1.1 GENERAL

Today microprocessors are very popular among the scientists and engineers. The electronics was mainly hardware oriented. Till early seventies and the logic IC's had fixed inbuilt logic blocks, using which a complex system could be built. A new device took the lead towards combination of software and hardware, which is microprocessor [1-3].

The microprocessors are a hardware device that can be controlled by software. The microprocessor, which has thousands of electronic logic elements on a chip, has lead to an industrial revolution. The microprocessor applications have covered almost every conceivable field and have started finding a place in household appliances like office equipments, communication systems, industrial processes and control systems entertainment and educational equipments [4, 5].

There are many microprocessors available from different manufactures like 8085, 8086, Z80, 6800, 8088, and 68000 which are 8-bits, 16-bits or 20-bits configurations. For any microprocessor some common features determine its power and capability. They are data bus width, address bus width, control bus, interrupt, processing capability, instruction set capability etc. The processor that is most suitable for a particular operation can be selected amongst available microprocessors after a comparative study of these common features. The power of microprocessor, availability of microprocessor and its support chips documentation and software, application update etc. are all very essential in determining the type of microprocessor to be used in an application [6, 7].

In 1960's, the million dollar computing capacity is available in integrated circuit called microprocessor. A Computing designed
using the microprocessor is called a computer. The use of Microprocessor is making an impact of an every aspect of a lives and it will play a significant role in industrialist society. The advancement in civilization always urged man to develop better and better machine.

The integrated circuit technology leads to the ability to integrate larger and larger number of logic gates on a single integrated circuit chip. The integration are divides into four categories [8].

(a) Small scale integration (SSI)

(b) Medium scale integration (MSI)

(c) Large scale integration (LSI)

(d) Very large scale integration (VLSI)

(a) SSI

The small-scale integration allowed only 12 logic gates to the integrated on the single chip. These IC's provides basic logic function like AND, OR, NOT along with the basic storage elements like D and J K type Flip flop. This type of integration is use only in the traditional switching circuits.

(b) MSI

This type of integration allowed from 13 to 99 logic gates on a single chip. There are limited number of pins and used as decoder and register comparator.

(c) LSI

LSI integrates 100 to 1000 of logic gates on to the single chip. The number of complexes fixed function is limited and are definable by the user of the LSI device.
(d) VLSI

Very large scale integration, integrate 1000 and above logic gates which integrated on the single chip.

1.2 MICROPROCESSOR

The microprocessors are digital devices able to receive information in the digital form, process this information according to a stored program and output information in the form of digital signal. A storage device whose contents can be read and altered at specific address (RAM) is used for storing temporary data. Storage devices, whose contents cannot be normally altered (ROM) may also be required for storing program or data that do not change and must always be available to the microprocessors. Both ROMs and RAMs are IC (Integrated circuit) packages. The microprocessors are compared of a register section, arithmetic and logic unit (ALUs) and a control unit. Depending the register section, the microprocessors can be classified either as an accumulator based or a general-purpose register (GPR) based machine [9, 10].

The basis elements of microprocessors are the central processing unit (CPU), program counter, stack, memory, I/O device. The CPU translates instruction, performs arithmetic and logic operation and temporarily stores instruction and data in its internal high-speed register. The memory stores program and data. The I/o unit interface the microprocessors with internal devices such as keyboard, display etc. with the advent of semi conductor technology it is possible to integrate the CPU in a single chip. Metal oxide semi conductor (MOS) technology is typically used to fabricate the microprocessors. Are PMOS, NMOS, CMOS, BIPOLAR, CCD and other technologies? MOS technology is used to create transistor and other component on the surface of small piece of silicon chip. This technology is currently used to achieve the densities that
characterize LSI microprocessors. Microprocessor is an integrated circuit like other integrated circuit but it is a powerful IC because it contains the ALU (arithmetic to logic unit), Register (for keeping data) and control unit (to control all devices) [11, 12].

In the earlier time when microprocessor were invented they are using PMOS technology, which provides low costs slow speed, and low output current. As a consequence these microprocessors are not compatible to TTL logic. After 1973 the microprocessors are designed using NMOS technology which offered the high speed, higher density and are compatible to TTL logic after 1978 microprocessors are designed using HMOs technology and provides the following advantages over NMOS. One factor which is very important is the Speed power factor of HMOS, which is 4 times better than that NMOS. The speed power factor is nothing but the multiplication of speed and power and having units in picojule. Circuit densities produced by HMOS one twice than that of NMOS. It is 4128 $\mu m^2$/gate in HMOS as compare to the 1852 $\mu m^2$/gate in NMOS [13].

1.3 EVOLUTION OF MICROPROCESSORS

The microprocessors are going to manufacture from 1971, therefore, they are classified for their generation [14].

1.3.1. First Generation

At the end of 70s a group of engineers developed a single chip, which is capable of processing data, and was given the name, processors chip. The design team headed by Ted Hoff on Intel Corporation developed. The first such controlled processors is given the name Intel in the year 1969. Intel Corporation started marketing microprocessors in the year 1971. The first microprocessor was a 4-bit microprocessors having 16-pins housed in a single chip of PMOS technology and also called the nibble
microprocessor. This was the first generation of microprocessor. The 4-bit microprocessors worked with 4-bit word. The Intel 4004 along with few other devices was used for making calculators. The ability of changing function of any system by just changing the programming rather than redesigning the hardware [15].

The Intel 8008 was developed in the year 1972 that worked with 8-bit word. It required about 20 or more additional devices to design the function of C.P.U. A few first generation microprocessors are listed in table 1.1.

1.3.2 Second Generation Microprocessors

The second-generation microprocessors are based on NMOS technology appeared in the market in the year 1973. Second generation microprocessors and lists in table 1.1.

The Intel 8080 is an 8-bit microprocessors of NMOS Technology was developed in the year 1974 which required only two additional devices to design a functional CPU. The 8080 microprocessor was much faster and had many more instruction than 8008 that facilitated the programming. The advantages of second generation microprocessor is:

1. Large chip size (170*200 unit) with 40-pins.
3. Ability to address large memory space (64-KB) and I/O Ports (256).
5. Dissipate less power.
7. Cycle time reduced to half (1.3 to 9 sec).
8. Size (170*200) mil with 40-pins.
9. Less Support chip required.
10. Used single power supply
11. Faster operation.
### Table 1.1: List of various microprocessors

<table>
<thead>
<tr>
<th>Microprocessors</th>
<th>Word size</th>
<th>Microprocessors</th>
<th>Word size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel 4004 &amp; 4040</td>
<td>4 bit</td>
<td>Intel 8008</td>
<td>8-bit</td>
</tr>
<tr>
<td>FAIR CHILD PPS-25</td>
<td>4 bit</td>
<td>NATIONAL IMP-8</td>
<td>8-bit</td>
</tr>
<tr>
<td>NATIONAL IMP -4</td>
<td>4 bit</td>
<td>ROCKWELL PPS-8</td>
<td>8-bit</td>
</tr>
<tr>
<td>ROCKWELL PPS-4</td>
<td>4 bit</td>
<td>AMI 7200</td>
<td>8-bit</td>
</tr>
<tr>
<td>MICROSYSTEM MICROCONTROLLER-1</td>
<td>4 bit</td>
<td>MOSTEK 5065</td>
<td>8-bit</td>
</tr>
<tr>
<td>Intel 8080/8085</td>
<td>8-bit</td>
<td>SIGNETICS 2650</td>
<td>8-bit</td>
</tr>
<tr>
<td>FAIRCHILD F8</td>
<td>8-bit</td>
<td>ZILOG Z-80</td>
<td>8-bit</td>
</tr>
<tr>
<td>MOTOROLA M 6800</td>
<td>8-bit</td>
<td>RCACOSMAC</td>
<td>8-bit</td>
</tr>
<tr>
<td>NATIONAL CMP-8</td>
<td>8-bit</td>
<td>INTERSIL 6100</td>
<td>12-bit</td>
</tr>
<tr>
<td>MOS TECH. 6500</td>
<td>8-bit</td>
<td>TOSHIBA TLCS-12</td>
<td>12-bit</td>
</tr>
</tbody>
</table>
1.3.3 Third Generation Microprocessor

The single chip third generation microprocessors having 64-pins started with the introduction of 16-bit Intel 8086 in the year 1978. The other important third generation microprocessor was Zilog Z-8000, Motorola M68000, National NS16016, and Texas Instrument TMS 99000 Series etc. The 16-bit microprocessors using the HMOS technology and achieved enhanced performance Parameters with respect to the 8-bit microprocessors. In additional to enhanced performance, it contained the process of multiplication and decision. The memory addressing capabilities were increased in the microprocessor of the order of 1M Byte to 16 M byte and also having the verity of flexible and powerful addressing modes.

The Intel 8088 was Identical to 8088 but for the 8-bit data bus. Hence 8088 could read or write 8-bit data at a time to or from the memory. The Intel 80186 and 80188 were the improved version of Intel 8086 and 8088, respectively. In additional to 16-bit CPU, the 80186 and 80188 had programmable peripheral devices integrated on the same Package. The program written for 80186 and 80188 may not work well on 8086 and 8088, but these written for 8086 and 8088 worked much difficulties on 80186 and 80188. These means they were upward compatible with 8086 and 8088. The Intel 80286 was the advanced version of 80186. It is designed for use in multi-user/Multitasking environment [16].

1.3.4 Fourth Generation Microprocessor

The single chip 32-bit microprocessor was introduced in the year 1981 by Intel as IAPX432. The other 4th generation microprocessors were; Bell single chip Bellmac-32, Hewlett-Packard National NS16032, Texas Instrument 99000, Motorola 68020 and 68030 [17].
The power of the microprocessors goes on increasing with the advancement in the integrated circuit Technology. The VLSI Technology eliminated in the extremely Complex microprocessors with as many as on billion transistors on a single chip. The Intel in the year 1985 announced and is also a 32-bit microprocessors most of the microprocessors were manufactured with HMOS (high density short channel MOS) Technology because of the following advantage [18].

1.4 PIN CONFIGURATION (8085)

The Intel 8085A is a 40-pin DIP dual-in-line Package. This is a HMOS device, which implemented with approximately 6200 transistors on 164222 check chip. It requires power supply of +5V and also has its built-in clock and control circuit. In this microprocessors the data can the transfer to and form the data bus, therefore it is capable of addressing $2^{16}=65,536=64K$ memory locations. The connected crystal oscillator produced 6.25MHz frequency from which 3.125 MHz is used by the microprocessors and its driving frequency is 1 MHz.

1.4.1 X1 and X2 (PIN 1 and 2)

The 8085 have an inbuilt oscillator on the chip. The crystal oscillator and LC tank network or RC network to control the frequency of oscillator. It is connected across X1 and X2 terminals. The oscillator output from the Schmitt trigger drives is feed to a FF which divides it by a factor of 2 and produces two phase clock signal as shown in figure 1.2. These clocks drive the internal circuit of 8085 A. The TTL level clock output CLK (OUT), drive from phase of the internal clock, Provides a clock signal that can be used for synchronization.
Fig. 1.1: Pin diagram of 8085 microprocessor
Fig. 1.2: Schematic diagram of clock signal

Fig. 1.3: Schematic diagram of SOD line
1.4.2 RESE TOUT (PIN-3)

This signal is taken to reset peripherals used in microprocessors system. A high logic on this pin indicates that the up is being RESET, if means that all resistors and counters are being RESET to zero. This signal is also used to RESET the other devices. This output remains high as long as the RESET input is kept low. When the power is switched ON the whole system in including the 8085A are RESET or initialized. A high on Pin 3 indicates that up are being RESET, all register and counters. After the RFSET OUT becomes low the processing starts.

1.4.3 SOD (PIN-4)

Serial output data (SOD) line provides a 1-bit output port on the 8085 A. The SIM (send interrupt mark) instruction is used to output data serially from the SOD line. As shown in figure 1.3, the SIM instruction, the value of D7 (bit-7) of the accumulator is loaded is SOD patch only when D6 (bit-6) of the accumulator is logic-1 (high). If D6=0, the SOD latch remains unattended when the 8085A is reset, the SOD patch is also set to logic zero automatically (Fig 1.3).

1.4.4 SID (PIN-5)

It is serial input data pin, which is 1-bit input port of the 8085A. A RIM (read interrupt mask) instruction is used to transfer the input present of this pin to bit-7 of the accumulator.

1.4.5 INTERRUPTS (PIN-10)

The 8085A chip has five interrupts pins are present on the following pins they are TRAP (PEN 6), RST 7.5 (PIN 7), RST 6.5 (PINS 8), RST 5.5 (PIN9), INTR (PIN 10) All these interrupts are used in this parity order.
TRAP has the highest priority and the INTER has the lowest. The TRAP is the non-maskable first priority interrupts. The created pulse is edge and the level sensitive.

The second priority is assigned to RST 7.5, which is positive edge sensitive where as the RST 6.5 and RST 5.5 arc level sensitive.

The INTER is the lowest priority interrupt and working when a request is demanded when this pin goes high it indicates that peripheral devices as requesting. At this request the microprocessor execute the current distinction the microprocessors to stop and switch over to the other work for peripheral. These interrupts are vectorized because their memory locations are fixed.

The program control is transformed to the corresponding memory location.

1.4.6 INTA (PIN-11)

It stands for interrupt whenever the interrupt request is demanded the microprocessors send acknowledge. A low (0) on this pin indicates that out an acknowledgement through this pin. This means that the request from the user is granted.

1.4.7 Multiplexed address and Data bus (Pin 12-19)

These buses are also called the AD (Address data bus) which are from pins 12 to 19. These buses are bi-directional, so that the data is transfer to and from the microprocessor. As shown in figure 1.4, these buses are connected to the latch 8282 which contain the D-type FF and higher order AD7 pin is connected to the D impact of this. The clock of the FF is connected to the ALE pin of the microprocessor, which creates the lower order address bus. Otherwise they are used as data bus.

The lower order address byte can be used to address $2^8 = 256$ I/o devices also.
1.4.8 LATCH

The LATCH is used to connect the output devices to the microprocessor. The data is available for a few uses on the AD buses. The data in the form of address should be present on the address bus till the reading or writing operation is not completed. If it is not so the wrong data transfer will take place. The lower order address byte is present in the 1st clock of the machine cycle therefore it (A7 - A0) should be latched to hold the lower order address byte as shown in the figure 1.5. This device contains the tri state buffer. The STB is directly connected to the CLK of the FF and is active high when STB goes low the data is latched from the data bust. The output enable is always at active low, which enables the tri state buffer to output data for output devices.

1.4.9 Address Bus (PIN 21-28)

The address buses of 8085 a microprocessor is unidirectional because address is sends only from the microprocessor. These 8 bits are used for memory address and they remain in the high impedance state during HOLD, HALT and RESET modes.

1.4.10 Control and Status Signals

The RD and WR are control signals and IO/m stand and so are status signals. The ALE is also a special control signal.

1.4.10.1 So, S1, and IO/m pins

So and S1 are two status signal similar to IO/m. The IO/m, so and S1 are used to identify various operation as shown in table 1.2.
Fig. 1.4: Figure showing multiplex address data bus

Fig. 1.5: Figure showing D LATCH
### Table 1.2: Status and control signals of 8085 A

<table>
<thead>
<tr>
<th>Machine cycle</th>
<th>Io/m</th>
<th>S1</th>
<th>S0</th>
<th>Control Signals</th>
</tr>
</thead>
<tbody>
<tr>
<td>OP code Fetch</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>RD = 0</td>
</tr>
<tr>
<td>Memory read</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>RD = 0</td>
</tr>
<tr>
<td>Memory write</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>WR = 0</td>
</tr>
<tr>
<td>I/o Read</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>RD = 0</td>
</tr>
<tr>
<td>I/o Write</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>WR = 0</td>
</tr>
<tr>
<td>Interrupt Ack</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>INTA = 0</td>
</tr>
<tr>
<td>HALT</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>HOLD</td>
<td>Z</td>
<td>X</td>
<td>X</td>
<td>RD.WR = Z</td>
</tr>
<tr>
<td>RESET</td>
<td>Z</td>
<td>X</td>
<td>X</td>
<td>INTA = 1</td>
</tr>
</tbody>
</table>

**Legends:** Z = HIGH IMPEDANCE STATE; X = UNSPECIFIED
1.4.10.2 ALE (PIN 30)

A memory chip contains a MAR (memory Address Register), which is also called the address latch. The ALE stores the address from the address bus when it is connected to the memory chip. The falling edge of the ALE signal loads (strokes) the address bus into the MAR of the memory chip.

The ALE is used to show the information carried on the multiplexed address/data bus. This is the lower order address byte when a high condition appears on the line. This control line has the advantage to latch the lower order address from it during the first clock of the machine cycle as shown in figure 1.6. During the this state when ALE goes high, the latch is seen as transparent because the output changes in accordance with the input data. During the tri-state the output of the latch is a data byte when ALE becomes low, the data byte is latched until the next ALE. The output of the latch represents low order address depicts the entire address bus after the latching operation.

1.4.10.3 WR (PIN - 31)

The WR indicating that write operation worked at low signal, which is generated by the microprocessor (to write) data into I/O devices or memory.

1.4.10.4 : RD (PIN-32)

A low condition RD line generated by the microprocessor to indicate the microprocessor reads the data from the I/O devices or from memory locations.

1.4.10.5 IO/M (Pin - 34)

The IO/M is used to define that the address on the address line (bus) is either for I/O devices or memory location. A high condition on this pin indicates that the address on the address bus is meant for I/O devices otherwise for memory.
Fig. 1.6: Figure showing multiplex address data bus control
1.4.11 Ready (PIN-35)

There is a incompatibility between the memory, I/O devices and the speed of the microprocessor operation therefore a signals are generated, which is high to match the cycle and after that an READY signal is generated. This condition is indicated by input ring a low condition on this pin with READY = 0, the microprocessor waits till Ready = 1. As soon as Ready = 1, the microprocessor received the data which is available from the memory or I/O devices.

1.4.12 Reset In (PIN-36)

When this signal is low at least for 600 ns, the microprocessor is forced to do the following steps for 3 clock cycle.

(PC) - 20

Introduction Register cleared

All Interrupts (Except TRAP) disabled.

SOD - 0

Data, address and control bus are floated.

1.4.13 CLK OUT (PIN-37)

There is a inbuilt oscillator to proceed clock pulse, which is coming on to from the chip. The CLK drives the peripherals to synchronize their timings.

1.4.14 HOLD (PIN-38)

A high condition hold the microprocessor to 1 clock cycle for DMA transfer.
1.4.15 HLDA (PIN-39)

HLDA stands for hold acknowledge. A high condition on this pin indicates that the microprocessor has received the information of request from the I/o devices and will relinquish the data, address and control buses after completing the current instruction, i.e. after completion of the current bus transfer, if any.

1.4.16 Vcc (PIN-40)

An external D.C. supply voltage of +5v (17 MA) is connected to this pin for the operation of the 8085A.

1.5 THE 8086

A basic 8086 microprocessor-based systems consists of microprocessor, memory devices, I/O ports, clock generators, bus controller, latches, butters, and decoders. The system uses bus controller, latches and buffers to provide separated and fully buffered address, data and control buses for interfacing memory and I/O devices to the microprocessors. Several techniques are employed to interface the memory and I/O devices to microprocessor. Decoders assign a range of addresses for the memory and I/O devices. Digital to analog converters and analog to digital converters are interfaced to microprocessors based systems for several instrumentation applications (19).

1.5.1 Pin Configuration

The microprocessor can be operated in either minimum or maximum mode by applying power logic to MN/MX input. It is operated in the minimum mode in single processor systems and in maximum mode in multiprocessor systems where more than one processor is used. Pins 24 to 31 have different functions in maximum and minimum modes as shown in figure 1.7.
Fig 1.7: Pin diagram of 8086 microprocessor
1.5.1.1 Common signals

**MN/MX (pin 33):** minimum / maximum line; logic high selects the minimum mode and low selects maximum mode of operation for the microprocessor. It is an input line.

**AD15 – ADO (pins 39 and 16-2):** Multiplexed Address / Data lines; when ALE is high, the lines carry A19 – A16 bits of address. When ALE is low, the lines carry S6 – S3 status signals. The address / status lines are outputs.

**BHE / S7 (pin 37):** Multiplexed bus high enable / status line; when ALE is high, the line serves as BHE. When ALE is low, it serves as status signal S7. The BHE / S7 is an output line. The BHE signal is low if ADO–AD8 lines are to be used in data transfer operations, else high. The S7 signal is always high.

**RD (pin 32):** The microprocessor outputs low on this Read line when it reads data from memory or I/O devices. It is an output and active low line. It goes to high – impedance state during hold acknowledge.

**READY (pin 22):** The external slow devices in the system control the microprocessor through this Ready line to remain idle till the device gets ready for data transfer. It enables the microprocessor to insert wait states during memory I/O access. It is an input and active high line.

**NMI and INTR (pin 18 and pin 17):** The non-maskable interrupt and interrupt Request are hardware interrupt input lines. External devices in the system request the attention of the microprocessor through these lines. The processors ignore interrupt through INTR input if interrupt enable flag IF is reset. The interrupt inputs are active high and inputs lines.
TEST (pin 23): If the Test signal is low, the WAIT instruction in the program being executed is treated as NOP. If high, the microprocessor when executes the WAIT instruction wait till the TEST signal goes low. It is active low input line.

RESET (pin 21): High on the Reset input causes the microprocessor to reset all its activities and begin the execution from FFFF: 0000 H location. It is an active high Input line.

CLK (pin 19): The clock signal provides the required timing signal. It is an input line.

VCC, and GND (pin 40 and 20): The supply, +5V and Ground lines.

1.5.1.2 Minimum mode signals

M/IO (pin 28): The M/IO signal differentiates the memory and I/O accesses. Logic high enables memory and 10w enables I/O device. The signal goes high–impedance during Hold Acknowledge. It is an output line.

WRC (pin 29): The write signal is active when the microprocessor writes data into memory or I/O device. It goes to high – impedance state during Hold Acknowledge. It is an active 10w output line.

AIE (pin 25): The address latch enable signal is used to multiplex address, data and status signals on AD15-ADO, A19/56-A16/53 and BHE/S7 lines. The signal is high in first I state of all machine cycles. It is an active high output line.

DT/R, DEN (pin 27 and 26): The data transmit/receive indicates the direction of data flow on data bus. It is high when the processor is transmitting and low when the processor receiving data, it is an
output line. The data enable signal enables external data bus butters for data bus activity on AD15–ADO lines. It is an active–low output line.

**INTA (pin 24):** The microprocessor responds to interrupt requests through INTR input the interrupt Acknowledge Signal. It is an active low output line.

**HOLD and HLDA (pin 31 and 30):** The other bus masters in the system request the microprocessor the release the control of system bus through the hold request with the Hold Acknowledge signal and tri–states the system bus. It is an active high output line in maximum mode Signals

**RO/GT1, RO/GT0 (pin 31 and 30):** The Request/Grant Signals. The other bus masters in the system request the microprocessor to release the local bus and the microprocessor relinquishes the bus and in forms the requesting master. The lines are active low and bidirectional.

**OS1–OS0 (pin 25 and 24):** The microprocessor indicates the queue states on these lines.

**S2, S1 and S0 (pin 28, 27 and 26):** The Status signals indicate the type of current bus cycle. 000 – interrupt Acknowledge 001 I/O Read 010 – I/O write, 011 – Hold, 100 – OP code fetch, 101–memory write, 111 – inactive. The lines are active low output line.

**LOCK (pin 29):** The LOCK signal prevents other bus masters going control of system busses. The signal is activated by a Lock instruction Prefix and remains activated till the end of the next instruction. It is an active – low output line.
1.5.2 Architecture

The 8086 are Intel's first 16-bit microprocessor. Its design is based on the 8080 but it is not directly compatible with the 8080. The HMOS technology is used to design 8086, which contain approximately 29000 transistors. The 8086 is available in 40-pin Dual in line package and requires a single 5 V power supply, with three different clock speeds. The standard 8086 runs at 5 MHz internal clock frequency, whereas the 8086-2 and 8086-4 run at internal clock frequencies of 8 and 4 MHz, respectively. The 8086 divides the external clock connected at CLK pin internally connected. The 8086 have a 20-bit address bus therefore; it can directly address one megabyte of memory. The 8086 family consists of two types of 16 – bits microprocessors – the 8086 and 8088 (20).

The 8086 is divided into two units one is interface unit (BIU) and other is called execution unit (EU) as shown in figure 1.8. The 8086 is also uses the segmented memory.

The BIU perfetches six instructions bytes from memory and queues them in order to speed up instruction execution.

The 8086 access a 16 - bit word to or from memory and it can read a 16 - bit word in one operation. If the first byte of the word is at an odd address. In this case, it decodes the unwanted byte of each.

The main difference between the 8086 and 8088 is that the 8088 have an 8-bit data path to memory and I/O, while the 8086 have a 16-bit external data path.
The Intel 8086 microprocessor is a 16-bit processor. The internal registers and data bus are 16-bit wide and it also uses 16-bit instructions. The microprocessor can read or write a 16-bit data to or form the memory. It can also read or write a 16-bit data to and form the I/O devices. It uses one I/O read or I/O write operation. The address bus is 20-bit wide (A0-A19) to make a physical address. The microprocessor can address up to 1 MB memory locations (21).

1.5.2.1 Bus Interface Unit (BIU)

The BIU’S instruction queue is a First – In – First – Out (FIFO) group of registers in which up to 6 bytes of instruction code are prefetched from memory ahead of time. This is done in order to speed up program execution by overlapping instruction fetch with execution. This mechanism is known as pipelining. The BIU fetches instructions, reads operands and writes results. The BIU provides all external bus operations. The BIU contains segments registers and instruction pointer, instruction queue and address generation or bus control circuitry to provide functions such as the fetching and queuing of instructions and bus control.

1.5.2.2 Queue

The BIU contains first in first out register of size 6 bytes in length called queue. The BIU prefetches instructions from memory and stores them in the queue. The EU, after completing the execution of current instruction, gets the next instruction from the queue and executes.

Fetching instructions from memory while the execution of an instruction is in progress is called as pipelining. This method increasing the speed of the processing.
1.5.2.3 Segments and offsets

The 8086 microprocessor with its 20-bit address bus can access up to 1 MB of memory, it groups 64 KB as one segment and uses a 16-bit binary number called segment base to identify it. The 8086 use another 16-bit address called offset to locate one of the 64 KB within the segment. The start of a segment is the address of the location at an offset of FFFFH in the segment. The 20 – bit physical address of a location is represented in segment.

1.5.2.4 Segment registrar

The BIU contains four 16 bit segment registers designated as code segment (CS), stack segment (SS), extra segment (ES) and data segment (DS) registers. The segment registers hold the base or start of the respective segments. The 8086 can access a maximum of four segments and work within these four segments at any one time.

CS Register

The 8086- store instruction code segment, which holds the segment base of code segment. The BIU has a 16-bit Instruction Pointer (IP) register. The purpose of the IP register is to hold the offset of next instruction byte to be fetched from memory within code segment. The BIU generates 20-bit address using contents of CS and IP registers, and sends to memory to fetch the instruction byte.

SS Register

Stack is portion of memory reserved to store addresses and data temporarily while the program is executing. The 8086 reserves one segment (64 KB) for stack purposes; SS in BIU holds the base of stack segment.

DS Register

The 8086 stores data in data segment. DS holds the segment base of data segment. BIU uses the contents of other registers
holding the offset 20-bit physical address to access a data in the data segment.

**ES Register**

The ES register holds the segment base of an additional data segment. BIU generates the physical address using the offset and generate 20-bit physical address to default by some string instructions.

The special feature of three 16-bit register-BP, SI, and DI- in execution unit is that they hold offset of data word in one of the segment. BP usually holds current offset of top of the stack in a stack segment for future use. SI holds offset of data in the segment and DI holds offset of data in extra segment.

The main advantages of segmentation schemes is that it uses separate blocks for code, data and stack in memory. Programs are relocatable. Near or short type jump instructions specify only the relative addresses and not the absolute addresses for branching instructions can be loaded anywhere in memory and executed properly. The segment registers determine the locations for the program and the contents of segment registers can alone be changed to load the program at any designed locations. Hence, the programs are said to be relocatable.

Segment offset representation uses only 16-bit addresses and not directly the 20-bit addresses. It enables easy interface between 16-bit internal registers and 8-bit or 16-bit memory or I/O devices.

It enables multiprogramming. Each program in memory has a separate set of logical segments for code and data. Switching from one program to another program in memory is made possible by simply changing the contents of four segment registers.
1.5.3 Execution Unit (EU)

The execution unit (EU) decodes and executes instructions. A decoder in the EU controls system, which translates instructions. The EU has a 16-bit ALU for performing arithmetic and logic operations.

The EU has nine – 16 bit registers, AX, BX, CX, DX, SP, SI, and DI, and flag register and the 16-bit general registers. AX, BX, CX and DX can be used as two 8 – bit register namely AH, AL, BH, BL, CH, CL, DH, DL.

1.5.3.1 The AX register

It is a 16-bit register, called 16-bit accumulator and can be used as 8-bit accumulator, Al and AH for high order and lower order 8-bits. The multiplication and division instructions also use the AX or AL. The AL register is the same as the 8085 A register.

1.5.3.2 The BX register

A base register is called BX register. This is only general-purpose register whose contents can be used for addressing 8086 memories.

1.5.3.3 The CX register

A counter register is called CX register. Because some instructions such as SHIFT, ROTATE and LOOP, use the contents of as a counter.

1.5.3.4 The DX register

A data register is called the DX register. It is used to hold the high 16 bit data (result) in 16 × 16 multiplication or the high 16–bit divided (data) before a 32 ÷16 division and the 16 – bit remainders after the divisions.
1.5.3.5 The Two-Pointer register

Stack pointer (SP) and base pointer (BP) are the two-pointer register, which are used to access data in the stack segment. The SP is used as an offset from the current SS during execution of instructions that involve the stack segment in external memory. The SP contents are automatically updated due to execution of a POP or PUSH instructions. The BP contains an offset address in the current SS. This offset is used by instructions utilizing the based addressing mode.

1.5.3.6 Two index registers

There are two index registers one is source index (SI) and other is destination index (DI) are used in indexed addressing. During an tring processes the SI and DI index register are used with DS and ES respectively.

1.5.3.7 The flag register

The flag register in the EU holds the status flag after an ALU operation. The EU sets or resets these flags to reflect certain properties of results.

1.5.4 Flag of 8086

The 8086 has six bit status flags

**AF (Auxiliary Carry Flag)** AF flag is set if there is carry from the low nibble into the high nibble or a borrow from the high nibble into low nibble of the low order 8 bits of a 16 – bit number. This flag is used by BCD arithmetic instructions: otherwise, AF is zero.

**CF (Carry Flag)** Carry flag is set if there is carry from addition or a borrow from subtraction.

**OF (Overflow Flag)** Overflag is set if there is an arithmetic overflow. If the size of result exceeds the capacity of the destination
location. An interrupt on overflow instruction is available to
generate on interrupt in this situation otherwise it is zero.

**SF (Sign Flag)** Sign flag is set if the most significant bit of the
result is one otherwise, it is zero.

**PF (Parity Flag)** Parity flag is set if the result has even parity. PF is
zero for odd of the result.

**ZF (Zero Flag)** Zero flag is set if the result is zero. ZF is zero for a
non-zero result.

### 1.6 ADDRESSING MODES

The 8086 provides various addressing modes to access
instructions operands. Operands may be contained in registers.
Within the instruction opcode, in memory or in I/O ports. The 8086
addressing modes can be classified into following group (22).

#### 1.6.1 Register Addressing

Transfer a copy of a byte or word from the source register or
memory location to the destination register or memory location.
Register addressing is the most common form data addressing. The
8086 contain the following 8-bit registers used with register
addressing AH, AL, BH, CH, CL, DH and DL and also present as the
following 16-bit registers AX, BX, CX, DX, SP, BP, SI and DI.

#### 1.6.2 Intermediate Addressing

The term intermediate implies that the data immediately
follow the hexadecimal opcode in the memory. The immediate data
are constant data while the data transferred from a register are
variable data. Immediate addressing operates upon a byte or word
of data. If hexadecimal data begin with a letter, the assembler
requires that the data start with a 0 (zero). Table 1.4 shows many
different variations of MOV instructions that apply immediate data.
### Table 1.3: Register Addressed Instructions

<table>
<thead>
<tr>
<th>Assembly Language</th>
<th>Size</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV AL, BL</td>
<td>8 - bits</td>
<td>Copies BL into AL</td>
</tr>
<tr>
<td>MOV CH, CL</td>
<td>8 - bits</td>
<td>Copies CL into CH</td>
</tr>
<tr>
<td>MOV AX, CX</td>
<td>16 - bits</td>
<td>Copies CX into AX</td>
</tr>
<tr>
<td>MOV SP, BP</td>
<td>16 - bits</td>
<td>Copies BP into SP</td>
</tr>
<tr>
<td>MOV DS, AX</td>
<td>16 - bits</td>
<td>Copies AX into DS</td>
</tr>
<tr>
<td>MOV SI, DI</td>
<td>16 - bits</td>
<td>Copies DI into SI</td>
</tr>
<tr>
<td>MOV BX, ES</td>
<td>16 - bits</td>
<td>Copies ES into BX</td>
</tr>
<tr>
<td>MOV ECX, EBX</td>
<td>32 - bits</td>
<td>Copies EBX into ECX</td>
</tr>
<tr>
<td>MOV ESP, EDX</td>
<td>32 - bits</td>
<td>Copies EDX into ESP</td>
</tr>
<tr>
<td>MOV ES, DS</td>
<td>-</td>
<td>Not Allowed</td>
</tr>
<tr>
<td>MOV BL, DX</td>
<td>-</td>
<td>Not Allowed</td>
</tr>
</tbody>
</table>

### Table 1.4: Instruction immediate address

<table>
<thead>
<tr>
<th>Assembly Language</th>
<th>Size</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV BL, 44</td>
<td>8 - bits</td>
<td>Copies a 44 decimal into BL</td>
</tr>
<tr>
<td>MOV AX, 44H</td>
<td>16 - bits</td>
<td>Copies a 0044H into AX</td>
</tr>
<tr>
<td>MOV SI, 0</td>
<td>16 - bits</td>
<td>Copies a 0000H into SI</td>
</tr>
<tr>
<td>MOV CH, 100</td>
<td>8 - bits</td>
<td>Copies a 100 decimal into CH</td>
</tr>
<tr>
<td>MOV AL, A</td>
<td>8 - bits</td>
<td>Copies a ASC11A into AL</td>
</tr>
<tr>
<td>MOV AX, AB</td>
<td>16 - bits</td>
<td>Copies a ASC11 BA into AX</td>
</tr>
<tr>
<td>MOV ESI, 12</td>
<td>32 - bits</td>
<td>Copies a 12 decimal into ESI</td>
</tr>
<tr>
<td>MOV EAX, 100Y</td>
<td>32 - bits</td>
<td>Copies a 100 binary into EAX</td>
</tr>
</tbody>
</table>
1.6.3 Direct Data Addressing

Direct data addressing is applied to many instructions in a typical program. There are two basic forms of direct data addressing: (1) direct addressing, which applies to MOV between a memory location and AL, AX or EAX and (2) displacement addressing, which applies to almost any instruction in the instruction set. In either case, the address formed by adding the displacement to the default data segment address or an alternate segment address.

1.6.4 Direct Addressing

Direct addressing, with a MOV instruction, transfer data between a memory location located within the data segment, and the AL (8-bit), AX (16-bit) register. A MOV instruction using this type of addressing is usually a 3-byte long instruction. The MOV AL, DATA instruction, as represented by most assemblers, loads AL from data segment memory location DATA (1234 H). Memory location DATA is a symbolic memory location, while the 1234 H is the actual hexadecimal location. With many assemblers, this instruction is represented as MOV AL, (1234 H) instruction.

The 1234 H is an absolute memory location that is not allowed by all assemblers programs.

1.6.5 Displacement Addressing

Displacement addressing is almost identical with direct addressing, except that the instruction is four bytes wide instead of three. This type of direct data addressing is much more flexible because most instructions use it.
### Table 1.5: Direct Addressed Instructions

<table>
<thead>
<tr>
<th>Assembly Language</th>
<th>Size</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV AL, Number</td>
<td>8-bits</td>
<td>Copies the byte contents of data segment memory location NUMBER into AL.</td>
</tr>
<tr>
<td>MOV AX, COW</td>
<td>16-bits</td>
<td>Copies the word contents of data segment memory location COW into AX.</td>
</tr>
<tr>
<td>MOV NEWS, AL</td>
<td>8-bits</td>
<td>Copies AL into data segment memory location NEWS.</td>
</tr>
<tr>
<td>MOV THERE, AX</td>
<td>16-bits</td>
<td>Copies AX into data segment memory location THERE.</td>
</tr>
</tbody>
</table>

### Table 1.6: Displacement Addressing Instructions

<table>
<thead>
<tr>
<th>Assembly Language</th>
<th>Size</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV CH, DOG</td>
<td>8-bits</td>
<td>Copies the byte contents of data segment memory location DOG into CH.</td>
</tr>
<tr>
<td>MOV CH, (1000 H)</td>
<td>8-bits</td>
<td>Copies the word contents of data segment offset address 1000 H into CH.</td>
</tr>
<tr>
<td>MOV ES, DATA 6</td>
<td>16-bits</td>
<td>Copies the word contents of data segment memory location DATA 6 into ES.</td>
</tr>
<tr>
<td>MOV DATA 7, BP</td>
<td>16-bits</td>
<td>Copies BP into data segment memory location DATA 7.</td>
</tr>
<tr>
<td>MOV NUMBER, SP</td>
<td>16-bits</td>
<td>Copies SP into data segment memory location NUMBER</td>
</tr>
</tbody>
</table>
1.6.6 Register Indirect Addressing

Register indirect addressing allows data to be addressed at any memory location through an offset address held in any of the following registers: BP, BX, DI and SI.

The data segment is used by default with register indirect addressing or any addressing mode that uses BX, DI, or SI to address memory. If register BP addresses memory, the stack segment is used by default. These are considered the default settings for these four indexes and base registers.

The operation of the MOV AX, [BX] instruction when BX = 1000 H and DS = 0100 H.

This instruction is shown after the contents of memory are transferred to AX is depicted in table 1.7.

1.6.7 Base-Plus-Index Addressing

Base-Plus-Index addressing is similar to indirect addressing because it indirectly addresses memory data. Thus type of addressing uses me base register (BP or BX) and one index register (DI or SI) to indirectly address memory. The base registers after holds the beginning location of a memory array, while the index register holds the relative position of an element in the array.
### Table 1.7: Register Indirect Addressing Instructions

<table>
<thead>
<tr>
<th>Assembly Language</th>
<th>Size</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV CX, [BX]</td>
<td>8-bits</td>
<td>Copies the byte contents of data segment memory location address by BX into CX.</td>
</tr>
<tr>
<td>MOV [BP], DL</td>
<td>8-bits</td>
<td>Copies DL into the stack segment memory location addressed by BP.</td>
</tr>
<tr>
<td>MOV [DI], BH</td>
<td>8-bits</td>
<td>Copies BH into the data segment memory location addressed by DI.</td>
</tr>
<tr>
<td>MOV [DI], [BX]</td>
<td></td>
<td>Memory to memory moves is not allowed except with string instructions.</td>
</tr>
<tr>
<td>MOV AL, [EDX]</td>
<td>8-bits</td>
<td>Copies the byte contents of the data segment memory location addressed by EDX into AL.</td>
</tr>
</tbody>
</table>

### Table 1.8: The Base-Plus-Index Addressing Instruction

<table>
<thead>
<tr>
<th>Assembly Language</th>
<th>Size</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV CX, [BX+DI]</td>
<td>16-bits</td>
<td>Copies the word contents of data segment memory location address by BX plus DI into CX.</td>
</tr>
<tr>
<td>MOV CH, [BP+SI]</td>
<td>8-bits</td>
<td>Copies the byte contents of the stack segment memory location addressed by BP plus SI into CH.</td>
</tr>
<tr>
<td>MOV [BP+SI], SP</td>
<td>16-bits</td>
<td>Copies SP into the data segment memory location addresses by BX plus SI.</td>
</tr>
<tr>
<td>MOV [BP+DI], AH</td>
<td>8-bits</td>
<td>Copies AH into the stack segment memory location addressed by BP plus DI.</td>
</tr>
<tr>
<td>MOV CL, [EDX+EDI]</td>
<td>8-bits</td>
<td>Copies the byte contents of the data segment memory location addressed by EDX plus EDI into CL.</td>
</tr>
</tbody>
</table>
1.6.8 Register Relative Addressing

Register relative addressing is similar to base – plus – index addressing and displacement addressing. In register relative addressing, the data in a segment of memory are addressed by adding the displacement to the contents of base or index register.

1.6.9 Base-Relative-Plus Index Addressing

The base relative plus index addressing mode is similar to the base plus index addressing mode but adds a displacement besides using a base register and an index register to form the memory address. This type of addressing mode often addresses a two-dimensional array of memory data.

1.6.10 Addressing Data with Base Relative plus index addressing

Base relative plus index addressing is the least used addressing mode. This addressing mode is too complex for frequent use in program.

1.6.11 Addressing Array with Base Relative plus index addressing

When that a file of many records exits in memory and each record contains many elements. This displacement addresses the file, the base register addresses a record and the index register addresses an element of record.
Table 1.9: Register Relative Addressing Instruction

<table>
<thead>
<tr>
<th>Assembly Language</th>
<th>Size</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV AX, [DI+100H]</td>
<td>16-bits</td>
<td>Copies the word contents of data segment memory location addressed by DI plus 100H into AX.</td>
</tr>
<tr>
<td>MOV ARRAY, [SI], BL</td>
<td>8-bits</td>
<td>Copies BL into the data segment memory location addressed by ARRAY plus SI</td>
</tr>
<tr>
<td>MOV LIST, [BP+DI], CL</td>
<td>8-bits</td>
<td>Copies CL into the data segment memory location addressed by sum of LIST, SI and 2.</td>
</tr>
<tr>
<td>MOV DI, SET, IT, [BX]</td>
<td>16-bits</td>
<td>Copies the word contents of the data segment memory location addressed by the sum of SET, IT and BX into DI.</td>
</tr>
<tr>
<td>MOV DI, [EDX + 10H]</td>
<td>16-bits</td>
<td>Copies the word contents of the data segment memory location addressed by the sum of EAX and 10H into DI.</td>
</tr>
</tbody>
</table>

Table 1.10: Base–Relative–Plus Index Addressing Instruction

<table>
<thead>
<tr>
<th>Assembly Language</th>
<th>Size</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV DH, [BX + DI + 20H]</td>
<td>8-bits</td>
<td>Copies the word contents of data segment memory location addressed by the sum of BX, DI and 20H into DH.</td>
</tr>
<tr>
<td>MOV AX, FILE [BX + DI]</td>
<td>16-bits</td>
<td>Copies the word contents of the data segment memory location addressed by the sum of FILE, BX and DI into AX.</td>
</tr>
<tr>
<td>MOV LIST, [BP + DI], CL</td>
<td>8-bits</td>
<td>Copies CL into the data segment memory location addressed by sum of LIST, BP and DI.</td>
</tr>
<tr>
<td>MOV LIST, [BP + SI + 4], DH</td>
<td>8-bits</td>
<td>Copies DH into the stack segment memory location addressed by the sum of LIST, BP, SI and 4.</td>
</tr>
</tbody>
</table>

37
1.6.12 Scaled-Index Addressing

Scaled-index addressing is the last type of the data-addressing mode. This data addressing mode is unique because it uses two 32-bit registers to access the memory. The scaling factor can be 1X, 2X, 4X or 8X. A scaling factor of 1X is implied and need not be included in the assembly language instruction.

A scaling factor of 2X is used to address word sized memory arrays, a scaling factor of 4X is used with double word sized memory arrays and a scaling factor of 8X is used with quad word sized memory arrays.

1.7 BASIC SYSTEM COMPONENTS

The microprocessor based system uses several basic components such as clock generator. It has READY CLOCK and RESET blocks and three generate the respective signals.

1.7.1 The READY Block

The two pairs of Ready (RDY1 / RDY2) and address enables (AEN1 and AEN2) input signals control the Ready (READY) output. If one at the two signals in both input pairs is mode inactive, the READY output goes low. The READY output is used to introduce wait states in the 8086 bus cycles and connected to the READY input of 8086.
Table 1.11: Scaled-Index Addressing Instructions

<table>
<thead>
<tr>
<th>Assembly Language</th>
<th>Size</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV EAX, [EBX + 4 * ECX]</td>
<td>32-bits</td>
<td>Copies the double word contents of the data segment memory location addressed my the sum of 4 times ECX plus EBX into EAX.</td>
</tr>
<tr>
<td>MOV [EAX + 2 * EDI + 100 H], CX</td>
<td>16-bits</td>
<td>Copies CX into the data segment memory location addressed by the sum of EAX, 100 H and 2 times EDI.</td>
</tr>
<tr>
<td>MOV AL, [EBP + 2 * EDI - 2]</td>
<td>8-bits</td>
<td>Copies the byte contents of the stack segment memory location addressed by the sum of EBP, -2 and 2 times EDI into AL.</td>
</tr>
</tbody>
</table>
1.7.2 The Clock Block

\textit{X1 and X2}

The crystal is connected to the two inputs. The crystal oscillator generates the square wave of frequency of the crystal.

\textit{OSC}

The output of crystal oscillator is buffered by an inverting buffer and presented at the oscillator output.

\textit{CLK, and DCLK}

The clock and peripheral clock outputs are used as clock signals for the systems components and peripheral. The CLK frequency is 1/3 of crystal frequency and its duty cycles 33%. The DCLK frequency is 1/2 of the CLK signal.

\textit{EFI}

The external frequency input allows external signal to be used for the generation of CLK and DCLK signals.

\textit{F/C}

The frequency of crystal input selects EFI input or crystal oscillator output to generate the CLK and DCLK outputs.

\textit{CSYNC}

The clock synchronization is used if EFI provides the CLK / DCLK signals.

The 8086 microprocessor multiplexes address, data and states signals on AD15 – ADO and A19/S6 – A16/S3 lines. It outputs address in 'TI' state of bus cycle and changes to data and status during the remaining period of the bus cycle. Address on address bus should be stable throughout a memory or I/O devices for data transfer. Hence, address and data must be demultiplexed.
1.7.3 Buffers

Buffers provide amplified current or power output to drive address and data buses in the microprocessor based systems.

1.7.3.1 Unidirectional octal buffer

A unidirectional octal buffer allows transmission of 8-bit data in one direction 74L5244 is a unidirectional octal buffer. It has two groups of four tri-state buffers. The two groups are separately controlled by two active-low enable (10E and 20E) inputs. When disabled are activated the device outputs logic levels presents at respective inputs. When disabled, the output goes to high – impedance state. The octal buffers are also used as input parts for interfacing peripherals to microprocessors. 74L5240 is another octal buffer, but it has tri – state inverted outputs.

1.7.3.2 Bi-directional octal buffer

A bidirectional octal buffer allows transmission of 8-bit data in either direction 74L5245 is bidirectional octal buffer. It has two groups of eight tri-state buffers. The two groups are controlled through direction control 'DIR' input Logic '1' DIR input enables one group of tri-state buffers and allows data flow in one direction. When one group of buffers is enabled, the other device also has active low chip enable (CE) input. The device is also referred to as transceiver. Intel 8286 is another octal bidirectional buffer providing the same functionally.

1.7.4 Latches

1.7.4.1 D – Latch

If CLK input of D-latch is high, the logic level present at D input is stored at its output. When the CLK is low the logic level present level present at D input will have no effect on its output.

1.7.4.2 Octal Latch

An octal latch has eight D-latch and eight tri-state buffers. The 74L5373 octal latch has two control ‘Enable’ and ‘output
control (OC). The 'Enable' control is active high and connected to CLK inputs of all 8 latches. Logic 1 in CLK input will store the logic levels present at D inputs in respective latches. The 'output control' signal eight tri-state buffers. Logic 0 will enable the buffers to output data from respective latches. The latch Intel 8282 provides the same functionality of 74L5373 but it is not pin-to-pin compatible. Octal latches are used as input/output part in microprocessor based systems.

1.8 BUS CONTROLLER

The minimum mode signals \( \overline{INTA}, \overline{ALE}, \overline{DEN}, DT/\overline{R}, M/IO, \overline{WR}, HLDA \) and HOLD (on pins 24 to 31) that are essential for interfacing memory and IO devices, are not available in the system if the 8086 is operated in maximum mode. A 8288 bus controller is used to generate the relevant signals for interfacing memory and I/O devices in the maximum mode. The bus controller has a command signal generator and a control signal generator.

The 8288 Input and Output Signals

\( S0, S1, \) and \( S2 \): The inputs (8086 status outputs) are decoded to generate command signals.

\( \overline{AEN} \): A low Address Enable Signal activates the memory control signals.

\( \overline{CEN} \): The Control Enable Signal enables the 8288 command outputs.

\( IOB \): High on the I/O Bus input operates the 8288 in the I/O bus mode in systems where there are separate system bus and I/O bus.

\( CLK \): The Clock Input.

\( \overline{DEN} \): The Data bus Enable Signal Controls the data bus buffers in the system. This signal is active-high in contrast to the \( \overline{DEN} \) Signal in the minimum mode.
ALE: The Address latch Enable Signal is used to demultiplex address and data line signals.

DT/\bar{R}: The Data Transmit / Receive Signal Controls bidirectional data bus buffer.

MRDC, MWTC, IORC, and IOWC: The 8288 generates the normal memory Road memory write I/O Road I/O write Control, Signals.

AMWC and ALOWC: These are advanced memory and Advanced I/O write control Signals.

INTA: The Interrupt Acknowledge output.

MCE/PDEN: The master cascade Enable, Enable Peripheral data. Enable output serves dual function. If IOB input is low it is selects cascading of interrupt controllers, and if high enables the I/O bus transceivers.

1.9 ADDRESS DECODERS

The 8086 processors can address up to 1MB memory locations and 64K I/O ports. An 8086 based system does not use all memory space by the memory devices. An I/O map for the system specifies the utilization of I/O space by the I/O devices. Address decodes enable splitting of memory space and I/O space into comfortable blocks and use separate device for each block. A decoder in normally used up to 1MB memory enable for many devices in the system (23).

1.9.1 Line-to-line Decoder

A simple 'n line-to-2^n-line decoder has n inputs and 2^n outputs. It decodes the n binary code applied on its n inputs and activates one of 2^n output. The 74LS139 is a dual 2 line-to-4 line decoder each decoder in 74LS139 has one active low chip-enable input, two inputs (A and B) and for active-low output (Y0–Y3).
1.9.2 PROM and PAL decoders

Programmable Logic Devices (PLD) is used as address decoders in most of the memory and I/O interfacing circuits, Programmable Read Only Memory (PROM), Programmable Array Logic (PAL) and Programmable Logic Array (PLA). All the devices have programmable AND & OR matrices. Though the AND & OR matrices have more than one input. The Fuse connects each horizontal and vertical line junctions. Devices are programmable to blow off the fuses at specified junctions for implementing the desired logic function.

1.10 INTERFACING MEMORY

Memory devices store binary data. The 8086 microprocessor has 20-bit address and 16-bit data buses. The 8086 based system can have the maximum of $2^{20}$ (equal to 1,048,576 or simply 1MB) memory locations. Each location stores an 8-bit data byte. The microprocessor can transfer a byte (8-bit) or a word 16-bit to or form memory in one bus cycle.

1.10.1 Memory Devices

Memory devices are broadly classified into two categories, read only memories (ROM) and random access memories (RAM). Two types of RAM are common. They are static RAM (SRAM) and dynamic RAM (DRAM).

1.10.2 The ROM

The feature of ROM is that is non – volatile. It means that binary word in such memories is not erased when power to the device is switched off. A ROM has two groups of lines address lines and date lines, varies depending on the capacity and width of data word.
The contents of ROM device are external during the manufacturing process and cannot be attend at any stage. Programmable read only memory (PROM) devices are different form ROM device and enable to user to enter the contents using special equipment called PROM. Programmer but or programmed the contents cannot be attend further. Contents of erasable programmable read only memory (EPROM) devices can be attend with EPROM programmers as many times as the user like. Electrically erasable PROM (EEPROM) devices are also called, as flash memory device are common computers today.

The other type of memory device is RAM and also known as read-write memory, which is volatile. The data in RAM remain stored till power to the device is on. The data are lost when power is off. Two types of RAM are most common; they are Static RAM (SRAM) and Dynamic RAM (DRAM).

1.10.3 SRAM

A SRAM user a flip-flop to store a bit of binary information advantage of SRAM is its speed. It is very fast. Disadvantages of SRAM are capacity and cost. The maximum of SRAM on a single chip is just 128K x 8. A 180 the SRAM is expensive.

1.10.4 DRAM

A DRAM uses a tiny capacitor to store a bit of binary information. Advantages of DRAM are that it stores a bit in less space then SRAM and it is inexpensive. Hence DRAM is easily available in capacities such as 16Mx1. The disadvantages of DRAM are that it is slow and its memory locations need to be refreshed at regular intervals. In a DRAM the charge on the capacitor may leak off in a period of time. Repeatedly reading and rewriting the contents at regular intervals using a DRAM refresh controller therefore refresh logic level stored on capacitor.
Fig. 1.8: Schematic diagram of ROM and RAM decoder

Fig. 1.9: Schematic diagram of DRAM decoder
1.11 MEMORY BANKS

Memory for 8086 systems is set upon two banks of 512 KB (= 5, 24, 288 bytes). The two banks are connected to A19 – A1 address lines. Memory bank is enable when A0 is low and takes even address (Since A0 = 0) for its locations. It is referred to as even bank or low bank. The low bank is connected to low data lines, D1 – D0. The other memory bank is enabling when BHE is low and takes add address (Since A0 = 1) for its locations. Its locations it is referred to as old high bank. The high bank is connected to the high data bus lines D15-D8. Here, the Microprocessor low on BHE line in T1 State if D15-D8 lines of data bus is used for data transfer. An external latch in the system captures. The signal and retains for use in the remaining T states). Signal on BHE and A0 identify the bank in which the addressed location resides and address on A19-A1 identifies the location in the bank. The microprocessor users the data bus in different ways for byte or word transfer to or from add or even addressed locations.

1.11.1 Byte Transfer to/from even Addressed Location

When the microprocessor transfers a byte to/from an even addressed location, the address bus carries the 20-bit physical address of low order. D1-D0 lines of the data bus carry the data byte. The D15-D8 lines are in high impedance state. The signal on BHE is high and A0 is low. It enable even addressed memory bank and disable the old addressed memory bank.
Fig. 1.10: Schematic diagram of memory bank
1.11.2 WordTransfer to/form even Addressed Location

For a word transfer to/form even addressed location, the address bus carries the 20-bit physical address of low order byte, the D15-D8 lines of the data bus carry the high order byte and D7-D0 carry the low order byte. The signals on BHE and A0 are low which enable both even and odd addressed memory banks.

1.11.3 Byte Transfer to/form odd Addressed Location

For a byte transfer to/form an odd addressed location, the address bus carries the address the D15-D8 lines of the data bus carry the data byte and the D7-D0 lines are high-impedance state. The signal on BHE is low and signal on A0 is high. It enables odd addressed memory bank and disable even addressed memory bank.

1.12 ADDRESS DECODING LOGIC

The address decoding logic in an 8086 system uses either simple decoders such as 74LS138 and 74LS139, or programmable devices such as PROM and PAL.

1.12.1 PROM Decoder

Figure 3.6 shows the PROM decoder. The PROM has 256 locations and holds 4-bit nibble in each location. The device when accessed with the address of location outputs the 4-bit data from the addressed location. System address lines A19-A12 select one of 256 locations in the PROM. Each location is accessed for a range of 4K addresses. The outputs are connected to chip enable (CE) inputs of memory devices. PROM is programmed for generating devices select signals. Programming the PROM for device.

1.12.2 PAL Decoder

Pal (Programmable array logic) is also widely used for address decoding. PAL decoder is 16L8; PAL16L8 has 10 dedicated inputs
and 2 dedicated outputs. It has 6 bidirectional (tri-state) lines that can be programmed either as input or output. 16L8 indicates that the device has a maximum of 16 inputs, maximum of 8 outputs, and the outputs are active-low. The PAL decoder is programmed to use 10 lines for inputs and 8 lines for outputs. It is programmed to generate device select outputs for the following input conditions.

### 1.12.3 SRAM Interfacing

It interfaces two 62128 RAM devices (16K×8) to the 16-bit data bus of 8086. Each RAM has 14 address 8 data a chip-enable (CE), a write enable (WE) and an output enable (OE) inputs. Both RAM-0 and RAM-1 have their address bus. The two OR gates provide chip enable signals for RAMs. Low on A0 enables RAM-0 and low on BHE enables RAM-1, RAM-0 has even addresses for its locations and has its data lines connected to lower 8-bit data lines D15-D8. Both RAM-0 and RAM-1 hold the block of continues 32K memory locations serving the address from 00000 H to 07FFFFH following is an example of interfacing.

### 1.12.4 DRAM Interfacing

DRAM interfacing requires a DRAM controller top read, write and refresh. The DRAM controller performs three basic tasks (i) multiplexes the row and column address (ii) translates the system MEMR and MEMW control signals to RAS, CAS and WE control signals compatible for the DRAM and (iii) Keeps the DRAM refreshed.

Intel 82C08 is a CMOS DRAM controller that can control two banks of 256K×16 DRAM chips, it contains a refresh counter refresh timer and address multiplexes to select to row column or refresh address it also includes combination and control logic to coordinate refresh cycles with microprocessor accesses and provides the required control signals top directly drive the DRAM.
Fig. 1.11: Schematic diagram of PROM decoder

Fig. 1.12: Schematic diagram of SRAM interfacing
Fig. 1.13: Schematic diagram of SRAM interfacing
1.13 INTERFACING I/O DEVICES

Input and output parts are interfaced to microprocessor to occupy memory or I/O space of two the system. I/O devices interfaced to occupy memory addresses and are referred as memory. Mapped I/O and I/O devices interfaced to occupy I/O addresses are referred to as I/O mapped I/O

1.13.1 Memory Mapped I/O Scheme

In the memory mapped I/O, the I/O devices are treated as memory locations and referenced by addresses of memory locations. The decoding logic for memory mapped I/O uses memory read (MEMR) and memory write (MEMW) signals. Any instruction that transfers data between processor and memory is used to transfer data between processor and memory is used to transfer data between the processor and the I/O devices.

1.13.2 I/O Mapped I/O Scheme

In the I/O mapped I/O, which is also called isolate I/O. The I/O devices are referenced by 8 – bit or 16 – bit addresses and the IN or OUT instructions are used for transferring data between the processor and I/O devices. The decoding logic for I/O mapped I/O uses I/O read (IOR) and I/O write (IOW) signals.

1.13.3 Interfacing to 8-bit Bus

In a simple system a tri-state such as 74LS244 can be used as an input port and a latch such as 74 LS374 can be used as an output part.
1.13.4 Interfacing Input Port

Figure 1.14 shows the interfacing of an 8-bit in put AND gate 74LS20 and 3-line-to-8 line decoder 745138. It decodes 8-bit part addresses. 8-bit decoding the less number of times. A0 and IOR signals at the two enable inputs of 742S138 (G2A and G2B) enables the decoder during I/O road operation from even part addresses. Connections to G1 enable – input enables the decoder to generate device select signals during I/O read from even I/O address F0H to FEH. The input part shown in the figure is enabled for the address FOH.

When an I/O instruction in AL, FOH is executed. The buffer is enabled it effectively inputs the status of switches to the processor. When the buffer is disabled at the end of I/O read operation. It is unidirectional disconnected from data bus.

1.13.5 Decoding 16-bit Port Address

The decoder generates device select signals for the when I/O is enabled for the address FFFOH. The input port shown in address type I/O instruction IIN, AL, DX (with FFFOH in DX) inputs the status information.

1.13.6 Interfacing Output Port

The circuit in figure 1.15 shows the interfacing of an 8-bit output port. A latch 74LS314 is used as output port and is interfaced to D7-D0 lines of the system data bus. The port outputs a data byte to eight LED displays.
Fig. 1.14: Schematic diagram interfacing input port

Fig. 1.15: Schematic diagram interfacing output port
1.13.7 Interfacing to 16-bit Bus

Single 16-bit I/O devices are rarely used to microprocessor-based systems. However, I/O for 16-bit 8086 system may also be set up on two 8-bit I/O banks like memory. The PAL decoder generates individual chip select signals for the ports for the input conditions given below and assigns I/O addresses 40H and 41H to the ports. The decoder in this scheme decodes the A0 and BHE signals also. It allows 16-bit data transfer on 8-bit data transfer to ports.

The decoding logic generates a common chip select signal. The input condition the generators the chip select is given below, it assigns I/O port address 40H to the and allows only 16-bit data transfer.

1.13.8 Interfacing Data Converters

Digital to analog converters and analog to digital converters are the two basic devices used in measurement and control instrumentation applications. The converters are often interfaced to microprocessor.

1.14 Digital to Analog Converters

Weighted resistor network uses a simple register network for digital to analog conversion. Resistors in the network are binary weighted i.e. $2^1R$, $2^2R$, $2^3R$, $2^4R$, ......, $2^nR$. Each of the digital inputs controls a (transistor) switch in the network. When a switch is closed, current flows the reference source through the binaryweighted resistor to the summing point (virtual ground). The OP amp performs the sum of currents and outputs a proportional voltage. Functionally the binary weighted resistors pass binary weighted currents that are summed and converted to voltage for a 4-bit DAC, the output I/O is given by following equation:

$$V_o = V_{ref} (S_3/2^1 + S_2/2^2 + S_1/2^3 + S_0/2^4)$$
Where $S_3$, $S_2$, $S_1$ and $S_0$ represent the status of the switches and take value 1 or 0 if the respective switch is closed or opened. The 4-bits of the digital input control the switches different discrete voltage at the output. In general, an n-bit digital quantity produce $2^n$ different discrete among voltage.

There is practical difficulty in implementation the weighted resistor network. The construction of an n-bit DAC with this type of simple resistor network need $n + 1$ resistor with values $2^0R$, $2^1R$, ......, $2^nR$. The value of LSB resistor in $2^n$ times the feedback resistor $R$. The nominal value of feedback resistor is SKR. For 8-bit and 12-bit DACs the LSB resistor will be 1.28 Mr ($2^8 \times 5$Kr) and 20.48 Mr ($2^{12} \times 5$Kr) respectively, such high values are not easily achievable in ICs.

The R-2R ladder resistor network solves the above problem in simple way. R to 2R, in ladder arrangement. It is functionally equivalent to the previous technique. The current entering a branch in the network splits into two equal halves at a node and further divides equally again at each node as it proceeds through the ladder. Each of the digital input controls a switch to steer current through the resistor either to the summing point or to the ground. The OP amp performs the sum of currents reaching the summing point (1s) and out proportional voltage. Analog output voltage for digital input quantity is given as following equation:

$$Vo = (RX1) \text{ if } (S_3/2^1 + S_3/2^1 + S_2/2^2 + S_1/2^3 + S_0/2^4) =$$

$$\text{Vet (} S_3/2^1 + S_3/2^1 + S_2/2^2 + S_1/2^3 + S_0/2^4),$$

Where, $S_3$, $S_2$, $S_1$ and $S_0$ represent the status of the switches and take value of respective digital input (i.e. 1 or 0).

Current output DAC. In the above techniques, the Op amp performs the sum of currents to voltage. The performance of such devices depends on the speed of output OP amp. Though devices
producing voltage outputs are may converted to use, better performance could be obtained with current output devices. The current output could then be easily converted to voltage.

**1.14.1 AD558**

AD558 is a popular 8-bit DAc from Analog devices, which is available on 16 pins DIP. Figure 3.11 shows the internal blocks and DAc interfaced to 8086 system.

**Internet Blocks**

The DAc has an internal 8-bit latch, R-R2 ladder network, onboard precision references voltage and an internal OP amp. It is microprocessor compatible and has CS, CE terminals. When both CS and CE are low, the binary input on D7-D0 terminals is transferred to DAc section. When either CS or CE goes high, the input is latched in the terminal resistor and hold until both signals go to low. The device has two separate ground terminals for digital and analog signals. The range of output is pin programmable with select and sense pins for 0 to 2.56 V 01 0 to 10.0 V. The device is configured to operate in the range of 0 to 2.56 V.

**1.14.2 Interfacing to 8089**

The PAL decoder connects the DAc to the system at I/O port address FEH. The PAL is programmed to generate device select signal at its V0 output for the input condition.

\[ V0 = A7 \times A6 \times A5 \times A4 \times A3 \times A2 \times A1 \times A0 \]

The decoder output V0 is connected to CS and the IOW is connected to CE input. An I/O write at port address FEH transfers the byte remains stored in the latch till next chip select occurs. The DAc generates an along equivalent for byte.
Fig. 1.16(a): Figure showing internal blocks

Fig. 1.16(b): Figure showing DAC interfaced to 8086 system
1.15 ANALOG TO DIGITAL CONVERTERS

Analog to digital converter produces digital data at its output that represents the magnitude of signal at its input. Several types of ADCs are commercially available. However, for choosing an ideal DAC among hundreds of devices for an application, better understanding of conversion techniques, their advantages and disadvantages is essential. Following are the types of ADC:

- Integration (Signal and dual slope)
- Successive approximation
- Parallel conversion

Analog to digital conversion is more complex than digital to analog conversion. Most ADCs (except parallel conversion types) do not convert instantaneously like DACs. They begin conversion on receipt of start of conversion (SOC) signal and indicate the completion by an end of conversion signal (EOC). The minimum time that is required for the ADC to complete the conversion is referred to as conversion time. It is then switched to known reference voltage with polarity opposite to Vin. The integrator output now amps in the opposite direction at constant rate and reaches zero level in time T2. Periods of integration of Vin (i.e. T1) and Vref (i.e. T2) are related as $T_2 = T_1 \times \frac{Vin}{Vref}$. During the period T2 the counter counts clock pulses from the clock generator. The counter’s content is the digital output. It is simply the frequency of the clock times the period of integration of reference, digital output = frequency $\times T_2$.

Conversion by dual slope technique is slower than any other technique. Integrating ADCs have conversion time between 30 ms to 300 ms. The conversion time is also not constant and depends on the level of input. However, the effect of non-linearity is introduced
during reference integration also. Since the conversion is based on integration this technique has excellent noise rejection capability and widely used in applications in noisy environment.

1.15.1 Successive Approximation ADC

Successive approximation is the popular technique implemented in median to high-speed converters. The technique is similar to determining the unknown weight of an object using standard weights. The successive approximation registers (SAR), a DAC and a comparator. The output of SAR is connected to the comparator. The unknown analog input is connected to the other input of the comparator. The output of compared and approximation is achieved.

Comparator compares unknown analog input \( V_{in} \) with DAC output \( V_{in} \) with DAC output \( V_0 \) and outputs signal to reset D2 if \( V_{in} < V_0 \) (i.e. binary code to 000 if \( V_{in} < V_0 \) 01 100 if \( V_{in} > V_0 \)). The next clock input sets the next bit D1 in SAR. The comparator again compares the input with DAC output and outputs signal to reset D1 if \( V_{in} < V_0 \). The comparison can parson continues till LSB D0 is tested. At the end of LSB test the SAR contains the best binary approximation of the input signal.

An important condition to the successive approximation ADC to function properly is that the input signal should remain constant throughout the conversion process. If the signal is changing during the conversion period, the ADC cannot predict its digital equivalent. Hence, the input voltage is held constant using a sample and hold device before the conversion is initiated.

1.15.2 AD574A

AD574A is a 12 bit successive approximation ADC from Analog devices.
1.15.3 Functional Blocks

The ADC is operated either in 10V or 20V range. Analog input is connected between pins 13 and 9 for 10V and between pins 14 and 9 for 20V range. It is operated either in unipolar or bipolar mode. In unipolar mode, the range of operation is either 0-10V (in 10V range) or 0-20V (in 20V range). In bipolar mode, the range of operation is either 0-10V (in 10V range) or 0-20V (in 20V range). In bipolar mode, the range of operation is −5.0 to +5.0 (in 10V range) or −10 to +10 (in 20V range). The tri-state output buffer circuitry provides facility to interface to 8-bit or 16-bit data bus. It has on broad high precision reference and a clock.

1.16 INTERRUPTS

Microprocessor fetches the instruction codes from memory in sequence and executes the microprocessor may by interrupted in the middle of a program to handle some emergency to handle some emergency tasks. The origin for the interrupts may be internal or external to the microprocessor. When an interrupt occurs in response to the microprocessor completes the execution to current instruction in the program jumps to a subroutine program called interrupt service procedure, and executes the instructions in the interrupt service procedure. After the interrupt service procedure is completed, the execution returns back to the interrupt program and continues.

1.16.1 8086 Interrupt Types

The 8086 interrupt are classified into three types. They are

- Predefined Interrupt
- Software Interrupt
- Hardware Interrupt

62
1.16.1.1 Predefined Interrupt

Interrupts types 0 through 31 have been defined as predefined interrupts and only the interrupt types 0 through 4 are used in 8086. Interrupt types 5 through 31 are reserved for use in advanced versions. The predefined interrupts are divide are by zero, single step, non-maskable, breakpoint interrupts.

Divide by zero interrupts, Type 1

The divide by zero interrupt occurs whenever a division by zero is attempted using a divide instruction. When the interrupt occurs, the microprocessor automatically performs type 0 interrupt response. This interrupt is non-maskable.

Single step interrupt, Type 1

The single step interrupt occurs after the execution of an instruction, if the trap flag (TF) in the program status word (PSW) is set. When the interrupt occurs, the microprocessor automatically performs type 1 interrupt response. The interrupt service procedure for the single step interrupt usually provides various diagnostic capabilities.

Non-Maskable interrupt (NM), Type 2

The non-maskable interrupt occurs when a low to high transition on NMI input pin of the microprocessor on interrupt, the microprocessor automatically performs type 2 interrupt response. This interrupt cannot be disabled or masked by any instruction. It is generally used to save critical data from volatile to non-volatile region in memory when power to the system fails.
**Breakpoint interrupts, Type 3**

The breakpoint interrupt is one-byte software interrupt instruction INT3. The instruction is inserted in a program where the execution should stop. It is useful for program debugging. The interrupt is non-maskable.

**Overflow interrupt, Type 4**

The overflow interrupt occurs when the overflow flag (OF) is set as a result of an arithmetic operation on two signed numbers and the interrupt on overflow instruction (INTO) is executed. The microprocessor automatically performs type 4 interrupt response when the interrupt occurs for the interrupt to occur the INTO instruction should immediately follow the arithmetic instruction. If the OF is not set after the signed arithmetic operation, the NEXT INTO instruction simply performs no operation, the overflow interrupt is non-maskable.

**1.16.2 Software Interrupts**

The microprocessor may be interrupted when it accepted a two byte interrupt INT 'n' instruction in the program. It is referred to as software interrupt. The 'n' in the instruction specifies the interrupt type and takes a value between 0 and 255. In response to the interrupt, the microprocessor gets the interrupt vector for the type 'n' from the table and executes the interrupt service procedure.

Basis input and output services (BIOS) interrupts in a PC are popular applications of software interrupts. BIOS are a collection of service routines that reside in ROM and directly control hardware components of a PC. Routines controlling the video display, keyboard, printer, and mouse of a PC are called by software interrupt instruction INT 10H, INT 16H, INT 17H, and INT 33H respectively.
1.16.3 Hardware Interrupts

The microprocessor has two interrupt inputs. They are non-maskable interrupt (NMI), and interrupt Request (INTR). The processor may be interrupted by external asynchronous signal applied to one of the two interrupt by an external asynchronous signal applied to one of the two interrupt inputs.

**NMI**

Whenever the NMI input is activated (a low to high transition), a type 2 interrupt occurs and the microprocessor automatically performs type 2 response. A common application for the NMI input is to save critical data in the event of power failure sends interrupt signal to the NMI input. Usually the capacitors in the system power supply can provide the voltage for a minimum period of 50 ms even after the power failure. This period is enough for the microprocessor to perform type 2 responses, which moves critical data from RAM to memory with battery backup.

**INTR**

External devices requiring the immediate attention of microprocessor interrupt through the interrupt request (INTR) input. Logic high to INTR pin activates the interrupt. When the interrupt is activated the 8086 the interrupt type from an external device such as a priority interrupt controller, 8259 and performs the interrupt response. This interrupt is useful in interacting slow devices to microprocessors. It is level-sensitive and can be enabled or disabled by setting or resetting the interrupt flag (IF) bit of the PSW. The IF of PSW is set by 'STI' instruction and reset by 'CLI' instruction (Fig. 1.17).
Priority of 8086 Interrupts

The 8086 assign priority among various interrupts. If more than one interrupt occurs as a time. The divide by zero and software instruction INTN and INTO move highest priority among all interrupts, NMI, INTR and single-step interrupts have priorities in the decreasing order. It means that simultaneous divide by zero and NMI/INT/ single-step interrupts cause the microprocessor to save divide by zero interrupt first, followed by the other interrupt. Similarly, simultaneous NMI and INTR interrupt cause the microprocessor to execute type 2 response (NMI) first and INTR interrupt type NEXT, the single-step interrupt has the test priority.

1.17 DIRECT MEMORY ACCESS

Microprocessor performs I/O read and memory writes operations to transfer a data from an I/O device to a memory. Similarly, it performs a memory read and an I/O writes operations to transfer a date from memory to I/O device. Every data transfer between a memory and an I/O device goes through the microprocessor. The direct memory access (DMA) technique provides direct access between memory and I/O device and enables fast and bulk data transfer without the of a microprocessor.

1.17.1 Basic DMA Operation

The 8086 in minimum mode have HOLD (DMA request) input and HLDA (DMA acknowledge) output pins to support DMA operation. DMA controlled data transfer I/O device to memory is
called as DMA write and data transfer from memory to I/O device is called as DMA read. An I/O device, requesting the DMA read or write operation, sends DREQ to the DMA controller, and in turn, the DMA controller sends hold request HRQ to the microprocessor samples the HOLD input in middle of every clock cycle. When the microprocessor detects the HOLD signal it stops executing further instructions floats its buses and sends out hold acknowledge (HLDA) signal to the DMA controller. The DMA controller then sends out a control signal to virtually disconnect the microprocessor from the address, data and control buses, and connect the DMA controller. The DMA controller thus gets the control of the buses and sends out the first byte of the I/O device is to be transferred. Then it sends DMA acknowledge (DACK) signal to the I/O device. Finally it sends out memory read (MEMR) and I/O write (IOW) control signals for DMA read or I/O read (IOR) and memory write (MEMW) control signals for DMA write on the control bus. It enables the byte of data directly to go between the I/O device and memory without passing through the microprocessor the DMA controller.

After the data transfer is complete, the DMA controller disables HOLD signal and releases the control of address, data and control buses back to the microprocessor by virtually disconnecting the buses from the DMA controller and connecting to the microprocessor.
Fig. 1.17: Schematic diagram of hardware interrupt request

Fig. 1.18: Schematic diagram of DMA operation