Question 2: Collatz Sequence Steps Calculation - MIPS Assembly Code

Bratin Mondal (21CS10016)
Somya Kumar (21CS30050)

August 16th, 2023

1 Introduction

This document provides an overview of an assembly code implemented in MIPS that calculates the number of steps required to reach the termination point of the Collatz sequence for a given positive integer n . The code demonstrates recursive function calls and basic arithmetic operations in MIPS assembly language.

2 Functionality

The MIPS assembly code presented here calculates the number of steps needed to reach the termination point of the Collatz sequence for a given positive integer 'n'. The Collatz sequence operates as follows:

1. If the number is even, divide it by 2.
2. If the number is odd, multiply it by 3 and add 1.

The code showcases the use of recursive function calls, user input and output, and basic arithmetic operations in MIPS assembly language.
The program prompts the user to input a positive integer 'n'. It checks for valid input (i.e., a positive integer) and displays a warning message and exits if an invalid input is entered. If the input is valid, the program calculates the number of steps needed to reach the termination point of the Collatz sequence for the input 'n'. The result is then displayed to the user.

3 Recursive Approach

The calculation of the number of steps in the Collatz sequence involves a recursive function that calculates the next value in the sequence based on whether the current value is even or odd.

If we define the number of steps to end reach the end of the Collatz Sequence as $f(x)$ for a positive integer x, then we can write

$$f(x) = \begin{cases} 1 + f(\frac{x}{2}), & x \text{ is even} \\ 1 + f(3x+1), & x \text{ is odd} \\ 0, & x \text{ is } 1 \end{cases}$$

3.1 Base Case

The function terminates when the current value becomes 1. The base case is when the current value becomes 1; in this case, the function returns 0 (indicating no more steps are needed).

3.2 Recursive Calls

For values greater than 1, the function calculates the next value using the Collatz sequence rules and recursively calls itself with the next value. The sum of these recursive calls represents the total steps needed to reach the termination point.