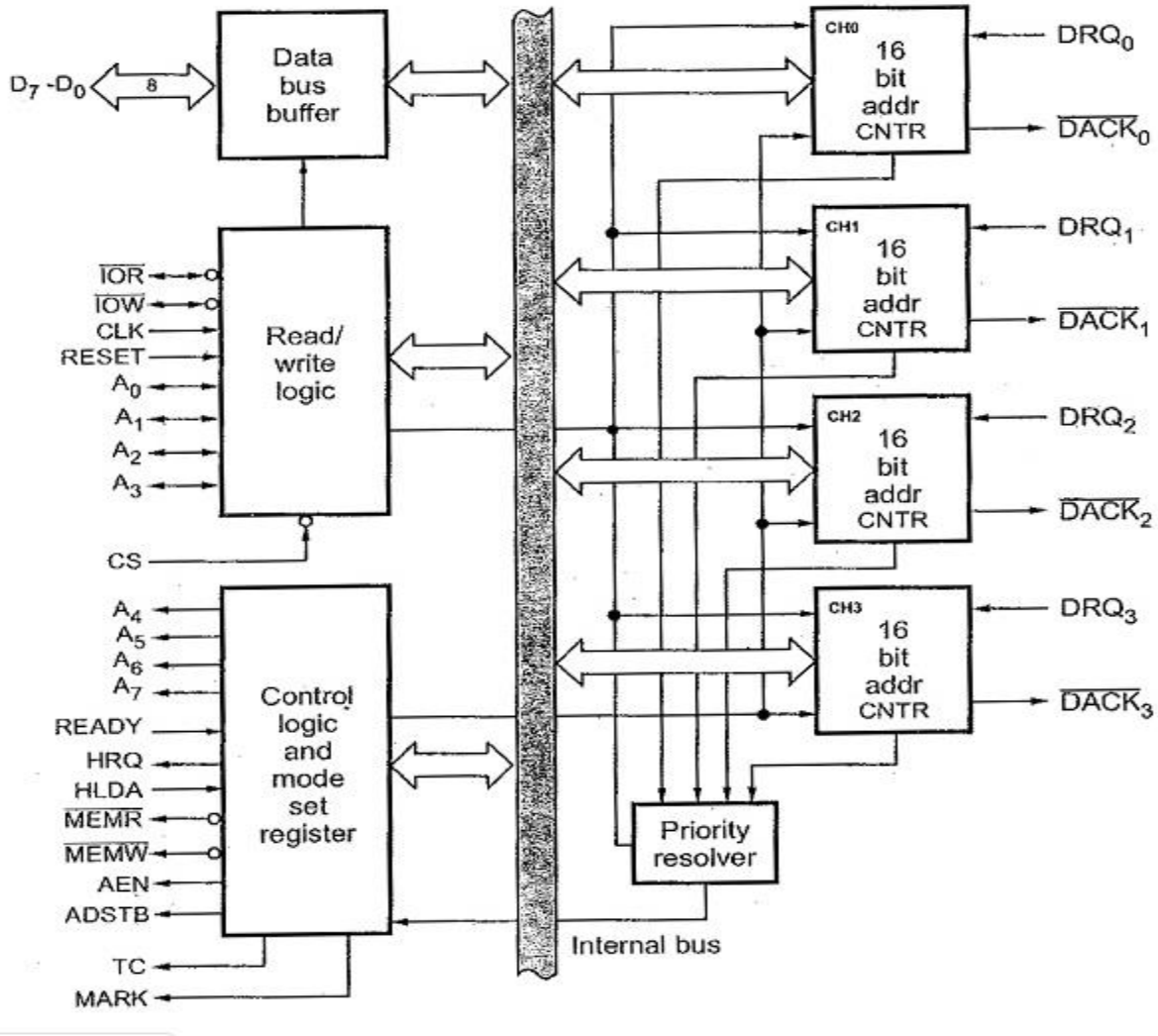


Mode 2 Bidirectional Data Transfer

DMA CONTROLLER

The term DMA stands for direct memory access. The hardware device used for direct memory access is called the DMA controller. DMA controller is a control unit, part of I/O device's interface circuit, which can transfer blocks of data between I/O devices and main memory with minimal intervention from the processor.

ARCHITECTURE OF 8257:



Register organization of 8257

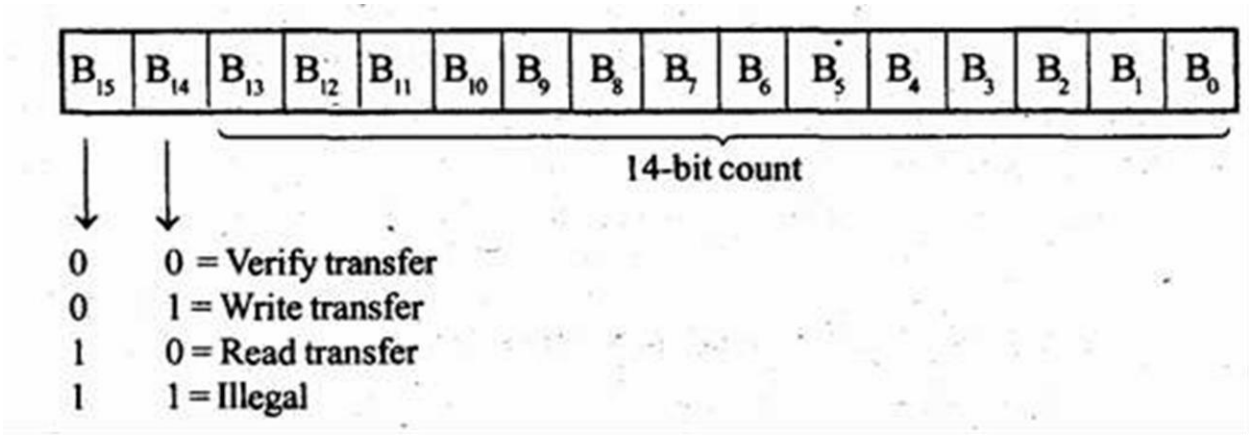
- The 8257 performs the DMA operation over four independent DMA channels. Each of the four channels of 8257 has a pair of two 16 bit registers viz. DMA address register and terminal count registers.
- There are two common registers for all channels namely, mode set registers and status registers .thus there are a total of ten registers the CPU selects one of these ten registers using address lines A0-A3

8257 Register Selection

Register	Address			
	A_3	A_2	A_1	A_0
Channel-0 DMA address register	0	0	0	0
Channel-0 Count register	0	0	0	1
Channel-1 DMA address register	0	0	1	0
Channel-1 Count register	0	0	1	1
Channel-2 DMA address register	0	1	0	0
Channel-2 Count register	0	1	0	1
Channel-3 DMA address register	0	1	1	0
Channel-3 Count register	0	1	1	1
Mode set register (Write only)	1	0	0	0
Status register (Read only)	1	0	0	0

DMA address register : It specifies the address of the first memory location to be accessed. It is necessary to load valid memory address in the DMA address register before channel is enabled.

Terminal Count Register : shows the format of Terminal Count register.



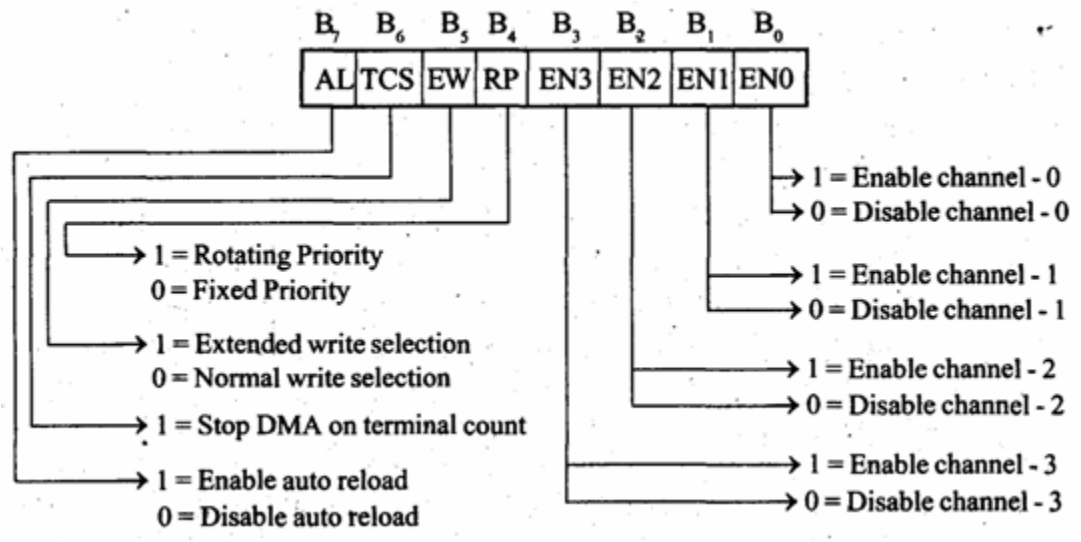
Note : N is number of bytes to be transferred.

The value loaded into the low order 14 bits (C13 — C0) of the terminal count register specifies the number of DMA cycles minus one before the terminal count (TC) output is activated. Therefore, for N number of desired DMA cycles it is necessary to load the value N-1 into the low order 14-bits of the terminal count register. The most significant 2 bits of the terminal count register specifies the type of DMA operation to be performed. It is necessary to load count for DMA cycles and operational code for valid DMA cycle in the terminal count register before channel is enabled.

Mode Set Register:

Fig. 14.65 gives the format of mode set register. Least significant four bits of mode set register, when set, enable each of the four DMA channels. Most significant four bits allow four different options for the Pin Diagram of 8257.

It is normally programmed by the CPU after initializing the DMA address registers and terminal count registers. It is cleared by the RESET input, thus disabling all options, inhibiting all channels, and preventing bus conflicts on power-up.



The bits B0, B1, B2, and B3 of mode set register are used to enable/disable channel -0, 1, 2 and 3 respectively. A one in these bit position will enable a particular channel and a zero will disable it • If the bit B4 is set to one, then the channels will have rotating priority and if it zero then the channels wilt have fixed priority. In rotating priority after servicing a channel its priority is made as lowest. In fixed priority the channel-0 has highest priority and channel-2 has lowest priority. • If the bit B5 is set to one, then the timing of low write signals (MEMW and IOW) will be extended. • If the bit B6 is set to one then the DMA operation is stopped at the terminal count. • The bit B7 is used to select the auto load feature for DMA channel-2. • When bit B7 is set to one, then the content of channel-3 count and address registers are loaded in channel-2 count and address registers respectively whenever the channel-2 reaches terminal count. When this mode is activated the number of channels available for DMA reduces from four to three.

STATUS REGISTER:

Fig. 14.66 shows the status register format. As said earlier, it indicates which channels have reached a terminal count condition and includes the update flag described previously.

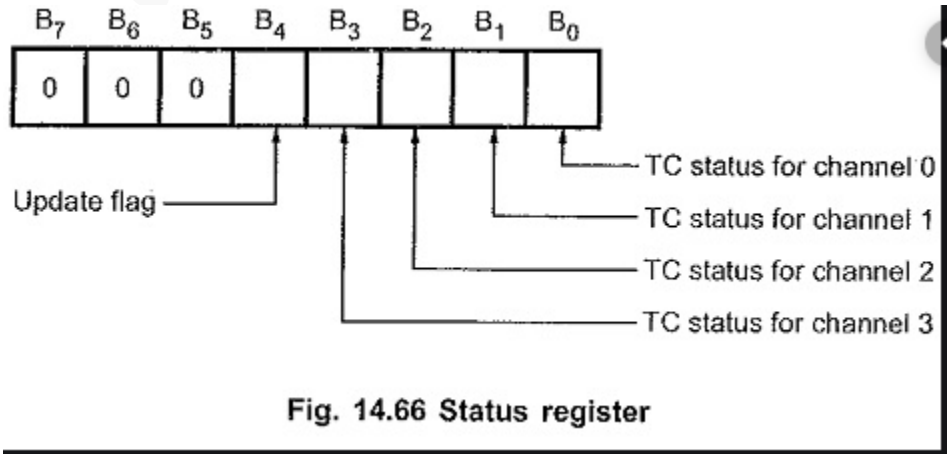


Fig. 14.66 Status register

The TC status bit, if one, indicates terminal count has been reached for that channel. TC bit remains set until the status register is read or the 8257 is reset. The update flag, however, is not affected by a status read operation.

The update flag bit, if one, indicates CPU that 8257 is executing update cycle. In update cycle 8257 loads parameters in channel 3 to channel 2.

The bit B0, B1, B2, and B3 of status register indicates the terminal count status of channel-0, 1,2 and 3 respectively. A one in these bit positions indicates that the particular channel has reached terminal count. • These status bits are cleared after a read operation by microprocessor. • The bit B4 of status register is called update flag and a one in this bit position indicates that the channel-2 register has been reloaded from channel-3 registers in the auto load mode of operation.

Data Bus Buffer:

It is a tri-state, bi-directional, eight bit buffer which interfaces the 8257 to the system data bus. In the slave mode, it is used to transfer data between microprocessor and internal registers of 8257. In master mode, it is used to send higher byte address (A8-A15) on the data bus.

Read/Write logic:

When the CPU is programming or reading one of the internal registers of Pin Diagram of 8257 (i.e, when the 8257 is in the slave mode), the Read/Write logic accepts the I/O Read (IOR) or I/O Write (IOW) signal, decodes the the least significant four addiess bits (A0 – A3) and either writes the contents of the data bus into the addressed register (if IOW is low) or places the contents of the addressed register onto the data bus (if IOR is low).

During DMA cycles (i.e. when the 8257 is in the master mode) the Read/Write logic generates the I/O read and memory write (DMA write cycle) or I/O write and memory read (DMA read cycle) signals which control the data transfer between peripheral and memory device.

Control logic:

It controls the sequence of operations during all DMA cycles (DMA read, DMA write, DMA verify) by generating the appiopriate control signals and the 16-bit address that specifies the memory location to be accessed. It consists of mode set register and status register. Mode set register is programmed by the CPU to configure 8257 whereas the status register is read by CPU to check which channels have reached a terminal count condition and status of update flag.

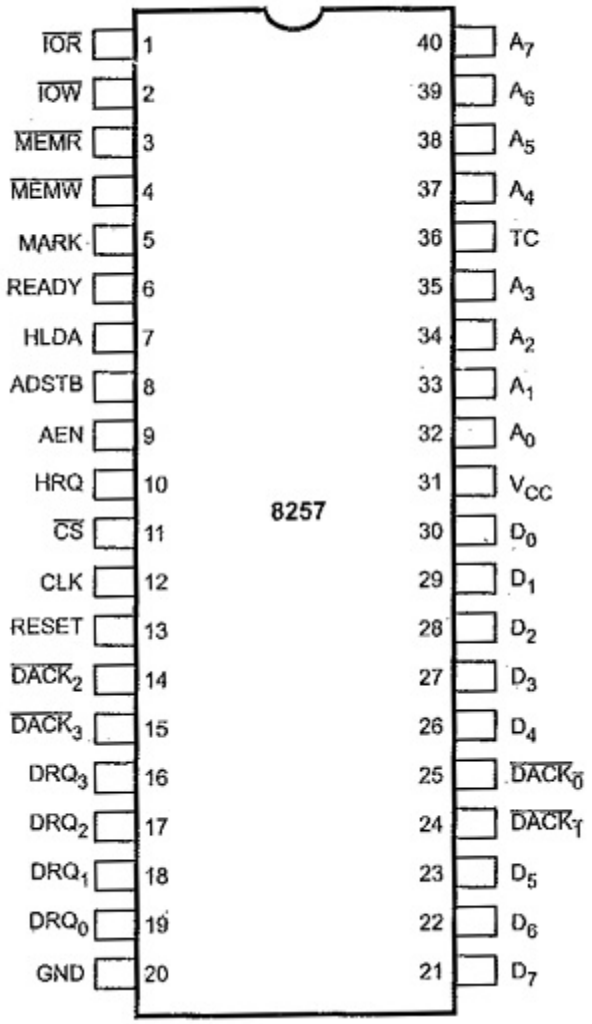
Priority Resolver:

It resolves the peripherals requests. It can be programmed to work in two modes, either in fixed mode or rotating priority mode.

PIN DIAGRAM OF 8257:

Fig. 14.61 shows Pin Diagram of 8257.

Data Bus (D_0-D_7) : These are bi-directional tri-state signals connected to the system data bus. When CPU is having control of system bus it can access contents of address register, status register, mode set register, and a terminal count register and it can also program, control registers of DMA controller, through the data bus.



During DMA cycles these lines are used to send the most significant bytes of the memory address from one of the

Address Bus (A0-A3 and A4-A7) : The four least significant lines A0-A3 are bi – directional tri – state signals. In the idle cycle they are inputs and used by the CPU to address the register to be loaded or read. In the Active cycle they output the lower 4 bits of the address for DMA operation. A4-A7 are unidirectional lines, provide 4-bits of address during DMA service.

Address Strobe (ADSTB) : This signal is used to demultiplex higher byte address and data using external latch.

Address Enable (AEN) : This active high signal enables the 8-bit latch containing the upper 8-address bits onto the system address bus. AEN can also be used to disable other system bus drivers during DMA transfers.

Memory Read and Memory Write (MEMR, MEMW) : These are active low tri-state signals. The MEMR signal used to access data from the addressed memory location during a DMA read or memory-to-memory transfer and MEMW signal is used to write data to the addressed memory location during DMA write or memory to memory transfer.

I/O Read and I/O Write (IOR and IOW) : These are active low bi-directional signals. In idle cycle, these are an input control signals used by CPU to read/write the control registers. In the active cycle IOR signal is used to access data from a peripheral and IOW signal is used to send data to the peripheral.

Chip Select (CS) : This is an active low input, used to select the 8257 as an I/O device during the idle cycle. This allows CPU to communicate with Pin Diagram of 8257.

Reset : This active high signal clears, the command, status, request and temporary registers. It also clears the first/last flip-flop and sets the Master Register. After reset the device is in the idle cycle.

Ready : This input is used to extend the memory read and write signals from the 8257 to interface slow memories or I/O devices.

Hold request (HRQ) : Any valid DREQ causes 8257 to issue the HRQ. It is used for requesting CPU to get the control of system bus.

Hold Ackmiwledge (HLDA) : The active high Hold Acknowledge from the CPU indicates that it has relinquished control of the system bus.

DREQ0-DREQ3 : These are DMA request lines, which are activated to obtain DMA service, until the corresponding DACK signal goes active.

DACK0-DACK3 : These are used to indicate peripheral devices that the DMA request is granted.

Terminal Count (TC) : This is active high signal concern with the completion of DMA service. The TC output signal is activated at the end of DMA service, i.e. when present cycle is a last cycle for the current data block.

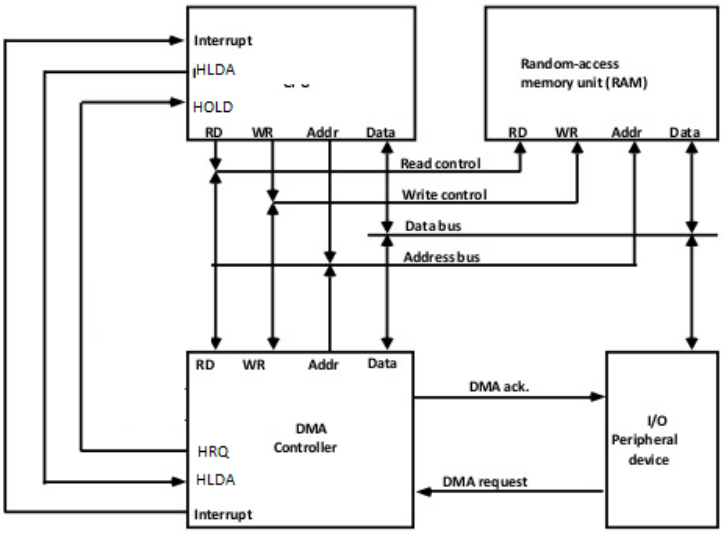
MARK : This output notifies the selected peripheral that the current DMA cycle is the 128th cycle since the previous MARK output. MARK always occurs at 128 (all multiplies of 128) cycles from the end of the data block.

Direct Memory Access TRANSFER

- DMA is a process of communication for data transfer between memory and input/output, controlled by an external circuit called DMA controller, without involvement of CPU.
- 8086 MP has two pins HOLD and HLDA which are used for DMA operation.
- First, DMA controller sends a request by making HRQ control line high. When MP receives high signal to HOLD pin, it first completes the execution of current machine cycle, it takes few clocks and sends HLDA signal to the DMA controller.
- After receiving HLDA TO DMA controller, the DMA controller takes control over system bus and transfers data directly between memory and I/O without involvement of CPU. During DMA operation, the processor is free to perform next job which does not need system bus.
- At the end of data transfer, the DMA controller terminates the request by sending low signal to HOLD pin and MP regains control of system bus by making HLDA low.
- Figure shows the block diagram of a typical DMA controller. The unit communicates with the MP via the data bus and control lines.
- The registers in the DMA are selected by the MP through the address bus by enabling the DS (DMA select) and RS (Register Select) inputs. The RD (read) and WR (write) inputs are bidirectional.
- When the bus grant (HRQ) input is 0, the MP can communicate with the DMA registers through the data bus to read from or write to the DMA registers. When HRQ=1, the processor does not have control over

the system buses and the DMA can communicate directly with the memory by specifying an address in the address bus and activating the RD or WR control.

- The DMA controller has three registers: an address register, a word count register and a control register.



- The address register contains an address to specify the desired location in memory. The address bits go though bus buffers into the address bus. The address register is incremented after each word that is transferred to memory.
- The word count register holds the number of words to be transferred. The register is decremented by one after each word transfer and internally tested for zero.

PROGRAMMABLE COMMUNICATION INTERFACE 8251 USART

8251 universal synchronous asynchronous receiver transmitter (USART) acts as a mediator between microprocessor and peripheral to transmit serial data into parallel form and vice versa.

1. It takes data serially from peripheral (outside devices) and converts into parallel data.
2. After converting the data into parallel form, it transmits it to the CPU.
3. Similarly, it receives parallel data from microprocessor and converts it into serial form.
4. After converting data into serial form, it transmits it to outside device (peripheral).

METHODS OF DATA COMMUNICATION

There are basically 3 modes of data communication:

Simplex: Data travels in one direction (from A to B). An example of a simplex link would be scoreboards such as those used in hockey, basketball, or other sports. The information is entered at a console by the score/timekeeper and sent serially to large displays that everybody can see.

Half-duplex: Data travels in one direction (from A to B) and then the other direction (from B to A) but not at the same time. The RS-485 is half-duplex.

Full-duplex: Data can travel in both directions at the same time.

Architecture of 8251 Microcontroller:

Fig. 14.37 shows the block diagram of IC Block Diagram of 8251 Microcontroller. It includes : Data bus buffer, Read/Write control logic, modem control, Transmit buffer, Transmit Control, Receiver Buffer and Receiver control.

Block Diagram of 8251 Microcontroller

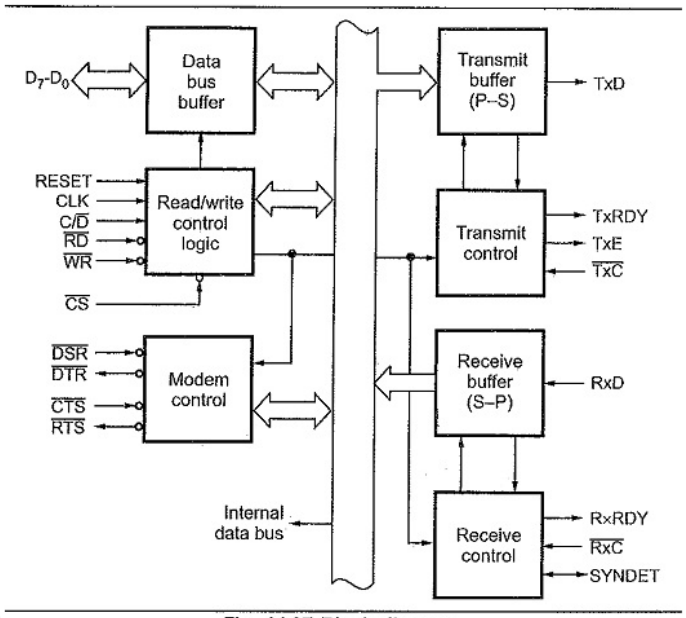


Fig. 14.37 Block diagram

Data Bus Buffer :

This tri-state, bi-directional, 8-bit buffer is used to interface Block Diagram of 8251 Microcontroller to the system data bus. Along with the data, control word, command words and status information are also transferred through the Data Bus Buffer.

Read/Write control logic :

This functional block accepts inputs from the system control bus and generates control signals for overall device operation. It decodes control signals on the 8085 control bus into signals which controls the internal and external I/O bus. It contains the control word register and command word register that stores the various control formats for the device functional definition.

Transmit Buffer:

The transmit buffer accepts parallel data from the CPU, adds the appropriate framing information, serializes it, and transmits it on the TxD pin on the falling edge of TxC.

It has two registers : A buffer register to hold eight bits and an output register to convert eight bits into a stream of serial bits. The CPU writes a byte in the buffer register, Which is transferred to the output register when it is empty. The output register then transmits serial data on the TxD pin.

In the asynchronous mode the transmitter always adds START bit; depending on how the unit is programmed, it also adds an optional even or odd parity bit, and either 1, 1 1/2, or 2 STOP bits. In synchronous mode no extra bits (other than parity, if enable) are generated by the transmitter.

Transmit Control :

It manages all activities associated with the transmission of serial data. It accepts and issues signals both externally and internally to accomplish this function.

TxRDY (Transmit Ready) : This output signal indicates CPU that buffer register is empty and the USART is ready to accept a data character. It can be used as an interrupt to the system or, for polled operation, the CPU can ‘check TxRDY using the status read operation. This signal is reset when a data byte is loaded into the bliffer register.

TxE (Transmitter Empty) : This is an output signal. A high on this line indicates that the output buffer is empty. In the synchronous mode, if the CPU has failed to load a new character in time, TxE will go high momentarily as SYN characters are loaded into the transmitter to fill the gap in transmission.

TxC (Transmitter Clock) : This clock controls the rate at which characters are transmitted by USART. In the synchronous mode TxC is equivalent to the ‘baud rate, and is supplied by the modem. In asynchronous mode TxC is 1, 16, or 64 times the baud rate. The clock division is programmable. It can be programmed by writing proper mode word in the mode set register.

Receiver Buffer:

The receiver accepts serial data on the RxD line, converts this serial data to parallel format, checks for bits or characters that are unique to the communication technique and sends an “assembled” character to the CPU.

When Block Diagram of 8251 Microcontroller is in the asynchronous mode an4 it is ready to accept a character, it looks for a low level on the RxD line. When it receives the low level, it assumes that it is a START bit and enables an internal counter, At a count equivalent to one-half of a hit time, the RxD line is sampled again. If the line is still low, a valid START bit is detected and the 8251A proceeds to assemble the character. After successful reception of a START bit the 8251A receives data, parity and STOP bits, and then transfers the data on the receiver input register. The data is then transferred into the receiver buffer register.

In the synchronous mode the receiver simply receives the specified number of data bits and transfers them to the receiver input register and then to the receiver buffer register.

Receiver Control:

It manages all receiver-related activities. Along with data reception, it does false start bit detection, parity error detection, framing error detection, sync detection and break detection.

RxRDY (Receiver Ready) : This is an output signal. It goes high (active), when the USART has a character in the buffer register and is ready to transfer it to the CPU. This line can be used either to indicate the status in the status register or to interrupt the CPU. This signal is reset when a data byte from receiver buffer is read by the CPU.

RxC (Receiver Clock) : This clock controls the rate at which the character is to be received by USART in the synchronous mode. RxC is equivalent to the baud rate, and is supplied by the modem. In asynchronous mode RxC is 1, 16, or 64 times the baud rate. The clock division is programmable. It can be programmed by writing proper mode word in the mode set register.

Modem Control:

The Block Diagram of 8251 Microcontroller has a set of control inputs and output's that can be used to simplify the interface to almost any modem. It provides control circuitry for the generation of RTS and DTR and the reception of CTS and DSR. In addition, a general purpose inverted output and a general purpose input are provided. The output is labeled DTR and the input is labeled DSR. DTR can be asserted by setting bit 2 of the command instruction; DSR can be sensed as bit 7 of the status register. When used as a modem control signal DTR indicates that the terminal is ready to communicate and DSR indicates that it is ready for communication.

8251A Control Words:

The control words defines the complete functional definition of Block Diagram of 8251 Microcontroller and they must be loaded before any transmission or reception. The control words of Block Diagram of 8251 Microcontroller are split into two formats

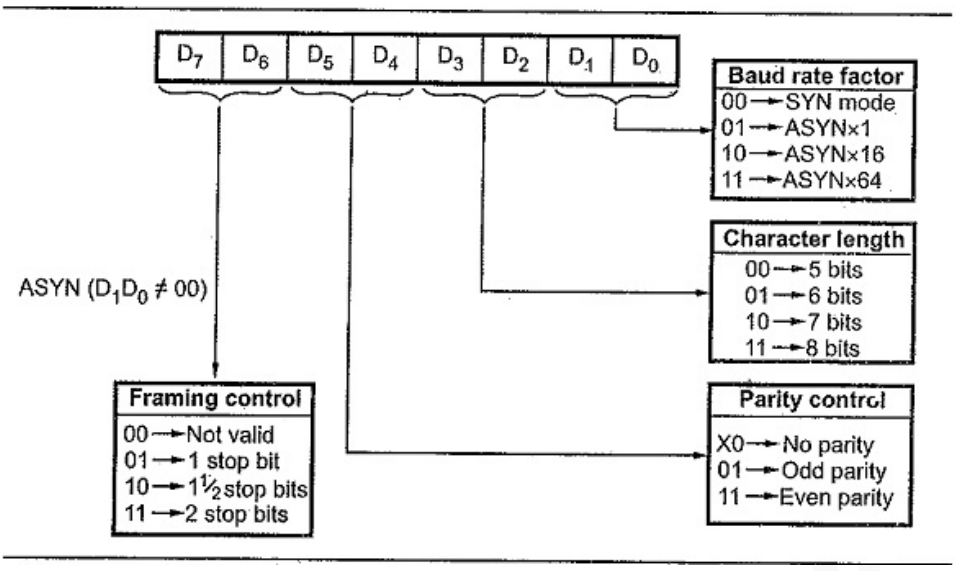
- Mode instruction
- Command instruction

Asynchronous Mode

Mode Instruction Control Word This defines the general operational characteristics of 8251A. After internal (reset command) or external (reset input pin) reset, this must be written to configure the 8251A as per the required operation. Once this has been written into 8251A, SYNC characters or command instructions may be programmed further as per the requirements. To change the mode of operation from synchronous to asynchronous or vice-a-versa, the 8251A has to be reset using master chip reset.

Figure 6.28 shows asynchronous mode instruction control word format.

Asynchronous Mode (Transmission) When a data character is sent to 8251A by the CPU, it adds start bits prior to the serial data bits, followed by optional parity bit and stop bits using the asynchronous mode instruction control word format. This sequence is then transmitted using TXD output pin on the falling edge of $\overline{TXC}$. When no data characters are sent by the CPU to 8251A the TXD output remains 'high', if a 'break' has not been detected.



Asynchronous Mode (Receive) A falling edge on RXD input line marks a start bit. At baud rates of 16x and 64x, this start bit is again checked at the center of start bit pulse and if detected low, it is a valid start bit which starts counting. The bit counter locates the data bits, parity bit and stop bit. If a parity error occurs, the parity error flag is set. If a low level is detected as the stop bit, the framing error flag is set. The receiver requires only one stop bit to mark end of the data bit string, regardless of the stop bit programmed at the transmitting end. This 8-bit character is then loaded into parallel I/O buffer of 8251A. RXRDY pin is then raised high to indicate to the CPU that a character is ready for it. If the previous character has not been read by the CPU, the new character replaces it, and the overrun flag is set indicating that the previous character is lost. These error flags can be cleared using an error reset instruction. Figure 6.29 shows asynchronous mode transmission and receiver data formats. If character length is 5 to 7 bits then the remaining bits are set to zero.

Synchronous Mode Figure 6.30 shows the synchronous mode instruction format with its bit definitions.

Synchronous Mode (Transmission) The TXD output is high until the CPU sends a character to 8251A which usually is a SYNC character. When $\overline{CTS}$ line goes low, the first character is serially transmitted out. All the characters are shifted out on the falling edges of $\overline{TXC}$. Data is shifted out at the same rate as $\overline{TXC}$, over TXD output line. If the CPU buffer becomes empty, the SYNC character or characters are inserted in the data stream over TXD output. The TXEMPTY pin is raised high to indicate that the 8251A is empty (i.e. it does not have any byte to transmit) and is transmitting SYNC characters. The TXEMPTY pin is reset, automatically when a data character is written to 8251A by the CPU. Figure 6.31 shows the relation between TXEMPTY and SYNC character insertion.

Synchronous Mode (Receiver) In this mode, the character synchronization can be achieved internally or externally. If this mode is programmed, then 'ENTER HUNT' command should be included in the first command instruction word written into the 8251A. The data on RXD pin is sampled on rising edge of the $\overline{RXC}$. The content of the receiver buffer is compared with the first SYNC character at every

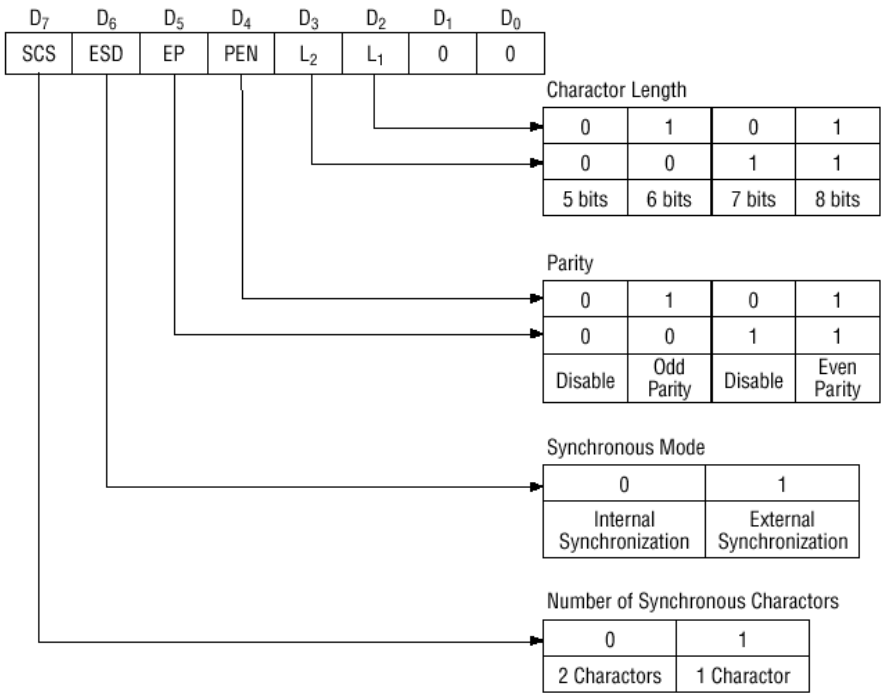


Fig. 3 Bit Configuration of Mode Instruction (Synchronous)

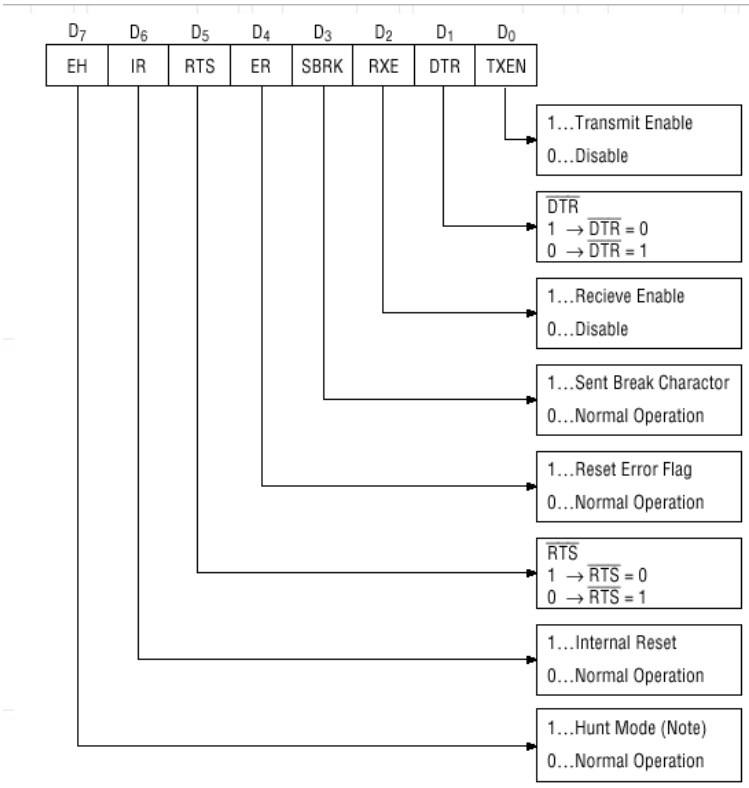
edge until it matches. If 8251A is programmed for two SYNC characters, the subsequent received character is also checked. When both the characters match, the hunting stops. The SYNDET pin is set high and is reset automatically by a status read operation. If a parity bit is programmed, the SYNDET signal will not go as high as the middle of parity bit, or till the middle of the last data bit.

Command Instruction:

After the mode instruction, command character should be issued to the USART. It controls the operation of the USART within the basic frame work established by the mode instruction. Fig. 14.39 shows command instruction format.

Items to be set by command are as follows:

- Transmit Enable/Disable
- Receive Enable/Disable
- DTR, RTS Output of data.
- Resetting of error flag.
- Sending to break characters
- Internal resetting
- Hunt mode (synchronous mode)



Note: Seach mode for synchronous charactors in synchronous mode.

Fig. 4 Bit Configuration of Command

8251A Status Word:

In the data communication systems it is often necessary to examine the "status" of the transmitter and receiver. It is also necessary for CPU to know if any error has occurred during communication. The Block Diagram of 8251 Microcontroller allow the programmer to read above mentioned information from the status register any time during the functional operation. Fig. 14.40 shows the format of status register.

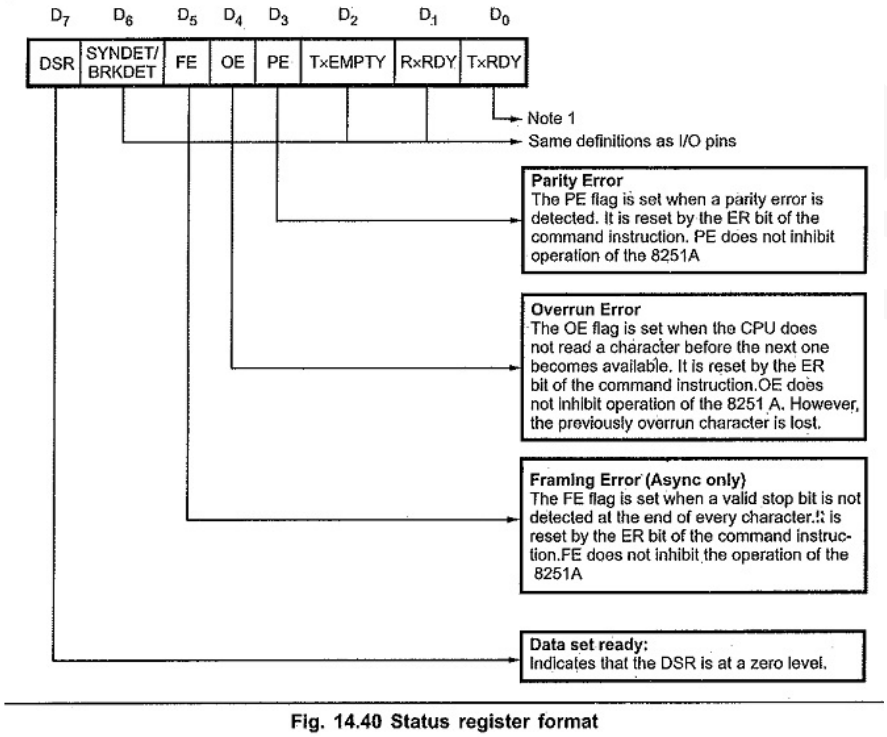


Fig. 14.40 Status register format

Error Definitions:

Parity Error : At the time of transmission of data an even or odd parity bit is inserted in the data stream. At the receiver end, if parity of the character does not match with the pre-defined parity, parity error occurs.

Overrun Error : In the receiver section received character is stored in the receiver buffer. The CPU is supposed to read this character before reception of the next character. But if CPU fails in reading the character loaded in the receiver buffer, the next the received character replaces the previous one and the OVERRRUN Error occurs.

Framing Error : If valid stop bit is not detected at the end each character framing error occurs.

All these errors, when occur, set the corrosponding bits in the status register. These error bits are reset by setting ER bit in the command instruction.

INTERFACING ANALOG TO DIGITAL DATA CONVERTERS:

 In most of the cases, the PIO 8255 is used for interfacing the analog to digital converters with microprocessor.

 We have already studied 8255 interfacing with 8086 as an I/O port, in previous section. This section we will only emphasize the interfacing techniques of analog to digital converters with 8255.

 The analog to digital converters is treated as an input device by the microprocessor that sends an initializing signal to the ADC to start the analogy to digital data conversation process. The start of conversation signal is a pulse of a specific duration.

 The process of analog to digital conversion is a slow

 Process and the microprocessor have to wait for the digital data till the conversion is over. After the conversion is over, the ADC sends end of conversion EOC signal to inform the microprocessor that the conversion is over and the result is ready at the output buffer of the ADC. The set asks of issuing an SOC pulse to ADC, reading EOC signal from the ADC and reading the digital output of the ADC are carried out by the CPU using 8255 I/O ports.

 The time taken by the ADC from the active edge of SOC pulse till the active edge of EOC signal is called as the conversion delay of the ADC.

 It may range anywhere from a few microseconds in caseof fast ADC to even a few hundred milliseconds in case of slow ADCs.

 The available ADC in the market use different conversion techniques for conversion of analog signal to digitals. Successive approximation techniques and dual slope integration techniques are the most popular techniques used in the integrated ADC chip.

General algorithm for ADC interfacing contains the following steps:

 Ensure the stability of analog input, applied to the ADC.

 Issue start of conversion pulse to ADC

 Read end of conversion signal to mark the end of conversion processes.

 Read digital data output of the ADC as equivalent digital output.

 Analog input voltage must be constant at the input of the ADC right from the start of conversion till the end of the conversion to get correct results. This may be ensured by as ample and hold circuit which samples the analog signal and holds it constant for specific time duration. The microprocessor may issue a hold signal to the sample and hold circuit.

 If the applied input changes before the complete conversion process is over, the digital equivalent of the analog input calculated by the ADC may not be correct.

ADC 0808/0809:

 The analog to digital converter chips 0808 and 0809 are 8-bit CMOS, successive approximation converters. This technique is one of the fast techniques for analog to digital conversion. The conversion delay is 100μs at a clock frequency of 640 KHz, which is quite low as compared to other converters. These converters do not need any external zero or full scale adjustments as they are already taken care of by internal circuits.

 These converters internally have a 3:8 analog multiplexer so that at a time eight different analog conversion by using address lines - ADD A, ADD B, ADD C, as shown. Using these address inputs, multichannel data acquisition system can be designed using a single ADC. The CPU may drive these lines using output port lines in case of multichannel applications. In case of single input applications, these may be hardwired to select the proper input.

 There are unipolar analog to digital converters, i.e. they are able to convert only positive analog input voltage to their digital equivalent. These chips do not contain any internal sample and hold circuit.

 If one needs a sample and hold circuit for the conversion of fast signal into equivalent digital quantities, it has to be externally connected at each of the analog inputs.

Analog I/P selected	Address lines		
	C	B	A
I/P 0	0	0	0
I/P 1	0	0	1
I/P 2	0	1	0
I/P 3	0	1	1
I/P 4	1	0	0
I/P 5	1	0	1
I/P 6	1	1	0
I/P 7	1	1	1

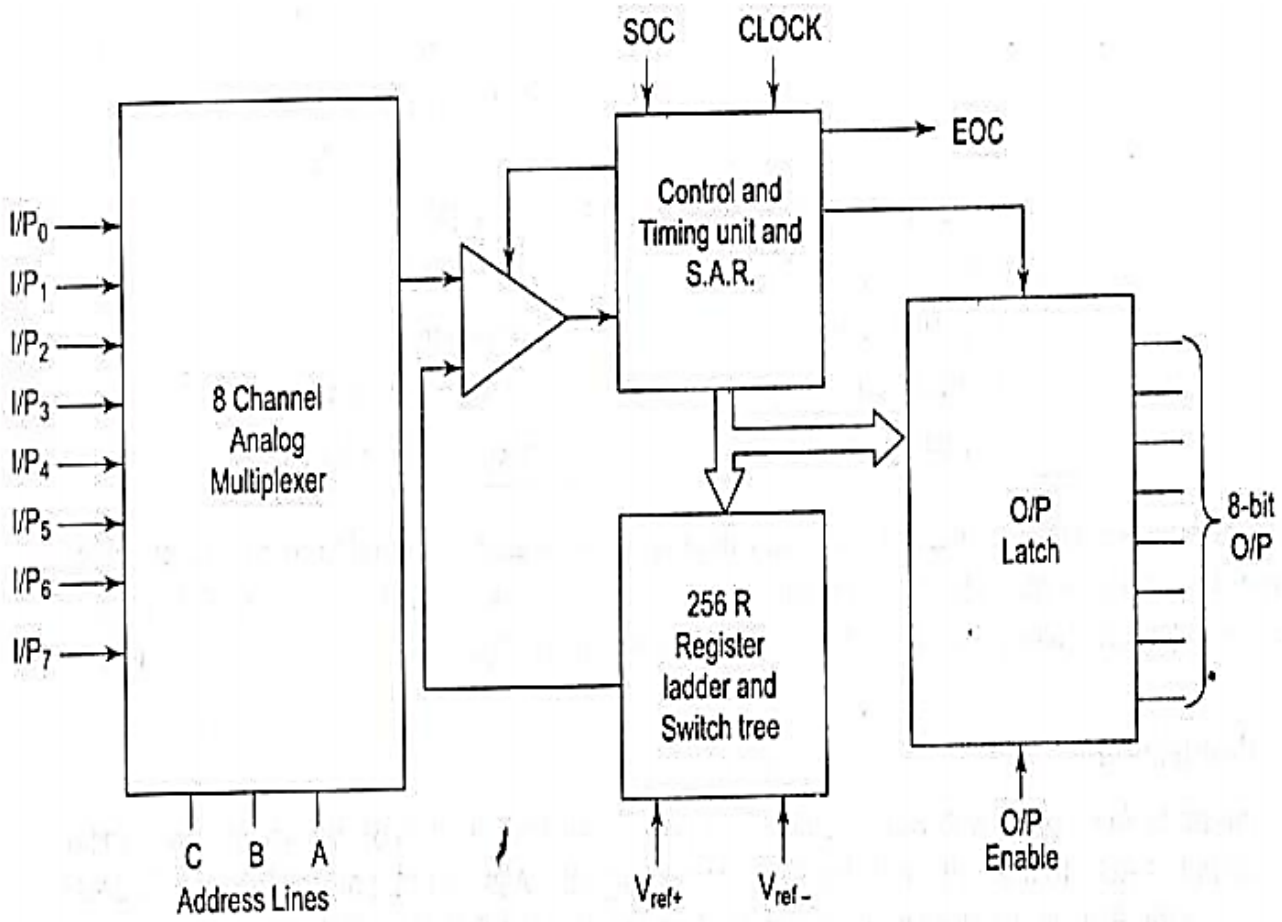
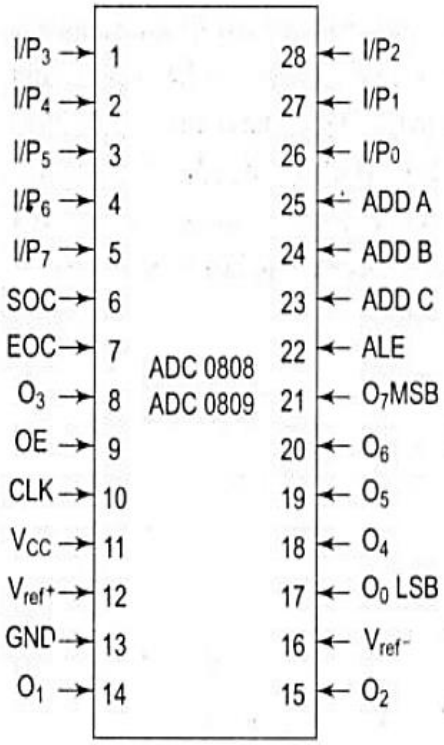


Fig.1 Block Diagram of ADC 0808/0809



I/P_0 - I/P_7	Analog inputs
ADD A, B, C	Address lines for selecting analog inputs
O_7 - O_0	Digital 8-bit output with O_7 MSB and O_0 LSB
SOC	Start of conversion signal pin
EOC	End of conversion signal pin
OE	Output latch enable pin, if high enable output
CLK	Clock input for ADC
V_{CC}, GND	Supply pins +5V and GND
V_{ref+} and V_{ref-}	Reference voltage positive (+5 Volts maximum) and Reference voltage negative (0V minimum)

Minimum SOC pulse width	100 ns
Minimum ALE pulse width	100 ns
Clock frequency	10 to 1280 kHz
Conversion time	100 ms at 640 kHz
Resolution	8-bit
Error	+/–1 LSB
V_{ref+}	Not more than +5V
V_{ref-}	Not less than GND
+ V_{cc} supply	+ 5 V DC
Logical 1 i/p voltage	minimum V_{cc} –1.5 V
Logical 0 i/p voltage	maximum 1.5 V
Logical 1 o/p voltage	minimum V_{cc}–0.4 V
Logical 0 o/p voltage	maximum 0.45 V

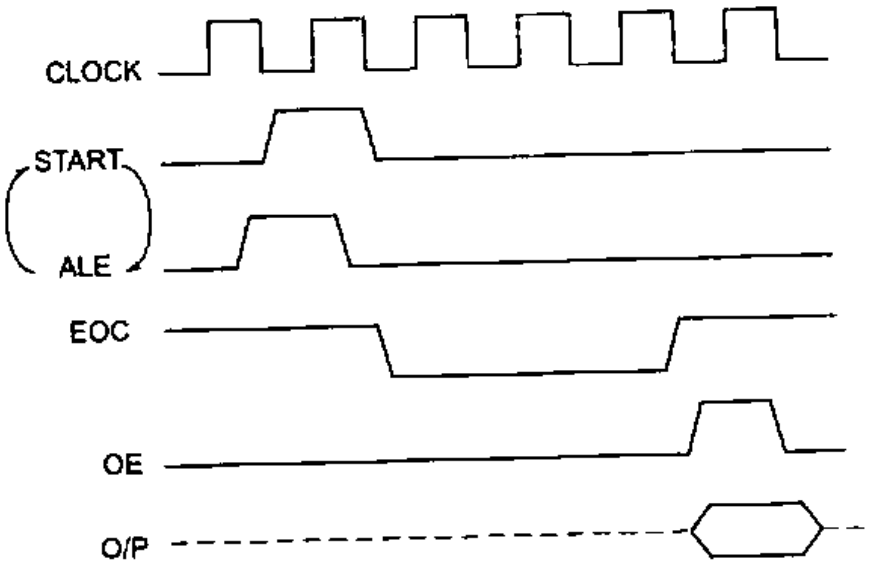
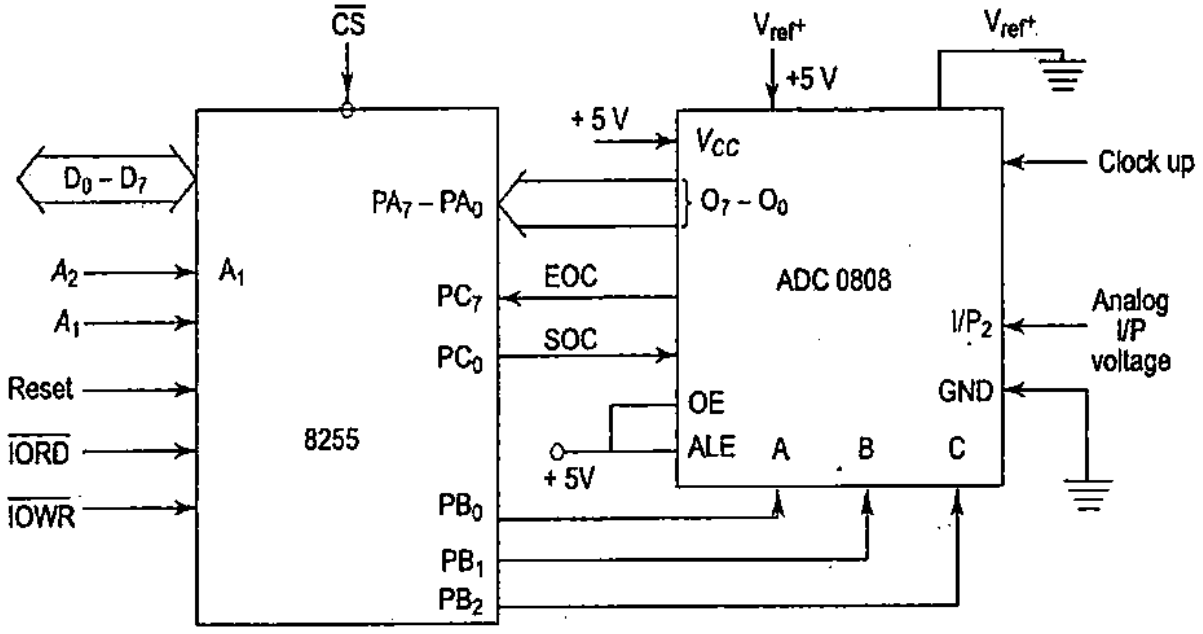


Fig.3 Timing Diagram Of ADC 0808.

Example : Interface ADC 0808 with 8086 using 8255 ports. Use Port A of 8255 for transferring digital data output of ADC to the CPU and Port C for control signals. Assume that an analog input is present at I/P2 of the ADC and a clock input of suitable frequency is available for ADC. Draw the schematic and write required ALP



Solution Figure 5.39 shows the interfacing connections of ADC0808 with 8086 using 8255. The analog input I/P_2 is used and therefore address pins A,B,C should be 0,1,0 respectively to select I/P_2. The OE and ALE pins are already kept at +5V to select the ADC and enable the outputs. Port C upper acts as the input port to receive the EOC signal while port C lower acts as the output port to send SOC to the ADC. Port A acts as a 8-bit input data port to receive the digital data output from the ADC. The 8255 control word is written as follows:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Control word
1	0	0	1	1	0	0	0	= 98 H

The required ALP is given as follows:

```
        MOV AL,98 H        ; Initialise 8255 as
        OUT CWR,AL         ; discussed above
        MOV AL,02H         ; Select I/P2 as analog
        OUT Port B,AL      ; input
        MOV AL,00H         ; Give start of conversion
        OUT Port C,AL      ; pulse to the ADC.
        MOV AL,01 H        ;
        OUT Port C,AL      ;
        MOV AL,00H         ;
        OUT Port C,AL      ;
WAIT :  IN AL,PortC        ; Check for EOC by
        RCL                ; reading port C upper and
        JNC WAIT           ; rotating through carry.
        IN AL,PortA        ; If EOC, read digital equivalent in AL
        HLT                ; Stop
```

Program 5.11 *ALP for Problem 5.16*

INTERFACING DIGITAL TO ANALOG CONVERTERS:

The digital to analog converters convert binary numbers into their analog equivalent voltages. The DAC find applications in areas like digitally controlled gains, motor speed controls, programmable gain amplifiers, etc.

DAC0800 8-bit Digital to Analog Converter

- The DAC 0800 is a monolithic 8-bit DAC manufactured by National Semiconductor.
- It has settling time around 100ms and can operate on a range of power supply voltages

i.e. from 4.5V to +18V. Usually the supply V+ is 5V or +12V.

- The V-pin can be kept at a minimum of -12V.

Problem 5.18

Write an assembly language program to generate a triangular wave of frequency 500 Hz using the interfacing circuit given in Fig. 5.48. The 8086 system operates at 8 MHz. The amplitude of the triangular wave should be +5 V.

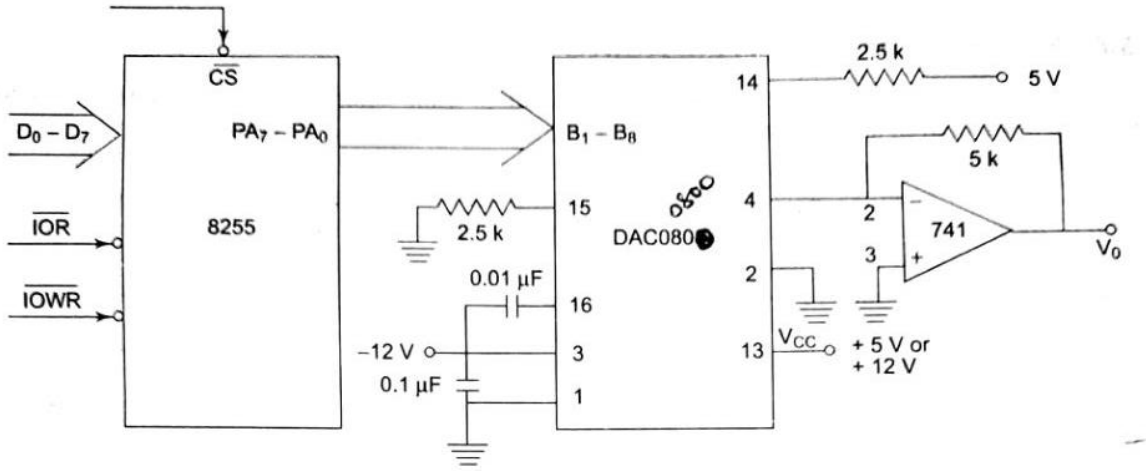


Fig. 5.48 *Interfacing DAC0800 with 8086*

Solution The V_{ref+} should be tied to +5 V to generate a wave of +5V amplitude. The required frequency of the output is 500 Hz, i.e. the period is 2 ms. Assuming the wave to be generated is symmetric, the waveform will rise for 1 ms and fall for 1 ms. This will be repeated continuously. In the previous program, we have already written an instruction sequence for period 1 ms. Using the same instruction sequence one can derive this triangular waveform. The ALP is given as follows:

```
ASSUME    CS : CODE
CODE      SEGMENT
START :   MOV AL,80 H       ; Initialise 8255 ports
          OUT CWR,AL        ; suitably.
          MOV AL,00H        ; Start rising ramp from
BACK :    OUT Port A,AL     ; 0V by sending 00H to DAC.
          INC AL            ; Increment ramp till 5V
          CMP AL,FFH        ; i.e. FFH.
          JB BACK           ; If it is FFH then,
BACK1 :   OUT Port A,AL     ; Output it and start the falling
          DEC AL            ; ramp by decrementing the
          CMP AL,00         ; counter till it reaches
          JA BACK1          ; zero. Then start again
          JMP BACK          ; for the next cycle.
CODE      ENDS
          END START
```

Program 5.14 *ALP for Generating a Triangular Wave Using DAC 0800*

AD7523 8BIT MULTIPLYING DAC

 The supply range extends from +5V to +15V , while Vref may be anywhere between -10V to +10V. The maximum analog output voltage will be +10V, when all the digital inputs are at logic high state. Usually a Zener is connected between OUT1 and OUT2 to save the DAC from negative transients.

 An operational amplifier is used as a current to voltage converter at the output of AD 7523 to convert the current output of AD7523 to a proportional output voltage.

 It also offers additional drive capability to the DAC output. An external feedback resistor acts to control the gain. One may not connect any external feedback resistor, if no gain control is required.

Problem 5.17

Interface DAC AD7523 with an 8086 CPU running at 8 MHz and write an assembly language program to generate a sawtooth waveform of period 1 ms with V_{max} 5V.

Solution Figure 5.46 shows the interfacing circuit of AD 7523 with 8086 using 8255. Program 5.13 gives an ALP to generate a sawtooth waveform using this circuit.

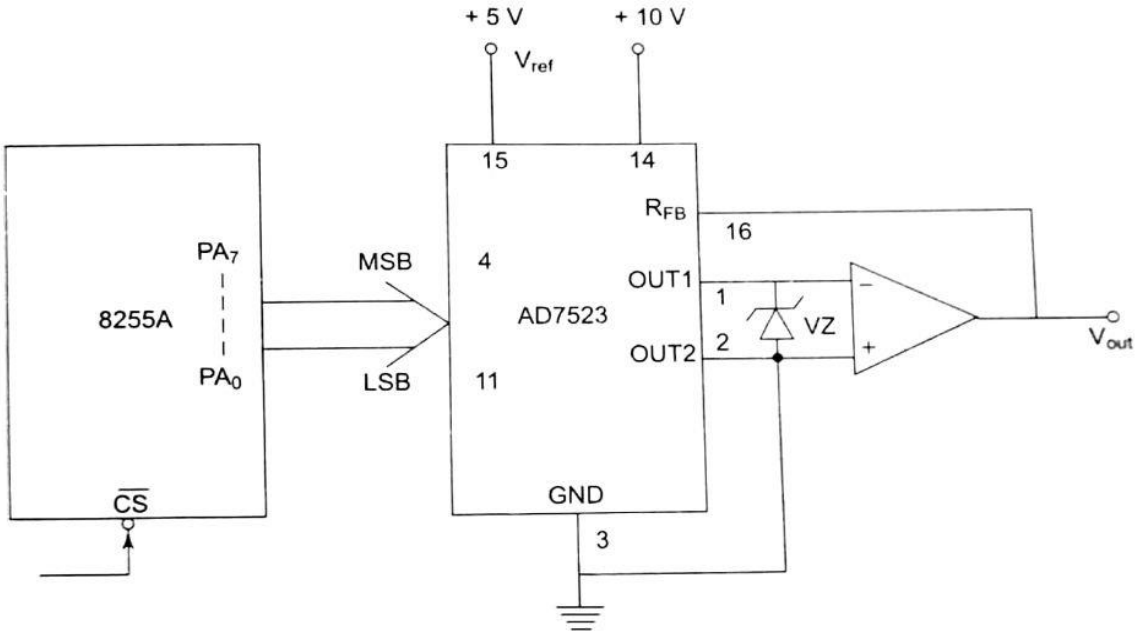


Fig. 5.46 *Interfacing of AD7523*

```
ASSUME    CS : CODE
CODE      SEGMENT
START:    MOV AL,80 H      ; Initialise port A as output
          OUT CWR,AL       ; port


AGAIN:    MOV AL,00H       ; Start the ramp from 0V
BACK :    OUT PortA,AL     ; Input 00H to DAC
          INC AL           ; Increment AL to increase ramp output
          CMP AL,0F2H      ; Is upper limit reached?
          JB BACK          ; If not, then increment the ramp
          JMP AGAIN        ; Else start again from 00H
CODE      ENDS
          END START
```

Program 5.13 *ALP for Generating Sawtooth Waveform Using AD 7523*

STEPPER MOTOR INTERFACING:

 A stepper motor is a device used to obtain an accurate position control of rotating shafts. It employs rotation of its shaft in terms of steps, rather than continuous rotation as in case of AC or DC motors. To rotate the shaft of the stepper motor, a sequence of pulses is needed to be applied to the windings of the stepper motor, in a proper sequence.

 The number of pulses required for one complete rotation of the shaft of the stepper motor is 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:

$X=360^0$/no. of rotor teeth

 After the rotation of the shaft through angel 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 fig.1.

 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.

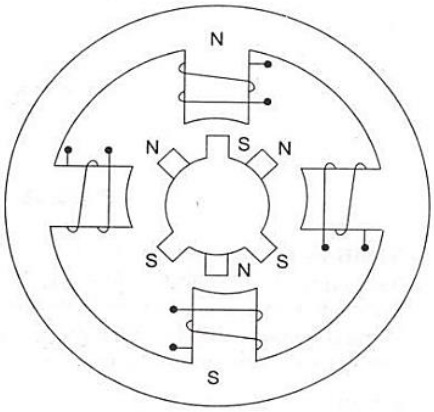


Fig.2 shows a typical winding arrangement of the stepper motor.

 Fig.3 shows conceptual positioning of the rotor teeth on the surface of rotor, for a six teeth rotor.

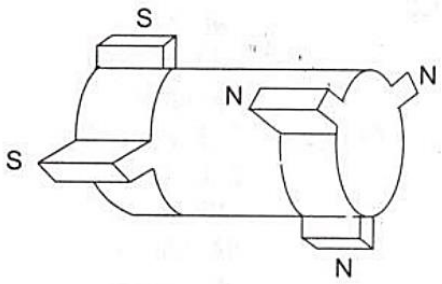


Fig.3 Stepper motor rotor

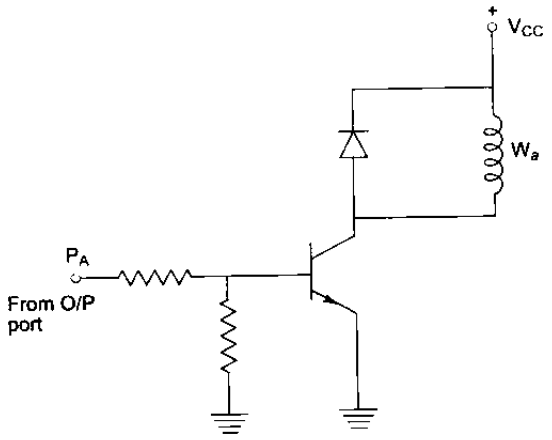


Fig.4 interfacing stepper motor winding.

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

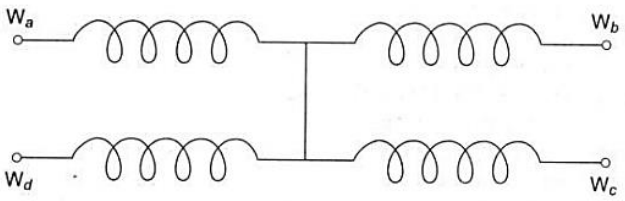


Fig.2 Winding arrangement of a stepper motor.

 Table.1 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 table2.

Table.1 Excitation sequence of a stepper motor using wave switching scheme.

Motion	step	A	B	C	D
Clock Wise Direction	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
Anti clock wise Direction	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

Table.2 An alternative scheme for rotating stepper motor shaft

Motion	step	A	B	C	D
Clock wise Direction	1	0	0	1	1
	2	0	1	1	0
	3	1	1	0	0
	4	1	0	0	1
	5	0	0	1	1
Anti clock wise Direction	1	0	0	1	1
	2	1	0	0	1
	3	1	1	0	0
	4	0	1	1	0
	5	0	0	0	0

EXAMPLE : Design a stepper motor controller and write an ALP to rotate shaft of a 4 phase stepper motor in clockwise 5 rotation and in anticlockwise 5 rotation

```
ASSUME CS : CODE
CODE SEGMENT
START:    MOV AL, 80H
          OUT CWR, AL
          MOV AL, 88H          ; Bit pattern 10001000 to start
          MOV CX, 1000         ; the sequence of excitation
AGAIN1:   OUT PORT A,AL        ; from WA. Excite WA, WB,
          CALL DELAY           ; WC and WD in sequence with delay. For 5 clockv
          ROL AL, 01           ; rotations the count is 200*5 = 1000.
          DEC CX               ; Excite till count = 0.
          JNZ AGAIN1
          MOV AH, 88H          ; Bit pattern to excite WA.
          MOV CX, 1000         ; Count for 5 rotations
```

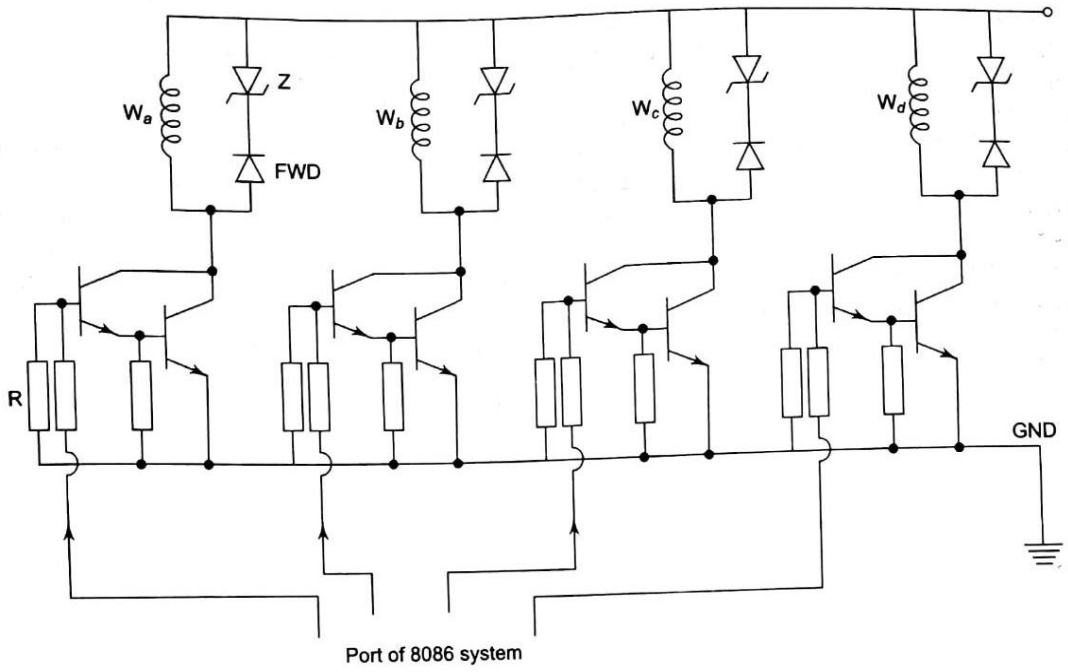


Fig. 5.51 *Stepper Motor Windings Connections*

```
AGAIN2:   OUT PORTA,AL         ; Excite WA, WB, WC, and WD.
          CALL DELAY           ; Wait.
          ROR AL, 01           ; Rotate bit pattern right to obtain anticlockwise
          DEC CX               ; motion of shaft. Decrement count.
          JNZ AGAIN2           ; If 5 rotations are completed
          MOV AH, 4CH          ; return to DOS else
          INT 21H              ; Continue
CODE      ENDS
          END START
```