Lecture Two
Arithmetic Operations

A. Binary Arithmetic Operation

Binary arithmetic operates on base-2 numbers using only digits **0** and **1**. Understanding these operations is fundamental to computer architecture, digital logic design, and low-level programming.

1. Binary Addition

Rules

Operation	Result	Carry
0 + 0	**0**	**0**
0 + 1	**1**	**0**
1 + 0	**1**	**0**
1 + 1	**0**	**1**
1 + 1 + 1	**1**	**1**

Step-by-Step Process

1. Align numbers from rightmost bit (LSB)
2. Add bits column by column
3. Propagate carries to the next left column

Example 1: Simple Addition

Add 1011_2(11) and 1101_2 (13)

```
  1 0 1 1
+ 1 1 0 1
----------
Carry:   1 1 1
Result:1 1 0 0 0
```

Check: 11 + 13 = 24 ✓

Example 2: With Multiple Carries

Add 1111_2 (15) and 0001_2 (1)

```
  1 1 1 1
+ 0 0 0 1
-----------
Carry:1 1 1 1
Result:1 0 0 0 0
```

Check: 15 + 1 = 16 ✓

Example 2: Float
Add (11.01)2 and (101.11)2

```
          11.01
        + 101.11
         -----------
Result:  1001.00
```

---

2. Binary Subtraction

Rules

Operation	Result	Borrow
0 - 0	**0**	**0**
1 - 0	**1**	**0**
1 - 1	**0**	**0**
0 - 1	**1**	**1**

Step-by-Step Process

1. Align numbers from rightmost bit
2. If subtracting 1 from 0, borrow 1 from next left column
3. Borrowed bit becomes 2 in current column

Example 1: Simple Subtraction
Subtract 0101_2 (5) from 1001_2 (9)

```
  1 0 0 1
- 0 1 0 1
-----------
```

Step 1: 1 - 1 = 0
Step 2: 0 - 0 = 0 (but need to check borrow)
Step 3: 0 - 1 → borrow → 10 - 1 = 1

Step 4: 0 (after borrow) - 0 = 0

Result: 0 1 0 0_2 = 4
Check: 9 - 5 = 4 ✓

Example 2: Multiple Borrows
Subtract 00111_2 (7) from 1000_2 (8)

```
  1 0 0 0
- 0 1 1 1
-----------
```
Step-by-step:
Column 1: 0 - 1 → borrow → 10 - 1 = 1 (borrow from col2)
Column 2: 0 (after borrow) - 1 → borrow → 10 - 1 = 1 (borrow from col3)
Column 3: 0 (after borrow) - 1 → borrow → 10 - 1 = 1 (borrow from col4)
Column 4: 1 (after giving borrow) - 0 = 0

Result: 0 0 0 1_2 = 1
Check: 8 - 7 = 1 ✓

Example 3: Float
1 1 0 . 0 1 - 1 0 0 . 1 = (0 0 1 . 1 1)$_2$

---

3. Binary Multiplication

Rules

Operation	Result
0 × 0	**0**
1 × 0	**0**
0 ×1	**0**
1 × 1	**1**

Process

1. Multiply multiplicand by each digit of multiplier
2. Shift partial products left based on digit position
3. Add all partial products

Example 1: 2-bit × 2-bit

Multiply 101_2 (5) by 011_2 (3)

```
   1 0 1   (5)
 × 0 1 1   (3)
-----------
   1 0 1   (101 × 1)
  1 0 1    (101 × 1, shift left 1)
0 0 0      (101 × 0, shift left 2)
-----------
0 1 1 1 1   (15)
```

Check: 5 × 3 = 15 ✓

Example 2: Larger Numbers

Multiply 1101_2 (13) by 101_2 (5)

text

```
   1 1 0 1   (13)
 ×   1 0 1   (5)
-----------
   1 1 0 1   (1101 × 1)
  0 0 0 0    (1101 × 0, shift left 1)
1 1 0 1      (1101 × 1, shift left 2)
-----------
1 0 0 0 0 0 1 (65)
```

Check: 13 × 5 = 65 ✓

Example 3: Float

$(1 . 0 1)_2 \times (1 0 . 1)_2$

```
        1 0 1
        1 0 1
   ----------------
        1 0 1
     0 0 0
   1 0 1
-----------------
   1 1.0 0 1
```

4. Binary Division

Process (Similar to Long Division)

1. Compare divisor with first bits of dividend
2. If divisor ≤ current bits, put 1 in quotient and subtract
3. Bring down next bit
4. Repeat until all bits processed

Example 1: Simple Division

$(1111)_2 \div (101)_2$ =

```
             11
           ________
     101  | 1111
           -101
           ---------
            0101
              101
           ----------
              000
```

Example 2: With Remainder

Divide 1010_2 (10) by 011_2 (3)

```
     1 1   (quotient = 3)
   _______
011) 1 0 1 0
    -0 1 1   (first 3 bits: 101 - 011 = 010)
     ------
     1 0 0
    -0 1 1 (100 - 011 = 001)
     ------
      0 0 1 (remainder = 1)
```

Check: 10 ÷ 3 = 3 remainder 1 ✓

B. Arithmetic operations in other systems:

عند إجراء العمليات الرياضية الاعتيادية من المحتمل أن ينتج رقم أكبر أو يساوي أساس النظام، في هذه الحالة يتم قسمة الرقم الناتج على أساس النظام حيث يمثل باقي القسمة ناتج العملية وناتج القسمة هو **carry**يحمل إلى المرتبة اللاحقة .وفي عملية الطرح تتم الاستدانة من المرتبة الأعلى وقيمة الاستدانة تكون مساوية لأساس النظام المستخدم.

Ex: Perform the following operations: -

1. $(471)_8+(635)_8=(1326)_8$
2. $(2A4)_{16}+(CB4)_{16}=(F58)_{16}$
3. $(405)_8-(267)_8=(116)_8$
4. $(A85)16-(5D4)_{16}=(4B1)_{16}$
5. $(652)_{12}-(480)_{12}=(172)_{12}$

Practice Problems
Try these:

1. **Add: $101101_2+110011101101_2+110011_2$**
2. **Subtract: $11000_2-101111000_2-1011_2$**
3. **Multiply: $1101_2\times10101101_2\times1010_2$**
4. **Divide: $11110_2\div10111110_2\div101_2$**
5. **$(1010.01)_2\div(1.1)_2$**