Dokuz Eylul University
Computer Engineering

DEU Electronic Universal

Automatic Reduced Computer (DeuARC) Simulator

Prepered by...
2014510013 Seren BOLAT
2014510023 Anıl DURSUN
2014510060 Merve RUMELLİ

29.05.2017

CONTENTS

Describtion

In this project, we are expected to design and develop a simulator for a basic computer with reduced instruction set using object oriented programming paradigm. We use Java programming language and Swing widget toolkit will to implement DEUARC has 9 registers and 3 memory segments .

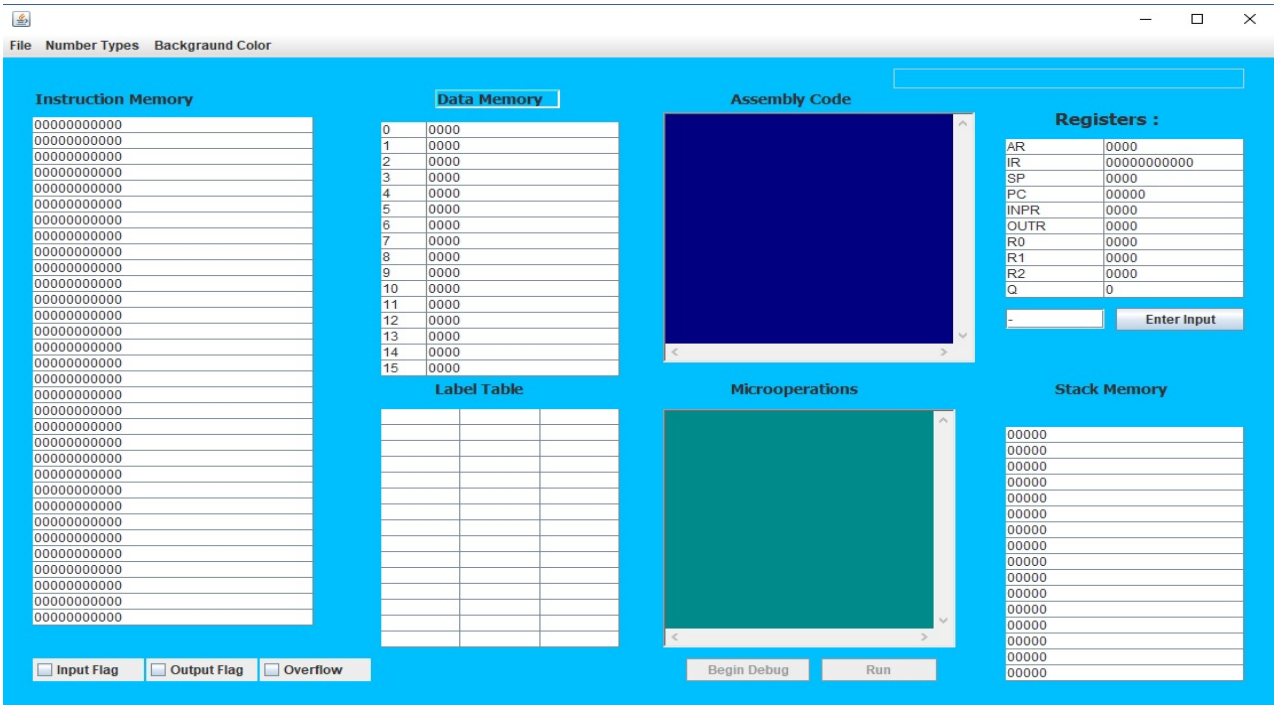
Registers are Address Register, Program Counter, Stack Pointer, Input Register, Output Register, Instruction Register and 3 general purpose registers. Each register has data, load, clear and clock inputs. Additionally, program counter has increment input and stack pointer has increment and decrement inputs.

Memory has three segments which are instruction, data and stack memory segments. Each has read enable and data inputs. Data and stack memory segments have write enable input as an extra. Also DEUARC has control unit and ALU (arithmetic logic unit). Control unit processes instructions to direct the micro-operations for computer's memory, registers and arithmetic/logic unit. ALU operates arithmetic and logic operations such as ADD and AND.

DEUARC simulator converts the assembly code to machine code and simulates the program execution phases. It runs the program step by step while showing the phases of instruction cycle (fetching, decoding, execution). DEUARC simulator reads and parses the assembly code, then shows its label table and memory content table. The simulator shows contents of the registers, memory segments, computer operations and their micro operations. It can simulate all operations that DEUARC supports (Table 1). It provides switching between binary / hexadecimal / decimal numbers and exporting hex or mif file of the machine code (HEX code or binary code).

The input files (asm or basm file) include assembly (symbolic) codes. The assembly language of the basic computer is defined by a set of rules. Data is indicated by ‘#’ and address is indicated by ‘@’ characters. An example for assembly code is given in the Code 1. Each line of the language has three columns called fields.

GUI



Instruction Memory

In Instruction memory the table shows us the converted assembly codes in binary representation.Each row shows Instruction Memory cell.If it is wanted,The table show's in Hex or Dec format.

Data Memory

In Data memory the table shows us the converted assembly data codes in binary representation.Each row shows Data Memory cell.If it is wanted,The table show's in Hex or Dec format.

Lable Table

This table shows us data's in Dec format to easly see.

Assambly Code

Our Assambly Code presentation

Microoperations

Show microoperations step-by-step.

Stack Memory

In Stack memory the table shows us the converted assembly Stack Memory in Dec representation.Each row shows Data Memory cell.

Registers

Table in which each register change is observed.

File

A menu where we can select the file asm or basm .

Number Types

Select the type of number’s presentation(BIN,HEX,DEC).

Weekly Progress

First&&Second Week

Project members try to finish GUI Programming using Java Swing but,encountered a lot of problems to use swing.First week The GUI finished without missing.

Third&&Fourth Week

The parsing operations and class design was completed in second week.Third week class design and parsing was updated and debugging.

Last Week

We hope the project will be finished in last week but we encountered some problems which non-useless.

Problem Encountered

The only problem encountered during the project was updating the requirements of the project every week - which meant that the project would change very frequently, which meant more time was needed to meet these requirement.
Unfortunately, no extra time was brought by these requirements, which meant that some of them could not be met.

Conclusion

As a conclusion, the project was finished.The harmony and order between the project members was an amusing project because of the perfect diverse project environment and peace of mind. No major problems were encountered except that the project owners were out of order of the project. The project was delivered on time at the delivery date.