**COMP 380**

**Team 1**

**Team OakSim**

**Jesus Barahona**

**Jospeh Winthrope**

**Peter Jones**

**Qingxin Wang**

**Sean Lee**

**Software Requirements Specification**

**Document**

**Version: 1.0** **Date: (2/01/2017)**

**Table of Contents**

**1. Introduction**

*1.1 Purpose*

*1.2 Scope*

*1.3 Definitions, Acronyms, and Abbreviations*

*1.4 References*

*1.5 Overview*

**2. The Overall Description**

*2.1 Product Perspective*

2.1.1 System Interfaces

2.1.2 Interfaces

2.1.3 Hardware Interfaces

2.1.4 Software Interfaces

2.1.5 Communications Interfaces

2.1.6 Memory Constraints

*2.2 Product Functions*

*2.3 User Characteristics*

*2.4 Constraints*

*2.5 Apportioning of Requirements*

**3. Specific Requirements**

*3.1 External interfaces*

*3.2 Functions*

*3.3 Performance Requirements*

*3.4 Software System Attributes*

3.4.1 Maintainability

3.4.2 Portability

*3.5 Organizing the Specific Requirements*

3.5.1 System Mode

3.5.2 User Class

3.5.3 Objects

3.5.4 Stimulus

3.5.5 Response

3.5.6 Functional Hierarchy

**4. Change Management Process**

# 1. Introduction

## 1.1 Purpose

This SRS is being maintained in order to catalogue product requirements, and focus group efforts and resources towards the total fulfillment of our agreed design specification. For internal group use, editing of this document after final group approval is only to be undertaken after full agreement of the development team.

## 1.2 Scope

The development team will be producing OakSim, a cross-platform ARM emulator. OakSim is a Javascript web application capable of being hosted on a website, or downloaded, saved, and utilized locally. It allows users to simulate the execution of ARM assembly language programs, utilizing both the 16-bit (THUMB) and 32-bit (ARM) instruction sets, on a system based on the ARM7TDMI processor. OakSim includes both an assembler and also provides features not often found in similar applications. They enable users both to debug ARM assembly programs and to monitor the state of the system’s context while a program executes. The monitoring information includes both cache states and timing information.

## 1.3 Definitions, Acronyms, and Abbreviations.

ARM7TDMI - The ARM7TDMI (ARM7 + 16 bit Thumb + JTAG Debug + fast Multiplier + enhanced ICE) processor implements the ARMv4 instruction set. It is primarily a group of older 32-bit RISC ARM processor cores licensed by ARM Holdings for microcontroller use.

GAS (GNU assembler syntax) - The GNU Assembler, commonly known as gas or simply as, its executable name, is the assembler used by the GNU Project. It is the default backend of GCC. It is used to assemble the GNU operating system and the Linux kernel, and various other software.

GNU( GNU Not Unix) - The GNU Project was launched in 1984 to develop a complete Unix-like operating system which is free software: the GNU system. (GNU is a recursive acronym for "GNU's Not Unix") Variants of the GNU operating system, which use the kernel Linux, are now widely used; though these systems are often referred to as "Linux", they are more accurately called GNU/Linux systems.

Assembler - A program for converting instructions written in low-level symbolic code into machine code.

Linker - A program used with a compiler or assembler to provide links to the libraries needed for an executable program.

## 1.4 References

***ARM7TDMI Reference Manual v.r4p1***

[***http://infocenter.arm.com/help/topic/com.arm.doc.ddi0210c/DDI0210B.pdf***](http://infocenter.arm.com/help/topic/com.arm.doc.ddi0210c/DDI0210B.pdf)

***GAS Assembler Directives***

<https://www.sourceware.org/binutils/docs-2.12/as.info/Pseudo-Ops.html>

## 1.5 Overview

For an overall description of OakSim, its capabilities and its constraints. Please refer to section 2 of this document on the following page. All relevant non-implementation related information will be collated here.

For a technical description of OakSim, including design paradigms, features, and their implementation please refer to section 3 of this document, starting on page \_\_.

# 2. The Overall Description

## 2.1 Product Perspective

OakSim is a web-based Javascript application built with the aim creating a modern implementation of a simulated ARM embedded environment. Designed for use in education, OakSim is a standalone product designed for use on any personal computer with minimal setup required. Users need only download the program to their chosen environment to be able to run and utilize it. This portability, combined with the application’s full range of features as an assembler and linker, are the main features of the tool.

### 2.1.1 System Interfaces

OakSim’s only form of system interface is through the use of any modern Javascript equipped web browser. An internet connection is only required for the initial download to the user’s system, after which the program can be run locally from the web page individual files.

### 2.1.2 Interfaces

OakSim will provide a GUI front-end within the user’s chosen web browser, providing textual feedback for both the assembly source code and the emulation context(registers, memory,etc). The end-user will require both a keyboard and a pointer device to utilize OakSim. Mobile users will also be able to utilize OakSim using both the virtual keyboard and touchscreen.

### 2.1.3 Hardware Interfaces

OakSim is capable of running within any browser that is capable of running the JAVAscript virtual machine. This environment is at a very high and abstracted level that does not interface with any specific hardware in particular besides the user’s input devices(keyboard and mouse) and *simulated hardware* such as the eventual implementation of virtual buttons and LEDs and other feedback peripherals for the emulated embedded environment.

### 2.1.4 Software Interfaces

WebBrowser(Javascript 1.8.5/EMCAScript 5)

The user will be running the software within a web browser which is capable of running within the user-space of any operating system or device, so long as the web browser minimally supports Javascript 1.8.5 which is equivalent to the EMCAScript 5 specification.

### 2.1.5 Memory Constraints

The user’s runtime environment requires access to an approximate minimum of 256 megabytes of RAM, for the the simultaneous operation of Javascript equipped web browser and the OakSim program.

## 2.2 Product Functions

OakSim is a web-based ARM development environment, capable of running on any modern web browser. Specifically, OakSim supports the following features:

* Full GAS assembly syntax implementation complete with preprocessor macros
* Simulation of ARM and Thumb instructions
* Step-by-step instruction debugging
* Assembler error feedback and program diagnostic information

## 2.3 User Characteristics

OakSim is designed for use in an educational environment by students seeking to learn how to program in the ARM embedded environment. User’s should have some knowledge of computer architecture and programming, but no knowledge of any specific embedded system is required. Some familiarity with the GAS/Intel assembler syntax is necessary.

## 2.4 Constraints

***(1)*****Availability**: OakSim is designed to be a web executable and as so is vulnerable to server failure if the user is not utilizing a locally saved copy. After download of the OakSim software the software is available for use so long as the local system supports its execution.Note: OakSim makes no use of external resources, that is, it does not utilize a back-end for computation or reporting. All executional processes are completed utilizing the local environment of the user’s machine.

***(2)******Safety and security considerations:*** OakSim acts only on, and accesses only, the browser files necessary for its execution and the assembly textual files that the user designates as input. OakSim can make no distinction between personal data and textual data, and contains no functionality related to the storage of either. OakSim does not require communication with any software outside of the browser environment it is running in, and the textual assembly files that are passed to it.

***(3)******Higher-order language requirements:*** OakSim requires the presence of JAVAscript in the local environment for the execution of its code. Presence of the ARM assembler/simulator on the local system is not required, as OakSim provides the requisite files. OakSim may utilize cross-compiled assembly code from high-order languages but does not require them.

***(4) Control functions:*** OakSim will have a full range of functions as an assembler and simulator.

It will have the standard Assemble, Run, Debug, and Terminate functions/commands.

* During assembly will be a diagnostics test. If all is well the user will receive a report stating that there are no syntactical errors, and if not the user will receive a full listing of all the errors found and where they exist within the source file.
* The Run command will then run the user’s program and enable the OakSim emulator to simulate the assembled binary within a simulated embedded context.
* The Debug feature will allow the user to run the program step by step and fix any logical errors that the diagnostics test cannot catch.
* The Terminate command will terminate the program whenever the user wishes and reset the embedded simulation context into a default state.

**(5)** **Portability:** Any and all programs that are made in OakSim will be easily moved from one computer to another. In other words the user should have the same experience running their program at a friends computer as they would on their own.

**(6) Understandability:** 95% of novice users can learn to use OakSim without outside assistance.

**(7) Usability:** OakSim must be usable to anyone that meets the minimum system requirements. There may be some interfacing limitations to users with disabilities.

**(8) Recoverability:** During a system restart, OakSim will return to a functioning state, and will contain all of the user’s code up to the most recent cache. The same must be true if OakSim itself were to crash.

## 2.5 Apportioning of Requirements.

Future desired features include:

* Translation of the User Interface into non-english native languages. (Localization)
* Support of other assembly languages.
* Environment peripheral support for the user interface.

# 3. Specific Requirements

## 3.1 External Interfaces

OakSim is designed primarily for users to input pre-written program files through a JAVAscript enabled web browser. Specific files chosen by the user from their local system are inputted to OakSim and then processed. From there, after OakSim has processed the user’s files, commands can be entered through the GUI to access OakSim’s logging functions.

## 3.2 Functions

OakSim shall support the following functions:

* Allow the user to input their desired program via an uploaded .txt file.
  + If the inputted file is not of the proper type, OakSim shall immediately return an error.
* Process the user’s code using the ARM x86 assembly language.
  + Report invalid code within the user’s entered file.
    - Highlight the incorrect code within the GUI.
    - Print a log regarding the error for the user to read.
* Display the result of the program for the user to see after successful execution.
* Allow for the user to request logs of the emulator’s activity after the successful completion of their program.
  + User can request a log of their programs total activity over the course of execution.
  + User can request a log of changes made to the emulator’s simulated registers over the course of program execution.
  + User can request a log of changes made to the emulator’s memory registers over the course of program execution.

## 3.3 Performance Requirements

OakSim involves the process of one user interfacing with their textual assembly data at a time. Textual source code data is assembled at run-time while the user is editing the assembly source text. Source data will typically not exceed character limits within the hundreds or possibly thousands. The user will typically have to wait 100 milliseconds or less for the source assembly to be compiled into its associated binary.

## 3.6 Software System Attributes

### 3.6.1 Maintainability

OakSim is designed to be a standalone program hosted on the user’s local system, and does not interact with the user’s local network. As such, there is no need to maintain the program dynamically. Program updates and bugfixes will be dispersed via the original hosting website.

### 3.6.2 Portability

OakSim is designed for use with any operating system capable of the use of any modern Javascript equipped web browser. Designed as an emulator, OakSim is not reliant on the hardware or software present on the system that it is on.

## 3.7 Organizing the Specific Requirements

### 3.7.1 System Mode

OakSim operates in one of three intuitive contexts:

* Pre-execution phase: Where the user’s textual assembly file is fed into OakSim and scanned for irregularities.
* Assembly phase: Where the user’s assembly file is parsed and assembled into its associated binary. It is here that any syntax errors are reported within the user’s source code. During compilation no user-input is utilized by the program.
* Runtime Phase: The assembled binary is simulated within a simulated embedded environment. This mode requires user input to ensure source code practicality at an instruction level. During the runtime phase the user may debug and ensure the functionality of their assembly source code. The simulator may also report any run-time errors during this phase(division by zero, access violations, etc)

### 3.7.2 User Class

OakSim does not support different levels of users and exposes no different privileges to different end-users.

### 

### 3.7.3 Objects

OakSim will maintain objects related to the different contexts of the assembly and simulation environments. The assembly will require maintaining the assembler state as well as the production of diagnostic syntax errors. The simulation environment will require context-objects to maintain the current state of the simulation and the current state of the associated debugger and other peripherals.

### 3.7.4 Stimulus

The OakSim assembler is indirectly invoked by any change made to the text editor. Should the user type, delete, paste or edit the text within this region in any way, the OakSim assembler will then immediately attempt to parse and provide diagnostic error messages to the user.

### 3. 7.5 Response

OakSim is an active software that does not function passively or in a daemon form. All OakSim functions are at the response of immediate user input.

### 3.7.6 Functional Hierarchy

OakSim will not provide a simulation or debugging context unless the source assembly code of the user is capable of being properly assembled and compiled. Parameters of the simulated embedded environment must also be valid for the simulation context to be enabled. The ability to edit the source assembly text will always be available to the user.

# Change Management Process

OakSim is an independent team-based development, changes to the SRS will be made as new features are conceived and approved by the team.

# Document Approvals

Sean Matthew Lee

Joseph Andrew Winthrope