**Experiment Project Documentation**

**Introduction:**

This document captures the technical details related to the experiment development

**Project**

**Domain Name :**  Computer Science & Engineering

**Lab Name :** Computer Organization

**Experiment Name :** ARM Simulator

An Assembly language ( or assembler language), often abbreviated as **asm**, is any low-level programming language in which there is a very strong correspondence between the instructions in the language and the architecture’s machine code instructions. Assembly language may also be called “symbolic machine code”.

Assembly code is converted into executable machine code by a utility program referred to as assembly, as in assembling the source code. Assembly language usually has one statement per machine instruction, but comments and statements that are assembler directives, macros and symbolic labels of program and memory locations are often also supported

ARM, is a family of reduced instruction set computing(RISC) architectures for computer processors, configured for various environments.

With over 100 billion ARM processors produced as of 2017, ARM is the most widely used instruction set architecture and the instruction set architecture produced in largest quantity. Currently, the widely used Cortex cores, older classic cores, and specialized SecurCore core variants are available for each of these to include or exclude optional capabilities.

**Purpose of the project**

The purpose of the project is to convert the ARM Simulator project from **Java** to **Javascript**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Name** | **Year of Study** | **Role** | **Email-ID** | **github handle** |
| 1 | Poojith Chandra | IV Year |  | [poojithchadra@gmail.com](mailto:poojithchadra@gmail.com) | github.com/Noobie777 |

**Project Developer Details**

**Technologies and Libraries:**

1. **HTML**
2. **CSS**
3. **Javascript**

**Development Environment**

**OS:** Windows 10

Documents:

|  |  |  |
| --- | --- | --- |
| S.No | Link to Document | Role |
| 1 | Procedure | This document captures the instructions to run the simulations |
| 2 | Test Cases | This document captures the functional test cases of experiment simulation |
| 3 | Code Documentation | This document captures the details related to the code |

**Process Followed to convert the experiment**

1. Understand the assigned experiment Java simulation
2. Understanding the experiment concept
3. Re-implement the same in javascript

**Value Added by our Project**

1. It would be beneficial for engineering students
2. Highly beneficial for tier 2 and tier 3 college students who can use this to learn and understand the concept of perception learning.

**Risks and Challenges**

Understanding the Java simulation to figure out the address allocation of registers

**Issues :**

The project was very unresponsive in it’s initial stages, took way longer to load.