CHAPTER 1

INTRODUCTION TO MICROPROCESSORS

1.1 TERMS USED IN MICROPROCESSOR LITERATURE

Bit : A digit of the binary number or code is called a bit.

Nibble : The 4-bit (4-digit) binary number or code is called a nibble.

Byte : The 8-bit (8-digit) binary number or code is called a byte.

Word : The 16-bit (16-digit) binary number or code is called a word.

Double Word : The 32-bit (32-digit) binary number or code is called a double word.

Multiple Word : The 64, 128, ... bit/digit binary numbers or codes are called multiple words.

Data : The quantity (binary number/code) operated by an instruction of a program is called data. The size of data is specified as bit, byte, word,etc.

Address : Address is an identification number (in binary) for memory locations. The 8086 processor uses a 20-bit address for memory.

Memory Word Size (or Addressability) : The memory word size or addressability is the size of binary information that can be stored in a memory location. The memory word size for an 8086 processor-based system is 8-bit.

[Address and program codes in a microprocessor system are given in binary (i.e., as a combination of "0" and "1"). With n-bit binary we can generate 2^n different binary codes or addresses.]

Microprocessor : The microprocessor is a program-controlled semiconductor device (IC), which fetches instruction and data (from memory), decodes and executes instructions. It is used as CPU (**C**entral **P**rocessing **U**nit) in computers.

The basic functional blocks of a microprocessor are ALU (**A**rithmetic **L**ogic **U**nit), an array of registers and a control unit. The microprocessor is identified with the size of data the ALU of the processor can work with at a time. The 8085 processor has a 8-bit ALU; hence, it is called a 8-bit processor. The 8086 processor has a 16-bit ALU; hence, it is called a 16-bit processor.

Bus : A bus is a group of conducting lines that carries data, address and control signals. Buses can be classified into Data bus, Address bus and Control bus.

The group of conducting lines that carries data is called a data bus.

The group of conducting lines that carries address is called an address bus.

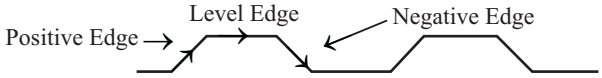
The group of conducting lines that carries control signals is called a control bus.

CPU Bus : The group of conducting lines that are directly connected to the microprocessor is called a CPU bus. In a CPU bus, the signals are multiplexed, i.e., more than one signal is passed through the same line but at different timings.

System Bus : The group of conducting lines that carries data, address and control signals in a microcomputer system is called System bus. Multiplexing is not allowed in a system bus.

[In microprocessor-based systems, each bit of information (data/address/control signal) is sent through a separate conducting line. Due to practical limitations, the manufacturers of microprocessors may provide multiplexed pins, i.e., one pin is used for more than one purpose. This leads to a multiplexed CPU bus. For example, in an 8086 processor, the address and data are sent through the same pins but at different timings. But when the system is formed, the multiplexed bus lines should be demultiplexed by using latches, ports, transceivers, etc. The demultiplexed bus lines are called system bus. In a system bus, separate conducting lines will be provided for each bit of data, address and control signals.]

Clock : A clock is a square wave used to synchronize various devices in the microprocessor and in the system. Every microprocessor system requires a clock for its functioning. The time taken for the microprocessor and the system to execute an instruction or program are measured only in terms of the time period of its clock.



A clock has three edges: rising edge (positive edge), level edge and falling edge (negative edge). The device is made sensitive to any one of the edges for better functioning (it means that the device will recognize the clock only when the edge is asserted or arrived).

Tristate Logic : Almost all the devices used in a microprocessor-based system use tristate logic. In devices with tristate logic, three logic levels will be available: **High** state, **Low** state and **High impedance** state.

The **high** and **low** level states are normal logic levels for data, address or control signals. The **high impedance** state is an electrical open-circuit condition. The **high impedance** state is provided to keep the device electrically isolated from the system. The tristate devices will normally remain in the **high impedance** state and their pins are physically connected in the system bus but electrically isolated. In the **high impedance** state, they cannot receive or send any signal or information. These devices are provided with chip enable/chip select pins. When the signal at this pin is asserted to the right level, they come out from the **high impedance** state to normal levels.

1.2 EVOLUTION OF MICROPROCESSORS

History tells us that it was the ancient Babylonians who first began using the abacus (a primitive calculator made of beads) in about 500 BC. This simple calculating machine eventually sparked the human mind into the development of calculating machines that use gears and wheels (Blaise Pascal in 1642). The giant computing machines of the 1940s and 1950s were constructed with relays and vacuum tubes. Next, the transistor and solid-state electronics were used to build the mighty computers of the 1960s. Finally, the advent of the **I**ntegrated **C**ircuit (IC) led to the development of the microprocessor and microprocessor-based computer systems.

In 1971, INTEL Corporation released the world's first microprocessor the INTEL 4004, a 4-bit microprocessor. It addresses 4096 memory locations of 4-bit word size. The instruction set consists of 45 different instructions. It is a monolithic IC employing large-scale integration in PMOS technology. The INTEL 4004 was soon followed by a variety of microprocessors, with most of the major semiconductor manufacturers producing one or more types.

First-Generation Microprocessors

The microprocessors introduced between 1971 and 1973 were the first-generation processors. They were designed using PMOS technology. This technology provided low cost, slow speed and low output currents and was not compatible with TTL (**T**ransistor **T**ransistor **L**ogic) levels.

The first-generation processors required a lot of additional support ICs to form a system, sometimes as high as 30 ICs. The 4-bit processors are provided with only 16 pins, but 8-bit and 16-bit processors are provided with 40 pins. Due to limitations of pins, the signals are multiplexed. A list of first-generation microprocessors are as follows:

1.	**INTEL 4004**	4-bit processors
2.	**INTEL 4040**	
3.	**FAIR CHILD PPS - 25**	
4.	**NATIONAL IMP - 4**	
5.	**ROCKWELL PPP - 4**	
6.	**MICRO SYSTEMS INTL. MC - 1**	
7.	**INTEL 8008**	8-bit processors
8.	**NATIONAL IMP - 8**	
9.	**ROCKWELL PPS - 8**	
10.	**AMI 7200**	
11.	**MOSTEK 5065**	
12.	**NATIONAL IMP/16**	16-bit processors
13.	**NATIONAL PACE**	

Second-Generation Microprocessors

The second-generation microprocessors appeared in 1973 and were manufactured using the NMOS technology. The NMOS technology offers faster speed and higher density than PMOS and it is TTL compatible. Some of the second-generation processors are as follows:

1.	**INTEL 8080**	8-bit processors
2.	**INTEL 8085**	
3.	**FAIRCHILD F - 8**	
4.	**MOTOROLA M6800**	
5.	**MOTOROLA M6809**	
6.	**NATIONAL CMP -8**	
7.	**RCA COSMAC**	
8.	**MOS TECH. 6500**	
9.	**SIGNETICS 2650**	
10.	**ZILOG Z80**	
11.	**INTERSIL 6100**	12-bit processors
12.	**TOSHIBA TLCS - 12**	
13.	**TI TMS 9900**	16-bit processors
14.	**DEC - W.D. MCP - 1600**	
15.	**GENERAL INSTRUMENT CP 1600**	
16.	**DATA GENERAL μN601**	

Characteristics of Second-Generation Microprocessors

1. **Larger chip size (170 × 200 mil). [1mil = 10^{-3} inch]**
2. **40 pins.**
3. **More numbers of on-chip decoded timing signals.**
4. **The ability to address large memory spaces.**
5. **The ability to address more IO ports.**
6. **Faster operation.**
7. **More powerful instruction set.**
8. **A greater number of levels of subroutine nesting.**
9. **Better interrupt-handling capabilities.**

Third-Generation Microprocessors

After 1978, the third-generation microprocessors were introduced. These are 16-bit processors and designed using HMOS (**H**igh density **MOS**) technology. Some of the third generation microprocessor are given below:

1. **INTEL 8086**	4. **INTEL 80286**	7. **ZILOG Z8000**
2. **INTEL 8088**	5. **MOTOROLA 68000**	8. **NATIONAL NS 16016**
3. **INTEL 80186**	6. **MOTOROLA 68010**	9. **TEXAS INSTRUMENTS TMS 99000**

The HMOS technology offers better **S**peed **P**ower **P**roduct (SPP) and higher packing density than NMOS.

Speed Power Product (SPP) = Speed × Power

Unit of SPP = Nanoseconds × Milliwatts

= Picojoules

1. **Speed Power Product of HMOS is four times better than NMOS.**
   SPP of NMOS = 4 picojoules (pJ)
   SPP of HMOS = 1 picojoules (pJ)
2. **Circuit densities provided by HMOS are approximately twice those of NMOS.**
   Packing density of NMOS = 1852.5 gates/mm^2
   Packing density of HMOS = 4128 gates/mm^2 (1 mm = 10^{-6} metre)

Characteristics of Third-Generation Microprocessors

- **Provided with 40/48/64 pins.**
- **High speed and very strong processing capability.**
- **Easier to program.**
- **Allow for dynamically relocatable programs.**
- **Size of internal registers are 8/16/32 bits.**
- **The processor has multiply/divide arithmetic hardware.**
- **Physical memory space is from 1 to 16 megabytes.**
- **The processor has segmented addresses and virtual memory features.**
- **More powerful interrupt-handling capabilities.**
- **Flexible IO port addressing.**
- **Different modes of operations (e.g., user and supervisor modes of M68000).**

Fourth-Generation Microprocessors

The fourth-generation microprocessors were introduced in the year 1980. These generation processors are 32-bit processors and are fabricated using the low-power version of the HMOS technology called HCMOS. These 32-bit microprocessors have increased sophistications that compete strongly with mainframes. Some of the fourth-generation microprocessors are given below:

1. **INTEL 80386**
2. **INTEL 80486**
3. **NATIONAL NS16032**
4. **MOTOROLA M68020**
5. **BELLMAC - 32**
6. **MOTOROLA M68030**
7. **MOTOROLA MC88100**

Characteristics of Fourth-Generation Microprocessors

1. **Physical memory space of 2^{24} bytes = 16 MB (megabytes).**
2. **Virtual memory space of 2^{40} bytes = 1TB (terabytes).**
3. **Floating-point hardware is incorporated.**
4. **Supports increased number of addressing modes.**

Fifth-Generation Microprocessors

In microprocessor technology, INTEL has taken a leading edge and is developing more and more new processors. The latest processor by INTEL is the **pentium** which is considered a fifth-generation processor. The pentium is a 32-bit processor with 64-bit data bus and is available in a wide range of clock speeds from 60 MHz to 3.2 GHz. With improvement in semiconductor technology, the processing speed of microprocessors has increased tremendously. The 8085 released in the year 1976 executes 0.5 **M**illion **I**nstructions **P**er **S**econd (0.5 MIPS). The 80486 executes 54 Million Instructions Per Second. The pentium is optimized to execute two instructions in one clock period. Therefore, a pentium processor working at 1 GHz clock can execute 2000 **M**illion **I**nstructions **P**er **S**econd (2000 MIPS). The various processors released by INTEL are listed in Appendix I.

1.3 FUNCTIONAL BUILDING BLOCKS OF A MICROPROCESSOR

A microprocessor is a programmable IC which is capable of performing arithmetic and logical operations. The basic functional block diagram of a microprocessor is shown in Fig. 1.1.

The basic functional blocks of a microprocessor are ALU, Flag register, Register array, **P**rogram **C**ounter (PC)/**I**nstruction **P**ointer (IP), Instruction decoding unit, and the Timing and Control unit.

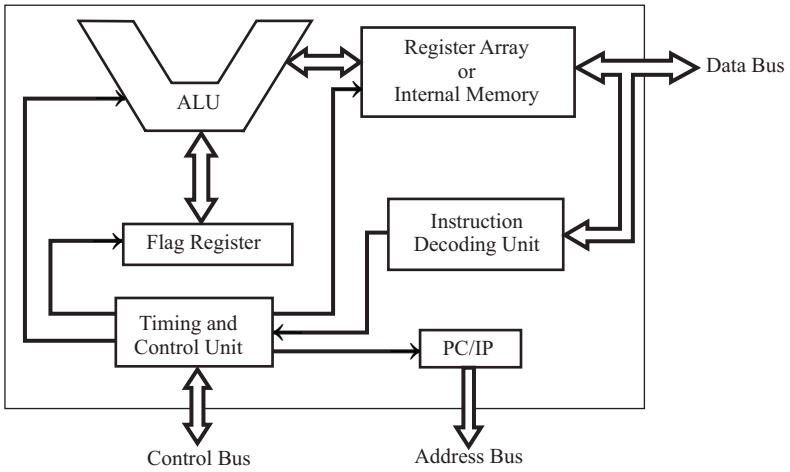


Fig. 1.1: Block diagram showing functional blocks of a microprocessor.

ALU is the computational unit of the microprocessor which performs arithmetic and logical operations on binary data. The various conditions of the result are stored as status bits called flags in the flag register. For example, consider sign flag. One of the bit positions of the flag register is called sign flag and it is used to store the status of sign of the result of the ALU operation (output data of ALU). If the result is negative then "1" is stored in the sign flag and if the result is positive then "0" is stored in the sign flag.

The register array is the internal storage device and so it is also called internal memory. The input data for ALU, the output data of ALU (result of computations) and any other binary information needed for processing are stored in the register array.

For any microprocessor, there will be a set of instructions given by its manufacturer. For doing any useful work with the microprocessor, we have to first write a program using these instructions, and store them in a memory device external to the microprocessor.

The instruction pointer generates the address of the instructions to be fetched from the memory and sends it through the address bus to the memory. The memory will send the instruction codes and data through the data bus. The instruction codes are decoded by the decoding unit and it sends information to the timing and control unit. The data is stored in the register array for processing by the ALU.

The control unit will generate the necessary control signals for internal and external operations of the microprocessor.

1.4 MICROPROCESSOR-BASED SYSTEM (ORGANIZATION OF A MICROCOMPUTER)

A microprocessor is a semiconductor device (or integrated circuit) manufactured by using the VLSI (**V**ery **L**arge **S**cale **I**ntegration) technique. It includes the ALU, the register arrays and the control circuit on a single chip. To perform a function or useful task we have to form a system by using the microprocessor as a CPU (**C**entral **P**rocessing **U**nit) and interfacing the memory, input and output devices to it. A system designed by using a microprocessor as its CPU is called a microcomputer or a single board microcomputer. A microprocessor-based system consists of a microprocessor as the CPU, semiconductor memories like EPROM and RAM, an input device, an output device and interfacing devices. The memories, input devices, output devices and interfacing devices are called peripherals.

The commonly used EPROM and static RAM in microcomputers are given below:

EPROM	Static RAM
INTEL 2708 (1 kB)	MOTOROLA 6208 (1kB)
INTEL 2716 (2 kB)	MOTOROLA 6216 (2kB)
INTEL 2732 (4kB)	MOTOROLA 6232 (4kB)
INTEL 2764 (8kB)	MOTOROLA 6264 (8kB)

Note: *kB refers to kilo bytes.*

Popular input devices are keyboard, floppy disk, etc., and output devices are printer, LED and LCD displays, CRT monitor, etc.

The block diagram of a microprocessor-based system (or organization of microcomputer) is shown in Fig. 1.2. In this system the microprocessor is the master and all other peripherals are slaves. The master controls all the peripherals and initiates all operations.

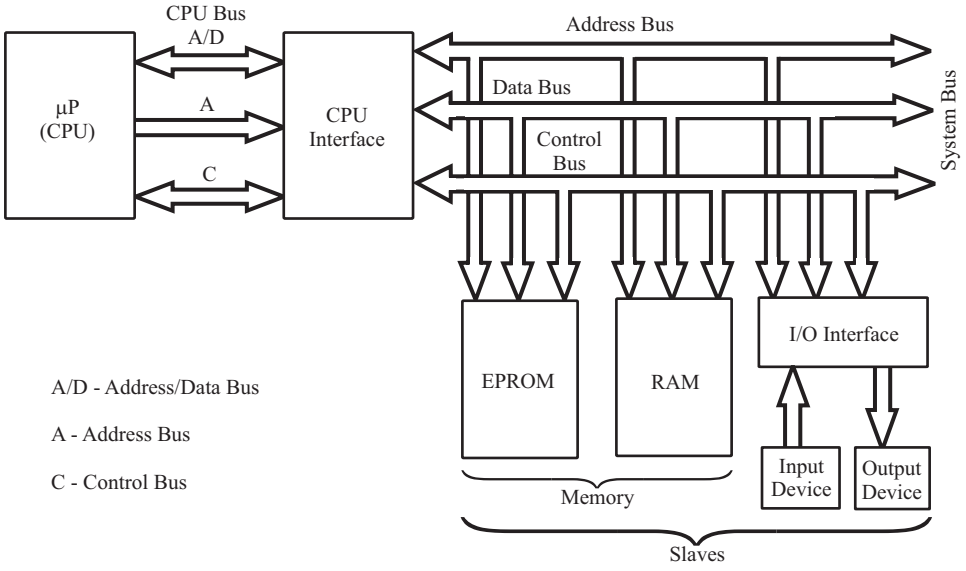


Fig. 1.2: Microprocessor-based system (organization of microcomputer).

Buses are groups of lines that carry data, addresses or control signals. The CPU bus has multiplexed lines, i.e., the same line is used to carry different signals. The CPU interface is provided to demultiplex the multiplexed lines to generate chip select signals and additional control signals. The system bus has separate lines for each signal.

All the slaves in the system are connected to the same system bus. But communication takes place between the master and one of the slaves at any one time. All the slaves have tristate logic and hence normally remain in **high impedance** state. The processor selects a slave by sending an address. When a slave is selected, it comes to the normal logic and communicates with the processor.

The EPROM memory is used to store permanent programs and data. The RAM memory is used to store temporary programs and data. The input device is used to enter the program, data and to operate the system. The output device is used for examining the results. Since the speed of IO devices does not match with the speed of the microprocessor, an interface device is provided between the system bus and the IO devices. Generally IO devices are slow devices.

The work done by the processor can be classified into the following three groups:

1. **Work done internal to the processor.**
2. **Work done external to the processor.**
3. **Operations initiated by the slaves or peripherals.**

The work done internal to the processor are additions, subtractions, logical operations, data transfer within registers, etc. The work done external to the processor are reading/writing the memory and reading/writing the IO devices or the peripherals. If the peripheral requires the attention of the master, then it can interrupt the master and initiate an operation.

The microprocessor is the master which controls all the activities of the system. To perform a specific job or task, the microprocessor has to execute a program stored in the memory. The program consists of a set of instructions stored in consecutive memory locations. In order to execute the program, the microprocessor issues address and control signals to fetch the instructions and data from the memory one by one. After fetching each instruction it decodes the instructions and performs the task specified by the instruction.

1.4.1 Concept of Multiplexing in a Microprocessor

Multiplexing is transferring different information at different well-defined times through the same lines. A group of such lines is called a multiplexed bus. The result of multiplexing is that fewer pins are required for microprocessors to communicate with the outside world.

Due to the pin number limitations, most microprocessors cannot provide simultaneously similar lines (such as address, data, status signals, etc.). Hence multiplexing of one or more of these buses is performed. Most often data lines are multiplexed with some or all address lines to form an address/data bus. (e.g., In 8085 the lower 8-address lines are multiplexed with data lines.) The status signals emitted by the microprocessor are sometimes multiplexed either with the data lines (as done in the INTEL 8080A) or with some of the address lines (as done in the INTEL 8086).

Whenever multiplexing is used, the CPU interface of the system must include the necessary hardware to demultiplex those lines to produce separate address, data and control buses required for the system. Demultiplexing of a multiplexed bus can be handled either at the CPU interface or locally at appropriate points in the system. Besides a slower system operation, a multiplexed bus also results in additional interface hardware requirements.

1.4.2 Demultiplexing of Address/Data Lines in an 8085 Microprocessor

In order to demultiplex the address/data lines (of the processor), the processor provides a signal called the ALE (**A**ddress **L**atch **E**nable). The ALE is asserted **high** and then **low** by the processor at the beginning of every machine cycle. At the same time the low byte address is given out through the AD_0 - AD_7 lines. The demultiplexing of address/data lines using an 8-bit D-latch 74LS373 is shown in Fig. 1.3.

The ALE is connected to the enable pin (EN) of an external 8-bit latch. When the ALE is asserted **high** and then **low**, the addresses are latched into the output lines of the latch. It holds the low byte of the address until the next machine cycle. After latching the address, the AD_0 - AD_7 lines are free for data transfer. The first T-state of every machine cycle is used for address latching in 8085 and the remaining T-states are used for reading or writing operation.

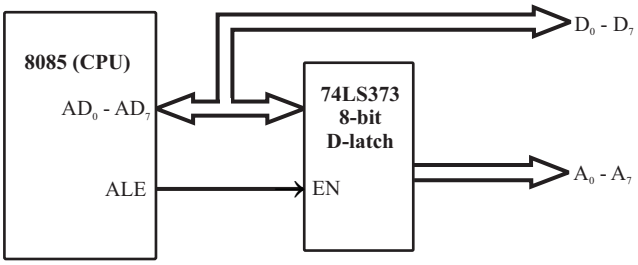


Fig. 1.3: Demultiplexing of address and data lines in an 8085 processor.

1.4.3 Demultiplexing of Address/Data Lines in an 8086 Microprocessor

In order to demultiplex the address/data lines (of the processor), the processor provides a signal called ALE (**A**ddress **L**atch **E**nable). The ALE is asserted **high** and then **low** by the processor at the beginning of every bus cycle. At the same time, the address is given out through AD_0-AD_{15} lines and A_{16}-A_{19}/status lines. Demultiplexing of address/data lines and address/status lines using 8-bit D-latch 74LS373 is shown in Fig. 1.4.

The ALE is connected to the **En**able Pin (EN) of the external 8-bit latches. When ALE is asserted **high** and then **low**, the addresses are latched into the output lines of the latch. It holds the address until the next bus cycle. After latching the address, the AD_0- AD_{15} lines are free for data transfer and A_{16} -A_{19}/status lines are free for carrying status information. The first T-state of every bus cycle is used for address latching in 8086 and the remaining T-states are used for reading or writing operation.

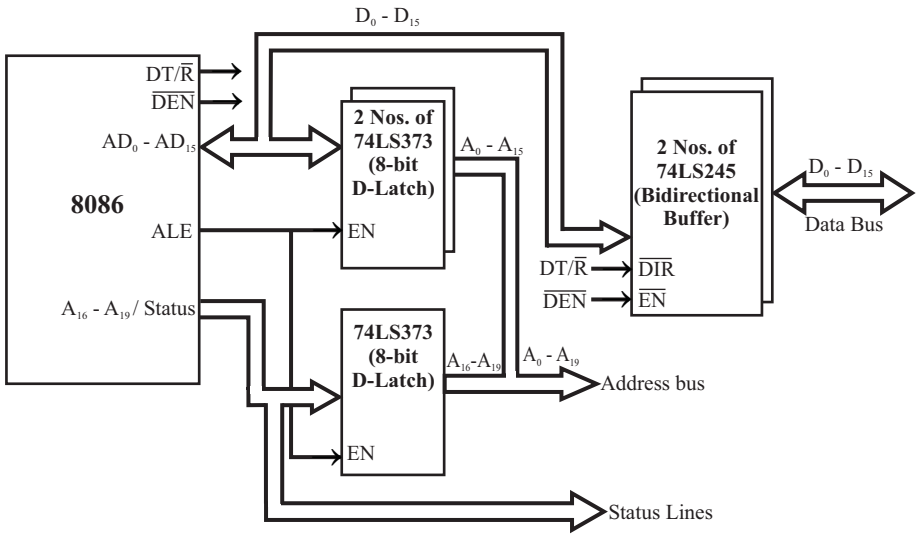


Fig. 1.4: Demultiplexing of address and data lines in an 8086 processor.

The data bus is provided with a bidirectional buffer in order to drive the data to a longer distance in the bus. The 8086 provides two control signals, DT/$\overline{R}$ and $\overline{DEN}$, for controlling the data buffers. DT/$\overline{R}$ is used to decide the direction of data flow and $\overline{DEN}$ is used to enable the data buffer.

1.5 INTEL 8085 MICROPROCESSOR

The INTEL 8085 is an 8-bit microprocessor released in the year 1976. The 8085 was originally designed using NMOS technology but now it is manufactured using HMOS technology and contains approximately 6500 transistors. The 8085 is packed in a 40-pin DIP (**D**ual **I**n-line **P**ackage) and requires a single 5V supply.

The 8085 has an internal clock oscillator. It generates a clock signal internally and divides by two for use as internal clock. This internal clock is also given out through the CLK pin for the clock requirement of peripheral devices.

The NMOS 8085 is available in two versions: 8085A and 8085A-2, with a maximum internal clock frequency of 3.03 MHz and 5 MHz respectively. The enhanced version of the 8085 is designed with HMOS transistors. It is available in three versions: 8085AH, 8085AH-2 and 8085AH-1 with maximum internal clock of 3 MHz, 5 MHz and 6 MHz respectively.

The basic data size of an 8085 is 8-bit. Therefore the memory word size of the memories interfaced with a 8085 processor is also 8-bit or byte. The 8085 uses a 16-bit address to access memory and hence it can address up to $2^{16} = 65{,}536_{10} = 64$ k memory locations. Since, one byte of information can be stored in one memory location, the maximum memory capacity of an 8085-based system is 64 kilobytes. For accessing IO-mapped devices, the 8085 uses a separate 8-bit address and so it can generate $2^8 = 256_{10}$ IO addresses.

1.5.1 Pin Configuration of an 8085 Microprocessor

The pin configuration of an 8085 microprocessor is shown in Fig. 1.5. The signals of the 8085 are listed in Table 1.1. The 8085 has 8 pins AD_0 to AD_7 for data transfer, which are multiplexed with low byte of address. The 8085 provides a signal ALE (**A**ddress **L**atch **E**nable) to demultiplex the low byte address and data using an external latch. The demultiplexing of address and data lines in an 8085 is shown in Fig. 1.3 in Section 1.4.2.

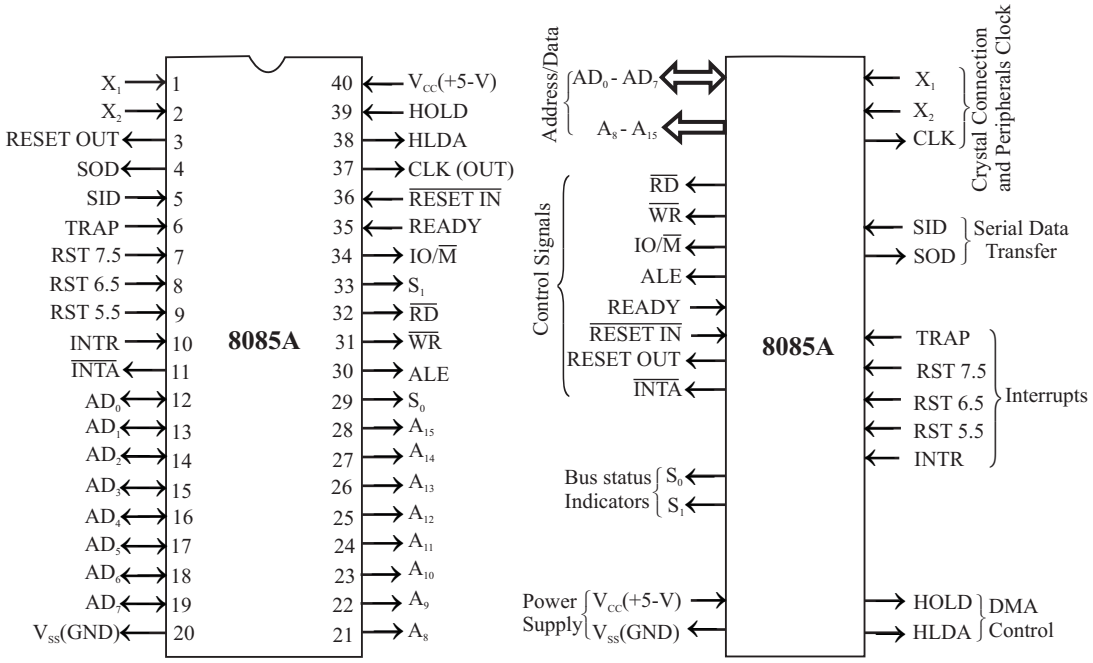


Fig. 1.5: 8085 microprocessor signals and pin assignment.

During memory access, the 16-bit memory address are output on AD_0 to AD_7 and A_8 to A_{15} lines. During IO access of IO-mapped devices the 8-bit IO address are output on both AD_0 to AD_7 and A_8 to A_{15} lines.

The 8085 processor differentiates the memory and IO address using the signal IO/$\overline{M}$. When the processor outputs a memory address, the IO/$\overline{M}$ is asserted **low** and when the processor outputs an IO address, the IO/$\overline{M}$ is asserted **high**.

The $\overline{RD}$ signal is asserted **low** by the processor during a memory or IO read operation. The $\overline{WR}$ signal is asserted **low** by the processor during a memory or IO write operation. The S_0 and S_1 are bus status indicators. The output signals on these lines during various bus activity (or machine cycles) are listed in Table 1.2.

Table 1.1: 8085 Signal Description Summary

Pin Name	Description	Type
AD_0 - AD_7	Address/Data	Bidirectional, Tristate
A_8 - A_{15}	Address	Output, Tristate
ALE	Address latch enable	Output, Tristate
$\overline{RD}$	Read control	Output, Tristate
$\overline{WR}$	Write control	Output, Tristate
IO/$\overline{M}$	IO or memory indicator	Output, Tristate
S_0, S_1	Bus state indicators	Output
READY	Wait state request	Input
SID	Serial input data	Input
SOD	Serial output data	Output
HOLD	Hold request	Input
HLDA	Hold acknowledge	Output
INTR	Interrupt request	Input
TRAP	Nonmaskable interrupt request	Input
RST 5.5	Hardware vectored interrupt request	Input
RST 6.5	Hardware vectored interrupt request	Input
RST 7.5	Hardware vectored interrupt request	Input
$\overline{INTA}$	Interrupt acknowledge	Output
$\overline{RESET\ IN}$	System reset	Input
$\overline{RESET\ OUT}$	Peripherals reset	Output
X_1, X_2	Crystal or RC connection	Input
CLK (OUT)	Clock signal	Output
V_{cc}	+5 V	Power supply
V_{ss}	Ground	Power supply

Note: *A overbar on the signal, indicates that it is active low. (i.e., the signal is normally high and when the signal is activated it is low).*

Table 1.2: Bus Status Signals

IO/$\overline{M}$	S_1	S_0	Operation performed by the 8085
0	0	1	Memory write
0	1	0	Memory read
1	0	1	IO write
1	1	0	IO read
0	1	1	Opcode fetch
1	1	1	Interrupt acknowledge

READY is an input signal that can be used by slow peripherals to get extra time in order to communicate with the 8085. The 8085 will work only when READY is tied to logic **high**. Whenever READY is tied to logic **low**, the 8085 will enter a wait state. When the system has slow peripheral devices, additional hardware is provided in the system to make the READY input **low** during the required extra time while executing a machine cycle, so that the processor will remain in wait state during this extra time.

The HOLD and HLDA signals are used for **D**irect **M**emory **A**ccess (DMA) type of data transfer. This type of data transfers are achieved by employing a DMA controller in the system. When DMA is required, the DMA controller will place a **high** signal on the HOLD pin of the 8085. When the HOLD input is asserted **high**, the processor will enter a wait state and drive all its tristate pins to a **high impedance** state and send an acknowledgement signal to the DMA controller through the HLDA pin. Upon receiving the acknowledgement signal, the DMA controller will take control of the bus and perform DMA transfer and at the end it asserts HOLD signal **low**. When HOLD is asserted **low** the processor will resume its execution.

The 8085 has five interrupt pins. The order of priority of the interrupts is TRAP, RST 7.5, RST 6.5, RST 5.5 and INTR. The interrupts TRAP, RST 7.5, RST 6.5 and RST 5.5 are hardware vectored interrupts and are enabled by appropriate signals at the appropriate pins of the 8085. When a vectored interrupt is enabled and if it is accepted then the program execution branches to the vector addresses specified by INTEL. The interrupts RST 7.5, RST 6.5 and RST 5.5 are maskable interrupts by software.

The INTR is enabled by appropriate signals at its pin. In order to service the INTR, one of the eight opcodes (RST 0 to RST 7) has to be provided on the AD_0 - AD_7 bus by external logic. The 8085 then executes this instruction and vectors to the appropriate address to service the interrupt. The vector address for an interrupt RST n is given by $\mathbf{(08 \times n)_H}$. The vector addresses of the interrrupts of 8085 are listed in Table 1.3. (The interrupt TRAP is RST 4.5.)

Table 1.3: Vector Addresses of Interrupts

Interrupt	Vector address	Interrupt	Vector address
RST 0	0000_H	RST 5	0028_H
RST 1	0008_H	RST 5.5	$002C_H$
RST 2	0010_H	RST 6	0030_H
RST 3	0018_H	RST 6.5	0034_H
RST 4	0020_H	RST 7	0038_H
TRAP	0024_H	RST 7.5	$003C_H$

The 8085 has the clock generation circuit on the chip but an external quartz crystal or LC circuit or RC circuit should be connected at the pins X_1 and X_2. The frequency at X_1 and X_2 is divided by two internally and used as an internal clock. The frequency of the output clock signal at the CLK(OUT) pin is same as that of the internal clock.

$\overline{\text{RESET IN}}$ is the system reset input signal and it is used to bring the processor to a known state. For proper reset, the $\overline{\text{RESET IN}}$ pin should be held **low** for at least three clock periods. When pin is asserted **low**, the program counter, instruction register, interrupt mask bits and all internal registers are cleared/reset. Also the RESET OUT signal is asserted **high** to clear/reset all the peripheral devices in the system. After a reset, the content of the program counter will be 0000_H and so the processor will start executing the program stored at 0000_H.

The pins SID and SOD can be used for serial data communication between the 8085 and any serial device under software control.

1.5.2 Driving X_1 and X_2 Inputs in an 8085 Microprocessor

The X_1 and X_2 pins of an 8085 processor are provided to connect an external quartz crystal or LC circuit. It can also be driven by an RC circuit or an external clock source. This connection is necessary for the internal oscillator to generate the clock signal for the processor. An oscillator consists of an amplifier and a feedback circuit. The feedback circuit of an oscillator can be of RC type, LC type or quartz crystal (a quartz crystal is electrically equivalent to an RLC circuit.) Also the feedback circuit decides the frequency of the signal generated by the oscillator.

In an 8085 processor, the oscillator circuit is provided internally except the feedback circuit. This feature facilitates the system designer to choose his own frequency for clock signals. But this frequency should not exceed the maximum clock frequency specified by the manufacturer.

Another reason for keeping feedback circuit external to the processor is that the high Q circuits (quartz crystal or large values of L) cannot be fabricated by IC technology.

In an 8085, the frequency generated by the oscillator circuit will be double that of the internal clock frequency. (The maximum clock frequencies specified by the manufacturer are internal clock frequencies.) In other words, the frequency at X_1 - X_2 pins of an 8085 is divided by two internally. This means that in order to obtain an internal clock of 3.03 MHz, a clock source of 6.06 MHz must be connected to X_1 - X_2. (Crystal/LC/RC should be designed for double the internal frequency.)

Quartz crystals are the best choice for connecting at X_1 - X_2, because they are less expensive, highly stable, have a large Q, occupy a very small space and frequencies do not drift with ageing. For crystals with less than 4 MHz, a capacitor of 20 pF should be connected between X_2 and ground to ensure the starting up of the crystal at the right frequency.

When an LC circuit is used, the value of L_{ext} and C_{ext} can be chosen using the formula,

$$f = \frac{1}{2\pi L_{ext}(C_{ext} + C_{int})}$$

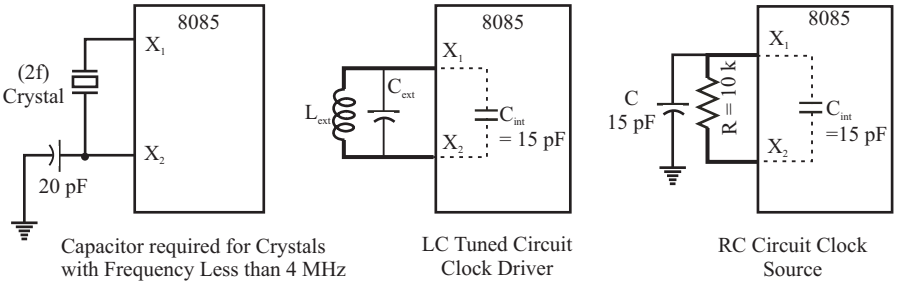


Fig. 1.6: Clock driver circuits for an 8085 microprocessor.

To minimize the variations in frequency, it is recommended that the value for C_{ext} should be chosen which is twice that of C_{int} or 30 pF. The use of LC circuit is not recommended for external frequencies higher than 5 MHz.

An RC circuit may also be used as the clock source for the 8085A if an accurate clock frequency is of no concern. Its advantage is the low component cost. The values shown in Fig. 1.6 are for generating an approximate external frequency of 3 MHz. Note that frequencies higher or lower than 3 MHz should not be attempted on this circuit.

1.5.3 Hardware Architecture of an 8085 Microprocessor

The architecture of an 8085 is shown in Fig. 1.8. The 8085 includes an ALU, a timing and control unit, a instruction register and a decoder, a register array, an interrupt control and a serial IO control.

The ALU performs the arithmetic and logical operations. The operations performed by the ALU of an 8085 are **addition, subtraction, increment, decrement, logical AND, OR, EXCLUSIVE-OR, compare, complement and left/right shift.** The accumulator and temporary register are used to hold the data during an arithmetic/logical operation. After an operation the result is stored in the accumulator and the flags are set or reset according to the result of the operation. The accumulator and flag register together is called the **P**rogram **S**tatus **W**ord (PSW).

There are five flags in an 8085: **Sign Flag (SF), Zero Flag (ZF), Auxiliary Carry Flag (AF), Parity Flag (PF) and Carry Flag (CF).** The bit positions reserved for these flags in the flag register are shown in Fig. 1.7.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SF	ZF		AF		PF		CF

Fig. 1.7: Bit positions of various flags in the flag register of 8085.

After an ALU operation if the most significant bit of the result is 1, the sign flag is set. The zero flag is set if the ALU operation results in zero and it is reset if the result is nonzero. In an arithmetic operation, when a carry is generated by the lower nibble, the auxiliary carry flag is set. After an arithmetic or logical operation if the result has an even number of 1's, the parity flag is set, otherwise it is reset.

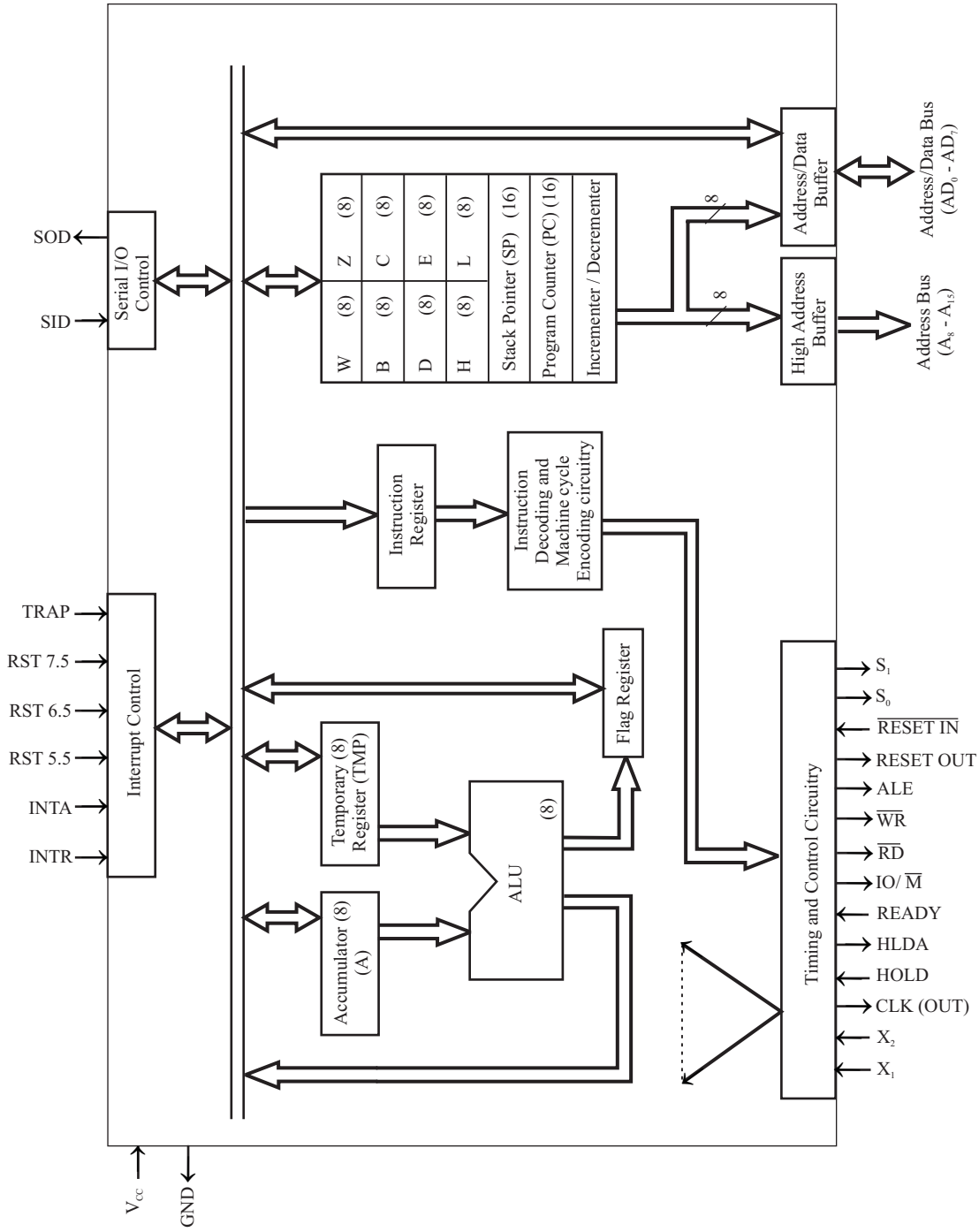


Fig. 1.8: Architecture of INTEL 8085 microprocessor.

If an arithmetic operation results in a carry, the carry flag is set, otherwise it is reset. Among the five flags, the AF flag is used internally for BCD arithmetic and other four flags can be used by the programmer to check the conditions of the result of an operation.

The **timing and control uni**t synchronizes all the microprocessor operations with the clock, and generates the control signals necessary for communication between the microprocessor and the peripherals.

When an instruction is fetched from the memory it is placed in the instruction register. Then it is decoded and encoded into various machine cycles. Apart from the **Accumulator** (A-register) there are six general purpose programmable registers B, C, D, E, H and L.

They can be used as 8-bit registers or paired to store 16-bit data. The allowed pairs are BC, DE and HL. The temporary registers TMP, W and Z cannot be used by the programmer.

The **S**tack **P**ointer SP holds the address of the stack top. The stack is a sequence of RAM memory locations defined by the programmer. The stack is used to save the content of the registers during the execution of a program.

The **P**rogram **C**ounter (PC) keeps track of program execution. To execute a program the starting address of the program is loaded in the program counter. The PC sends out an address to fetch a byte of instruction from memory and increment its content automatically. Hence when a byte of instruction is fetched, the PC holds the address of the next byte of the instruction or the next instruction.

1.5.4 Instruction Execution and Data Flow in an 8085 Microprocessor

The program instructions are stored in the memory, which is an external device. In order to execute a program in an 8085, the starting address of the program should be loaded in the program counter. The 8085 outputs the contents of the program counter to the address bus and asserts the read control signal **low**. Also, the program counter is incremented.

The address and the read control signal enables the memory to output the content of the memory location on the data bus. Now the content of the data bus is the opcode of an instruction.

The read control signal is made **high** by the timing and control unit after a specified time. At the rising edge of read control signals, the opcode is latched into the microprocessor internal bus and placed in the instruction register.

The instruction decoding unit decodes the instructions and provides information to the timing and control unit to take further action.

1.6 ZILOG Z80

The ZILOG Z80 is an 8-bit microprocessor manufactured using NMOS technology. The Z80 is available in a 40-pin DIP (**D**ual **I**n-line **P**ackage). It requires a single external clock and a single 5-V power supply. The maximum internal clock of standard Z80 is 2.5 MHz and for Z80-A it is 4 MHz. The Z80 provides more registers, extra addressing modes and a much larger instruction set than an 8085. It also has a built-in logic to refresh its dynamic RAM memories.

The signals of Z80 microprocessor and its functional block diagram (architecture) are shown in Figs. 1.9 and 1.10 respectively. The Z80 communicates with other system modules via three functionally separate buses: data, address and control buses.

The Z80 has separate pins for data and address. It operates on an 8-bit data and uses a 16-bit memory address. The physical memory size of the Z80 system is 64 kB. The IO devices can be mapped by memory-mapping or IO-mapping similar to that of 8085. For IO-mapped devices an 8-bit address is allotted. During memory refresh time, the seven lower-order bits of the address bus (A_0 - A_6) contain a valid refresh address.

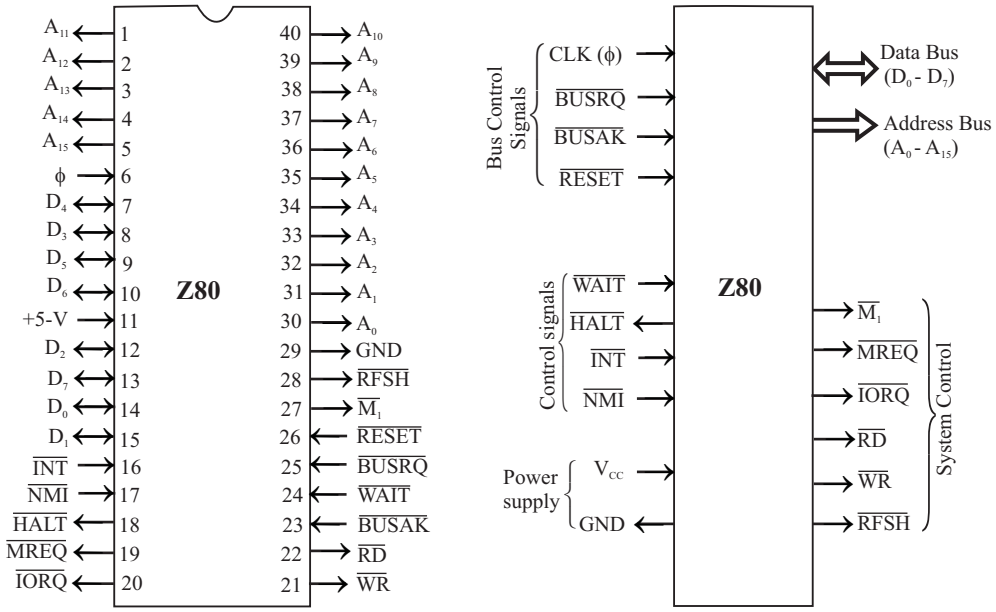


Fig. 1.9: Signals of Z80.

The control bus has three types of control signals. They are listed below:

1. System control signals

$\overline{M_1}$	-	**First machine cycle of an instruction**
$\overline{MREQ}$	-	**Memory request**
$\overline{IORQ}$	-	**IO request**
$\overline{RD}$	-	**Read control**
$\overline{WR}$	-	**Write control**
$\overline{RFSH}$	-	**Refresh cycle**

2. CPU control signals

$\overline{WAIT}$	-	**Wait request**
$\overline{HALT}$	-	**Halt request**
$\overline{INT}$	-	**Interrupt request**
$\overline{NMI}$	-	**Non-maskable interrupt**

3. Bus control signals

$\overline{\text{BUSRQ}}$	-	**BUS request**
$\overline{\text{BUSAK}}$	-	**BUS acknowledge**
$\overline{\text{RESET}}$	-	**System reset**
$\overline{\text{CLK}}$ (ϕ)	-	**Clock input**

The ALU is eight bits wide and performs similar functions to those of the 8085 ALU. The Z80 has two independent 8-bit accumulators A and A′ and two independent flag registers F and F′. The ALU operation involving accumulator A affects the flag register F. The ALU operation involving accumulator A' affects the flag register F′.

The flag registers have six flags: sign (S and S′), zero (Z and Z′), carry (C and C′), parity/over flow (P/V and P′/V′), half carry (H and H′) and subtract (N and N').

The Z80 has two sets of 8-bit general purpose register. Each set has 6 registers. They are B, C, D, E, H & L and B′, C′, D′, E′, H′, & L′. They can be used individually as 8-bit registers or as 16-bit register pairs. The allowed pairs are BC, DE & HL and B′C′, D′E′ & H′L′.

At any time instant the programmer can select and work with either the main register set or the alternate register set. To work in an alternate register set, the programmer has to use a single **Ex**change **I**nstruction (EXI) for the entire set of instructions. This alternate set allows the background mode of operation or handling fast interrupt response requirements while servicing an interrupt or executing a subroutine. While executing a program if one set of registers is not sufficient then we need not push them to stack, alternatively we can deactivate them without destroying its contents and switch to an alternate set of registers through exchange instructions.

The 16-bit **P**rogram **C**ounter (PC) and **S**tack **P**ointer (SP) registers are same as that of a 8085 microprocessor and operate in exactly the same way. The registers IX and IY allow two independent indexed addressing modes.

The Z80 includes an 8-bit interrupt vector (I). It is used in one of the interrupt response modes of the processor. It holds the upper eight bits of a memory pointer (or vector address). The lower eight bits of this pointer are supplied (as a vector number) by the interrupting device that requests service. The CPU then uses this 16-bit vector address to make an indirect call to the memory location that holds the first instruction of the interrupt service routine. This feature allows the vector table to be located anywhere in the memory.

The Z80 also contains an 8-bit memory refresh register (R) that contains the current memory refresh address, thus providing for automatic, totally transparent refresh of external dynamic RAM memories. Although the programmer can load this register for testing purposes, the R register is not normally used by the programmer.

The Z80 can execute 158 instruction types. The microprocessor includes all the instructions of an 8080A microprocessor with total software compatibility at the machine code level.

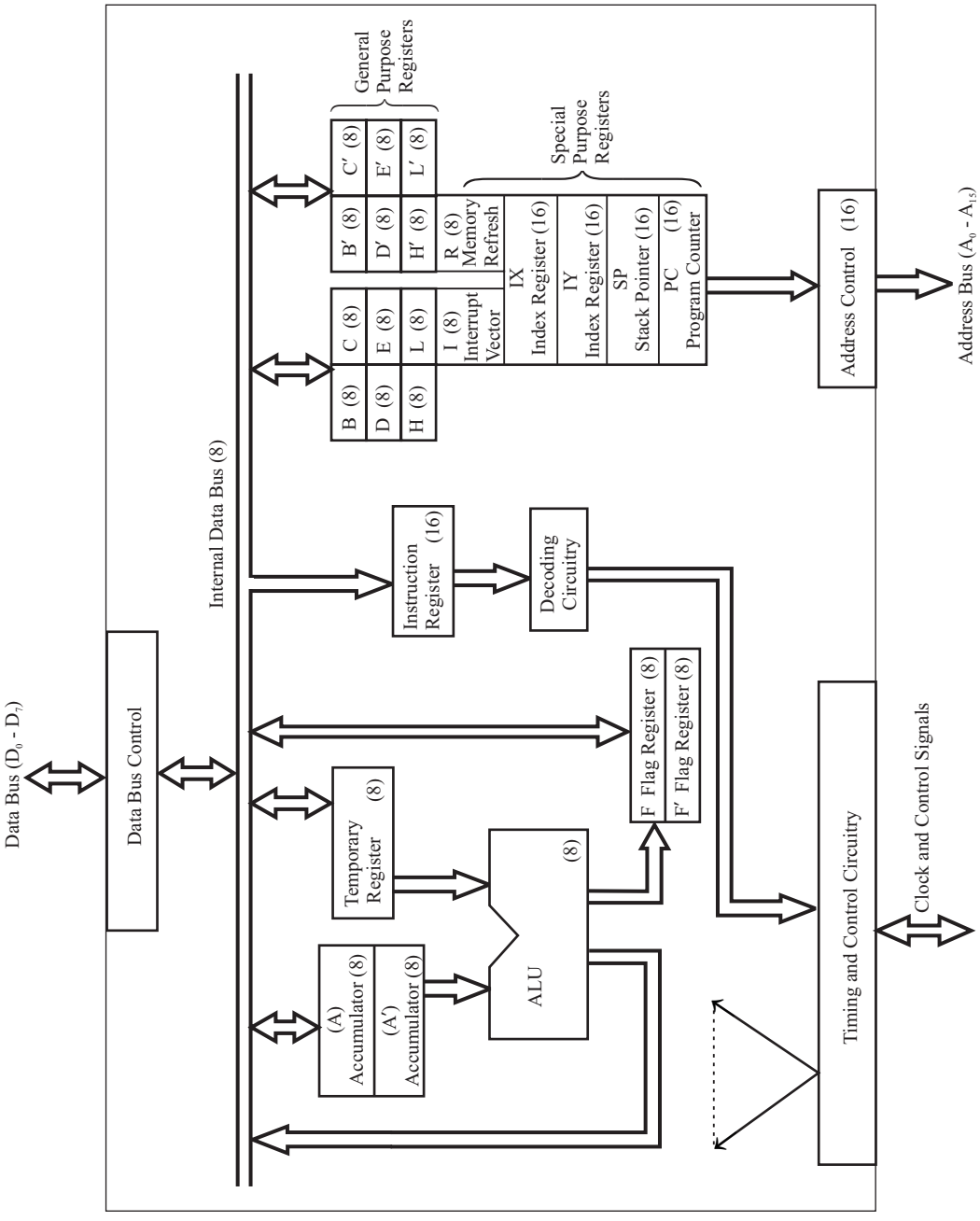


Fig. 1.10: Architecture of Z80 microprocessor.

> ***Note:*** *The 8085 has same instructions of 8080 except two new instructions SIM and RIM. Hence 8085 is also software compatible with Z80.*

The new instructions in Z80 include 1/4/8/16-bit operations, exchange instructions, block-transfer and block-search instructions and a full set of rotate and shift instructions applicable to any register, rather than just to the accumulator.

The size of a Z80 instruction is one to four bytes. A 1-byte instruction has just one-byte opcode. A 2-byte instruction has one or two byte opcode plus data-byte/device-number/displacement.

In multibyte instructions the opcode is one or two bytes. The remaining bytes are data/device-number/displacement/address.

The device-number is an 8-bit IO port address. The data-byte is the immediate operand. The displacement is a signed 2's complement number which is added to a 16-bit number residing in an index register, during indexed addressing.

Every Z80 instruction consists of one to six machine cycles. All types of machine cycles consist of either three or four states. Some Z80 instructions always insert wait states (T_w) between the states T_2 and T_3.

The basic operation of the Z80 is analogous to that of the INTEL 8085. The main difference is that instead of IO/$\overline{M}$ of 8085, the Z80 has $\overline{MREQ}$ and $\overline{IORQ}$. They are activated along with $\overline{RD}$ and $\overline{WR}$ for the memory or IO access.

1.7 MOTOROLA 6800

The Motorola 6800 product family was originally introduced in 1974. The 6800 microprocessor CPU is manufactured in NMOS technology on a 40-pin chip, has TTL compatible pins and it is the first 8-bit single chip microprocessor to exploit a single 5-V power supply.

The 6800 CPU can drive from seven to ten 6800 family devices without buffering. A two-phase external clock (1 MHz, maximum) must be externally supplied.

The signals of a motorola 6800 and its simplified functional block diagram are shown in Figs. 1.11 and 1.12 respectively.

The 6800 CPU has three buses to communicate with the other system modules, they are data, address and control buses. The data bus is bidirectional and has 8-lines, D_0-D_7.

The address bus has 16-lines, A_0-A_{15}. The processor operates on 8-bit data and uses a 16-bit address for memory and IO devices.

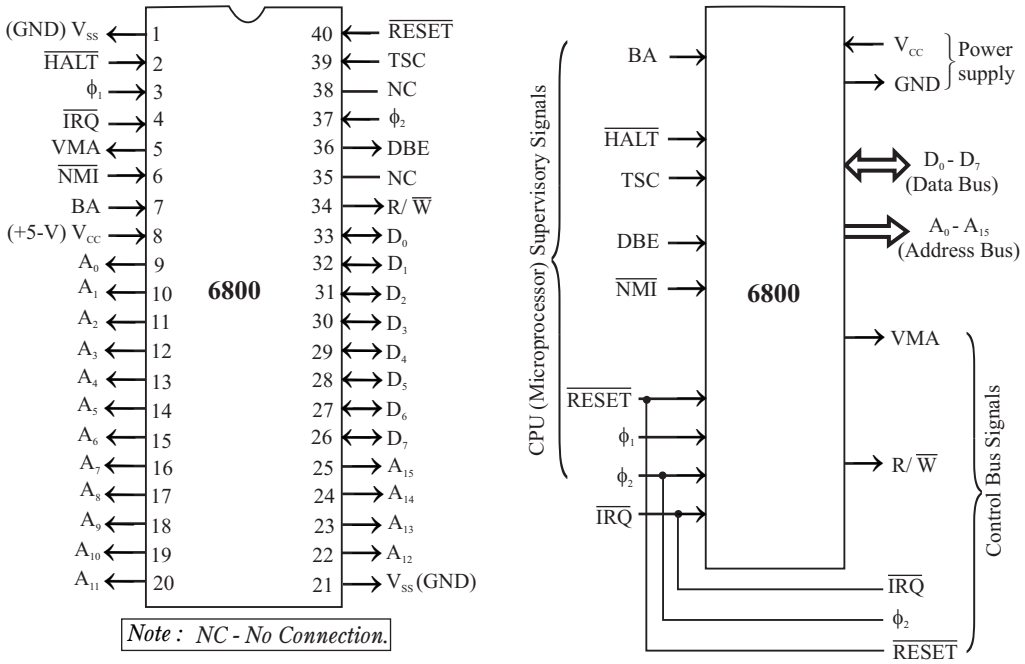


Fig.1.11: Signals of Motorola 6800.

This microprocessor does not distinguish between memory and peripheral addresses. Therefore some of the 64 k addresses must be reserved for peripheral addresses. The control bus carries two types of signals called **control bus signals** and **CPU (microprocessor) supervisory signals.**

Control bus signals :	**VMA**	-	**Valid memory address**
	R/$\overline{W}$	-	**Read/Write control**
	$\overline{IRQ}$	-	**Interrupt request**
	ϕ_2	-	**Phase-2 of clock**
	$\overline{RESET}$	-	**System reset**
CPU supervisory signals :	**BA**	-	**Bus acknowledge**
	$\overline{HALT}$	-	**Halt request**
	TSC	-	**Tristate control**
	DBE	-	**Data bus enable**
	$\overline{NMI}$	-	**Non-maskable interrupt**
	ϕ_1	-	**Phase-1 of clock**
	$\overline{IRQ}$	-	**Interrupt request**
	$\overline{RESET}$	-	**System reset**

The $\overline{HALT}$ pin is used for DMA data transfer in block-transfer mode or a cycle-stealing mode. When $\overline{HALT}$ is asserted **low** the microprocessor halts all its activity at the completion of the current instruction.

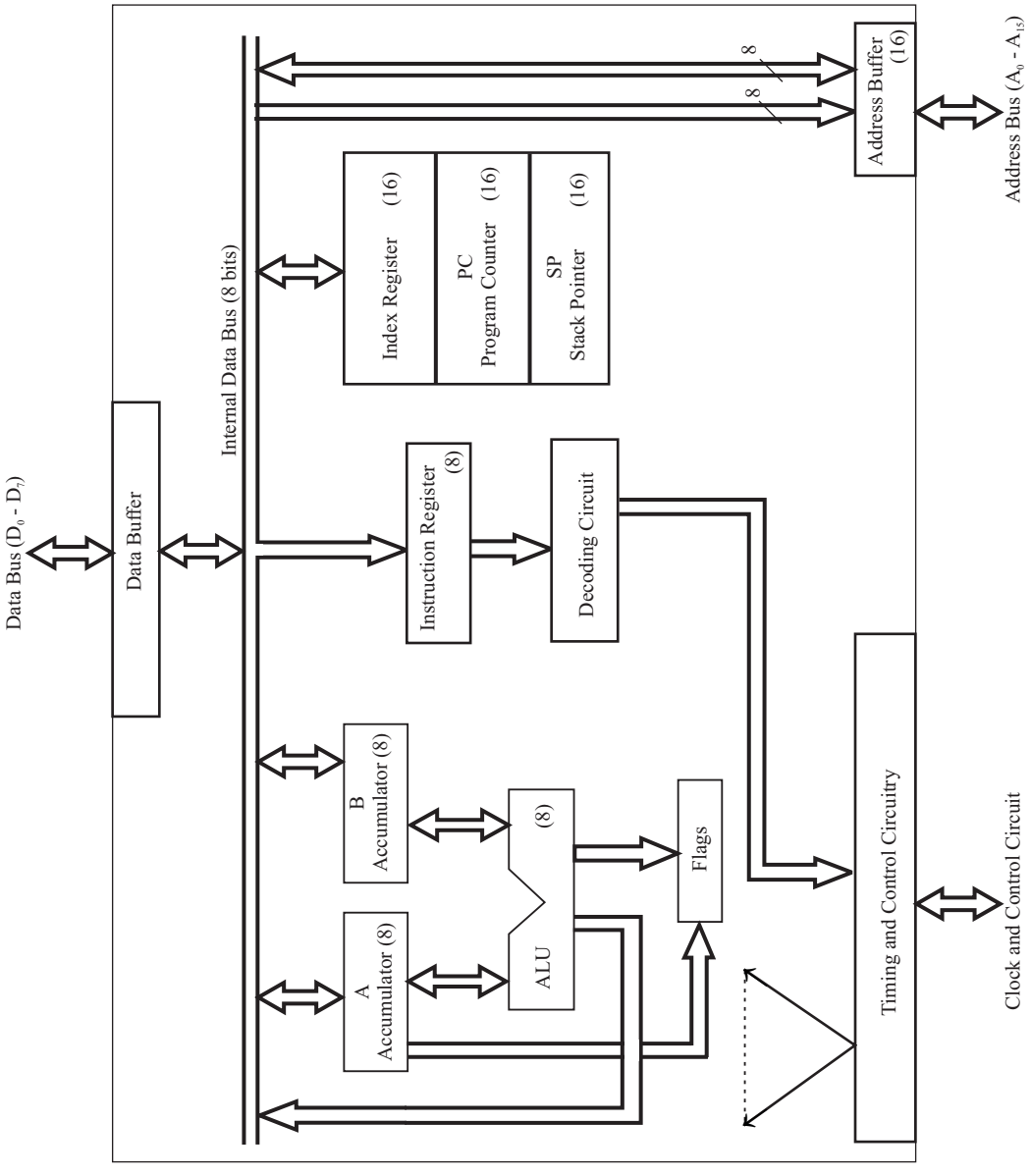


Fig. 1.12: Architecture of Motorola 6800 microprocessor.

The **Tri**state **C**ontrol (TSC) may be used to implement DMA on a cycle-stealing basis. If TSC is placed in a **high** state, the address bus and the R/W line go to a **high impedance** state 500 ns later. The data bus is not affected by TSC and has its own enable (DBE). This approach assures rapid response to the DMA request. Since the internal memory of the 6800 is dynamic, the TSC terminal cannot be held in **high** state for longer than 5 μs, if loss of data in the microprocessor is to be avoided.

The architecture of 6800 includes the ALU, a 16-bit **P**rogram **C**ounter (PC), a 16-bit stack pointer, a 16-bit index or general purpose register, a two 8-bit accumulators and a condition code register.

The stack pointer allows a **L**ast-**I**n-**F**irst-**O**ut (LIFO) stack to be implemented at any address in the memory and to be limited in size only by the memory space. The index register may be used to store data or a 16-bit memory address for use in the indexed mode of addressing. The **C**ondition **C**ode **R**egister (CCR) indicates the results of an ALU operation. The flags in the CCR are **N**egative (N), **Z**ero(Z), **O**verflow (O), **C**arry(C), **H**alf carry (H) and **I**nterrupt enable/disable (I). The unused bits of the CCR are the 1's.

The ALU performs arithmetic and logical operations including AND, OR, EXCLUSIVE-OR, NEGATE, COMPARE, ADD, SUBTRACT and DECIMAL ADJUST which allows BCD arithmetic to be performed. Immediate, direct, indexed and relative addressing modes are used in 6800.

In the indexed addressing mode, the address contained in the second byte of the instruction is added to the lowest eight bits of the index register. The carry is then added to the higher order bits of the index register. The result is used to address memory.

In relative addressing the address contained in the second byte of the instruction is added to the lowest eight bits of the PC. To this result, a value of +**2** is added, which allows the user to address the data within a range of – 125 to + 129 bytes of the present instruction.

The 6800 has a set of 72 instructions. They are classified as data handling, arithmetic, logic, control transfer, data test, condition codes, address maintenance and interrupt handling.

The data handling instructions include several instructions for moving data between two accumulators, memory and the stack. Data may be altered with Clear, Increment, Decrement, Complement (1's and 2's), Rotate and Shift instructions.

The arithmetic instructions include Add, Subtract and Decimal Adjust Accumulator. The AND, OR and EXCLUSIVE-OR comprise the logical instructions.

The control transfer instructions include Unconditional Branch, Jump and Jump-to-subroutine. The Branch instruction uses relative addressing while the Jump instruction uses direct or indirect addressing. A number of conditional branches are available which test the condition of one or more bits of the condition code register.

The data test instructions set the condition codes (alter the flags) without altering the data. They include Bit Test (for comparing individual bits of accumulator A or B with a memory word), Compare and Test (for determining the sign of a number).

Condition code instructions are provided which enable the programmer to set or reset directly the Carry, Interrupt or Over flow flags. The entire contents of the condition code register may be moved to or from the accumulator A with a single instruction. Eleven instructions are provided for address maintenance. These instructions allow operations on the index register, e.g., Compare, Increment, Decrement and Transfer to or from the memory or the stack pointer. Similar instructions are available for operation on addresses stored in the stack pointer.

The interrupt handling instructions include a **s**oft**w**are **i**nterrupt (SWI) which stores the status of the processor in the stack before processing the interrupt and a **Re**t**u**rn from **I**nterrupt (RTI) instruction which restores the status of the microprocessor after an interrupt is processed. A **Wa**it for **I**nterrupt (WAI) instruction causes the status to be stored in the stack and places the processor in a halt condition until a hardware interrupt occurs.

A 6800 instruction may be one, two or three bytes long, its length being closely related to the addressing mode used.

Usually every 6800 instruction cycle consists of two to eight machine cycles, all of which are identical in length (except interrupt instructions which require longer instruction execution cycles.) In the 6800, a machine cycle is one and the same thing as a clock cycle (or state.)

The operation of the 6800 is very simple, since it consists of only three types of machine cycles: a read machine cycle (during which a byte of data is input into the CPU), a write machine cycle (during which a byte of data is output by the CPU) and an interrupt operation machine cycle (during which the CPU is busy and no activity occurs on the system buses.) The timing of any 6800 instruction is simply a concatenation of these three basic machine cycle types.

The control signals required to access a memory location are R/$\overline{W}$, VMA and DBE. Under normal circumstances, DBE is identical to ϕ_2. The signal R/$\overline{W}$ controls the reading or writing operation. For read operation R/$\overline{W}$ is asserted **high** and for write operation R/$\overline{W}$ is asserted **low**.

1.8 INTEL 8086 MICROPROCESSOR

The INTEL 8086 is the first 16-bit processor released by INTEL in the year 1978. The 8086 is designed using the HMOS technology and now it is manufactured using HMOS III technology and contains approximately 29,000 transistors. The 8086 is packed in a 40-pin DIP and requires a single 5-V supply.

The 8086 does not have an internal clock circuit. The 8086 requires an external asymmetric clock source with 33% duty cycle. An 8284 clock generator is used to generate the required clock for 8086. The maximum internal clock of 8086 is 5 MHz. The other versions of 8086 with different clock rates are 8086-1, 8086-2 and 8086-4 with maximum internal clock frequency of 10 MHz, 8 MHz and 4 MHz respectively.

The 8086 uses a 20-bit address to access memory and hence it can directly address up to one megabytes (220 = 1 Mega) of memory space. The one megabyte (1MB) of addressable memory space of 8086 are organized as two memory banks of 512 kilobytes each (512 kB + 512 kB = 1MB).

The memory banks are called even (or lower) bank and odd (or upper) bank. The address line A_0 is used to select even bank and the control signal $\overline{BHE}$ is used to select odd bank.

For accessing IO-mapped devices, the 8086 uses a separate 16-bit address, and so the 8086 can generate 64k (2^{16}) IO addresses. The signal M/$\overline{IO}$ is used to differentiate the memory and IO addresses. For memory address the signal M/$\overline{IO}$ is asserted **high** and for IO address the signal M/IO is asserted **low** by the processor.

The 8086 can operate in two modes: minimum mode and maximum mode. The mode is decided by a signal at MN/$\overline{MX}$ pin. When the MN/$\overline{MX}$ is tied **high** it works in minimum mode and the system is called a uniprocessor system. When MN/$\overline{MX}$ is tied **low** it works in maximum mode and the system is called a multiprocessor system. Usually the pin MN/$\overline{MX}$ is permanently tied to **low** or **high** so that the 8086 system can work in any one of the two modes. The 8086 can work with an 8087 coprocessor in maximum mode. In this mode an external bus controller 8288 is required to generate bus control signals.

The 8086 has two families of processors. They are 8086 and 8088. The 8088 uses 8-bit data bus externally but 8086 uses 16-bit data bus externally. The 8086 access memory is in words but 8088 access memory is in bytes. IBM designed its first **P**ersonal **C**omputer (PC) using an INTEL 8088 microprocessor as the CPU.

1.8.1 Pins and Signals of INTEL 8086

The 8086 pins and signals are shown in Fig. 1.13. The 8086 is a 40-pin IC and all the 8086 pins are TTL compatible. The signal assigned to pins 24 to 31 is different for minimum and maximum mode of operation. The signal assigned to all the other pins are common for minimum and maximum mode of operation.

Table 1.4: Common Signals

Name	Description/Function	Type
AD_{15} - AD_0	Address/Data	Bidirectional, Tristate
A_{19}/S_6 - A_{16}/S_3	Address/Status	Output, Tristate
$\overline{BHE}/S_7$	Bus high enable/Status	Output, Tristate
MN/$\overline{MX}$	Minimum/Maximum mode control	Input
$\overline{RD}$	Read control	Output, Tristate
$\overline{TEST}$	Wait on test control	Input
READY	Wait state control	Input
RESET	System reset	Input
NMI	Non-maskable interrupt request	Input
INTR	Interrupt request	Input
CLK	System clock	Input
V_{cc}	+ 5-V	Power supply input
GND	Ground	Power supply ground

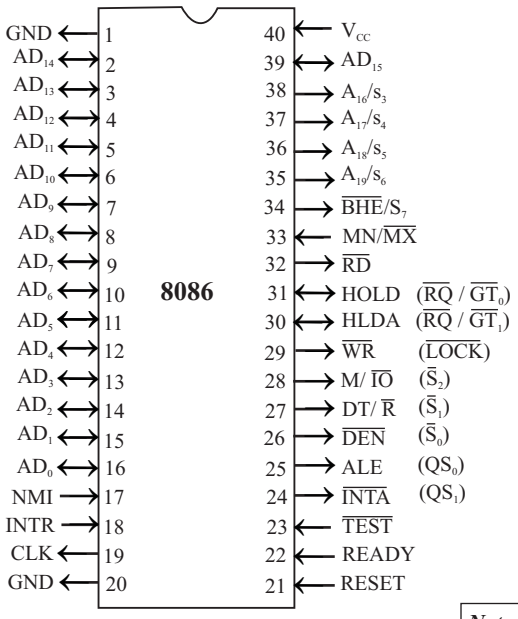


Fig. a : 8086 pin assignments.

***Note:** Signals shown in parenthesis are maximum mode signals.*

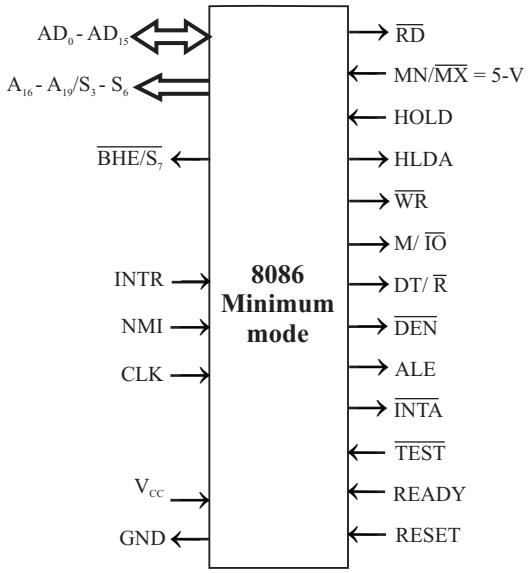


Fig. b: 8086-Minimum mode.

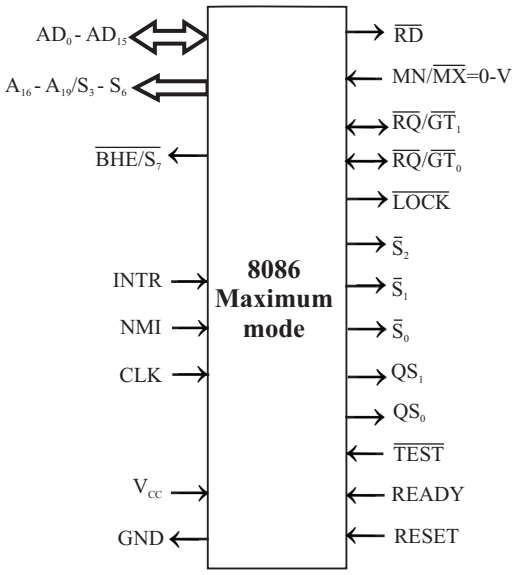


Fig. c: 8086-Maximum mode.

Fig. 1.13: 8086 signals and pin assignment.

Table 1.5: Minimum Mode Signals [MN / $\overline{MX}$ = V_{cc} (Logic high)]

Name	Description / Function	Type
HOLD	Hold request	Input
HLDA	Hold acknowledge	Output
$\overline{WR}$	Write control	Output, Tristate
M/$\overline{IO}$	Memory / IO control	Output, Tristate
DT/$\overline{R}$	Data transmit / Receive	Output, Tristate
$\overline{DEN}$	Data enable	Output, Tristate
ALE	Address latch enable	Output
$\overline{INTA}$	Interrupt acknowledge	Output

Table 1.6: Maximum Mode Signals [MN / $\overline{MX}$ = GROUND (Logic low)]

Name	Description / Function	Type
$\overline{RQ}/\overline{GT}_1$, $\overline{RQ}/\overline{GT}_0$	Request / Grant bus access control	Bidirectional
$\overline{LOCK}$	Bus priority lock control	Output, Tristate
$\overline{S}_2$, $\overline{S}_1$, $\overline{S}_0$	Bus cycle status	Output, Tristate
QS_1, QS_0	Instruction queue status	Output

1.8.2 Common Signals

The signals common for minimum and maximum mode are listed in Table 1.4. The lower sixteen lines of address are multiplexed with data and the upper four lines of address are multiplexed with status signals. During the first clock period of a bus cycle the entire 20-bit address is available on these lines. During all other clock periods of a bus cycle, the data and status signals will be available on these lines.

The status signals on S_3 and S_4 specifies the segment register used for calculating the physical address. The output on the status lines S_3 and S_4 when the processor is accessing various segments are listed in Table 1.7.

Table 1.7: Status Signals During Memory Segment Access

Status signal		Segment register
S_4	S_3	
0	0	Extra segment
0	1	Stack segment
1	0	Code or no segment
1	1	Data segment

The status lines S_3 and S_4 can be used to expand the memory up to 4 Mb. The status line S_5 indicates the status of an 8086 interrupt enable flag. A **low** on the line S_6 indicates that 8086 is on the bus (i.e., it indicates that 8086 is the bus master) and during hold acknowledge this pin is driven to **high impedance** state. The output signal $\overline{BHE}$ on the first T-state of a bus cycle is maintained as status signal S_7 on the same pin.

The 8086 outputs a **low** on $\overline{BHE}$ pin during read, write and interrupt acknowledge cycles when the data is to be transferred to the high-order data bus. The $\overline{BHE}$ can be used in conjunction with AD_0 to select memory banks.

When the processor reads from memory or an IO location it asserts $\overline{RD}$ **low**. The $\overline{TEST}$ input is tested by the WAIT instruction. The 8086 will enter a wait state after execution of the WAIT instruction, and it will resume execution only when $\overline{TEST}$ is made **low** by an external hardware. This is used to synchronize an external activity to the processor internal operation. $\overline{TEST}$ input is synchronized internally during each clock cycle on the leading edge of the clock signal.

INTR is the maskable interrupt and INTR must be held **high** until it is recognized to generate an interrupt signal. NMI is the non-maskable interrupt input activated by a leading edge signal.

RESET is the system reset input signal. For power-ON reset it is held **high** for 50 micro-second. For reset while working, it is held **high** for at least four clock cycles. When the processor is resetted, the DS, SS, ES, IP and flag register are cleared, **C**ode **S**egment (CS) register is initialized to $FFFF_H$ and queue is emptied. After reset the processor will start fetching instruction from 20-bit physical address $FFFF0_H$.

READY is an input signal to the processor, used by the memory or IO devices to get extra time for data transfer or to introduce **wait states** in the bus cycles. Normally READY is tied **high**. If the READY is tied **low**, the 8086 introduces wait states after second T-state of a bus cycle and it will complete the bus cycle only when READY is made **high** again.

CLK input is the clock signal that provides basic timing for the 8086 and bus controller. The 8086 does not have an on-chip clock generation circuit. Hence the 8284 clock generator chip is used to generate the required clock. A quartz crystal whose frequency is thrice that of the internal clock of 8086 must be connected to the 8284. The 8284 generates the clock at crystal frequency. It divides the generated clock by three and modifies the duty cycle to 33% and output on the CLK pin of the 8284. This CLK output of the 8284 must be connected to the 8086 CLK pin. The 8284 also provides the RESET and READY signal to an 8086.

1.8.3 Minimum Mode Signals

The minimum mode signals of an 8086 are listed in Table 1.5. For minimum mode of operation the MN/$\overline{MX}$ pin is tied to V_{cc}(logic **high**). In minimum mode, the 8086 itself generates all bus control signals. The minimum mode signals are explained below:

DT/$\overline{R}$ - [***D**ata **T**ransmit / **R**eceive*] It is an output signal from the processor to control the direction of data flow through the data transceivers.

$\overline{DEN}$ - (***D**ata **E**nable*) - It is an output signal from the processor used as output enable for the data transceivers.

ALE - (***A**ddress **L**atch **E**nable*) - It is used to demultiplex the address and data lines using external latches.

M/$\overline{IO}$ - It is used to differentiate memory access and IO access. For IN and OUT instructions it is asserted **low**. For memory reference instructions it is asserted **high.**

$\overline{WR}$ - It is a write control signal and it is asserted **low** whenever the processor writes data to memory or IO port.

$\overline{INTA}$ - (*Interrupt Acknowledge*) - The 8086 output is asserted **low** on this line to acknowledge when the interrupt request is accepted by the processor.

HOLD - It is an input signal to the processor from other bus masters as a request to grant control of the bus. It is usually used by the DMA controller to get control of the bus.

HLDA - (*Hold Acknowledge*) - It is an acknowledge signal by the processor to the master requesting the control of the bus through HOLD. The acknowledge is asserted **high** when the processor accepts the HOLD. [*On accepting the hold the processor drives all the tristate pins to* ***high impedance*** *state and sends an acknowledgement to the device which requested HOLD. On receiving the acknowledgement the other master will take control of the bus*.]

1.8.4 Maximum Mode Signals

The maximum mode signals of an 8086 are listed in Table 1.6. An 8086-based system can be made to work in maximum mode by grounding the MN/$\overline{MX}$ pin (i.e., MN/$\overline{MX}$ is tied to logic **low**). In maximum mode, the pins 24 to 31 are redefined as follows:

$\overline{S}_0$, $\overline{S}_1$, $\overline{S}_2$ - These are status signals and they are used by the 8288 bus controller to generate bus timing and control signals. The status signals are decoded as shown in Table 1.8.

Table 1.8: Status Signals During Various Machine Cycles

Status Signal			Machine Cycle
S_2	S_1	S_0	
0	0	0	Interrupt acknowledge
0	0	1	Read IO port
0	1	0	Write IO port
0	1	1	Halt
1	0	0	Code access
1	0	1	Read memory
1	1	0	Write memory
1	1	1	Passive/Inactive

$\overline{RQ}/\overline{GT}_0$, $\overline{RQ}/\overline{GT}_1$ - (Bus Request/Bus Grant) These requests are used by the other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle These pins are bidirectional. The request on GT_0will have higher priority than GT_1.

The bus request to an 8086 works as follows:

1. **When a local bus master requires an system bus control, it sends a low pulse to the 8086.**
2. **At the end of the current bus cycle, the processor (8086) drives its pins to high impedance state and sends an acknowledgement as a low pulse on the same pin to the device which had requested the bus control.**

3. **On receiving the acknowledgement the local master will take control of the system bus. After completing its work, at the end, the local bus master sends a low signal on the same pin to 8086 to inform the end of control. Now 8086 regains the control of the bus.**

$\overline{\text{LOCK}}$ - It an output signal activated by the LOCK prefix instruction and remains active until the completion of the instruction prefixed by LOCK. The 8086 asserts the **$\overline{\text{LOCK}}$** pin low while executing an instruction prefixed by LOCK to prevent other bus masters from gaining control of the system bus.

QS_1, QS_0 - (Queue Status) - The processor provides the status of queue on these lines. The queue status can be used by the external device to track the internal status of the queue in an 8086. The QS_0 and QS_1 are valid during the clock period following any queue operation. The output on QS_0 and QS_1 can be interpretted as shown in Table 1.9.

Table 1.9: Queue Status

Queue status		Queue operation
QS_1	QS_0	
0	0	No operation
0	1	First byte of an opcode from the queue
1	0	Empty the queue
1	1	Subsequent byte from the queue

1.8.5 Architecture of INTEL 8086

The 8086 has a pipelined architecture. In pipelined architecture the processor will have a number of functional units and the execution time of the functional units are overlapped. Each functional unit works independently most of the time. The simplified block diagram of the internal architecture of an 8086 is shown in Fig. 1.14. The architecture of the 8086 can be internally divided into two separate functional units: **B**us **I**nterface **U**nit (BIU) and **E**xecution **U**nit (EU).

The BIU fetches instructions, reads data from memory and IO ports and writes data to memory and IO ports. The BIU contains segment registers, an instruction pointer, an instruction queue, an address generation unit and a bus control unit. The EU executes instructions that have already been fetched by the BIU. The BIU and EU function independently.

The instruction queue is a FIFO (**F**irst-**I**n-**F**irst-**O**ut) group of registers. The size of the queue is 6 bytes. The BIU fetches the instruction code from memory and stores it in queue. The EU fetches the instruction codes from the queue.

The BIU has four 16-bit segment registers: **C**ode **S**egment (CS) register, **D**ata **S**egment (DS) register, **S**tack **S**egment (SS) register and **E**xtra **S**egment (ES) register. The 8086 memory space can be divided into segments of 64 kB. The 4-segment registers are used to hold four segment base addresses. Hence 8086 can directly address 4 segments of 64 kB at any time instant ($4 \times 64 = 256$ kB within 1MB memory space). This feature of the 8086 allows the system designer to allocate separate areas for storing program codes and data.

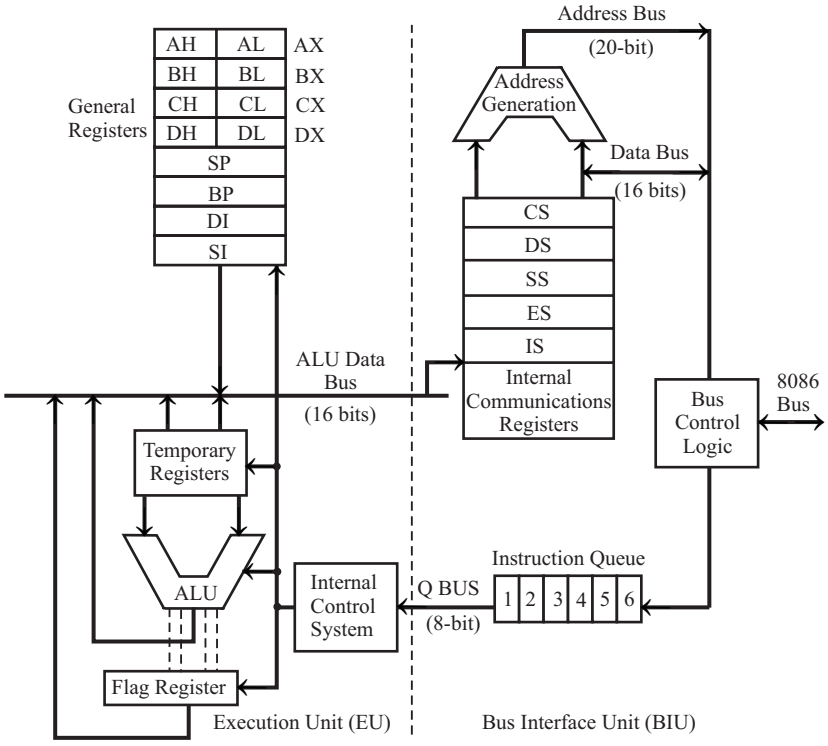


Fig. 1.14: Internal architecture of 8086.

The contents of the segment registers are programmable. Hence the processor can access the code and data in any part of the memory by changing the contents of the segment registers. The memory segment can be continuous, partially overlapped, fully overlapped or disjointed.

> ***Note:*** *Since segment registers are programmable it is possible to design multitasking and multiuser systems using 8086. The program code and data for each task/user can be stored in separate segments. The program execution can be switched from one task/user to another by changing the contents of the segment registers.*

The dedicated address generation unit generates a 20-bit physical address from the segment base and an offset or effective address. The segment base address is logically shifted left four times and added to the offset. *[logically shifting left four times is equal to multiplying it by 16_{10}.]*

The address for fetching instruction codes is generated by logically shifting the content of the CS to the left four times and then adding it to the content of the IP (**I**nstruction **P**ointer). The IP holds the offset address of the program codes. The content of the IP gets incremented by two after every bus cycle. *[In one bus cycle the processor fetches two bytes of the instruction code.]*

The data address is computed by using the content of the DS or ES as the base address and an offset or effective address specified by the instruction.

The stack address is computed by using the content of the SS as the base address and the content of the SP (**S**tack **P**ointer) as the offset address or effective address.

The bus control logic of the BIU generates all the bus control signals such as read and write signals for memory and IO. The EU consists of the ALU, the flag register and the general purpose registers. The EU decodes and executes the instructions. A decoder in the EU control system translates the instructions.

The EU has a 16-bit ALU to perform arithmetic and logical operations. The EU has eight numbers of 16-bit general purpose registers. They are AX, BX, CX, DX, SP, BP, SI and DI.

Some of the 16-bit registers can also be used as two numbers of 8-bit registers as given below:

AX - can be used as AH and AL

BX - can be used as BH and BL

CX - can be used as CH and CL

DX - can be used as DH and DL

The general purpose registers can be used for data storage when they are not involved in any special functions assigned to them. These registers are named after special functions carried out by each one of them as given in Table 1.10.

Table 1.10: Special Functions of 8086 Registers

Register	Name of the register	Special function
AX	16-bit Accumulator	Stores the 16-bit result of certain arithmetic and logical operations.
AL	8-bit Accumulator	Stores the 8-bit result of certain arithmetic and logical operations.
BX	Base Register	Used to hold the base value in base addressing mode to access memory data
CX	Count Register	Used to hold the count value in SHIFT, ROTATE and LOOP instructions.
DX	Data Register	Used to hold data for multiplication and division operations.
SP	Stack Pointer	Used to hold the offset address of top of stack memory.
BP	Base Pointer	Used to hold the base value in base addressing using stack segment register to access data from stack memory.
SI	Source Index	Used to hold the index value of source operand (data) for string instructions.
DI	Destination Index	Used to hold the index value of destination operand (data) for string instruction.

1.8.6 8086 Flag Register

The size of an 8086 flag register is 16 bits and in this nine bits are defined as flags. The six flags are used to indicate the status of the result of the arithmetic or logical operations. Three flags are used to control the processor operation and so they are also called control bits. The various flags of an 8086 processor and their bit position in flag register are shown in Fig. 1.15.

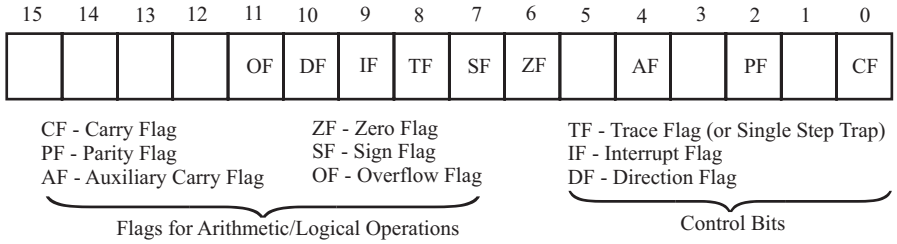


Fig. 1.15: Bit positions of various flags in the flag register of 8086.

The **C**arry **F**lag (CF) is set if there is a carry from the addition or borrow from the subtraction. **A**uxiliary carry **F**lag (AF) is set if there is a carry from low nibble to high nibble of the low order 8-bit of a 16-bit number.

The **O**verflow **F**lag (OF) is set to **one** if there is an arithmetic overflow, that is, if the size of the result exceeds the capacity of the destination location. **S**ign **F**lag (SF) is set to **one** if the most significant bit of the result is **one** and SF is cleared to **zero** for non-negative result. The **P**arity **F**lag (PF) is set to **one** if the result has even parity and PF is cleared to **zero** for odd parity of the result. The **Z**ero **F**lag (ZF) is set to **one** if the result is zero and ZF is cleared to **zero** for a non zero result.

The three control bits in the flag register can be set or reset by the programmer. The **D**irection **F**lag (DF) is set to **one** for autodecrement and reset to **zero** for autoincrement of the SI and DI registers during string data accessing. Setting **I**nterrupt **F**lag (IF) to **one** causes the 8086 to recognize the external maskable interrupts, and clearing IF to **zero** disables the interrupts.

Setting **T**race **F**lag (TF) to **one**, places the 8086 in the single step mode. In this mode the 8086 generates an internal interrupt after execution of each instruction. The single stepping is used for debugging a program.

1.8.7 Instruction and Data Flow in 8086

The 8086 microprocessor allows the user to define different memory areas for storing the program and data. The program memory can be accessed using CS-register and the data memory can be accessed using DS, ES and SS registers.

The program instructions are stored in the program memory which is an external device. To execute a program in 8086, the base address and offset address of the first instruction of the program should be loaded in CS-register and IP respectively.

The 8086 computes the 20-bit physical address of the program instruction by multiplying the content of the CS-register by 16_{10} and adding to the content of the IP. The 20-bit physical address is given out on the address bus.

Then $\overline{RD}$ is asserted **low**. Also other control signals necessary for program memory read operation are asserted. The IP is incremented by two to point the next instruction or the next word of the same instruction.

The address and control signals enable the memory to output one word (two bytes) of program memory on the data bus. After a predefined time the $\overline{RD}$ is asserted **high** and at this instant the content of the data bus is latched into two empty locations of the instruction queue. Then the BIU starts fetching the next word, of the program code as explained above. The BIU keeps on fetching the program codes, word by word, from consecutive memory locations whenever two locations of queue is empty. When a branch instruction is encountered, the queue is emptied and then filled with program codes from the new address loaded in the CS and IP by the branch instruction.

The EU reads the program instructions from the queue, decodes and executes them one by one. If the execution of an instruction requires data from memory (or to store data in memory) then BIU is interrupted to read (or write) data in memory. When BIU is interrupted it completes the fetching of the current instruction word and then starts reading/writing the data by generating a 20-bit data memory address. The 20-bit data memory address is obtained by multiplying the content of the segment base register specified by the instruction by 16_{10} and adding to an effective or offset address specified by the instruction.

1.9 MOTOROLA MC68000

The MC68000 is Motorola's first 16-bit microprocessor. It has a 16-bit data bus and 24-bit address bus to address up to 16 MB of physical memory space. The 16 MB physical memory space is organized as two banks of 8 MB each. The MC68000 is designed using HMOS transistors and requires a single +5-V supply.

In the 68000 family Motorola has released a number of processors which share a common base architecture but differ in data bus size, address bus size, instruction set, operating system support and performance. The 68000 family of processors are listed in Table 1.11.

Table 1.11: 68000 Family of Processors

Processor	Address bus size	Data bus size
MC68000	24	16
MC6808	20	8
MC68010	24	16
MC68012	31	16
MC68020	32	8/16/32
MC68030	32	8/16/32

The 68000 does not have an on-chip clock circuitry and hence it requires an external clock generator to generate the required clock and supply to the processor. Its maximum internal clock is 25 MHz. The 68000 is available with a maximum clock rating of 6, 8, 10, 12.5, 16.67 and 25 MHz.

The 68000 has a general register based architecture and in this architecture any register can be used as the accumulator or scratch pad register. The internal address and data registers of 68000 are 32-bit wide and its ALU is 16-bit wide. It operates on five different data types: 4-bit BCD, 8-bit, 16-bit and 32-bit binary data. The 68000 has 56 basic instructions and has more than 1000 opcodes. It has 14 addressing modes and supports only memory-mapped IO.

The 68000 has two operating modes: the supervisor mode and the user mode. The supervisor mode is also called the operating system mode, and in this mode the processor can execute all the instructions. Upon hardware reset the 68000 enters the supervisor mode. The processor can switch from the supervisor mode to the user mode by clearing the S-bit in the status register. The processor can switch from the user mode to the supervisor mode by recognition of a trap/reset/interrupt.

1.9.1 Pins and Signals of 68000

The MC68000 microprocessor is a 64-pin IC available in **D**ual **I**n-line **P**ackage (DIP). It is also available as a 68-pin IC in quad or **P**in **G**rid **A**rray (PGA) package or available as a 68-terminal chip carrier. In the 68-pin version the extra four pins are either ground or NC (**N**o **C**onnection).

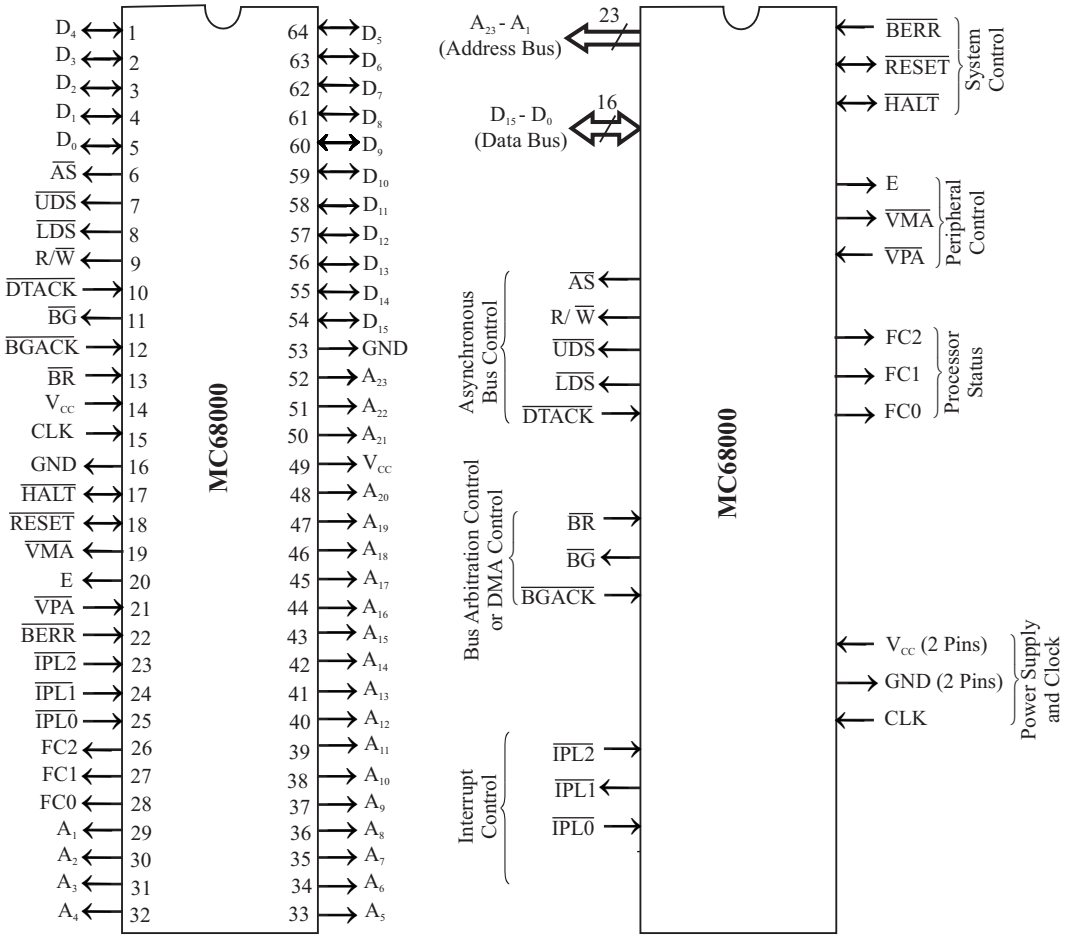


Fig. 1.16: Pin description of MC68000 microprocessor.

The pin configuration of the 64-pin version of MC68000 in DIP is shown in Fig. 1.16. The signals of a 68000 processor are listed in Table 1.12.

The 68000 has 16 data pins D_{15} - D_0 to form a 16-bit data bus and 23 address pins A_{23} - A_1 to address up to 8 MB of physical memory space. The address bit A_0 is internally decoded and supplied as two memory bank select signals $\overline{UDS}$ and $\overline{LDS}$. Hence, 68000-based systems can have two memory banks with a capacity of 8 MB in each bank and so the total addressable memory space is 16 MB. The two memory banks are called odd bank and even bank. The odd bank is enabled by $\overline{LDS}$ and the even bank is enabled by $\overline{UDS}$. The data lines D_{15} to D_8 are connected to the even bank and the data lines D_7-D_0 are connected to the odd bank.

The processor can access bytes from the even bank via D_{15} - D_8 lines by asserting $\overline{UDS}$ as **low** or can access bytes from the odd bank via D_7 - D_0 lines by asserting $\overline{LDS}$ as **low**. For word access both the banks are simultaneously enabled so that one byte from each bank forms a word.

The MC68000 can perform synchronous or asynchronous data transfer with peripherals. The synchronous data transfer can be employed for peripherals having compatible timing with the processor and asynchronous data transfer can be employed for slow peripherals.

The synchronous data transfer between the processor and the peripheral involves the following operations:

1. **The 68000 initiates read/write operation by sending an address and asserting $\overline{AS}$ as low to inform the peripheral that a valid address is on the bus.**
2. **The peripheral will assert $\overline{VPA}$ as low to inform the processor that it requires synchronization with clock.**
3. **The 68000 synchronizes read/write operation with E-clock when it is low.**
4. **Then 68000 asserts $\overline{VMA}$ as low to inform the peripheral that it is synchronized.**
5. **When E-clock is high the peripheral transfer the data.**

The asynchronous data transfer between the processor and the peripheral involves the following operations:

1. **The 68000 initiates read/write operation by sending as address and asserting $\overline{AS}$ as low to inform the peripheral that a valid address is on the bus.**
2. **Then 68000 remains in wait state until it gets a $\overline{DTACK}$ signal from the peripheral.**
3. **When the data is ready, the peripheral will assert $\overline{DTACK}$ as low.**
4. **On receiving the $\overline{DTACK}$ signal the processor will accomplish the read/write operation using R/W, UDS and $\overline{LDS}$.**

The $\overline{BERR}$ is an error signal sent by the external timer/peripheral device to inform the processor that some error has occurred in the machine cycle. (Usually in a 68000-based system an external timer is used to monitor the timing of the machine cycle.) When the 68000 receives an error signal it either returns the instruction cycle which caused the error or executes an error service routine.

Table 1.12: Signals of a 68000 Processor

Signal	Type	Description
A_{23}-A_1	Output	23 pins for address. Used to form 23-bit address bus to address up to 8 MB memory space.
$\overline{UDS}$	Output	Upper Data Strobe. Used to select even memory bank.
$\overline{LDS}$	Output	Lower Data Strobe. Used to select odd memory bank.
D_{15}- D_0	Bidirectional	16 pins for data. Used to form 16-bit data bus.
$\overline{AS}$	Output	Address strobe. It is asserted **low** by the processor to indicate a valid address whenever an address is output on the address bus.
R/$\overline{W}$	Output	Read/Write control signal. For a read operation it is asserted **high** and for write operation it is asserted **low**.
$\overline{DTACK}$	Input	Data acknowledge. Used for asynchronous data transfer between processor and peripheral. It is an acknowledge signal supplied by the peripheral to complete the bus cycle.
E	Output	Enable or E-clock. It is a clock output at one-tenth(1/10) of processor clock.
$\overline{VMA}$	Output	Valid Memory Address. An acknowledge from processor in response to $\overline{VPA}$.
$\overline{VPA}$	Input	Valid Peripheral Address. A request from peripheral to synchronize with clock.
$\overline{BERR}$	Input	Bus error. Input signal from peripheral to indicate bus error.
$\overline{RESET}$	Bidirectional	Reset input/output. As input it is used to bring the processor to known state. As output it is used to reset the peripherals.
$\overline{HALT}$	Bidirectional	Halt control. Used to drive the processor to **high impedance** state during single step or error condition or DMA.
$\overline{BR}$	Input	Bus Request. Used by other bus master to make a request for the bus.
$\overline{BG}$	Output	Bus Grant. It is an acknowledge signal from the processor in response to $\overline{BR}$ signal.
$\overline{BGACK}$	Input	Bus Grant Acknowledge. It is an acknowledgement from the other master controlling the bus.
$\overline{IPL2}$ to $\overline{IPL0}$	Input	Interrupt Privilege Level (or priority level).
FC2-FC0	Output	Function code.
CLK	Input	Clock input.
V_{CC}	Input	Power supply (+5-V).
GND	Output	Power supply ground (0-V).

The $\overline{\text{HALT}}$ input signal is used to drive the processor to **high impedance** state. The $\overline{\text{HALT}}$ is asserted **low** in order to perform single stepping or to indicate double error. It can also be used by other bus masters (such as the DMA controller) to take control of the system bus. The $\overline{\text{HALT}}$ signal can also be used as the output signal. During a major failure the processor asserts the $\overline{\text{HALT}}$ signal as **low** to inform the peripheral and goes to **high impedance** state.

In order to reset the system both $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ pins must be asserted **low** at the same time. The 68000 $\overline{\text{RESET}}$ pin can also be used as an output signal. In the supervisor mode, when the RESET instruction is executed, the processor outputs **low** signal on the $\overline{\text{RESET}}$ line in order to reset the peripherals connected to this pin. During a hardware reset through $\overline{\text{RESET}}$ pin the **P**rogram **C**ounter (PC), **S**tack **P**ointer (SP) and **S**tatus **R**egister (SR) are initialized, and other registers are not altered.

The input signals $\overline{\text{IPL2}}$, $\overline{\text{IPL1}}$ and $\overline{\text{IPL0}}$ are used to initiate seven hardware interrupts. The input signal 000_2 to 110_2 will initiate interrupt level-7 to level-1. (The complement of the input gives the interrupt level.) The interrupt level-7 has the highest priority and level-1 has the lowest priority. Level-7 is a nonmaskable interrupt while the other interrupts are maskable.

The signals $\overline{\text{BR}}$, $\overline{\text{BG}}$ and $\overline{\text{BGACK}}$ are used as bus arbitration signals in multi-master systems. In a 68000 processor-based multimaster system, the 68000 processor will be the main/primary master and the other masters such as DMA controller will be the secondary master. The $\overline{\text{BR}}$ signal is used by the secondary master to make a request for the control of the system bus.

On receiving the $\overline{\text{BR}}$ signal the processor will send an acknowledge via the $\overline{\text{BG}}$ line to the secondary master which made a request for the bus. The secondary master has to check for completion of the current bus cycle and then assert $\overline{\text{BGACK}}$ signal as **low** to drive the processor to **high impedance** state. The processor will come to a normal state only when $\overline{\text{BGACK}}$ is asserted **high** by the secondary master.

The 68000 processor outputs the bus status signal on pins FC2 to FC0 to indicate whether the processor is fetching data/ program from user/supervisor memory or the processor is servicing an interrupt.

The status signals of a 68000 processor are listed in Table 1.13. In a 68000 processor the status signals are also known as function codes.

Table 1.13: Status Signalsof the MC68000

FC2	FC1	FC0	Operation performed by bus
0	0	1	User data access
0	1	0	User program access
1	0	1	Supervisor data access
1	1	0	Supervisor program access
1	1	1	Interrupt acknowledge

Note: *Other codes are not defined.*

1.9.2 Architecture of a MC68000 Microprocessor

The architecture of a MC68000 microprocessor is shown in Fig. 1.17. It has a general register-based architecture in which any data register can be used as the accumulator or scratch-pad register. The 68000 processor has seventeen numbers of 32-bit registers in which eight registers are called data registers and the remaining nine registers are called address registers. The data registers are denoted as D0 - D7 and the address registers are denoted as A0 - A7, A7′.

The data registers are used to store 8/16/32 bit data. The address registers are used to store 24-bit address. (The upper 8-bits of address registers are ignored in 68000 processor.) The address register A7 is the **S**tack **P**ointer (SP) or **U**ser **S**tack **P**rinter (USP) and the address register A′ is the **S**upervisor **S**tack **P**ointer (SSP).

The 68000 processor has a 16-bit ALU (**A**rithmetic **L**ogic **U**nit) which can operate on 8, 16 or 32-bit data. The architecture includes a 32-bit temporary register which is used to hold the operand or intermediate result during the execution of an instruction.

The 68000 processor has a 16-bit **I**nstruction **R**egister (IR) which holds the first word of the currently executing instruction. The control unit decodes the instructions and controls the other blocks to fetch and execute the instructions.

The architecture includes a 32-bit **P**rogram **C**ounter (PC) in which the lower 24-bit portion is used to hold the 24-bit address of instruction being executed. When an instruction word is fetched the PC will hold the address of next word of same instruction or the address of next instruction.

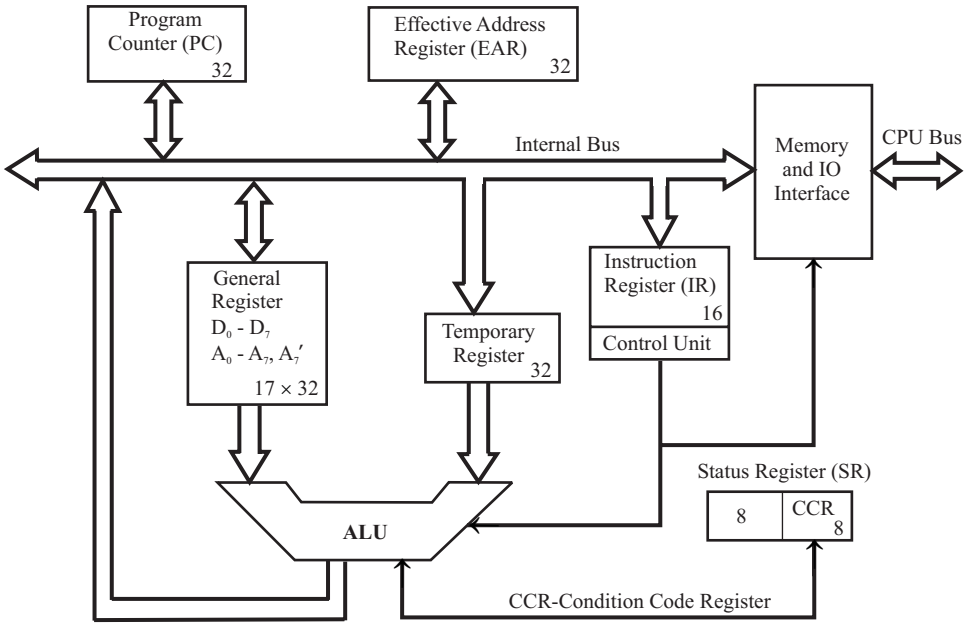


Fig. 1.17: Architecture of MC68000 microprocessor.

The memory and IO interface takes care of the read/write operation with memory or IO device as commanded by the control unit. The 68000 processor has a 16-bit status register in which the lower 8-bit portion is also known as **C**ondition **C**ode **R**egister (CCR).

The format of the status register of a 68000 microprocessor is shown in Fig. 1.18. The content of the status register is also known as status word in which the lower byte is called user byte and upper byte is called system byte.

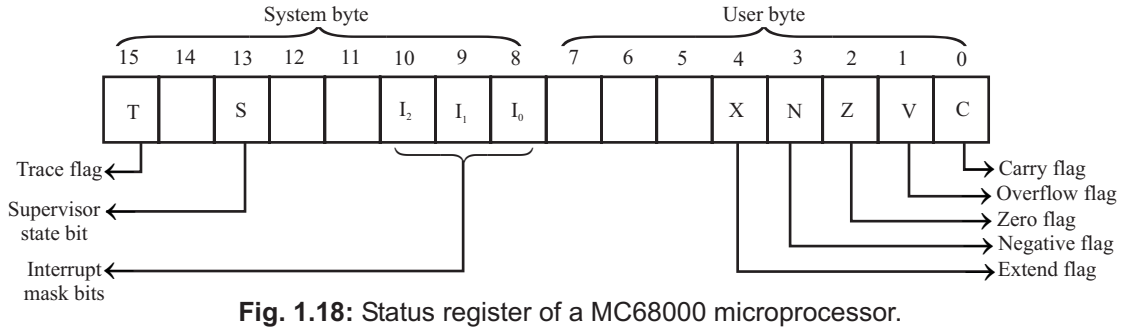


Fig. 1.18: Status register of a MC68000 microprocessor.

The user byte consists of arithmetic flags like carry, overflow, negative, zero and extend flags. During arithmetic operations the carry and extend flag are affected in an identical manner. The instruction set of a 68000 processor includes instructions which uses extend flag for addition with carry or subtraction with borrow.

The system byte consists of trace flag, supervisor state bit and interrupt mask bits. When the trace flag is set to one, the 68000 processor generates an internal interrupt called trap after execution of each instruction and so the trace flag can be used for single step execution of programs for debugging.

The supervisor state bit is used to switch the operating mode from the supervisor mode to the user mode or vice versa. When S-bit is set to one, the processor operates in the supervisor mode and when S-bit is cleared to zero, the processor operates in the user mode. The interrupt mask bits provide the status of 68000 interrupt input pins IPL2, IPL1 and IPL0. The signals on these pins are inverted and then stored as I_2, I_1 and I_0 respectively.

1.9.3 Programming Model of a 68000 Microprocessor

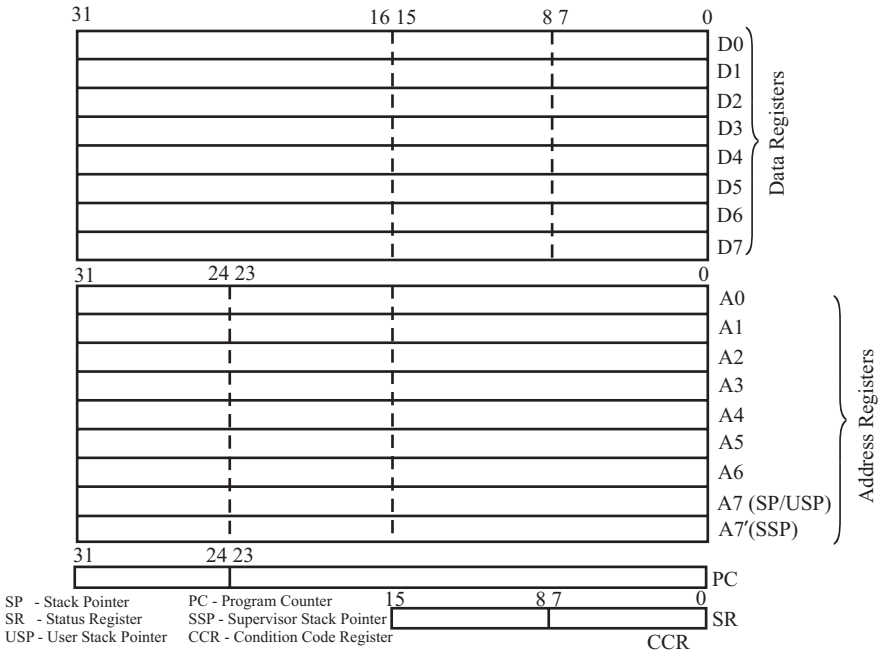


Fig. 1.19: Programming model of a MC68000 microprocessor.

The set of registers whose contents are altered while executing instructions constitute the programming model of a processor. In a 68000 processor the contents of the program counter, data registers, (D0 - D7), address registers (A0 - A7, A7′) and status register are altered during execution of instructions and so these set of registers constitute the programming model of a 68000 processor. The programming model of a 68000 processor is shown in Fig. 1.19.

1.10 SHORT-ANSWER QUESTIONS

Q1.1 What is a microprocessor ?

A microprocessor is a program controlled semiconductor device (IC), which fetches, decodes and executes instructions.

Q1.2 What are the basic functional blocks of a microprocessor ?

The basic functional blocks of a microprocessor are ALU, an array of registers and control unit.

Q1.3 What is a bus ?

Bus is a group of conducting lines that carries data, addresses and control signals.

Q1.4 Define bit, byte and word.

A digit of the binary number or code is called bit. The bit is also the fundamental storage unit of computer memory.

The 8-bit (8-digit) binary number or code is called byte and 16-bit binary number or code is called word. (Some microprocessor manufacturers refer to the basic data size operated by the processor as word.)

Q1.5 State the relation between the number of address pins and physical memory space?

The size of the binary number used to address the memory decides the physical memory space. If a microprocessor has n-address pins then it can directly address 2^n memory locations. (The memory locations that are directly addressed by the processor are called physical memory space.)

Q1.6 Why is data bus is bidirectional?

The microprocessor has to fetch (read) the data from memory or input device for processing and after processing it has to store (write) the data in memory or output device. Hence, the data bus is bidirectional.

Q1.7 Why is address bus unidirectional?

The address is an identification number used by the microprocessor to identify or access a memory location or IO device. It is an output signal from the processor. Hence, the address bus is unidirectional.

Q1.8 State the difference between CPU and ALU.

The ALU is the unit that performs the arithmetic or logical operations. The CPU is the unit that includes ALU and control unit. Apart from processing the data, the CPU controls the entire system functioning. Usually, a microprocessor will be the CPU of a system and it is called the brain of the computer.

Q1.9 What is a tristate logic? Why it is needed in microprocessor system?

In a tristate logic, three logic levels are used **high, low** and **high impedance** state. The **high** and **low** are normal logic levels and **high impedance** state is electrical open circuit condition.

In a microprocessor system, all the peripheral/slave devices are connected to a common bus. But communication (data transfer) takes place between the master (microprocessor) and one slave (peripheral) at any time instant. During this time instant, all other devices should be isolated from the bus. Therefore, normally all the slaves (peripherals) will remain in **high impedance** state (i.e., in electrical isolation). The master will select a slave by sending address and chip select signal. When the slave is selected, it comes to normal logic and it can communicate with the master.

Q1.10 What is HMOS and HCMOS.

The HMOS is High density n-type Metal Oxide Silicon field effect transistors. The third generation microprocessors are fabricated using HMOS transistors.

The HCMOS is High density n-type Complementary Metal Oxide Silicon field effect transistors. It is the low power version of HMOS and the fourth generation microprocessors are fabricated using HCMOS transistors.

Q1.11 What are the drawbacks of first generation microprocessors.

The first generation processors are fabricated using PMOS technology and it has the drawbacks like slow speed, provides low output currents and was not compatible with TTL logic levels.

Q1.12 What is a microcomputer? Explain the difference between a microprocessor and a microcomputer.

A system designed using a microprocessor as its CPU is called microcomputer. The term microcomputer refers to the whole system, whereas the microprocessor is the CPU of the system.

Q1.13 What is the function of microprocessor in a system?

The microprocessor is the master in the system, which controls all the activity of the system. It issues address and control signals and fetches the instruction and data from memory. Then it executes the instruction to take appropriate action.

Q1.14 List the components of microprocessor-based (single board microcomputer) system.

The microprocessor-based system consist of microprocessor as CPU, semiconductor memories like EPROM and RAM, input device, output device and interfacing devices.

Q1.15 Why interfacing is needed for IO devices?

Generally IO devices are slow devices. Therefore, the speed of IO devices does not match with the speed of microprocessor. And so an interface is provided between system bus and IO devices.

Q1.16 What is the difference between CPU bus and system bus?

The CPU bus has multiplexed lines but the system bus has separate lines for each signal. (The multiplexed CPU lines are demultiplexed by the CPU interface circuit to form system bus.)

Q1.17 What is multiplexing and what is its advantage?

Multiplexing is transferring different information at different well-defined times through same lines. A group of such lines is called multiplexed bus. The advantage of multiplexing is that fewer pins are required for microprocessors to communicate with the outside world.

Q1.18 How the address and data lines are demultiplexed in 8085?

The low order address and data lines of 8085 are demultiplexed using an external 8-bit D-Latch (74LS373) and the ALE signal of 8085, as shown in Fig. Q1.18.

At the beginning of every machine cycle, ALE is asserted **high** and then **low**. Also, the low byte of address is given out through AD_0 - AD_7 lines. Since, the ALE is connected to enable of latch, whenever ALE is asserted **high** and then **low,** the addresses are latched into the output lines of the latch then the lines AD_0 - AD_7 are free for data transfer.

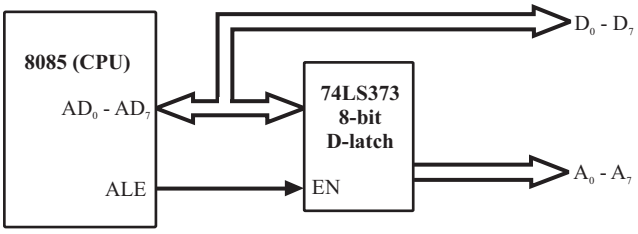


Fig. Q1.18: Demultiplexing of address and data lines in 8085.

Q1.19 What do you mean by 16 and 8-bit processors? Mention a few 8-bit and 16-bit processors.

The processors are classified into 8-bit or 16-bit depending on the basic data size handled by the ALU of the processor.

8-bit microprocessors : 8085, Z80, Motorola 6800.
16-bit microprocessors : 8086, Z8000, MC68000.

Q1.20 What is the fabrication technology used for 8085?

The 8085A is fabricated used NMOS technology and 8085AH is fabricated using HMOS technology.

Q1.21 What is the physical memory space in 8085?

The 8085 uses 16-bit address to access memory locations. Hence, it can directly address 64 k memory locations (2^{16} = 65,536 = 64 k). Since 8085 has 8 data lines, it can read or write 8-data bits from a memory address. Therefore, the physical memory space is 64 k × 1byte = 64 kilobytes (64 kB).

Q1.22 What is ALE?

The ALE (**A**ddress **L**atch **E**nable) is a signal used to demultiplex the address and data lines using an external latch. It is used as enable signal for the external latch.

Q1.23 Explain the function of IO/$\overline{M}$ in 8085.

The IO/$\overline{M}$ is used to differentiate memory access and IO access. For IN and OUT instruction it is asserted **high**. For memory reference instructions it is asserted **low**.

Q1.24 How the READY signal is used in microprocessor system?

The READY is an input signal that can be used by slow peripherals to get extra time in order to communicate with 8085. The 8085 will work only when READY is tied to logic **high**. Whenever READY is tied to logic **low**, the 8085 will enter a wait state. When the system has slow peripheral devices, additional hardware is provided in the system to make the READY input **low** during the required extra time while executing a machine cycle, so that the processor will remain in wait state during this extra time.

Q1.25 What is HOLD and HLDA? How is it used?

The HOLD and HLDA signals are used for the **D**irect **M**emory **A**ccess (DMA) type of data transfer. These type of data transfers are achieved by employing a DMA controller in the system. When DMA is required, the DMA controller will place a **high** signal on the HOLD pin of 8085. When HOLD input is asserted **high,** the processor will enter a wait state and drive all its tristate pins to **high impedance** state and send an acknowledge signal to DMA controller through HLDA pin. Upon receiving the acknowledge signal, the DMA controller will take control of the bus and perform DMA transfer and at the end it asserts HOLD signal **low**. When HOLD is asserted **low,** the processor will resume its execution.

Q1.26 How clock signals are generated in 8085 and what is the frequency of the internal clock?

The 8085 has the clock generation circuit on the chip but an external quartz crystal or LC circuit or RC circuit should be connected at the pins X_1 and X_2 in order to generate a clock signal. The 8085 clock generation circuit, generate a clock whose frequency is double as compared to that of internal clock. The generated clock is divided by two and then used as internal clock. The maximum internal clock frequency of 8085A is 3.03 MHz.

Q1.27 What happens to the 8085 processor when it is resetted?

When $\overline{\text{RESET IN}}$ pin is asserted **low**, the program counter, instruction register, interrupt mask bits and all internal registers are cleared or resetted. Also the RESET OUT signal is asserted **high** to clear or reset all the peripheral devices in the system. After a reset, the content of program counter will be 0000_H and so the processor will start executing the program stored at 0000_H.

Q1.28 What are the operations performed by ALU of 8085?

The operations performed by ALU of 8085 are addition, subtraction, logical AND, OR, Exclusive-OR, compare, complement, increment, decrement and left/right shift.

Q1.29 Mention the names of various registers in 8085 along with its size.

Register	Size (bits)	Register	Size (bits)
Accumulator (A)	8	Stack pointer	16
Temporary register	8	Program counter	16
Instruction register	8		
SGeneral purpose register (B, C, D, E, H and L)	8		

Q1.30 What is a flag?

Flag is a flip- flop used to store the information about the status of the processor and the status of the instruction executed most recently.

Q1.31 List the flags of 8085.

There are five flags in 8085. They are sign flag, zero flag, auxiliary carry flag, parity flag and carry flag.

Q1.32 What are the hardware interrupts of 8085?

The hardware interrupts in 8085 are TRAP, RST 7.5, RST 6.5 and RST 5.5.

Q1.33 Which interrupt has highest priority in 8085? What is the priority of other interrupts?

The TRAP has the highest priority, followed by RST 7.5, RST 6.5, RST 5.5 and INTR.

Q1.34 Show the bit positions of various flags in 8085 flag register.

The bit positions of various flags in the flag register of 8085 is shown in Fig. Q1.34.

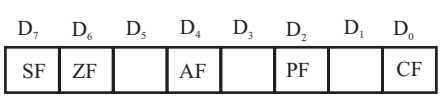


Fig. Q1.34: Bit positions of various flags in the flag register of 8085.

Q1.35 Define stack.

Stack is a sequence of RAM memory locations defined by the programmer.

Q1.36 What is program counter? How is it useful in program execution?

The program counter keeps a track of program execution. To execute a program, the starting address of the program is loaded in program counter. The PC sends out an address to fetch a byte of instruction from memory and increment its content automatically.

Q1.37 How is the microprocessor synchronized with peripherals ?

The timing and control unit synchronizes all the microprocessor operations with clock and generates control signals necessary for communication between the microprocessor and peripherals.

Q1.38 What are the additional features in Z80, as compared to an 8085?

The Z80 has separate pins for data and address. The Z80 provides more register, extra addressing modes, a larger instruction set than 8085 and it has built-in logic to refresh dynamic RAM memories. The Z80 has an indexed addressing mode.

Q1.39 What are shadow registers of Z80?

Each register of Z80 has an alternate register. The set of alternate registers are called shadow registers.

Q1.40 How are the control signals classified in Z80?

The control signals of Z80 are classified into bus control, CPU control and system control signals.

Q1.41 List the register pairs of Z80.

The registers pairs of Z80 are BC, DE, HL, B'C', D'E' and H'L'.

Q1.42 List the flags of Z80.

The Z80 has six flags:

1. **Sign flag (S and S')**
2. **Zero flag (Z and Z')**
3. **Carry flag (C and C')**
4. **Parity/Overflow flag (P/V and P'/V')**
5. **Half carry flag (H and H')**
6. **Subtract flag (N and N')**

Q1.43 What are the common features of 8085 and Z80?

The common features of 8085 and Z80 are the following:

1. **Both are fabricated using NMOS technology and have 40 pins.**
2. **Memory is accessed by a 16-bit address and the IO device by an 8-bit address.**
3. **The 8085 is software compatible with Z80.**

Q1.44 List the difference between 8085 and Z80.

8085	Z80
1. Low order address and data lines are multiplexed.	1. Separate lines are provided for address and data.
2. A single signal is IO/$\overline{M}$ used to differentiate IO access and memory access.	2. Separate signals are used to differentiate the memory address and the IO address.
3. The instruction size is one to three bytes.	3. The instruction size is one to four bytes.
4. The flag register has five flags	4. The flag register has six flags.
5. It has five hardware interrupts.	5. The Z80 has two hardware interrupts.
6. It has 74 types of instructions.	6. It has 156 types of instructions.

Q1.45 What is the data and address size in Motorola 6800?

In Motorola 6800 the data size is 8-bit and address size is 16-bit.

Q1.46 How are IO devices addressed in M6800?

The Motorola 6800 does not have a separate address for its memory and IO devices. Hence some of the memory addresses are used to address IO devices.

Q1.47 How are the control signals of M6800 classified?

The control signals of M6800 are classified into control bus signals and CPU (microprocessor) supervisory signals.

Q1.48 What is the clock requirement of M6800?

The Motorola 6800 requires an external 2-phase clock whose maximum frequency can be 1 MHz.

Q1.49 What is CCR or what is the name of the flag register in M6800?

The flag register in the Motorola processor is called **C**ondition **C**ode **R**egister (CCR).

Q1.50 What are the flags of M6800?

The M6800 has six flags:

1. Negative (N)
2. Zero (Z)
3. Overflow (V)
4. Carry (C)
5. Half Carry (H)
6. Interrupt enable/disable (I).

Q1.51 What are the addressing modes available in Motorola 6800?

The addressing modes of Motorola 6800 are immediate, direct, indexed and relative addressing.

Q1.52 List the differences between 8085 and M6800.

8085	M6800
1. Low order address and data lines are multiplexed.	1. Separate lines are provided for address and data.
2. It has a 16-bit address for memory and an 8-bit address for IO-mapped devices.	2. It does not have a separate address for memory and IO-mapped devices.
3. The flag register has five flags.	3. The flag register has six flags.
4. It has five hardware interrupts.	4. It has two hardware interrupts.

Q1.53 What are the different types of instructions available in Motorola 6800?

Data handling, arithmetic, logic, control transfer, data test, condition codes, address maintenance and interrupt handling are the different types of instructions available in Motorola 6800.

Q1.54 What are the modes in which 8086 can operate?

The 8086 can operate in two modes and they are minimum (or uniprocessor) mode and maximum (or multiprocessor) mode.

Q1.55 What is the data and address size in 8086?

The 8086 can operate on either 8-bit or 16-bit data. The 8086 uses 20-bit address to access memory and 16-bit address to access IO devices.

Q1.56 What is the difference between 8086 and 8088?

The external data bus in 8086 is 16-bit and that of 8088 is 8-bit, i.e., the 8086 access memory in words but 8088 access memory in bytes.

Q1.57 Explain the function of $M/\overline{IO}$ in 8086.

The signal $M/\overline{IO}$ is used to differentiate memory address and IO address. When the processor is accessing memory locations $M/\overline{IO}$ is asserted **high** and when it is accessing IO-mapped devices it is asserted **low**.

Q1.58 What are the hardware interrupts of 8086?

The hardware interrupts of 8086 are INTR and NMI. The INTR is general maskable interrupt and NMI is non-maskable interrupt.

Q1.59 How is clock signal generated in 8086? What is the maximum internal clock frequency of 8086?

The 8086 does not have on-chip clock generation circuit. Hence the clock generator chip, 8284 is used to generate the required clock. The frequency of clock generated by 8284 is thrice that of internal clock frequency of 8086. The 8284 divides the generated clock by three and modifies the duty cycle to 33% and then supply as clock signal to 8086. The maximum internal clock frequency of 8086 is 5 MHz.

Q1.60 What is pipelined architecture?

In pipelined architecture, the processor will have the number of functional units and the execution time of functional units overlapped. Each functional unit works independently most of the time.

Q1.61 What are the functional units available in 8086 architecture?

The **B**us **I**nterface **U**nit (BIU) and **E**xecution **U**nit (EU) are the two functional units available in 8086 architecture.

Q1.62 List the segment registers of 8086.

The segment registers of 8086 are **C**ode **S**egment (CS), **D**ata **S**egment (DS), **S**tack **S**egment (SS) and **E**xtra **S**egment (ES) registers.

Q1.63 What is the difference between segment register and general purpose register?

The segment registers are used to store 16-bit segment base address of the four memory segments. The general purpose registers are used as the source or destination register during data transfer and computation, as pointers to memory and as counters.

Q1.64 What is queue? How is queue implemented in 8086?

A data structure which can be accessed on the basis of first in first out is called queue. The 8086 has six numbers of 8-bit FIFO registers, which are used as instruction queue.

Q1.65 Write the flags of 8086.

The 8086 has nine flags. They are:

1. **Carry Flag (CF)**
2. **Parity Flag (PF)**
3. **Auxiliary carry Flag (AF)**
4. **Zero Flag (ZF)**
5. **Sign Flag (SF)**
6. **Overflow Flag (OF)**
7. **Trace Flag (TF) (or Single step trap)**
8. **Interrupt Flag (IF)**
9. **Direction Flag (DF)**

Q1.66 Write the special functions carried out by the general purpose registers of 8086.

The special functions carried out by the registers of 8086 are the following:

Register	Name of the register	Special function
AX	16-bit Accumulator	Stores the 16-bit result of certain arithmetic and logical operations.
AL	8-bit Accumulator	Stores the 8-bit result of certain arithmetic and logical operations.
BX	Base Register	Used to hold the base value in base addressing mode to access memory data
CX	Count Register	Used to hold the count value in SHIFT, ROTATE and LOOP instructions.
DX	Data Register	Used to hold data for multiplication and division operations.
SP	Stack Pointer	Used to hold the offset address of top of stack memory.
BP	Base Pointer	Used to hold the base value in base addressing using stack segment register to access data from stack memory.
SI	Source Index	Used to hold the index value of source operand (data) for string instructions.
DI	Destination Index	Used to hold the index value of destination operand (data) for string instruction.

Q1.67 What are control bits?

The flags TF, IF and DF of 8086 are used to control the processor operation and hence are called control bits.

Q1.68 List the internal registers of a 68000 processor.

The 68000 processor has eight numbers of 32-bit data registers D_0-D_7 and nine numbers of 32-bit address registers A_0-A_7, A'_7. Also it has a 32-bit **P**rogram **C**ounter (PC) and a 16-bit **S**tatus **R**egister (SR). For addressing, the 68000 processor uses only lower 24-bit position of address registers and PC.

Q1.69 What is status word in a 68000 processor?

In a 68000 processor, the content of the status register is called the status word, in which the lower byte is called the user byte and the upper byte is called the system byte.

The user byte consists of arithmetic flags like carry, overflow, negative, zero and extend flags. The system byte consists of the trace flag, the supervisor state bit and the interrupt mask bits.

Q1.70 List few differences between 8086 and 68000 processor?

8086	68000
1. The address pins are multiplexed with data and status signals.	1. Separate (or Non-multiplexed) pins are provided for address and data.
2. Uses a 20-bit address for memory and so physical address space is 1MB.	2. Uses a 24-bit address for memory and so the physical address space is 16 MB.
3. Separate IO address space is available. Therefore both memory mapping and IO mapping of IO devices are possible.	3. The Motorola processor does not have a separate IO address space and so only memory-mapped IO is possible
4. All the internal registers are 16 bits wide.	4. All the internal registers (except SR) are 32 bits wide.

1.11 EXERCISES

I. Fill in the blanks with appropriate words

1. A digit of the binary number or code is callled _____ .
2. The group of conducting lines that carry control signals is called _____ bus.
3. The _____ state is used to keep the device electrically isolated from the system.
4. INTEL 8088 is an _____ bit processor.
5. The third generation microprocessors were designed using _____ technology.
6. Transfering different information at different well defined times through the same lines is called _____.
7. 1 mil is equivalent to _____ inch.
8. The _____ and _____ signals of 8085 are used for serial data communication between 8085 and any serial device.
9. The _____ register of 8085 points to the next instruction to be executed.
10. The two independent 8-bit accumulators of Z80 processor are _____ and _____.
11. The two registers of Z80 processor used for indexed addressing mode are _____ and _____.
12. _____ register is used to indicate the results of an ALU operation in motorola 6800.
13. _____ signal is used to differentiate the minimum mode and maximum mode operation in 8086 processor.
14. _____ flag is set to 1 if the most significant bit of the result is one.
15. _____ register is used to hold the upper 16 bits of the starting address of the code segment in 8086 processor.
16. _____ pins are used to track the internal status of the instruction queue in 8086.
17. _____ register is used as the counter register in 8086.
18. _____ flag is used for single step execution in 8086.
19. The address line _____ is used to select the even bank of 8086.
20. _____ pin of 8086 is made low by external hardware when the instruction WAIT is executed.
21. The status signals _____ and _____ are use to indicate the selection of segments in 8086.
22. The maximum internal clock frequency of motorola MC68000 is _____.
23. The _____ bit in the status register of MC68000 is reset to zero to switch the processor from supervisor mode to user mode.
24. The _____ pin is asserted low by MC68000 to indicate a valid address whenever an address is output on the address bus.

Answers

1. bit	7. 10^{-3}	13. MN/$\overline{MX}$	19. A_0
2. control	8. SID, SOD	14. Sign	20. TEST
3. high impedance	9. Program Counter(PC)	15. CS	21. S_3,S_4
4. eight	10. A,A'	16. QS_1 and QS_0	22. 25 MHz
5. High Density MOS(HMOS)	11. IX,IY	17. CX	23. S
6. multiplexing	12. Condition Code	18. TF	24. $\overline{AS}$

II. State whether the following statements are True/False.

1. The address bus is bidirectional.
2. The CPU bus is directly connected to the microprocessor.
3. The high impedance state is an electrical open-circuit condition .
4. The NMOS technology offers faster speed and higher density than HMOS technology.
5. Registers can be read/written faster than memory chips.
6. The 8085 has an internal clock oscillator.
7. The 8085 interrupts RST 7.5, RST 6.5 and RST 5.5 are non-maskable interrupts.
8. To reset the 8085 microprocessor the $\overline{\text{RESET IN}}$ pin should be held low for atleast three clock pulses.
9. The stack pointer (SP) holds the address of the stack top.
10. Z80 processor does not require a separate circuit to refresh dynamic RAM.
11. Z80 is software compatible with 8085.
12. The physical memory size of Z80 is larger than that of 8085.
13. The motorola 6800 uses I/0 mapping technique to interface the peripherals.
14. The 8086 processor does not have an internal clock circuit
15. The 8086 uses 16-bit address to access memory.
16. The 8088 uses 8-bit data bus externally.
17. The 8086 processor executes single bus cycle to read/write a word from/to the memory address 50013_H.
18. The 8086 and 8088 have a common instruction set.
19. The minimum mode signals of 8086 are used in multiprocessor environment.
20. The memory segments in 8086 cannot be overlapped.
21. SP register of 8086 always points to the top of the stack.
22. Data segment is the default segment in 8086.
23. The motorola MC68000 can address up to 16 Mb of physical memory space.
24. The Motorola MC68000 enters the supervisor mode when it is reset.
25. MC68000 consists of only one accumulator.

Answers

1. False	6. True	11. True	16. True	21. True
2. True	7. False	12. False	17. False	22. True
3. True	8. True	13. False	18. True	23. True
4. False	9. True	14. True	19. False	24. True
5. True	10. True	15. False	20. False	25. False

III. Choose the right answer for the following questions.

1. *Microprocessors are intended to be a _____ computer.*

   a) general-purpose *b)* special purpose *c)* hybrid *d)* analog

2. *Group of 4-bits is called*

   a) byte *b)* nibble *c)* word *d)* double word

3. *What would be the total memory capacity of a microprocessor with 10 address lines*

   a) 1 MB *b)* 1 GB *c)* 1 KB *d)* 512 MB

4. *The first 8-bit processor introduced by INTEL is*

   a) 8080 *b)* 8008 *c)* 8085 *d)* 8086

5. *Which of the following is used to store temporary programs and data?*

   a) EPROM *b)* ROM **c)** RAM *d)* all the three

6. *The 1-bit register provided to store the results of certain program instructions is*

   a) status register *b)* instruction register *c)* program counter *d)* flag

7. *Which of the following is not a 16-bit processor?*

   a) 8086 *b)* 80186 *c)* 8088 *d)* 8096

8. *Which of the following signals is used to demultiplex the address/data lines in 8085 and 8086 processors?*

   a) DT/$\overline{R}$ *b)* ALE *c)* SOD *d)* READY

9. *The maximum internal clock frequency of 8085A microprocessor is*

   a) 5.03 MHz *b)* 6 MHz *c)* 10.6 MHz *d)* 3.03 MHz

10. *The total memory capacity of 8085 processor is*

    a) 64 KB *b)* 1 MB *c)* 64 MB *d)* 10 MB

11. *Which of the following 8085 signals are used for DMA operation?*

    a) $\overline{RD}$, $\overline{WR}$ *b)* $\overline{RESET\ IN}$, $\overline{RESET\ OUT}$ *c)* HOLD, HLDA *d)* SOD, SID

12. *When initiated, certain interrupts can be delayed or rejected but when allowed, the program execution starts from a fixed location. Such an interrupt is known as,*

    a) non maskable and non vectored *b)* maskable and non vectored
    c) non maskable and vectored *d)* maskable and vectored

13. *The vector address for the 8085 interrupt TRAP is,*

    a) 0028_H *b)* 0020_H *c)* 0024_H *d)* 0000_H

14. *Which of the following flag bit is available in Z80 processor but not in 8085 processor*

    a) sign flag *b)* zero flag *c)* subtract flag *d)* carry flag

15. *Which of the following is the first 8-bit single chip microprocessor to exploit a single 5-V power supply?*

    a) 8080 *b)* 8085 *c)* motorola 6800 *d)* Z80

16. What is the total memory capacity of 8086 microprocessor?

a) 1 GB ***b)*** 1 MB ***c)*** 2 MB ***d)*** 64 KB

17. Which of the following is the maximum mode signal of 8086?

a) HOLD ***b)*** ALE ***c)*** DT/$\overline{R}$ ***d)*** $\overline{LOCK}$

18. Which of the following signal is used to select the odd memory bank of 8086?

a) READY ***b)*** A_0 ***c)*** $\overline{BHE}$ ***d)*** ALE

19. What is the size of the instruction queue in 8086?

a) 4 bytes ***b)*** 6 bytes ***c)*** 8 bytes ***d)*** 5 bytes

20. What will be the content of CS register if the 8086 is reset?

a) $FFF0_H$ ***b)*** $FF00_H$ ***c)*** $FFFF_H$ ***d)*** 0000_H

21. Which of the following 8086 registers is used to hold the index value of destination operand for string instructions?

a) SP ***b)*** BP ***c)*** SI ***d)*** DI

22. Which of the following signals is used by slow peripherals to get extra time in order to communicate with 8086?

a) TEST ***b)*** READY ***c)*** DEN ***d)*** none of the above

23. What is the significance of instruction queue in 8086?

a) overlapping ***b)*** multitasking ***c)*** pipeling ***d)*** multiprogramming

24. Which of the following signals of MC68000 are used to enable the odd bank and even bank respectively?

a) $\overline{UDS}$, $\overline{LDS}$ ***b)*** $\overline{LDS}$, $\overline{UDS}$ ***c)*** A_0, $\overline{BHE}$ ***d)*** $\overline{BHE}$, A_0

25. Which of the following signals of MC68000 is used to indicate the bus error from peripheral to the processor?

a) $\overline{BERR}$ ***b)*** $\overline{BR}$ ***c)*** BG ***d)*** $\overline{DTACK}$

Answers

1. a	5. c	9. d	13. c	17. d	21. d	25. a
2. b	6. d	10. a	14. c	18. c	22. b	
3. c	7. d	11. c	15. c	19. b	23. c	
4. b	8. b	12. d	16. b	20. c	24. b	

IV. Answer the following questions.

E1.1 What is meant by addressability of a microprocessor?

E1.2 State the significance of clock pulse in microprocessor based system.

E1.3 List some of the 4-bit microprocessors with specifications.

E1.4 Define the term speed power product(SPP).

E1.5 Mention the packing density of NMOS and HMOS technology

E1.6 Define the term MIPS.

E1.7 Draw the basic functional blocks of a microprocessor.

E1.8 What is meant by active low signal? Explain with an example.

E1.9 Write down the significance of bus status signals IO/$\overline{M}$, S_0 and S_1.

E1.10 What is the use of signals SOD and SID in 8085?

E1.11 Define maskable and non-maskable interrupts.

E1.12 What is meant by hardware interrupt?

E1.13 What is meant by software interrupt?

E1.14 What is meant by vectored and non-vectored interrupts?

E1.15 Write down the vector address of all the interrupts of 8085.

E1.16 When power on, how does the CPU know the starting address of the first instruction it has to execute? What is that first instruction? why?

E1.17 What is the use of stack pointer in 8085 microprocessor?

E1.18 Why the program counter and stack pointer of 8085 are 16-bit registers?

E1.19 How is the dynamic RAM refreshed in Z80?

E1.20 What is the use of shadow registers in Z80?

E1.21 How is the interrupt service routine serviced in Z80?

E1.22 Mention the clock frequency of different versions of 8086 processor.

E1.23 How are the even and odd bank of memory accessed in 8086?

E1.24 What is meant by memory segmentation? What are its advantages?

E1.25 Name the different segments of 8086 and mention the segment size.

E1.26 How is the physical address generated in 8086?

E1.27 Write down the status of all the six status flag after execution of an 8086 instruction that adds AL = 28_H and BL = $1D_H$.

E1.28 How many bus cycles are required for accessing a word which is stored at memory location 20013_H? Which (higher or lower) data lines are used for placing lower byte and higher byte?

E1.29 Is it possible for a segment to begin at a memory address that is not divisible by 16 in 8086? Why?

E1.30 An 8-bit data 50_H is stored at the address 75380_H in a data segment with base address 7500_H. What will be the physical address if the same data is stored in another data segment with same offset address but the base address is changed to 6000_H?

E1.31 Which pin of 8086 is used to synchronize the slow peripherals with 8086? How?

E1.32 Show the bit positions of various flags in an 8086 flag registers.

E1.33 Differentiate the 8086 signals READY and $\overline{TEST}$.

E1.34 What is the significance of WAIT instruction in 8086?

E1.35 How does the 8086 processor read the status of instruction queue?

E1.36 What is meant by supervisor mode? Also state its importance in MC68000.

E1.37 What is meant by scratch pad register?

E1.38 How is the motorola MC68000 processor reset?

E1.39 Draw the programming model of MC68000 processor.

E1.40 Differentiate conditional and control flags.