**Documentation for the program**

The program is designed to convert the given assembly language instruction to machine code which is then understood by the computer system. The scope of the program is limited to 15 instructions for the 8086/8088 microprocessor. We have considered eight 8 bits registers and 16 memory locations for simplicity. The program can translate the given instruction as well as update and display the contents of the registers and memory locations. The program is written in python while the front-end GUI is made using pygame. Tkinter window is used as well. Dictionaries are made to store the values of opcodes for the instructions, contents of registers and memory locations, and codes for registers.

**How does it work?**

Cases are made for each instruction. In each instruction, we can perform the function for a register to register, register to a memory location, and memory location to register. When the instruction is given, the program splits it to get opcode and operands, one or two, depending on the instruction. Respective values are then accessed from and updated in the dictionaries. In the case of MUL, IMUL, DIV, and IDIV, no update is made in the contents of the registers. Instructions with memory locations used as operands are only valid with the opcodes that take two operands else not.

Github link: https://github.com/abdullah-aleem/86-88Simulator