Computer Organization

II-II CSE(autonomous)

Units:

1. Basic Structure of Computers
2. Register Transfer Language and Microoperations
3. Control Unit Design, Arithmetic & Logic Operations
4. 8086 Microprocessors architecture
5. ALP using 8086 Microprocessors
6. 8255-PPI

Books:

- "Computer System Architecture" by Morris Mano.
- " Computer Organization and Architecture" by William Stallings.
- "Computer Organization" by Carl Hamacher

Basic Structure of Computers

Unit-1

Contents:

Introduction:

Computer Organization

- It describes the function and design of the various units of digital computers that store and process information.

- It also deals with the units of computer that receive information from external sources and send computed results to external destinations.

1. Computer types

Digital computer :

It is a fast electronic calculating machine that accepts digitized input information, processes it according to a list of internally stored instructions, and produces the resulting output information.

(1) Personal computer :

- It is the most common form of desktop computers.
- Desk top computers have processing and storage units, visual display and audio output units, and a keyboard that can all be located easily on a home or office desk. The storage media include hard disks, CD-ROMs and diskettes.
- Portable notebook computers are a compact version of the personal computers with all of these components packaged into single unit the size of a thin briefcase

Computer types (contd.,)

(2) Workstations:

- Work stations with high resolution graphics input/output capability, although still retaining the dimensions of desktop computers, have significantly more computational power than personal computers.
- Workstations are often used in engineering applications, especially for interactive design works.

(3) Enterprise systems or mainframes:

- These are used for business data processing in medium to large corporations that require much more computing power and storage capacity than workstations can provide.

Computer types (contd.,)

4) Servers :

- Servers contain sizable database storage units and are capable of handling large volumes of requests to access the data.
- The Internet and its associated servers have become a dominant world wide source of all types of information.

5) Super Computers :

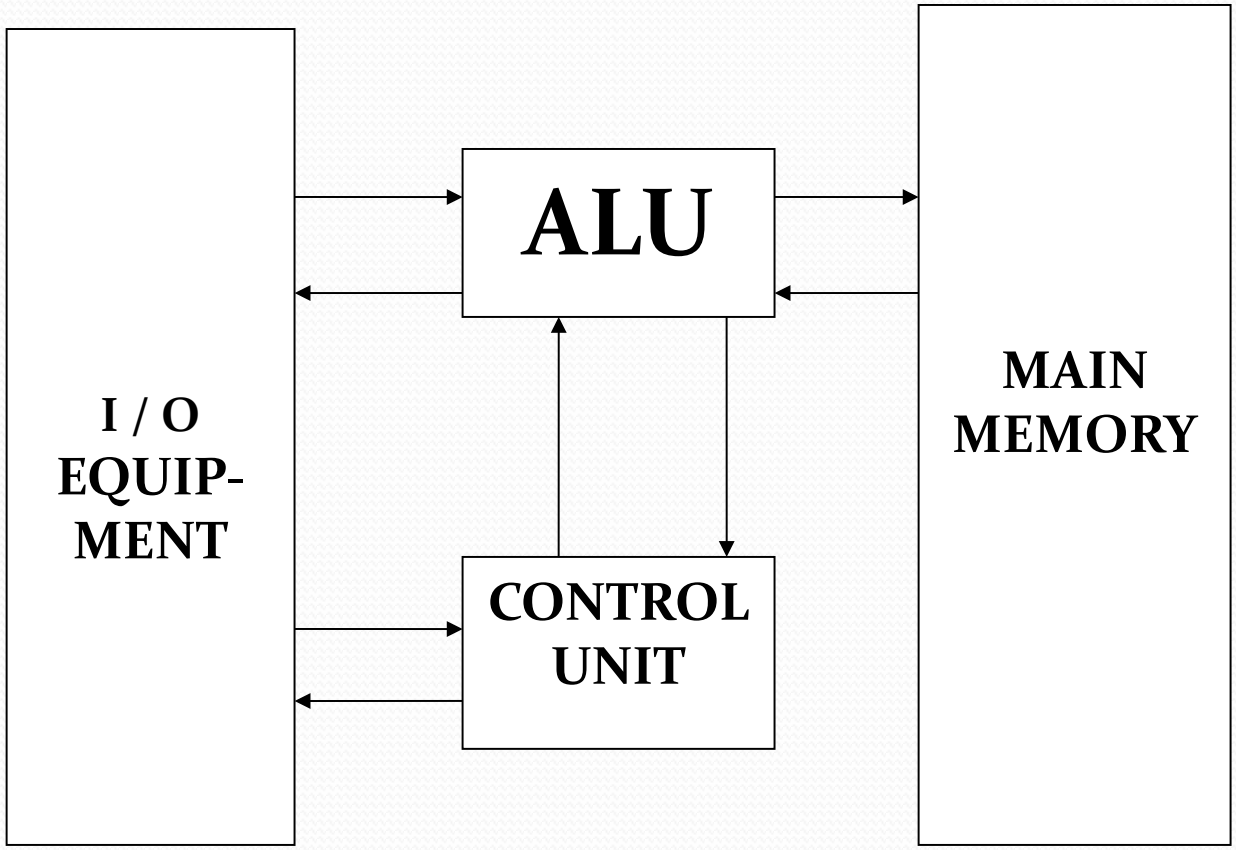
- Super Computers are used for the large scale numerical calculations required in applications such as weather forecasting, aircraft design and simulation.

2. Functional units

A computer consists of five functionally independent main parts:

- Input unit
- Memory unit
- Arithmetic and logic unit (ALU)
- Output unit
- Control unit

The structure of a digital computer is shown below



Operation of a computer

- The computer accepts information in the form of programs and data through an input unit and stores it in the memory.
- Information stored in the memory is fetched, under program control, into an arithmetic and logic unit, where it is processed.
- Processed information leaves through an output.
- All activities inside the machine are directed by the control unit.

Input unit

- Computers accept coded information through input units.
- The most well known input device is key board.
- Whenever a key is pressed, the corresponding letter or digit is automatically translated into its corresponding binary code and transmitted over a cable to either the memory or the processor.

Memory unit

The function of the memory is to store programs and data.

- There are two classes of storage, called primary and secondary.
- Primary storage is a fast memory that operates at electronic speeds.

Programs must stay in memory while they are being executed.

The memory contains a large number of semiconductor storage cells, each capable of storing one bit of information.

Random access memory: Memory in which any location can be reached in a short and fixed amount of time after specifying its address is called random access memory (RAM).

Memory unit (Contd.,)

- The time required to access one word is called the memory access time.
- Cache memory: The small, fast, RAM units are called caches.

  They are tightly coupled with the processor and are often contained on the same integrated circuit chip to achieve high performance.
- Main memory: The largest and slowest unit is referred to as the main memory.

Memory unit (Contd.,)

- Secondary storage:
- Although primary storage is essential, it tends to be expensive.
- Thus additional, cheaper, secondary storage is used when large amounts of data and many programs have to be stored, particularly for information that is accessed infrequently.
- A wide selection of secondary storage devices is available, including magnetic disks, tapes and optical disks (CD-ROMs)

Arithmetic and logic unit (ALU)

- Most computer operations are executed in the arithmetic and logic unit (ALU) of the processor.
- For example, Suppose two numbers are to be added.
- They are brought into the processor, and the actual addition is carried out by the ALU.
- The sum may then be stored in the memory or retained in the processor for immediate use.
- When operands are brought into the processor, they are stored in high-speed storage elements called registers.

Output unit

- The output unit is the counterpart of input unit.
- Its function is to send processed results to the outside world.
- The most familiar example of such a device is a printer.
- Some units, such as graphic displays, provide an input function and an output function. The dual role of such units is the reason for using the single name I/O unit in many cases.

Control unit

- The memory, arithmetic and logic, and I/O units store and process information and perform input and output operations.
- The operations of these units are coordinated by control unit.
- The control unit is effectively the nerve center that sends control signals to other units and senses their states.

3. Basic operational concepts

- To perform a given task, an appropriate program consisting of a list of instructions is stored in the memory.
- Individual instructions are brought from the memory into the processor, which execute the specified operations.
- Data to be used as operands are also stored in the memory

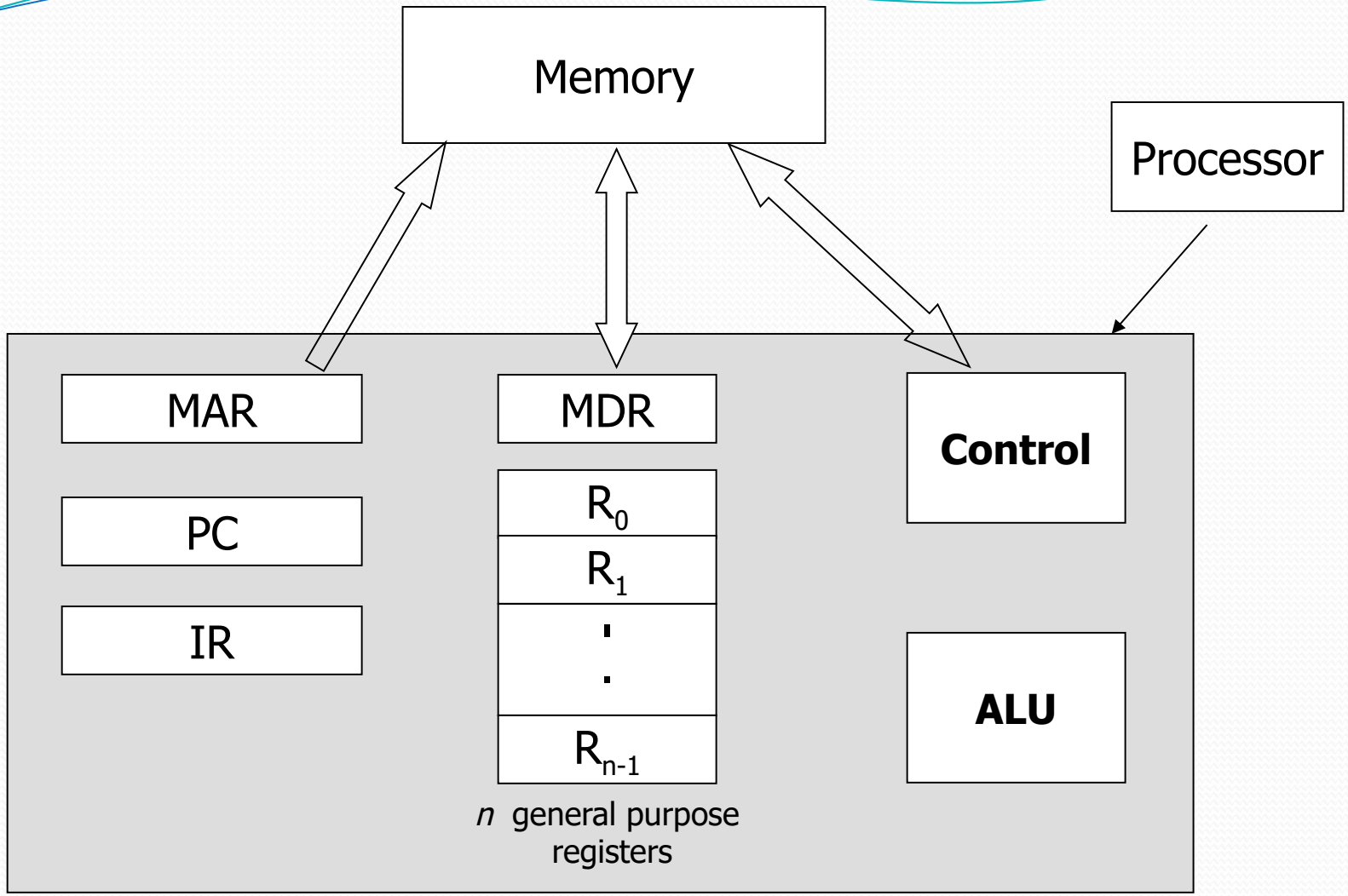
Basic operational concepts (Contd.,)

- Consider, the two instruction sequence

  Load LOCA, R1

  Add R1, Ro

- The first of these instructions transfers the contents of memory location LOCA into processor register R1, and second instruction adds the contents of register R1 as well as those of Ro and places the sum into Ro.
- Note that this destroys the former contents of R1 as well as Ro, where as the original contents of memory location LOCA are preserved.



Connections between the processor and the memory

Instruction register (IR)

- The instruction register holds the instruction that is currently being executed.
- Its output is available to the control circuits, which generate the timing signals that control the various processing elements involved in executing the instruction

Program Counter (PC)

- The program counter is another specialized register.
- It keeps track of the execution of a program.
- It contains the memory address of the next instruction to be fetched and executed.
- During the execution of an instruction, the contents of the PC are updated to correspond to the address of the next instruction to be executed.
- It is customary to say that PC points to the next instruction that is to be fetched from the memory.

Memory address register (MAR) & Memory data register (MDR)

These two registers facilitate communication with the memory.

- The MAR holds the address of the location to be accessed.
- The MDR contains the data to be written into or read out of the addressed location.

Operating steps for Program execution

- Programs are stored in the memory through the input unit.
- Execution of the program starts when the PC is set to point to the first instruction of the program.
- The contents of the PC are transferred to the MAR and a Read control signal is sent to the memory.
- After the time required to access the memory elapses, the addressed word (in this case, the first instruction of the program) is read out of the memory and loaded into the MDR.

Operating steps for Program execution (Contd.,)

- Next, the contents of the MDR are transferred to the IR .
- At this point, the instruction is ready to be decoded and executed.
- If the instruction involves an operation to be performed by the ALU, it is necessary to obtain the required operands.
- If an operand resides in memory (it could also be in a general-purpose register in the processor), it has to be fetched by sending its address to the MAR and initiating a Read cycle.

Operating steps for Program execution (Contd.,)

- When the operand has been read from the memory into the MDR, it is transferred from the MDR to the ALU.
- After one or more operands are fetched in this way, the ALU can perform the desired operation.
- If the result of the operation is to be stored in the memory, then the result is sent to the MDR.
- The address of the location where the result is to be stored is sent to the MAR, and a write cycle is initiated.

Operating steps for Program execution (Contd.,)

- At some point during the execution of the current instruction, the contents of the PC are incremented so that the PC points to the next instruction to be executed.
- Thus, as soon as the execution of the current instruction is completed, a new instruction fetch may be started.
- In addition to transferring data between the memory and the processor, the computer accepts data from input devices and sends data to output devices. Thus, some machine instructions with the ability to handle I/O transfers are provided.

Interrupt service routine

- Normal execution of a programs may be preempted if some device requires urgent servicing.
- For example, a monitoring device in a computer-controlled industrial process may detect a dangerous condition.

  In order to deal with the situation immediately, the normal execution of the current program must be interrupted.

  To do this, the device raises an interrupt signal.

- An interrupt is a request from an I/O device for service by the processor.

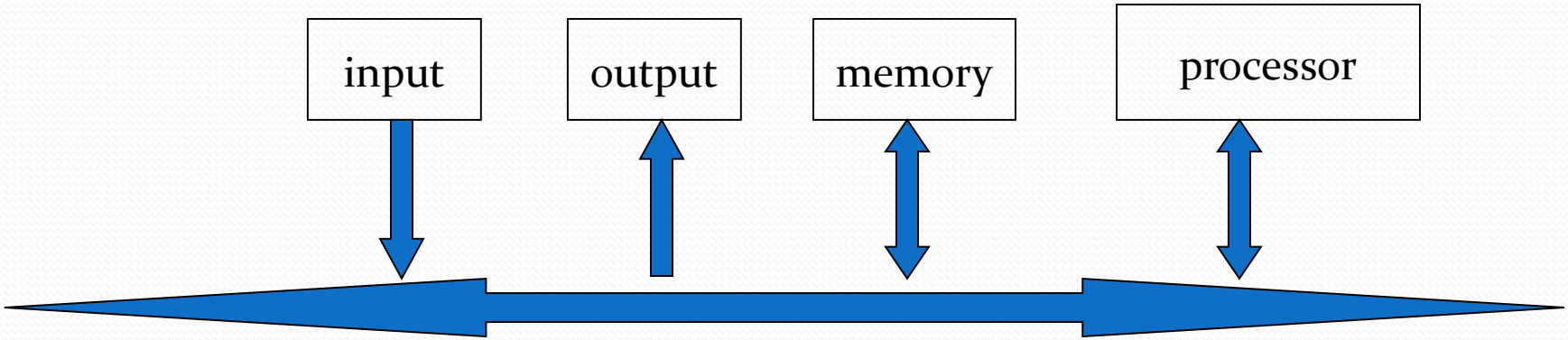
  The processor provides the requested service by executing an appropriate interrupt-service routine.

Interrupt service routine (contd.,)

- Because such diversions may alter the internal state of the processor, its state must be saved in the memory locations before servicing the interrupt.
- Normally, the contents of the PC, the general registers, and some control information are stored in memory.
- When the interrupt service routine is completed, the state of the processor is restored so that the interrupted program may continue.

4. Bus structures

- A group of lines(wires) that serves as a connecting path for several devices of a computer is called a bus.
- In addition to the lines that carry the data, the bus must have lines for address and control purposes.
- The simplest way to interconnect functional units is to use a single bus, as shown below.



Bus structures (contd.,)

- All units are connected to this bus. Because the bus can be used for only one transfer at a time, only two units can actively use the bus at any given time.
- Bus control lines are used to arbitrate multiple requests for use of the bus.
- The main virtue of the single-bus structure is its low cost and its flexibility for attaching peripheral devices.
- Systems that contain multiple buses achieve more concurrency in operations by allowing two or more transfers to be carried out at the same time. This leads to better performance but at an increased cost.

How timing differences are smoothed out among processors, memories and I/O devices?.

- The devices connected to a bus vary widely in their speed of operation.
- Electro mechanical devices such as key board and printers are relatively slow.
- Others like magnetic or optical disks, are considerably faster.
- Memory and processor units operate at electronic speeds.
- A common approach to smooth out the timing differences is to include buffer registers with the devices to hold the information during transfers.
- They prevent a high speed processor from being locked to a slow I/O device during a sequence of data transfers.

Contd.,

- This allows the processor to switch rapidly from one device to another, interweaving its process activity with data transfers involving several I/O devices.
- Thus, buffer registers smooth out timing differences among processors, memories and I/O devices.

5. Software

The role of system software in a computer.

- System software is responsible for the coordination of all activities in a computing system.
- System software is a collection of programs that are executed as needed to perform functions such as

  1) Receiving and interpreting user commands.

  2) Entering and editing application programs and storing them as files in secondary storage devices.

  3) Managing the storage and retrieval of files in secondary storage devices.

Contd.,

4) Running standard application programs such as word processors, spread sheets, or games, with data supplied by the user.

5) Controlling I/O units to receive input information and produce output results.

6) Translating programs from high level language to low level language.

7) Linking and running user-written application programs with existing standard library routines, such as numerical computation packages.

Various components of system software.

- **Compiler** : A system software program which translates the high-level language program into a suitable machine language program.

- **Text editor**: Another important system program that all programmers use is a text editor.
- It is used for entering and editing application programs.
- The user of this program interactively execute commands that allow statements of a source program entered at a keyboard to be accumulated in a file.

Contd.,

- **Operating system** (OS)
- It is a key system software component.
- This is a large program, or actually a collection of routines , that is used to control the sharing of and interaction among various computer units as they execute application programs.
- The OS routines perform the tasks required to assign computer resources to individual application programs.
- These tasks include assigning memory to program and data files, moving data between memory and disk units, and handling I/O operations.

6. Performance

Various parameters for improving the performance of a computer.

- The most important measure of the performance of a computer is how quickly it can execute a programs.
- The speed with which a computer executes programs is affected by the design of its hardware and its machine language instructions.

- **Elapsed time**: The total time required to execute a program .
- This elapsed time is a measure of the performance of the entire computer system.
- It is affected by the speed of the processor, the disk and the printer.

Contd.,

Processor time:

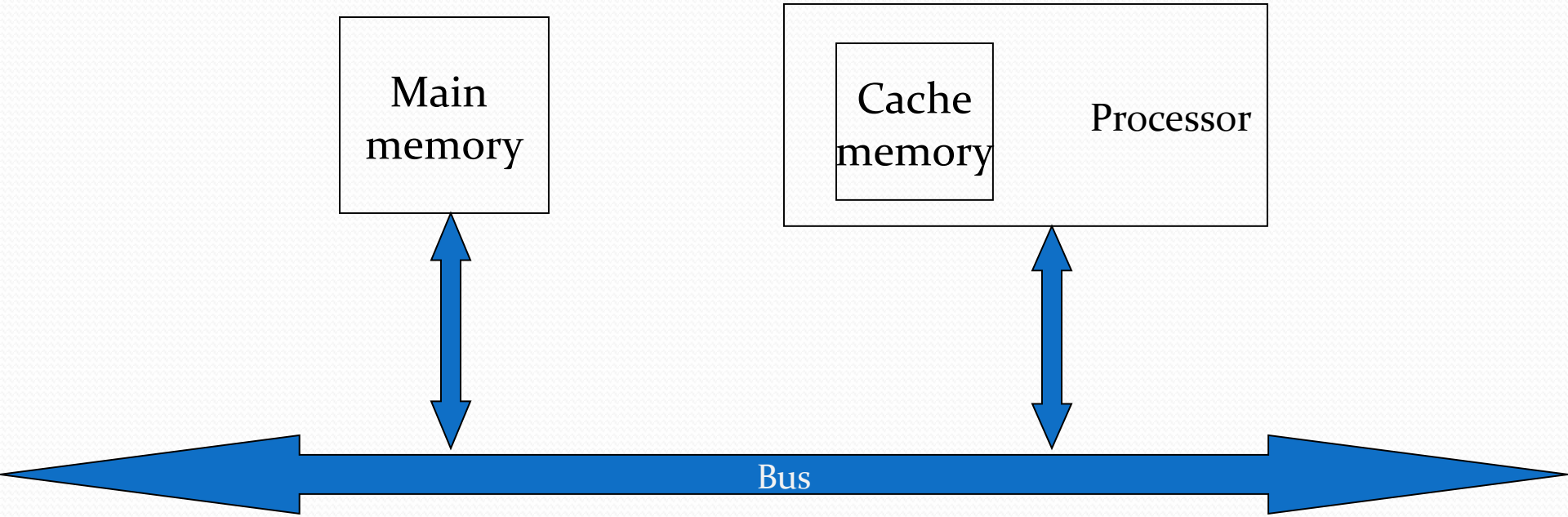
- Here we have to consider only those periods of the elapsed time, during which the processor is active.
- The sum of these periods is called processor time.
- The processor time depends on the hardware involved in the execution of individual machine instructions.

Cache memory:

- The processor and a relatively small cache memory can be fabricated on a single IC chip.
- The internal speed of performing the basic steps of instruction processing on such chips is very high and considerably faster than the speed at which instructions and data can be fetched from the main memory.

Contd.,

- A program will be executed faster if the movement of instructions and data between the main memory and processor is minimized, which is achieved by using the cache.



Processor clock

- Processor circuits are controlled by a timing signal called a clock .
- The clock defines regular time intervals, called clock cycles.
- To execute a machine instruction, the processor divides the action to be performed into a sequence of basic steps, such that each step can be completed in one clock cycle.
- The length P of one clock cycle is an important parameter that affects processor performance.
- Its inverse is the clock rate, R = 1/P , which is measured in cycles per second .
- If the clock rate is 500 million cycles per second, then the corresponding clock period is 2 nanoseconds.

Basic performance equation

- Let 'T' be the processor time required to execute a program that has been prepared in some high level language.
- Assume that the complete execution of the program requires the execution of 'N' machine language instructions.
- Suppose that the average number of basic steps needed to execute one machine instruction is 'S'.
- If the clock rate is 'R' cycles per second, the program execution time is given by
- $$T = \frac{N \cdot S}{R}$$
- This is often referred to as the basic performance equation.

Pipelining and Super scalar operation

- A substantial improvement in performance can be achieved by overlapping the execution of successive instructions, using a technique called pipelining .
- Consider the instruction Add R1, R2, R3
- Which adds the contents of registers R1 and R2, and places the sum into R3.
- The contents of R1 and R2 are first transferred to the inputs of the ALU.
- After the add operation is performed, the sum is transferred to R3.
- Processor can read the next instruction from the memory while the addition operation is being performed.

Pipelining (contd.,)

- Then, if that instruction also uses the ALU, its operands can be transferred to the ALU inputs at the same time that the result of Add instruction is being transferred to R3.
- Thus, pipelining increases the rate of executing instructions significantly and cause the effective value of S to approach 1.

Super scalar operation

- A higher degree of concurrency can be achieved if multiple instruction pipelines are implemented in the processor.
- This means that multiple function units are used, creating parallel paths through which different instructions can be executed in parallel.
- With such an arrangement, it becomes possible to start the execution of several instructions in every clock cycle.
- This mode of execution is called Super scalar operation.

Clock rate

- There are two possibilities for increasing the clock rate, R.
- First, improving the IC technology makes logic circuit faster, which reduces the time needed to complete a basic step. This allows the clock period, P, to be reduced and the clock rate, R, to be increased.
- Second, reducing the amount of processing done in one basic step also makes it possible to reduce the clock period , P.

Instruction set : CISC and RISC

- The terms CISC and RISC refer to design principles and techniques.
- RISC : Reduced instruction set computers
- Simple instructions require a small number of basic steps to execute.
- For a processor that has only simple instructions, a large number of instructions may be needed to perform a given programming task.
- This could lead to a large value of N and a small value for S.
- It is much easier to implement efficient pipelining in processors with simple instruction sets.

Instruction set : CISC and RISC (contd.,)

- CISC: Complex instruction set computers.
- Complex instructions involve a large number of steps.
- If individual instructions perform more complex operations, fewer instructions will be needed, leading to a lower value of N and a larger value of S.
- Complex instructions combined with pipelining would achieve good performance.

Compiler

- A compiler translates a high-level language program into a sequence of machine instructions.
- To reduce N, we need to have a suitable machine instruction set and a compiler that makes good use of it.
- An **optimizing compiler** takes advantage of various features of the target processor to reduce the product N.S .
- The compiler may rearrange program instructions to achieve better performance.

Performance measurement

- SPEC rating.
- A non profit organization called "System Performance Evaluation Corporation" (SPEC) selects and publishes representative application programs for different application domains.
- The SPEC rating is computed as follows
- SPEC rating = Running time on the reference computer
- Running time on the computer under test
- Thus SPEC rating of 50 means that the computer under test is 50 times faster than the reference computer for this particular benchmark.

Contd.,

- The test is repeated for all the programs in the SPEC suite, and the geometric mean of the results is computed.

- Let $SPEC_i$ be the rating for program 'i' in the suite.

  The overall SPEC rating for the computer is given by

$$\text{SPEC rating} = \left[\prod_{i=1}^{n} \left(SPEC_i \right) \right]^{1/n}$$

  where n is the number of programs in the suite.

7. Multiprocessors and Multi-computers

- Large computer systems may contain a number of processor units, in which case they are called multiprocessor systems.
- These systems either execute a number of different application tasks in parallel, or they execute subtasks of a single large task in parallel.
- All processors usually have access to all of the memory in such systems, and the term shared-memory multiprocessor systems is often used to make this clear.

Multiprocessors (contd.,)

- The high performance of these systems comes with much increased complexity and cost.
- In addition to multiple processors and memory units, cost is increased because of the need for more complex interconnection networks.

Multi-Computers:

- In contrast to multiprocessor systems, it is possible to use an interconnected group of complete computers to achieve high total computational power.
- The computers normally have access only to their own memory units. When the tasks they are executing need to communicate data, they do so by exchanging **messages** over a communication network.
- This property distinguishes them from shared-memory multiprocessors, leading to the name **message-passing multi-computers**.

8. Data Representation

Data Types

- Binary information is stored in ***memory*** or ***processor registers***
- Registers contain either ***data*** or ***control information***
  - ***Data*** are numbers and other binary-coded information
  - ***Control information*** is a bit or a group of bits used to specify the sequence of command signals
- Data types found in the registers of digital computers
  - ***Numbers*** used in arithmetic computations
  - ***Letters*** of the alphabet used in data processing
  - ***Other discrete symbols*** used for specific purpose
- Number Systems
  - ***Base*** or ***Radix r system*** : uses distinct symbols for ***r digits***
  - Most common number system :Decimal, Binary, Octal, Hexadecimal
  - Positional-value(weight) System : $\mathbf{r^2\ r\ ^1r^0.r^{-1}\ r^{-2}\ r^{-3}}$
    - Multiply each digit by an integer power of r and then form he sum of all weighted digits

- Decimal System/Base-10 System
  - Composed of 10 symbols or numerals(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0)
- Binary System/Base-2 System
  - Composed of 10 symbols or numerals(0, 1)
  - Bit = Binary digit
- Hexadecimal System/Base-16 System
  - Composed of 16 symbols or numerals(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)
- Binary-to-Decimal Conversions

$1011.101_2 = (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) + (1 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3})$

$= 8_{10} + 0 + 2_{10} + 1_{10} + 0.5_{10} + 0 + 0.125_{10}$

$= 11.625_{10}$

- Decimal-to-Binary Conversions

37 / 2 = 18 remainder 1 (binary number will end with 1) : **LSB**
18 / 2 = 9 remainder 0
9 / 2 = 4 remainder 1
4 / 2 = 2 remainder 0
2 / 2 = 1 remainder 0
1 / 2 = 0 remainder 1 (binary number will start with 1) : **MSB**

Read the result upward to give an answer of $37_{10} = 100101_2$

0.375 x 2 = 0.750 integer 0 **MSB**
0.750 x 2 = 1.500 integer 1
0.500 x 2 = 1.000 integer 1 **LSB**
Read the result downward $.375_{10} = .011_2$

Hex-to-Decimal Conversion

$2AF_{16} = (2 \times 16^2) + (10 \times 16^1) + (15 \times 16^0)$

$= 512_{10} + 160_{10} + 15_{10}$

$= 687_{10}$

Decimal-to-Hex Conversion

423_{10} / 16 = 26 remainder 7 (Hex number will end with 7) : **LSB**

26_{10} / 16 = 1 remainder 10

1_{10} / 16 = 0 remainder 1 (Hex number will start with 1) : **MSB**

Read the result upward to give an answer of $423_{10} = 1A7_{16}$

Hex	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Hex-to-Binary Conversion

$9F2_{16} =$ 9 F 2

↓ ↓ ↓

$= 1001 \quad 1111 \quad 0010$

$= 100111110010_2$

Binary-to-Hex Conversion

$1\,1\,1\,0\,1\,0\,0\,1\,1\,0_2 = 0\,0\,1\,1 \quad 1\,0\,1\,0 \quad 0\,1\,1\,0$

3 A 6

$= 3A6_{16}$

Binary-Coded-Decimal Code

- Each digit of a decimal number is represented by its binary equivalent

  8 7 4 **(Decimal)**

  ↓ ↓ ↓

  1000 0111 0100 **(BCD)**

- Only the four bit binary numbers from 0000 through 1001 are used
- Comparison of BCD and Binary

  $137_{10} = 10001001_2$ **(Binary) -** *require only 8 bits*

  $137_{10} = 0001\ 0011\ 0111_{BCD}$ **(BCD) -** *require 12 bits*

Alphanumeric Representation

- Alphanumeric character set
  - » 10 decimal digits, 26 letters, special character($, +, =,….)
  - » A complete list of ASCII : p. 384, Tab. 11-1, Morris Mano Text Book
- ASCII(American Standard Code for Information Interchange)
  - » Standard alphanumeric binary code uses seven bits to code 128 characters

Complements

- ***Complements*** are used in digital computers for simplifying the ***subtraction operation*** and for logical manipulation
- There are two types of complements for base r system
  - 1) r's complement 2) (r-1)'s complement
    - Binary number : 2's or 1's complement
    - Decimal number : 10's or 9's complement
- (r-1)'s Complement
  - (r-1)'s Complement of N = $(r^n-1)-N$
    - 9's complement of N=***546700***

      $(10^6-1)-546700= (1000000-1)-546700= 999999-546700$

      $= 453299$
    - 1's complement of N=***101101***

      $(2^6-1)-101101= (1000000-1)-101101= 111111-101101$

      $= 010010$
- r's Complement
  - r's Complement of N = r^n-N
    - 10's complement of ***2389***= 7610+1= ***7611***
    - 2's complement of ***1101100***= 0010011+1= ***0010100***

N : given number
r : base
n : digit number

546700(N) + 453299(9's com) =999999

101101(N) + 010010(1's com) =111111

* ***r's Complement***
(r-1)'s Complement +1 =$(r^n-1)-N+1= r^n-N$

- Subtraction of ~~Unsigned~~ Numbers

(M-N), N≠0

  - 1) M + (r^n-N)
  - 2) M ≥ N : Discard end carry, Result = M-N
  - 3) M < N : No end carry, Result = - r's complement of (N-M)

- ***Decimal Example)***

M ≥ N

72532(M) - 13250(N) = 59282

72532

\+ ~~*86750*~~ (10's complement of 13250)

Discard End Cary

1 59282

Result = ***59282***

M < N

13250(M) - 72532(N) = -59282

13250

\+ *27468* (10's complement of 72532)

No End Carry

0 40718

Result = -(10's complement of 40718)

= -(59281+1) = ***-59282***

- ***Binary Example)***

X ≥ Y

1010100(X) - 1000011(Y) = 0010001

1010100

\+ ~~*0111101*~~ (2's complement of 1000011)

1 0010001

Result = ***0010001***

X < Y

1000011(X) - 1010100(Y) = -0010001

1000011

\+ *0101100* (2's complement of 1010100)

0 1101111

Result = -(2's complement of 1101111)

= -(0010000+1) = ***-0010001***

9. Fixed-Point Representation

*Numeric Data
1) Fixed Point
2) Floating Point

- Computers must represent everything with 1's and 0's, including the sign of a number and fixed/floating point number
- Binary/Decimal Point
  - The position of the binary/decimal point is needed to represent ***fractions***, ***integers***, or ***mixed integer-fraction*** number

* 32.25
1) 0.25, 2) 32.0, 3) 32.25

- Two ways of specifying the position of the binary point in a register
  - 1) Fixed Point : the binary point is always fixed in one position
    - A binary point in the ***extreme left*** of the register(***Fraction*** : 0.xxxxx)
    - A binary point in the ***extreme right*** of the register(***Integer*** : xxxxx.0)
      - *The binary point is not actually present, but the number stored in the register is treated as a fraction or as an integer*
  - 2) Floating Point : the second register is used to designate the position of the binary point in the first register.
- Integer Representation
  - Signed-magnitude representation
  - Signed-1's complement representation
  - Signed-2's complement representation

Most Common

	+14	-14
→	0 0001110	1 0001110
→	0 0001110	1 1110001
→	0 0001110	1 1110010

* MSB for Sign
"0" is plus +
"1" is minus -

- Arithmetic Addition
  - Addition Rules of Ordinary Arithmetic
    - The signs are ***same*** : sign= ***common*** sign, result= ***add***
    - The signs are ***different*** : sign= ***larger*** sign, result= ***larger-smaller***
  - Addition Rules of the signed 2's complement
    - Add the two numbers including their sign bits
    - Discard any carry out of the sign bit position
- Arithmetic Subtraction
  - Subtraction is changed to an Addition
    - (± A) - (+ B) = (± A) + (- B)
    - (± A) - (- B) = (± A) + (+ B)

(-12) + (-13) = -25
(+12) + (+13) = +25

(+25) + (-37)
= 37 - 25 = -12

***Addition Example**

+ 6	00000110	- 6	11111010
+ 13	00001101	+ 13	00001101
+ 19	00010011	+ 7	00000111
+ 6	00000110	- 6	11111010
- 13	11110011	- 13	11110011
- 7	11111001	- 19	11101101

* **Subtraction Example** (- 6) - (- 13) = +7
11111010 - 11110011 = 11111010 + 2's comp of 11110011
= 11111010 + 00001101
= 100000111 = +7

Discard End Carry

- Overflow
  - Two numbers of n digits each are added and the sum occupies n+1 digits
  - n + 1 bit cannot be accommodated in a register with a standard length of n bits(*many computer detect the occurrence of an overflow, and a corresponding F/F is set*)

- Overflow
  - An overflow may occur if the two numbers added are both positive or both negative
    - When two unsigned numbers are added
      - an overflow is detected from the end carry out of the MSB position
    - When two signed numbers are added
      - the MSB always represents the sign

        - the sign bit is treated as part of the number

        - the end carry does not indicate an overflow

•Overflow Example

Carries: 0 1		carries: 1 0	
+ 70	0 1000110	- 70	1 0111010
+ 80	0 1010000	- 80	1 0110000
+ 150	1 0010110	- 150	0 1101010

Overflow Detection

- Detected by observing the ***carry into*** the sign bit position and the ***carry out*** of the sign bit position
- If these two carries are not equal, an overflow condition is produced(***Exclusive-OR gate = 1***)

***Decimal Example (+375) + (-240)**

375 + (10's comp of 240)= 375 +9 760

0 375	(0000 0011 0111 0101)
+9 760	(1001 0111 0110 0000)
0 135	(0000 0001 0011 0101)

Decimal Fixed-Point Representation

* Advantage * Computer I/O data are generated by people who use the decimal system

- A 4 bit decimal code requires four F/Fs for each decimal digit
- The representation of 4385 in BCD requires 16 F/Fs (0100 0011 1000 0101)
- The representation in decimal is *wasting a considerable amount of storage* space and the circuits required to perform decimal arithmetic are *more complex*

10. Floating-Point Representation

* Decimal + 6132.789

Fraction	*Exponent*
+0.6132789	+4

- The floating-point representation of a number has two parts
  - 1) Mantissa : signed, fixed-point number
  - 2) Exponent : position of binary(decimal) point
- Scientific notation : $m \times r^e$ $(+0.6132789 \times 10^{+4})$
  - ***m*** : mantissa, ***r*** : radix, ***e*** : exponent
- Example : $m \times 2^e = +(.1001110)_2 \times 2^{+4}$

Fraction	*Exponent*
01001110	000100

- Normalization
  - Most significant digit of mantissa is ***nonzero***

- Other Binary Codes
  - Gray Code

    changes by only one bit

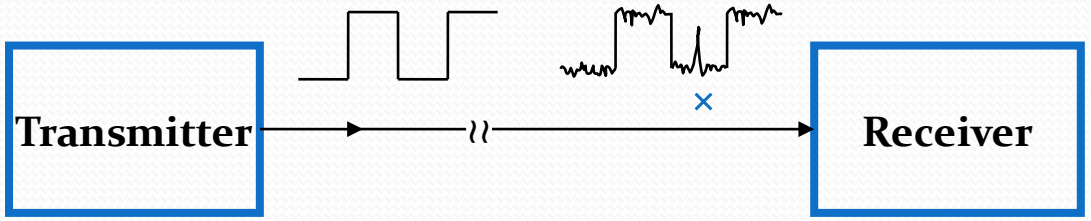
- Other Decimal Codes
  - Self-Complementing : excess-3 code
    - 9's complement of a decimal number is easily obtained complement(=changing 1's to 0's and 0's to 1's)
  - Weighted Code : 2421 code
    - The bits are multiplied by the weights, and the sum of the weighted bits gives the decimal digit
- Other Alphanumeric Codes
  - ASCII Code
  - The code consists of 128 characters. 95 characters represent graphic symbols that include upper and lower case letters, numerals zero to nine, punctuation marks and special symbols. 23 characters represent format effectors.
    - Format effector : Functional characters for controlling the layout of printing or display devices(carriage return-CR, line feed-LF, horizontal tab-HT,…)
    - The other 10 characters are for Data communication flow control(acknowledge-ACK, escape-ESC, synchronous-SYN,…)
  - EBCDIC(Extended BCD Interchange Code)
    - Used in IBM equipment

*** Self-Complement Example**

4_{10} = 0111 (3-excess)
= 1000 (1's comp)
= 5_{10} (3-excess in Tab. 3-6)
= 5_{10}(*9's comp of 4*)

11. Error Detection Codes

- Binary information transmitted through some form of communication medium is subject to external noise



- Parity Bit
  - An extra bit included with a binary message to make the total number of 1's either odd or even.
- Even-parity method
  - The value of the parity bit is chosen so that the total number of 1s *(including the parity bit)* is an ***even*** number

1 1 0 0 0 0 1 1

Added parity bit

- Odd-parity method
  - Exactly the same way except that the total number of 1s is an ***odd*** number

1 1 0 0 0 0 0 1

Added parity bit

- Parity Generator/Checker
  - At the sending end, the message is applied to a parity generator
  - At the receiving end, all the incoming bits are applied to a parity checker

1 1 0 0 0 0 1 1	**1 1 0 0 0 0 1 0**
(Even-parity Generator)	**(Even-parity Checker)**

  - Can not tell which bit in error
  - Can detect only single bit error(odd number of errors)

- Odd Parity Generator/Checker
  - Truth Table

A	B	C	D	E	O
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	0	1

End of Unit-1