User Guide: Simple Memory Implementation

CSCI 6461: Computer System Architecture Team 10

February 27, 2025

1 Overview

This document serves as the user guide for the simple memory module implemented for the CS6461 project simulator. The memory module simulates a single-port memory containing 2048 words, each 16-bit wide. This guide explains how to compile, run, and test the memory module, and describes its key operations.

2 System Requirements

- **Java Development Kit (JDK)**: Version 1.8 or later.
- **Development Environment**: Any Java IDE (e.g., Eclipse, NetBeans) or a text editor with command line tools.
- **Operating System**: Windows, macOS, or Linux.

3 Installation and Compilation

3.1 Compiling the Code

1. Download the `Memory.java` file.
2. Open a terminal or command prompt and navigate to the directory containing `Memory.java`.
3. Compile the code using the command:

```
javac Memory.java
```

4. Upon successful compilation, a `Memory.class` file will be created.

3.2 Creating an Executable JAR File (Optional)

1. Create a manifest file named `manifest.txt` with the following content:

```
Main-Class: Memory
```

2. Package the compiled class into a JAR file:

```
jar cfm Memory.jar manifest.txt Memory.class
```

3. Run the JAR file with:

```
java -jar Memory.jar
```

4 Running the Memory Module

4.1 Test Execution

The `Memory` class includes a `main` method that performs a basic test:

- Reads and displays the initial value at memory address 0 (expected to be 0).
- Writes a sample value (e.g., 12345) to memory address 0.
- Reads and displays the value at memory address 0 after the write operation.

4.2 Expected Output

Running the program should produce output similar to:

```
Initial value at address 0: 0
Value at address 0 after write: 12345
```

5 Memory Operations

The memory module supports the following operations:

- **readWord(int address)**: Returns the 16-bit word stored at the specified address.
- **writeWord(int address, short data)**: Writes a 16-bit word to the specified address.
- **reset()**: Clears the memory by setting all memory locations to zero.

All operations include bounds checking to ensure that the address is within the valid range (0 to 2047).

6 Troubleshooting

6.1 Compilation Issues

- Verify that JDK 1.8 or later is installed.
- Ensure that the file is named `Memory.java` and is in the correct directory.

6.2 Runtime Issues

- An `IllegalArgumentException` indicates an attempt to access an address outside the valid range (0 to 2047). Check the address values being used.
- Ensure your system meets the required specifications for running the simulator.

7 Conclusion

This user guide outlines how to compile, run, and test the simple memory module, a critical component of the CS6461 project simulator. For further details, consult the design notes and additional project documentation. The memory module is designed to be robust and easy to integrate with the overall simulator, and serves as a solid foundation for executing load and store instructions.