

Computer Architecture

CPU simulator-Part 2

I. Introduction

In Part 1, we programmed a CPU simulator that can execute few instructions (STR, LDR). In Part 2, we realized all the Load/Store, Transfer, Arithmetic and Logical instructions. We also designed the cache and memory banking. At the same time, we revised the UI to make it more utility and friendly. Our simulator has three kinds of working mode, which are “Run”, “debug” and “SingleStep”. For “Run” and “Debug” mode, the CPU runs all the instructions stored in memory. However, there are differences between these two modes. Besides running all the instructions, CPU running in “Debug mode” prints the log of every instruction. When running on “SingleStep” mode, the CPU execute only one instruction.

II. Program design

Our source code consists of eight packages, and each of them realizes different function:

Name of Packages	Function
edu.gwu.computerarchitecture.alu	Realize Arithmetic Logical Unit
edu.gwu.computerarchitecture.cache	Realize cache operation
edu.gwu.computerarchitecture.cpu	Realize CPU operation
edu.gwu.computerarchitecture.entity	Define register and memory
edu.gwu.computerarchitecture.main	Entrance of the entire program
edu.gwu.computerarchitecture.operation	Define instructions, and Realize fetch, decode, writeback
edu.gwu.computerarchitecture.operation.execute	Execution of each instructions
edu.gwu.computerarchitecture.ui	Design the UI of the program

III. UI

We revised the UI, and now it is consist of more interfaces:

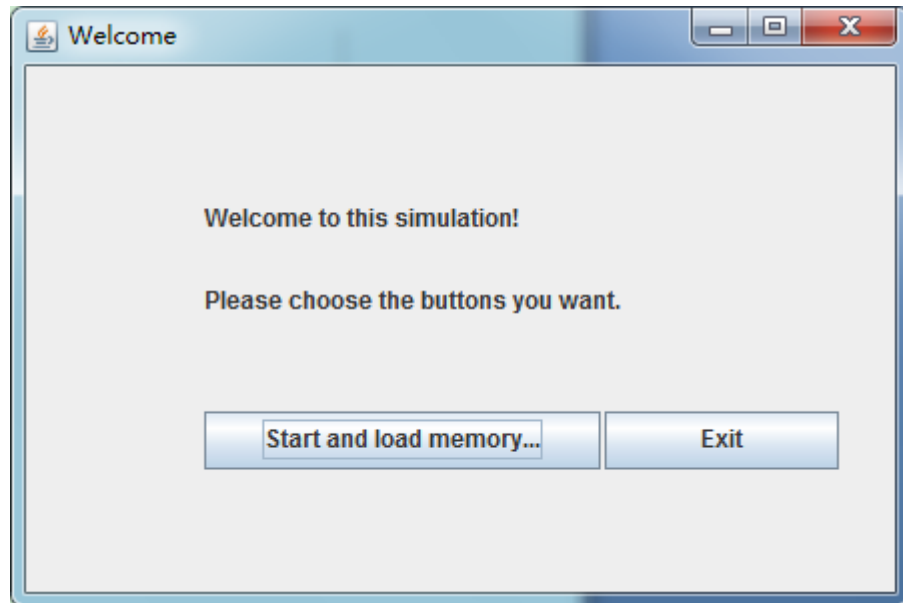


Fig.1 start the simulator

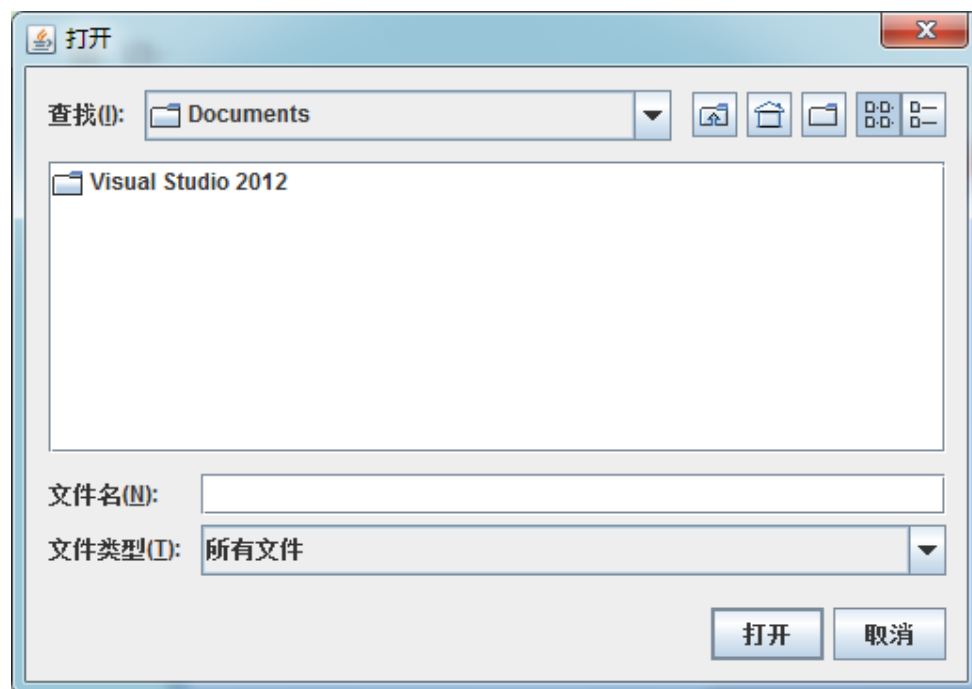


Fig.2 choose file to load for memory

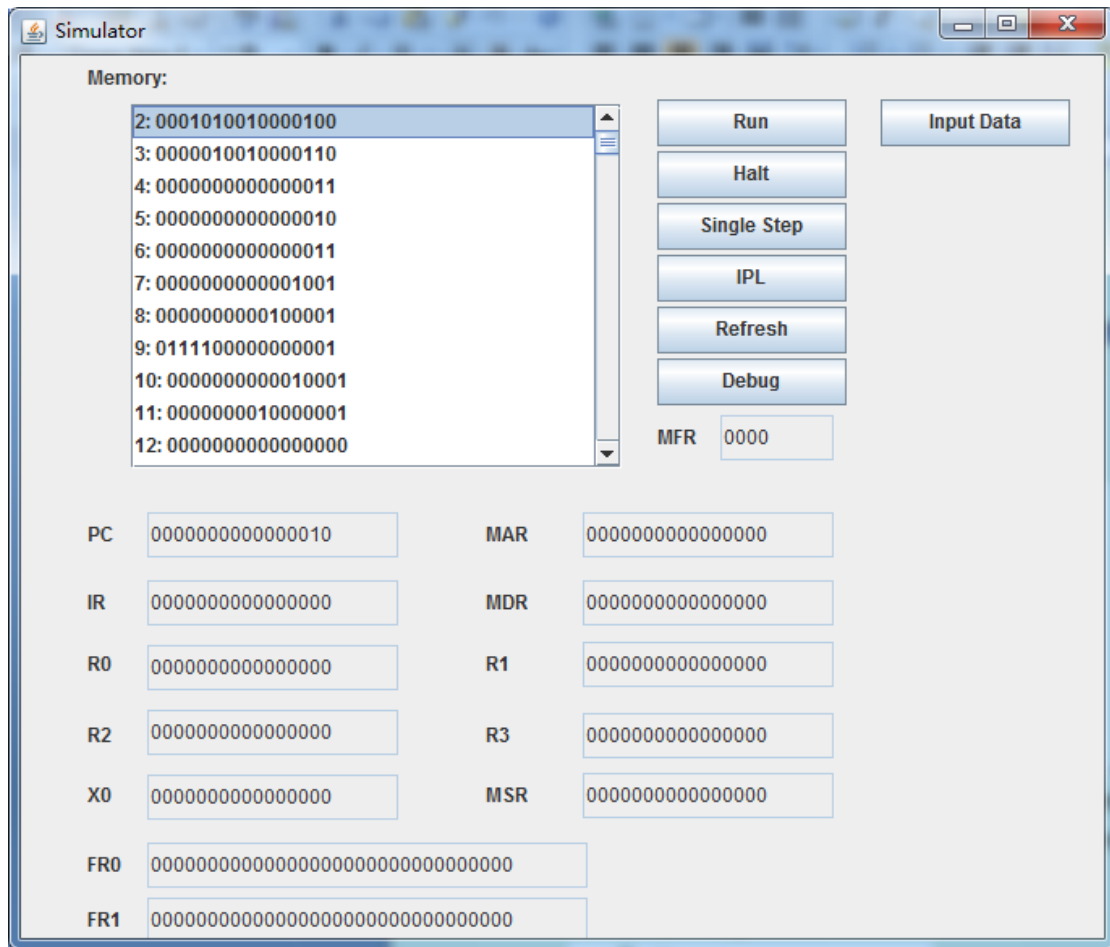


Fig.3 the panel of the simulator

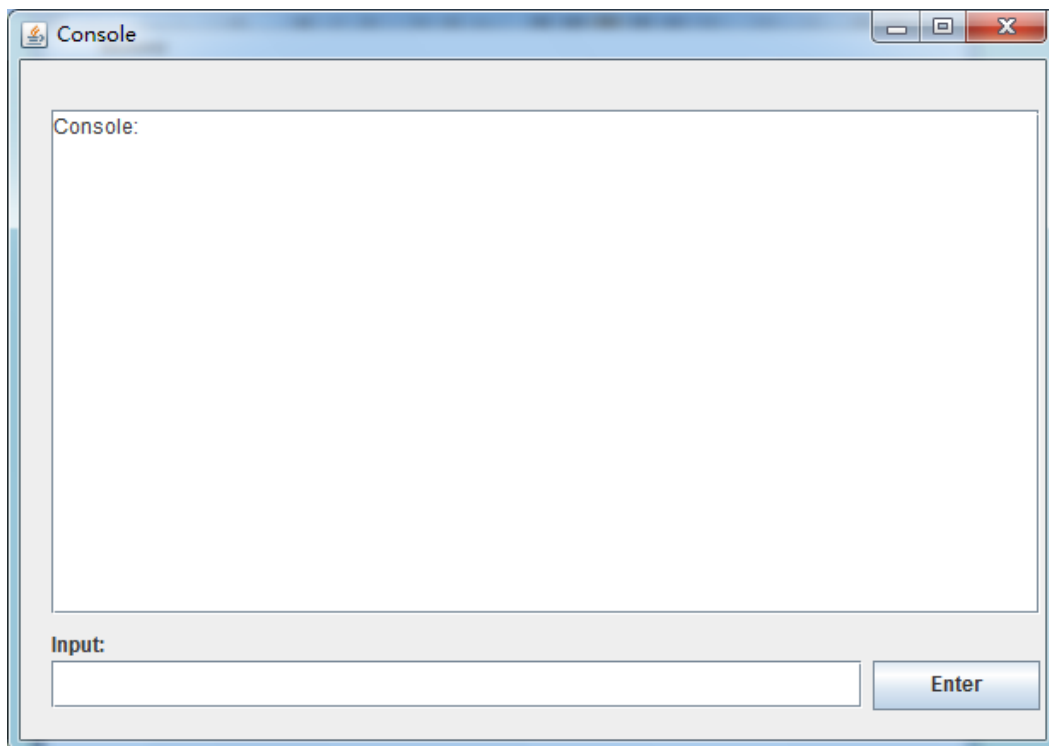


Fig.4 console for debug printing