Component Based MMIX Simulator using Multiple Programming Paradigms

Stephen Edmans

Contents

List of Figures

1 Introduction

There is a prevailing design paradigm to break systems up into smaller components The idea behind this paradigm is that these components can be swapped out for alternatives when the need requires. They can also be developed in parallel and potentially by separate teams of developers. There is now a proliferation of

2 Existing Implementations

3 Outline Design

The MMIX simulator application will consists of three separate components. The task of converting the MMIX assembly language source text into the corresponding binary representation will be performed by a component I am calling the Assembler. The task of actually simulating an MMIX computer will be performed by a component that I am calling the Simulation Engine. The final component will be responsible for representing the current state of the MMIX computer to the user, along with orchestrating all of the interactions with the other components. I am calling this component the User Interface. The interactions between the components can be illustrated by the .

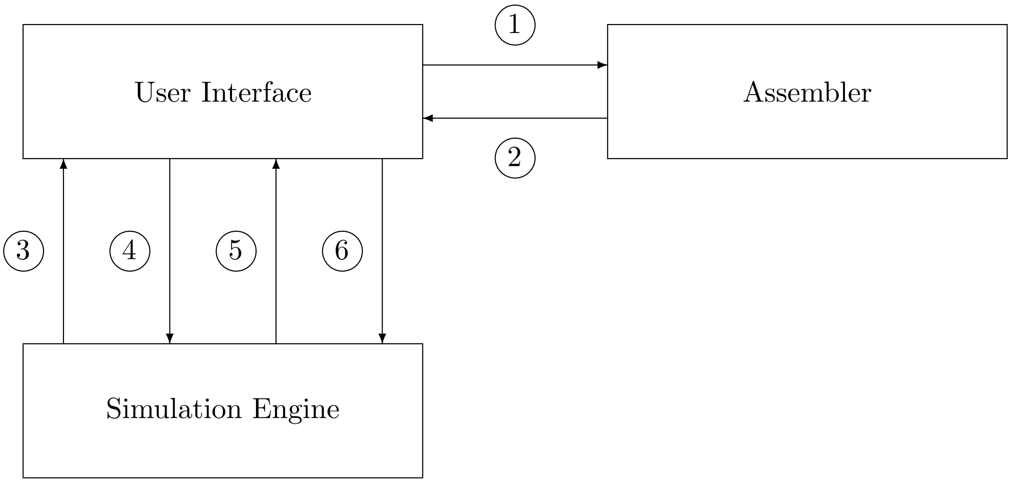

Figure 1: Component Interaction

1. Source Text
2. Binary Representation
3. Binary Representation
4. Current State

5. Process Next Step
6. Change of State

3.1 Assembler

3.2 Simulation Engine

3.3 Graphical User Interface

The main way that users will interact with the simulator is through a graphical user interface (GUI). The GUI will be responsible for all of the interactions with the other components. The GUI can be broken up into 4 separate sections as illustrated in figure

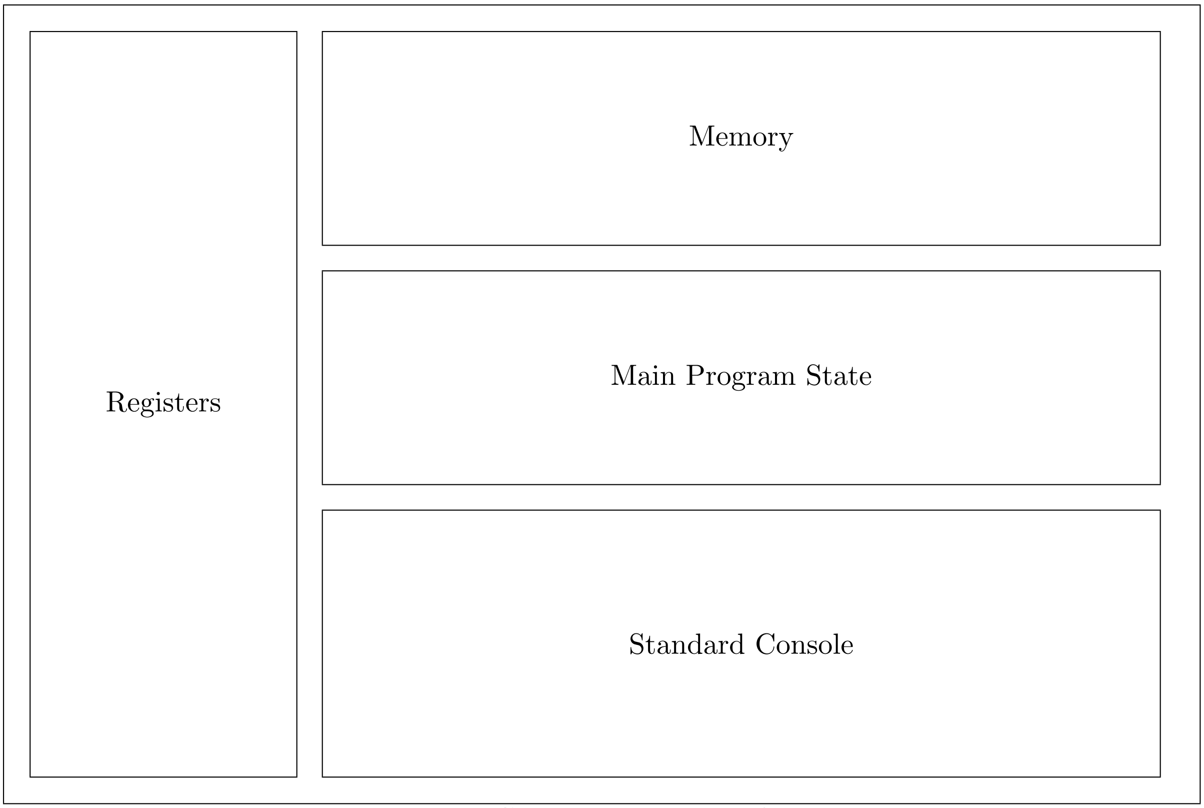

Figure 2: Graphical User Interface