Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Fall, Year: 2024), B.Sc. in CSE (Day)

Multi-Base Converter Using 8086

Course Title: Microprocessors & Microcontrollers Lab
Course Code: CSE 304
Section: 223 D4

Students Details

Name	ID
Nadia Islam Rupa	223002047
Kawser Miah	223002063

Submission Date: 16/11/2024
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 Number System Converter project is a comprehensive application developed using the `emu8086` microprocessor emulator to demonstrate the practical capabilities of assembly language programming. The primary goal of this project is to enable users to convert numbers between various number systems, such as binary, decimal, octal, and hexadecimal, with precision and efficiency. By leveraging the fundamental operations of the 8086 assembly language, the program processes user inputs and performs conversions through well-structured computational logic. The project adopts a modular approach, where each conversion process is implemented independently, ensuring simplicity and maintainability. A user-friendly interface guides users to input a number, select its current format, and choose the desired target system for conversion. This project aims to bridge the gap between theoretical knowledge of number systems and their real-world implementation while serving as an educational resource for understanding low-level programming concepts. Additionally, it highlights the importance of assembly language in developing efficient computational solutions for foundational problems.

1.2 Motivation

The motivation behind choosing the Number System Converter project stems from the critical role number systems play in computer science and digital electronics. As a student of assembly language programming, understanding how computers handle and represent data in various formats is essential. While modern programming languages provide built-in functions for number system conversions, the opportunity to implement these operations at a low level using the `emu8086` emulator offers a unique learning experience.

This project was chosen to bridge the gap between theoretical knowledge and practical implementation. By working with assembly language, I aimed to gain deeper insights into the internal workings of conversions, such as binary to decimal or hexadecimal to octal. Additionally, this project allows for mastering the 8086 instruction

set, developing problem-solving skills, and fostering a hands-on understanding of low-level programming concepts. It also serves as a practical demonstration of how foundational knowledge can be applied to create efficient computational tools, emphasizing the relevance of assembly language in solving real-world problems. [**?**].

1.3 Problem Definition

1.3.1 Problem Statement

The primary problem addressed in this project is the lack of understanding of how number system conversions are performed at a low level. Modern tools abstract these operations, leaving students with limited insights into the computational logic involved. This project bridges that gap by implementing a Number System Converter in assembly language, enabling a deeper understanding of binary, decimal, octal, and hexadecimal conversions while showcasing the practical application of low-level programming using **emu8086**.

1.3.2 Complex Engineering Problem

The complex engineering problem addressed by this project includes several key challenges:

- **Limited Resources of Assembly Language:** The 8086 microprocessor offers limited registers and simple instructions, making number system conversions challenging. The project must efficiently utilize available resources without causing overflow or errors.
- **Handling Multiple Number Systems:** The conversion logic must handle binary, decimal, octal, and hexadecimal systems. Each system has unique characteristics, and the challenge is to implement accurate conversion algorithms within the constraints of assembly.
- **Efficient Computation:** Ensuring that the conversion process is both accurate and computationally efficient in a low-level programming environment, where performance optimization is crucial.
- **User Input and Output Management:** Designing a clear and simple user interface within the limitations of the emulator, where inputs can be easily given and results can be displayed without overwhelming the user.
- **Avoiding Errors in Conversion:** Preventing errors such as overflow and incorrect results, particularly due to the manual handling of memory and registers in assembly language.
- **Understanding and Implementing Assembly Logic:** The complexity of translating high-level number system conversion logic into assembly code while ensuring it is both accurate and efficient.

Table 1.1: Summary of the attributes touched by the mentioned project

Name of the Attributes	Explain how to address
P1: Depth of knowledge required	The project requires a thorough understanding of assembly language, the 8086 architecture, and number system conversion logic, which is crucial for handling low-level computational tasks.
P2: Range of conflicting requirements	Balancing user-friendly input handling with the constraints of 8086 assembly, such as limited registers and instruction sets, is a challenging requirement.
P3: Depth of analysis required	Analyzing how different number systems operate and translating that logic into efficient assembly code demands meticulous attention to detail.
P4: Familiarity of issues	Many students struggle to connect theoretical knowledge of number systems with practical assembly implementation, making this project highly relevant for bridging that gap.
P5: Extent of applicable codes	The project adheres to assembly language programming standards, using efficient instruction sequences and interrupts within the constraints of the 8086 emulator.
P6: Extent of stakeholder involvement and conflicting requirements	Stakeholder involvement is limited to user interaction, requiring simple input mechanisms and clear output without overwhelming technical complexity.
P7: Interdependence	Each module (e.g., binary-to-decimal conversion) is interdependent but operates as a standalone process, ensuring modularity and simplifying debugging.

1.4 Design Goals/Objectives

The primary goal of this project is to design and implement a Number System Converter using 8086 assembly language in the `emu8086` microprocessor emulator. The specific objectives of this project are as follows:

- **Conversion Functionality:** To develop a program that can accurately convert numbers between binary, decimal, octal, and hexadecimal systems. The converter will handle both integer and non-integer inputs, ensuring flexibility and comprehensive coverage of different number system conversions.
- **Efficiency and Optimization:** To ensure that the conversion process is performed in an efficient manner by optimizing the assembly code. This includes

minimizing the use of registers and memory while maximizing processing speed and accuracy.

- **User Interface:** To design a simple yet effective user interface that allows easy input of numbers and selection of the desired number system for conversion. The interface should be intuitive, even with the constraints of assembly language.
- **Error Handling:** To incorporate error detection and handling mechanisms, ensuring that incorrect inputs (such as invalid number formats) are managed appropriately, providing clear feedback to the user.
- **Educational Value:** To create a learning tool for students and beginners to understand number systems and their conversions at a low level. This includes demonstrating the use of assembly language to perform complex calculations and data handling tasks.
- **Code Modularity and Maintenance:** To design the program with modularity in mind, allowing for easy maintenance and future improvements. This will ensure that each conversion process (e.g., binary-to-decimal, hexadecimal-to-octal) is implemented in separate modules for clarity and ease of debugging.

These objectives collectively ensure that the project not only fulfills its functional requirements but also serves as an educational tool, offering insights into low-level programming and the internal workings of number systems.

1.5 Application

The Number System Converter project has significant practical applications in various fields, particularly in computer science, digital electronics, and embedded systems. Understanding and converting between different number systems is foundational to these areas, and the implementation of this project provides a concrete example of how such conversions can be performed at the hardware level. Below are some of the key real-world applications of this project:

- **Computer Architecture and Digital Electronics:** Number system conversions are fundamental to computer operations, especially in the areas of memory management, data storage, and processing. Binary numbers are the core language of computers, while hexadecimal is commonly used for efficient representation of binary data. This project can be used as an educational tool to help students and professionals understand how computers work with different base systems at the hardware level.
- **Assembly Language Programming:** In low-level programming, especially when working directly with hardware or embedded systems, assembly language is often used to optimize critical operations. The number system converter is a useful demonstration of how to implement core functionality like arithmetic operations and data conversion using assembly. This has applications in embedded system development, firmware programming, and microcontroller programming.

- **Software Development and Debugging:** Many programming environments and compilers require the use of different number systems for debugging and testing, such as when working with memory addresses (hexadecimal) or bitwise operations (binary). The knowledge and implementation of a number system converter can assist developers in debugging code that involves bit manipulation or low-level data encoding and decoding.

- **Networking and Cryptography:** In networking, addressing schemes often use hexadecimal for IP addresses or MAC addresses, while binary is essential in protocol design, data transfer, and error checking. The ability to convert between these systems is essential for network engineers and cybersecurity experts who deal with low-level packet inspection or cryptographic algorithms that require precise control over data representation.

- **Computer Graphics and Digital Media:** In graphics programming, colors are often represented using hexadecimal codes, and working with pixel data may require converting between various number systems. For example, RGB values for colors are typically expressed in hexadecimal, while bitmap images may use binary for pixel data encoding. Understanding and applying number system conversions can be crucial for media developers working at a lower level in graphics engines or video processing.

- **Learning Tool for Education:** As an educational project, this converter serves as an invaluable resource for teaching the fundamentals of number systems, assembly language programming, and computational theory. It helps students visualize how data is represented in various number formats and how these conversions occur at the hardware level, enhancing their overall understanding of computer science principles.

In conclusion, the Number System Converter project is not just a theoretical exercise; it has broad applications in both the real-world technical domains and educational settings. Its implementation provides valuable insights into computer systems, data representation, and low-level programming, making it an essential tool for both aspiring and experienced professionals.