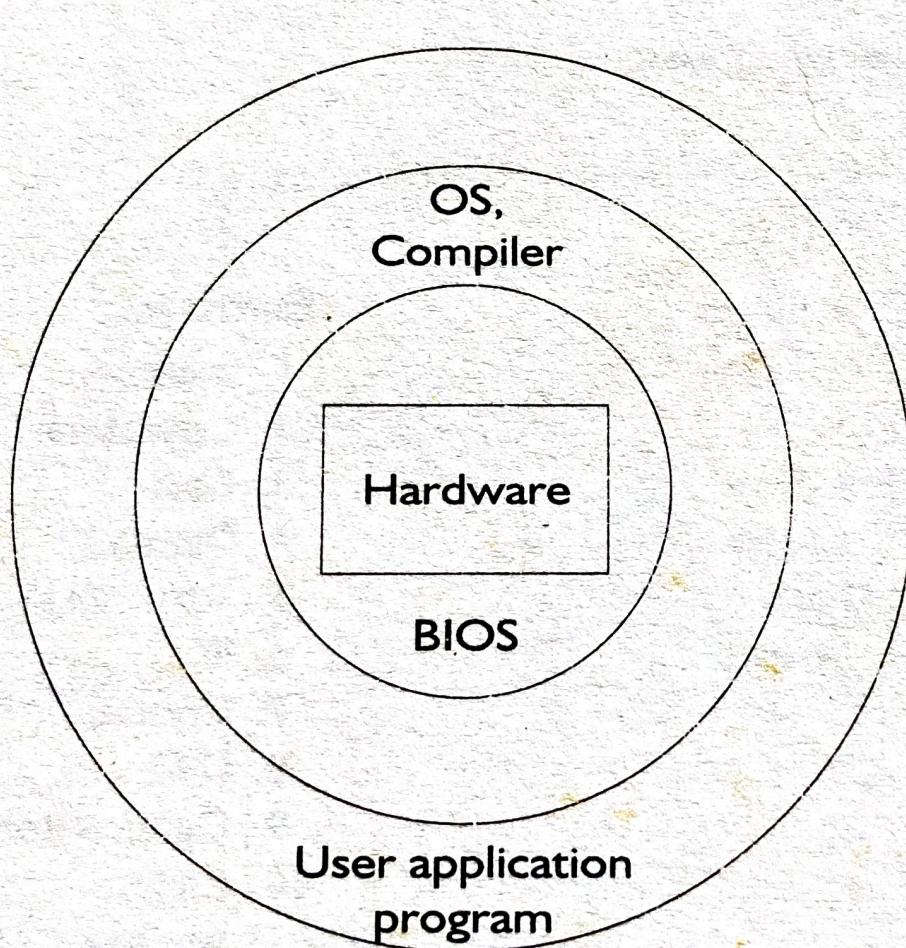


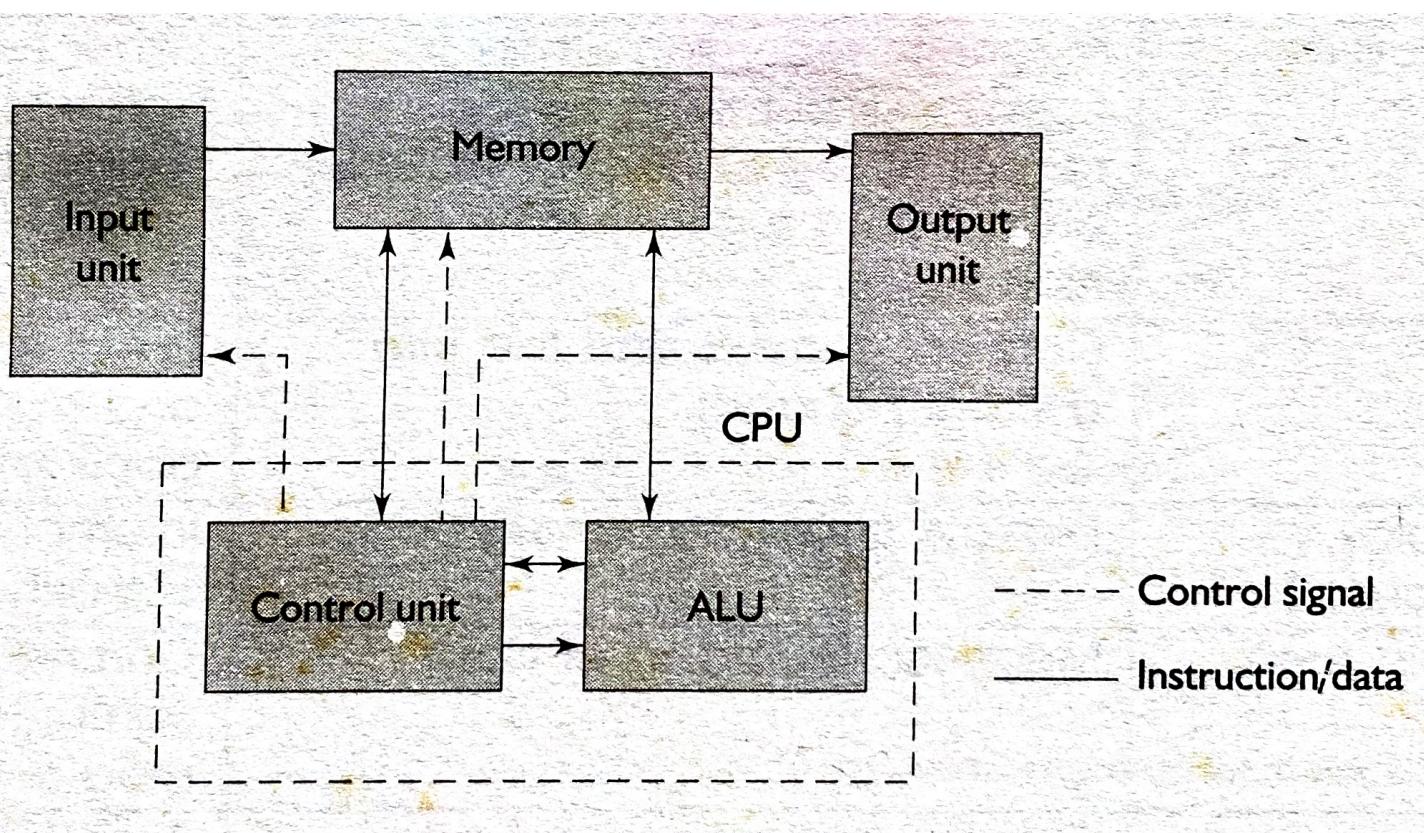
Layers in a computer



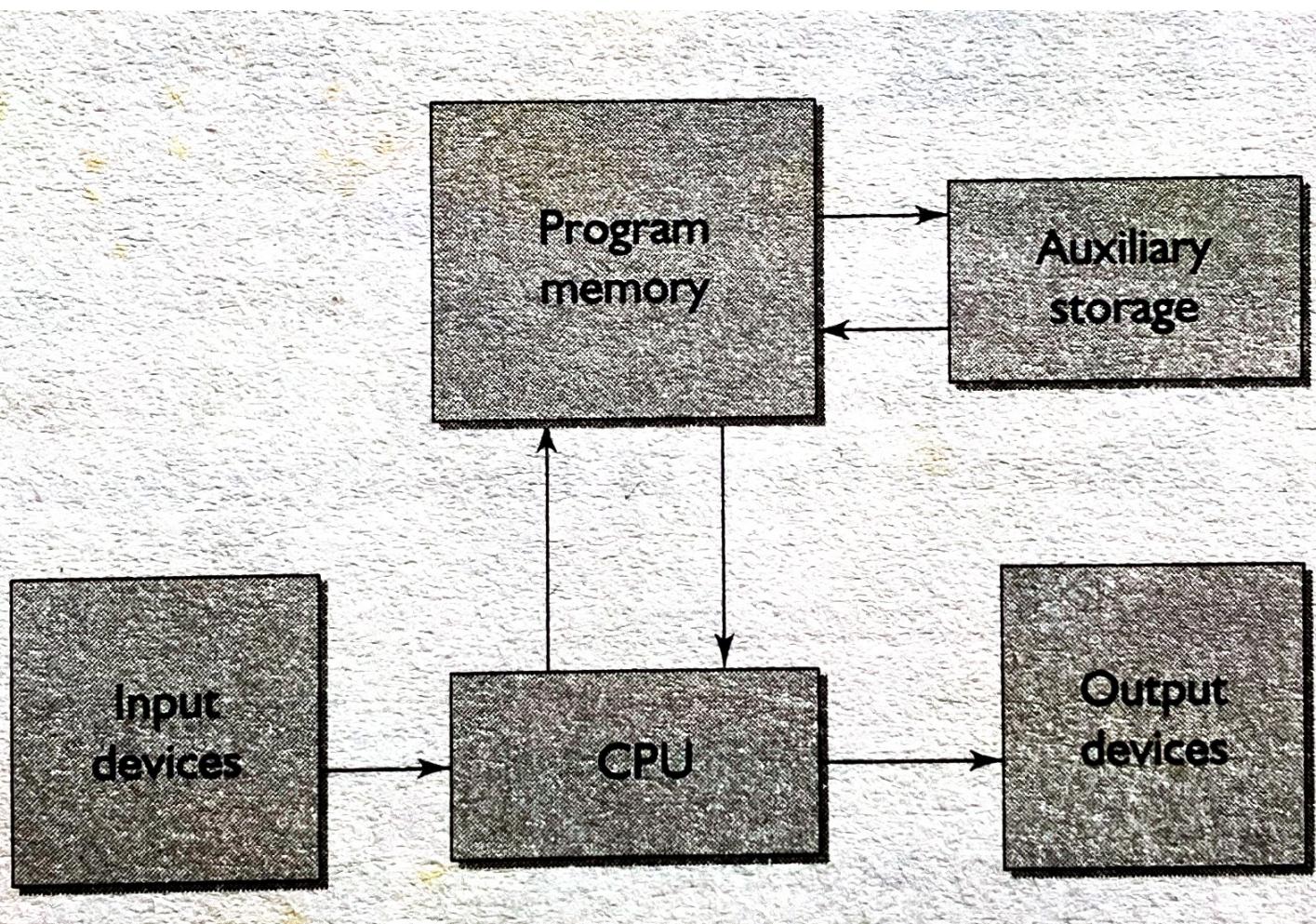
BIOS—Basic input output
control system

OS—Operating system

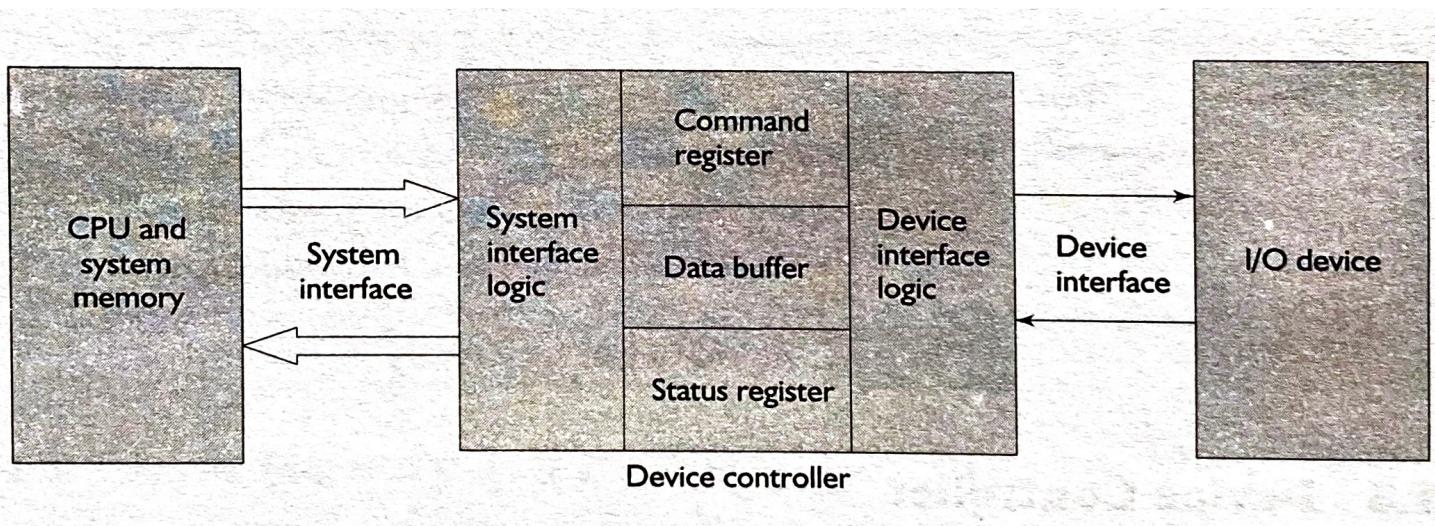
Functional units in a computer



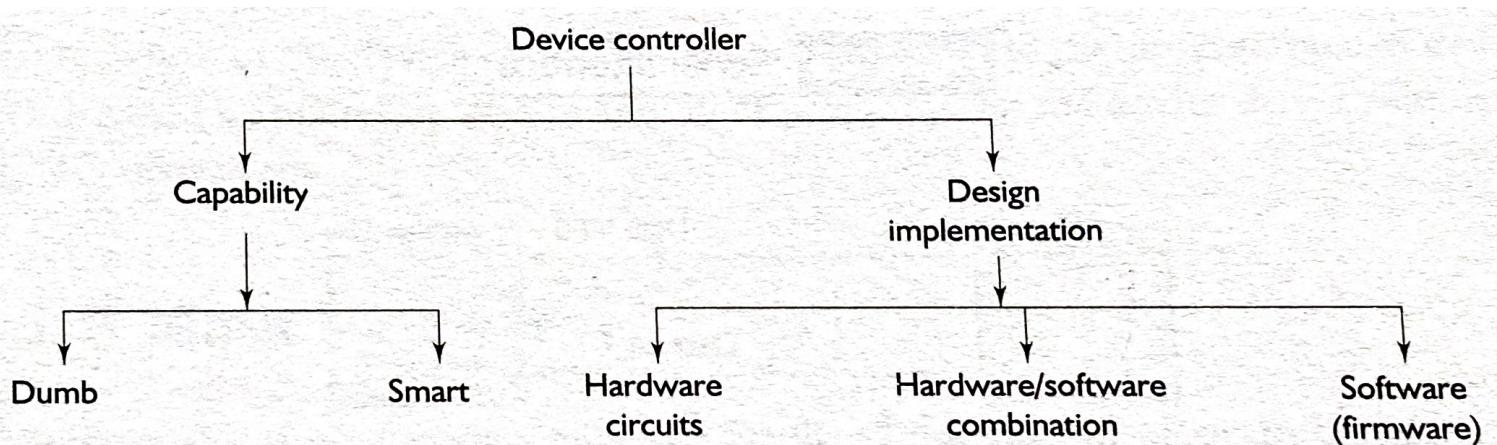
Auxiliary storage

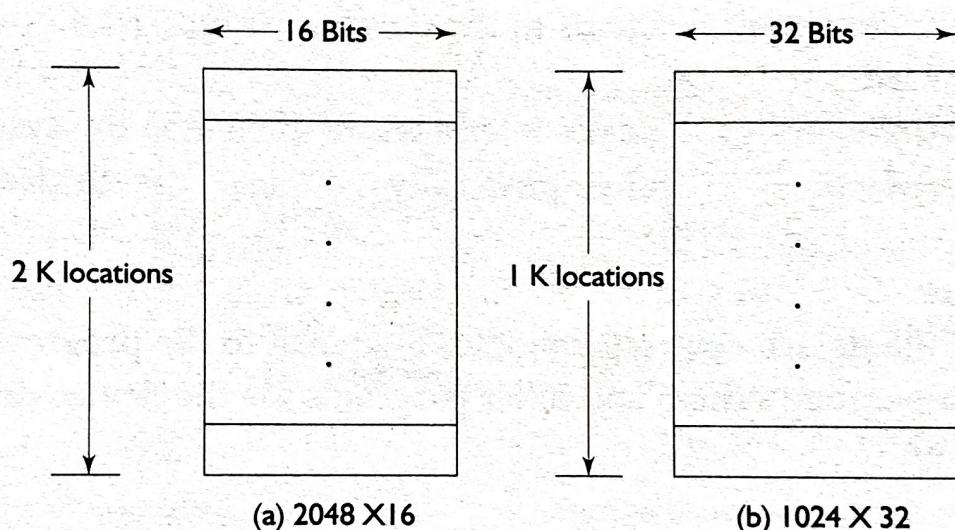


Device controller: device and system interface



Device controller: classification





Memory capacity and organization

Example

A computer has a main memory with 1024 locations of each 32-bits. Calculate the total memory capacity:

Word length = 32 bits = 4 bytes;

No. of locations = 1024 = 1 kilo = 1K;

Memory capacity = $1K \times 4 \text{ bytes} = 4 \text{ KB}$

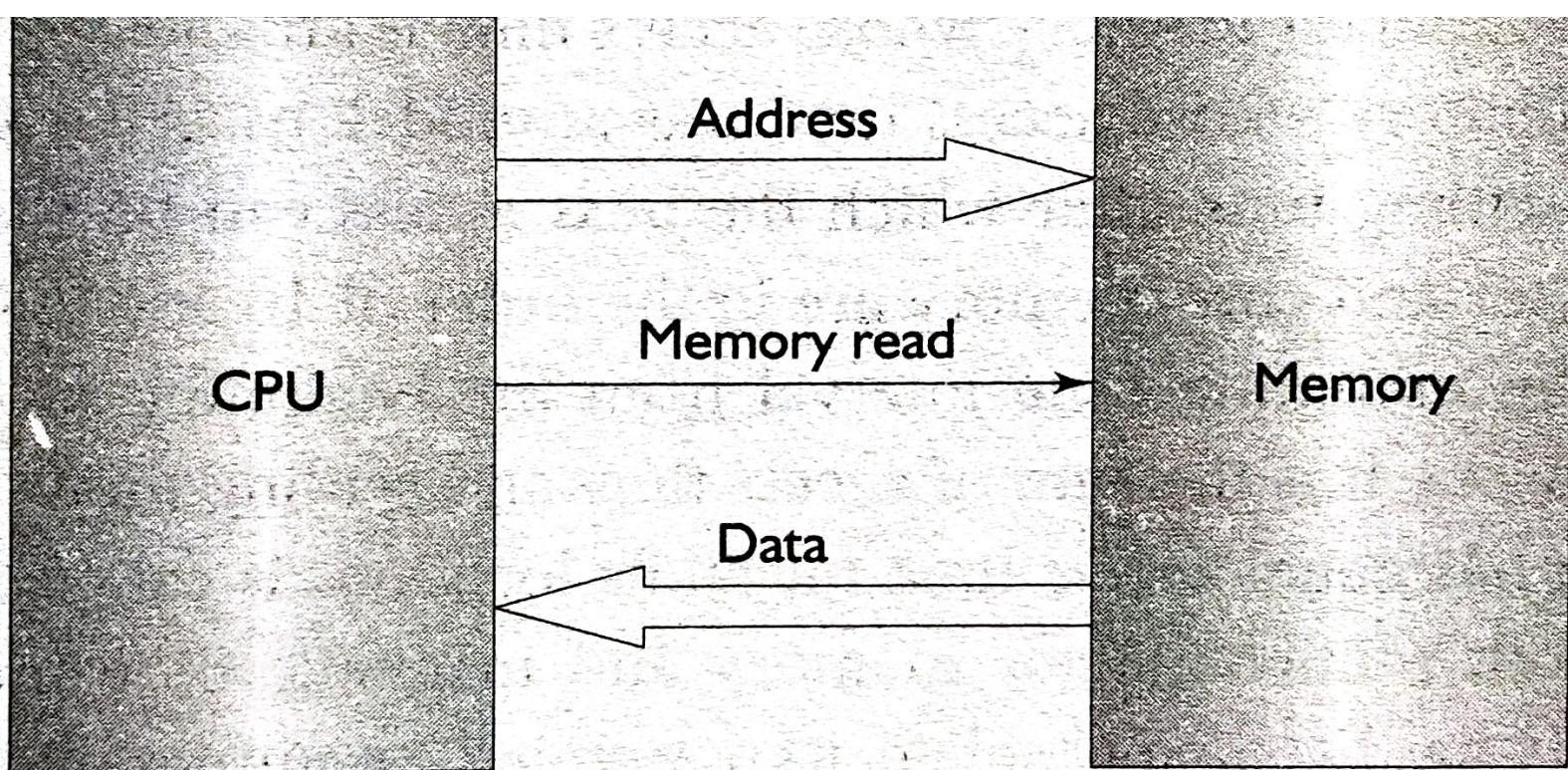
Example A main memory has an access time of 45 ns. A 5 ns time gap is necessary from the completion of one access to beginning of next access. Calculate the bandwidth of the memory.

Access time = 45 ns; settling time = 5 ns;

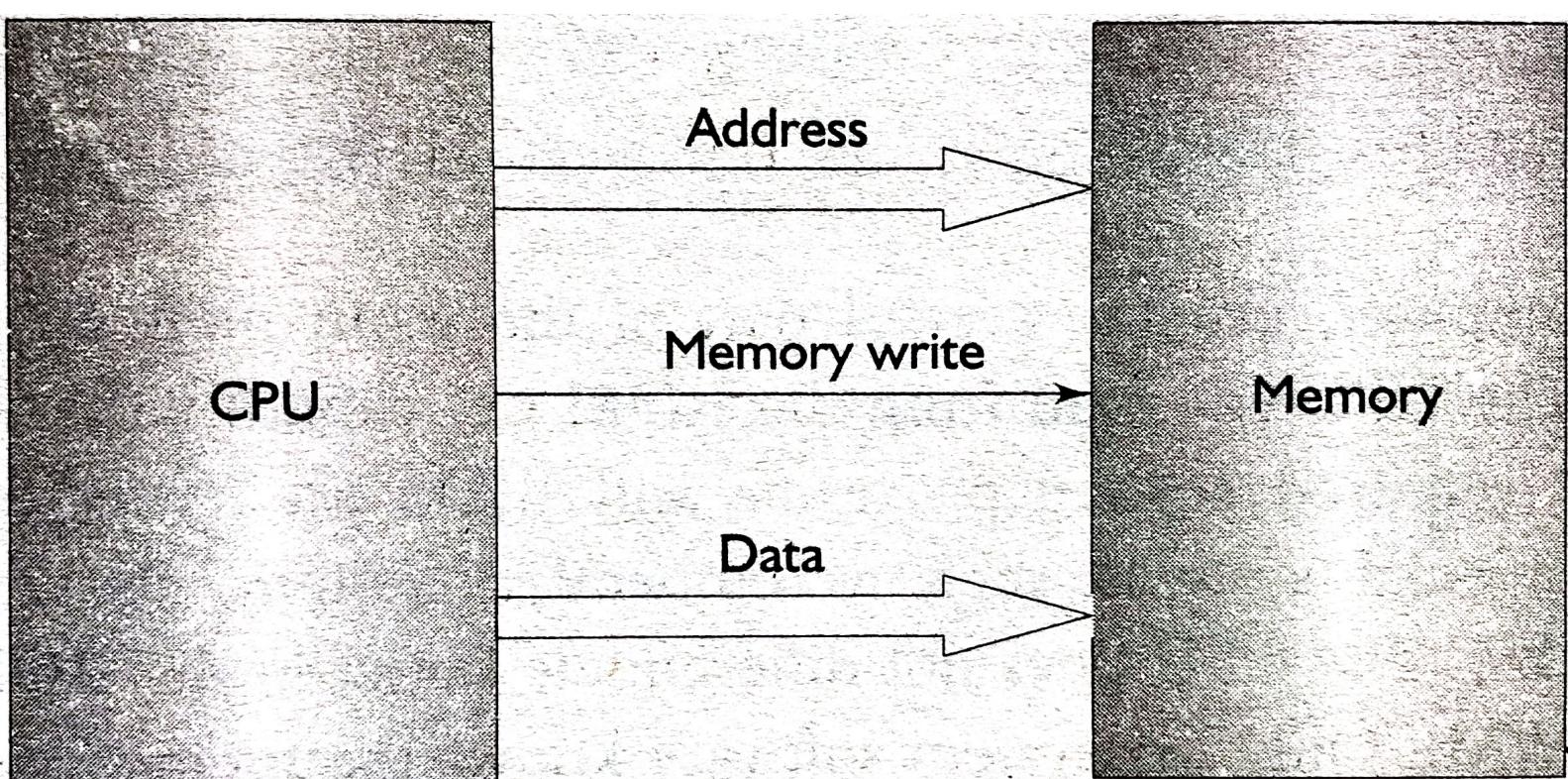
Cycle time = access time + settling time = 45 ns + 5 ns = 50 ns;

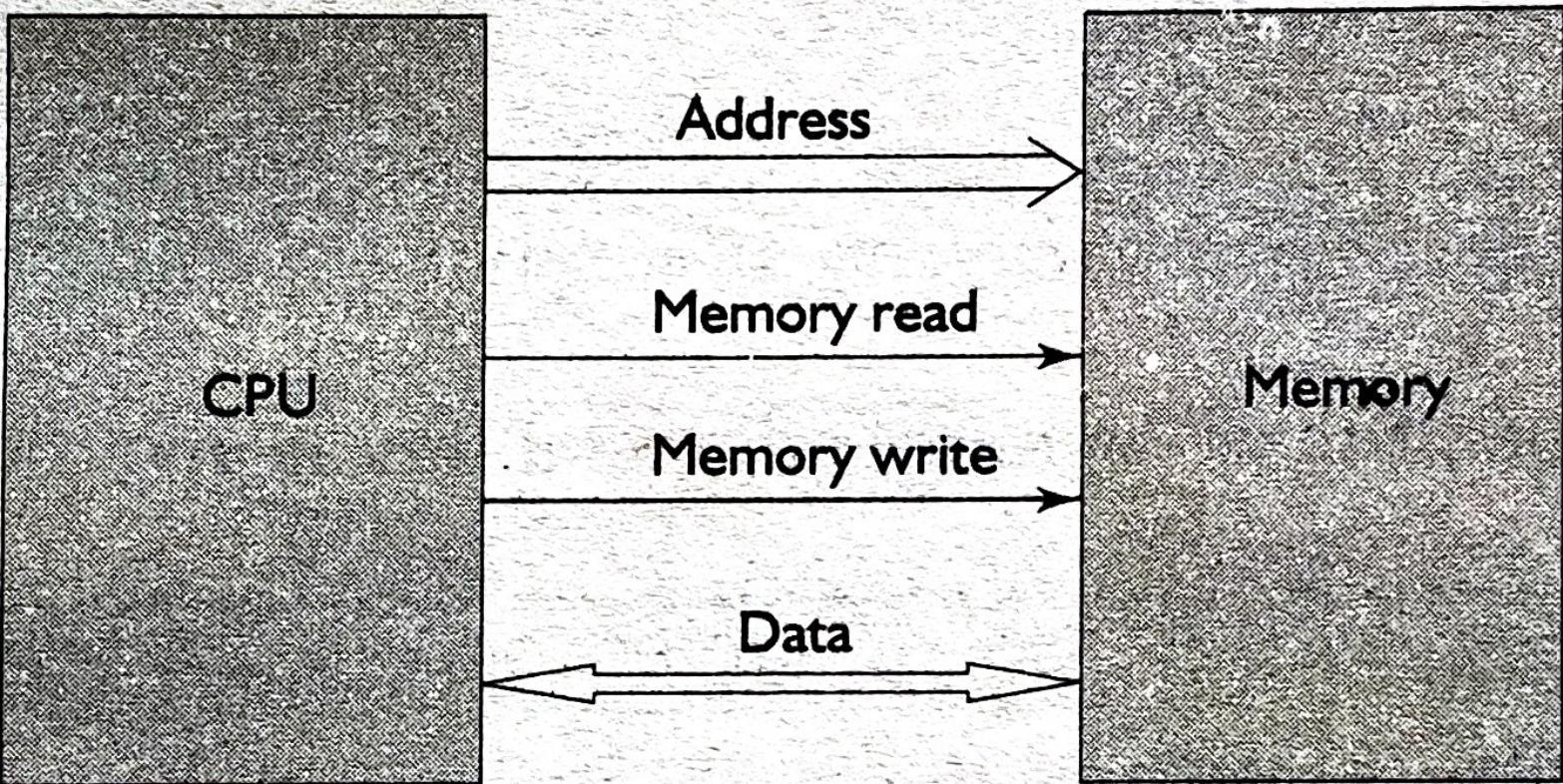
Bandwidth = 1/cycle time = 1/50 ns = 20 MHz.

Memory read operation

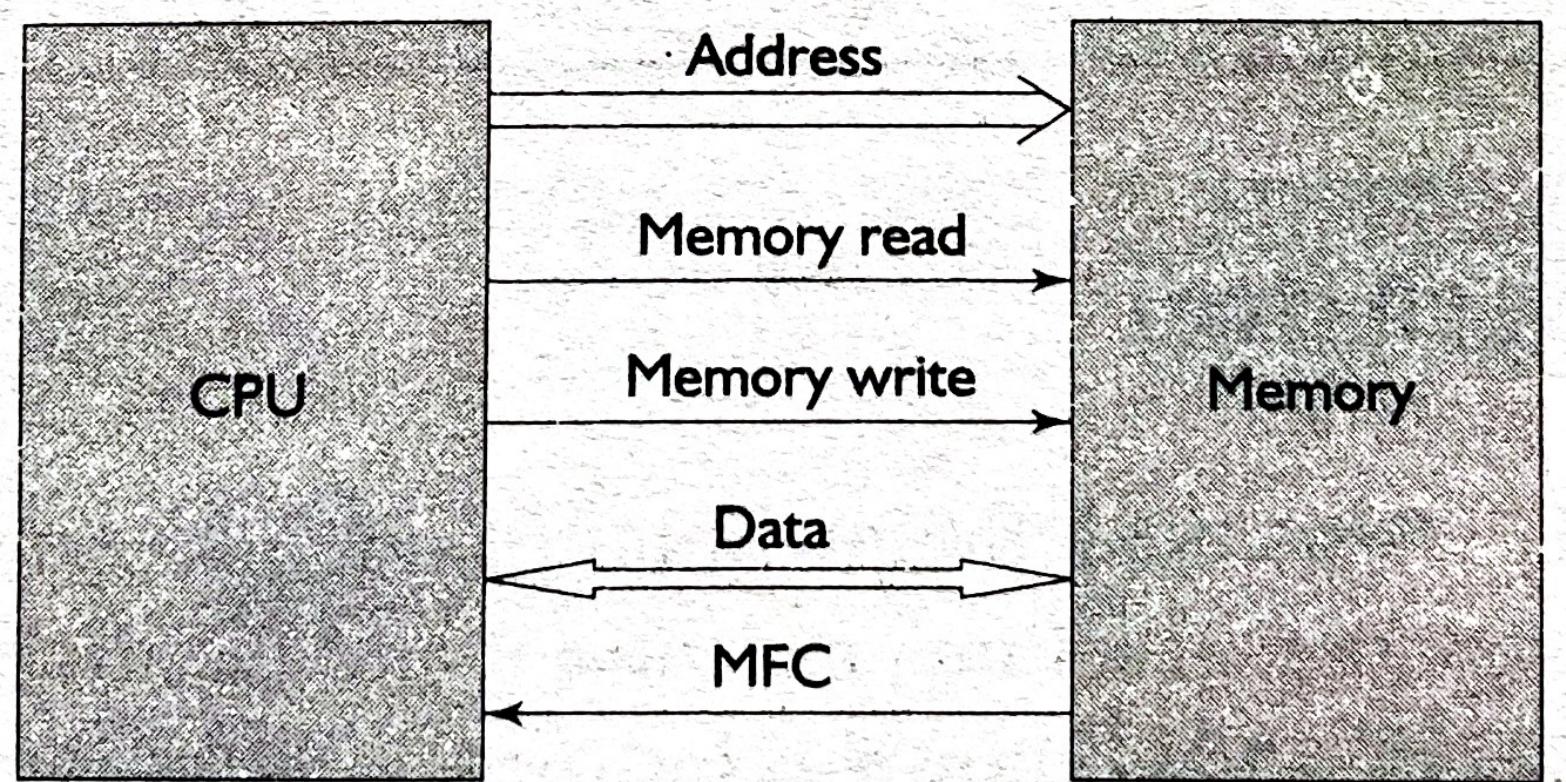


Memory write operation





Synchronous memory interface



Asynchronous memory interface

Memory addressability

Example A CPU has a 12 bit address for memory addressing: (a) What is the memory addressability of the CPU ? (b) If the memory has a total capacity of 16 KB, what is the word length of the memory?

No. of address bits = 12;

Memory addressability = $2^{12} = 4$ kilolocations;

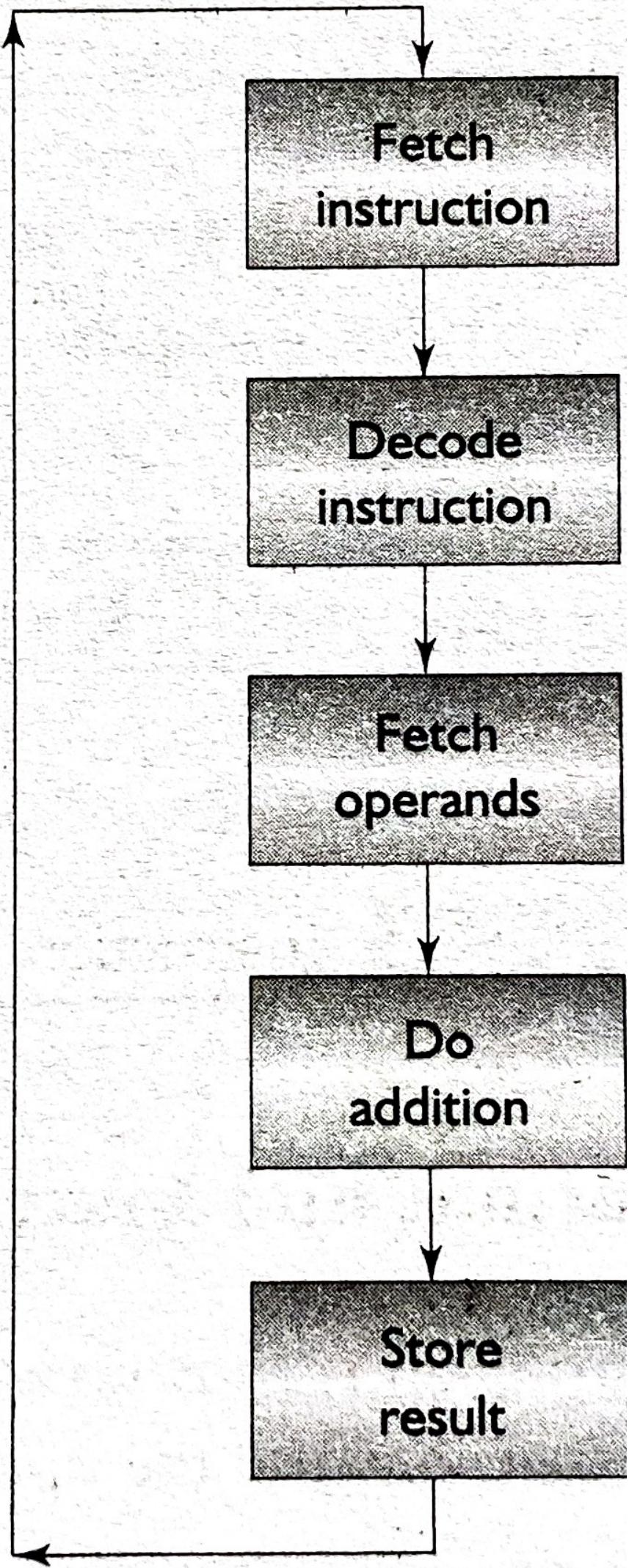
Memory capacity = 16 KB;

Word length = memory capacity/no. of locations = 16 KB/4K = 4B = 4 bytes.

CPU	No. of address bits	Memory addressability
IBM System 360/40	24	16 mega
Intel 8080	16	64 kilo
Intel 8088	20	1 mega
Pentium	32	4 giga
Unknown	40	1 tera

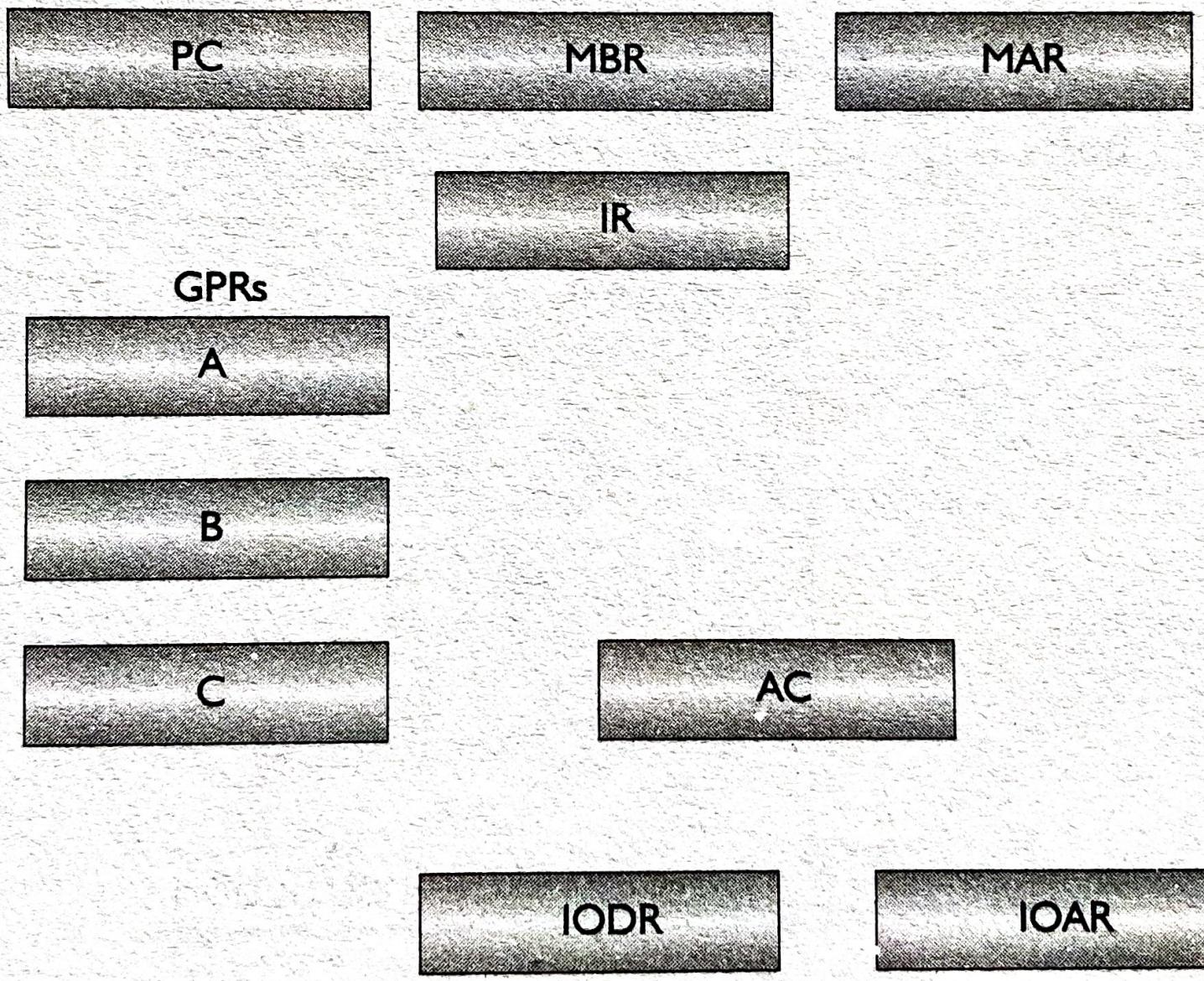
Instruction
cycle
steps

Next
instruction

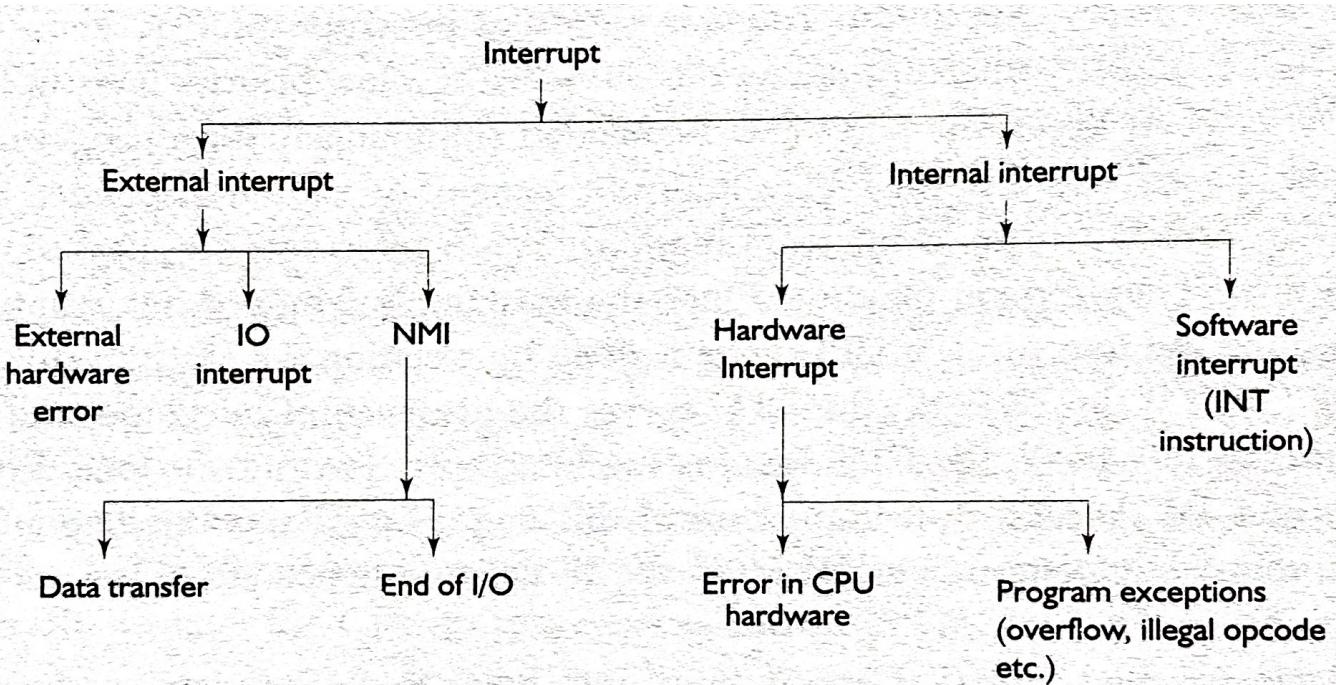


S. no.	Step	Action responsibility	Remarks
1	Instruction fetch	Control unit; external action	Fetches next instruction from main memory
2	Instruction decode	Control unit; internal action	Analyses opcode pattern in the instruction and identifies the exact operation specified
3	Operand fetch	Control unit: external (memory) or internal action depending on the location of operands	Fetches the operands, one by one, from main memory or CPU registers and supply them to ALU
4	Execute (ADD)	ALU; internal action	Specified arithmetic or logical operation is done
5	Result store	Control unit; external or internal action	Stores the result in memory or registers

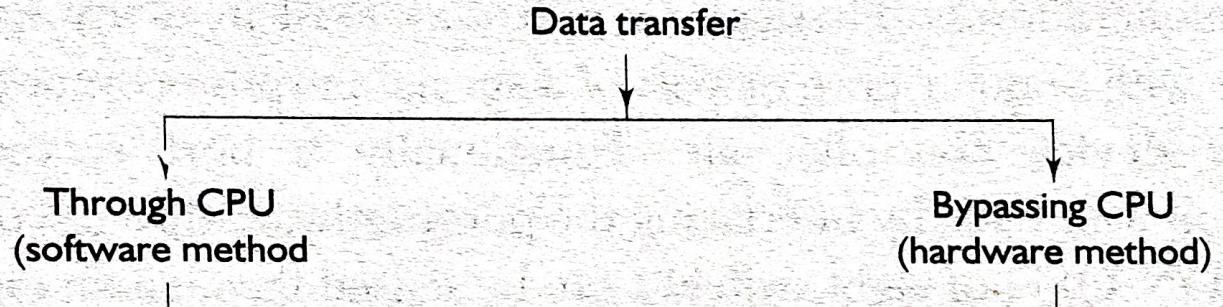
CPU registers



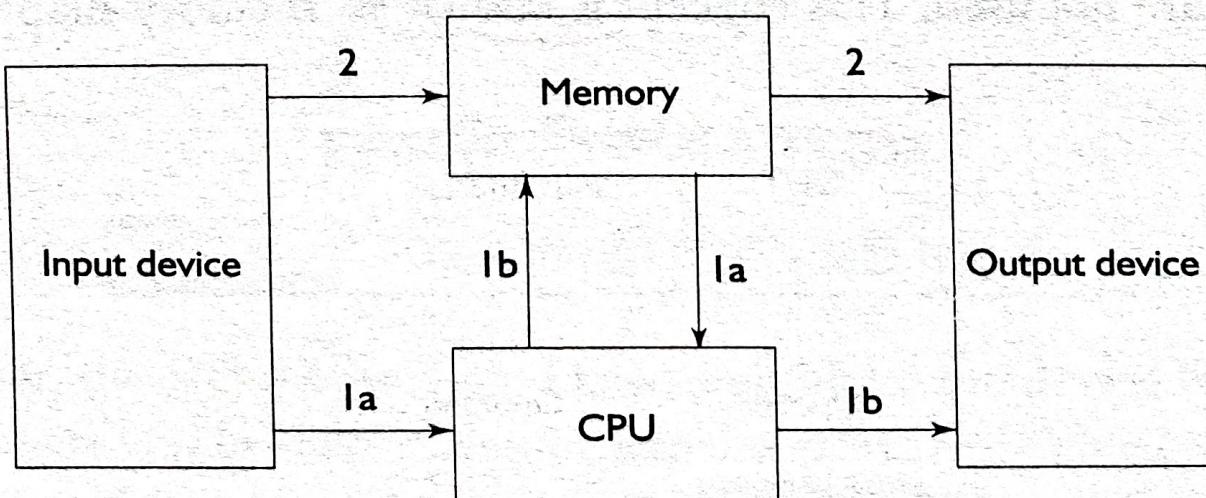
Classification of Interrupts



S. No.	NMI Cause	Definition	ISR action
1	Power fail	Advance warning that AC power is likely to be cut off	Control may be given to SAVE routine which saves CPU status. If a power backup (UPS etc.) is available, transfer of power source is done
2	Memory parity error	While reading from memory, it is detected that some data bits have failed	Control is given to ERROR RECOVERY routine which may either cancel the program or ask for operator response
3	Bus cycle malfunction	Current bus cycle has some malfunction	Control is given to MACHINE CHECK routine which takes a decision whether to retry or abort with a message

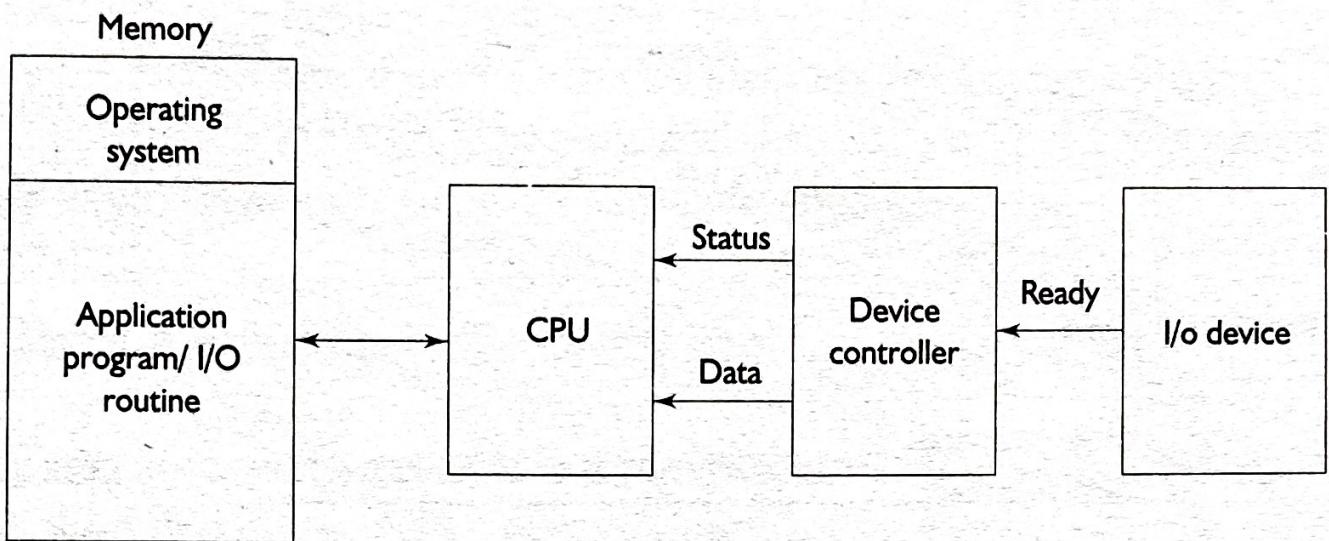


Methods of data transfer

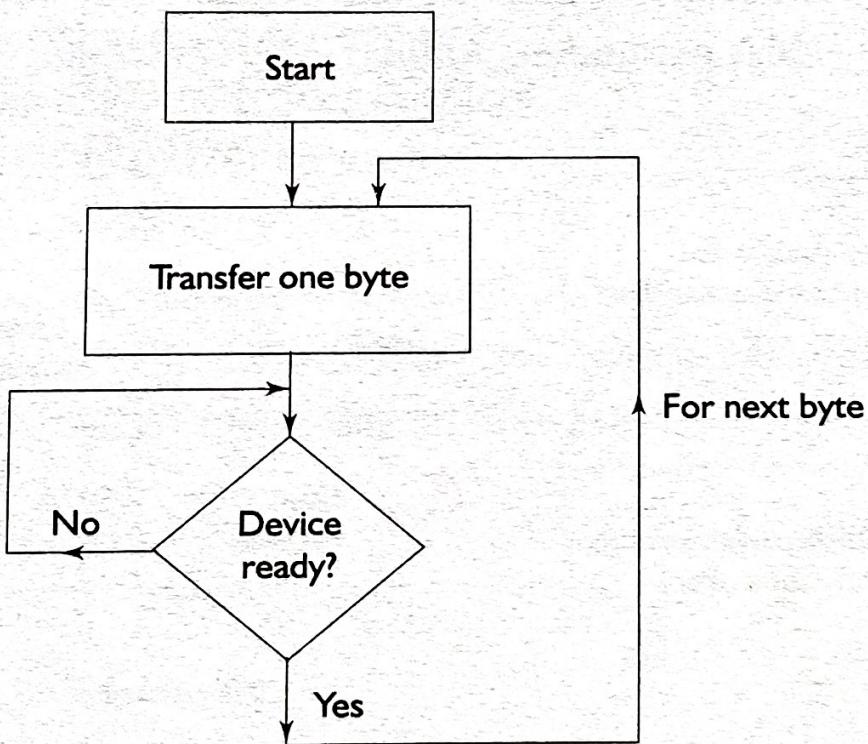


Ia, Ib—Two steps in software method; 2—hardware method

Principles of data transfer

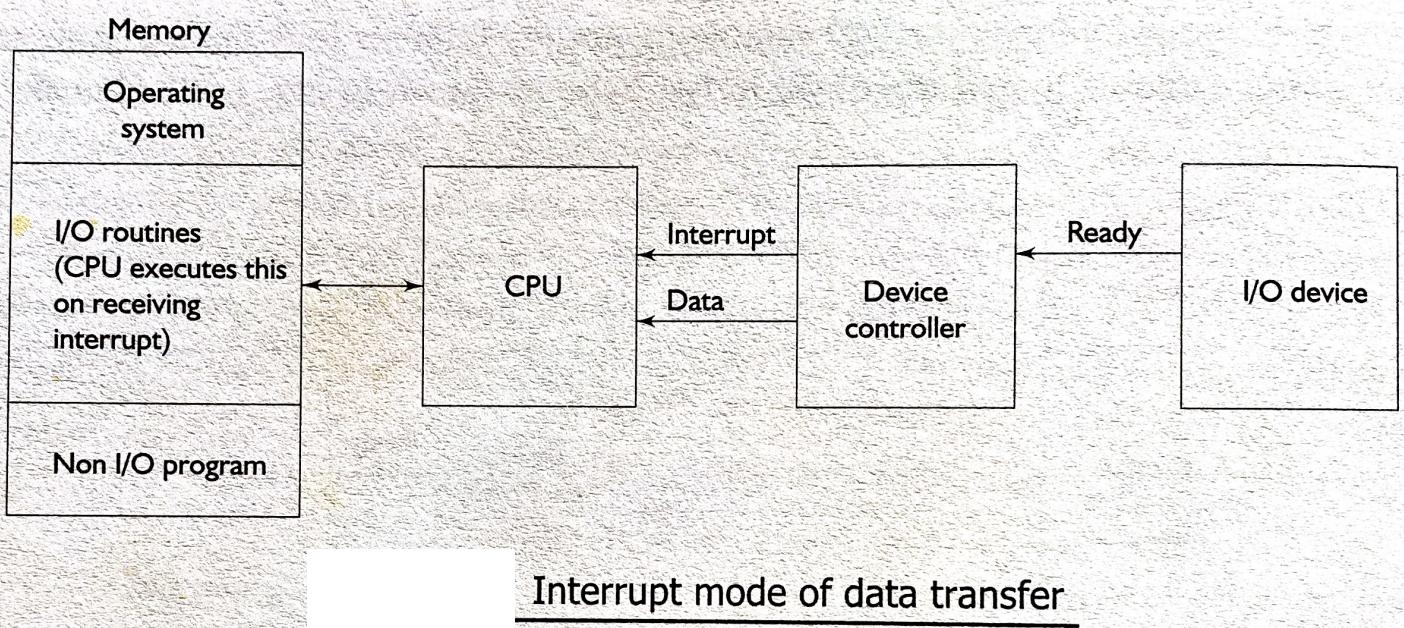


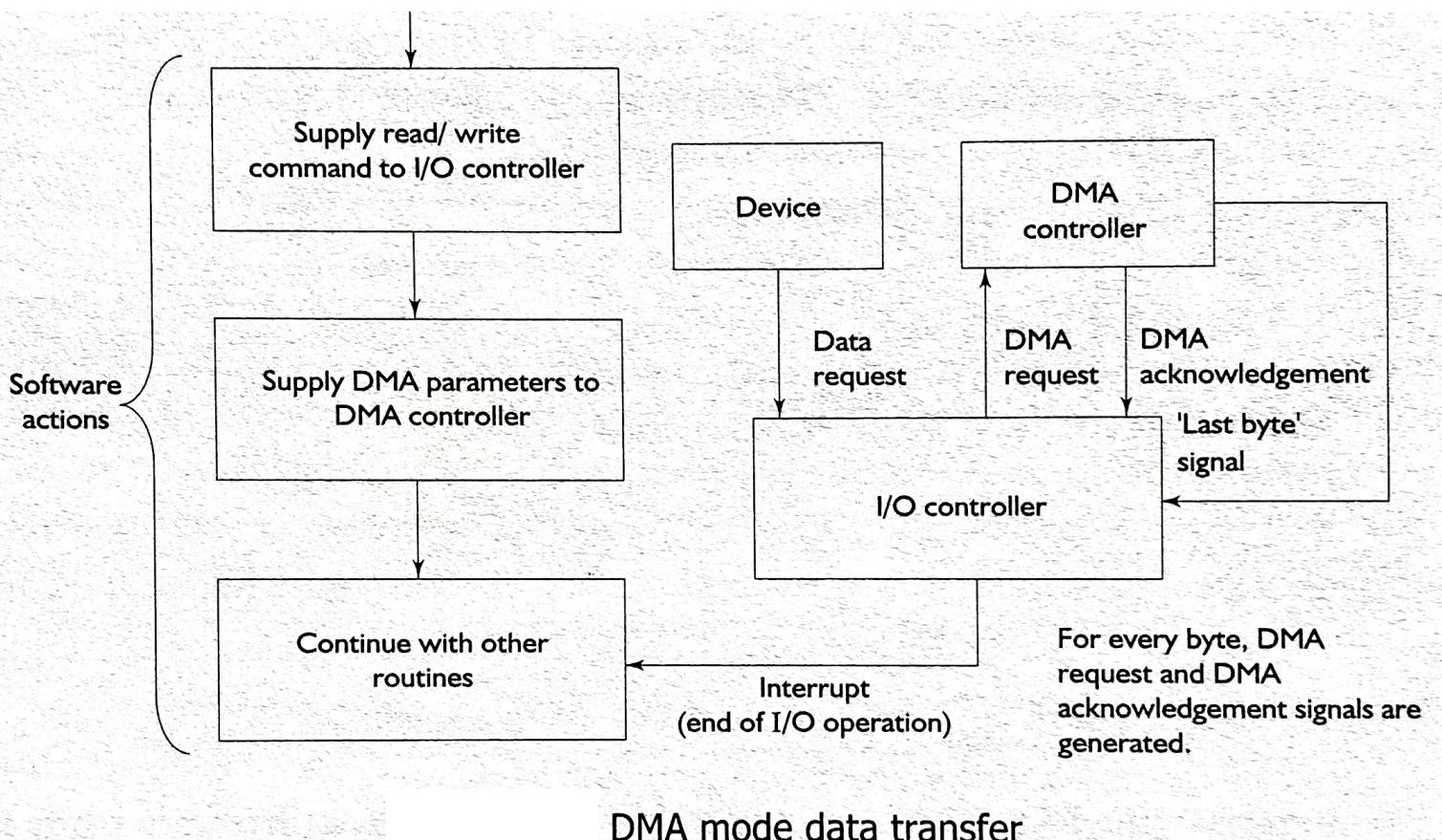
(a) Block Diagram



(b) Sequence

Programmed mode data transfer

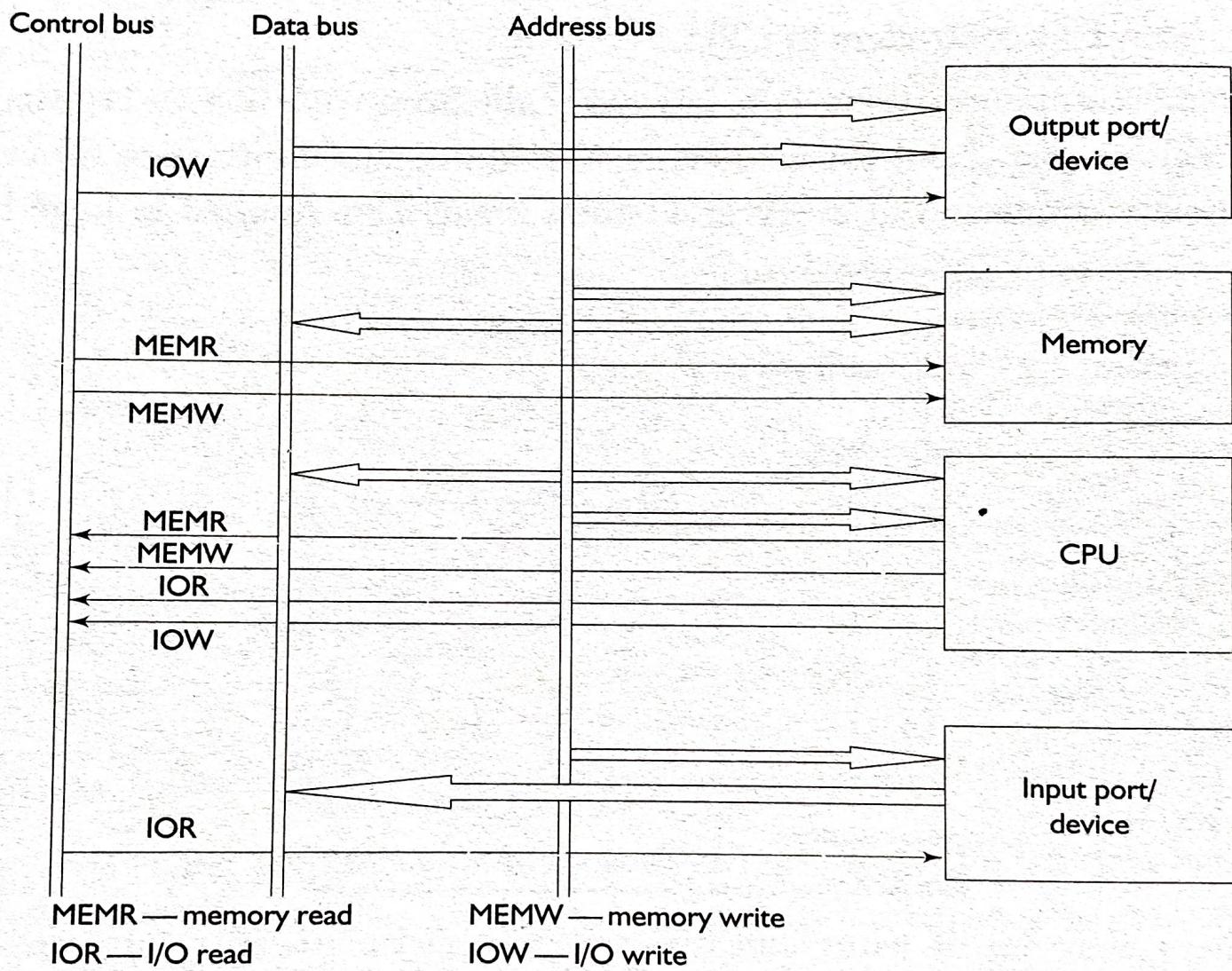




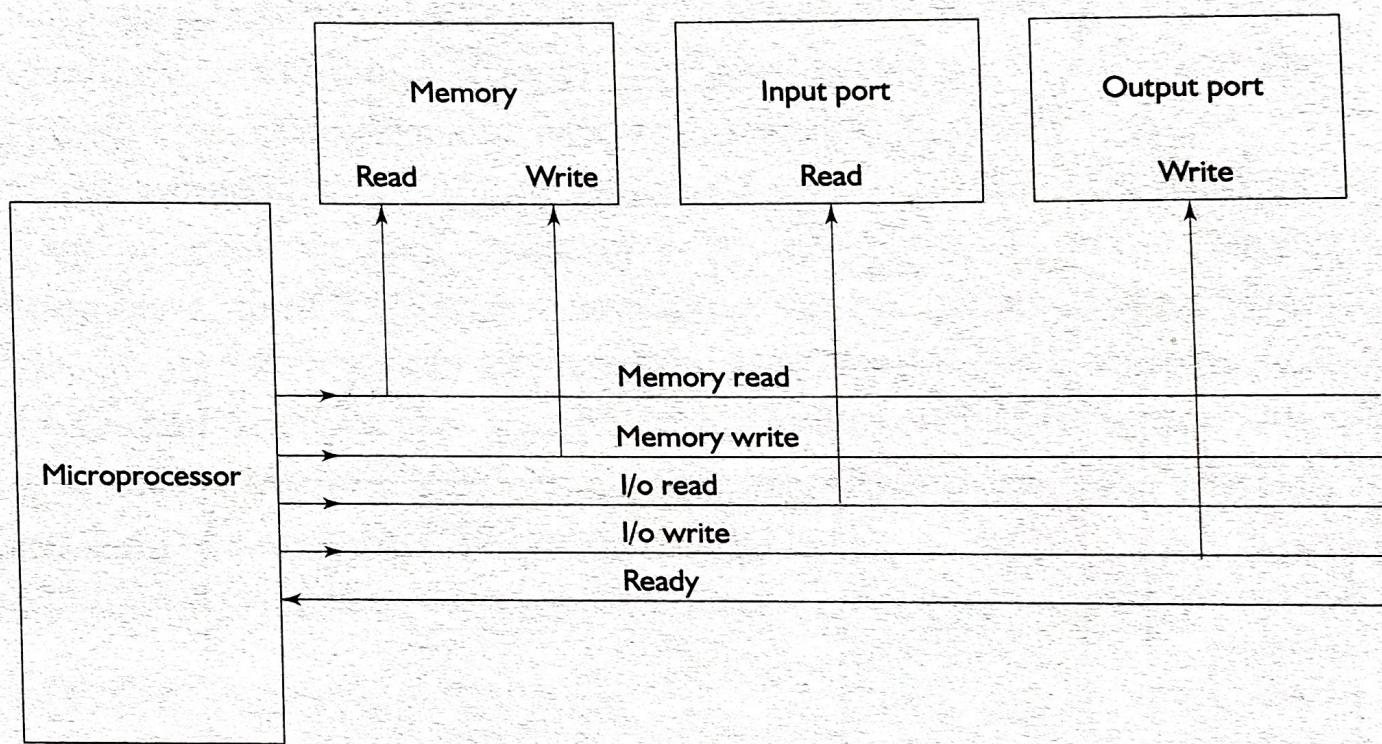
Buses

Connecting paths are necessary inside a computer for carrying following types of information between the subsystems (CPU, memory, and I/O controllers):

1. Instruction from memory to CPU.
2. Data from memory to CPU.
3. Data from CPU to memory.
4. Memory Address from CPU to memory.
5. Port Address from CPU to I/O controllers.
6. Command from CPU to I/O controllers.
7. Status from I/O controllers to CPU.



Simple bus structure



Control bus signals

Control bus signals definition

S. No.	Bus control signal	Description
1	Memory read	Output signal from CPU; indicates that the addressed memory has to put data on data bus
2	Memory write	Output signal: indicates that the addressed memory has to take data from data bus
3	I/O read	Output signal from CPU; indicates that the selected input port has to put data on data bus
4	I/O write	Output signal from CPU; indicates that the selected output port has to take data from data bus
5	Ready	Input signal to CPU; indicates that the addressed subsystem (memory or I/O port) has completed the read/write operation

Computer performance factors

- Instruction fetch: memory access time;
- Instruction decode: control unit speed;
- Operand address calculation: (1) GPRs access time/memory access time (2) address addition time;
- Operand fetch: memory access time/GPRs access time;
- Execute: addition time;
- Store result: main memory access time/GPRs access time.

execution time for a program (T_p) is related to the clock speed and the actual program by the following equation:

$$T_p = \frac{N_{ie} \times CPI}{F}$$

where N_{ie} is the number of instructions executed (encountered) by the CPU (not total instructions in the program), CPI the average number of clock cycles needed for an instruction and F the clock frequency. It should be noticed that N_{ie} is not equal to the total number of instructions in the program. Some of the instructions in the program may not be executed at all i.e. they may be skipped because of program control. Similarly some instructions may be executed several times because of loops.

From the above performance equation, it appears that to reduce program execution time T_p , the following approaches can be taken: reducing N_{ie} , reducing CPI and increasing F . Reducing N_{ie} involves having less instructions in the compiled program which is related to the compiler efficiency and the instruction set. Reducing CPI involves better CPU design to shorten instruction cycle time. Increasing F involves higher clock frequency which depends on technology. However while designing a computer system, these points have to be considered together since improving one parameter may affect other parameters.

System Performance Measurement

In the early days, the term MIPS (Millions of Instructions executed Per Second) was commonly used to indicate the speed of a computer. Since instruction cycle time is different for different instructions, the MIPS value will differ for different instruction mix in the program. Hence, the MIPS value of a computer gives only a rough idea about system performance.

Benchmark programs are specially developed evaluation programs which can be used to compare the performance of different computers. The time taken by a computer system to execute a benchmark program serves as a measure of its performance. This is used for decision making by buyers, developers and marketing teams. There are two important aspects to be considered here:

1. Different instruction types have different execution times.
2. Number of occurrence of an instruction type varies with different types of programs.

The ability of a benchmark program to give reasonably accurate information about the system performance of a computer depends on the instruction mix used in the benchmark program. Since application programs of different types (scientific, commercial etc.) have different instruction mix, it is very difficult to develop a benchmark program which will truly stimulate all real life programs. Hence, separate benchmark programs are developed with different instruction mix for each type of application: scientific, commercial, educational etc. While no benchmark can fully represent overall system performance, the results of a group of carefully selected benchmarks can give valuable information about real performance.

SPEC Rating

The SPEC (System Performance Evaluation Corporation) is a non-profit organization dedicated for performance evaluation. It selects typical application programs for different application areas and publishes (announces) the performance results for important commercial computers. The performance of one of the commercially available computer is taken as the reference computer. The performance of other computers are rated as a relative measure of the standard computer. The following are some of the standards released by SPEC:

SPEC 1995, SPEC 2000, Java benchmarks: JBB2000 and JVM98

Web server

benchmarks: SPECweb99_SSL, SPECweb99 and SPECweb96

A Mail server

benchmark: MAIL2001
an NFS benchmark: SFS97

The SPEC rating is specified as follows:

$$\text{SPEC rating for } X = \frac{\text{Program execution time for standard computer}}{\text{Program execution time for } X}$$

In practice, a suit of programs of various types is run and the SPEC rating is calculated for each. The SPEC rating is a reflection of multiple factors: CPU performance, Memory performance, System organization, Compiler efficiency, Operating system performance etc.

SPEC does two different roles: developing benchmarks and publishing results.

1. *Developing suites of benchmarks*: These suites have sets of benchmark programs with source codes and tools. The programs in the suites are developed by SPEC from programs donated by various sources (generally system manufacturers). SPEC aims on portability and creates tools and meaningful functions for the benchmarks programs.
2. *Publishing news and benchmark results*: Along with performance results, SPEC also provides additional information such as submitted results, benchmark descriptions, background information, and tools; these help in performance comparisons.

SPEC Groups

Presently SPEC has become an umbrella organization with three groups: OSG, HPG, and GPC.

Open Systems Group (OSG): OSG group develops benchmarks for desktop systems, workstations and multi-user servers supporting open operating system environments.

High Performance Group (HPG): HPG group develops benchmarks for symmetric multiprocessor systems, workstation clusters, distributed memory parallel systems, and traditional vector and vector parallel supercomputers. These benchmarks represent large, real applications, in scientific and technical computing.

Graphics Performance Characterization Group (GPC): Responsible for industry standard graphics benchmarks for graphical and multimedia applications, subsystems, OpenGL, etc.