Fundamental Of Computer Organization

UNIT 1 — Number System + Logic Gates

PART A — Number System Basics

What is Number System?

A number system is a way to **represent numbers using symbols**.

Types of Number Systems

System	Base	Digits Used
Decimal	10	0–9
Binary	2	0,1
Octal	8	0–7
Hexadecimal	16	0–9, A–F

Conversion

Binary → Decimal

Multiply each bit with 2^n and add.

Example:
$(1011)_2 = 1\times8 + 0\times4 + 1\times2 + 1\times1 = \mathbf{11}$

Decimal → Binary

Divide by 2 → write remainders in reverse.

Example:
$13 \rightarrow 1101_2$

Binary → Octal

Group bits in 3

Example:
$101110 \rightarrow 101 \mid 110 = \mathbf{56_8}$

Binary → Hexadecimal

Group bits in 4

Example:
1110 0111 = **$E7_{16}$**

PART B — Binary Arithmetic

Binary Addition

Rules
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 (carry 1)

Example:
1011+0101
= 10000

PART C — Basic Logic Gates

AND Gate

Output is **1 only if both inputs are 1**

Truth Table:

A	B	A·B
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

Output is **1 if any input is 1**

Truth Table:

A	B	A+B
0	0	0
0	1	1
1	0	1

1	1	1

NOT Gate

Inverts input

$0 \rightarrow 1$
$1 \rightarrow 0$

NAND Gate

Opposite of AND

Symbol: (AB)’

NOR Gate

Opposite of OR

Symbol: (A+B)’

XOR Gate

Output **1 when inputs are different**

XNOR Gate

Output **1 when inputs are same**

Now we do practice (from zero again)

Q1) Convert $(11010)_2$ to decimal

Options:
(a) 18
(b) 20
(c) 26
(d) 24

Q2) Convert $(10101)_2 \rightarrow$ Decimal

Options:
(a) 21
(b) 19
(c) 15
(d) 17

Q3) Convert $(10011)_2$ → Decimal

Options:
(a) 17
(b) 19
(c) 21
(d) 23

(Decimal → Binary)

Q4) Convert 18 decimal → binary

Options:

(a) 10010
(b) 10101
(c) 11001
(d) 10011

Question 1:

For **AND Gate**, output kab 1 hota hai?

(a) When both inputs are 1
(b) When any one input is 1

Question 2:

For **OR Gate**, output kab 1 hota hai?

(a) Only when both inputs are 1
(b) When at least one input is 1

Question 3:

For **XOR Gate**, output kab 1 hota hai?

(a) When inputs are same
(b) When inputs are different

Question 4:

For **XNOR Gate**, output kab 1 hota hai?

(a) When inputs are same
(b) When inputs are different

UNIT-2

1) Types of Computers

Micro Computers — PCs, laptops
Mini Computers — small organisations
Mainframe — banking systems
Super Computers — high speed scientific tasks

2) Functional Units of a Computer

Input Unit
Output Unit
Memory Unit
ALU (Arithmetic Logic Unit)
Control Unit

➡ Together these form the **CPU working model**

3) Basic Operational Concepts

Computer follows **Input → Processing → Output → Storage** cycle

4) Bus Structure

Bus = set of wires for communication

Data Bus
Address Bus
Control Bus

5) Performance Factors

Computer performance depends on:

CPU speed
Memory speed
Cache size
Bus width
I/O speed

6) Floating vs Fixed Point Representation

Fixed point — integer representation
Floating point — decimal (fraction) numbers

7) Error Detection Codes

Used to check whether transmitted data is correct.

Example: **Parity Bit**

Q1) CPU ke main parts kaun se hote hain?

(a) Input and Output
(b) Control Unit and ALU
(c) Monitor and Keyboard
(d) Printer and Scanner

Q2) Which bus carries data?

(a) Address Bus
(b) Control Bus
(c) Data Bus
(d) Power Bus

Q3) Which computer is used for scientific research and weather forecasting?

(a) Micro computer
(b) Mini computer
(c) Super computer
(d) Mainframe

Q4) ALU ka full form?

Arithmetic Logic Unit

Q5) Control Unit kya karta hai?

Instruction ko decode karta hai aur execution control karta hai

Q6) Bus kya hota hai?

Communication path between components

Q7) Floating point numbers kis type ke hote hai?

Fractional / decimal numbers

UNIT-3 — Register Transfer Language & Micro-Operations

1) What is a Register?

A register is a **small, high-speed storage location inside CPU** used to hold data temporarily.

Registers store data while CPU is executing instructions.

Examples:

- PC (Program Counter)
- MAR (Memory Address Register)
- IR (Instruction Register)
- AC (Accumulator)

2) What is Register Transfer Language (RTL)?

RTL is a notation used to **describe how data moves between registers**.

It is written in a symbolic way.

Example:

R1 ← R2

Meaning: value of **R2 is transferred to R1**

➡ This is called **Register Transfer Operation**

3) BUS — data path between registers

To move data between registers, CPU uses **bus structure**.

A **bus** is a set of wires that carries data between registers, memory and CPU.

4) Micro-Operations

Micro-operations are **elementary operations performed on data stored in registers.**

Unit-3 syllabus clearly mentions 3 types:

- Arithmetic
- Logic
- Shift Operations

Let’s learn them 👇

A) Arithmetic Micro-operations

Used to perform arithmetic on register contents.

Examples:

1. Addition:

R1 ← R1 + R2

1. Subtraction:

R1 ← R1 - R2

1. Increment:

R1 ← R1 + 1

1. Decrement:

R1 ← R1 - 1

👉 These operations are done inside **ALU**

B) Logic Micro-operations

Used for bit wise logical manipulation.

Examples:

R1 ← R1 AND R2

R1 ← R1 OR R2

R1 ← NOT R1

R1 ← R1 XOR R2

These operations help in **decision-making and data manipulation**.

C) Shift Micro-operations

Used to shift data bits left or right.

Examples:

R1 ← SHL R1 (Shift Left)

R1 ← SHR R1 (Shift Right)

➡ Useful in multiplication, division, encryption etc.

ALSU — Arithmetic Logic Shift Unit

This is a CPU component that performs:

Arithmetic operations
Logic operations
Shift operations

ALSU combines **ALU + shifter** so processor executes micro-operations efficiently.

Instruction Cycle — How CPU executes instructions

Based on Unit-3 syllabus: "Instruction cycle"

CPU executes instructions in 3 major steps:

1) Fetch

• Control Unit takes instruction from memory
• Stores it in Instruction Register (IR)

2) Decode

• Instruction decoded to understand operation
• Addressing mode and operand located

3) Execute

• ALU performs operation
• Result stored back to register or memory

This cycle runs continuously while computer is ON.

Instruction Format

Instruction has fields like:

Opcode | Operand | Address

Opcode — tells operation
Operand — data or address
Mode bits — addressing style

Addressing Modes (important for exam)

Gets asked frequently

✔ Immediate Addressing
Operands directly present in instruction

✔ Direct Addressing
Instruction contains **memory address**

✔ Indirect Addressing
Instruction points to a memory location which stores actual address

✔ Register Addressing
Operand stored in register

✔ Register Indirect
Register holds address of operand

✔ Indexed Addressing
Effective address = base + index

These modes help instruction access operands efficiently.

Quick Practice

Q1) What does RTL notation R1 ← R2 mean?

(a) R2 gets value of R1
(b) R1 gets value of R2
(c) Exchanging values

Next Question

Q2) What is a micro-operation?

(a) A large operation performed outside CPU
(b) A small internal operation performed on register data
(c) An internet operation

Arithmetic Logic Shift Unit (ALSU) — Explanation

CPU me ek special block hota hai jo:

Arithmetic operations karta hai
Logic operations karta hai
Shift operations karta hai

Isi block ko **Arithmetic Logic Shift Unit** kehte hain.

➡ Ye unit register data par operation perform karke result return karta hai.

Diagram idea (exam me draw kar sakta hai):

Registers → ALSU → Result

Instruction Cycle — Detailed Explanation

CPU executes instructions in **three major stages**:

1) Fetch Cycle

• Control Unit memory se instruction laati hai
• Instruction IR (Instruction Register) me store hoti hai

2) Decode Cycle

• CU instruction ko decode karti hai
• Operation aur addressing mode samajhti hai

3) Execute Cycle

• ALU/memory/register operation perform hota hai
• Result store hota hai

➡ Then next instruction fetch hota hai — *cycle continues forever*.

Instruction Format — Explanation

Instruction ke andar normally 2 parts hote hain:

Opcode | Operand

Opcode → which operation (add, move, load, store etc.)
Operand → data or address required

Addressing Modes — Detailed Set (Exam Favourite)

Addressing mode batata hai **operand kaha milta hai?**

Here are common modes:

Immediate Addressing

Operand directly inside instruction
Example: MOV A, #5

Direct Addressing

Instruction contains address of data
Example: LOAD A, 1000

Indirect Addressing

Instruction contains address of a memory location
which stores actual address.

Register Addressing

Operand stored in register
Example: ADD A, B

Register Indirect Addressing

Register contains memory address where operand exists

Indexed Addressing

Effective address = Base + Index

➡ Fast execution ke liye modes select kiye jaate hain.

Q3) CPU executes instruction in sequence — what is this called?

(a) Boot cycle
(b) Instruction cycle
(c) Memory cycle

Register Transfer Bus & Memory Transfer

👉 CPU me registers ek dusre se data exchange kar sakte hain
👉 Memory se register me data transfer hota hai

Example RTL:

R1 ← Memory[1000]

Memory[2000] ← R1

✔ First one loads data from memory into register
✔ Second one stores register value into memory

➡ Ye process **Address Bus + Data Bus + Control signals** se hota hai

Instruction Format Types

Instructions ke kuch structure types:

✔ **Single Address Instruction**
Opcode + Address

✔ **Two Address Instruction**
Opcode + Src + Destination

✔ **Three Address Instruction**
Opcode + Src1 + Src2 + Destination

➡ Program complexity aur hardware ke hisaab se formats select hote hain.

📌 **Computer Registers — Types**

Some important CPU registers:

✔ **PC (Program Counter)**

Stores **address of next instruction**

✔ **IR (Instruction Register)**

Stores **current instruction**

✔ **MAR (Memory Address Register)**

Carries **address sent to memory**

✔ **MDR / MBR (Memory Data Register)**

Carries **data sent to or received from memory**

✔ **AC (Accumulator)**

Used to store intermediate results during calculation

➡ Ye exam me 5 marks ke short note me aata hai.

Input–Output and Interrupt

✔ I/O devices CPU se communicate karte hain

✔ Interrupt system CPU ko notify karta hai important events ka

Example: keyboard input, timer event, device request

➡ CPU interrupts handle karta hai and returns back to instruction cycle.

Addressing Modes summary table for exam:

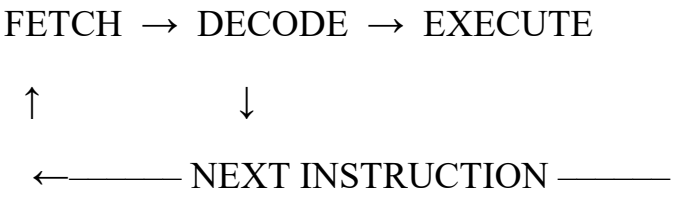
Mode	Where operand is found?
Immediate	Inside instruction
Direct	Memory address in instruction
Indirect	Memory holds actual address
Register	Stored in register
Register Indirect	Register holds memory address
Indexed	Base + index

Q4) Which register stores the address of next instruction?

(a) IR
(b) PC
(c) MAR

Instruction Cycle Diagram (Exam Drawing)

Aise diagram draw kar sakta hai 👇



UNIT – 4: COMPUTER ARITHMETIC

Q1) Explain Binary Addition and Subtraction rules.

Binary Addition Rules

Operation	Result
0 + 0	0
0 + 1	1
1 + 0	1
1 + 1	0 (carry 1)

Example:

1011 + 0101

10000

Binary Subtraction Rules

Operation	Result
0 − 0	0
1 − 0	1
1 − 1	0
0 − 1	1 (borrow 1)

Example:

1010 - 0011

0111

Q2) Explain Binary Multiplication Algorithm.

Binary multiplication works like decimal multiplication — but shifted additions.

Rules:

- 0 × anything = 0
- 1 × anything = same number

Example:

Multiply 110 × 101

```
   110
 × 101
-------------
   110      (1 × 110)
  000       (0 × 110 shifted left)
+ 110        (1 × 110 shifted twice)
-------------
 11110
```

Final Result: 11110_2

➡ Multiplication in processor performed by **shift + add operations**.

Q3) Explain Binary Division Algorithm.

Binary division same as decimal — repeated subtraction + shifting.

✔ **Example:**

1101 ÷ 10 (13 ÷ 2)

Steps:

- Compare
- Subtract
- Bring next bit

Final quotient = 110
Remainder = 1

➡ Processor performs division using **subtract + shift + restore method**

Q4) Explain Floating-Point Representation and Arithmetic.

Floating-point representation stores **real/fractional numbers**.

Representation Form:

Number = Mantissa × Base^Exponent

Example:

1.101×2^3

Advantages:

Wide range
Supports fractional values

Floating-Point Operations:

Addition, subtraction, multiplication and division using exponent adjustment.

Q5) Explain Decimal Arithmetic Unit.

Processors also perform arithmetic in **decimal form**, especially for financial data.

Decimal arithmetic unit handles:

Decimal addition
Decimal subtraction
Decimal multiplication
Decimal division

It often uses BCD (Binary-coded decimal) form for operations.

UNIT-4 Important Theory Concepts (Exam Writing)

1) Binary Adder

A binary adder is a digital circuit that adds binary numbers.

Types:

- **Half Adder** → adds two bits
- **Full Adder** → adds 3 bits (A, B, Carry)

2) Half Adder

Inputs: A, B
Outputs: Sum (A XOR B), Carry (A AND B)

3) Full Adder

Inputs: A, B, Cin
Outputs:

Sum = A XOR B XOR Cin
Carry = AB + Cin(A XOR B)

4) Subtractor Circuits

Half subtractor & full subtractor work like adders but perform subtraction.

Used in processor arithmetic units.

UNIT-5 — THE MEMORY SYSTEM

1) What is computer memory?

Memory is a storage unit used to:

store data
store instructions
store execution results

It works closely with CPU to support program execution.

Ek computer me different types ki memory hoti hai
jo **speed, size aur cost** ke hisaab se arranged hoti hai.

2) Memory Hierarchy

Memory ko **fastest se slowest** order me arrange kiya jata hai:

Register (Fastest)

Cache

RAM (Main memory)

Hard disk / SSD (Secondary memory)

External storage (Slowest)

Jitni fast memory hoti hai utni hi costly hoti hai.

Hierarchy ka purpose: speed aur capacity ka balance maintain karna.

3) Primary Memory

✔ RAM — Random Access Memory

• Volatile (power off ho jaaye to data erase ho jata hai)
• Fast read/write
• CPU executes programs directly from RAM

Types of RAM:

1. **SRAM (Static RAM)**
   – Very fast
   – Used in cache memory
2. **DRAM (Dynamic RAM)**
   – Cheaper
   – Used in main memory

ROM — Read Only Memory

• Non-volatile (power off hone ke baad bhi data safe)
• Instructions permanently stored

Example:
BIOS / firmware

Types of ROM:
• PROM — programmed once
• EPROM — erased using UV light
• EEPROM — erased electrically

4) Cache Memory

Cache CPU aur RAM ke beech me hota hai.
Purpose = CPU ko fast data supply karna.

Features:

✔ Very fast
✔ Small capacity
✔ Stores frequently used instructions

Cache Levels:

- L1 (fastest, smallest)
- L2 (faster, bigger)
- L3 (slowest among cache, biggest)

➡ Cache hit = data available in cache → fast
➡ Cache miss = data fetched from RAM → slower

5) Secondary Storage

Long-term storage for keeping data permanently.

Examples:

Hard Disk Drive (HDD)
Solid State Drive (SSD)
Pen drive
Memory card

Secondary storage **non-volatile** hota hai.

6) Virtual Memory

Virtual memory ek technique hai jo **RAM ko extend karta hai**
storage device (hard disk / SSD) ka use karke.

When RAM becomes full,
program ka kuch data HDD me move ho jata hai.

OS page file / swap area use karta hai.

This gives effect of **larger memory** than actual RAM.

7) Performance Considerations

Memory speed CPU performance ko effect karti hai.

Cache build kiya jata hai isliye
CPU ko har baar RAM ka wait nahi karna padta.

More cache / faster RAM → better system performance.

8) RAID — Redundant Array of Independent Disks

RAID ek **storage technology** hai jisme multiple disks combine hote hain.

Purpose:

Better performance
Better reliability
Fault tolerance

Common RAID levels:

• RAID 0 — striping, high speed but no safety
• RAID 1 — mirroring, data copied on two disks
• RAID 5 — striping + parity (speed + safety balance)

RAID servers, banking systems, large databases me use hota hai.

9) Summary Table — Types of Memory

Memory Type	Speed	Capacity	Volatility	Usage
Register	Fastest	Very low	Volatile	CPU internal
Cache	Very fast	Low	Volatile	Frequent data
RAM	Medium	High	Volatile	Program execution
ROM	Medium	Low	Non-volatile	Firmware
HDD/SSD	Slow	Very high	Non-volatile	Storage
Virtual Memory	Slow	Depends	Software-based	Extend RAM