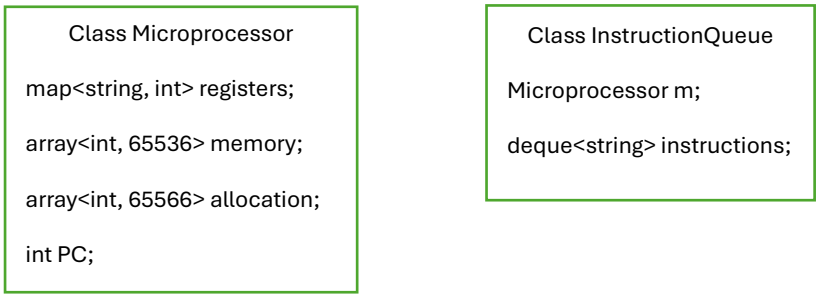
Emulating a Simple 16-Bit Microprocessor

Class Diagram



Emulation Details

The Microprocessor is emulated by using:

Class Microprocessor

Data Members

1. Map for registers (AX, BX, CX and DX)
2. An int array of size 65536 to emulate memory, considering each byte as an int.
3. An int array to keep track of allocation
4. A Program Counter variable

Function Members

1. bool isRegister(string);
    - To check if the operand is a register
2. bool isMemoryAddr(string);
    - To check if the operand is a memory address
3. bool isDirect(string);
    - To check if the operand is a direct value
4. int readRegister(string);
    - To read the register
5. int readMemoryAddr(int);
    - To read the memory address
6. bool writeRegister(string, int);
    - To write to the register
7. bool writeMemoryAddr(int, int);
    - To write to the memory address
8. void displayRegisters();
    - To display the registers
9. void displayMemory();

- To display the first 16 “bytes”/blocks of memory

10. int getAddr(string);
    - To remove the ‘[‘ and ‘]’ from the memory address and return it in int format
11. Microprocessor();
    - Constructor to initialize the data members
12. void setPC(int);
    - To set the Program Counter
13. int getPC();
    - To get the Program Counter
14. bool mov(string, string);
    - For the “MOV” operation
15. bool add(string, string);
    - For the “ADD” operation
16. bool sub(string, string);
    - For the “SUB” operation
17. bool mul(string, string);
    - For the “MUL” operation
18. bool div(string, string);
    - For the “DIV” operation
19. void hlt();
    - For the “HLT” operation and to call the display function

Class InstructionQueue

Data Members:

1. Microprocessor m;
    - An object of class Microprocessor
2. deque<string> instructions;
    - Deque to store the instructions

Function Members:

1. array<string, 3> tokenizeInstruction(string);
    - To tokenize the instruction string into 3 parts (Except for the HLT instruction)
2. bool addInstruction(string);
    - Push Back the instruction to the queue
3. void displayInstructions();
    - Display all the instructions in the queue

4. void execute();
    - Execute the tokenized instruction, by calling the appropriate functions of object m

Overview

1. The program reads the lines from “instructions.txt”, and pushes it to the deque member of the InstructionQueue class.
2. Then, the execute() function of the InstructionQueue class, tokenizes the string, and calls the appropriate function members of the Microprocessor class to emulate the instruction execution.
3. The Microprocessor operation functions carry out their operations using the private member functions to read and write to the registers and memory.
4. In cases where, the value to be written to the memory exceeds 256, the value is split across multiple free blocks, if available.
5. On encountering, the HLT instruction, the display functions are called