Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Fall, Year: 2025), B.Sc. in CSE (Day)

Banking Simulation

Course Title:Microprocessors, Microcontrollers and Embedded System Lab
Course Code:CSE 303
Section:232-D1

Students Details

Name	ID
Ashab Uddin	232002274
Rokunozzaman Topu	232002280

Submission Date: 02-12-2025
Course Teacher's Name: Jarin Tasnim Tonvi

[For teachers use only: Don't write anything inside this box]

Lab Project Status	
Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Overview

The **Banking Simulation** project is developed using 8086 assembly language. This system simulates the basic operations of a bank account, such as:

- Creating an initial account balance
- Depositing money
- Withdrawing money
- Checking available balance

The project is implemented in the EMU8086 emulator, which allows us to execute instructions at low level and visualize the working mechanism of microprocessor-based computations. Through this project, users can understand how simple banking operations are processed internally using registers, memory, loops, and arithmetic instructions.

1.2 Motivation

Banking systems today rely heavily on software to automate financial operations. Students usually learn these concepts using high-level languages like Python, Java, or C. However, many underestimate the power of assembly language to manage such real-world computations.

This project aims to show:

- how banking logic works at the lowest hardware-interaction level,
- how deposits, withdrawals, and balance checks are implemented without any built-in library,
- how microprocessors handle arithmetic and logic directly using registers and memory.

This system helps bridge the gap between theoretical microprocessor concepts and practical financial applications.

1.3 Problem Definition

1.3.1 Problem Statement

Modern banking applications automate balance updates, deposits, and withdrawals. But understanding these operations at the machine level is difficult for students. This project solves that problem by building a simplified banking model entirely using 8086 assembly language.

Main challenges include:

- implementing deposit and withdrawal logic with limited instruction sets,
- avoiding invalid memory access,
- ensuring correct arithmetic operations such as preventing negative balance.

1.3.2 Complex Engineering Problem

Table 1.1 summarizes the key engineering attributes addressed in this project.

Attribute	**Explanation**
Depth of knowledge	Requires understanding of registers, arithmetic instructions, and memory management.
Conflicting requirements	Needs simplicity while maintaining correctness of all banking operations.
Depth of analysis	Must analyze how deposits, withdrawals, and balance storage work efficiently.
Familiarity of issues	Students may face issues like underflow, incorrect jumps, and invalid memory access.
Applicable codes	Only 8086 assembly instructions can be used. No high-level features allowed.
Stakeholder requirements	Teachers expect a clear demonstration of microprocessor logic without unnecessary complexity.
Interdependence	Balance calculation, user input, and data storage are interconnected; failure in one breaks the system.

Table 1.1: Summary of attributes addressed by the Banking Simulation project

1.4 Design Goals/Objectives

The primary objectives of this project are:

- To design a simple banking simulation using 8086 assembly language.
- To implement realistic operations such as deposit, withdrawal, and balance checking.
- To strengthen understanding of microprocessor architecture.
- To demonstrate practical use of assembly language in real-world inspired applications.
- To improve debugging, problem-solving, and register-level programming skills.

1.5 Application

1.5.1 Educational Use

The system can be used by students to understand:

- low-level arithmetic operations,
- how basic banking logic works internally,
- how microprocessors execute financial calculations.

1.5.2 Embedded System Simulation

Devices like:

- ATM prototypes,
- Vending machines,
- Prepaid balance counters,

use similar low-level arithmetic concepts. This project can be a foundation for such embedded designs.

1.5.3 Understanding Banking Logic

Although simplified, this simulation includes the core logic behind:

- account management,
- deposit/withdraw processing,
- balance update mechanisms.

It helps beginners understand how financial operations work inside bank software.

1.5.4 Microprocessor Training Projects

Training institutes often use small but meaningful projects. This one covers:

- input handling,
- data storage,
- arithmetic processing,
- control flow,

making it ideal for microprocessor laboratory practice.