MIPS SIMULATOR

202101697 Lebona N.J
202101683 Nchoba S.J
202100027 Nketu F
202100119 Motsabitsabi K
202101692 Rasetoromo M.J
202101716 Sethojana L
202101694 Mpatlise T.J

Introduction

A simulator is an application that helps computers to run programs. This paper discusses an implementation of a web based simulator that takes inspiration from MIPS32 simulators.

Baic front-end tool(HTML,CSS,JavaScript) were used to create a basic MIPS assembly code simulator. The simulator allows users to input MIPS assembly code into a text area, and upon clicking the "Assemble" button, the code is processed by a JavaScript class called "Assembler." The Assembler class tokenizes and converts the MIPS assembly instructions into machine code. This machine code is then displayed in the output section.

The Assembler class includes a mapping for opcodes and registers, enabling the translation of human-readable MIPS assembly instructions into their binary machine code representations. The code handles various MIPS instructions such as "add," "sub," "div," "beq," "LW" (load word), "SW" (store word), and "jump." Each instruction is associated with a specific opcode and function code, ensuring accurate translation.

Instruction Opcodes

Instruction	Opcode
add	001100
sub	001101
div	001110
beq	001000

Testing

This is the demonstration of Assembling part of the simulator. It accepts a Assembly code in the text area and provides a corresponding Machine Code

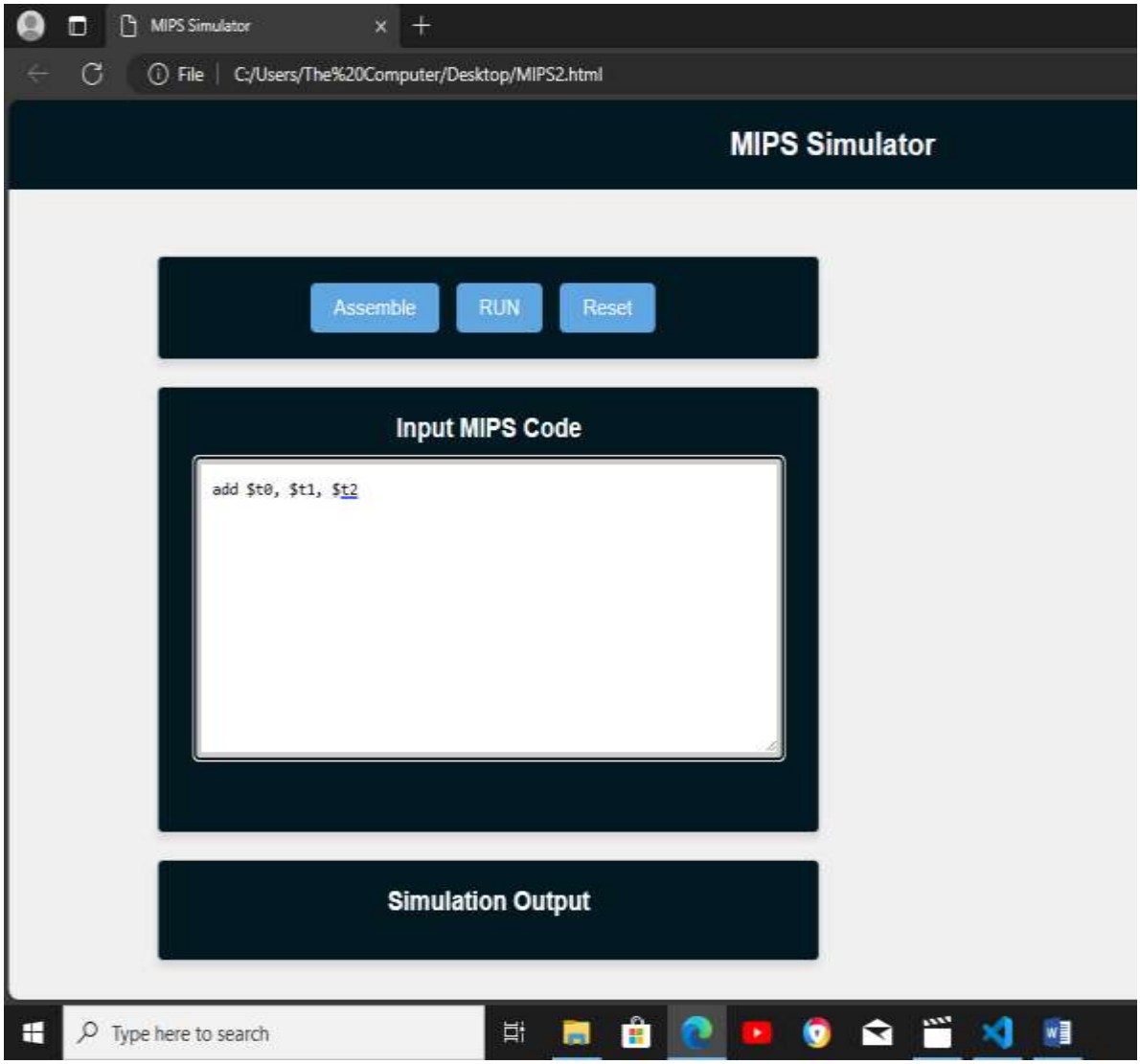
Instruction Example:
add $t0, $t1, $t2

Opcode for add instruction: **001100**
Register mappings:
$t0: **0000**
$t1: **0001**

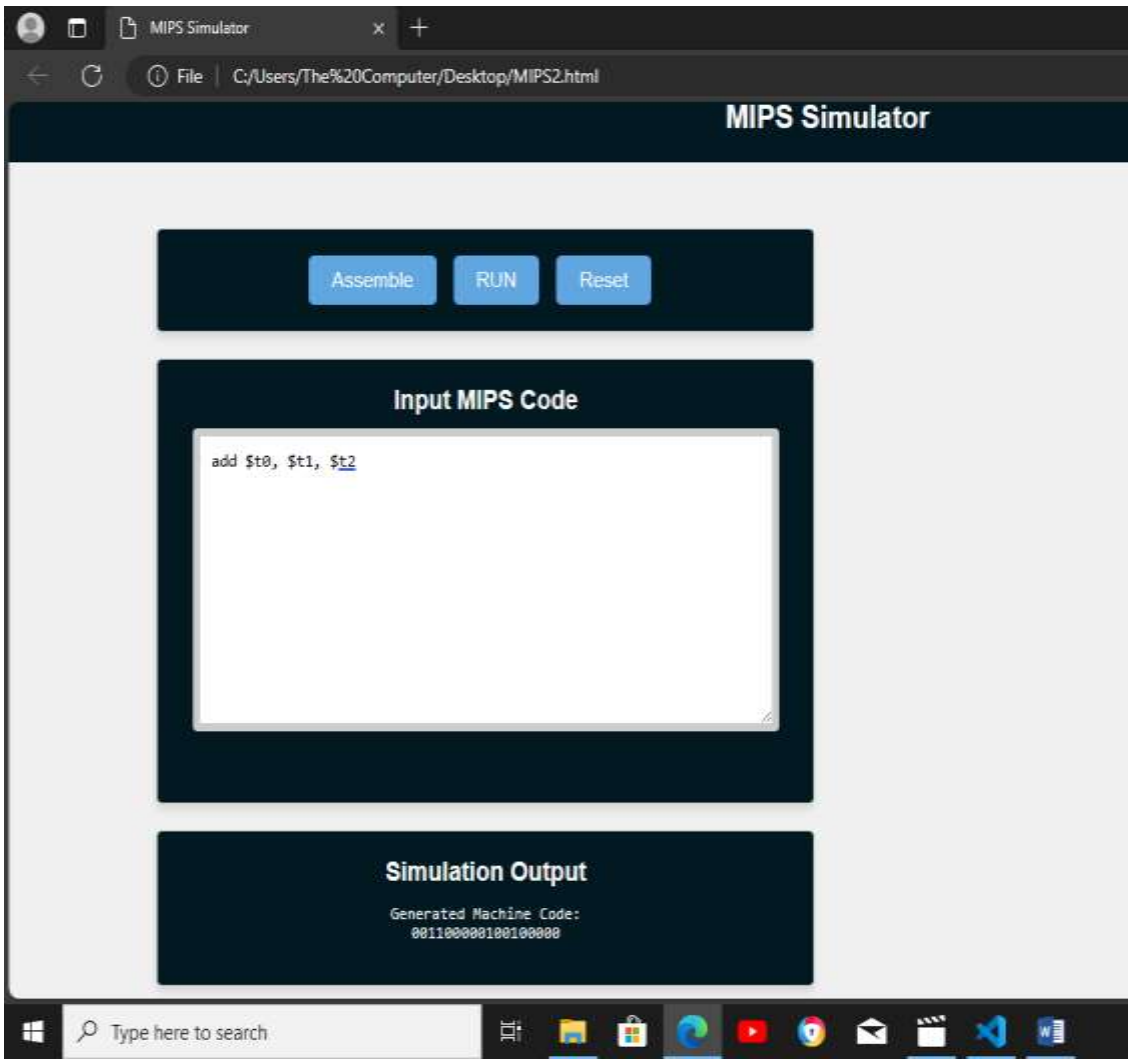
$t2: **0010**

The expected machine equivalent for the instruction;

00110000100100000



The generated machine code:



Conclusion

The assembler works as expected however the simulator, as it stands, lacks the implementation of the "Run" button and any execution logic for the generated machine code. The core functionality lies in the assembly of MIPS code to machine code. Future enhancements may involve integrating a MIPS processor simulation, execution of the generated machine code, and providing a comprehensive MIPS simulation experience.