Project Requirements

Rev 1.4

4/11/18

|  |  |  |
| --- | --- | --- |
| Date | Change | Changes made by |
| 1/30/18 | Document created | Matthew Michaels, Reagan Craddock, Milton Griffin, Michael Farden, Matthew Strenk |
| 2/8/18 | Completed Document | Matthew Michaels, Reagan Craddock, Milton Griffin, Michael Farden, Matthew Strenk |
| 2/14/18 | Made revisions to Requirements | Matthew Michaels |
| 2/27/18 | Updated requirements to reflect new design | Matthew Michaels |
| 4/11/18 | Made updates to reflect removal of mm | Matthew Michaels |
| 4/17/18 | Made some small changes | Matthew Michaels |

Table of Contents

[**Scope**](#_apyxwsfqv9z) **3**

[Purpose](#_lxllmb773rli) 3

[Document Overview](#_82dosteh0kkj) 3

[System Overview](#_c3s1dxus7hpz) 3

[**Definitions**](#_owffkim8yxx1) **3**

[**Requirements/Attributes**](#_zhjdrhu8c87r) **4**

[Constraints & Functional Requirements](#_djo4e4z22ub5) 4

[Qualitative Attributes](#_nlg3xep2iive) 4

[Quantitative Attributes](#_uy4ajewp63kk) 4

[**Verification**](#_o7xtsbumkj01) **4**

[**References**](#_o5yq9isrvikf) **5**

# Scope

## Purpose

The purpose of this document is to give a detailed description of the requirements for the Multicore Scratchpad and Cache Hybrid Simulator project. The document will give details to the entirety of the design and development of the system. This document will give a breakdown of the generic requirements, functional requirements, constraints, and both the qualitative and quantitative attributes of the project. This document will be the basis of all of the design of the project and will stand as a reference when implementing the design and delivering the project.

## Document Overview

This document is laid out in a few pieces. The first piece of the document is the scope, which encompasses this purpose, the document overview and the system overview. This section will give a general overview of this document and the project as a whole. The next piece of this document is the definitions; this will list any terms used within this document and their meanings. The next section of this document is the requirements and attributes section. This section will go very in depth with all constraints given by the customer, all system requirements based on the constraints, and any attributes of the system. The fourth section will be the verification section which will discuss how the system was/will be tested to verify that it meets all of the requirements set forth in this document. The last section of this document is the reference section that will give credit and link to any sources that were used in the creation of this document.

## System Overview

Our system will be comprised of two main pieces, the cache, and the scratchpad. The purpose for cache is that it is memory that can be accessed quicker than RAM and can store frequently reference information to increase the overall performance of the system. The one downside of cache though is that the data might not always be there, and the data request can miss and not be captured in one clock cycle; this is where the scratchpad comes in handy. Unlike cache, the scratchpad is managed in mostly software and is guaranteed to have a data access time of one clock cycle with no misses. Our system will incorporate these two different memory management systems together in a simulator and have it be a multithreaded application.

# Definitions

* 1. Multi-Core: Two or more CPUs that work together on the same chip.
  2. Scratchpad: A small, fast memory for the temporary storage of data.
  3. Cache: An auxiliary memory from which high-speed retrieval is possible.
  4. Simulation: Imitation of a situation or process.
  5. CPU: Central Processing Unit
  6. RAM: Random Access Memory

# Requirements/Attributes

## Constraints & Functional Requirements

* + 1. The system will simulate a multi-core cache/scratchpad.
    2. The system will simultaneous run a scratchpad simulation and a cache simulation each time it is run
    3. The user will chose what kind of simulation they would like to run by entering a 1, 2, 3, 4, or 5 which each correspond to a type of simulation or exiting
    4. The system will check to ensure the user enters correct data and it will prompt the user to try again if the data is incorrect
    5. The system will run the same type of simulation on both the cache and scratchpad memory
    6. The user will have four different ways to simulate the memories: hash, heap, stride, and trace
    7. The predefined trace file that can be used to simulate the memory will be formatted in the following way: R/W<address>:<size> or I<cycles>
       1. R denotes a read command, W denotes a write command, and I denotes an nop command
       2. <address> is the location of where to read from or write to
       3. <size> is how much data to read or write
       4. <cycles> is how many cycles the system will sit and wait

## Qualitative Attributes

* + 1. The utility will be run via command line by using the executable that controls the simulator
    2. The command line interface will display statistics based on the results of each simulation

## Quantitative Attributes

* + 1. The utility will simulate a two core system.
    2. Each simulated core will simulate one of the two memory types
    3. After the simulations are completed, the results will be displayed to the command line and to respective log files
    4. After the simulations are completed, any errors will be displayed to respective log files

# Verification

The detailed methods used for verification of the above requirements are discussed in the Test Plan document. However, the generalized versions of these methods will be stated here.

The testing methods for the system are based on the contents of the input file used for simulation. This input file will contain sets of data values with a corresponding command (read or write) and a destination address. This information will be pushed through the scratchpad/cache simulator and processed. The results of the simulation will verify the success rate, based on how many hits and misses of the memory there were during the simulation, of the simulator. The number of commands that are correctly executed as well as the number of failures will be recorded. This will accurately measure the success rate of the simulator. The average processing times per core will also be recorded. This will allow us to see how fast data is being processed so we can optimize the code as needed for maximum efficiency. This process will be repeated for both the cache and scratchpad portions of the system.

# References

* “Cache-Simulator”, Author: Ananth Rao, 4/9/15, <https://github.com/ananthamapod/Cache-Simulator>
* “What is the difference between scratchpad and cache memories”, Author: Shane Ryoo, 7/24/2015, <https://www.quora.com/What-is-the-difference-between-scratchpad-and-cache-memories>
* “Scratchpad memory vs Caches - Performance and Predictability comparison”, Author: David Langguth, <https://es.cs.uni-kl.de/publications/data/Lang15.pdf>