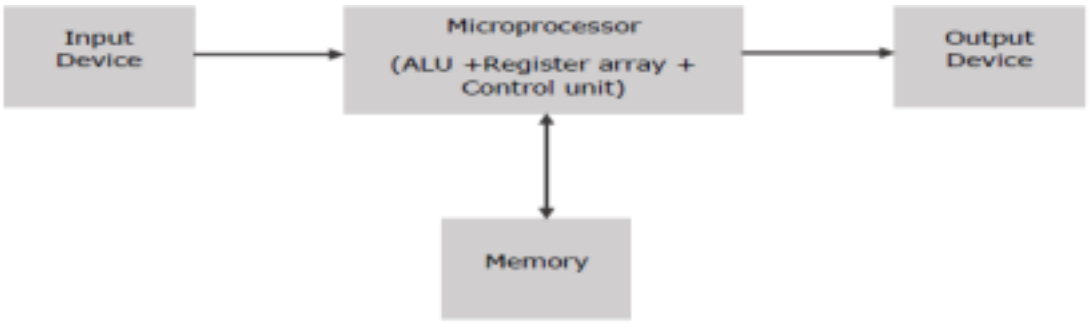
Module 4 CO

Role of Microprocessor in Micro Computer



- Microprocessor is a controlling unit of a micro-computer, fabricated on a small chip capable of performing ALU (Arithmetic Logical Unit) operations and communicating with the other devices connected to it.
- Microprocessor consists of an ALU, register array, and a control unit.
- ALU performs arithmetical and logical operations on the data received from the memory or an input device.
- Register array consists of registers identified by letters like B, C, D, E, H, L and accumulator. The control unit controls the flow of data and instructions within the computer.
- The microprocessor follows a sequence: Fetch, Decode, and then Execute.
- Initially, the instructions are stored in the memory in a sequential order.
- The microprocessor fetches those instructions from the memory, then decodes it and executes those instructions till STOP instruction is reached.
- Later, it sends the result in binary to the output port. Between these processes, the register stores the temporary data and ALU performs the computing functions.

Central Processing Unit (CPU)

- It performs the necessary arithmetic and logic operations and controls the timing and general operation of the complete system

Input/Output (I/O) Devices

- Input devices are used for feeding data into the CPU, examples of these devices are toggle switches, analog-to-digital converters, paper tape readers, card readers, keyboards, disk etc.
- The output devices are used for delivering the results of computations to the outside world; examples are light emitting diodes, cathode ray tube (CRT) displays, digital-to-analog

converters, card and paper-tape punches, character printers, plotters, communication lines etc.

- The inputoutput subsystem thus allows the computer to usefully communicate with the outside world. Input-output devices are also called as peripherals.

I/O Interface

- Communications between an input/output device and the MPU take place through an interface.
- The interface converts the data from a form used by one of these devices to a form acceptable by the other.
- It must also adjust for speed differences between the processor and the other device. • The interface circuits of microcomputers correspond to the I/O control units used on larger computer systems.

Memory

- It stores both the instructions to be executed (i.e., the program) and the data involved.
- It usually consists of both RAMs (random-access memories) and ROMS (read-only memories).

A microprocessor is an integrated circuit designed to function as the CPU of a microcomputer.

Brief history of Microprocessors

4-bit Microprocessors

- The first microprocessor was introduced in 1971 by Intel Corp.
- It was named Intel 4004 as it was a 4 bit processor.
- It was a processor on a single chip.
- It could perform simple arithmetic and logic operations such as addition, subtraction, boolean AND and boolean OR.
- It had a control unit capable of performing control functions like fetching an instruction from memory, decoding it, and generating control pulses to execute it.
- It was able to operate on 4 bits of data at a time.
- This first microprocessor was quite a success in industry.
- Soon other microprocessors were also introduced. Intel introduced the enhanced version of 4004, the 4040.

8-bit Microprocessors

- The first 8 bit microprocessor which could perform arithmetic and logic operations on 8 bit words was introduced in 1973 again by Intel.
- This was Intel 8008 and was later followed by an improved version, Intel

8088.

• Some other 8 bit processors are Zilog-80 and Motorola M6800.

16-bit Microprocessors

• The 8-bit processors were followed by 16 bit processors.

• They are Intel 8086 and 80286.

32-bit Microprocessors

• The 32 bit microprocessors were introduced by several companies but the most popular one is Intel 80386.

Pentium Series

• Instead of 80586, Intel came out with a new processor namely Pentium processor. Its performance is closer to RISC performance.

• Pentium was followed by Pentium Pro CPU.

• Pentium Pro allows multiple CPUs in a single system in order to achieve multiprocessing. • The MMX extension was added to Pentium Pro and the result was Pentiuum II.

• The Pentium III provided high performance floating point operations for certain types of computations by using the SIMD extensions to the instruction set.

• These new instructions makes the Pentium III faster than high-end RISC CPUs.

Features of 8086 Microprocessor

The features of 8086 Microprocessor are:

1) The 8086 is a 16-bit microprocessor. The term “16-bit” means that its arithmetic logic unit, internal registers and most of its instructions are designed to work with 16-bit binary words.

2) The 8086 has a 16-bit data bus, so it can read data from or write data to memory and ports either 16 bits or 8 bits at a time. The 8088, however, has an 8-bit data bus, so it can only read

data from or write data to memory and ports 8 bits at a time.

3) The 8086 has a 20-bit address bus, so it can directly access 2 20 or 10,48,576 (1Mb) memory locations. Each of the 10, 48, 576 memory locations is byte Therefore, a sixteen-bit words are stored in two consecutive memory locations. The 8088 also has a 20-bit address bus, so it can also address 2 20 or 10, 48, 576 memory locations.

4) The Features of 8086 Microprocessor can generate 16-bit I/O address, hence it can access 2 16 = 65536 I/O ports.

5) The 8086 provides fourteen 16-bit registers.

6) The 8086 has multiplexed address and data bus which reduces the number of pins needed, but does slow down the transfer of data (drawback).

7) The 8086 requires one phase clock with a 33% duty cycle to provide optimized internal timing.

8) The Features of 8086 Microprocessor is possible to perform bit, byte, word and block operations in 8086. It performs the arithmetic and logical operations on bit, byte, word and decimal numbers including multiply and divide.

9) The Intel 8086 is designed to operate in two modes, namely the minimum mode and the maximum mode. When only one 8086 CPU is to be used in a microcomputer system, the 8086 is used in the minimum mode of operation. In this mode the CPU issues the control signals required by memory and I/O In multiprocessor (more than one processor in the
system) system 8086 operates in maximum mode. In maximum mode, control signals are generated with the help of external bus controller (8288).

10) The Intel 8086 supports multiprogramming. In multiprogramming, the code for two or more processes is in memory at the same time and is executed in a time-multiplexed fashion.

11) An interesting feature of the 8086 is that it fetches up to six instruction bytes (4 instruction bytes for 8088) from memory and queue stores them in order to speed up instruction execution.

12) The Features of 8086 Microprocessor provides powerful instruction set with the following addressing modes: Register, immediate, direct, indirect through an index or base, indirect through the sum of a base and an index register, relative and implied.

8086 Internal Architecture

- It is internally divided into two separate functional units. These are the Bus Interface Unit (BIU) and the Execution Unit (EU).
- These two functional units can work simultaneously to increase system speed and hence the throughput.
- Throughput is a measure of number of instructions executed per unit time. Fig. 6.2 shows a block diagram of the 8086 internal architecture.

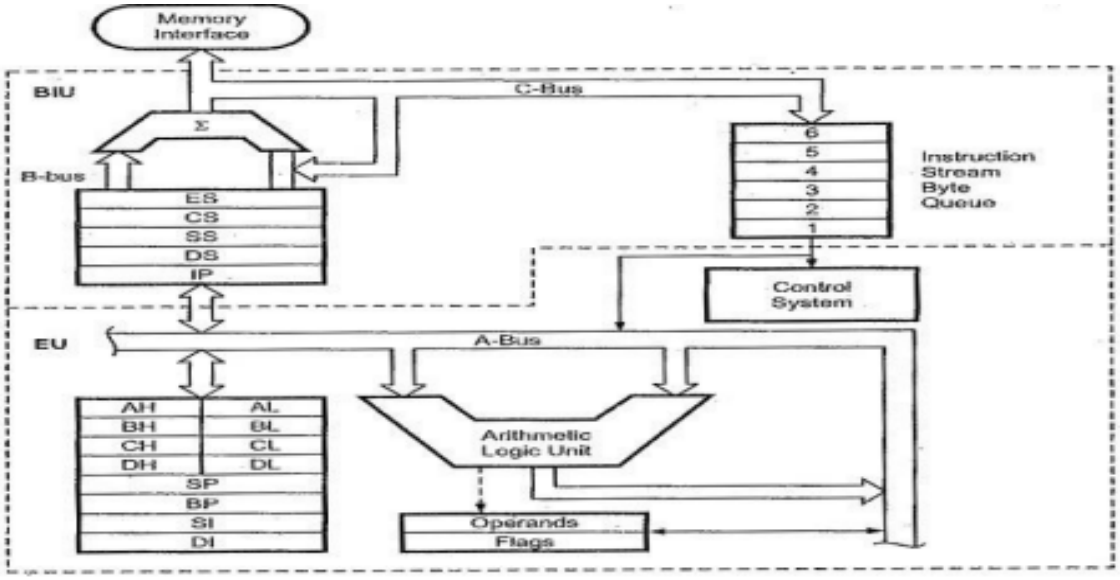


Fig. 6.2 8086 Internal block diagram

Functions of Bus Interface Unit

- It sends address of the memory or I/O.
- It fetches instruction from memory.
- It reads data from port/memory.
- It Writes data into port/memory.
- It supports instruction queuing.
- It provides the address relocation facility.

To implement these functions the BIU contains the instruction queue, segment registers instruction pointer, address summer and bus control logic.

Instruction Queue:

- To speed up program execution, the BIU fetches six instruction bytes ahead of time from the memory.
- These pre-fetched instruction bytes are held for the execution unit in a group of registers called Queue.
- With the help of queue it is possible to fetch next instruction when current instruction is in execution.
- During this execution time the BIU fetches the next instruction or instructions from memory into the instruction queue instead of remaining idle.
- The BIU continues this process as long as the queue is not full.
- Due to this, execution unit gets the ready instruction in the queue and instruction fetch time

is eliminated.

- The queue operates on the principle first in first out (FIFO).
- So that the execution unit gets the instructions for execution in the order they are fetched.
- In case of JUMP and CALL instructions, instructions already fetched in queue are of no use.
- Hence, in these cases queue is dumped and newly formed by loading instructions from new address specified by JUMP or CALL instruction.
- Feature of fetching the next instruction while the current instruction is executing is called pipelining.

Segment Registers:

- The physical address of the 8086 Internal Architecture is 20-bits wide to access 1 Mbyte memory locations.
- However, its registers and memory locations which contain logical addresses are just 16-bits wide.
- Hence 8086 uses memory segmentation.
- It treats the 1 Mbyte of memory as divided into segments, with a maximum size of a segment as 64 Kbytes.
- Thus any location within the segment can be accessed using 16 bits. The 8086 Internal Architecture allows only four active segments at a time, as shown in the Fig. 6.4.
- For the selection of the four active segments the 16-bit segment registers are provided within the BIU of the 8086. These four registers are :
- Code segment (CS) register,
- the data segment (DS) register,
- the stack segment (SS) register,
- and the extra segment (ES) register.

Instruction Pointer:

- The instruction pointer register holds the 16-bit address of the next Code byte within the code segment.
- The value contained in the IP is referred to as an offset.
- This value must be offset from (added to) the segment base address in CS to produce the required 20-bit physical address.

Generation of 20-bit Address:

The contents of the CS register are multiplied by 16 i.e. shifted by 4 position to the left by inserting 4 zero bits and then the offset i.e. the contents of IP register are added to the shifted

contents of CS to generate physical address.

Execution Unit [EU]:

The execution unit of 8086 Internal Architecture tells the BIU from where to fetch instructions or data, decodes instructions and executes instructions. It contains

- Control Circuitry
- Instruction Decoder
- Arithmetic Logic Unit (ALU)
- Flag Register
- General Purpose Registers
- Pointers and Index Registers

Control Circuitry, Instruction Decoder, ALU:

- The control circuitry in the EU directs the internal operations.
- A decoder in the EU translates the instructions fetched from memory into a series of actions wlifeh the EU performs.
- ALU is 16-bit. It can add, subtract, AND, OR, XOR, increment, decrements, complement and shift binary numbers.

Flag Register:

A flag is a flip–flop which indicates some condition produced by the execution of an instruction or controls certain operations of the EU

General Purpose Registers:

• The EU has. 8 general purpose registers labeled ATI, AL, BH, BL, CH, CL, DH, and DL.
• These registers can be used individually for temporary storage of 8 bit data.

• The AL register is also called accumulator.

• Certain pairs of these general purpose registers can be used together to store 16-bit data, such as AX, BX, CX and DX.

Pointers and Index Registers:

- All segment registers are 16-bit. But it is necessary to put 20-bit address (physical address) on the address bus.
- To get 20-bit physical address one more register is associated with each segment register the way IP is associated with CS.
- These additional registers belong to the pointer and index group.
- The pointer and index group consists of instruction pointer (IP), stack pointer (SP), BP (base pointer), source index (SI) and destination index (DI) registers.

Stack Pointer (SP) : The stack pointer (SP) register contains the 16-bit offset from the start of the segment to the top of stack. For stack operation, physical address is produced by adding the contents of stack pointer register to the segment base address in SS

MULTICORE PROCESSOR

A multicore processor is a single integrated circuit (a.k.a., chip multiprocessor or CMP) that contains multiple core processing units, more commonly known as *cores*. There are many different multicore processor architectures, which vary in terms of

- **Number of cores**. Different multicore processors often have different numbers of cores. For example, a quad-core processor has four cores. The number of cores is usually a power of two.
- **Number of core types**.
    - **Homogeneous (symmetric) cores.** All of the cores in a homogeneous multicore processor are of the same type; typically the core processing units are general-purpose central processing units that run a single multicore operating system.
    - **Heterogeneous (asymmetric) cores**. Heterogeneous multicore processors have a mix of core types that often run different operating systems and include graphics processing units.
- **Number and level of caches**. Multicore processors vary in terms of their instruction and data caches, which are relatively small and fast pools of local memory.
- **How cores are interconnected.** Multicore processors also vary in terms of their bus architectures.
- **Isolation**. The amount, typically minimal, of in-chip support for the spatial and temporal isolation of cores:
    - **Physical isolation** ensures that different cores cannot access the same physical hardware (e.g., memory locations such as caches and RAM).
    - **Temporal isolation** ensures that the execution of software on one *core* does not impact the temporal behavior of software running on another *core.*

Pros of Multicore Processing

1. **Energy Efficiency**. By using multicore processors, architects can decrease the number of embedded computers. They overcome increased heat generation due to Moore's Law which in turn decreases the need for cooling. The use of multicore processing reduces power consumption (less energy wasted as heat), which increases battery life.
2. **True Concurrency**. By allocating applications to different cores, multicore processing increases the intrinsic support for actual (as opposed to virtual) parallel processing within individual software applications across multiple applications.
3. **Performance**. Multicore processing can increase performance by running

multiple applications concurrently.

4. **Isolation**. Software running on one core is less likely to affect software on another core than if both are executing on the same single core. **Reliability and Robustness**. Allocating software to multiple cores increases reliability and robustness (i.e., fault and failure tolerance) by limiting fault and/or failure propagation from software on one core to software on another.
5. **Obsolescence Avoidance**. The use of multicore processors enables architects to avoid technological obsolescence and improve maintainability.
6. **Hardware Costs**. By using multicore processors, architects can produce systems with fewer computers and processors.

Cons of Multi Core Processor

1. **Shared Resources.** Cores on the same processor share both *processor-internal* resources (L3 cache, system bus, memory controller, I/O controllers, and interconnects) and *processor-external* resources (main memory, I/O devices, and networks). These shared resources imply (1) the existence of single points of failure, (2) two applications running on the *same* core can interfere with each other, and (3) software running on one core can impact software running on *another* core (i.e., interference can violate spatial and temporal isolation because multicore support for isolation is limited). The diagram below uses the color red to illustrate six shared resources.
2. **Interference**. Interference occurs when software executing on one core impacts the behavior of software executing on other cores in the same processor.