Assignment Activity Unit 4

Department of Computer Science, UoPeople

CS 2203-01 - AY2025-T1

Instructor Ada Ajunwa

**3th October 2023**

# Designing a Digital Calculator for Binary Arithmetic Operations

In this assignment, we will explore the design and development of a digital calculator tailored to perform arithmetic operations on binary numbers. Our objective is to create modules for binary addition, subtraction, multiplication, and division, and to integrate these into a central processing unit (CPU) for seamless execution. This project aims to highlight the efficiency and flexibility of binary arithmetic in solving mathematical problems, as well as the benefits of understanding number systems in the broader context of computer science and engineering.

## 1. Design and Development of Binary Arithmetic Modules

The digital calculator will be constructed using four fundamental binary arithmetic modules: the binary adder, subtractor, multiplier, and divider. Each module will perform its respective arithmetic operation on binary inputs.

### 1.1 Binary Adder

A binary adder is designed to add two binary numbers and produce a sum and a carry-out value. We will utilize a **full-adder circuit**, which takes three inputs: two binary numbers and an incoming carry bit from the previous operation. The adder outputs the sum and a carry-out bit, which can be fed into subsequent adders in a chain for multi-bit addition. The sum (S) and carry-out (C) can be expressed as:

S=A⊕B⊕Cin​

Cout​=(A⋅B)+(Cin​⋅(A⊕B))

### 1.2 Binary Subtractor

The binary subtractor module will perform subtraction by using the concept of two's complement. We will invert the bits of the second operand and add 1, which allows us to transform a subtraction problem into an addition problem. The design is similar to that of the binary adder but incorporates a control signal to handle the two’s complement transformation.

### 1.3 Binary Multiplier

Binary multiplication can be visualized as a series of shifts and additions. We will implement a **shift-and-add algorithm**, which iterates through each bit of the multiplier, shifting the multiplicand and adding it to the result if the corresponding multiplier bit is 1. This operation is repeated for each bit position, resulting in a product that is efficiently computed.

### 1.4 Binary Divider

The binary divider module will be designed using the **restoring division algorithm**, which divides a binary dividend by a binary divisor to obtain a quotient and remainder. The algorithm repeatedly shifts and subtracts the divisor from the dividend and restores it if the subtraction results in a negative value. This ensures a precise computation of the division operation.

## 2. Integration and Organization of Arithmetic Modules

To ensure seamless operation of the calculator, the four binary modules will be integrated into a single CPU. The integration will involve creating a control unit that selects the desired operation based on user input. The control unit will manage data flow between the modules and ensure that each module receives the appropriate inputs and generates the correct outputs. Additionally, a memory unit will store intermediate results to facilitate complex calculations.

The calculator’s architecture will include the following components:

* **Arithmetic Logic Unit (ALU)**: Incorporates the binary adder, subtractor, multiplier, and divider, performing arithmetic operations based on control signals.
* **Control Unit**: Manages the selection of arithmetic operations and directs data flow between modules.
* **Input/Output Interface**: Allows users to enter binary numbers and receive results.
* **Memory Unit**: Stores intermediate values and results to support multi-step operations.

## 3. Examples of Binary Calculations

### 3.1 Binary Addition

Example: 1011(11)+1101(13)

 ​Carry:1110

Binary Addition:1011+1101=11000(24)​

### 3.2 Binary Subtraction

Example: 1010(10)−0110(6)

 ​Binary Subtraction:1010−0110=0100(4)​

### 3.3 Binary Multiplication

Example: 101(5)×11(3)

 Product 1) Product 2, shifted left)​101×11101(Partial Product 1)1010(Partial Product 2, shifted left)Result:1111(15)​

### 3.4 Binary Division

Example: 10100(20)÷101(5)

The result is a quotient of 100 (4) and a remainder of 0.

## 4. Advantages and Challenges of Binary Arithmetic

Binary arithmetic offers several advantages, including:

* **Simplified Circuit Design**: Binary systems rely on simple on-off states, making digital circuits easier to design and implement.
* **Reduced Ambiguity**: Unlike decimal arithmetic, binary arithmetic reduces the potential for ambiguity since it involves only two states (0 and 1).
* **Efficiency in Computation**: Binary arithmetic enables faster computation due to its straightforward nature and the efficient manipulation of binary bits.

However, challenges include:

* **Complexity in Representation**: Larger numbers can be cumbersome to represent in binary format compared to decimal.
* **Difficulty in Manual Calculation**: Manual calculations in binary can be more complex for individuals unfamiliar with the system.

## 5. Significance of the Calculator Design

Designing a digital calculator for binary arithmetic promotes a deeper understanding of number systems and fundamental arithmetic operations. By working with binary numbers, students gain insights into the inner workings of digital computers, which rely heavily on binary logic. This project bridges theoretical concepts and practical application, preparing students for advanced studies in computer science, digital electronics, and related fields.

In conclusion, the development of a digital calculator for binary arithmetic not only demonstrates the versatility and efficiency of binary operations but also serves as a valuable educational tool. The project emphasizes the importance of mastering number systems and lays a strong foundation for further exploration in digital systems and computational theories.