Most of the implemented operations are implemented in the ISA class, and added the trap instruction. But I added a Cache class. This class simulates the cache part, using two lists to represent the address list and the content list. At the same time, it also realizes the function of reading and writing. This class also has a flush function that clears content currently in the cache.

The Main GUI class has input widgets for hexadecimal and binary values. This is where you load the program to start implementing it. Inputs can be loaded into your choice of GPR registers, IXR registers, PC, MAR, MBR, IR, and CC.

Debug GUI class allows the user to set the CPU speed, print or flush cache, load, or store addresses or content. It also prints out console output with the instructions and contents and mentions whether the cache was hit or missed.

The I/O GUI class allows the user to enter a value on the console, and it reads the entered number one at a time. The print will output results based on the values ​​you enter.