IT201 COMPUTER ORGANIZATION AND ARCHITECTURE

1)BASIC STRUCTURE OF COMPUTERS: Basic Computer Organization – CPU Organization , Functional unit, Basic OPERATIONAL concepts, Bus structures,Performance,Data Representation. Fixed Point.Representation. Floating – Point Representation. Error Detection codes.

2) REGISTER TRANSFER LANGUAGE AND MICRO OPERATIONS : Register Transfer language. Register Transfer Bus and memory transfers, Arithmetic Micro operations, logic micro operations, shift micro operations, Arithmetic logic shift unit.

3) Basic Computer Organization and Design: Instruction codes. Computer Registers Computer instructions, Timing and Control, Instruction cycle. Memory Reference Instructions, Input – Output and Interrupt,

4) Micro Programmed Control: Control memory, Address sequencing, micro program example, design of control unit, micro Programmed control

5) Computer Processing Unit Organization: General Register Organization ,STACK organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation ,Program Control. CISC and RISC.

6) Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating – point Arithmetic operations. BCD Adder

7) Input-Output Organization: Peripheral Devices, Input-Output Interface, Asynchronous data transfer Modes of Transfer, Priority Interrupt Direct memory Access, Input –Output Processor (IOP) DMA controller

8) Pipeline And Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline,Dependencies, Vector Processing.

9) Memory Organization: Memory Hierarchy, Main Memory –RAM And ROM Chips, Memory Address map, Auxiliary memory-magnetic Disks, Magnetic tapes, Associate Memory,-Hardware Organization, Match Logic, Cache Memory –Associative Mapping , Direct Mapping, Set associative mapping ,Writing in to cache and cache Initialization , Cache Coherence ,Virtual memory-Address Space and memory Space ,Address mapping using pages, Associative memory page table ,page Replacement .

TEXTBOOKS

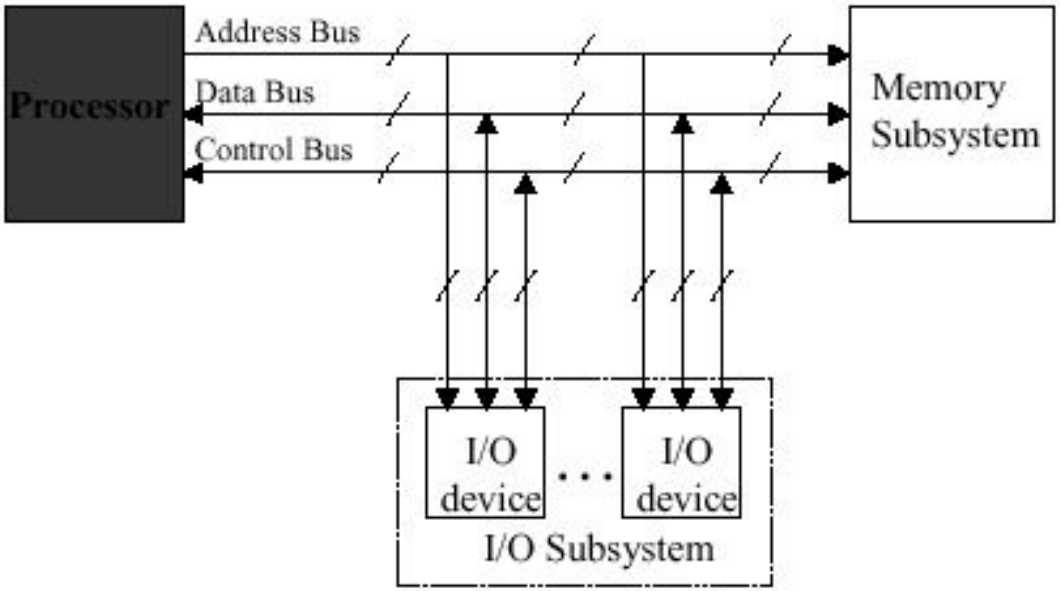
- Carl Hamacher et al., Computer Organization and Embedded Systems, Sixth Edition, McGraw-Hill, 2014.
- Vincent P Heuring, Harry F Jordan, T. G. Venkatesh, Computer Systems Design and Architecture, Pearson, 2008.
- Miles Murdocca and Vincent Heuring, Computer Architecture & Organization An Integrated Approach, Wiley, 2007.
- J. Hennesy and D. Patterson, Computer Architecture –A Quantitative Approach, 6th Ed., Morgan Kaufmann, 2017.
- Computer Organization" special edition-MRCET,TATA McGraw Hill. 2017
- Computer Systems Architecture – M.Moris Mano, IIIrd Edition, Pearson/PHI

REFERENCES:

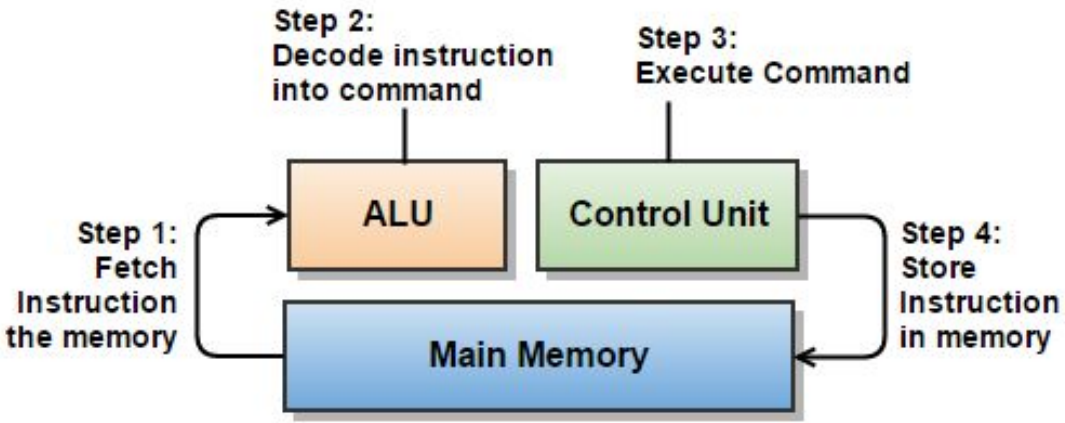
- Computer Organization and Architecture – William Stallings Sixth Edition, Pearson/PHI
- Structured Computer Organization – Andrew S. Tanenbaum, 4th Edition PHI/Pearson
- Fundamentals or Computer Organization and Design, - Sivaraama Dandamudi Springer Int. Edition.

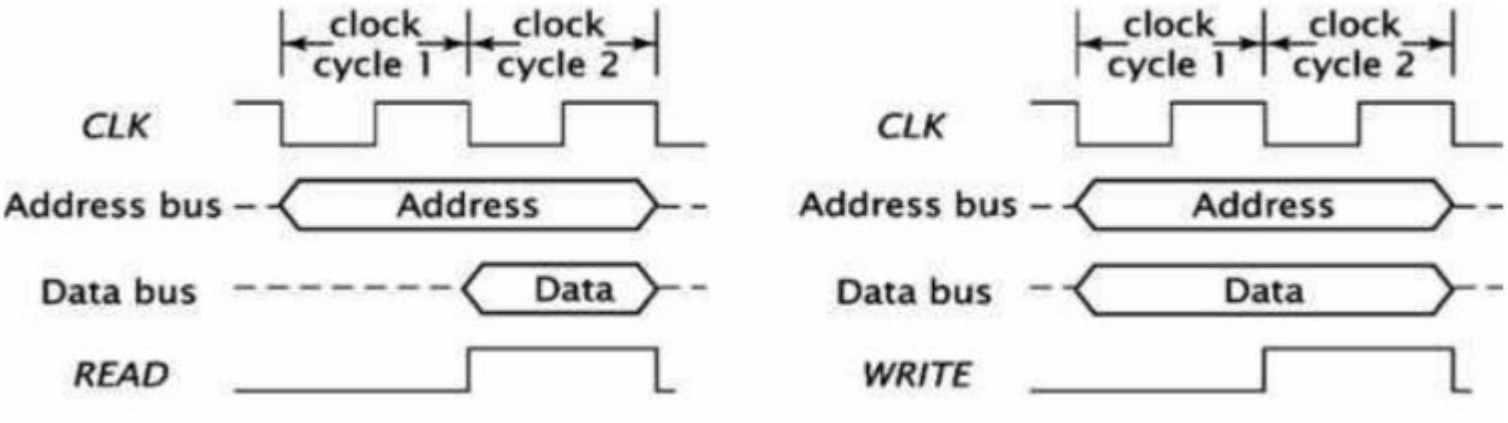
1) Quizzes : 20 Marks
2) Mid-sem : 30 Marks
3) End-sem : 50 Marks

VON NEUMANN ARCHITECTURE



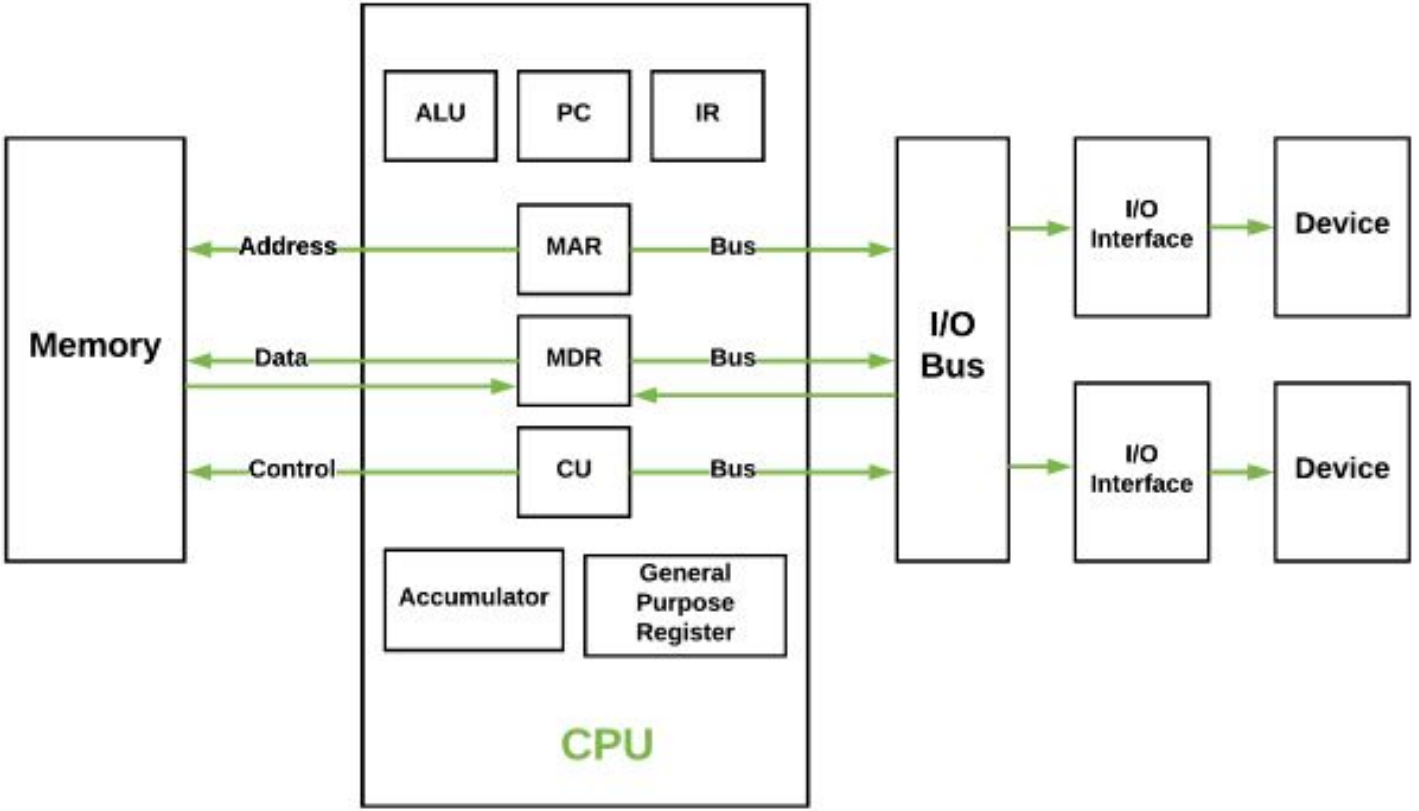
1. System bus: wires
2. Instruction cycles: fetch , decode and execute instructions





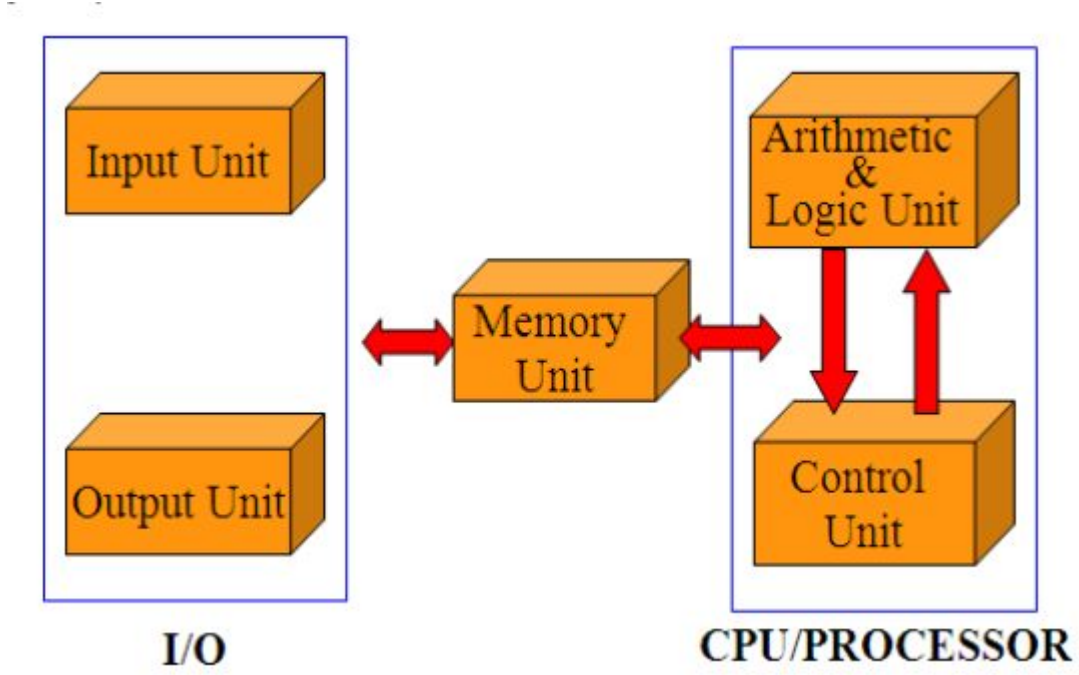
Timing diagram for read and write operations

CPU Organization



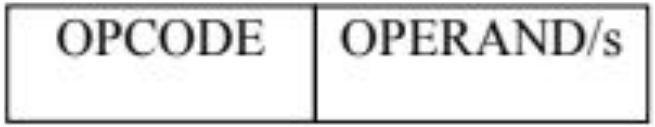
- Main Memory Unit (Registers) –
  - Accumulator
  - Program Counter (PC)
  - Memory Address Register (MAR)
  - Memory Data Register (MDR)
  - Current Instruction Register (CIR)
  - Instruction Buffer Register (IBR)
- Input/Output Devices
- Buses
  1. Data Bus
  2. Address Bus
  3. Control Bus

Functional Unit



Basic Operational Concepts

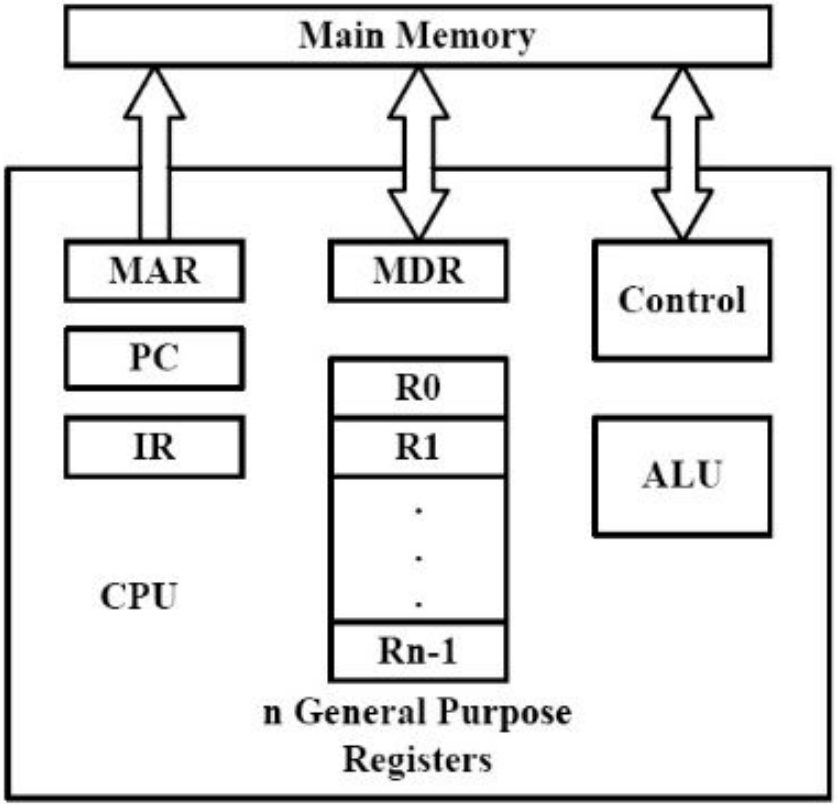
An Instruction consists of two parts, an Operation code and operand/s



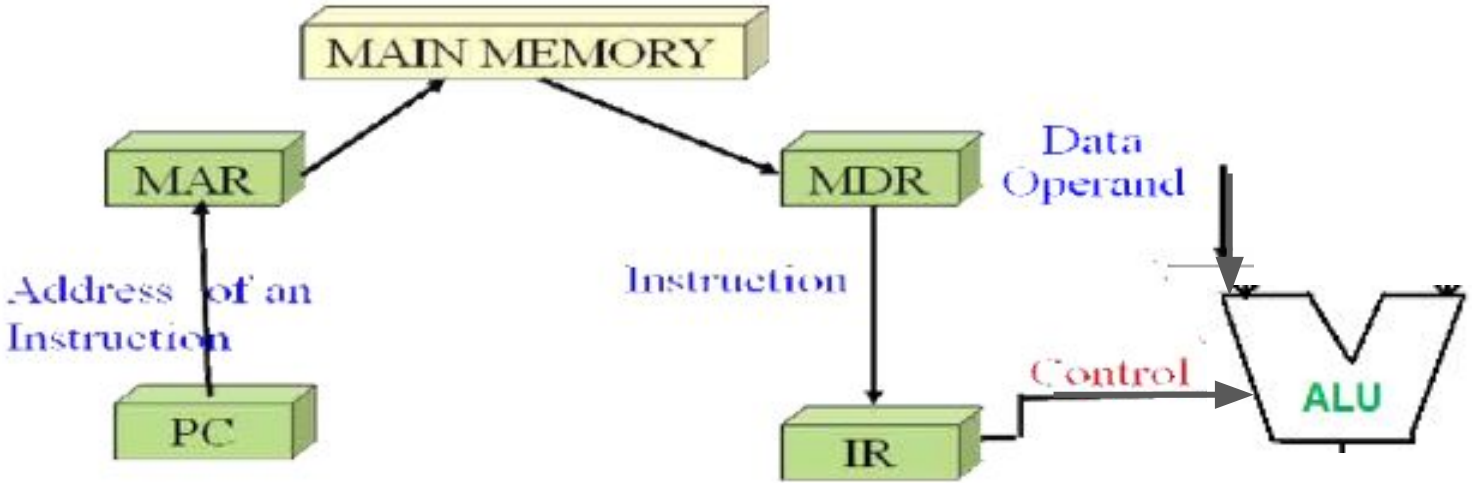
Eg 1) ADD LOCA, R0

Operation mnemonic	Name	Number of bits transferred	Description
L	Load	32	Transfer from memory in register
LH	Store half-word	16	Transler from memory to register
ST	store	32	Transfer from register to memory
STH	Store half-word	16	Transfer from register to memory

Connections between the processor and the memory

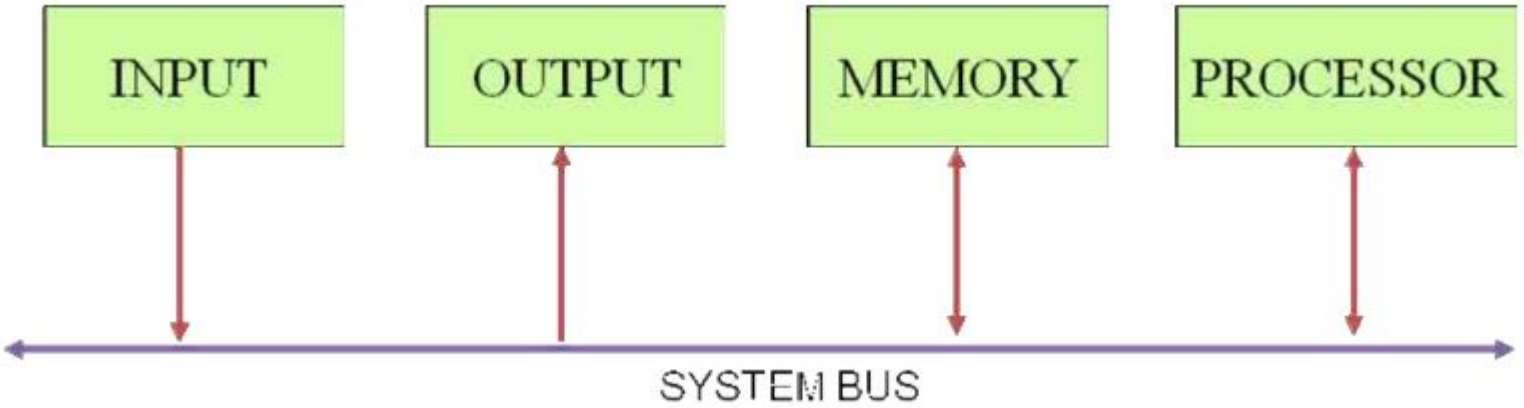


Interaction between the memory and the ALU



BUS STRUCTURES

Group of lines that serve as connecting path for several devices is called a bus



USER PROGRAM and OS ROUTINE INTERACTION

Assume computer with 1 processor, 1 disk and 1 printer and application program is in machine code on disk.

t0: the OS loads the program from the disk to

memory

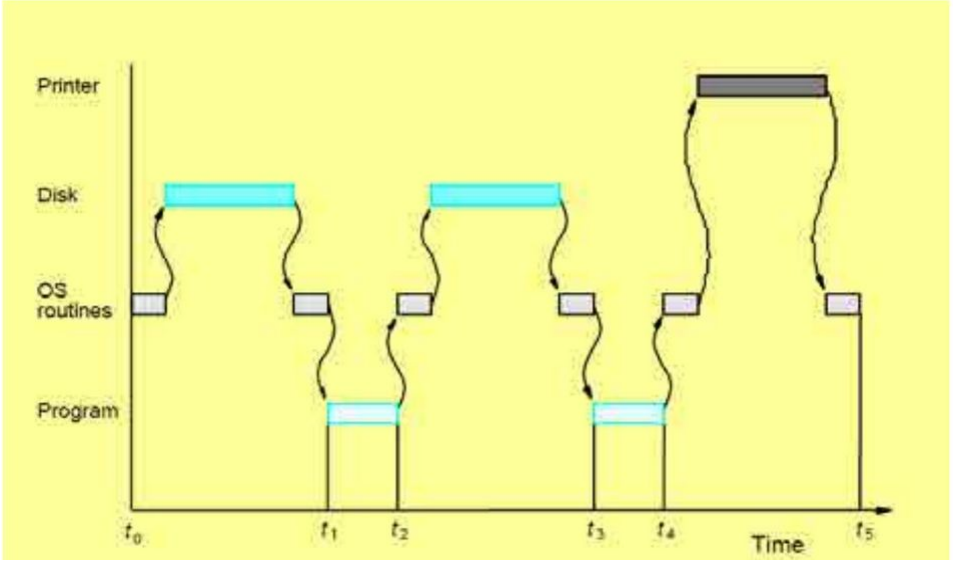
t1: program executes

t2: program accesses disk

t3: program executes

t4: program accesses printer

t5: program terminates



GENERATION OF COMPUTERS

First generation: 1946 to 1955: Computers of this generation used **Vacuum Tubes**. The computers were built using stored program concept. Ex: ENIAC, EDSAC, IBM 701. Computers of this age typically used about ten thousand vacuum tubes. They **were bulky in size had slow operating speed, short life time and limited programming facilities.**

Second generation: 1955 to 1965: Computers of this generation used the **germanium transistors** as the active switching electronic device. Ex: IBM 7000, B5000, IBM 1401. Comparatively **smaller in size About ten times faster operating speed** as compared to first generation vacuum tube based computers. Consumed **less power, had fairly good reliability.** Availability of large memory was an added advantage.

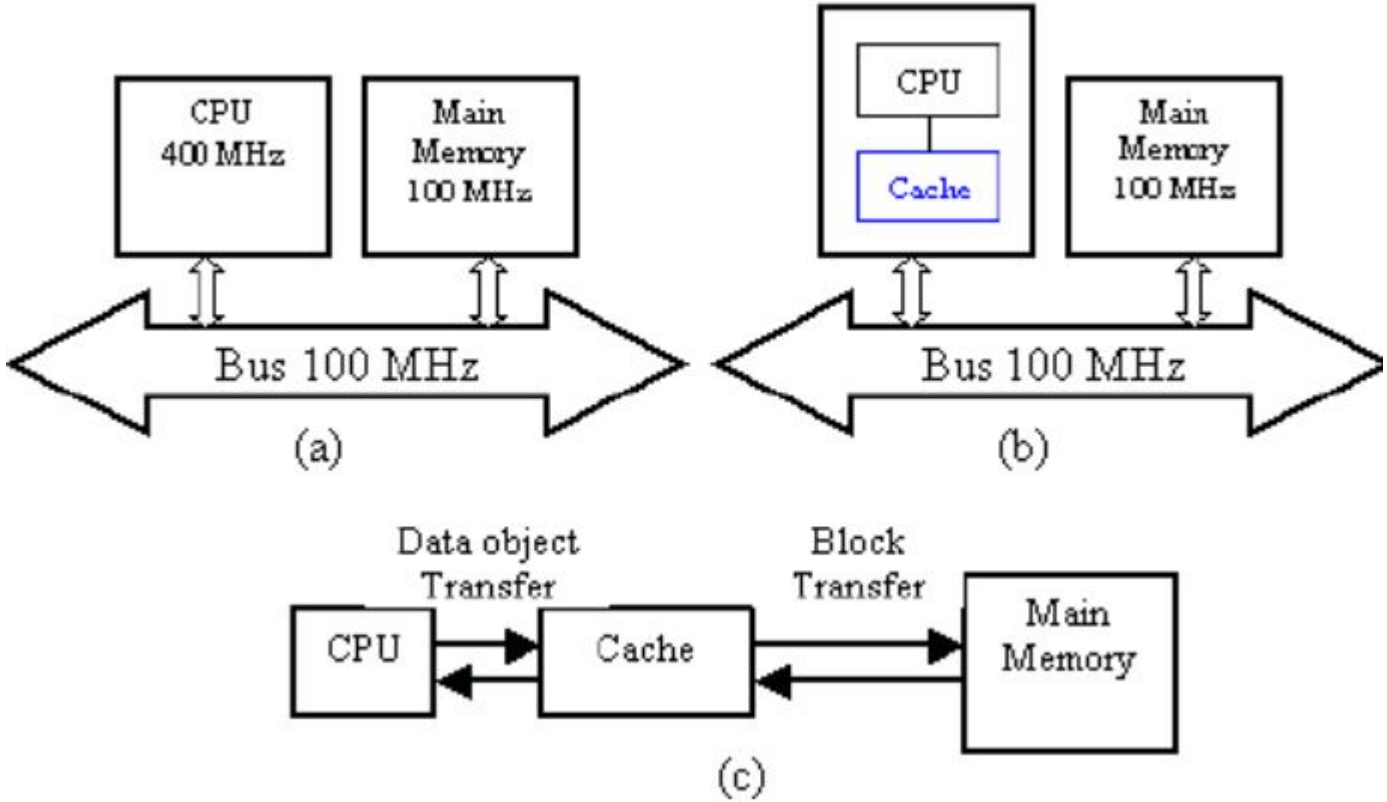
Third generation: 1965 to 1975: The computers of this generation used the **Integrated Circuits** as the active electronic components. Ex: IBM system 360, PDP minicomputer etc. They were still s**maller in size.** They had powerful CPUs with the capacity of executing 1 million instructions per second (MIPS). Used to consume **very less power consumption.**

Fourth generation: 1976 to 1990: The computers of this generation used the **LSI chips** like microprocessor as their active electronic element. HCL horizen III, and WIPRO‟S Uniplus+ HCL‟s Busybee PC etc. **They used high speed microprocessor as CPU.** They were more user friendly and highly reliable systems. They had large storage capacity disk memories.

Beyond Fourth Generation: 1990 onwards: Specialized and dedicated **VLSI chips** are used to control specific functions of these computers. Modern Desktop PC‟s, Laptops or Notebook Computers.

PERFORMANCE

- The total time required to execute the program is elapsed time is a measure of the performance of the entire computer system.
- The time needed to execute a instruction is called the processor time.



a) CPU, main memory, and bus; (b) Processor cache; (c) Data transfer between CPU and cache and between cache and main memory

Processor clock

- Processor circuits are controlled by a timing signal called clock.
- The clock designs at the regular time intervals called clock cycles.

Performance equation

$$T=N*S/R$$

- T be the processor time required to execute a program
- N is the number of instruction
- S average number of basic steps executed in one clock cycle.
- R cycles per second

Data Representation

Information that a Computer is dealing with is Data

Numeric Data: Numbers(Integer, real)

Non-numeric Data: Letters, Symbols

Relationship between data elements:Data Structures -Linear Lists, Trees, Rings, e

Program(Instruction)

Numeric Data Representation

Fixed Point

Decimal	Binary	Octal	Hexadecimal
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

1. Sign and Magnitude Representation: The **most significant** (leftmost) bit in the word as a sign bit. If the sign bit is 0, the number is positive; **if the sign bit is 1, the number is negative.**

2. One's Complement (1's) Representation: Forming the 1s complement of a given number is equivalent to **subtracting that number from 2n -1** Eg.1s complement of 0101 is 1010

3.Two"s Complement (2"s) Representation: Forming the 2s complement of a number is done by subtracting that number from 2n**.** So 2s complement of a number is obtained by adding 1 to 1s complement of that number. Eg: 2"s complement of 0101 is 1010 +1 = 1011

In all systems,the leftmost bit is 0 for positive number and 1 for negative number.

Floating-point representation

Floating-point numbers are so called as the decimal or binary point floats over the base depending on the exponent value. It consists two components.

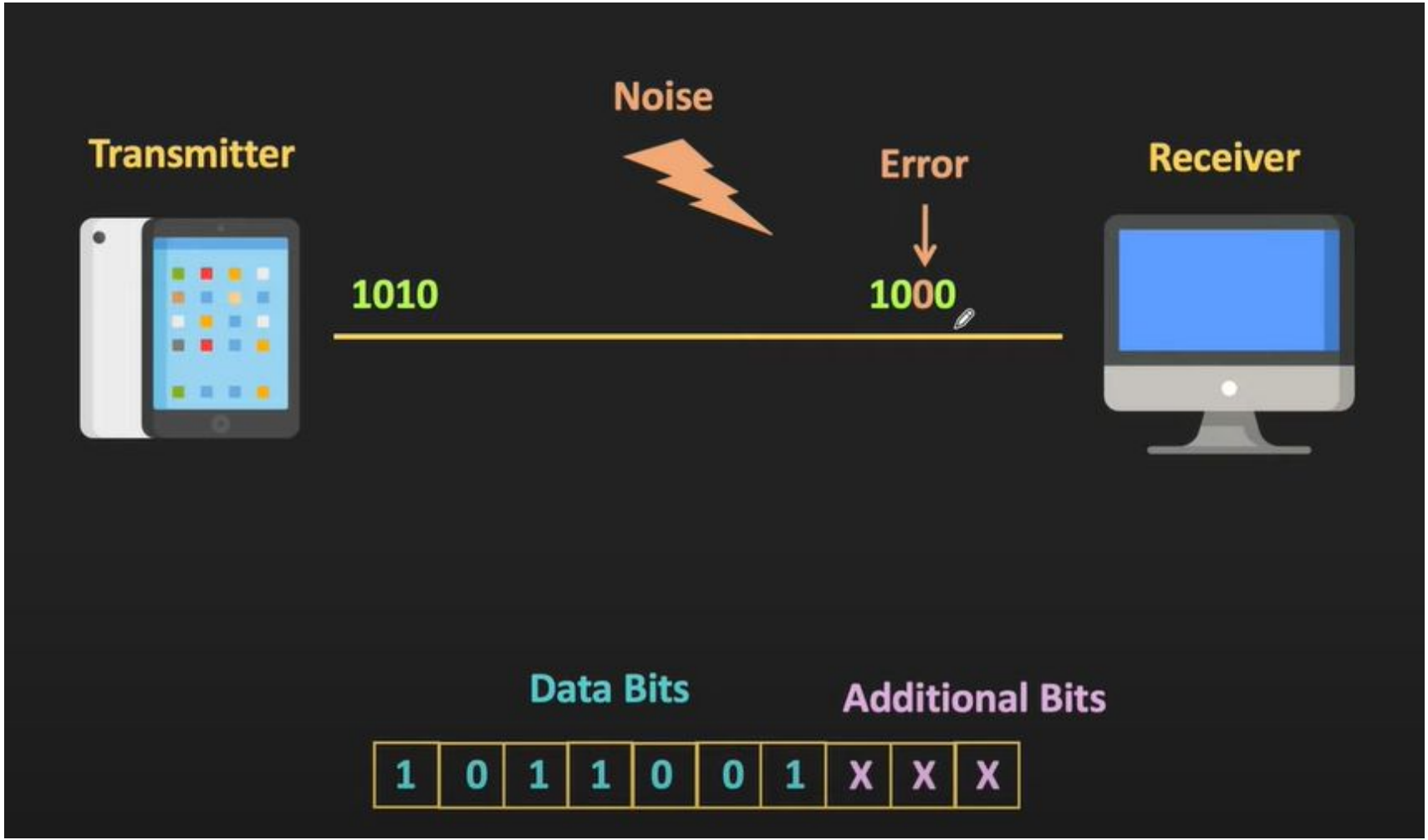
- Exponent
- Mantissa

Error Detection

- The corrupted bits leads to spurious data being received by the receiver and are called errors.
- Error – detecting codes ensures messages to be encoded before they are sent over noisy channels. The encoding is done in a manner so that the decoder at the receiving end can detect whether there are errors in the incoming signal with high probability of success.

Features of Error Detecting Codes

- Reliable data transmission: Acknowledgment message
- Block codes: Bits are added
- Error has occurred or not.: The number of error bits and the type of error does not matter.

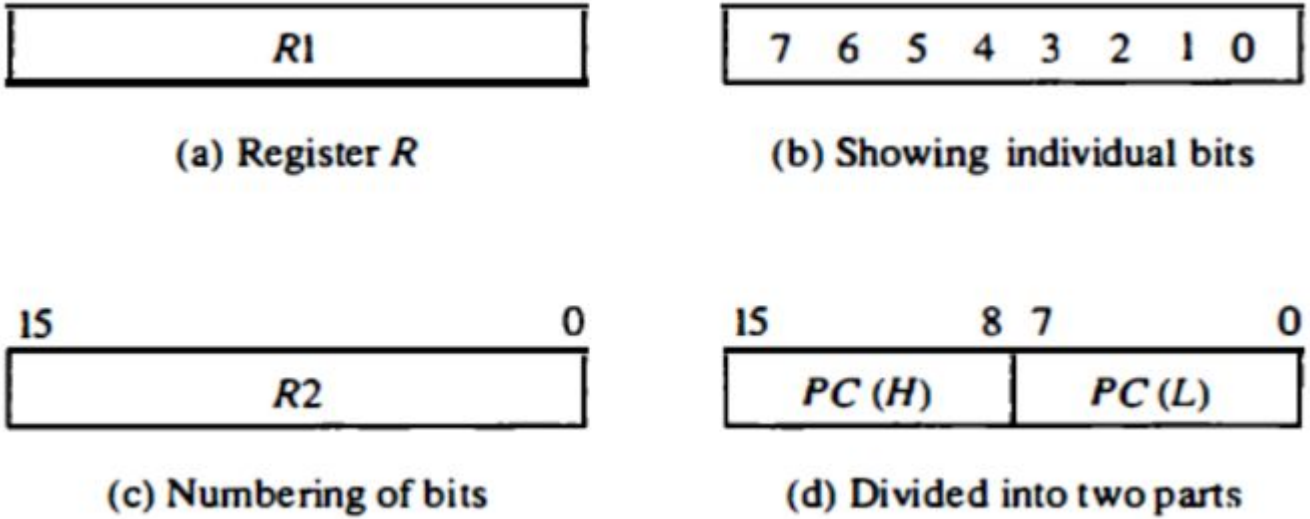

Noise
Transmitter
Error
Receiver
1010
1000
Data Bits
Additional Bits
1 0 1 1 0 0 1 X X X

1. Parity Code
2. Two-dimensional Parity check
3. Hamming Code
4. Checksum
5. Cyclic redundancy check

Register Transfer and Microoperations

Register Transfer Language

- Digital modules: constructed by digital components as registers, decoders, arithmetic elements, and control logic.
- The operations executed on data stored in registers are called microoperations eg shift, count,clear, and load.
- Register transfer language: symbolic notation for microoperation transfers among registers



(a) **Represent a register is by a rectangular box with the name of the register inside,**
(b) **Individual bits can be distinguished**
(c) **The numbering of bits in a 16-bit register can be marked on top of the box**
(d) **A 16-bit register is partitioned into two parts in Bits 0 through 7 are assigned the symbol L (for low byte) and bits 8 through 15 are assigned the symbol H (for high byte).** The name of the 16-bit register is PC. The symbol PC (0-7) or PC(L) refers to the low-order byte and PC(8-15) or PC(H) to the high-order byte.