ARM Cluster

Senior Design Final Documentation

Team Name

Andrew Hoover Samantha Kranst Christine Sorensen

October 7, 2015

Contents

List of Figures

List of Tables

List of Algorithms

Overview Statements

0.1 Mission Statement

To create the fastest and most cost efficient cluster of single-board computers and investigate alternative communication modes. CNS

0.2 Elevator Pitch

Our goal is to build a cluster of single-board computers that produces as many Floating Point Operations possible per Watt per dollar. We are testing three computers, ODROID, Raspberry Pi, and PcDuino, to determine the best for the cluster. Once we have the cluster, we will investigate alternative communication modes. CNS

Document Preparation and Updates

Current Version [1.1.0]

Prepared By:
Andrew Hoover
Samantha Kranst
Christine Sorensen

Revision History

Date	***Author***	***Version***	***Comments***
09/14/15	*Christine Sorensen*	*1.0.0*	*Initial version*
10/02/15	*Christine Sorensen*	*1.1.0*	*Completion of Sprint # 1*

1

Overview and concept of operations

The overview should take the form of an executive summary. Give the reader a feel for the purpose of the document, what is contained in the document, and an idea of the purpose for the system or product.

1.1 Scope

What scope does this document cover?

1.2 Deliverables

1.3 Purpose

What is the purpose of the system or product?

1.3.1 Major System Component #1

Describe briefly the role this major component plays in this system.

1.3.2 Major System Component #2

Describe briefly the role this major component plays in this system.

1.3.3 Major System Component #3

Describe briefly the role this major component plays in this system.

1.4 Systems Goals

Briefly describe the overall goals this system plans to achieve. These goals are typically provided by the stakeholders. This is not intended to be a detailed requirements listing. Keep in mind that this section is still part of the Overview.

1.5 System Overview and Diagram

Provide a more detailed description of the major system components without getting too detailed. This section should contain a high-level block and/or flow diagram of the system highlighting the major components. See Figure 1.1. This is a floating figure environment. LaTeX will try to put it close to where it was typeset but will not allow the figure to be split if moving it can not happen. Figures, tables, algorithms and

many other floating environments are automatically numbered and placed in the appropriate type of table of contents. You can move these and the numbers will update correctly.

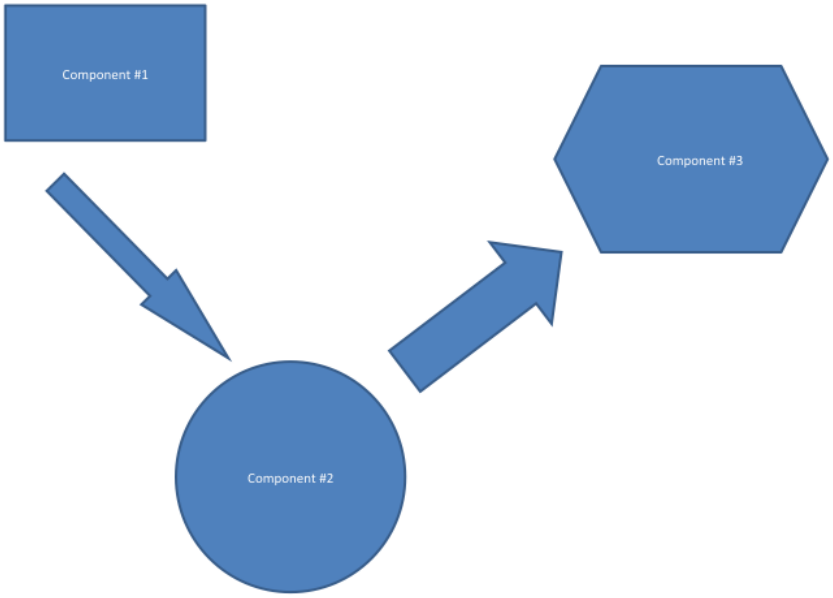

Figure 1.1: A sample figure .... System Diagram

1.6 Technologies Overview

This section should contain a list of specific technologies used to develop the system. The list should contain the name of the technology, brief description, link to reference material for further understanding, and briefly how/where/why it was used in the system. See Table 1.1. This is a floating table environment. LaTeX will try to put it close to where it was typeset but will not allow the table to be split.

Table 1.1: A sample Table ... some numbers.

7C0	hexadecimal
3700	octal
11111000000	binary
1984	decimal

2

Project Overview

This section provides some housekeeping type of information with regard to the team, project, etc.

2.1 Team Members and Roles

Describe who was involved and what role(s) were played.

2.2 Project Management Approach

This section will provide an explanation of the basic approach to managing the project. Typically, this would detail how the project will be managed through a given Agile methodology. The sprint length (i.e. 2 weeks) and product backlog ownership and location (ex. Trello) are examples of what will be discussed. An overview of the system used to track sprint tasks, bug or trouble tickets, and user stories would be warranted.

2.3 Phase Overview

If the system will be implemented in phases, describe those phases/sub-phases (design, implementation, testing, delivery) and the various milestones in this section. This section should also contain a correlation between the phases of development and the associated versioning of the system, i.e. major version, minor version, revision.

2.4 Terminology and Acronyms

Provide a list of terms used in the document that warrant definition. Consider industry or domain specific terms and acronyms as well as system specific.

3

User Stories, Backlog and Requirements

3.1 Overview

The overview should take the form of an executive summary. Give the reader a feel for the purpose of the document, what is contained in the document, and an idea of the purpose for the system or product.

The userstories are provided by the stakeholders. You will create he backlogs and the requirements, and document here. This chapter should contain details about each of the requirements and how the requirements are or will be satisfied in the design and implementation of the system.

Below: list, describe, and define the requirements in this chapter. There could be any number of subsections to help provide the necessary level of detail.

3.1.1 Scope

What scope does this document cover? This document would contain stakeholder information, initial user stories, requirements, proof of concept results, and various research task results.

3.1.2 Purpose of the System

What is the purpose of the system or product?

3.2 Stakeholder Information

This section would provide the basic description of all of the stakeholders for the project. Who has an interest in the successful and/or unsuccessful completion of this project?

3.2.1 Customer or End User (Product Owner)

Who? What role will they play in the project? Will this person or group manage and prioritize the product backlog? Who will they interact with on the team to drive product backlog priorities if not done directly?

3.2.2 Management or Instructor (Scrum Master)

Who? What role will they play in the project? Will the Scrum Master drive the Sprint Meetings?

3.2.3 Investors

Are there any? Who? What role will they play?

3.2.4 Developers –Testers

Who? Is there a defined project manager, developer, tester, designer, architect, etc.?

3.3 Business Need

Use this section to define what business need exist and how this software will meet and/or exceed that business need.

3.4 Requirements and Design Constraints

Use this section to discuss what requirements exist that deal with meeting the business need. These requirements might equate to design constraints which can take the form of system, network, and/or user constraints. Examples: Windows Server only, iOS only, slow network constraints, or no offline, local storage capabilities.

3.4.1 System Requirements

What are they? How will they impact the potential design? Are there alternatives?

3.4.2 Network Requirements

What are they?

3.4.3 Development Environment Requirements

What are they? Is the system supposed to be cross-platform?

3.4.4 Project Management Methodology

The stakeholders might restrict how the project implementation will be managed. There may be constraints on when design meetings will take place. There might be restrictions on how often progress reports need to be provided and to whom.

- What system will be used to keep track of the backlogs and sprint status?
- Will all parties have access to the Sprint and Product Backlogs?
- How many Sprints will encompass this particular project?
- How long are the Sprint Cycles?
- Are there restrictions on source control?

3.5 User Stories

This section can really be seen as the guts of the document. This section should be the result of discussions with the stakeholders with regard to the actual functional requirements of the software. It is the user stories that will be used in the work breakdown structure to build tasks to fill the product backlog for implementation through the sprints.

This section should contain sub-sections to define and potentially provide a breakdown of larger user stories into smaller user stories.

3.5.1 User Story #1

As a user, I want a cluster of at least 6 and no more than 12 single-board computers.

3.5.1.a User Story #1 Breakdown

The cluster will be made of ODROIDs, PcDuinos, or Raspberry Pi's, depending on which performs best in the benchmark tests.

3.5.2 User Story #2

As a user, I want the fasest, most efficient in both cost and operation cluster.

3.5.2.a User Story #2 Breakdown

Testing will be done on the single-board computers compared with prices to determine which will be best for the ARM cluster.

3.5.3 User Story #3

I want to the cluster to be at or below the maximum budget of $1,200.00.

3.5.3.a User Story #3 Breakdown

The budget must include all components of the cluster: the computer boards, cost of power, switch, memory, cables, and power strips.

3.5.4 User Story #4

I want to know which of the single-board computers is the fastest in GFlops/$/Watt.

3.5.4.a User Story #4 Breakdown

Testing will take place on the ODROID, PcDuino, and Raspberry Pi to determine which is the fastest in this metric.

3.5.5 User Story #5

I want a different communication mode beyond standard Ethernet.

3.5.5.a User Story #5 Breakdown

Utilize the other pins and ports to find an alternative form of communication.

3.5.6 User Story #6

Develop a message passing protocol for the communication.

3.5.6.a User Story #6 Breakdown

There is no message passing protocol for the other modes of communication. They must be developed and benchmarked.

3.6 Research or Proof of Concept Results

This section is reserved for the discussion centered on any research that needed to take place before full system design. The research efforts may have led to the need to actually provide a proof of concept for approval by the stakeholders. The proof of concept might even go to the extent of a user interface design or mockups.

3.7 Supporting Material

This document might contain references or supporting material which should be documented and discussed either here if appropriate or more often in the appendices at the end. This material may have been provided by the stakeholders or it may be material garnered from research tasks.

4

Design and Implementation

This section is used to describe the design details for each of the major components in the system. Note that this chapter is critical for all tracks. Research tracks would do experimental design here where other tracks would include the engineering design aspects. This section is not brief and requires the necessary detail that can be used by the reader to truly understand the architecture and implementation details without having to dig into the code. Sample algorithm: Algorithm 1. This algorithm environment is automatically placed - meaning it floats. You don't have to worry about placement or numbering.

Algorithm 1 Calculate $y = x^n$

Require: $n \geq 0 \vee x \neq 0$
Ensure: $y = x^n$
  $y \Leftarrow 1$
  if $n < 0$ **then**
    $X \Leftarrow 1/x$
    $N \Leftarrow -n$
  else
    $X \Leftarrow x$
    $N \Leftarrow n$
  end if
  while $N \neq 0$ **do**
    if N is even **then**
      $X \Leftarrow X \times X$
      $N \Leftarrow N/2$
    else {N is odd}
      $y \Leftarrow y \times X$
      $N \Leftarrow N - 1$
    end if
  end while

Citations look like [2, 1, 3] and [6, 4, 5]. These are done automatically. Just fill in the database `designrefs.bib` using the same field structure as the other entries. Then pdflatex the document, bibtex the document and pdflatex twice again. The first pdflatex creates requests for bibliography entries. The bibtex extracts and formats the requested entries. The next pdflatex puts them in order and assigns labels. The final pdflatex replaces references in the text with the assigned labels. The bibliography is automatically constructed.

4.1 Major Component #1

4.1.1 Technologies Used

This section provides a list of technologies used for this component. The details for the technologies have already been provided in the Overview section.

4.1.2 Component Overview

This section can take the form of a list of features.

4.1.3 Phase Overview

This is an extension of the Phase Overview above, but specific to this component. It is meant to be basically a brief list with space for marking the phase status.

4.1.4 Architecture Diagram

It is important to build and maintain an architecture diagram. However, it may be that a component is best described visually with a data flow diagram.

4.1.5 Data Flow Diagram

It is important to build and maintain a data flow diagram. However, it may be that a component is best described visually with an architecture diagram.

4.1.6 Design Details

This is where the details are presented and may contain subsections. Here is an example code listing:

```
#include <stdio.h>
#define N 10
/* Block
 * comment */

int main()
{
    int i;

    // Line comment.
    puts("Hello world!");

    for (i = 0; i < N; i++)
    {
        puts("LaTeX is also great for programmers!");
    }

    return 0;
}
```

This code listing is not floating or automatically numbered. If you want auto-numbering, but it in the algorithm environment (not algorithmic however) shown above.

4.2 Major Component #2

4.2.1 Technologies Used

This section provides a list of technologies used for this component. The details for the technologies have already been provided in the Overview section.

4.2.2 Component Overview

This section can take the form of a list of features.

4.2.3 Phase Overview

This is an extension of the Phase Overview above, but specific to this component. It is meant to be basically a brief list with space for marking the phase status.

4.2.4 Architecture Diagram

It is important to build and maintain an architecture diagram. However, it may be that a component is best described visually with a data flow diagram.

4.2.5 Data Flow Diagram

It is important to build and maintain a data flow diagram. However, it may be that a component is best described visually with an architecture diagram.

4.2.6 Design Details

This is where the details are presented and may contain subsections.

4.3 Major Component #3

4.3.1 Technologies Used

This section provides a list of technologies used for this component. The details for the technologies have already been provided in the Overview section.

4.3.2 Component Overview

This section can take the form of a list of features.

4.3.3 Phase Overview

This is an extension of the Phase Overview above, but specific to this component. It is meant to be basically a brief list with space for marking the phase status.

4.3.4 Architecture Diagram

It is important to build and maintain an architecture diagram. However, it may be that a component is best described visually with a data flow diagram.

4.3.5 Data Flow Diagram

It is important to build and maintain a data flow diagram. However, it may be that a component is best described visually with an architecture diagram.

4.3.6 Design Details

This is where the details are presented and may contain subsections.

5

System and Unit Testing

This section describes the approach taken with regard to system and unit testing.

5.1 Overview

Provides a brief overview of the testing approach, testing frameworks, and general how testing is/will be done to provide a measure of success for the system.

5.2 Dependencies

Describe the basic dependencies which should include unit testing frameworks and reference material.

5.3 Test Setup and Execution

Describe how test cases were developed, setup, and executed. This section can be extremely involved if a complete list of test cases was warranted for the system.

6

Development Environment

The basic purpose for this section is to give a developer all of the necessary information to setup their development environment to run, test, and/or develop.

6.1 Development IDE and Tools

Describe which IDE and provide links to installs and/or reference material.

6.2 Source Control

Which source control system is/was used? How was it setup? How does a developer connect to it?

6.3 Dependencies

Describe all dependencies associated with developing the system.

6.4 Build Environment

How are the packages built? Are there build scripts?

6.5 Development Machine Setup

If warranted, provide a list of steps and details associated with setting up a machine for use by a developer.

7

Release – Setup – Deployment

This section should contain any specific subsection regarding specifics in releasing, setup, and/or deployment of the system.

7.1 Deployment Information and Dependencies

Are there dependencies that are not embedded into the system install?

7.2 Setup Information

How is a setup/install built?

7.3 System Versioning Information

How is the system versioned?

8

User Documentation

This section should contain the basis for any end user documentation for the system. End user documentation would cover the basic steps for setup and use of the system. It is likely that the majority of this section would be present in its own document to be delivered to the end user. However, it is recommended the original is contained and maintained in this document.

8.1 User Guide

The source for the user guide can go here. You have some options for how to handle the user docs. If you have some `newpage` commands around the guide then you can just print out those pages. If a different formatting is required, then have the source in a separate file `userguide.tex` and include that file here. That file can also be included into a driver (like the senior design template) which has the client specified formatting. Again, this is a single source approach.

8.2 Installation Guide

8.3 Programmer Manual

9

Class Index

9.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

10

Class Documentation

10.1 Poly Class Reference

Public Member Functions

- Poly ()
- ~Poly ()
- int myfunction (int)

10.1.1 Constructor & Destructor Documentation

10.1.1.a Poly::Poly ( )

My constructor

10.1.1.b Poly::~Poly ( )

My destructor

10.1.2 Member Function Documentation

10.1.2.a int Poly::myfunction (int *a*)

my own example function fancy new function
new variable
The documentation for this class was generated from the following file:

- hello.cpp

11

Experimental Log

A log of all research activities.

11.1 Benchmarking the Individual Computers

9/17/15 PcDuino isn't working according to Dr. Karlsson. The PcDuinos are about $160 each, so chances were that the PcDuino wasn't going to be selected for the cluster. We will not put the PcDuino in consideration with our benchmarking.

9/17/15 PcDuino isn't working according to Dr. Karlsson. The PcDuinos are about $160 each, so chances were that the PcDuino wasn't going to be selected for the cluster. We will not put the PcDuino in consideration with our benchmarking.

9/22/15 We begin work on benchmarking the remaining candidates. The code we will be using to benchmark the two devices will test the addition, multiplication, division, trigonmetric function in single and double point precision of two massively large arrays filled with random numbers.

9/29/15 OpenMP is added to the benchmark code so the program runs on all cores. Results are as follows:

Length of Time (seconds)				
Device	Addition	Multiplication	Division	Sine
ODroid 4xU	29.925	31.341	37.032	227.40
Raspberry Pi 2B	221.645	221.034	297.204	1468.63

9/30/15 The gigaflops are calculated.

Gigaflops				
Device	Addition	Multiplication	Division	Sine
ODroid 4xU	0.311	0.297	0.251	0.0410
Raspberry Pi 2B	0.0420	0.0421	0.0313	0.00634

10/1/15 The wattage is measured when the devices are running these operations. Using the wattage, the metric of GFlops/Dollar/Watt is calculated.

Gigaflops per Dollar per Watts				
Device	Addition	Multiplication	Division	Sine
ODroid 4xU	0.00028	0.000268	0.000226	0.0000369
Raspberry Pi 2B	0.0003	0.0003	0.000224	0.0000453

10/1/15 The results show that the Raspberry Pi and the ODroid perform nearly the same. The Raspberry Pi in our benchmarking proved the best. However, it is inconclusive as to which computer will be used.

12

Research Results

This chapter describes the results and conclusions of your research. This would be the final report for a research project.

12.1 Result 1

12.2 Result 2

12.3 Conclusions

12.4 Further work

Bibliography

[1] R. Arkin. *Governing Lethal Behavior in Autonomous Robots*. Taylor & Francis, 2009.

[2] Howie Choset, Kevin M. Lynch, Seth Hutchinson, George A Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, Cambridge, MA, June 2005.

[3] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at http://planning.cs.uiuc.edu/.

[4] V. Lumelsky and A. Stepanov. Path planning strategies for point mobile automation moving amidst unknown obstacles of arbirary shape. *Algorithmica*, pages 403–430, 1987.

[5] S.A. NOLFI and D.A. FLOREANO. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. A Bradford book. A BRADFORD BOOK/THE MIT PRESS, 2000.

[6] Wikipedia. Asimo — Wikipedia, the free encyclopedia. `http://upload.wikimedia.org/wikipedia/commons/thumb/0/05/HONDA_ASIMO.jpg/450px-HONDA_ASIMO.jpg`, 2013. [Online; accessed June 23, 2013].

SDSMT SENIOR DESIGN
SOFTWARE DEVELOPMENT AGREEMENT

This Software Development Agreement (the "Agreement") is made between the SDSMT Computer Science Senior Design Team ARM Cluster, consisting of team members Andrew Hoover, Samantha Kranz & Christine Sorensen, and Sponsor Dr. Christer Karlsson, with address: 501 E. St Joseph St., Rapid City, SD, 57701.

1 RECITALS

1. Sponsor desires Senior Design Team to develop software for the research project of developing an ARM Cluster.
2. Senior Design Teams willing to develop such Software.

NOW, THEREFORE, in consideration of the mutual covenants and promises herein contained, the Team and Sponsor agree as follows:

2 EFFECTIVE DATE

This Agreement shall be effective as of 10/2/15

3 DEFINITIONS

1. "Software" shall mean the computer programs in machine readable object code form and any subsequent error corrections or updates supplied to Sponsor by Senior Design Team pursuant to this Agreement.
2. "Acceptance Criteria" means the written technical and operational performance and functional criteria and documentation standards set out in the design document.
3. "Acceptance Date" means 2015, April 18 (before the Design Fair) for each Milestone when all Deliverables included in that Milestone have been accepted by Dr. C. Karlsson in accordance with the Acceptance Criteria and this Agreement.
4. "Deliverable" means a deliverable specified in the design document deliverable section.
5. "Delivery Date" shall mean the date on which University has delivered to Dr. C. Karlsson all of the Deliverables in accordance with the design document and this Agreement.
6. "Documentation" means the documents, manuals and written materials (including end-user manuals) referenced, indicated or described in the design document or otherwise developed pursuant to this Agreement.
7. "Milestone" means the completion and delivery of all of the Deliverables or other events which are included or described in the design document scheduled for delivery and/or completion on a given target date; a Milestone will not be considered completed until the Acceptance Date has occurred with respect to all of the Deliverables for that Milestone.

4 DEVELOPMENT OF SOFTWARE

1. Senior Design Team will use its best efforts to develop the Software described in design document deliverable section. The Software development will be under the direction of or his/her successors as mutually agreed to by the parties "Team Lead" and will be conducted by the Team Lead. The Team will deliver the Software to the satisfaction of the course instructor that reasonable effort has been made to address the needs of the client. The Team understands that failure to deliver the Software is grounds for failing the course.
2. Sponsor understands that the Senior Design course's mission is education and advancement of knowledge, and, consequently, the development of Software must further that mission. The Senior Design Course does not guarantee specific results or any results, and the Software will be developed only on a best efforts basis. The Software is considered PROOF OF CONCEPT only and is NOT intended for commercial, medical, mission critical or industrial applications.
3. The Senior Design instructor will act as mediator between Sponsor and Team; and resolve any conflicts that may arise.

5 CONSULTATION AND REPORTS

1. Sponsor's designated representative for consultation and communications with the Team Lead shall be Christine Sorensen or such other person as Sponsor may from time to time designate to the Team Lead (Christine Sorensen).
2. During the Term of the Agreement, Sponsor's representatives may consult informally with course instructor regarding the project, both personally and by telephone. Access to work carried on in University facilities, if any, in the course of this Agreement shall be entirely under the control of University personnel but shall be made available on a reasonable basis.
3. The Team Lead will submit written progress reports. At the conclusion of this Agreement, the Team Lead shall submit a comprehensive final report in the form of the formal course documentation at the conclusion of the Senior Design II course.

6 CONFIDENTIAL INFORMATION

1. The parties may wish, from time to time, in connection with work contemplated under this Agreement, to disclose confidential information to each other ("Confidential Information"). Each party will use reasonable efforts to prevent the disclosure of any of the other party's Confidential Information to third parties for a period of three (3) years after the termination of this Agreement, provided that the recipient party's obligation shall not apply to information that:
   (a) is not disclosed in writing or reduced to writing and so marked with an appropriate confidentiality legend within thirty (30) days of disclosure;
   (b) is already in the recipient party's possession at the time of disclosure thereof;
   (c) is or later becomes part of the public domain through no fault of the recipient party;
   (d) is received from a third party having no obligations of confidentiality to the disclosing party;
   (e) is independently developed by the recipient party; or
   (f) is required by law or regulation to be disclosed.
2. In the event that information is required to be disclosed pursuant to subsection (6), the party required to make disclosure shall notify the other to allow that party to assert whatever exclusions or exemptions may be available to it under such law or regulation.

7 INTELLECTUAL PROPERTY RIGHTS

All deliverable become property of Dr. C. Karlsson and South Dakota School of Mines and Tecnology.

8 WARRANTIES

The Senior Design Team represents and warrants to Sponsor that:

1. the Software is the original work of the Senior Design Team in each and all aspects;
2. the Software and its use do not infringe any copyright or trade secret rights of any third party.

No agreements will be made beyond items (1) and (2).

9 INDEMNITY

1. Sponsor is responsible for claims and damages, losses or expenses held against the Sponsor.
2. Sponsor shall indemnify and hold harmless the Senior Design Team, its affiliated companies and the officers, agents, directors and employees of the same from any and all claims and damages, losses or expenses, including attorney's fees, caused by any negligent act of Sponsor or any of Sponsor's agents, employees, subcontractors, or suppliers.
3. NEITHER PARTY TO THIS AGREEMENT NOR THEIR AFFILIATED COMPANIES, NOR THE OFFICERS, AGENTS, STUDENTS AND EMPLOYEES OF ANY OF THE FOREGOING, SHALL BE LIABLE TO ANY OTHER PARTY HERETO IN ANY ACTION OR CLAIM FOR CONSEQUENTIAL OR SPECIAL DAMAGES, LOSS OF PROFITS, LOSS OF OPPORTUNITY, LOSS OF PRODUCT OR LOSS OF USE, WHETHER THE ACTION IN WHICH RECOVERY OF DAMAGES IS SOUGHT IS BASED ON CONTRACT TORT (INCLUDING SOLE, CONCURRENT OR OTHER NEGLIGENCE AND STRICT LIABILITY), STATUTE OR OTHERWISE. TO THE EXTENT PERMITTED BY LAW, ANY STATUTORY REMEDIES WHICH ARE INCONSISTENT WITH THE PROVISIONS OF THESE TERMS ARE WAIVED.

10 INDEPENDENT CONTRACTOR

For the purposes of this Agreement and all services to be provided hereunder, the parties shall be, and shall be deemed to be, independent contractors and not agents or employees of the other party. Neither party shall have authority to make any statements, representations or commitments of any kind, or to take any action which shall be binding on the other party, except as may be expressly provided for herein or authorized in writing.

11 TERM AND TERMINATION

1. This Agreement shall commence on the Effective Date and extend until the end of classes of the second semester of Senior Design (CSC 467), unless sooner terminated in accordance with the provisions of this Section (Fall 2015-Spring 2016).
2. This Agreement may be terminated by the written agreement of both parties.

3. In the event that either party shall be in default of its materials obligations under this Agreement and shall fail to remedy such default within thirty (30) days after receipt of written notice thereof, this Agreement shall terminate upon expiration of the thirty (30) day period.
4. Any provisions of this Agreement which by their nature extend beyond termination shall survive such termination.

12 GENERAL

1. This Agreement constitutes the entire and only agreement between the parties relating to the Senior Design Course, and all prior negotiations, representations, agreements and understandings are superseded hereby. No agreements altering or supplementing the terms hereof may be made except by means of a written document signed by the duly authorized representatives of the parties.
2. This Agreement shall be governed by, construed, and enforced in accordance with the internal laws of the State of South Dakota.

14 SIGNATURES

Andrew Hoover — 10-2-2015

Samantha Kranstz — 10-2-15

Christine Sorensen — 10/2/15

Dr. Christer Karlsson — 10/2/15

A

Product Description

Write a description of the product to be developed. Use sectioning commands as neccessary.

NOTE: *This is part of the contract.*

B

Publications

Research Track: This chapter will include any publications generated from the research. Most likely these will be preprints and one will just include the pdf.

C

Sprint Reports

1 Sprint Report #1

Deliverables

- Mission Statement
- User Stories
- Number Generating Code
- Benchmark Code
- Benchmark Log
- Signed Software Contract
- Updated Design Document

Work for this sprint included:

- Wrote Mission Statement and Elevator Speech
- Drew up Software Contract
- Wrote user stories
- Obtained ODroid 4xU, Raspberry Pi 2B, and PcDuino 8 single-board computers
- Wrote number generating code
- Wrote benchmark code that ran addition, multiplication, division, and sine floating point operations
- Added OpenMP to run the benchmark code on all cores
- Ran the code each of the single-board computers
- Logged times
- Calculated the GFlops
- Calculated the GFlops/Dollar/Watts
- Determined best computer

Work that is carried over into Sprint 2 is as follows:

- Using the benchmark results to determine which computer to use
- Order more of the computers that proved best from Sprint 1 and maintain the given budget of $1,200
- Find a topology that best fits the cluster

Backlog

- Decide on a computer based on the results of the benchmarking
- Calculate prices on supplies and computers while maintaining below the budget
- Ordering said supplies and computers
- Build the cluster to perform floating-point operations
- Benchmark the cluster
- Experiment with different connections
- Create a new mode of communication

2 Sprint Report #2

The sprint reports should be inserted here. Reports focus on process. Design elements can be inserted into the design chapter with the report discussing the design element in more of an overview fashion.

3 Sprint Report #3

4 Sprint Report ...

D

Industrial Experience and Resumes

1 Resumes

Your resumes are included here. See the source file (industrial.tex) and uncomment the PDF includes to see how this works. If your resume is written in LaTeX then you can just insert the LaTeX source code.

2 ABET: Industrial Experience Reports

2.1 Name1

2.2 Name2

2.3 Name3

E

Acknowledgment

Thanks

F

Supporting Materials

This document will contain several appendices used as a way to separate out major component details, logic details, or tables of information. Use of this structure will help keep the document clean, readable, and organized.

LaTeX Example

LaTeX sample file: Remove from submitted materials

1 Introduction

This is a sample input file. Comparing it with the output it generates can show you how to produce a simple document of your own.

2 Ordinary Text

The ends of words and sentences are marked by spaces. It doesn't matter how many spaces you type; one is as good as 100. The end of a line counts as a space.

One or more blank lines denote the end of a paragraph.

Since any number of consecutive spaces are treated like a single one, the formatting of the input file makes no difference to TeX, but it makes a difference to you. When you use LaTeX, making your input file as easy to read as possible will be a great help as you write your document and when you change it. This sample file shows how you can add comments to your own input file.

Because printing is different from typewriting, there are a number of things that you have to do differently when preparing an input file than if you were just typing the document directly. Quotation marks like "this" have to be handled specially, as do quotes within quotes: "'this' is what I just wrote, not 'that'".

Dashes come in three sizes: an intra-word dash, a medium dash for number ranges like 1–2, and a punctuation dash—like this.

A sentence-ending space should be larger than the space between words within a sentence. You sometimes have to type special commands in conjunction with punctuation characters to get this right, as in the following sentence. Gnats, gnus, etc. all begin with G. You should check the spaces after periods when reading your output to make sure you haven't forgotten any special cases. Generating an ellipsis ... with the right spacing around the periods requires a special command.

TeX interprets some common characters as commands, so you must type special commands to generate them. These characters include the following: $ & % # { and }.

In printing, text is emphasized by using an *italic* type style.

A long segment of text can also be emphasized in this way. Text within such a segment given additional emphasis with Roman *type. Italic type loses its ability to emphasize and become simply distracting when used excessively.*

It is sometimes necessary to prevent TeX from breaking a line where it might otherwise do so. This may be at a space, as between the "Mr." and "Jones" in "Mr. Jones", or within a word—especially when the word is a symbol like *itemnum* that makes little sense when hyphenated across lines.

Footnotes[1] pose no problem.

TeX is good at typesetting mathematical formulas like $x - 3y = 7$ or $a_1 > x^{2n}/y^{2n} > x'$. Remember that a letter like x is a formula when it denotes a mathematical symbol, and should be treated as one.

[1] This is an example of a footnote.

3 Displayed Text

Text is displayed by indenting it from the left margin. Quotations are commonly displayed. There are short quotations

> This is a short a quotation. It consists of a single paragraph of text. There is no paragraph indentation.

and longer ones.

> This is a longer quotation. It consists of two paragraphs of text. The beginning of each paragraph is indicated by an extra indentation.
>
> This is the second paragraph of the quotation. It is just as dull as the first paragraph.

Another frequently-displayed structure is a list. The following is an example of an *itemized* list.

- This is the first item of an itemized list. Each item in the list is marked with a "tick". The document style determines what kind of tick mark is used.
- This is the second item of the list. It contains another list nested inside it. The inner list is an *enumerated* list.
  1. This is the first item of an enumerated list that is nested within the itemized list.
  2. This is the second item of the inner list. LaTeX allows you to nest lists deeper than you really should.

  This is the rest of the second item of the outer list. It is no more interesting than any other part of the item.
- This is the third item of the list.

You can even display poetry.

> There is an environment for verse
> Whose features some poets will curse.
>
> For instead of making
> Them do *all* line breaking,
> It allows them to put too many words on a line when they'd rather be forced to be terse.

Mathematical formulas may also be displayed. A displayed formula is one-line long; multi-line formulas require special formatting instructions.

$$x' + y^2 = z_i^2$$

Don't start a paragraph with a displayed equation, nor make one a paragraph by itself.

4 Build process

To build LaTeX documents you need the latex program. It is free and available on all operating systems. Download and install. Many of us use the TexLive distribution and are very happy with it. You can use a editor and command line or use an IDE. To build this document via command line:

```
alta>  pdflatex SystemTemplate
```

If you change the bib entries, then you need to update the bib files:

```
alta>  pdflatex SystemTemplate
alta>  bibtex SystemTemplate
alta>  pdflatex SystemTemplate
alta>  pdflatex SystemTemplate
```

The template files provided also contain a Makefile, which will make things much easier.

Acknowledgment

Thanks to Leslie Lamport.