UNIT -02 COA DETAILED NOTES WITH EASY EXPLAINATION AND DIAGRAMS

Addition and Subtraction

> ⭐⭐⭐⭐⭐ Most Important Topic
>
> Addition aur Subtraction Computer Arithmetic ka basic aur frequently asked topic hai.
>
> Is topic se numericals bhi aate hain aur theory questions bhi.
>
> Ye answer **3–4 pages** aasani se cover karega.

Addition and Subtraction

Introduction

Computer sirf binary numbers (0 aur 1) ko samajhta hai.

Isliye computer ke andar saare arithmetic operations binary form me perform kiye jate hain.

Addition aur Subtraction sabse basic arithmetic operations hain jinko ALU (Arithmetic Logic Unit) perform karta hai.

Definition

Binary Addition

"Binary Addition is the process of adding two binary numbers according to binary arithmetic rules."

Binary Subtraction

"Binary Subtraction is the process of subtracting one binary number from another using binary arithmetic rules."

Need of Addition and Subtraction

Addition aur Subtraction ka use:

- Arithmetic calculations
- Data processing
- Scientific computations
- Computer programming
- Digital systems

me hota hai.

Binary Number System Review

Binary system me sirf do digits hoti hain:

```
0 and 1
```

Base = 2

Rules of Binary Addition

Binary Addition ke basic rules:

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Important Rule

```
1 + 1 = 10
```

Sum = 0

Carry = 1

Example 1: Binary Addition

Add:

```
1010
+
0101
```

Solution:

```
  1010
+ 0101
-------
  1111
```

Verification:

```
1010 = 10

0101 = 5
```

Result:

```
1111 = 15
```

Therefore:

```
10 + 5 = 15
```

Correct.

Example 2: Binary Addition with Carry

Add:

```
1111
+
0001
```

Solution:

```
  1111
+ 0001
-------
10000
```

Decimal Check:

```
15 + 1 = 16
```

Correct.

Hardware for Binary Addition

Addition operation ke liye:

Half Adder

Full Adder

use kiye jate hain.

Half Adder

Definition

Half Adder ek combinational circuit hai jo do binary bits ko add karta hai.

Block Diagram

```
 A ----\
        \
         > Half Adder
        /
 B ----/

Outputs:

Sum
Carry
```

Truth Table

A	B	Sum	Carry

0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Full Adder

Definition

Full Adder teen inputs ko add karta hai.

```
A

B

Carry-In
```

Block Diagram

```
 A ----\
        \
 B ----- > Full Adder
        /
Cin ----/

Outputs:

Sum
Carry
```

Truth Table

A	B	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Binary Subtraction

Binary subtraction bhi specific rules follow karti hai.

Rules of Binary Subtraction

A	B	Result
0	0	0
1	0	1
1	1	0
0	1	1 (Borrow Required)

Borrow Concept

```
0 - 1
```

Possible nahi hai.

Isliye left bit se borrow liya jata hai.

Example 1: Binary Subtraction

Subtract:

```
1010
-
0011
```

Solution:

```
  1010
- 0011
--------
  0111
```

Verification:

```
10 - 3 = 7
```

Correct.

Example 2: Subtraction Using Borrow

```
1000
-
0001
```

Solution:

```
  1000
- 0001
--------
  0111
```

Decimal:

```
8 - 1 = 7
```

Correct.

Subtraction Using 2's Complement

Computer generally subtraction ko direct perform nahi karta.

Subtraction ko addition me convert kar deta hai.

Formula:

```
A - B

=

A + (2's Complement of B)
```

Example

```
7 - 3
```

Binary:

```
7 = 0111

3 = 0011
```

2's Complement of 3:

```
0011

↓

1100 + 1

↓

1101
```

Now:

```
0111
+
1101
-------
10100
```

Discard Carry:

```
0100
```

Answer:

```
4
```

Correct.

Addition and Subtraction Circuit

```
        A
        |
        |
        V

    +--------+
    |  ALU   |
    +--------+
```

```
        ^
        |
        |
        B

        |
        V

      Result
```

ALU addition aur subtraction dono perform karta hai.

Advantages of Binary Addition and Subtraction

1. Simple Hardware

Digital circuits me implementation easy hai.

---

2. High Speed

Fast calculations possible.

---

3. Reliable

Error chances kam.

---

4. Suitable for Computers

Binary system computer ke liye natural hai.

Disadvantages

1. Long Binary Numbers

Large numbers difficult lag sakte hain.

---

2. Borrow and Carry Complexity

Manual calculations me complexity badh sakti hai.

Applications

ALU Operations

CPU calculations.

Scientific Computing

Complex computations.

Embedded Systems

Microcontrollers.

Digital Signal Processing

DSP systems.

Difference Between Addition and Subtraction

Addition	Subtraction
Numbers Add Hote Hain	Numbers Subtract Hote Hain

Carry Generate Hota Hai	Borrow Generate Hota Hai
Easier Operation	Comparatively Complex
Example: 5+3	Example: 5−3

Viva Questions

Q1. What is Binary Addition?

Adding two binary numbers.

Q2. What is Binary Subtraction?

Subtracting one binary number from another.

Q3. What is Carry?

Extra bit generated during addition.

Q4. What is Borrow?

Bit borrowed during subtraction.

Q5. Which unit performs addition and subtraction?

ALU.

Frequently Asked RGPV Questions

2 Marks

1. Define Binary Addition.
2. Define Binary Subtraction.
3. What is Carry?
4. What is Borrow?

---

5 Marks

1. Explain Binary Addition.
2. Explain Binary Subtraction.
3. Explain Half Adder.

---

7 Marks

1. Explain Binary Addition with examples.
2. Explain Binary Subtraction with examples.
3. Explain Full Adder.

---

14 Marks

1. Explain Binary Addition and Subtraction with suitable examples.
2. Discuss Half Adder and Full Adder with truth tables.
3. Explain subtraction using 2's complement method.

Expected 2026 Questions

🔥 Explain Binary Addition and Binary Subtraction with examples.

🔥 Explain Half Adder and Full Adder.

🔥 Explain subtraction using 2's complement.

🔥 Perform binary addition and subtraction numerically.

One-Minute Revision

✅ Binary Addition:

```
1 + 1 = 10
```

✅ Binary Subtraction:

```
0 - 1 = Borrow
```

✅ Half Adder = 2 Inputs

✅ Full Adder = 3 Inputs

✅ ALU performs arithmetic operations

✅ Subtraction:

```
A - B

=

A + 2's Complement(B)
```

Conclusion

Addition and Subtraction computer arithmetic ke fundamental operations hain. Inhe binary number system ke rules ke according perform kiya jata hai. ALU in operations ko execute karta hai aur Half Adder, Full Adder aur 2's Complement techniques ki madad se fast aur efficient calculations perform ki jaati hain. Ye Computer Organization ka bahut important aur exam-oriented topic hai. 🎯

2's Complement Representation

> ⭐⭐⭐⭐⭐ Most Important Topic
>
> 2's Complement Representation Computer Arithmetic ka backbone hai.
>
> RGPV me theory + numerical dono puchhe jate hain.
>
> Ye answer **3–4 pages** aasani se cover karega.

2's Complement Representation

Introduction

Computer positive aur negative numbers ko binary form me represent karta hai.

Negative numbers ko represent karne ke liye sabse common technique **2's Complement Representation** hai.

Computer arithmetic me subtraction aur signed arithmetic operations ko perform karne ke liye 2's Complement ka use kiya jata hai.

Definition

"2's Complement is a method of representing negative binary numbers in which 1 is added to the 1's complement of a binary number."

Why 2's Complement is Used?

1. Negative Numbers Representation

Negative numbers ko easily represent karta hai.

2. Simplifies Arithmetic

Subtraction ko addition me convert kar deta hai.

3. Single Hardware

Same adder circuit addition aur subtraction dono perform kar sakta hai.

4. Efficient Processing

Computer arithmetic fast ho jati hai.

Binary Complements

Binary arithmetic me do types ke complements use hote hain:

1. 1's Complement

2. 2's Complement

1's Complement

Binary number ke har bit ko invert kar diya jata hai.

Rule

```
0 → 1

1 → 0
```

Example

Binary Number:

```
1010
```

1's Complement:

```
0101
```

2's Complement

2's Complement nikalne ke liye:

Step 1

1's Complement nikalo.

Step 2

Usme 1 add karo.

Formula

```
2's Complement

=

1's Complement + 1
```

Example 1

Find 2's Complement of:

```
1010
```

Step 1: 1's Complement

```
1010

↓

0101
```

Step 2: Add 1

```
0101
+
0001
------
0110
```

Answer:

```
0110
```

Example 2

Find 2's Complement of:

```
1100
```

Step 1

```
1100

↓

0011
```

Step 2

```
0011
+
0001
------
0100
```

Answer:

```
0100
```

Shortcut Method

Rule

Right side se first 1 tak bits same rakho.

Baaki sab bits invert kar do.

Example

```
101100
```

Rightmost 1 tak same:

```
100
```

Remaining invert:

```
010
```

Answer:

```
010100
```

Representation of Positive and Negative Numbers

Positive Number

MSB = 0

Example:

```
+5

00000101
```

Negative Number

MSB = 1

Example:

```
-5
```

First:

```
5 = 00000101
```

1's Complement:

```
11111010
```

2's Complement:

```
11111011
```

Therefore:

```
-5 = 11111011
```

Range of Numbers in 8-bit 2's Complement

Formula:

```
-(2^(n-1))

to

(2^(n-1)-1)
```

For 8-bit:

```
-128 to +127
```

Representation Table

Decimal	Binary (8-bit)
+5	00000101
-5	11111011
+10	00001010
-10	11110110

Subtraction Using 2's Complement

Most Important

Formula

```
A - B

=

A + 2's Complement(B)
```

Example

Find:

```
7 - 3
```

Step 1

Convert to binary

```
7 = 0111

3 = 0011
```

Step 2

Find 2's Complement of 3

```
0011

↓

1100

↓

1101
```

Step 3

Add

```
0111
+
1101
------
10100
```

Discard Carry

```
0100
```

Answer:

```
4
```

Correct.

Example 2

Find:

```
10 - 6
```

Binary

```
10 = 1010

6 = 0110
```

2's Complement of 6

```
0110

↓
```

```
1001

↓

1010
```

Add

```
1010
+
1010
------
10100
```

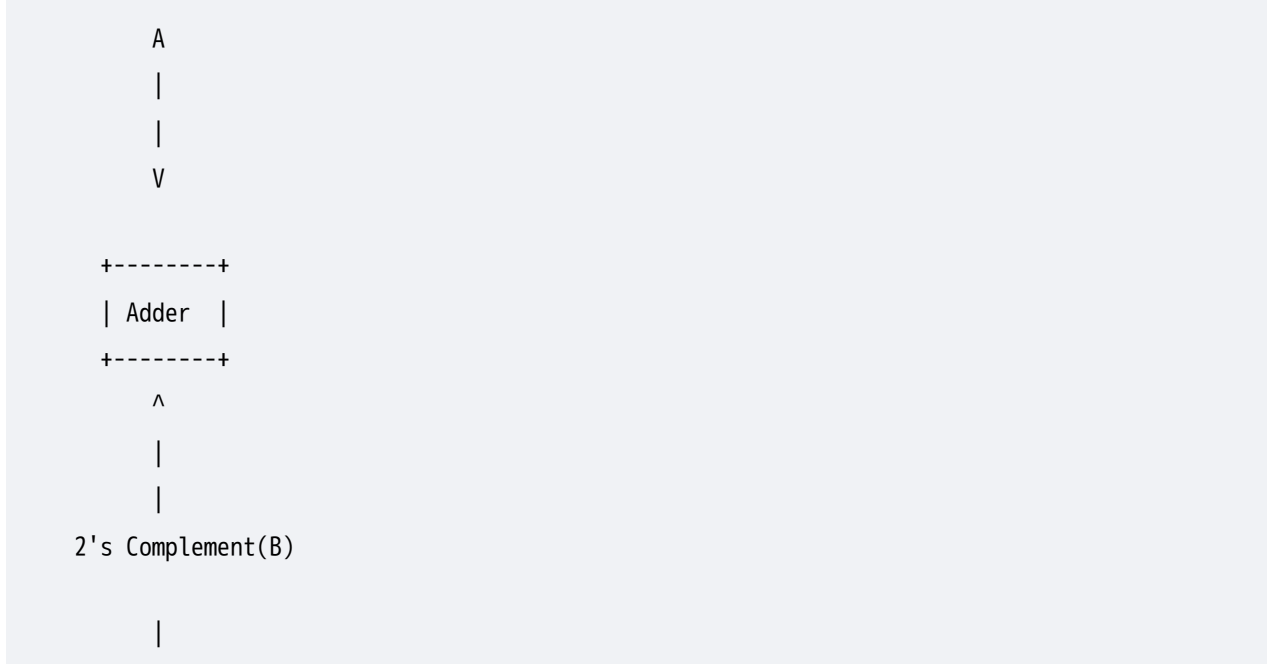
Discard Carry

```
0100
```

Answer:

```
4
```

Hardware Representation



```
          V

       Result
```

Advantages of 2's Complement

1. Easy Representation

Negative numbers easily represent hote hain.

---

2. Simplifies Subtraction

Subtraction addition me convert ho jati hai.

---

3. Single Arithmetic Circuit

Same hardware use hota hai.

---

4. Fast Processing

Computer speed improve hoti hai.

---

5. No Separate Sign Handling

Sign automatically handle ho jata hai.

Disadvantages

1. Concept Difficult for Beginners

Initial understanding difficult ho sakti hai.

---

2. Fixed Range

Bit length ke according range limited hoti hai.

Applications

CPU Arithmetic

Addition and subtraction.

ALU Design

Arithmetic circuits.

Digital Computers

Negative number representation.

Embedded Systems

Microcontrollers.

Signal Processing

DSP applications.

1's Complement vs 2's Complement

1's Complement	2's Complement
Bits Inverted	1's Complement + 1
Two Zeros Possible	Single Zero
Complex Arithmetic	Simple Arithmetic
Less Efficient	More Efficient
Rarely Used	Widely Used

Viva Questions

Q1. What is 2's Complement?

Method of representing negative binary numbers.

Q2. How is 2's Complement obtained?

1's Complement + 1

Q3. Why is 2's Complement used?

To represent negative numbers and simplify subtraction.

Q4. What is the range of 8-bit 2's Complement?

```
-128 to +127
```

Q5. Which complement is most commonly used?

2's Complement.

Frequently Asked RGPV Questions

2 Marks

1. Define 2's Complement.
2. How is 2's Complement obtained?
3. What is 1's Complement?
4. Why is 2's Complement used?

---

5 Marks

1. Explain 2's Complement Representation.
2. Explain method of finding 2's Complement.
3. Compare 1's and 2's Complement.

---

7 Marks

1. Explain 2's Complement with examples.
2. Explain subtraction using 2's Complement.
3. Discuss advantages of 2's Complement.

---

14 Marks

1. Explain 2's Complement Representation with suitable examples.
2. Discuss subtraction using 2's Complement method.
3. Compare 1's Complement and 2's Complement and explain their applications.

PYQ Trend Analysis

Topic	Frequency
2's Complement Basics	⭐⭐⭐⭐⭐
Numerical Problems	⭐⭐⭐⭐⭐
Subtraction Using 2's Complement	⭐⭐⭐⭐⭐
1's vs 2's Complement	⭐⭐⭐⭐

Expected 2026 Questions

🔥 Explain 2's Complement Representation with examples.

🔥 Perform subtraction using 2's Complement.

🔥 Compare 1's Complement and 2's Complement.

🔥 Explain advantages of 2's Complement.

One-Minute Revision

✅ 2's Complement = 1's Complement + 1

✅ Negative numbers represented using 2's Complement

✅ Formula:

```
A - B = A + 2's Complement(B)
```

✅ 8-bit Range:

```
-128 to +127
```

✅ Most commonly used representation method

Conclusion

2's Complement Representation computer arithmetic me negative numbers ko represent karne ki sabse efficient technique hai. Iski madad se subtraction ko addition me convert kiya ja sakta hai aur hardware design simplify ho jata hai. Fast processing, simple arithmetic aur efficient representation ke karan ye modern computer systems me sabse adhik use kiya jata hai. 🎯

Signed Addition and Subtraction

> ⭐⭐⭐⭐⭐ Most Important Topic
>
> RGPV me Signed Addition & Subtraction se theory aur numerical dono puchhe jaate hain.
>
> Booth Algorithm aur Floating Point samajhne ke liye bhi ye topic important hai.
>
> Ye answer **3–4 pages** aasani se cover karega.

Signed Addition and Subtraction

Introduction

Computer me numbers do types ke hote hain:

1. Unsigned Numbers

Sirf positive numbers represent karte hain.

Example:

```
5, 10, 25
```

---

2. Signed Numbers

Positive aur Negative dono numbers represent karte hain.

Example:

```
+5, -5, +10, -10
```

Signed arithmetic me sign bit ka use hota hai.

Definition

"Signed Addition and Subtraction are arithmetic operations performed on positive and negative binary numbers using sign representation methods such as 2's Complement."

Sign Bit Representation

Most Significant Bit (MSB) sign ko represent karti hai.

MSB	Meaning
0	Positive Number
1	Negative Number

Examples

Positive Number

```
+5

00000101
```

MSB = 0

Negative Number

```
-5

11111011
```

MSB = 1

Need of Signed Arithmetic

Signed arithmetic ka use:

- Banking Calculations
- Scientific Computing
- Engineering Applications
- Computer Programming
- Digital Signal Processing

me hota hai.

Rules of Signed Addition

Signed Addition ke 4 possible cases hote hain.

Case 1: Positive + Positive

Example

```
+5 + +3
```

Binary:

```
00000101
+
00000011
----------
00001000
```

Answer:

```
+8
```

Rule

Signs same hain.

Simply add kar do.

Result positive hoga.

Case 2: Negative + Negative

Example

```
-5 + -3
```

Binary Representation

```
-5 = 11111011

-3 = 11111101
```

Addition:

```
11111011
+
11111101
```

```
----------
11111000
```

Result:

```
-8
```

Rule

Signs same hain.

Numbers add karo.

Result negative hoga.

Case 3: Positive + Negative

Example

```
+8 + (-3)
```

Binary

```
+8 = 00001000

-3 = 11111101
```

Addition:

```
00001000
+
11111101
----------
00000101
```

Answer:

```
+5
```

Rule

Signs different hain.

Smaller number ko larger number se subtract karo.

Result larger number ke sign ka hoga.

Case 4: Negative + Positive

Example

```
-8 + 3
```

Result:

```
-5
```

Rule

Same as Case-3

Signed Subtraction

Formula

Subtraction ko addition me convert karte hain.

```
A - B

=

A + (2's Complement of B)
```

Example 1

Find:

```
7 - 3
```

Binary

```
7 = 0111

3 = 0011
```

2's Complement of 3

```
0011

↓

1100

↓

1101
```

Addition

```
0111
+
1101
```

```
-------
10100
```

Discard Carry

```
0100
```

Answer:

```
4
```

Example 2

Find:

```
5 - 8
```

Binary

```
5 = 0101

8 = 1000
```

2's Complement of 8

```
1000

↓

0111

↓

1000
```

Addition

```
0101
+
1000
-------
1101
```

Result:

```
-3
```

Overflow in Signed Arithmetic

Definition

Overflow tab hota hai jab result available bits me represent nahi ho sakta.

Example

4-bit Representation

```
0111
+
0001
-------
1000
```

Expected:

```
7 + 1 = 8
```

Lekin 4-bit signed range exceed ho gayi.

Overflow occur hua.

Overflow Detection Rules

Positive + Positive = Negative

Overflow

Negative + Negative = Positive

Overflow

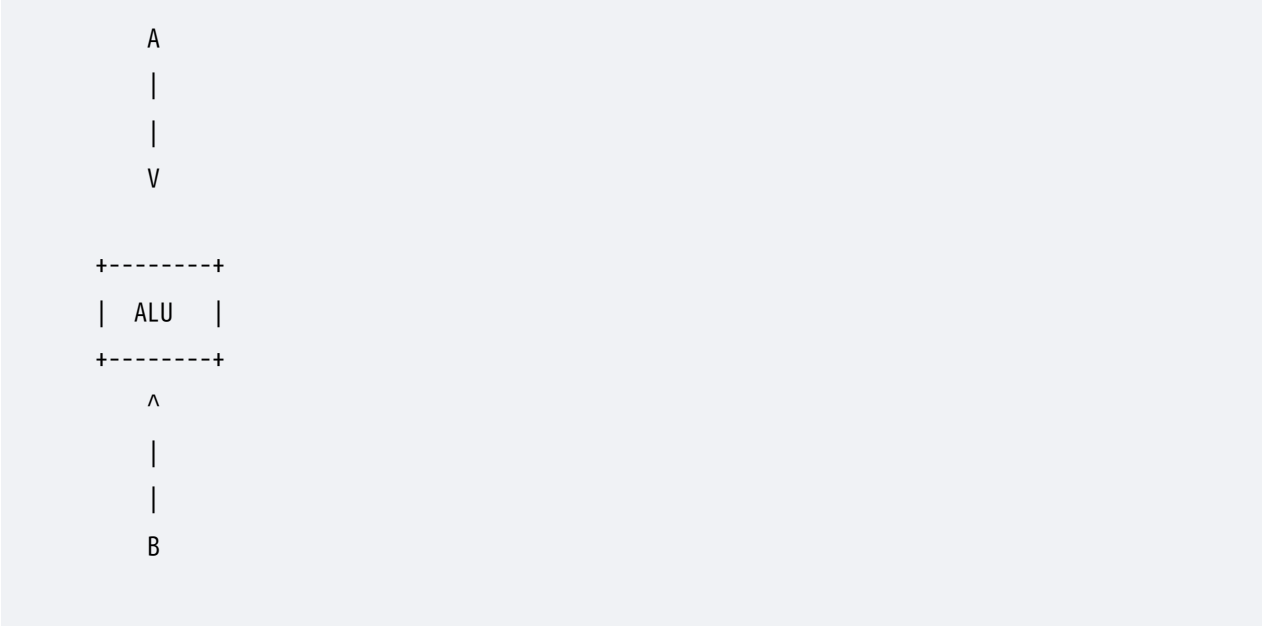
Positive + Negative

No Overflow

Hardware for Signed Arithmetic

Signed arithmetic ALU ke through perform hota hai.

Block Diagram



```
          |
          V

      Result
```

Steps of Signed Addition

```
Load Operand A
        ↓

Load Operand B
        ↓

Check Sign Bits
        ↓

Perform Addition
        ↓

Check Overflow
        ↓

Store Result
```

Steps of Signed Subtraction

```
Load A
   ↓

Find 2's Complement of B
   ↓

Add A + (-B)
   ↓

Check Overflow
   ↓
```

```
Store Result
```

Advantages of Signed Arithmetic

1. Supports Negative Numbers

Negative values represent kar sakte hain.

---

2. Simplifies Calculations

Subtraction addition me convert ho jati hai.

---

3. Efficient Hardware

Same circuit use hota hai.

---

4. Fast Processing

CPU operations fast ho jati hain.

---

5. Useful in Real Applications

Scientific aur engineering calculations.

Disadvantages

1. Overflow Problem

Large values par overflow ho sakta hai.

---

2. Complex for Beginners

Sign handling difficult lag sakta hai.

Applications

Banking Systems

Profit/Loss calculations.

Scientific Computing

Positive & Negative values.

Signal Processing

Wave calculations.

Embedded Systems

Microcontroller arithmetic.

Computer Graphics

Coordinate calculations.

Signed vs Unsigned Numbers

Signed Numbers	Unsigned Numbers
Positive & Negative	Only Positive

Uses Sign Bit	No Sign Bit
Smaller Positive Range	Larger Positive Range
More Flexible	Simpler

Viva Questions

Q1. What is a Signed Number?

Number having a sign bit.

Q2. What is MSB?

Most Significant Bit.

Q3. What does MSB = 1 indicate?

Negative Number.

Q4. What is Overflow?

Condition when result exceeds available bits.

Q5. How is subtraction performed?

Using 2's Complement.

Frequently Asked RGPV Questions

2 Marks

1. Define Signed Number.
2. What is Sign Bit?
3. What is Overflow?
4. Define Signed Arithmetic.

---

5 Marks

1. Explain Signed Addition.
2. Explain Signed Subtraction.
3. Explain Overflow.

---

7 Marks

1. Explain Signed Addition with examples.
2. Explain Signed Subtraction using 2's Complement.
3. Explain Overflow Detection.

---

14 Marks

1. Explain Signed Addition and Signed Subtraction with suitable examples.
2. Discuss overflow in signed arithmetic.
3. Explain arithmetic operations on signed numbers using 2's Complement.

PYQ Trend Analysis

Topic	Frequency
Signed Addition	⭐⭐⭐⭐⭐
Signed Subtraction	⭐⭐⭐⭐⭐
Overflow	⭐⭐⭐⭐

Numerical Problems	⭐⭐⭐⭐⭐

Expected 2026 Questions

🔥 Explain Signed Addition with examples.

🔥 Explain Signed Subtraction using 2's Complement.

🔥 Explain Overflow and its detection.

🔥 Perform signed arithmetic numericals.

One-Minute Revision

✅ MSB = Sign Bit

```
0 → Positive

1 → Negative
```

✅ Same Signs → Add

✅ Different Signs → Subtract

✅ Subtraction:

```
A - B

=

A + 2's Complement(B)
```

✅ Overflow:

```
(+)+(+) = (-)

or
```

```
(-)+(-) = (+)
```

Conclusion

Signed Addition and Subtraction computer arithmetic ke important operations hain jo positive aur negative binary numbers par perform kiye jate hain. 2's Complement representation aur ALU ki madad se ye operations efficiently execute hote hain. Overflow detection aur sign-bit handling is topic ke key concepts hain. Ye RGPV exams ka highly important numerical aur theory topic hai. 🎯

Multiplication and Division

> ⭐⭐⭐⭐⭐ Most Important Topic
>
> RGPV me Multiplication, Division aur Booth Algorithm ek hi unit ke most scoring topics hain.
>
> Theory + Numerical dono puchhe ja sakte hain.
>
> Ye answer **3–4 pages** aasani se cover karega.

Multiplication and Division

Introduction

Computer Arithmetic me Multiplication aur Division do important arithmetic operations hain.

Computer directly decimal numbers par kaam nahi karta. Isliye ye operations binary numbers par perform kiye jaate hain.

ALU (Arithmetic Logic Unit) multiplication aur division operations ko execute karta hai.

Definition

Binary Multiplication

"Binary Multiplication is the process of multiplying two binary numbers using binary arithmetic rules."

Binary Division

"Binary Division is the process of dividing one binary number by another binary number using binary arithmetic rules."

Need of Multiplication and Division

Multiplication aur Division ka use:

- Scientific Calculations
- Banking Systems
- Engineering Applications
- Computer Graphics
- Digital Signal Processing

me hota hai.

Binary Multiplication

Binary multiplication decimal multiplication ki tarah hi hoti hai.

Rules of Binary Multiplication

A	B	Result
0	0	0
0	1	0
1	0	0
1	1	1

Important Rule

```
1 × 1 = 1
```

Example 1

Multiply:

```
101
× 11
```

Step 1

```
      101
    × 011
--------------
      101
+    1010
--------------
     1111
```

Verification

$$101_2 = 5$$

$$11_2 = 3$$

Result:

$$1111_2 = 15$$

Therefore:

$$5 \times 3 = 15$$

Correct.

Example 2

Multiply:

$$\begin{array}{r} 110 \\ \times\ 10 \end{array}$$

Solution

$$\begin{array}{r} \quad 110 \\ \times \quad 10 \\ \hline \quad 000 \\ +\ 1100 \\ \hline 1100 \end{array}$$

Verification

$110_2 = 6$

$10_2 = 2$

$6 \times 2 = 12$

$1100_2 = 12$

Correct.

Hardware Multiplication Process

```
Multiplicand
      |
      V

Multiplier
      |
      V

 Shift & Add
      |
      V

 Product
```

Steps of Binary Multiplication

Step 1

Take Multiplicand.

Step 2

Take Multiplier.

Step 3

Check multiplier bit.

Step 4

If bit = 1 → Add

If bit = 0 → Ignore

Step 5

Shift left.

Step 6

Repeat until all bits processed.

Advantages of Binary Multiplication

1. Simple Logic

Easy hardware implementation.

---

2. Fast Processing

Efficient ALU operations.

---

3. Suitable for Digital Systems

Used in processors.

Binary Division

Binary Division decimal division ke similar hoti hai.

Terms Used

```
Dividend ÷ Divisor
```

Example

```
1010 ÷ 10
```

Binary Division Rules

A	B	Result
0 ÷ 1	0	
1 ÷ 1	1	
0 ÷ 0	Undefined	
1 ÷ 0	Not Possible	

Example 1

Divide:

```
1010 ÷ 10
```

Solution

```
       101
     ------
10 ) 1010
     10
     ----
      001
```

Verification

$1010_2 = 10$

$10_2 = 2$

$10 \div 2 = 5$

$101_2 = 5$

Correct.

Example 2

Divide:

$1100 \div 10$

Solution

```
       110
     ------
10 ) 1100
     10
     ----
      100
      10
      ---
       00
```

Answer:

```
110₂
```

Verification

```
12 ÷ 2 = 6
```

Correct.

Hardware Division Process

```
Dividend
    |
    V

 Divisor
    |
    V

Compare
Subtract
Shift
    |
    V

 Quotient
```

Division Algorithm

Step 1

Load Dividend.

Step 2

Load Divisor.

Step 3

Compare Divisor and Dividend.

Step 4

Subtract Divisor.

Step 5

Generate Quotient Bit.

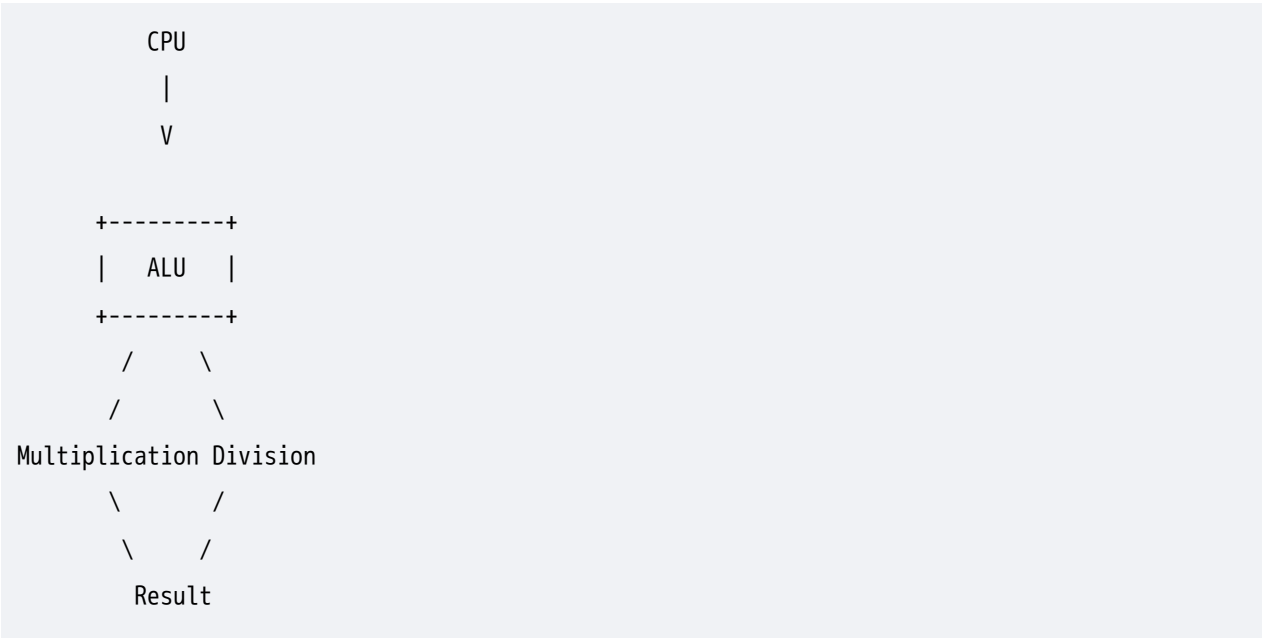
Step 6

Repeat until completion.

Multiplication vs Division

Multiplication	Division
Repeated Addition	Repeated Subtraction
Product Obtained	Quotient Obtained
Faster	Comparatively Slower
Shift Left Used	Shift Right Used
Example: 5×3	Example: 15÷3

Hardware Block Diagram



Applications

Scientific Computing

Complex calculations.

Computer Graphics

Image processing.

Banking Systems

Interest calculations.

Digital Signal Processing

Signal operations.

Embedded Systems

Microcontroller calculations.

Advantages

1. Fast Computation

Complex calculations quickly perform hoti hain.

---

2. High Accuracy

Binary arithmetic errors kam hote hain.

---

3. Efficient Hardware

ALU implementation easy hoti hai.

---

4. Essential for Modern Computing

Almost every application me use hota hai.

Disadvantages

1. Large Binary Numbers

Manual calculations difficult.

---

2. Division Slower Than Addition

More clock cycles lagte hain.

---

3. Hardware Complexity

Advanced multiplication circuits complex hote hain.

Viva Questions

Q1. What is Binary Multiplication?

Multiplication of two binary numbers.

Q2. What is Binary Division?

Division of one binary number by another.

Q3. Which unit performs multiplication?

ALU.

Q4. Which operation is slower?

Division.

Q5. What is Product?

Result of multiplication.

Frequently Asked RGPV Questions

2 Marks

1. Define Binary Multiplication.

2. Define Binary Division.
3. What is Product?
4. What is Quotient?

---

5 Marks

1. Explain Binary Multiplication.
2. Explain Binary Division.
3. Write multiplication rules.

---

7 Marks

1. Explain multiplication with example.
2. Explain division with example.
3. Compare multiplication and division.

---

14 Marks

1. Explain Binary Multiplication and Binary Division with suitable examples.
2. Discuss multiplication and division operations in computer arithmetic.
3. Explain hardware implementation of multiplication and division.

PYQ Trend Analysis

Topic	Frequency
Binary Multiplication	⭐⭐⭐⭐⭐
Binary Division	⭐⭐⭐⭐⭐
Numerical Problems	⭐⭐⭐⭐⭐

Hardware Implementation	⭐⭐⭐⭐

Expected 2026 Questions

🔥 Explain Binary Multiplication with example.

🔥 Explain Binary Division with example.

🔥 Compare Multiplication and Division.

🔥 Discuss arithmetic operations performed by ALU.

One-Minute Revision

✅ Multiplication = Repeated Addition

✅ Division = Repeated Subtraction

✅ Rule:

```
1 × 1 = 1
```

✅ Product = Multiplication Result

✅ Quotient = Division Result

✅ ALU performs both operations

Conclusion

Multiplication and Division computer arithmetic ke fundamental operations hain jo binary numbers par perform kiye jate hain. ALU in operations ko execute karta hai aur shift-add aur

repeated subtraction techniques ki madad se results generate karta hai. Ye Computer Organization ka important aur frequently asked topic hai aur numericals ke liye bhi bahut scoring hai. 🎯

Booth's Algorithm

> ⭐⭐⭐⭐⭐ Most Important & Most Repeated Numerical Topic
>
> RGPV me Booth's Algorithm se direct **7 marks**, **10 marks** aur **14 marks** ke questions aate hain.
>
> Exam me numerical solve karne ke chances bahut high hote hain.
>
> Ye answer **4–5 pages** aasani se cover karega.

Booth's Algorithm

Introduction

Computer me signed binary numbers ka multiplication karna normal multiplication se thoda difficult hota hai.

Is problem ko solve karne ke liye **Andrew D. Booth** ne Booth's Algorithm diya tha.

Booth's Algorithm signed positive aur signed negative binary numbers ke multiplication ke liye use kiya jata hai.

Definition

"Booth's Algorithm is an efficient multiplication algorithm used for multiplying signed binary numbers in 2's complement form."

Need of Booth's Algorithm

1. Signed Number Multiplication

Positive aur Negative dono numbers multiply kar sakta hai.

2. Fast Computation

Number of additions reduce karta hai.

3. Efficient Hardware

Processor implementation easy hoti hai.

4. Used in ALU

Modern processors me use hota hai.

Basic Principle

Booth's Algorithm multiplier ke consecutive bits ko observe karta hai.

Hum do bits check karte hain:

```
Q0   Q-1
```

Decision Table

Q0	Q-1	Operation

0	0	No Operation
1	1	No Operation
0	1	Add Multiplicand (M)
1	0	Subtract Multiplicand (M)

Registers Used

M

Multiplicand

Q

Multiplier

A

Accumulator

Q-1

Extra Bit

Count

Number of bits

Booth's Algorithm Flow

```
Initialize Registers
        ↓
Check Q0 and Q-1
        ↓
Add/Subtract M
        ↓
Arithmetic Right Shift
        ↓
Count - 1
        ↓
Repeat
        ↓
Final Product
```

Algorithm Steps

Step 1

Initialize:

```
A = 0000

Q = Multiplier

M = Multiplicand

Q-1 = 0
```

Step 2

Check:

```
Q0 and Q-1
```

Step 3

Apply Rule

Q0Q-1	Operation
00	Nothing
11	Nothing
01	A = A + M
10	A = A - M

Step 4

Arithmetic Right Shift

```
(A,Q,Q-1)
```

Step 5

Count Reduce

Step 6

Repeat Until Count = 0

Numerical Example

Multiply:

```
5 × 3
```

Binary Representation

```
M = 0101

Q = 0011

A = 0000

Q-1 = 0

Count = 4
```

Step-1

Current:

```
A = 0000

Q = 0011

Q-1 = 0
```

Check:

```
Q0Q-1

1 0
```

Rule:

```
10 → A = A - M
```

A:

```
0000

-
```

```
0101

=

1011
```

Arithmetic Right Shift

```
A = 1101

Q = 1001

Q-1 = 1
```

Step-2

Check:

```
Q0Q-1

1 1
```

No Operation

Shift

```
A = 1110

Q = 1100

Q-1 = 1
```

Step-3

Check:

```
Q0Q-1

0 1
```

Rule:

```
01 → A = A + M
```

Addition:

```
1110

+

0101

=

0011
```

Shift

```
A = 0001

Q = 1110

Q-1 = 0
```

Step-4

Check:

```
Q0Q-1

0 0
```

No Operation

Shift

```
A = 0000

Q = 1111
```

Final Result

```
AQ

00001111
```

Decimal:

```
15
```

Hence:

```
5 × 3 = 15
```

Correct.

Hardware Block Diagram

```
          M
          |
          |
          V

      +---------+
      | Adder/  |
      |Subtractor|
      +---------+
           |
           V

          A
          |
```

```
          |
          V

          Q
          |
          |
         Q-1
```

Arithmetic Right Shift

Most Important Concept

Example:

Before Shift

```
A = 1010

Q = 1100

Q-1 = 1
```

After Shift

```
A = 1101

Q = 0110

Q-1 = 0
```

Sign bit preserve hoti hai.

Advantages of Booth's Algorithm

1. Signed Multiplication

Positive aur negative dono handle karta hai.

---

2. Fast Processing

Additions reduce hoti hain.

---

3. Efficient Hardware

ALU implementation easy.

---

4. Reduced Computation

Less arithmetic operations.

---

5. Widely Used

Modern processors me use hota hai.

Disadvantages

1. Complex Algorithm

Normal multiplication se difficult.

---

2. Shift Operations Required

Extra hardware lag sakta hai.

---

3. Difficult Manual Calculation

Exam me mistakes ho sakti hain.

Applications

CPU Design

Processor multiplication unit.

Digital Signal Processing

DSP calculations.

Scientific Computing

Fast arithmetic.

Embedded Systems

Microcontrollers.

Computer Architecture

Signed arithmetic operations.

Booth's Algorithm vs Ordinary Multiplication

Booth Algorithm	Ordinary Multiplication
Signed Numbers	Mostly Unsigned
Faster	Slower
Less Additions	More Additions
Uses Shifting	Simple Multiplication

Hardware Efficient	Less Efficient

Viva Questions

Q1. What is Booth's Algorithm?

Algorithm used for signed binary multiplication.

Q2. Who proposed Booth's Algorithm?

Andrew D. Booth.

Q3. Which representation is used?

2's Complement.

Q4. Why is Q-1 used?

To determine operation.

Q5. Which shift is used?

Arithmetic Right Shift.

Frequently Asked RGPV Questions

2 Marks

1. Define Booth's Algorithm.
2. What is Arithmetic Right Shift?
3. What is Q-1?

4. Why is Booth's Algorithm used?

---

5 Marks

1. Explain Booth's Algorithm.
2. Explain Arithmetic Right Shift.
3. Write decision table of Booth's Algorithm.

---

7 Marks

1. Explain Booth's Algorithm with flowchart.
2. Perform multiplication using Booth's Algorithm.
3. Explain registers used in Booth's Algorithm.

---

14 Marks

1. Explain Booth's Algorithm with suitable numerical example.
2. Discuss signed binary multiplication using Booth's Algorithm.
3. Explain Booth's Algorithm with block diagram and working.

PYQ Trend Analysis

Topic	Frequency
Booth Algorithm Theory	⭐⭐⭐⭐⭐
Numerical Problems	⭐⭐⭐⭐⭐
Arithmetic Right Shift	⭐⭐⭐⭐
Decision Table	⭐⭐⭐⭐

Expected 2026 Questions

🔥 Explain Booth's Algorithm with numerical example.

🔥 Perform signed multiplication using Booth's Algorithm.

🔥 Explain Arithmetic Right Shift in Booth's Algorithm.

🔥 Discuss advantages of Booth's Algorithm.

One-Minute Revision

✅ Used for Signed Multiplication

✅ Uses 2's Complement

✅ Registers:

```
A

Q

M

Q-1
```

✅ Rules:

Q0Q-1	**Action**
00	Nothing
11	Nothing
01	Add M
10	Subtract M

✅ Arithmetic Right Shift after every step

Conclusion

Booth's Algorithm signed binary multiplication ke liye ek efficient technique hai jo 2's Complement representation aur Arithmetic Right Shift ka use karti hai. Ye multiplication process ko fast aur hardware-efficient banati hai. Computer Architecture, ALU design aur modern processors me iska bahut important role hai. RGPV exams me ye sabse important numerical topics me se ek hai. 🎯

Floating Point Arithmetic Operation

> ⭐⭐⭐⭐⭐ Most Important Topic
>
> RGPV me Floating Point Representation aur Floating Point Arithmetic se theory + numerical dono puchhe jate hain.
>
> Ye topic computer me **very large aur very small numbers** represent karne ke liye use hota hai.
>
> Ye answer **4–5 pages** aasani se cover karega.

Floating Point Arithmetic Operation

Introduction

Computer me kai baar bahut bade numbers aur bahut chhote decimal numbers ke saath calculations karni padti hain.

Example:

```
125000000

0.000000125
```

In numbers ko normal binary representation me store karna difficult hota hai.

Isliye Floating Point Representation ka use kiya jata hai.

Definition

"Floating Point Arithmetic is a method of representing and performing arithmetic operations on real numbers using a mantissa and exponent."

Scientific Notation Concept

Floating Point Representation scientific notation par based hoti hai.

Example

Decimal Number:

```
125000
```

Scientific Notation:

```
1.25 × 10⁵
```

Example

```
0.000125
```

Scientific Notation:

```
1.25 × 10⁻⁴
```

Floating Point Representation

Floating Point Number ke three parts hote hain:

```
---------------------------------
| Sign | Exponent | Mantissa |
---------------------------------
```

---

1. Sign Bit

Number positive hai ya negative.

```
0 = Positive

1 = Negative
```

---

2. Exponent

Decimal point ki position batata hai.

---

3. Mantissa

Actual significant digits store karta hai.

Example

Number:

```
13.25
```

Binary:

```
1101.01
```

Normalize:

```
1.10101 × 2³
```

Components

```
Sign = 0

Exponent = 3

Mantissa = 10101
```

Floating Point Format

32-bit IEEE 754 Format

Most Important

```
-------------------------------------
| Sign | Exponent | Mantissa |
-------------------------------------
| 1 bit| 8 bits   | 23 bits  |
-------------------------------------
```

IEEE 754 Representation

Sign Bit

1 Bit

Exponent

8 Bits

Fraction/Mantissa

23 Bits

Diagram

```
-----------------------------------------------
| S | EEEEEEEE | MMMMMMMMMMMMMMMMMMMMMMM |
-----------------------------------------------
```

Example

```
+5.75
```

Binary:

```
101.11
```

Normalize:

```
1.0111 × 2²
```

Components:

```
Sign = 0

Exponent = 2

Mantissa = 0111
```

Floating Point Addition

Most Important

Steps

Step 1

Equalize Exponents

Step 2

Add Mantissas

Step 3

Normalize Result

Step 4

Adjust Exponent

Example

Add:

```
1.5 × 10²

+

2.5 × 10²
```

Same exponent hai.

Add mantissa:

```
1.5 + 2.5

=

4.0
```

Result:

```
4.0 × 10²
```

Floating Point Addition Flow

```
Align Exponents
        ↓
Add Mantissas
        ↓
Normalize
        ↓
Store Result
```

Floating Point Subtraction

Steps

Step 1

Equalize Exponents

Step 2

Subtract Mantissas

Step 3

Normalize Result

Step 4

Store Result

Example

```
5.5 × 10²

-

2.5 × 10²
```

Subtract:

```
5.5 - 2.5

=

3.0
```

Result:

```
3.0 × 10²
```

Floating Point Multiplication

Most Important

Rule

Multiply Mantissas

Add Exponents

Formula

```
(M1 × 2E1)

×

(M2 × 2E2)

=

(M1 × M2)

×

2(E1+E2)
```

Example

```
1.5 × 10²

×

2.0 × 10³
```

Mantissa:

```
1.5 × 2.0

=

3.0
```

Exponent:

```
2 + 3

=

5
```

Result:

```
3.0 × 10⁵
```

Floating Point Division

Most Important

Rule

Divide Mantissas

Subtract Exponents

Formula

```
(M1 × 2E1)

÷
```

```
(M2 × 2E2)

=

(M1 ÷ M2)

×

2(E1-E2)
```

Example

```
8 × 10⁵

÷

2 × 10²
```

Mantissa:

```
8 ÷ 2

=

4
```

Exponent:

```
5 - 2

=

3
```

Result:

```
4 × 10³
```

Normalization

Most Important Concept

Definition

Floating point number ko standard form me convert karna.

Example

```
101.01
```

Normalize:

```
1.0101 × 2²
```

Overflow

Definition

Jab exponent maximum limit se exceed ho jaye.

Example

Very large number:

```
10³⁰⁰
```

Overflow generate kar sakta hai.

Underflow

Definition

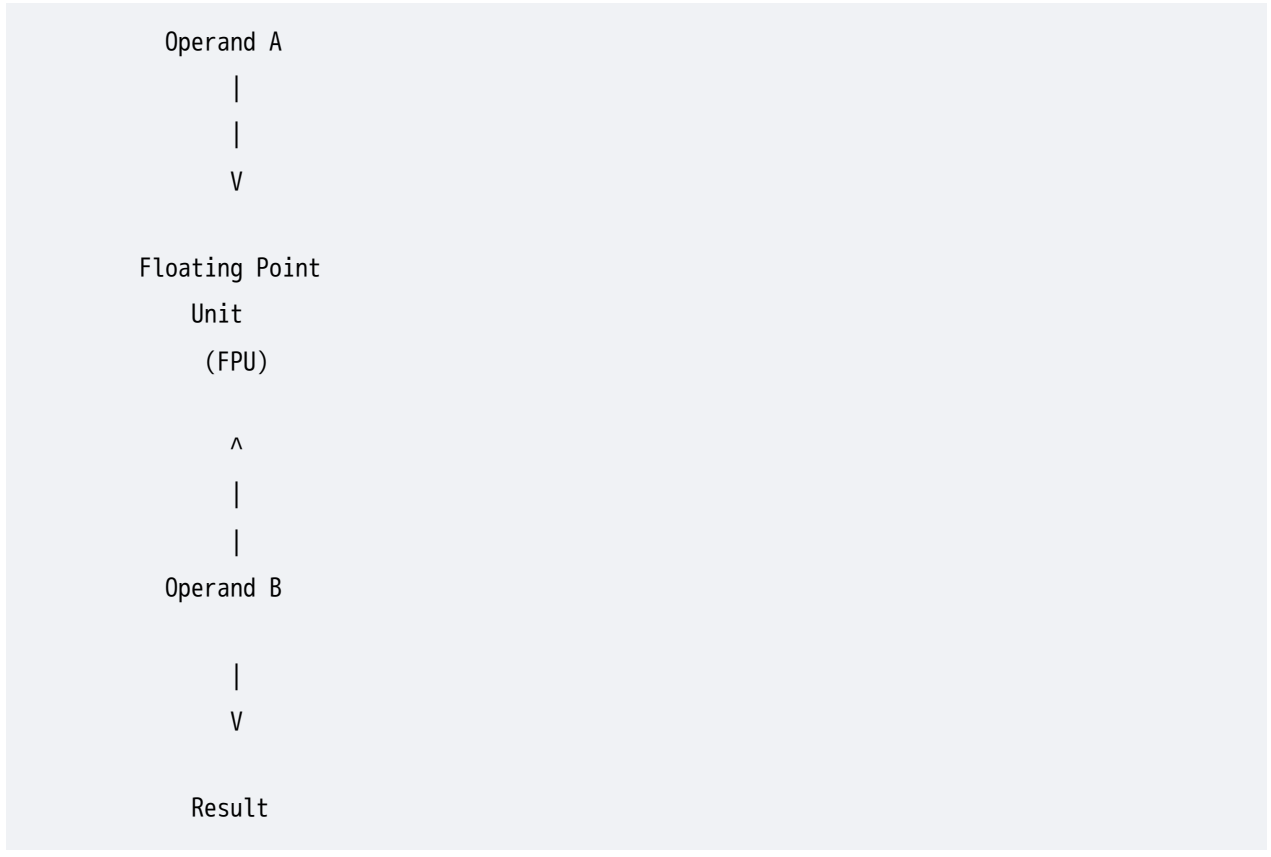
Jab exponent bahut chhota ho jaye.

Example

```
10^-300
```

Underflow generate kar sakta hai.

Hardware for Floating Point Arithmetic



Floating Point Unit (FPU)

Definition

Special hardware unit jo floating point operations perform karti hai.

Functions

- Addition
- Subtraction
- Multiplication
- Division

Advantages

1. Large Range

Bahut bade numbers represent kar sakta hai.

---

2. High Precision

Accurate calculations.

---

3. Scientific Applications

Engineering calculations ke liye suitable.

---

4. Fast Processing

Modern processors me dedicated FPU hota hai.

---

5. Real Number Support

Decimal values represent kar sakta hai.

Disadvantages

1. Complex Hardware

Implementation difficult hoti hai.

---

2. Rounding Errors

Precision loss ho sakti hai.

---

3. High Cost

FPU design expensive hoti hai.

Applications

Scientific Computing

Physics simulations.

Engineering

CAD/CAM systems.

Artificial Intelligence

Machine Learning calculations.

Computer Graphics

3D Rendering.

Weather Forecasting

Complex mathematical models.

Fixed Point vs Floating Point

Fixed Point	Floating Point
Decimal Position Fixed	Decimal Position Variable
Small Range	Large Range
Simple Hardware	Complex Hardware
Less Precision	High Precision
Faster	Slightly Slower

Viva Questions

Q1. What is Floating Point Arithmetic?

Arithmetic performed on real numbers using exponent and mantissa.

Q2. What is Mantissa?

Significant digits of number.

Q3. What is Exponent?

Position of decimal point.

Q4. What is IEEE 754?

Standard floating point representation.

Q5. What is FPU?

Floating Point Unit.

Frequently Asked RGPV Questions

2 Marks

1. Define Floating Point Representation.
2. What is Mantissa?
3. What is Exponent?
4. What is FPU?

---

5 Marks

1. Explain Floating Point Representation.
2. Explain IEEE 754 format.
3. Explain Normalization.

---

7 Marks

1. Explain Floating Point Addition.
2. Explain Floating Point Multiplication.
3. Explain Overflow and Underflow.

---

14 Marks

1. Explain Floating Point Arithmetic Operations with suitable examples.
2. Discuss Floating Point Representation and IEEE 754 format.
3. Explain Addition, Subtraction, Multiplication and Division in Floating Point Arithmetic.

PYQ Trend Analysis

Topic	Frequency
Floating Point Representation	⭐⭐⭐⭐⭐
IEEE 754 Format	⭐⭐⭐⭐⭐
Floating Point Arithmetic	⭐⭐⭐⭐⭐
Overflow/Underflow	⭐⭐⭐⭐
Normalization	⭐⭐⭐⭐

Expected 2026 Questions

🔥 Explain Floating Point Arithmetic Operations.

🔥 Explain IEEE 754 Floating Point Format.

🔥 Explain Floating Point Addition and Multiplication.

🔥 Explain Normalization, Overflow and Underflow.

One-Minute Revision

✅ Floating Point Number:

```
Sign + Exponent + Mantissa
```

✅ IEEE 754:

```
1 Bit Sign

8 Bit Exponent

23 Bit Mantissa
```

✅ Multiplication:

```
Multiply Mantissa

Add Exponents
```

✅ Division:

```
Divide Mantissa

Subtract Exponents
```

✅ Used for Real Numbers

Conclusion

Floating Point Arithmetic computer systems me real numbers ko represent aur process karne ki ek efficient technique hai. Isme Sign Bit, Exponent aur Mantissa ka use hota hai. IEEE 754 standard floating point representation ka widely used format hai. Floating Point Arithmetic scientific computing, AI, graphics aur engineering applications me bahut important role play karti hai aur RGPV exams ka highly important topic hai. 🎯

Design of Arithmetic Unit

> ⭐⭐⭐⭐⭐ Most Important Topic
>
> RGPV me "Design of Arithmetic Unit" se direct 7 marks aur 14 marks ke questions frequently puche jate hain.
>
> Ye topic ALU, Adders aur Arithmetic Operations se directly related hai.
>
> Ye answer **3–4 pages** aasani se cover karega.

Design of Arithmetic Unit

Introduction

Computer me saare arithmetic operations jaise:

- Addition
- Subtraction
- Increment
- Decrement
- Transfer

ek special hardware unit dwara perform kiye jate hain jise **Arithmetic Unit** kehte hain.

Arithmetic Unit ALU (Arithmetic Logic Unit) ka important part hota hai.

Definition

"Arithmetic Unit is a digital circuit that performs arithmetic operations on binary data under the control of selection inputs."

Need of Arithmetic Unit

Arithmetic Unit ka use:

✅ Binary Addition

✅ Binary Subtraction

✅ Increment

✅ Decrement

✅ Data Transfer

ke liye hota hai.

Arithmetic Unit in CPU

Basic Components of Arithmetic Unit

Arithmetic Unit mainly following components se milkar banti hai:

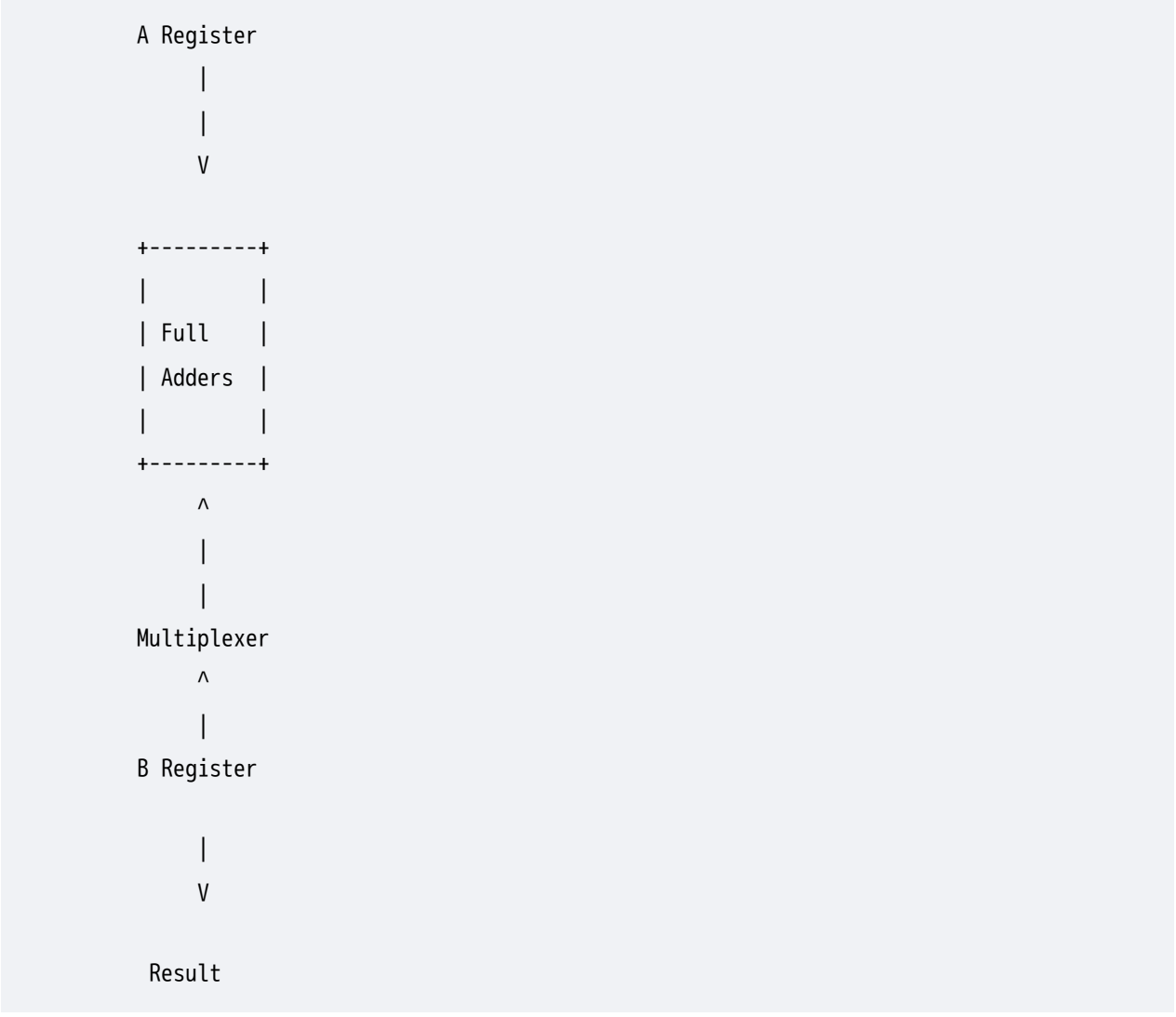
1. Full Adder

2. Multiplexer

3. Control Lines

4. Registers

Block Diagram of Arithmetic Unit



Working Principle

Arithmetic Unit do operands par operation perform karti hai.

Suppose:

```
A = Operand-1

B = Operand-2
```

Control Signals decide karte hain ki kaunsa operation perform hoga.

Arithmetic Circuit

Arithmetic Unit ka main circuit Full Adders ka combination hota hai.

4-Bit Arithmetic Unit

```
A3 A2 A1 A0
 |  |  |  |
 ----------------
 | Full Adders  |
 ----------------
 |  |  |  |
F3 F2 F1 F0
```

Selection Variables

Arithmetic Unit me generally:

```
S1

S0

Cin
```

use hote hain.

Arithmetic Operations Table

S1	S0	Cin	Operation
0	0	0	Transfer A
0	0	1	Increment A
0	1	0	Addition
0	1	1	Add with Carry
1	0	0	Subtraction
1	0	1	Decrement A
1	1	0	Transfer B
1	1	1	Increment B

> ⭐ Table exam me bahut important hai.

Operation 1: Transfer

RTL

```
F ← A
```

Meaning:

A ka data directly output me transfer hoga.

Operation 2: Increment

RTL

```
F ← A + 1
```

Example

```
A = 0101
```

Result:

```
0110
```

Operation 3: Addition

RTL

```
F ← A + B
```

Example

```
A = 0101

B = 0011
```

Result:

```
1000
```

Decimal:

```
5 + 3 = 8
```

Operation 4: Add with Carry

RTL

```
F ← A + B + Cin
```

Example

```
5 + 3 + 1 = 9
```

Operation 5: Subtraction

Most Important

Formula

```
A - B
=
A + 2's Complement(B)
```

Example

```
7 - 3
```

Binary:

```
0111

0011
```

2's Complement:

```
1101
```

Addition:

```
0111
+
1101
-------
0100
```

Answer:

```
4
```

Operation 6: Decrement

RTL

```
F ← A - 1
```

Example

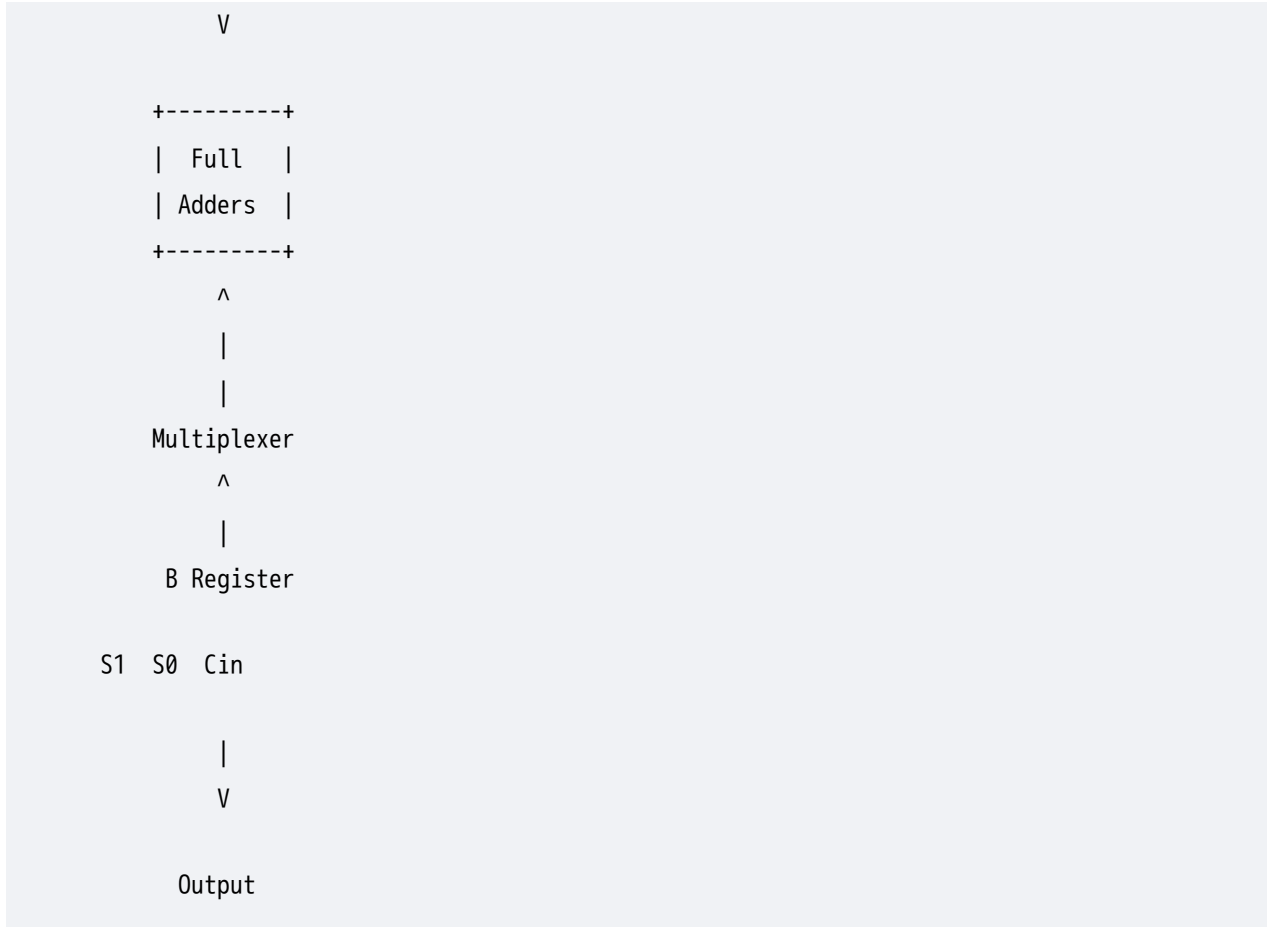
```
A = 0110
```

Result:

```
0101
```

Hardware Design of Arithmetic Unit

```
          A Register
              |
              |
```



Role of Multiplexer

Multiplexer arithmetic operation select karta hai.

Example:

```
S1 = 0

S0 = 1
```

Addition select ho sakta hai.

Role of Full Adder

Full Adder perform karta hai:

```
A + B + Cin
```

Har bit ke liye ek Full Adder use hota hai.

Advantages of Arithmetic Unit

1. Fast Calculations

High-speed arithmetic operations.

---

2. Hardware Efficient

Single circuit multiple operations perform karta hai.

---

3. Flexible

Different arithmetic functions perform kar sakta hai.

---

4. Used in ALU

CPU performance improve hoti hai.

---

5. Automatic Operation Selection

Control signals ke through.

Disadvantages

1. Complex Circuit Design

Large bit-width par complexity badh jati hai.

---

2. Hardware Cost

Advanced processors me cost increase hoti hai.

---

3. Power Consumption

Large arithmetic units more power consume karti hain.

Applications

CPUs

Arithmetic calculations.

Microprocessors

Data processing.

Digital Computers

Basic arithmetic operations.

Embedded Systems

Microcontroller calculations.

Scientific Computing

Complex mathematical operations.

Arithmetic Unit vs ALU

Arithmetic Unit	ALU
Performs Arithmetic Operations	Performs Arithmetic + Logical Operations
Sub-part of ALU	Complete Unit
Addition, Subtraction	Addition, Subtraction, AND, OR, NOT
Limited Functions	More Functions

Viva Questions

Q1. What is Arithmetic Unit?

Circuit used to perform arithmetic operations.

Q2. What are the main components?

Full Adder and Multiplexer.

Q3. Which unit performs subtraction?

Arithmetic Unit.

Q4. Why is Multiplexer used?

To select arithmetic operations.

Q5. Is Arithmetic Unit a part of ALU?

Yes.

Frequently Asked RGPV Questions

2 Marks

1. Define Arithmetic Unit.
2. What is the role of Full Adder?
3. What is Multiplexer?
4. What is Increment Operation?

---

5 Marks

1. Explain Arithmetic Unit.
2. Explain Arithmetic Operations.
3. Explain role of Multiplexer.

---

7 Marks

1. Explain Design of Arithmetic Unit.
2. Explain Arithmetic Circuit with diagram.
3. Discuss arithmetic operations performed by Arithmetic Unit.

---

14 Marks

1. Explain Design of Arithmetic Unit with neat block diagram.
2. Discuss various arithmetic operations performed by Arithmetic Unit.
3. Explain Arithmetic Unit using Full Adders and Multiplexers.

PYQ Trend Analysis

Topic	Frequency
Arithmetic Unit Design	⭐⭐⭐⭐⭐

Arithmetic Operations Table	⭐⭐⭐⭐
Full Adder Based Design	⭐⭐⭐⭐⭐
Multiplexer Role	⭐⭐⭐⭐

Expected 2026 Questions

🔥 Explain Design of Arithmetic Unit with block diagram.

🔥 Explain arithmetic operations performed by Arithmetic Unit.

🔥 Discuss role of Full Adder and Multiplexer.

🔥 Explain Arithmetic Unit using selection variables.

One-Minute Revision

✅ Arithmetic Unit = Part of ALU

✅ Components:

- Full Adder
- Multiplexer
- Registers

✅ Operations:

- Addition
- Subtraction
- Increment
- Decrement
- Transfer

✅ Formula:

```
A - B = A + 2's Complement(B)
```

✅ Selection Variables:

```
S1, S0, Cin
```

Conclusion

Arithmetic Unit CPU ka ek important component hai jo binary arithmetic operations perform karta hai. Iska design Full Adders, Multiplexers aur Control Signals par based hota hai. Arithmetic Unit Addition, Subtraction, Increment, Decrement aur Transfer operations ko efficiently execute karta hai aur Computer Architecture me central role nibhata hai. 🎯

Exam Tip (RGPV 14 Marks)

Answer me ye sequence follow karo:

1. Definition
2. Need of Arithmetic Unit
3. Block Diagram
4. Components
5. Working
6. Arithmetic Operations Table
7. Role of Full Adder
8. Role of Multiplexer
9. Advantages
10. Applications
11. Conclusion

Is format me likhne par answer **3–4 pages** aasani se bhar jayega aur full marks ke chances bahut badh jayenge.