GENERIC ON-BOARD-COMPUTER HARDWARE AND SOFTWARE DEVELOPMENT FOR NANOSATELLITE APPLICATIONS

KONSTANTIN BORSCHIOV

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER'S OF SCIENCE

GRADUATE PROGRAM IN EARTH AND SPACE SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

NOVEMBER 2012

GENERIC ON-BOARD-COMPUTER HARDWARE AND SOFTWARE DEVELOPMENT FOR NANOSATELLITE APPLICATIONS

by **Konstantin Borschiov**

A thesis submitted to the Faculty of Graduate Studies of York University in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

GENERIC ON-BOARD-COMPUTER HARDWARE AND SOFTWARE DEVELOPMENT FOR NANOSATELLITE APPLICATIONS

by **Konstantin Borschiov**

By virtue of submitting this document electronically, the author certifies that this is a true electronic equivalent of the copy of the thesis approved by York University for the award of the degree. No alteration of the content has occurred and if there are any minor variations in formatting, they are as a result of the conversion to Adobe Acrobat format (or similar software application).

Examination Committee members:

1. Dr. Regina Lee

2. Dr. Michael Daly

3. Dr. Brendan Quine

4. Dr. George Vukovich

Abstract

This study outlines the results obtained from the development of a generic nanosatellite on-board-computer (OBC). The nanosatellite OBC is a non-mission specific design and as such it must be adaptable to changing mission requirements in order to be suitable for varying nanosatellite missions. Focus is placed on the commercial-off-the-shelf (COTS) principle where commercial components are used and evaluated for their potential performance in nanosatellite applications. The OBC design is prototyped and subjected to tests to evaluate its performance and its feasibility to survive in space.

Acknowledgements

I would like to thank all those who have supported me and encouraged my work through my years at York University. In particular I would like to thank my supervisor, Professor Regina Lee, for her continued support and motivation as I have progressed with my research.

I would like to thank the entire YuSEND team at York University, both past and present members, for their friendship, continued motivation and willingness to assist me whenever I have come across problems. In particular, Mark Post, whose knowledge of electronics and software has proved most invaluable as my research has progressed. I would also like to acknowledge Hugh Chesser for keeping the group on track and making us always think of the big picture. I would like to thank Anthony Dineen for providing with very needed distractions to the otherwise stressful days.

I would also like to express my gratitude to Professor Brendan Quine for his assistance and support during the TVAC testing stage.

I would like to thank COM DEV and the Natural Science and Engineering Council for their financial support.

I would like to express my gratitude and thanks to Beth Duffus for keeping me in line and on track as I was working towards the completion of my work and the writing of my thesis. Lastly, I want to thank my parents whose patience and support during my entire life has brought me to where I am today.

Table of Contents

List of Figures

List of Tables

List of Acronyms

AC	Alternating Current
ADC	Analog-to-Digital Converter
API	Application Programming Interface
ARM	Advanced RISC Machine
ASIC	Application Specific Integrated Circuits
BGA	Ball Grid Array
CANOE	Canadian Advance Nanosatellite Operating Environment
COTS	Commercial Off The Shelf
DDR	Double Data Rate
DRAM	Dynamic Random Access Memory
EDAC	Error Detection and Correction
EEPROM	Electrically Erasable Programmable Read Only Memory
ENIAC	Electronic Numerical Integrator And Computer
FPGA	Field Programmable Gate Array
FRAM	Ferroelectric Random Access Memory
GNB	Generic Nanosatellite Bus
GPIO	General Purpose Input/Output
HDL	Hardware Description Language
JFFS2	Journaling Flash File System version 2
JTAG	Joint Test Action Group

KiB	Kibibyte (1024 bytes)
LEO	Low Earth Orbit
MiB	MebiByte (1024 KiB)
MRAM	Magnetoresistive Random Access Memory
NASA	National Aeronautics and Space Administration
OBC	On-Board Computer
PCB	Printed Circuit Board
P-POD	Poly-Picosatellite Orbital Deployer
RISC	Reduced Instruction Set Computing
ROM	Read Only Memory
SDR	Single Data Rate
SDRAM	Synchronous Dynamic Random Access Memory
SEE	Single Event Effects
SoC	System on a Chip
SRAM	Static Random Access Memory
TID	Total Ionizing Dose
TMR	Triple Modular Redudnacy
TVAC	Thermal Vacuum
TWI	Two Wire Interface
UART	Universal Asynchronous Receiver/Transmitter
UBI	Unsorted Block Image
UBIFS	Unsorted Block Image FileSystem

UTIAS/SFL	University of Toronto Institue for Aerospace Studies - Space Flight Laboratory
YAFFS2	Yet Another Flash File System version 2
YuSEND	York University Space Engineering Nanosatellite Demonstration

Chapter 1 - Introduction

The purpose of the research is to develop a nanosatellite on-board-computer (OBC) for generic nanosatellite missions. The OBC design takes into account limitations of nanosatellites and the environment they operate in.

A nanosatellite is a sub-classification of satellite types by mass. Several range definitions for nanosatellites exist but the one that is being used in this thesis is the range of 1kg to 10kg. Definitions of the mass range can vary from different countries to different institutions. In Canada the accepted definition is the one mentioned above [1] while the CubeSat standard defines that range as a picosatellite [2].

Nanosatellites are becoming increasingly popular due to being small enough that they are launched for less cost than a traditional large satellite. In order to accomplish useful tasks, nanosatellites tend to focus on using commercial-off-the-shelf (COTS) products in order to drive cost down further [1]. Advances in electronics also mean that commercial electronics are more powerful and smaller in comparison to older technology even as recent as 10 years ago and they allow nanosatellites to be capable of accomplishing tasks that in the past required a larger satellite.

Since the nanosatellite subcategory only defines the mass of the satellite, it is often left to the designers to choose other parameters. A class of nanosatellites known as CubeSat refers to a specific set of physical parameters of nanosatellites including shape and tolerances. CubeSat specifications define three categories of satellite where the smallest one is a cube with dimensions of 10x10x13.5 cm^3 and a mass limit of 1.33kg [2], known

as a 1U CubeSat, where the U stands for unit. Figure 1 below depicts a 1U CubeSat under development at York University.

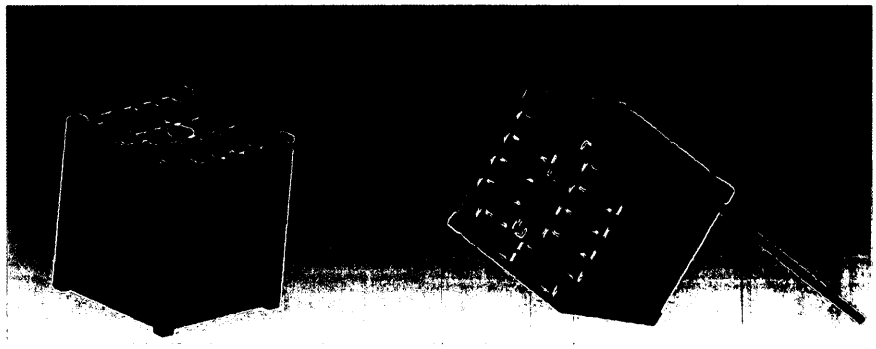

Figure 1 - 1U CubeSat Model Developed at York University

The specifications define the largest CubeSat as being a 3U CubeSat with dimensions 10x10x34.5 cm^3. The specifications are an attempt at creating a standardized form factor such that multiple nanosatellites can be loaded onto a single launcher known as a Poly Picosatellite Orbital Deployer (P-POD) [3]. Note that the CubeSat specifications use the picosatellite designation for the CubeSat as opposed to the nanosatellite designation as mentioned above. The P-POD is a container into which CubeSats are loaded while it in turn is mounted onto a launch vehicle. The P-POD can accommodate three 1U CubeSats or a single 3U CubeSat inside of it. The use of a P-POD and hence the CubeSat standard, eliminates the need for satellite designers to be concerned about the mounting of the satellite in the launch vehicle. Instead, this is a concern for the designers of the P-POD, of which commercial solutions are available which outline the requirements for how the nanosatellites fit in the commercial P-POD.

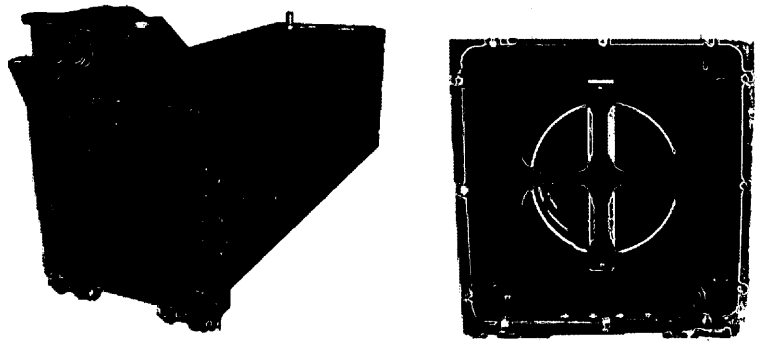

Figure 2 – Poly-Picosatellite Orbital Deployer (P-POD) Model and Cross Section
Image Credit: Cal Poly [2]

Adherence to CubeSat specifications will help the final product to be compatible with other systems on the market, e.g. a team can design a 1U CubeSat and use a commercial P-POD while possibly sharing launch costs with other nanosatellites designers that want their satellite to be in the same P-POD.

A computer system is central to traditional nanosatellite designs; it provides the capability to perform processing of data as well as control of autonomous tasks. The OBC is connected to the subsystems of a nanosatellite and provides a centralized on-board control center as is shown in Figure 3. A connection between the receiver and the power system exists so that the power system may be able to power cycle the OBC in case of an emergency on command.

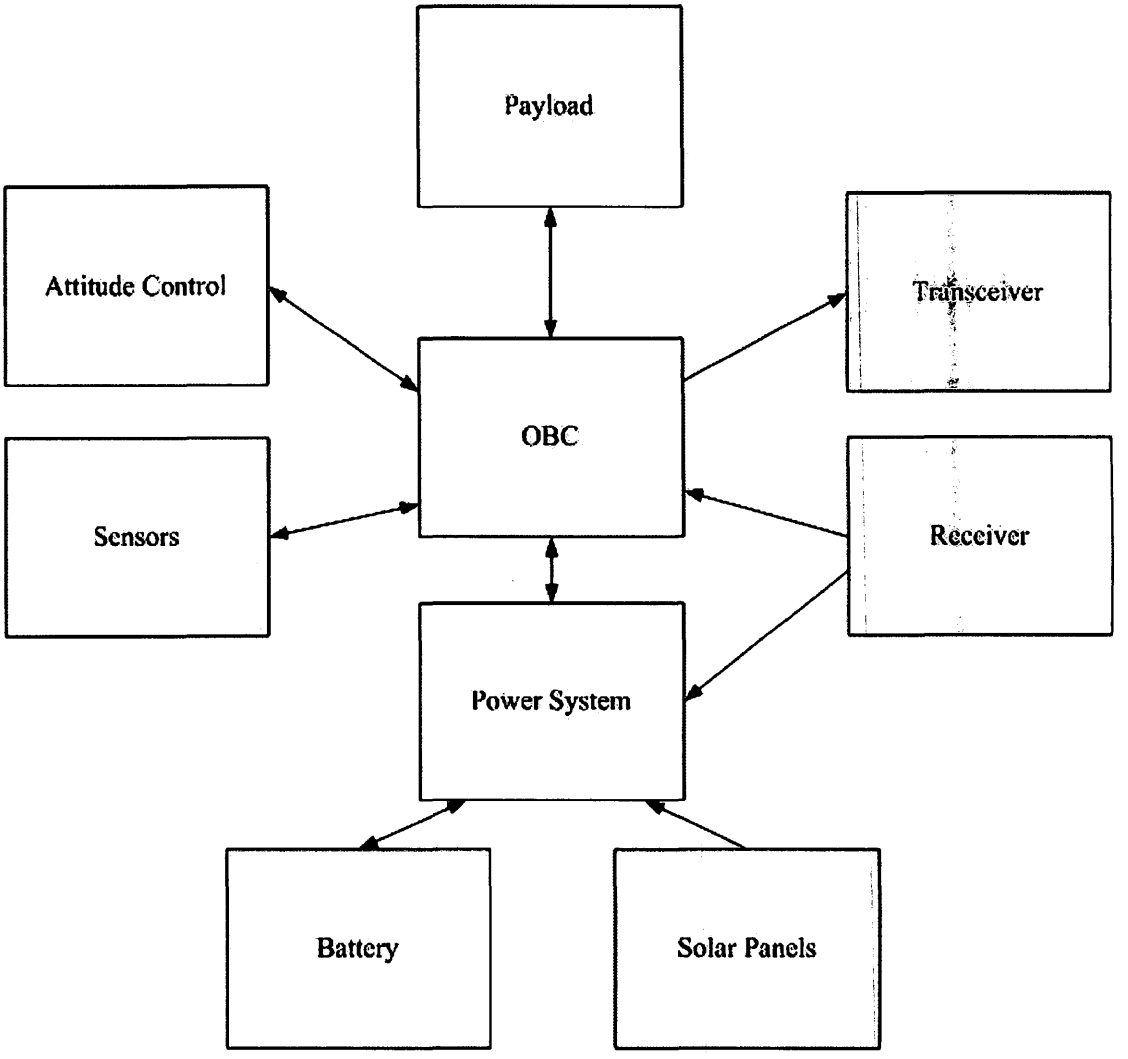

Figure 3 - Common Nanosatellite Configuration

Computers in the past used to be large and occupy whole rooms such as the Electronic Numerical Integrator And Computer (ENIAC) which was built in 1946 [4] for the purpose of computing firing tables and was used for other intensive computations as well. The ENIAC used vacuum tubes, which were used to control flow of electric current. The first major step to miniaturization came with the creation of transistors which could replace vacuum tubes and were smaller in size. Initial transistors were used in circuits as

discrete components. Development continued and integrated circuits were created which allowed the formation of large circuits incorporating transistors as well as other electronic components in a discrete package.

A modern computer can be broken down into two fundamental functional blocks: memory and a central processing unit [5]. The memory of a computer is used to store data and commands. Relevant information can be read from the memory and depending on the type of memory, information can be written to it. The central processing unit performs mathematical computations and is able to access the memory to retrieve information to process. The arrangement of the fundamental functional blocks, their connectivity and the resources arrangement is referred to as the computer architecture.

There are two main design architectures for a satellite regarding its OBC design. The first design involves a centralized architecture where the OBC acts as the main processing system and all data collected is passed on to the OBC to perform the necessary work. The second design, also referred to as a distributed architecture, involves subsystems that are able to perform their own processing and pass on the results to the OBC. The OBC in this case plays a more managerial role where it can relay data between subsystems and send data requests to the subsystems.

The centralized architecture is more common for simpler systems where subsystems can wait for the OBC to finish processing previous data. The subsystems can also share the OBC's processing power to work on several tasks using a scheduler and time allocation techniques. Such an approach may be necessary due to lack of space or power to support additional computing devices.

The distributed architecture is appropriate for systems where there is a large amount of data to be processed and subsystems are operating continuously. This is used to reduce the workload on the OBC and allows faster operation of the system as a whole. This approach is more appropriate for nanosatellite design as it contains several critical subsystems (e.g. power, communication), often as separate modules, which may require real time processing and collection of data continuously.

A hybrid approach is also possible where critical subsystems will have their own processing power while others will rely on the OBC to perform the necessary tasks.

The generic nature of the proposed OBC design allows it to be implemented within a larger system using any of the above approaches; although for a nanosatellite, the distributed or hybrid architectures are preferred.

The OBC design developed as part of the research can be configured for either design option. Such configurations are investigated on a mission specific basis and begin through identification of systems and whether processing power can be shared with other systems.

It is desired for the developed OBC to possess real time capabilities. Real time when referring to computer systems means that there are set time limits as to when a task must be accomplished by. A generic non real time system such as a desktop computer is designed to improve average time response. This does not prevent a task from waiting a long time before it is able to complete. A system with real time capabilities is designed to set and attempt to reduce the maximum time that a task may require to complete. Real time systems may be divided into two categories: soft real time and hard real time [6].

Soft real time systems guarantees that real time tasks will receive priority for execution but there is no guarantee that they will meet their timing constraints. Hard real time systems must guarantee that tasks meet their timing constraints and failure to do so produces a critical error.

1.1. Space Environment and Its Effects on Computer Systems

The present OBC design is intended for use aboard a nanosatellite and as such it is subject to the effects of the space environment. The space environment that affects the present OBC design includes vacuum, temperature extremes and radiation, each of which presents a challenge in the design of the system.

When a satellite is launched into space it is typically placed into a low Earth orbit (LEO) with various configurations as per mission requirements. The low Earth orbit range is generally accepted to be up to an altitude of 2000 km above sea level [7]. At these altitudes in typical nanosatellite missions, the atmospheric pressure and density are very low and the condition is considered a 'vacuum'. The vacuum is not a perfect vacuum as particles are present at low quantities which vary with altitude.

There are two effects of vacuum that are significant for satellites. Liquids in vacuum evaporate at temperatures well below their boiling temperatures. Liquid filled components that are not hermetically sealed and able to withstand the pressure difference rupture in vacuum. For example, electronic components such as electrolytic capacitors, which are filled with liquid, will rupture and be destroyed under vacuum conditions. Solid components are subject to an effect called outgassing where gasses that are trapped

within are released in vacuum condition. Outgassing materials are detrimental to spacecraft operations as they can attach themselves to parts of the spacecraft, such as solar panels and lenses, and reduce their performance.

The second effect of vacuum on satellites involves the lack of convection. Convection is an effective method of removing heat from a system. In a desktop and laptop computers, convection with the assistance of a fan prevents the system from overheating. Overheating of electronic components beyond their maximum rated temperature is generally destructive.

In low Earth orbit, the lack of an atmosphere causes extreme temperature variations dependent on being in view of the sun. When a satellite in low Earth orbit is hidden from the sun by Earth, the lack of atmosphere causes a rapid cool down due to a lack of energy being absorbed as well as heat energy being dissipated away from the satellite. When in view of the sun, the satellite is not protected by Earth's atmosphere and will be subject to the sun's radiation, raising its temperature. Combined with the lack of convection, an operational satellite may reach extremely high temperatures. The temperatures that a satellite can reach are dependent upon the altitude of the orbit, heat generated by satellite systems as well as time spent in view of the sun. This is important for electronic systems as components are rated to work within a particular temperature range. The operational temperature range of electronic components varies between components and it is a property that must be examined when selecting a device.

Radiation causes both failures and errors in electronics. The majority of radiation in low Earth orbit consists of energetic protons and electrons as well as heavier ions [8] that strike circuits. Electronic components are made out of doped semiconductor materials and when energetic particles strike them, they may cause single event effects (SEEs) and generation of charge within the semiconductor known as total ionizing dose (TID).

The effects of TID and SEEs are varied and may lead to build up of charge, shorting of regions that are normally insulated and switching in state. The results of these effects are memory errors and improper operation of components as well as complete failure.

Electronic components can be produced that are suited for operation in an environment containing radiation such as space. Such components are known as radiation tolerant and radiation hardened components. Radiation hardened devices are capable of operating in a radiation environment up to their defined limits. Radiation tolerant devices are capable of operating in radiation environments if protection schemes are implemented. Such components are more expensive than their equivalent non hardened or tolerant devices.

1.2. Historical Overview

Nanosatellites are an increasingly popular platform for educational institutions to educate their students in the area of satellite and space mission design with tangible results and even the possibility of an actual launch. The Nanosatellite Launch Program which is headed by University of Toronto Institute for Aerospace Studies (UTIAS) has launched several nanosatellites [9] built by universities and organizations from different countries. The advantages of nanosatellites are their lower cost and mission scope, which result in a

short design and construction phase in comparison to that of a larger satellite that can involve several years of development, high priced contractors and equipment, and proprietary development.

1.2.1. Previous Nanosatellite Computer Hardware Development

Nanosatellites have been flown from educational, commercial and governmental institutions and each has varying techniques of implementing their OBC. Research from the University of Stellenbosch has outlined the design of an Advanced RISC Machine (ARM) based OBC for nanosatellites [10]. Focus was placed on the type of microcontroller that is used (the AT91SAM7A2 ARM7 based processor) and its functionality in their research. The FPGA used for error detection and correction (EDAC), and on the qualification of the processor for the space environment were also examined in their study.

Research from UTIAS, who have successfully launched several nanosatellites, outlines the top level architecture of the OBC used on the CanX series satellites [11]. While the study focuses mostly on the overall nanosatellite configuration and the software for the on-going missions, it also includes a detailed description of the major components on the OBC. While specific parts are not named in the document, their functionality is described. In [11], it is seen that the OBC is composed of the bare essentials needed including a microcontroller for the processing and execution of software, volatile memory to provide an area for execution of tasks and non-volatile memory to store the necessary data as outlined by the mission and operational requirements. Focus is placed

on EDAC and it is done through the implementation of a triple modular redundancy (TMR) system in hardware by supplying three memory chips that allows the memory controller to use triple voting through a hardware voter.

Another design approach to a satellite OBC is to create a redundant scheme where there are two central units that are capable of working independently from each other. Researchers at the Politecnico di Torino have designed an architecture for a small satellite in which the OBC consists of two processing units which are not identical and can work independently or together [12]. This leads to a design where one faulty component of the OBC may be turned off and the other becomes the main and only processor. This provides an extra layer of protection against a critical fault in the OBC.

A common theme with the aforementioned designs is that they do not specify the use of any radiation hardened equipment as part of the system. Any component that is mentioned in all designs is a generic commercial component. This is a key issue in developing nanosatellites as the avoidance of radiation hardened equipment will result in lower cost of hardware.

In addition to the standard components used in existing nanosatellite OBC designs. The OBC that was developed as part of this research also contains Magnetoresistive Random Access Memory (MRAM), a type of non-volatile memory. MRAM is a promising technology and has been flown on Sprite-Sat as part of the magnetometer subsystem [13]. The magnetometer subsystem was designed by ÅAC Microtec and the satellite was

launched in 2009. The satellite was operating successfully; however, there is no specific information released about the performance of the MRAM component.

1.2.2. Previous Nanosatellite Software Development

A functional OBC must accomplish a set of tasks as outlined by mission requirements which may involve computations and data collection and storage. Those tasks are implemented and controlled through software.

Several educational nanosatellite developers have designed custom, low-level software to run their OBCs. The CanX series nanosatellites use custom software designed and written in house where all required functionality was implemented using direct calls to OBC hardware [11][14].

Developing custom software from the ground up requires significant work and therefore most likely results in increased cost and time. A way to reduce the time dedicated to the software aspect is to use an existing operating system that can be adapted to work on the OBC. Several spacecraft have used such commercial solutions. For example, VxWorks is a proprietary real time operating system that has been used on several space missions such as the Deep Impact spacecraft [15].

1.3. Thesis Statement and Contribution

This thesis outlines a hardware design of a nanosatellite OBC as well as the necessary software components in order to produce a working OBC subsystem that is suitable for

space missions. While focus is placed on nanosatellite applications, the present OBC design is also suitable to be used on ground applications such as rovers.

The objectives that drive the design are: power consumption, space environment survivability, low cost and standard interfaces. The power consumption of the system if important due to on orbit power availability which is limited to power generation through the use of solar panels. The space environment poses risks to electronics operation because of the temperatures the equipment is subject to as well as limited heat dissipation in space due to lack of convection in a vacuum. Radiation poses further risks as it may interfere with the operation of electronics.

In order for the design to be competitive with other commercial products it must be cost competitive as well as capable of easy integration into a nanosatellite system. Emphasis is placed on using the commercial-off-the-shelf (COTS) principle in order to create a low cost system through the selection of widely available and commonly used components.

The present OBC design is put through benchmark tests to evaluate its performance in comparison to commercial embedded systems. In additions, results from a qualifications test are presented as well to demonstrate the feasibility of using commercial components for the design of space systems.

1.4. Outline

The thesis is divided into 4 main sections. Chapter 1 details background information about the developments done with nanosatellites computers from a hardware and software perspective as well as presenting the main topic of the research.

Chapter 2 outlines several considerations in designing a computer system and describes the rationale behind the selection of the OBC components.

In Chapter 3, the focus shifts towards how the OBC runs and executes the required tasks. The types of software required are outlined and existing solutions are presented. After the options are investigated and the software components are chosen, a discussion is presented regarding how to integrate the software components together as well as the kind of customization that is required in order to make the software operational on the present OBC design.

Chapter 4 presents results from functional tests performed on the system. This includes functional tests that demonstrate the operation of the system as well as qualification tests that examine system operation under extreme conditions.

The final chapter summarizes important concepts and provides a closing discussion of the development of the present OBC design.

Chapter 2 – Hardware Design

The hardware development phase of the present OBC design is presented in detail. Initial stages prior to designing the hardware consist of evaluating constraints imposed on the design and comparison with existing commercial systems. Design of the hardware involves creating electronic schematic development in conjunction with component selection. Changing of a component may lead to changes in electronic schematics as it may change the electrical connections.

The printed circuit board (PCB) layout phase is started after the schematics are completed and involve creating physical connection between electrical components. Care is taken to create an efficient design that takes makes an effort to reduce noise in the system.

2.1. Hardware Constraints

The present OBC design is developed for nanosatellites and the design is based on the CubeSat specifications by the California Polytechnic State University. The specifications define 2 form factors which are the 1U CubeSat and 3U CubeSat. The 1U CubeSat dimensions are 100.0±0.1mm × 100.0±0.1mm × 113.5±0.3mm [2]. The limitation of the CubeSat standard as of 2012 is that it defines a satellite with a maximum mass of 4kg. A nanosatellite does not have to adhere to the CubeSat specifications in any way and therefore nanosatellites can exist in varying shapes and sizes. An example of such a nanosatellite would be the AISSat-1 which is constructed by UTIAS/SFL for the government of Norway. This nanosatellite adheres to UTIAS/SFL's own standard of the

Generic Nanosatellite Bus (GNB) and it measures 200mm × 200mm ×200mm [16]. The CubeSat standard was chosen as a guideline for the present OBC design due to it being an established standard with international flight heritage and as such it has compatibility with designs that also follow the CubeSat standard. Other nanosatellite designs can also make use of the present OBC design with the additional requirement that extra modifications may be necessary in order to mount and connect the present OBC design within those nanosatellite designs.

To adhere to the CubeSat standard, the chosen printed circuit board (PCB) form factor is the Pumpkin modified PC/104 Plus standards; PC/104 Plus standards define many aspects of a circuit board including dimensions, mounting holes as well as headers and many other features. Pumpkin Corporation has taken the PC/104 Plus standard and modified the types of headers that are used and have focused on the form factor and mounting holes that the standard defines. The headers that are defined in the PC/104 Plus specifications are shown in Figure 4. For comparison, the header mounting holes of the modified design are shown in Figure 5.The mounting holes at one side of the board in the PC/104 Plus specifications are omitted and the mounting holes on the other side are different, the PCI/104 plus uses PCI compatible headers while the Pumpkin design uses headers with different spacing.

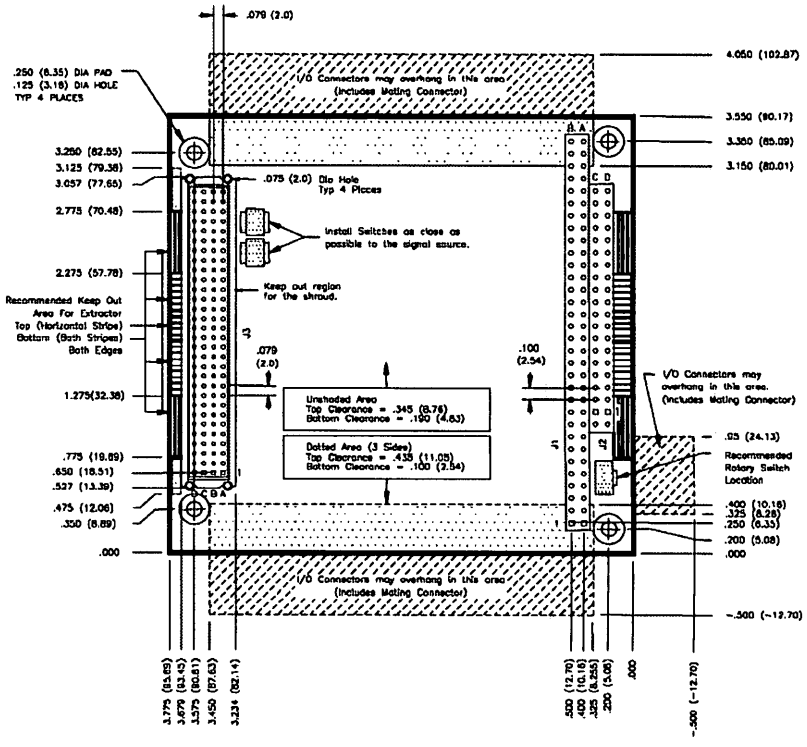

Figure 4 - PC/104 Plus Board with Headers Source Image: PC/104 Plus Specifications [17]

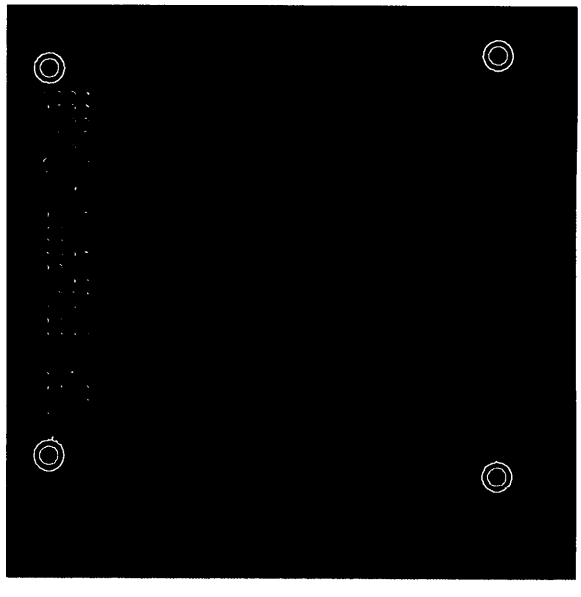

Figure 5 - Modified PC/104 Form Factor

For the present OBC design the Pumpkin PC/104 specifications are used as they are a commercial solution that provides all components necessary for constructing a CubeSat. Pumpkin CubeSat components have an extensive flight heritage [18], being used in over 10 satellites such as QbX1 and Rax. The present OBC design was created such that it is compatible with their equipment so as to provide an alternative to their motherboard that can be integrated with their equipment for use by nanosatellite designers.

The CubeSat and Pumpkin PC/104 specifications are driving guidelines for the present OBC design yet it should be noted that while the CubeSat is the ideal platform for the design, it can be used with other platforms with modification to either the design or the platform itself.

2.2. Available Embedded Computer Systems

As part of a trade-off study, several commercial embedded computer systems were examined in order to investigate and compare functionality, power consumption and price of those systems.

Gumstix is a company that sells several types of embedded system modules with varying components and capabilities. To get a complete overview of their system's capability, the specifications of the Overo FE COM system were reviewed as it is one of their top of the line systems with varying functionalities and enhanced operational temperature range. The dimensions of the Overo FE COM are the smallest of all the systems investigated but this comes at a cost where the system has headers at the bottom to allow connection to a daughter expansion board which then increase the size of the system, such boards can be

purchased from Gumstix or produced in house as the actual header connections are disclosed. The actual board design schematics are proprietary information [19].

Linuxstamp II is an embedded system designed and produced by The Linuxstamp Shop, owned and operated by Paul Thomas. The first iteration of the board was previously used at York University for the development of a micro-rover system for a Mars missions as well as by group of engineering students as part of their coursework project relating to nanosatellite communication systems. An advantage of this system is that the board schematics and layout are available to the public as it is an open hardware design [20] and the design can thus be modified as necessary.

The third embedded system that was examined is the LN2440SBC by LittleChips. This is a comprehensive system that includes many peripherals and board schematics are available with purchase of the board [21].

The final commercial embedded system that was investigated is the motherboard from Pumpkin Corporation. This system has the advantage that it is designed for nanosatellites and was designed with a large temperature range and low power consumption to accommodate the constraints of operating in the space environment. The motherboard does not have a processor by default and several options are sold separately by the company that are compatible with the motherboard. A summary of the different characteristics of the above systems are listed in Table 1.

Table 1: Summary of Commercial Embedded Systems

	Overo FE COM (Gumstix)	Linuxstamp II (Thomas, 2010)	LN2440SBC (LittleChips, 2008)	CubeSat Kit Flight Motherboard (Not including processor)
Price	$229.00	$120.00	$399.00	$1,200.00
Processor	Texas Instruments OMAP3530	Atmel AT91SAM9260	Samsung S3C2440A	Varies (socket accommodates several processor modules)
Instruction Set	ARMv7-A	ARMv5TEJ	ARMv4T	Varies
CPU	ARM Cortex-A8	ARM926EJ-S	ARM920T	Varies
Clock	600MHz	180MHz	400MHz	Varies
Temperature Range	-40°C - 85°C[2]	Unknown	Unknown	-40°C - 85°C
Memory	256MiB RAM	32MiB RAM	64MiB RAM	On separate processor module
	256MiB Flash	256MiB Flash	32MiB Flash	On separate processor module
	SD Card	SD Card	-	SD Card
Connectivity to System	Custom or prebuilt expansion board	Custom or prebuilt expansion board	Headers	Headers
Interfaces	Bluetooth, Wi-Fi, USB, SPI, I2C, UART, PWM, GPIO, JTAG, ADC	USB, SPI, I2C, PWM, GPIO, JTAG, ADC, Ethernet	USB, SPI, I2C, PWM, GPIO, JTAG, ADC, Ethernet	USB, SPI, I2C, PWM, GPIO, JTAG, ADC[3]
Design	Proprietary	Open hardware	Proprietary layout	Proprietary
Shipped OE	Open Embedded	EmDebian	Linux kernel	Purchased separately
Dimensions	17mm × 58mm × 4.2mm	41mm × 72.5mm × 15.3mm[1]	140mm × 78mm	96 mm × 90.2 mm × 12.5 mm

[1] *Linuxstamp II dimensions were obtained manually with calipers*
[2] *Exception to Bluetooth, Wi-Fi and SD card modules*
[3] *All require external processor module*

Examining the price ranges of the embedded systems it can be seen that the CubeSat Kit Flight MB is significantly more costly than the others, especially due to the fact that it

also requires a pluggable processor module to be complete, which will set its price even higher.

Knowledge of the processor, instruction set and processor that each embedded system uses is important for writing software and compiling it.

Only the Overo FE COM and the CubeSat Kit Motherboard provide a temperature range at which their systems are able to operate. With the Linuxstamp II being an open hardware design, the bill of materials is publicly available and the temperature ratings of all components can be obtained. Operational temperature range is an important characteristic of a system when considering operation in space. In space the system will be subject to extreme temperature ranges, which are dependent on several factors including but not limited to the satellite orbit. A system with a narrow operational range will require extra care to make sure that the temperature it is subject to is within its allowed limits.

2.3. Component selection

A computer system is composed of many types of circuits working in conjunction. Integrated circuits have revolutionized the electronics industry by providing complicated circuits on a single die and thus computers became smaller and more powerful. It is required to select the appropriate components in order to design and construct a functional system. The following sections present a description of the selection of the critical components and their role in the present OBC design.

2.3.1. Processor

The function of a digital computer is to perform computations. With the invention of integrated circuits, it is possible to manufacture commercial chips that act as an integrated computer. There are several types of integrated circuits that act as an integrated computer, and in this research the focus is placed on 3 types of such integrated circuits: application specific integrated circuits (ASIC), field programmable gate arrays (FPGA) and microcontrollers.

ASICs are designed to accomplish specific tasks, but not suited for general purpose use. Using an ASIC for a nanosatellite requires designing the ASIC and getting it fabricated. A mistake in the design may require a new design to be created. Redesigning and fabricating a new ASIC requires extra time and money [22]. As the research is focused on designing a general purpose OBC for nanosatellites, the inflexible nature of ASICs puts them at a disadvantage; other options on the market can provide the necessary flexibility without the need to fabricate custom components.

FPGAs are components that are mass produced and are available commercially but are still very flexible as developers can program the device to perform the functions they need. FPGAs work by having developers program the logic connections in them directly to form digital circuits [8]. The main drawback of FPGAs is the necessity to program the built in logic gates to create the required digital circuits.

Radiation tolerant and radiation-hardened FPGAs are designed to be more resilient to the effects of radiation. Both types cost significantly more than an equivalent FPGA due to

the different fabrication processes required, extra testing and validation in development and a limited market demand for them. Non-radiation hardened components are as susceptible to single event effects (SEE) as any other CMOS device [23], including memory modules and microcontrollers.

Radiation hardened and radiation tolerant FPGAs are commonly used in commercial spacecraft as they provide the necessary flexibility to be customized to perform a variety tasks from communication interfaces to digital signal processing as well as protection from effects of radiation that will prolong the life of the mission and reduce the chances of errors occurring. The flexibility of FPGAs means that less variety of radiation hardened components are required as they can be programmed to perform different functions.

FPGAs have been flown on a variety of missions such as the communication satellite Optus C1 launched in 2003. The popularity of FPGAs in space means that future missions also plan to use them as part of their design such as the UKube-1 CubeSat planned to launch in 2013. Both carry FPGAs designed by the company Xilinx. UKube-1 is planned to carry radiation hardened FPGAs in order to test and validate their performance in space conditions [24]. Optus C1 is carrying radiation tolerant FPGAs to perform data processing as part of the communication system [25].

There are 3 main types of FPGAs, they are antifuse, Static random access memory (SRAM) and Flash based. Antifuse FPGAs are one time programmable devices which contain antifuses. An antifuse is a component which conducts electricity when it is burnt.

By selecting the antifuses to burn, logic gates can be connected to form a digital circuit. There is no way to restore them to a non conductive state which is why they can be programmed only once.

SRAM based FPGAs are volatile and require constant power to maintain their configuration, they are programmed on every start up from either an external source or an internal Flash source. Flash based FPGAs are non volatile and will maintain their configuration when there is no power.

When examining the power consumption of the 3 types of FPGAs it can be seen that SRAM based FPGAs have high power demands due to their volatile nature as the entire configuration requires a constant power supply in order to maintain its state while Flash based FPGAs and antifuse FPGAs do not have such an issue and therefore have lower power consumption in general.

The third examined technology is microcontrollers. Microcontrollers are integrated circuits that contain circuits designed for specific functions. A key difference between FPGAs and microcontrollers is that with an FPGA a developer designs the circuits using the logic gates on the FPGA while with a microcontroller the circuits already exist and can be configured through programming.

Modern microcontrollers provide many capabilities such as timing circuits, communication peripherals and interfaces to common devices (external memories, clocks, etc.). Operation of microcontrollers is performed by writing to hardware registers and those in turn manipulate the circuits that they control.

Radiation hardened microcontrollers also exist such as Aeroflex's UT80C196KD, though the commercial selection of such devices appears to be more limited in comparison with radiation tolerant and hardened FPGAs. CanX-2 CubeSat from UTIAS/SFL contains an ARM7 microcontroller, model number AT91M48200A [14], which is a non radiation hardened or tolerant component. CanX-2 was launched in 2003 and is still operational as of 2012. SwissCube-1 is a CubeSat designed by Ecole Polytechnique Fédérale de Lausanne and it uses a microcontroller as part of the attitude control system. The microcontroller that SwissCube is using is the MSP430F169 made by Texas Instruments [26], which is also a non radiation hardened component.

Examining the 3 technologies, the microcontroller stands above the rest as it requires the least amount of customization. Microcontrollers are commonly used in ground based applications and have been used in space applications as well. FPGAs require the use of a complete configuration of each circuit to be used while microcontrollers have all circuits built in and are simply selected. Through the availability of common interfaces on a typical microcontroller, it can be connected to other systems through the use of its built in circuits and hardware registers.

There are many different types of microcontrollers that can be chosen and many of them are equivalent in capabilities such as the amount of peripherals present, operational temperature range, power requirements and price. The AT91SAM9G20 was chosen as The Linuxstamp I and II used its predecessors the AT91RM9200 and the AT91SAM9260 respectively. Several months after the AT91SAM9G20 was chosen to be used on the

present OBC design, a second version of the Linuxstamp II was released with the AT91SAM9G20 as its microcontroller. The OBC schematics were complete at that stage and work was focused on the layout.

2.3.2. Memory

Memory is used to store data and programs that are used by the computer. The microcontroller itself contains embedded Read Only Memory (ROM) and SRAM which are 64KiB and 32KiB in size respectively [27], and is configured to boot from external memory storage.

Memory can be classified into 2 categories: volatile and non-volatile. Volatile memory is a type of memory that requires constant power in order to maintain the information stored within it. Non-volatile memory will retain its data even when no power is supplied to it. Traditionally, volatile memories provide faster access times to the data stored within them and have a higher tolerance to wear than non-volatile memories. Exceptions may be found in emerging technologies and highly advanced and expensive products, one of which will be discussed later. The problem of such technologies is that they are expensive. An everyday example is seen when comparing solid state drives to hard drives where the solid state drives are more expensive per unit of memory. Due to the limitations and advantages of the 2 categories of memories as outlined above, it is therefore common practice to have both types of memory if space permits.

While it is possible to have only non-volatile memory on the present OBC design, all results from computations and executing processes have to be stored in a dedicated

partition of that memory, continuous operation causes wear on the memory circuitry and the slow access times are apparent when addressing large amounts of data.

The OBC therefore contains 2 main types of memory as well as a third type to be used as backup and to evaluate its performance.

Non-Volatile Memory

The selection of a non-volatile memory for a nanosatellite mission depends on several criteria: size, power consumption, price and durability. In space, access to the satellite may not be available at all times of day. Data that the satellite collects and produces must be stored in a memory location so that the satellite can access it at a later time, possibly to transmit it to a ground station.

A comparison of several types of non-volatile memory technologies are listed in Table 2, the devices chosen are commercial devices that are available directly from distributors. The characteristics portrayed are typical to devices from other manufacturers. Memory size is a critical criterion for selecting the non-volatile memory. The memory is used to store both the software for the nanosatellite operation as well as any data gathered by any sensors or other subsystems that are a part of the nanosatellite. As mentioned previously, constant access to a nanosatellite may not be possible and it may be desired to store data on-board until it can be retrieved. Being a generic system, the present OBC design may be connected to a variety of sensors, each with its own memory requirement. It is also not possible to predict the size of the control software itself.

Chapter 3 outlines minimum memory requirements for the software, in the form of a kernel and filesystem, to be used on the present OBC design. The kernel and filesystem provide a basic structure upon which applications to control a nanosatellite can be executed.

Table 2: Comparison of Non-Volatile Memories

Type	Nand Flash [28]	Nor Flash [29]	FRAM [30]	EEPROM
Device	MT29F2G08ABAEAWP-IT	MX29GL256FLT2I	FM23MLD16-60	M95M02-DR
Manufacturer	Micron Technology	Macronix	Ramtron	STMicroelectronics
Cost	$7.20	$5.78	$72.04	$6.11
Memory	256MiB	256MiB	1MiB	256KiB
Durability	10^5 cycles	10^5 cycles	10^{14} cycles	10^6 cycles
Power[1]	115mW	66mW	46.2mW	16.5mW

[1] *Power estimated from peak current estimates in datasheet and operational voltage*

The high cost and low memory size of Ferroelectric random access memory (FRAM) eliminate it as a viable option for use as a main memory. As outlined in chapter 3, the Linux kernel by itself is around 1.0MiB and therefore there would be no room for any other software or data. The Electrically Erasable Programmable Read Only Memory (EEPROM), though inexpensive, contains less memory than the FRAM and therefore is also not a viable option.

Flash memory is a commonly used technology, often seen in the form of Flash Drives. Further characteristics of the 2 types of Flash technologies are listed in Table 3. It is important to note that the timing characteristics were estimated based on information in the product datasheets and can deviate from the timings presented depending on the order of operations. The NAND Flash exhibits faster timing characteristics in all operations as

well as a lower pin count. A lower pin count leads to a simpler layout which is a strategic choice when dealing with limited PCB space.

Table 3: Comparison of Flash operations

Specific device	MT29F2G08ABAEAWP-IT [28]	MX29GL256FLT2I [29]
Manufacturer	Micron Technology	Macronix
Pins	23(8 bit)/31(16 bit)	51
Read time per byte[1]	12ns	20ns
Write time per byte[1]	100ns	5500ns
Erase time per byte[1,2]	5.5ns	570ns

[1]*Computed using estimates from datasheet*
[2]*Erase operation is performed on bulk sections and cannot be performed on smaller sections*

Volatile Memory

Volatile memory is needed for the OBC in order to prolong the lifetime of the system by reducing the wear on the non-volatile memory. This is achieved by allowing the microcontroller to store and retrieve temporary data from the volatile memory as opposed to the non-volatile memory. There are several types of volatile memory that can be used and they can be divided into 2 major groups: static random access memory (SRAM) and dynamic random access memory (DRAM).

The key difference between the 2 groups is that DRAM uses capacitors to store the memory state and because capacitors lose their charge over time, it must be refreshed periodically [31]. SRAM does not suffer from the same problem as the state of an individual bit is stored within a network of transistors [32] [33]. In both cases, once the power to the memory is lost, the state of the bit storage circuits is lost with it. A key

advantage of DRAM over SRAM is that it has higher memory density as well as cheaper cost per bit [34]; for the purpose of nanosatellite OBC design, therefore, DRAM is preferred to SRAM.

The most commonly used DRAM variants are the synchronous DRAM (SDRAM) type memory. SDRAM has also developed over the years and new types of SDRAM have appeared in the market known as double data rate SDRAM (DDR SDRAM, or simply DDR) and the original synchronous design has been renamed to single data rate SDRAM (SDR SDRAM, or simply SDRAM). The difference is that DDR is able to obtain double the data rate transfer by transferring the data on both the falling and rising edge of the clock while SDR uses only one of the edges.

The chosen microcontroller, AT91SAM9G20, provides a built in controller for SDRAM but it only supports the SDR SDRAM type as it does not provide pins for the additional control pins that are required for a DDR type memory [27]. The lack of support for DDR limits the system to the use of SDR SDRAM. This can be seen as a benefit to nanosatellites as the frequency of operation increases, power consumption may increase.

Several manufacturers sell SDR SDRAMs commercially but Micron Technology products dominate the market and are available for purchase at small volumes from distributors. Products from other manufacturers require minimum orders of 1,000 to 2,500 units.

There are several important criteria for choosing an SDR SDRAM (henceforth referred to as SDRAM), they are memory size, price, speed, temperature range and power

consumption. The SDRAM units manufactured by Micron Technology are similar in functionality and requirements and therefore the most cost effective solution was chosen.

The SDRAM that is used on the present OBC design is the MT48LC16M16A2 manufactured by Micron Technology. A summary of the characteristics of the chosen SDRAM are listed in Table 4.

Table 4: Specifications of Chosen SDRAM [35]

Part	MT48LC16M16A2P-75 IT:D TR
Manufacturer	Micron Technology
Voltage	3.3V
Capacity	32MiB
Bus Width	16 bit
Price	$8.73
Temperature Range	-40°C ~ 85°C
Power (Active Read/Write)	0.45W
Power (Active Standby)	0.13W

Experimental Memory

An additional 3rd type of memory was added to the OBC design for the purpose of evaluation of its performance in the space environment. The magnetoresistive random access memory (MRAM) is a non-volatile memory that works by changing the state of a bit cell structure through a magnetic field. The magnetic field causes the bit cell to become a high resistance structure or a low resistance structure which maintains its state once the magnetic field is removed [36], several technologies exist that accomplish that goal.

The often cited advantages of MRAM are its high speed, non-volatility, high data retention and unlimited endurance. Development of MRAM has a focus of the technology to become an alternative to both Flash and SDRAM. In reality the technology of MRAM is not yet at a stage where it is able to compete with Flash as the memory density of MRAM is significantly lower. The Flash that was chosen for the OBC is 256MiB in size while a typical commercial MRAM is 2MiB in size. In addition, the speed of MRAM, while surpassing that of Flash, is not on par with SDRAM. Nonetheless, the companies and researchers working on MRAM technology have made progress over the years and are forecasting the possible replacement of Flash by MRAM.

The inclusion of the MRAM on the present OBC design is to utilize the nanosatellite platform as a vehicle suited for technology demonstration. An MRAM component from ÅAC Microtec has flown on a Japanese satellite in 2009. The magnetometer that the MRAM was a part of was reported to have worked as intended; however, there are no public reports of the performance in space of the MRAM component such as bit errors, speed and power consumption.

The MRAM on the present OBC design is intended as a backup memory where a copy of the bootloaders and kernel will be stored. The software within the MRAM will be compared to software in the NAND Flash. The research of MRAM performance in space conditions is promising and ground tests deemed its performance satisfactory though there is concern of possible latchups occurring [37], though this is true for other CMOS based devices which are not radiation hardened.

The commercial availability of MRAM is limited as it is not a commonly used technology. The MRAM devices used on board Sprite-Sat are manufactured by Freescale Semiconductors though they are not easily obtainable in low volumes. Everspin semiconductor is another manufacturer that has MRAM components on the market available in low quantities. A comparison between 2 parts from the 2 manufacturers is displayed in Table 5. The determining factor for selecting a component was the ready availability of the Everspin Technologies product.

Table 5: Comparison of MRAM Products

Part Number	MR4A16BCYS35	MR2A16ATS35C-ND
Manufacturer	Everspin Technologies	Freescale Semiconductor
Price	$40.07	Unavailable
Memory Size	2MiB	2MiB
Power Supply	3.3V	3.3V
Operating Temperature	-40°C - 85°C	0°C-70°C
Operating Power Consumption	0.36W (write), 0.198W(read)	0.34W(write),0.181W(read)
Data Bus Width	16 bits	16 bits
Rated Data Lifetime	>20 years	>20 years
Package	54 TSOP	44 TSOP II

2.3.3. Voltage Regulator

The main function of a voltage regulator is to maintain a steady voltage level at its output. Voltage regulators are also able to convert an input voltage level into a stable output at a different level. The OBC is composed of several active integrated circuits that require 2 voltage levels to operate; Table 6 lists the components and their operational

voltage levels. It is of note that the microcontroller requires 2 voltage levels to operate the circuitry in it.

Table 6: Operational Voltage Levels of Active Components

Device	Voltage Level	
	Nominal (Acceptable Range)	
Microcontroller	3.3V (3.0V - 3.6V)	1.0V (0.9 - 1.1V)
NAND Flash	3.3V (2.7V - 3.6V)	
SDRAM	3.3V (3.0V - 3.6V)	
MRAM	3.3V (3.0V - 3.6V)	

The inclusion of a voltage regulator as part of the present OBC design stems from making the OBC generic and independent from the power subsystem on a nanosatellite. The implementation of a voltage regulator system allows the OBC to accept a single power input at a range of values and regulate it itself. The OBC is a critical subsystem of a nanosatellite and it is therefore desired to keep its power input steady. Connection to external power supplies can result in noise on the lines and voltage spikes can cause the voltage to go out of operational bounds. A voltage regulator on the OBC will minimize the noise and provide a buffer to protect the OBC from voltage spikes.

Voltage regulators are grouped into 2 main categories: linear regulators and switching regulators (hybrids also exist). The linear regulator uses a feedback loop from the output to adjust the output voltage level to the desired voltage level [38]. The greatest disadvantage of linear regulators is their inefficiency. The input and output current of a linear regulator must be identical and as such the difference in voltage between the input and the desired output is dissipated as heat.

Switching regulators work by switching its output on and off using a circuit that is more complex than that of a linear regulator. During its on stage, the regulator transmits all the power from its input and during its off stage, it transmits no power [38]; this makes the switching regulator more efficient than a linear regulator as it does not dissipate the power difference between the input and output. The time that the device is off and on (the duty cycle) controls the output voltage at the output.

For the present OBC design, a switching regulator is preferred to a linear regulator based on its efficiency. To meet the requirement of 3.3V and 1.0V supplies, a voltage regulator with dual outputs is needed.

The LTC3521 is a switching regulator from Linear Technology and is selected because it contains 2 buck converters and 1 buck-boost converter. The buck converters are set to output 1.0V and 3.3V each with a maximum current of 600mA to meet the voltage requirements of the OBC components. The buck-boost converter is set to output 5V; this output is mostly used as a generic output that can supply power to other components. A level of 5V is selected as it is a commonly used voltage level in digital electronics.

A buck converter steps down the input voltage to a lower voltage level up to the level of the input voltage. Therefore the LTC3521 in theory requires an input of at least 3.3V in order to output a stable 3.3V and 1.0V to power the OBC. Inefficiencies due to energy loss in the form of heat in the regulator as well as imperfections of control resistors cause a deviation of the output voltage compared with the theoretical output voltage. Figure 6 and Figure 7 show the output from the regulator for the 3.3V and 1.0V output

respectively. The 1.0V output is within an error of 1.2% of the theoretical value but the 3.3V output reaches an error below 1% only after an input voltage of 3.4V. At an input voltage of 3.3V, the output is 3.186V which amount to 3.45% difference. The 1.0V output is within 1.2% of the theoretical value because the input voltage is much higher and in fact the minimum input voltage to the regulator is rated at 1.8V.

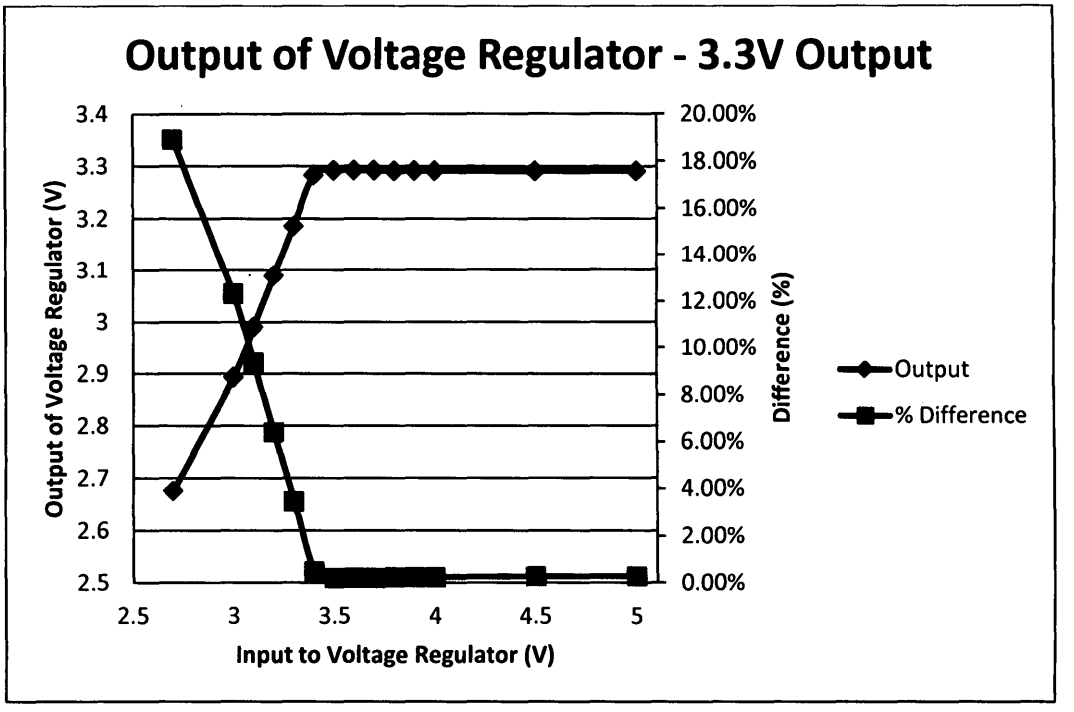

Figure 6 - Output of Voltage Regulator, 3.3V

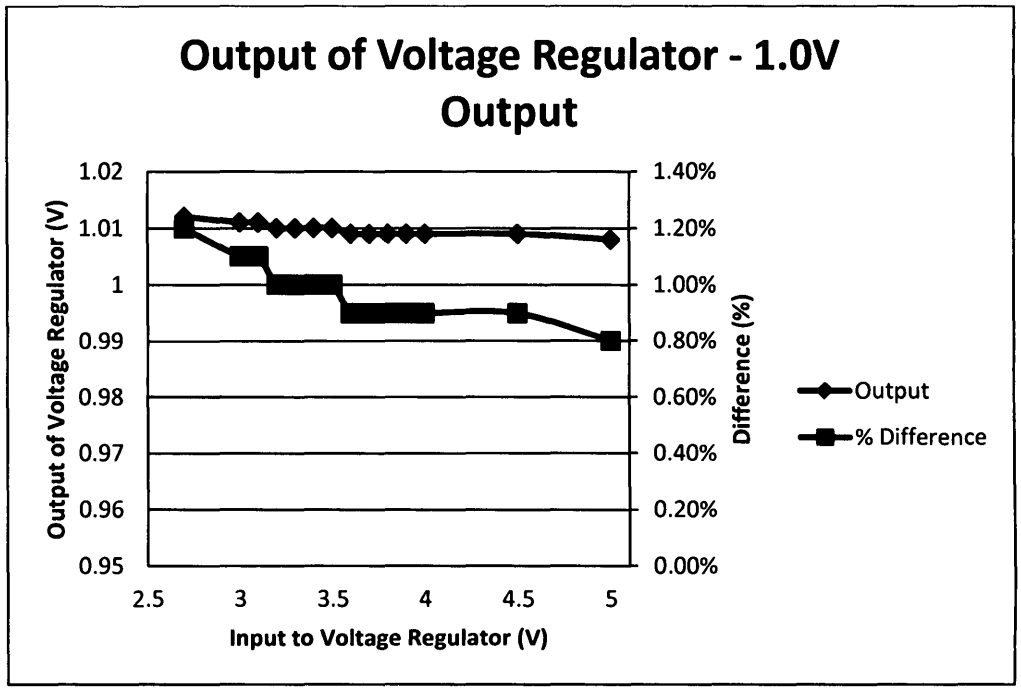

Figure 7 - Output of Voltage Regulator, 1.0V

2.3.4. Passive Components

Passive components require no input power to work and therefore cannot amplify power. Power amplification occurs in active components such as transistors where an input signal can be amplified through the use of the power input to the active component. The passive components used on the OBC include resistors, capacitors and inductors. Due to space limitations, only surface mount components are used for the present OBC design. While each component is expected to accomplish a simple task (e.g. a resistor impedes the flow of current) there are different manufacturing processes as well as materials that can be used to manufacture the components. Different components of the same type can have different properties due to the differing manufacturing processes and materials used.

Resistors

The resistors on the OBC are mostly used as part of an external circuitry that is required by the integrated circuits to function, for example, the voltage regulator requires resistors in a voltage divider configuration to set the output voltage.

The 4 main types of resistor technologies and their properties are shown in Table 7, The values are based on bulk commercial availability and does not indicate special orders directly from manufacturers.

Metal foil resistors possess characteristics, such as radiation tolerance, that make them desirable for satellite use from a performance standpoint. Radiation causes resistors formed from semiconductor material to change their resistance by doping. This is not an issue for resistors formed from thin metal, such as metal foil [39]. Their high stability and resistance to radiation effects makes them suitable for space applications according to the manufacturing companies. A disadvantage of metal foil resistors is the large price per resistor as well as their availability. Getting the resistors with the necessary resistance values per the OBC design requires getting them from manufacturers. The present OBC design requires 11 different resistance values to be used throughout the design. A single metal foil resistor costs on average $17 from distributors and few resistance values are available. The high cost of metal foil and the requirement to purchase in bulk directly from manufacturers results in a high component price. The various types of resistor technologies are listen in Table 7.

Table 7: Resistor Technologies

	Resistor Type			
	Thin Film	Thick Film	Metal Film	Metal Foil
Temperature coefficient	100-350 ppm/K	150-800 ppm/K	350-1500 ppm/K	0.2 ppm/K
Tolerance	0.01%-20%	0.1%-30%	0.1%-5%	0.01%-0.02%
Typical Price (200Ω)	$0.6 @ 0.1%, 25ppm/K	$0.08 @ 1%, 100ppm/K	$1.26 @ 0.1%, 25ppm/K	$17.53, not for 200 Ω
Resistance ranges (commercially available)	0 - 9.1MΩ	0 - 100GΩ	0.1Ω - 1MΩ	100Ω, 350Ω, 1 - 10KΩ
Composition	Thin layer of resistive material, deposited through sputtering, resistance adjusted through trimming.[40]	Mixture of metals and ceramics deposited onto a ceramic base. Can be 3 to 10 times thicker than thin film.[40]	Thin metal film applied onto a ceramic substrate.	Metal cold rolled foil is cemented onto a ceramic substrate and photoetched to a desired resistive value.[40]

Metal Film resistors are thicker than the metal foil resistors and tend to have higher temperature coefficient and they are often used as temperature sensing devices. Thick films, while significantly cheaper than the other types, have poor tolerances as well as large temperature coefficients while thin film resistors posses desirable characteristics. Research on specific types of thin film resistors showed that the radiation effects on the value of their resistance is deviated from their initial values by at most 2% [41].

For use in nanosatellite systems, such as the present OBC design, the thin film resistors are the optimal choice due to their availability, low cost, radiation tolerant properties and low thermal coefficient.

Capacitors

Capacitors are passive components that store energy when a voltage is applied across the two terminals. A useful property of capacitors is their ability to allow AC currents through while blocking DC currents. This property leads to capacitors being often used to filter noise from a system by passing the noise directly to the ground of the system. Like the resistors, the role of capacitors on the present OBC design is to serve as external components to integrated circuits as well as decoupling capacitors.

In addition to filtering out noise, the decoupling capacitors are used when devices change their state which in turn causes a change in their power requirements. When a component without a decoupling capacitor switches state, it draws energy from its power input. Time lag caused by the supply line not being able to instantaneously meet the new current demands causes a change in the voltage level of the device due to the change in power consumption. Putting a decoupling capacitor close to the power input of the device causes the necessary current to be obtained from the decoupling capacitor and not from the power supply directly. Obtaining the current from the decoupling capacitor prevents noise from leaking onto the power supply line. The key to decoupling a device is to place the capacitor as close as possible to the power and ground pins of the device [42].

The type of capacitors often used by the National Aeronautics and Space Administration (NASA) for space applications are glass capacitors. Properties of glass capacitors include high stability, radiation tolerance and large operational temperature range [43]. One of

the space applications of glass capacitors was on board the Cassini-Huygens spacecraft where the capacitors are used as part of the communication and control circuitry [44].

Glass capacitors are expensive and hard to obtain, only one capacitor was found at an online distributor for a price of $1,161 [45]. Several types of capacitor technologies exist and for this research, the most common were considered to make sure that they are commercially available for low volume orders. The types of capacitors examined are shown in Table 8. The various technologies used for manufacturing capacitors yield varied properties and limitations. An entry that is absent from Table 8 is the electrolytic capacitor. This type of capacitor contains a liquid inside known as an electrolyte. The vacuum environment of space will cause the electrolytic capacitor to burst due to the pressure of the electrolyte inside and will destroy the capacitor, causing outgassing. The electrolyte itself may then cause damage to surrounding components by attaching to sensors or shorting conductors.

Table 8: Capacitor Technologies

	Capacitor		
	Aluminum	Ceramic	Tantalum
Operating Temperature	-55°C - 105°C	-55°C - 125°C	-55°C - 125°C
Tolerance	20%	1%, 2%, 5%, 10%, 20%	10%, 20%
Capacitance Ranges	0.1uF - 10mF	0.1pF - 47uF	0.047uF - 100uF
Polarity	polar	non polar	polar
Composition	2 aluminum foils with dielectric materials between them [46].	Ceramic dielectric materials with metal layers acting as electrodes [47].	Anode composed of tantalum, dielectric material grown on anode and a conductive cathode surrounding it [48].

The capacitors that are most abundant on the present OBC design are decoupling capacitors. The datasheets for the different integrated circuits all recommend the use of 0.1uF value capacitors for that purpose. For decoupling capacitors, the ceramic types are most commonly used [42] because tantalum and aluminum capacitors can be easily damaged if polarity is reversed.

Tantalum capacitors are often used together with ceramic capacitors to provide filtering on power supplies. The only tantalum capacitors used on the present OBC design are part of the voltage regulator circuit. They provide higher capacitance and can store a larger amount of energy per unit volume. The switching regulator duty cycle causes the power requirements to constantly change and a larger store of current is needed.

Inductors

There are two applications for inductors in the present OBC design. First, inductors are used as part of the voltage regulator external circuitry. Inductors are required for every converter on the voltage regulator to store energy in order to produce the necessary output at the correct voltage level. The value of the inductor was taken from the LTC3521 datasheet to be 4.7uH to generate the required outputs.

The second application of an inductor requires the use of a specific type of inductor known as a ferrite bead. Ferrite beads are special components that have the properties of inductors at low frequencies and of resistors at high frequencies. This is useful in filtering noise as a high frequency noise will experience large electrical resistance from the ferrite bead and will be dissipated as heat. Ferrite beads are used in the present OBC design at

the input to the voltage regulator to filter the input. A ferrite bead is used at the USB connector power pin to filter the power coming from a USB connector that could be connected to the present OBC design.

Inductors generate a magnetic field during operation that can interfere with adjacent circuits. A method to contain the magnetic field of the inductor is to use a shielded inductor.

2.4. Design Layout

The layout of the present PCB design was created using Altium Designer [49], a software package for electronic systems design. All connections were done manually as opposed to automated routing in order to create a design with the shortest traces and least amount of vias. A simplified view of how the major components on the OBC are connected is shown in Figure 8.

A PCB is composed of layers of copper and insulating material pressed together with a resin. The copper layer is etched away to form a variety of features with the most common being pads and traces. Pads are formed on the external layers of the PCB and components (e.g. capacitors, resistors, integrated circuits) are attached to them with solder. Traces serve to conduct electricity throughout the PCB and connect the different components together. Traces can be formed on the external as well as internal layers of copper.

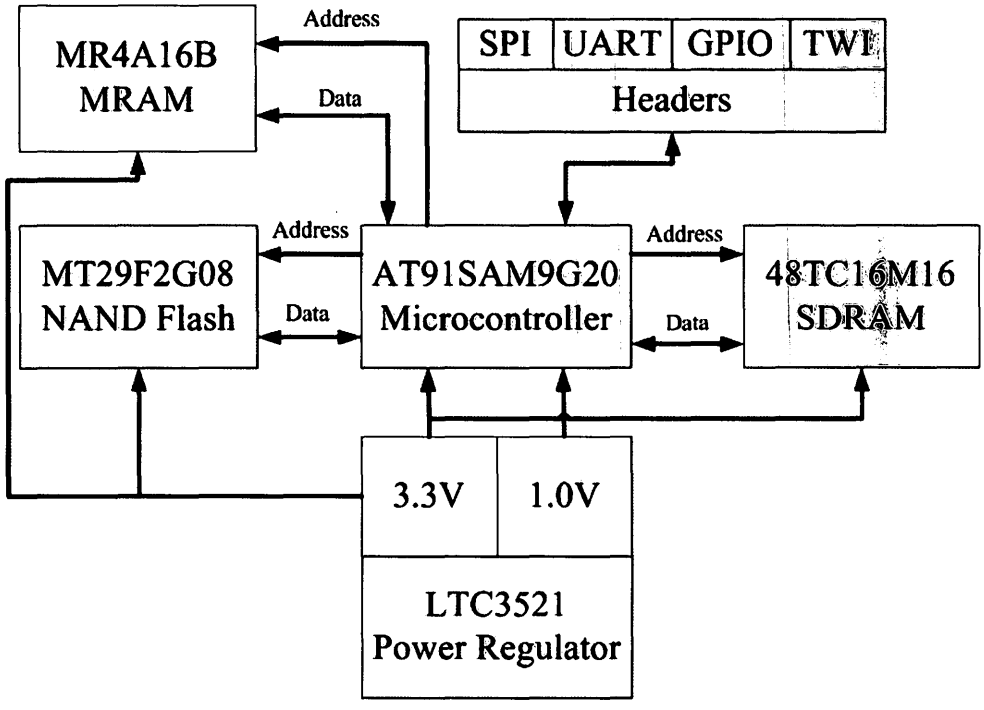

Figure 8 - Hardware Block Diagram

Connecting copper traces that are present on separate layers is accomplished through vias. A via is a small hole with walls coated in a conductive material. The conductivity allows a via to be used as a path for electricity to move between the layers of the PCB.

In order to produce a design that adheres to current technological fabrication abilities it was necessary to contact a company to fabricate the design. Investigating fabrication capabilities is an important step as it provides restrictions that have to be considered on the design.

Two very important aspects of PCB layout considered are the copper trace minimum size and the number of layers in the design. The choice of a minimum trace size is dependent

upon several factors including the (1) maximum current through the trace, (2) clearance between component pads and vias and (3) manufacturing capabilities.

Fabricators often publish their routing capabilities and required clearances based on a standard 1oz copper thickness. 1oz copper is defined as the resulting copper thickness when 1oz of copper is used to cover an area 1 square foot in size. This translates to a thickness of 1.4mil (0.03556mm).

It is possible to obtain larger thicknesses of copper such as 2oz and above. The drawback of using a thicker copper thickness is that it increases the minimum feature size and clearance that a manufacturer can achieve. It is required to examine the components on the PCB and determine whether the components themselves pose physical limitations. For the present OBC design, the microcontroller is in a ball grid array (BGA) package. A BGA package has all the pins of an integrated circuit distributed on the bottom side of the package and no leads are protruding, as is shown in Figure 9. The circles shown are the pins which are in a shape of small protrusions in the form of part of a ball. Routing of copper traces and vias that are placed between the pins of a BGA package must be small enough in order to provide access to the pins of the microcontroller and avoid causing shorts. Using traces that are too narrow may increase fabrication costs as more expensive methods are required, as well as the narrow traces have less current carrying capacity.

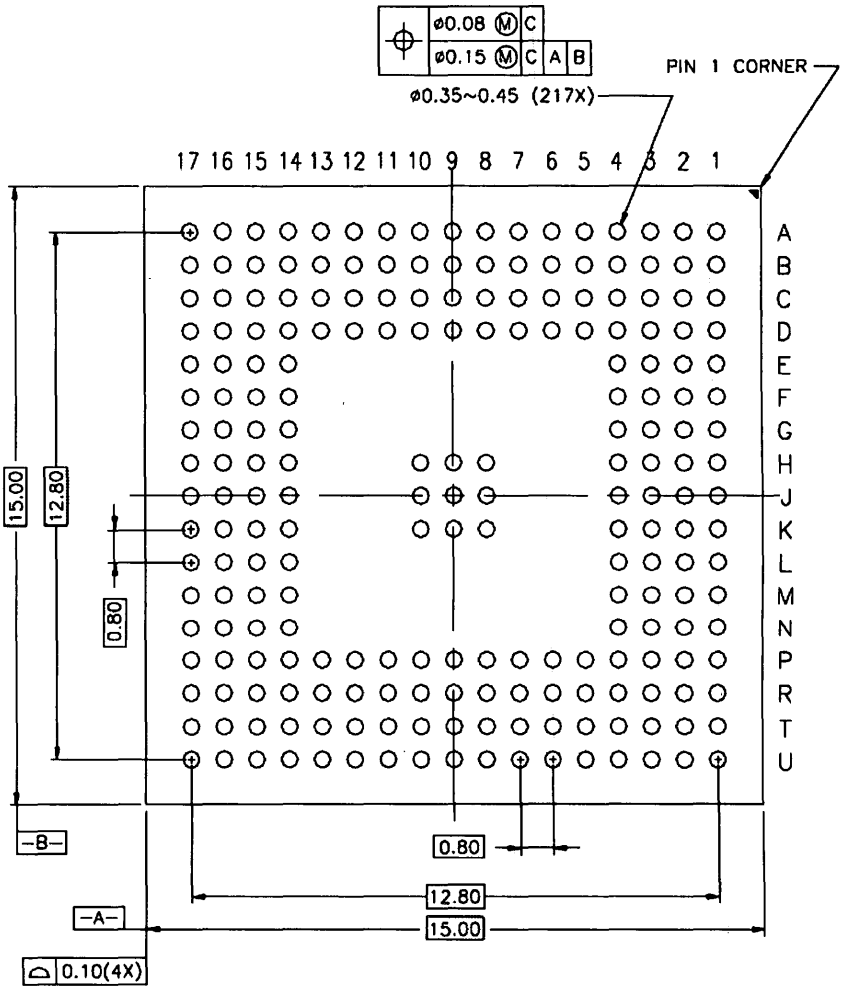

Figure 9 - Bottom view of AT91SAM9G20, dimensions are in mm [27]

The pins are connected using solder pads to the PCB and the pads need to be large enough so that they make contact with the pins. Because the pins on BGA packages are in the shape of balls, the area in contact with the solder pad is smaller than the ball diameter. The pad size is chosen by Altium Designer based on a set of input parameters of the mechanical details of the BGA component and is 13.78mil (0.35mm) in diameter. The chosen fabricator requires a minimum clearance of 5mil (0.127mm) and Figure 10 shows the clearance available with a trace width of 7mil (0.178mm).

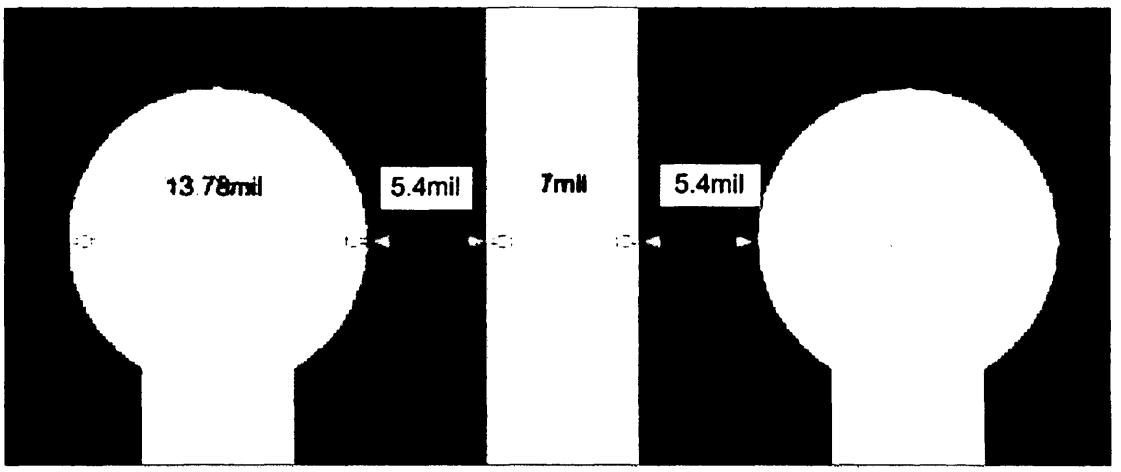

Figure 10 - Close up view of BGA microcontroller solder pads

Using 2oz copper increases the clearance and minimum feature size to the point that a different fabricator will have to perform the copper routing using more expensive techniques.

The present OBC design is composed of 6 layers. The choice of 6 layers is driven by the form factor of the AT91SAM9G20 microcontroller (a BGA component) which has a 0.8mm pitch and is meant for multiple layers to create a fan-out. The board stack up is shown in Figure 11.

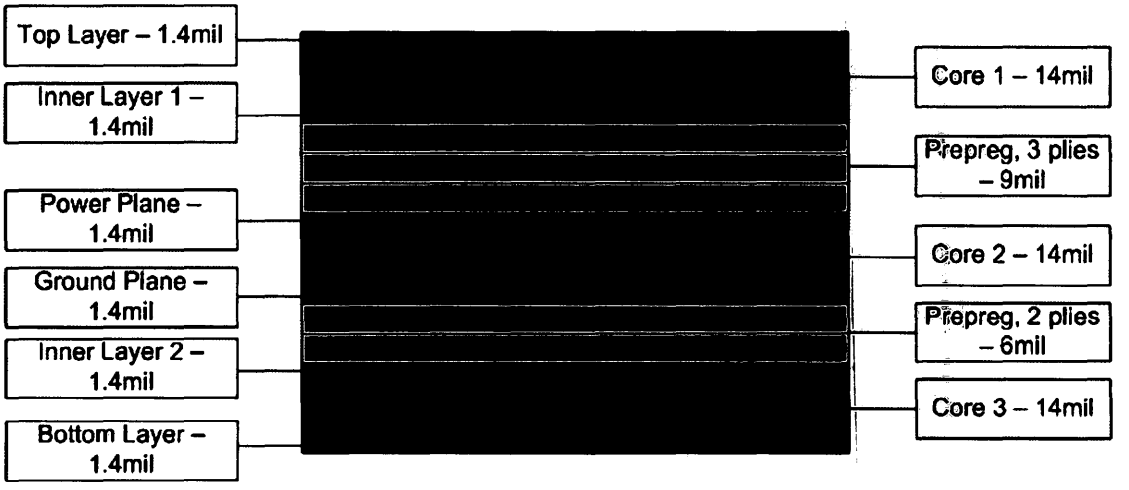

Figure 11 - OBC Board Stackup

The OBC board materials are divided into 3 distinct types: copper, core and prepreg. The copper is the material used for creating conductive traces and planes. The core and prepreg are made from FR-4 material, which is fibreglass bound together with epoxy.

The two inner most layers are the power and ground layers (also known as power plane and ground plane). Power and ground planes allow components to connect their power pins to the planes through the use of vias to keep the distances short. Long traces induce an electromagnetic field which interferes with electronic devices.

The ground and power planes also form a capacitor between them which will stabilize the voltage levels on them by filtering noise which would be seen as an AC component.

The microcontroller fan-out is a critical stage in the design. It is necessary to provide access to all the pins using the shortest traces and least amount of vias possible. Use of the Altium Designer built in auto router provided incomplete results. The pins that were routed included many loops and vias.

The strategy employed for creating the fan-out was to manually route all pins starting with the outside pins. The outside 2 rows of pins were routed out on the top layer as shown in Figure 12. The inner pins were routed onto the 2 inner layers and the bottom layers and connected as shown in Figure 13, Figure 14 and Figure 15 respectively. Ground and power pins are connected to the ground and power planes through the use of vias with their locations being close to the pins.

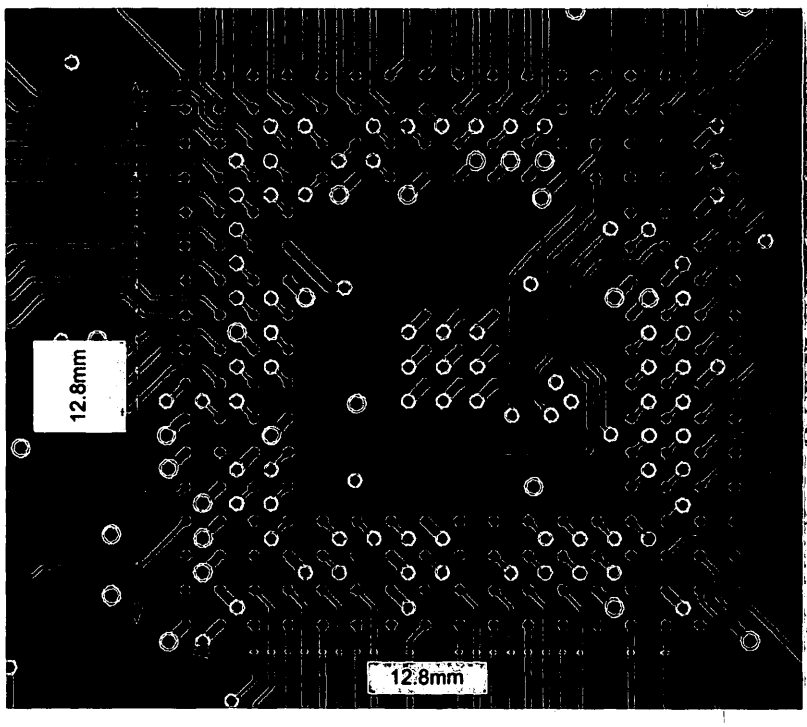

Figure 12 - Top Layer Fan-Out of Microcontroller Connections

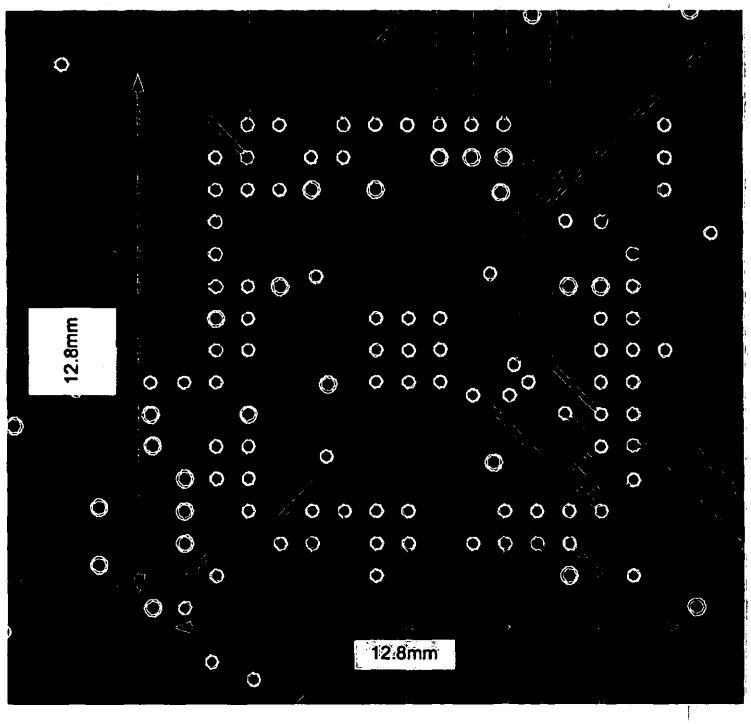

Figure 13 - First Inner Layer Fan-Out of Microcontroller Connections

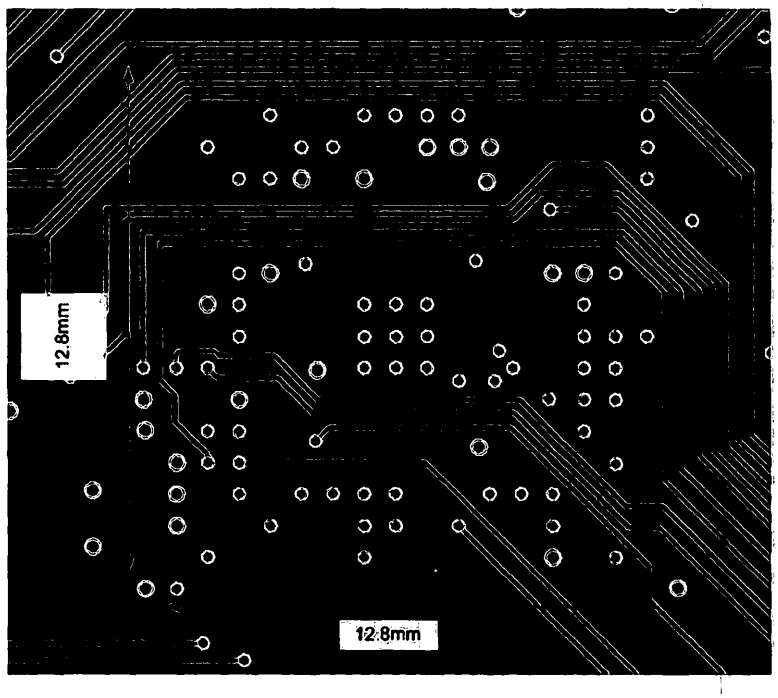

Figure 14 - Second Inner Layer Fan-Out of Microcontroller Connections

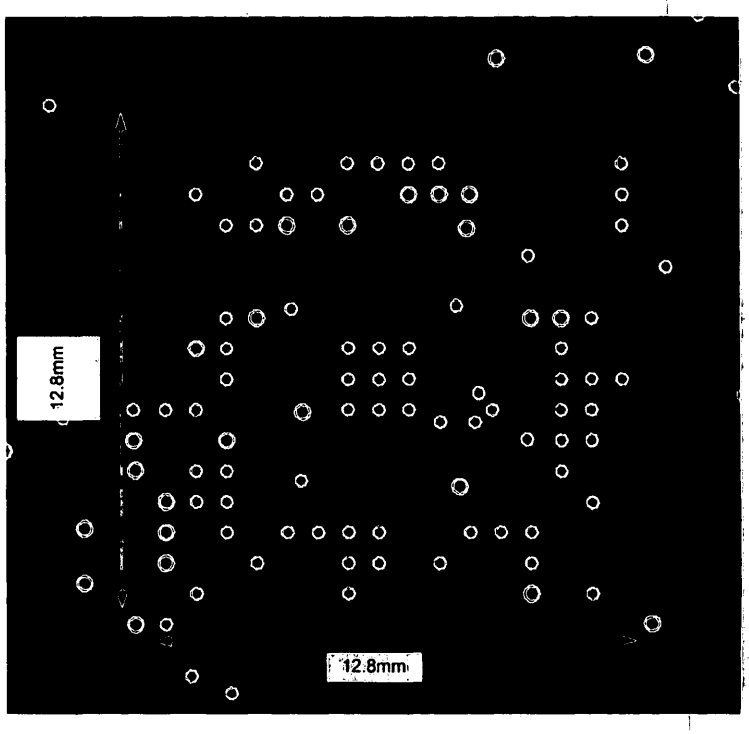

Figure 15 - Bottom Layer Fan-Out of Microcontroller Connections

2.5. Complete Design of the On Board Computer

The layout was created with the microcontroller at its center and the other integrated circuits around it in a way as to make the pins on the components to be close to their respective pins on the microcontroller. Passive components are added close to the circuits for minimum distance in order to reduce noise. Shorter traces contain less resistance and parasitic capacitance which improves the quality of the signal.

Debugging features such as the JTAG and DEBUG interface are broken out. They can be used to load software onto the board. The final layout with all signal layers visible is shown in Figure 16.

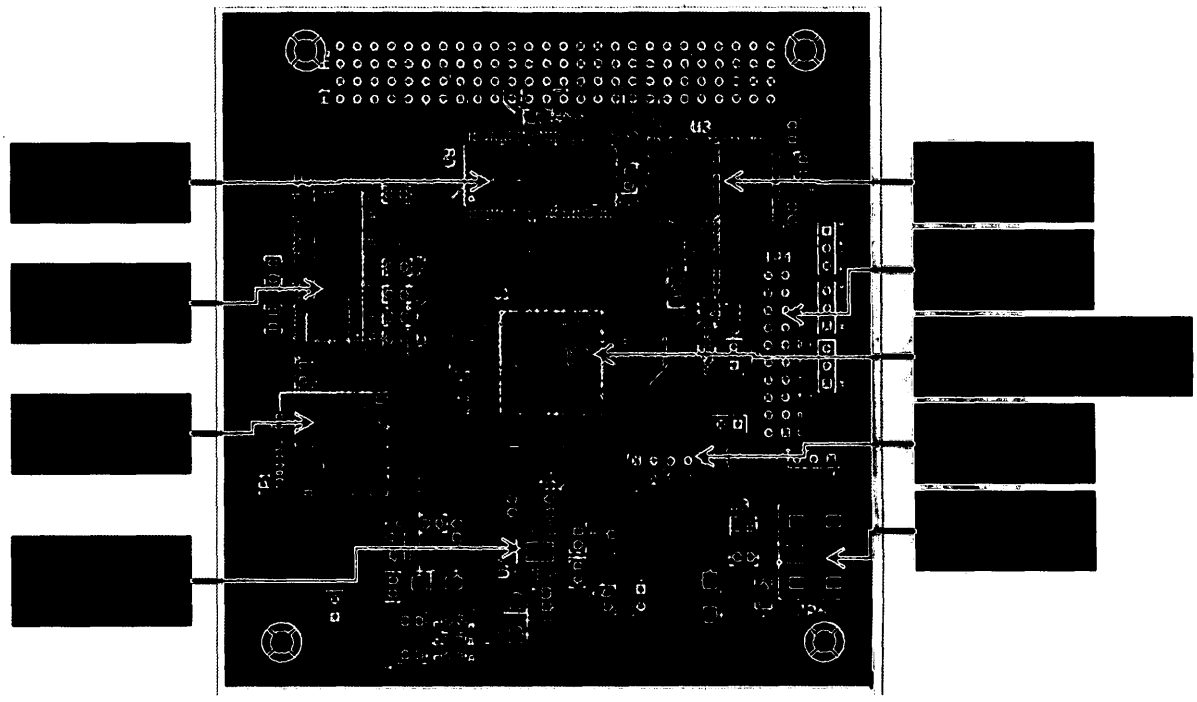

Figure 16 - Final Layout of OBC

2.6. Cost Analysis of Present OBC Design Hardware

In this section the cost of constructing the nanosatellite OBC hardware is examined in detail. When manufacturing a PCB, a manufacturer uses standard sizes of PCB material on which several boards may fit. Therefore using a whole sheet for the manufacturing of a small PCB is inefficient. For prototyping the present OBC design 7 boards are produced at a cost of approximately $500 in February 2012 at Crimp Circuits, a local PCB fabrication company. This brings the approximate cost of 1 board to $71.

In evaluating the component cost, note that many of the resistors and capacitors that are needed for the present PCB design are ordered in bulk quantities and their individual

prices are negligible while other components such as the varying integrated circuits cost a significant amount. The breakdown of the component costs is shown in Table 9; the cost of miscellaneous components such as resistors and capacitors was calculated to be under $5.00. The optional column lists components that can be omitted from assembly and involve the slot for a microSD card, a USB connector, LEDs that indicate power is on and the MRAM. The MRAM is listed as optional due to its large price tag and the fact that it may be omitted from the design as a technology demonstration experiment.

Table 9: Cost Breakdown of OBC Components[1]

Function	Part Number	Amount	Cost	Total Cost
Microcontroller	AT91SAM9G20B-CU	1	$16.32	$16.32
Flash Memory	MT29F2G08ABAEAWP-IT:E TR	1	$6.92	$6.92
SDRAM	MT48LC16M16A2P-75 IT:D TR	1	$8.39	$8.39
Power Regulator	LTC3521EFE#PBF	1	$7.89	$7.89
Crystal	ABM8-18.432MHZ-B2-T	1	$2.18	$2.18
Crystal	ECS-.327-12.5-17-TR	1	$1.61	$1.61
Tantalum Capacitor	T495C107K010ZTE250	5	$1.61	$8.05
Inductor	1812R-472J	3	$2.05	$6.15
Miscellaneous	varies	varies	-	$5.00
				$62.51
Optional				
MRAM	MR4A16BCYS35	1	$40.07	$40.07
MicroSD Connector	DM3AT-SF-PEJM5	1	$3.17	$3.17
EMI Filter	USBUF02W6	1	$1.19	$1.19
USB Connector	1734035-2	1	$2.03	$2.03
LED	LNJ308G8TRA	3	$0.49	$1.47
				$110.44

[1] *All prices in CAN$, obtained from Digi-Key Corporation on September 10th, 2012*

The total price for the components for a full board is therefore approximately $110.

Lastly, the assembly of the components onto the PCB is considered. Assembly is another process where large volume may lead to reduction in unit price if automation is used. The OBC was assembled using a pick-and-place machine that places the components on the board based on their coordinates and orientation on the board. The majority of the cost is associated with setting up of the machine and of loading it with the required surface mount components. After components are placed by the machine, the PCB is baked inside a reflow oven in order to melt the solder paste and bond the components.

A single prototype of the present OBC design was assembled for $450 in March 2012. Hand soldering was briefly considered, but it was noted that while it reduces the cost, hand soldering may also introduce an issue where areas of the PCB as well as components are subject to repeated heating and cooling as each component is soldered separately. This may lead to physical problems in the board due to improperly soldered connections.

2.7. Expected Power Consumption

Power consumption is an important parameter for embedded systems as they are often run on batteries and it is desired for the system to consume minimum power in order to prolong up-time between charges. It is even more important for space equipment as charging is more restrictive, being usually dependent on solar panel size and efficiency, battery capacity and exposure to the sun.

The power consumption of the OBC is estimated by examining the components that require the most power (which on the OBC are the memories and the microcontroller),

and estimating the power consumption from their datasheets. The results are shown in Table 10. While it may seem that the resulting power consumption is very high when all components are active, it is important to realize that the memory devices can only be accessed one at a time and thus only one memory module can be active at a time. The memory component that consumes the most power is the SDRAM due to its need for constant refreshing of the memory cells.

Table 10: Theoretical Power Consumption of OBC Components

	State	Power Consumption (mW)
Microcontroller	Active	165
	Idle	66
NAND Flash	Read/Write/Erase	115.5
	Idle	3.3
SDRAM	Active	445.5
	Standby	132
MRAM	Read	198
	Write	363
	Active Standby	36.3

Chapter 3 – Software

3.1. Trade-Off Study on Types of Software

In order for the present OBC to be able to operate and perform tasks in nanosatellite applications, it requires software that defines and controls the OBC functions.

Software for all applications, including nanosatellites, comes in several variations. A common approach in satellite development is to create custom software for the satellite subsystems. The custom software can be mission specific and may not be utilized on other satellite missions or it can be generic and provide an interface to the hardware on top of which developers can create customized applications. Such an approach is implemented by UTIAS/SFL for the CanX satellites and their GNB platform through the use of Canadian Advanced Nanosatellite Operating Environment (CANOE) [14][11]. The software developed is a real time operating environment which provides the fundamental infrastructure on top of which developers can develop mission specific applications. The operating environment is used to handle the hardware functionality of the system and provide an interface which masks the detailed operation of the hardware components.

A second approach is to use commercial software such as VxWorks, developed by Wind River Systems. VxWorks is a proprietary operating system designed for embedded systems with flight heritage. VxWorks has been flown on several space missions including the Mars Reconnaissance Orbiter [50]. VxWorks is similar to CANOE in the sense that it provides an interface to developers to allow development of mission specific

applications. The benefit of using a commercial system is the reduction in development time and complexity as the commercial package is interfaced with the hardware by the supplier. The development team is left with developing applications that use the commercial software's application programming interface (API).

An alternative to the proprietary commercial software is open source software. The source code of open source software is available for public use free of charge and can be modified. Open source software can also be commercial software, discussion about the distinction between commercial and non commercial open source software is beyond the scope of the research.

Open source software can be seen as a hybrid approach to custom and proprietary commercial software. The open source software can be modified by nanosatellite developers to operate on the nanosatellite hardware and it reduces the amount of custom development that is required.

For the present OBC design, the open source software solution was chosen. The availability of mature software with varied functionality reduces the development cycle to modifying the software to operate on the present OBC hardware. Nanosatellite developers using the present OBC design have access to the full source code of the software and can adapt it in any way that may be needed.

3.2. Trade-Off Study of Open Source Software

Several choices of open source software exist that can be used on the present OBC design. The software packages examined are eCos, FreeRTOS and Linux.

eCos and FreeRTOS are both real time operating systems designed for use in embedded computers. Linux is not a real time operating system by default, additional configuration using either the PREEMPT_RT kernel patch or with the use of an additional real time framework such as Xenomai add real time capabilities to the system.

Linux allows execution of binary files while it is running without modification of the kernel. eCos and FreeRTOS allow execution of tasks that were included into the system during compilation, addition of new features requires the recompilation of the whole system into a new binary file that must be written into the computer system.

The Linux kernel functionality comes at the price of higher memory cost. The present OBC design is a system that is able to provide both the storage and the run time memory for the Linux system.

The Linux kernel was chosen to be used on the present OBC design as it is extensively used in many applications both embedded and non-embedded.

3.3. OBC Boot Process

The present OBC design begins the boot process when power is applied to the board. The microcontroller contains a boot program within its Read Only Memory (ROM). When the

microcontroller is powered on, the ROM boot program begins searching for a valid program. A valid program for the AT91SAM9G20 is a binary file which contains 7 sets of data, each 4 bytes in size, in the beginning of its memory locations. The 7 data sets are known as ARM exception vectors and they are compared to patterns accepted by the microcontroller, they are also used to supply information about the program itself. From the datasheet for the AT91SAM9G20; the boot process is shown in Figure 17. When the microcontroller fails to find a valid program within a defined time limit, the ROM boot program moves on to the next interface. After the boot program checks the last interface and is unable to find a valid program, it executes the SAM-BA interface. SAM-BA is a software that is used on a computer to detect and connect to an Atmel microcontroller. SAM-BA is used in this research to load software onto the empty OBC prototype.

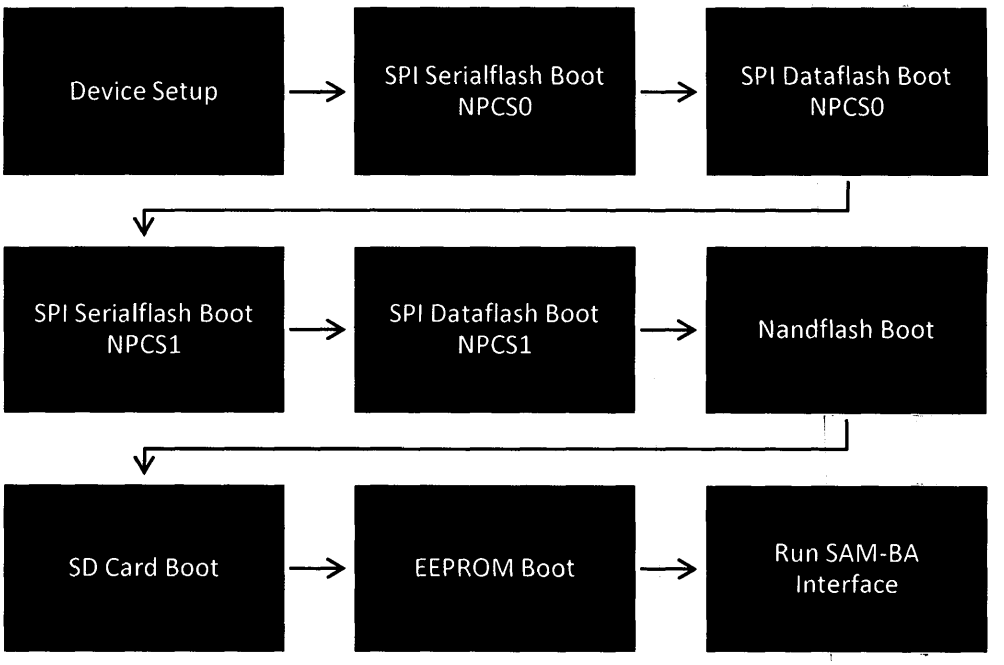

Figure 17 - Microcontroller Boot Process. Adapted from datasheet [27]

The software on the present OBC design is stored within a NAND Flash and as such, the boot program will find the initial bootloader at the beginning of the NAND Flash memory and copies it to the internal SRAM memory to begin its execution. The internal SRAM is part of the microcontroller circuitry and on the AT91SAM9G20 is 16KiB in size and therefore the initial bootloader program must be small enough to fit into that area. When the microcontroller has found a bootable program, that program can be used to configure connections further to provide other functionality.

The boot process begins by executing a built in program on the microcontroller that cannot be changed. The microcontroller looks for a bootable program in external memory by cycling through a predetermined set of peripherals interfaces.

In the case of the OBC, the boot program is located in NAND Flash and the execution is demonstrated in Figure 18.

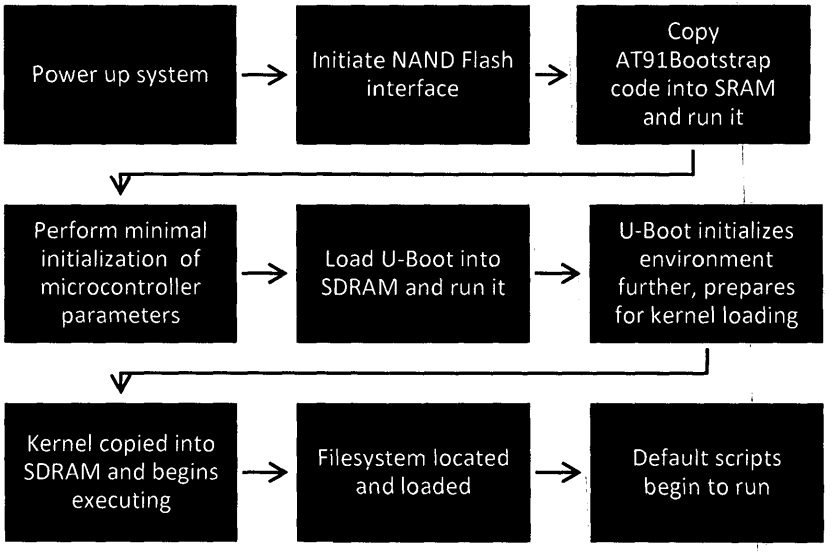

Figure 18 - Boot Process of Present OBC Design

3.4. Required Software Components

The necessary software needed in order to have an operational board is composed of several main packages. The first important step is to set up a cross compiler environment in order to create and modify software for the present OBC design. In order for a program to execute on a computer platform, the computer must understand how the program operates and what it needs. Programs written in a programming language must be converted to a language that the computer understands known as machine language [51]. Machine language is not easy for humans to understand and work with, which is why higher level languages such as C, JAVA and FORTRAN were developed. Machine languages are not universal and varying architectures have their own requirements as to how they require a program to operate. Software that is compiled for a PowerPC architecture will not be able to run on an ARM architecture because the system does not understand the program.

Cross-compilation is a method of compiling software for a computer architecture that is different than the one on which a compiler is running. The cross compilers that are used for software development for the present OBC design are the Embedded Linux Development Kit 4.2 (ELDK 4.2) from DENX Software Engineering and CodeSourcery from Mentor Graphics. The need for 2 cross-compilers arose due to AT91Bootstrap not being able to compile with the CodeSourcery compiler version used.

The first level bootloader is the program that is executed when the microcontroller is powered on. The first level bootloader is used to perform initialization of the board such

as configuring the peripherals. The software package that was chosen to act as a first stage bootloader is the AT91Bootstrap from Atmel Corporation. This open source software is designed to be used with their products and the existing code allows easy modifications in order to customize the package for a different board.

The AT91Bootstrap code for the AT91SAM9G20-EK was modified in order to accommodate the present OBC design. The majority of modifications were configuration changes as both boards use the same microcontroller and the interfaces to the SDRAM and Flash memory use standard interfaces. An MRAM driver was added to AT91Bootstrap in order to able to test basic read and write capabilities of the device.

The second level bootloader is Das U-Boot, often called simply U-Boot. U-Boot is developed by Denx Software Engineering and is open source software. It is a popular choice for embedded systems due to its wide support of a variety of architectures, including ARM, as well as a wide selection of code that is designed to work with commercial boards. U-Boot is a popular piece of software and its use as part of the present OBC design is due to large amount of support available and its continuous active development [52].

3.5. JTAG Interface

The Joint Test Action Group (JTAG) interface is a standardized interface that is used to test printed circuit boards and the components on them [53]. Devices that support JTAG contain internal circuitry that allows a JTAG apparatus to have control of the circuitry in the device. It is a powerful tool that simplifies the process of testing circuitry. The JTAG

interface is used on the present OBC design to test the connectivity of pins by modifying their state and measuring the output.

The JTAG interface combined with SAM-BA is the preferred method of loading software onto the present OBC design. SAM-BA is a program made by Atmel that is designed to communicate with their products. It is capable of initializing the NAND Flash and SDRAM memories and simplifies the task of writing software to NAND Flash.

3.6. MRAM Driver

The NAND Flash and SDRAM make use of the microcontroller's built in interfaces that allows them to be initialized and then accessed using memory addressing. Both the NAND Flash and SDRAM have their own dedicated range of memory to which data could be written and read from by directly accessing the memory location. The microcontroller itself regulates how the control lines on each device are used based on the user configuration.

The MRAM driver makes extensive use of the microcontroller external bus interface as well and it is connected to the static memory controller with additional general purpose pins to provide full control of the device. The use of the general purpose pins slows down the operation of the MRAM as they are a part of a separate circuit and not part of the memory circuit. As the MRAM is experimental hardware that is designed to be used as a backup memory, its decreased performance is deemed acceptable.

The MRAM interface requires the microcontroller to be configured for timing and the use of the correct memory lines.

3.7. XENOMAI Real Time Framework

Xenomai is a software package that is designed to provide real time functionality to a Linux based system. Xenomai works by embedding into the Linux system and providing an interface to the computer hardware separately from the Linux interface. Linux tasks contain no real time requirements while Xenomai tasks do.

A secondary option to Xenomai is to apply the PREEMPT_RT patch to the Linux kernel. The Linux kernel possesses a pre-emptive scheduler with limited capabilities and it is incapable of hard real time. Only select few processes are able to pre-empt other processes and only under certain circumstances. The PREEMPT_RT patch for Linux modifies the Linux kernel by making all processes pre-emptible. A pre-emptive task can be interrupted by another task and its execution is suspended until the task that interrupted it releases the resources back. Configuring which tasks can interrupt other tasks involves assigning a level of privilege to a task where more privileged tasks can interrupt lower privilege tasks.

Both Xenomai and the PREEMPT_RT patch require the use of their own interface in order to create real time tasks. It is not required for every task to be a real time task. Tasks that do not require real time execution should not be executed as real time tasks as they could reduce the execution time of tasks that do require real time response.

In a study that examined the performance of the unmodified Linux kernel, Linux kernel with the PREEMPT_RT patch and Xenomai it was found that Xenomai has on average better performance in a variety of tasks [54] though the PREEMPT_RT has performed comparably well in most situations.

To implement Xenomai on the present OBC design it is required to apply the Xenomai patch to the Linux kernel. Once the patch is applied, the kernel is compiled with Xenomai enabled.

3.8. OBC Filesystem

The filesystem is a means to organize the computer memory contents. The filesystem contains libraries and programs and data is stored in the filesystem and can be viewed as a high level interface to memory contents.

Several types of filesystems exist. One of the main requirements for the present OBC design is a capability to store new data within its non volatile memory. This requirement excludes several types of filesystem formats such as CRAMFS which are loaded into RAM and do not support the ability to permanently store new data.

The fileystem formats that are examined for the present OBC design are suited for Flash and are: Journaling Flash File System version 2 (JFFS2), Yet Another Flash Filesystem version 2 (YAFFS2) and Unsorted Block Image FileSystem (UBIFS). All 3 types are filesystem that are capable of storing new data in Flash memory for later use, as such, the data is persistent and remains in memory when power to the system is lost. Flash devices

such as SD cards and USB drives contain a built in Flash translation layer that presents the device as a hard drive when connected to a computer. The filesystems above do not use a Flash translation layer and instead organize access the memory through the use of blocks and pages.

JFFS2 keeps a log file that contains the information about all data stored in it, the entries in the log are known as nodes. Flash devices suffer from wear when they are written to. JFFS2 attempts to minimize the wear by organizing where data is written to in an effort to even out the use of memory [52]. JFFS2 works with erase block sizes. Nodes must be smaller than the erase block size as JFFS2 does not allow nodes to be spread out over several blocks. If a block does not contain enough free memory then a new block is selected, this may lead to inefficient use of space as blocks may contain amount of free space that may be too small to contain a whole node individually but together may accumulate to a large amount of space.

YAFFS2 uses page sizes as opposed to erase block sizes. Page sizes are smaller than erase blocks (exact numbers vary between Flash devices) and allow smaller files to be written more efficiently. In addition, YAFFS2 uses less RAM compared to JFFS2 to perform its operations [55].

UBIFS works on top of Unsorted Block Images (UBI). The UBI layer divides data being written to memory across the entire Flash memory; as such it provides wear leveling capabilities similar to JFFS2 and YAFFS2 [56]. UBI is an abstraction layer that separates

the UBIFS from the raw Flash. Access UBIFS to the raw Flash is done through virtual erase blocks that are mapped to physical erase blocks, this is managed by UBI.

Studies on the performance of the filesystems conclude that UBIFS is superior to JFFS2 and YAFFS2 as it provides faster read and write time, faster mount times [57]. There is overhead associated with the UBI inclusion though that becomes negligible for larger Flash sizes, including the one being used on the present OBC design.

3.9. Triple Modular Redundancy

Triple Modular Redundancy (TMR) is an error correction and detection scheme that is employed in satellites to correct the effects of radiation on memory devices. TMR works by comparing 3 identical copies of data in memory and detecting any differences. Differences in the data can be corrected if 2 of the memory addresses contain identical data. The third memory address is corrected with the data stored in the other 2 addresses.

TMR schemes are traditionally implemented using an FPGA as a voter. The computer system thinks that it accesses one memory location when in reality the voter accesses 3 memory locations where identical data is stored on its behalf. Comparisons are done within the FPGA and the result is passed to the system.

The present OBC design does not contain an FPGA component and a method to perform a form of TMR in software is developed.

TMR can be applied to the present OBC design. Storing multiple copies of critical software at known memory locations and allowing a run-time process to perform TMR on them will provide protection from radiation effects during run-time.

The voter technique was developed with the help of truth tables of potential situations that can arise. The problem simplifies when it is understood that the comparison is performed on bit level. The logic equation that was developed to perform the voting is seen in equation (1).

$$Result = \neg[(a \oplus b) \wedge (a \oplus c)] \oplus (\neg a) \qquad (1)$$

Where a, b and c are the 3 memory arrays to be compared and the result is the corrected array.

The AT91Bootstrap performs initialization of several interfaces including the NAND Flash and SDRAM memory, it is desired that the voter checks for inconsistencies in the data and corrects them before initialization. A problem arises that without initializing the NAND Flash, obtaining the other 2 copies of AT91Bootstrap is not guaranteed. The ROM bootloader remaps the memory and the NAND Flash interface may no longer be properly initialized.

A solution is developed that forces the ROM bootloader to load all 3 copies of the AT91Bootstrap into microcontroller SRAM and a comparison can be accomplished within SRAM. Limitations of this technique involve the limited memory available within

SRAM as well as the necessary modification of the binary structure of the AT91Bootstrap is necessary.

The SRAM memory bank that can be accessed during initial boot is 16KiB in size and the 3 copies of AT91Bootstrap must be smaller than it to fit within. The AT91Bootstrap is compact software though the addition of the voter code causes the memory size to approach the limit. To conserve memory space, the voter as implemented on the current OBC design scans through all of the memory at a rate of 32 bits at a time for each copy and overwrites the data in SRAM. Addition of extra checks that only correct the incorrect bits add too much code and the size becomes too large for the SRAM.

To make SAM-BA load 3 copies of AT91Bootstrap the binary file is opened in a hex editor and the size vector is modified. Note that the binary file is in little Endian format and thus the most significant byte is on the right of each set of 4 bytes. Figure 19 shows the unmodified binary code of an AT91Bootstrap file. The size is shown as 0×0020149C, this is inaccurate, the compiler appends to the size vector, and the real size of the file in the example is 0×0000149C, this will be discussed in the next section.

```
00000000 0E 00 FF EA 05 00 00 EA 05 00 00 EA 05 00 00 EA 05 00 00 EA
00000014 9C 14 20 00 05 00 00 EA 05 00 00 EA FE FF FF EA FE FF FF EA
00000028 FE FF FF EA FE FF FF EA FE FF FF EA FE FF FF EA FE FF FF EA
0000003C C1 D9 A0 E3 D0 00 9F E5 00 10 90 E5 01 20 A0 E3 02 10 11 E0
00000050 09 00 00 1A C0 00 9F E5 01 19 A0 E3 01 20 A0 E3 02 10 81 E1
00000064 00 10 80 E5 A8 00 9F E5 00 10 90 E5 01 20 A0 E3 02 10 11 E0
00000078 FB FF FF 0A 9C 00 9F E5 03 10 A0 E3 00 20 90 E5 01 20 02 E0
0000008C 00 10 A0 E3 02 00 51 E1 0E 00 00 1A 01 10 A0 E3 00 20 A0 E3
000000A0 02 10 81 E1 00 10 80 E5 68 00 9F E5 00 10 90 E5 08 20 A0 E3
```

Figure 19 - Unmodified AT91Bootstrap Binary Data

The binary code must be modified to a value 3 times of the actual size to force the ROM bootloader to load all 3 copies into SRAM. The value is changed manually using a hex editor, not that the "20" entry in the size vector must be removed at the same time. The new size vector for the above example is 0×3DD4 as is shown in Figure 20.

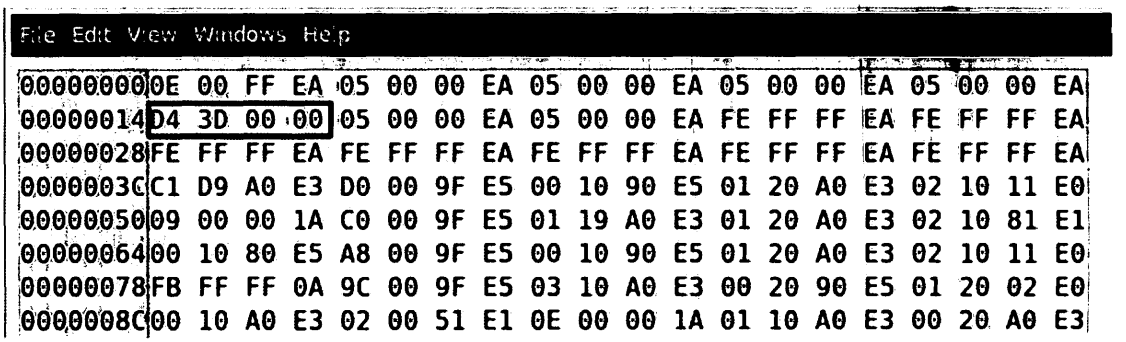

Figure 20 - Modified AT91Bootstrap Binary Data

The extra "20" that is originally in the size vector does not cause problems when loaded using SAM-BA because SAM-BA rewrites the vector on its own using the actual file size. Performing the above changes to the AT91Bootstrap file and loading through SAM-BA will overwrite the size vector and microcontroller will only load the first copy of AT91Bootstrap.

SAM-BA itself must be modified and the code in SAM-BA that modifies the size vector is located in the file "SAM-BA install directory/tcl_lib/common/generic.tcl". The line that overwrites the size vector and must be commented out is:

```
# If this is a boot file modify 6th vector with file size
if {[expr ($isBootFile == 1) && ($offset == 0)]} {
TCL_Write_Int $target(handle) $sizeToWrite [expr $bufferAddress + (5 * 4)]
dummy_err
```

Commenting out the SAM-BA size rewrite line and modifying the AT91Bootstrap binary file size vector causes the ROM bootloader to load 3 copies of AT91Bootstrap into the internal SRAM.

Chapter 4 – Assembled System Performance Evaluation

4.1. Power Consumption of OBC

The actual power consumption of the OBC is tested by connecting the OBC to a power supply and monitoring the voltage level and current drawn. The results for various operations of the system are shown in Table 11. It is shown that the power consumption is high during memory operations and it is therefore important to minimize those activities.

Table 11: Measured Power Consumption of OBC

	State	Power Consumption (mW)
System	Idle	570
System and NAND Flash	Write	792
	Read	710
	Erase	710
System and SDRAM	Write	825
	Read	726

4.2. General Performance Evaluation

The performance of the present OBC design is evaluated by comparing execution of mathematical functions between the OBC prototype and other embedded systems. Code

in the C language was written to use a 4th order Runge-Kutta method to numerically solve the equation for a mass spring system which is shown in equation (2).

$$x''(t) = -\frac{k}{m}x(t) \tag{2}$$

Where the variables in equation (2) are:

$x''(t) -$ *the acceleration of the mass over time, units of* $\frac{m}{s^2}$

$k -$ *The stiffness of the spring, set to* $1.0\frac{N}{m}$

$m -$ *the mass attached to the spring, set to* $1.0kg$

$x(t) -$ *the position of the mass with respect to time, units of* m

The initial conditions of the simulation were set to the following:

$x_0 = 1.0m$

$t_0 = 0s$

$x'(0) = 1.0\frac{m}{s}$

The code was set to compute the results up to 20 seconds with 500,000 time steps. The code was executed on 4 systems: present OBC design, Linuxstamp II-AT91SAM9260 version, Linuxstamp II-AT91SAM9G20 version and the TS7800 by Technologic Systems. The TS7800 system is an embedded system with a PC/104 form factor with an ARM9 processor. The Linuxstamp systems are chosen as the present OBC design is based on the Linuxstamp designs. The TS7800 system is chosen to represent a typical commercial embedded computer.

The code is executed with the built in "time" command. This command measures the execution time of a program and presents 3 time values: real, user and system. The real value is the total amount of time that the program has taken to execute from issuing the command until the program has completed execution. The user value is the total amount of time that the program spent executing in user space. System value is the total time that the program spent executing in kernel space [58]. Processor usage (%CPU) is computed as the total time in user and system mode together as a percentage of the total real time. When a program is executing, it must share the access to the processor with other processes. In the case of each system, the only processes that were running at the same time as the computations were the necessary processes for keeping the operating system running.

The code was executed on each platform 5 times and the results were averaged. The results for the time and power measurements of the software execution on each of the platforms are shown in Table 12. Power consumption of each system is measured throughout the computation process and is averaged.

Table 12: Benchmark Tests of Embedded Systems

	Real (s)	User (s)	System (s)	CPU (%)	Power (W)
OBC	24.41	20.74	0.29	86%	0.79
LS9G20	23.71	19.75	0.43	85%	0.60
LS9260	72.33	65.18	1.19	92%	0.72
TS7800	77.91	24.92	49.16	95%	4.13

The TS7800 has significantly higher power consumption than the other systems. It is a commercial system that contains additional components, including an FPGA, each requiring additional power even while idle. The Code executed slowly on the TS7800 and LS9260 while the LS9G20 and present OBC design had similar execution time.

Noticeably higher power consumption is apparent on the present OBC design in comparison to the LS9G20. Part of the higher power consumption can be attributed to the MRAM and the voltage regulator used. The present OBC design contains a voltage regulator with 3 outputs where 2 of the outputs are used to power all components on board. The LS9G20 contains a voltage regulator with 2 outputs; no 5V output option is given.

Communication interfaces were tested by connecting devices that require the equivalent communication interfaces. The following list outlines how the interfaces were tested.

The Universal Asynchronous Receiver/Transmitter (UART) interface was tested extensively as it is used for serial communication with a computer. The UART interface was tested with the C328-7640 JPEG camera module by CoMedia Limited. The OBC uses the UART interface to configure the camera and obtain images that are saved into the on-board NAND Flash.

The Two Wire Interface (TWI) interface was tested by communicating with 2 separate devices. The first device is the HMC5883L magnetometer which successfully returned data which varied with an applied magnetic field. The second device used to test the TWI was a motor assembly with a controller developed by Dr. Krishna Kumar's group at

Ryerson University. The motor controller receives command through the TWI interface and uses them to set motor speed. The present OBC design has successfully communicated with the device and was able to configure the motor speed.

General Purpose Input/Output (GPIO) pins are tested by toggling the pin state using software and monitoring the pin using an oscilloscope. GPIO pins successfully responded to software manipulation of their state.

The MRAM has successfully been written to and read from. Testing of the MRAM operation is performed using following methods:

- Write data to MRAM and read it back immediately
- Write data to MRAM, wait 10 minutes before reading data back
- Write data to MRAM, shut off system and wait 10 minutes, turn system on and read MRAM
- Write non identical data to MRAM in separate memory addresses and read them back in the order they were written

The MRAM was successful in performing the above tasks and is operating as expected.

4.3. Thermal Cycling Test

Prior to performing a thermal vacuum test, the present OBC design is subjected to a thermal cycling test. The purpose of the thermal cycling test is to perform a general evaluation of the system under extreme temperatures before it is subject to vacuum conditions. The OBC was placed into an anti-static bag in order to minimize

condensation during the cold stages of the temperature profile. Figure 21 shows the present OBC design after it is placed within the thermal chamber.

Figure 21 - Present OBC Design in Thermal Chamber

The temperature profile and test stages that the present OBC design was subject to during the thermal cycling test is:

- Ambient temperature – perform test of equipment
- -30°C – perform test of equipment
- -40°C – leave all equipment off
- -30°C – perform test of equipment
- 60°C – perform test of equipment
- 70°C – leave all equipment off
- 60°C – perform test of equipment
- -30°C – perform test of equipment
- 60°C – perform test of equipment
- Ambient temperature – perform test of equipment

The tests that are performed on the present OBC design are as follows:

- Apply power to OBC and see whether it turns on and boots successfully

- Execute a program to connect to the camera and store the resulting picture with a time stamp
- Execute code that connects to a motor outside of the thermal chamber using the two wire interface (TWI) and run it.
- Reset OBC and boot into U-Boot and perform a memory test where the microcontroller writes and immediately reads from the SDRAM for a prolonged period of time

The present OBC design successfully accomplished the tests outlined above at every stage. The temperature profile at a hot and cold stage are shown in Figure 22 and Figure 23 respectively.

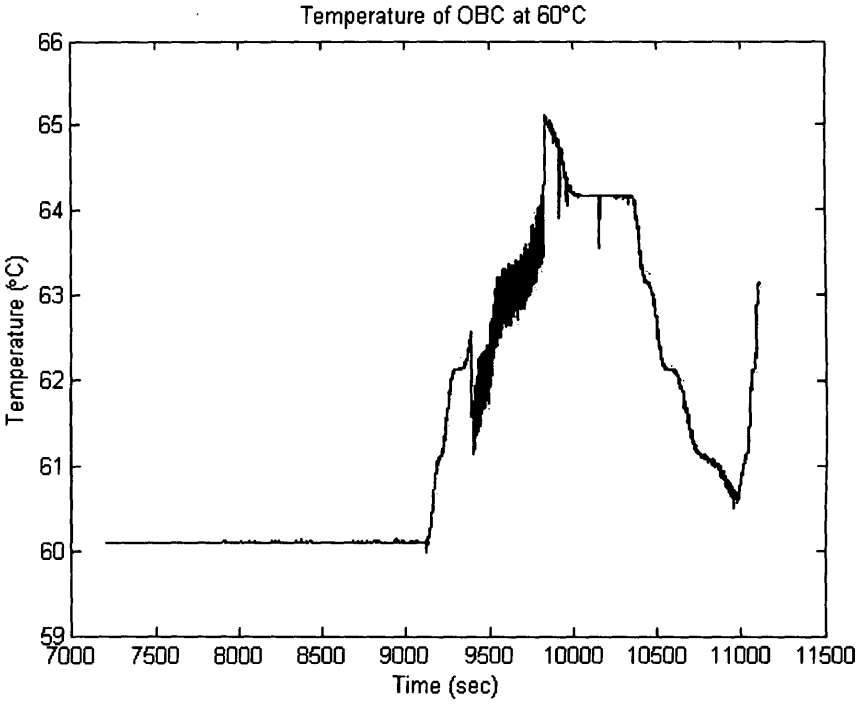

Figure 22 - OBC Operational Temperature at 60°C

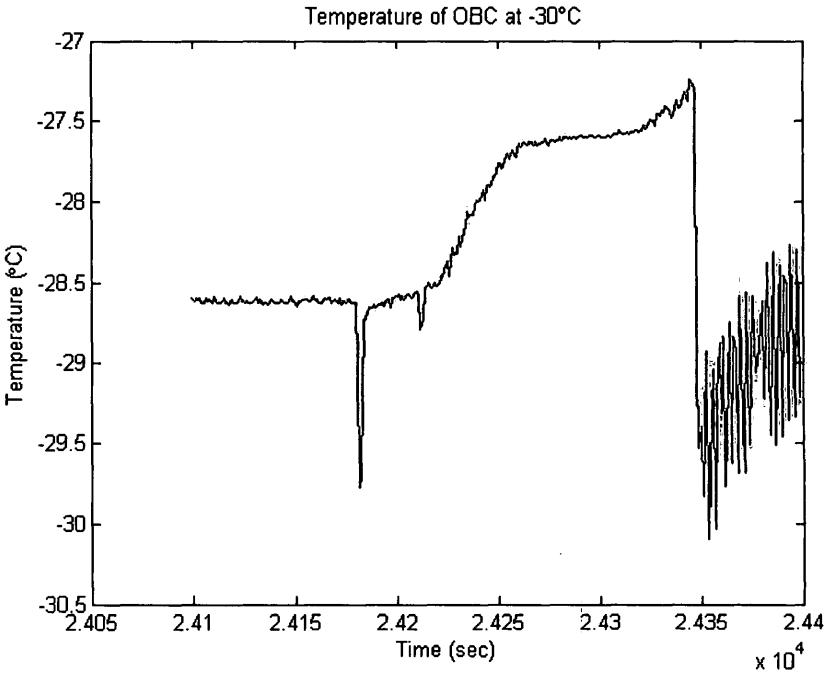

Figure 23 - OBC Operational Temperature at -30°C

The resulting temperature data contains significant amount of noise and it is particularly apparent in Figure 23 where the temperature drops suddenly. The noise is due to the temperature sensors being not well isolated from the OBC. A change in the state of the OBC, such as when transmitting data over the UART line, interferes with the data from the temperature sensors. The resulting temperature of the OBC is measured right after the OBC completes an operation.

The present OBC design has not failed and has not shown a decrease in performance at the extreme temperature stages. After the success of the thermal cycling test, the OBC was subject to a thermal vacuum test as the next stage of space qualification.

4.4. Thermal Vacuum Test

A thermal vacuum test was performed on the OBC prototype in conjunction with other equipment at the Thermal Vacuum (TVAC) chamber in York University, which is managed by Professor Brendan Quine. The OBC was chosen as the reference subject for temperature control of the equipment in the TVAC chamber and 2 temperature sensors were placed on it, this is done in the case that one sensor becomes detached, there is another one that can be used as a reference. The main sensor is placed on the microcontroller and the backup sensor was placed on the SDRAM. The microcontroller is chosen because it is central part of the system and the SDRAM is chosen due to its potential large power consumption under consistent operation.

4.4.1. Thermocouples

The temperature sensors used are calibrated thermocouples and were attached to the surfaces to be monitored using layers of Kapton and silver tape as recommended by Professor Brendan Quine, the order is shown in Figure 24.

A thermocouple is a device composed of 2 metals connected at the tip; the different thermal properties of the metals will produce a voltage that is related to the temperatures of the joint and the temperatures of the 2 metals at the other end, this phenomenon is known as the Seebeck effect [59].

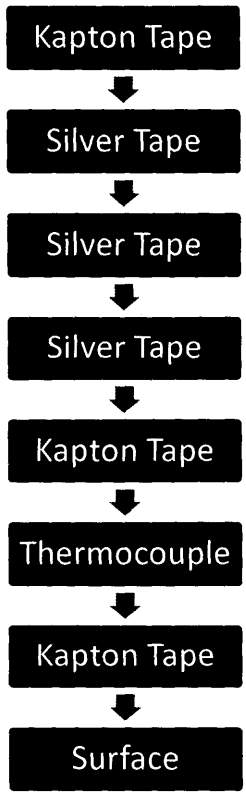

Figure 24 - Layering of Thermocouple Attachment

The Kapton tape is used due to its ability to bond to the surface better than the silver tape and thus it will hold the thermocouple in place with greater force, another desirable property of the Kapton tape is that it is an electrical insulator and it will prevent a short circuit from being formed by either the thermocouple or the silver tape. Kapton is a poor thermal conductor but due to the thin nature of the tape which is 0.005 inches, it does inhibit the capability of the thermocouples to measure the temperature of the equipment they are attached to.

The silver tape is used to transfer the heat around the thermocouple evenly to ensure that the thermocouple joint experiences the same temperature all around it. The placement of the thermocouples on the OBC is shown in Figure 25.

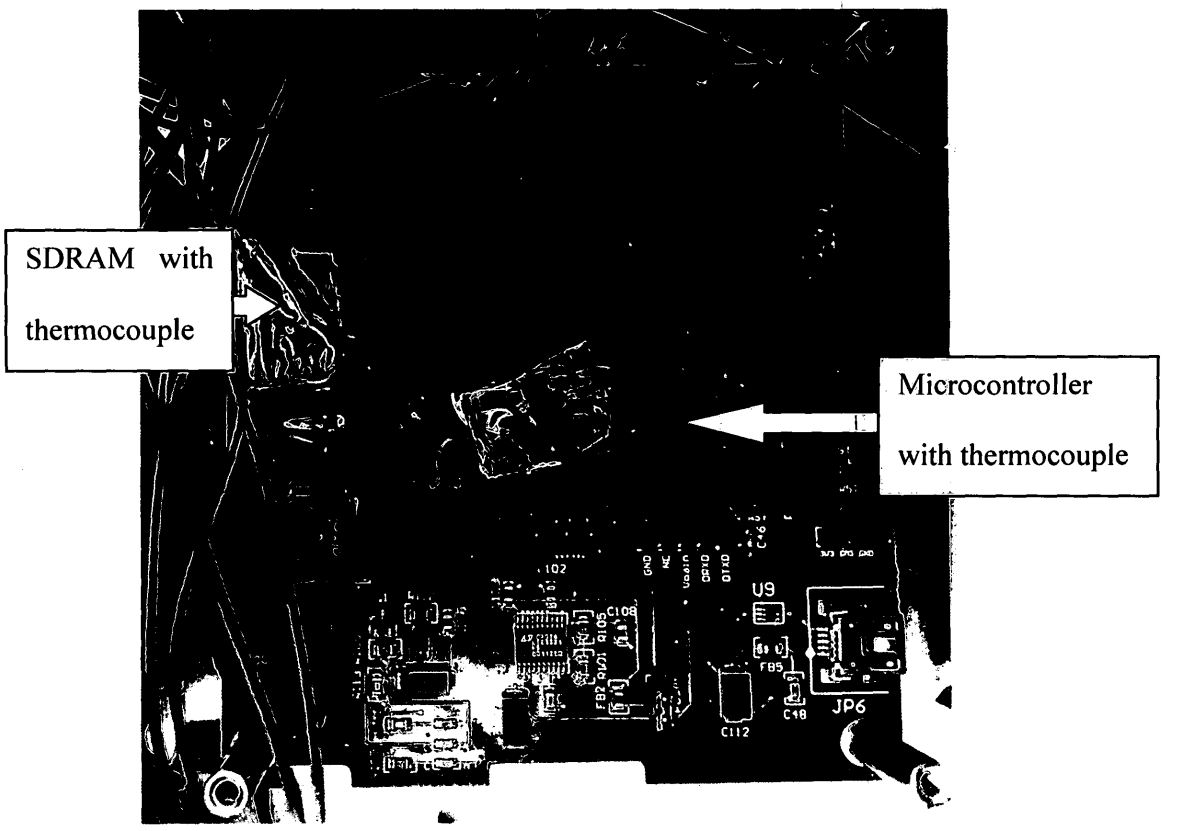

Figure 25 - OBC in TVAC with Thermocouples Attached

4.4.2. Test Setup

The TVAC test was to be performed at a vacuum with a pressure lower than 1E-5Pa and in fact the pressure was maintained between 1E-6Pa and 3E-6Pa throughout the test. The temperature profile that the equipment was to be subject to is identical to the temperature profile that was used in the thermal tests and is as follows:

- Ambient temperature – perform test of equipment
- -30°C – perform test of equipment
- -40°C – leave all equipment off
- -30°C – perform test of equipment
- 60°C – perform test of equipment
- 70°C – leave all equipment off
- 60°C – perform test of equipment
- -30°C – perform test of equipment
- 60°C – perform test of equipment
- Ambient temperature – perform test of equipment

In all temperatures, there was a 3°C allowable error for the target temperature. For the temperatures of 60°C and 70°C the temperature of the OBC could rise above by 3°C but should be maintained above the targets and for the -30°C and -40°C the temperature was allowed to go below by a maximum of 3°C and should be maintained at below the target levels.

The choice of the temperatures was determined by the tightest operational temperatures that were allowed for the components of the OBC. All the components were chosen to be of industrial quality and thus have an extended range; the minimum lowest temperature was common to most components and was -40°C. The maximum high temperature that component some components could be subjected to was 80°C.

The OBC is soaked at each temperature for an hour before performing a test or moving to the next step if no test is to be performed. After each test, the devices are left soaking to allow equipment to return to soaking temperature.

Inside the TVAC chamber the OBC is connected to the C328-7640 camera that is being tested as well as the HMC5883L magnetometer that collects data about the magnetic field being generated by 2 magnetorquers.

The tests that the OBC was subject to involved several different aspects and they were monitored through a serial connection to a computer beside the TVAC chamber and they are as follows:

- Apply power to OBC and see whether it turns on and boots successfully
- Execute a program to connect to the camera and store the resulting picture with a time stamp
- Execute code that connects to a magnetometer through the two wire interface (TWI) and read and store data over a 20 second period while magnetorquers are off
- Execute code that connects to a magnetometer through the TWI and read and store data over a 20 second period while magnetorquers are on
- Reset OBC and boot into U-Boot and perform a memory test where the microcontroller write and immediately reads from the SDRAM for a prolonged period of time

The resultant temperature profile from the entire test is shown in Figure 26. Sharp peaks designate a test point where the equipment is operational and the temperature of the equipment rises. The first -30°C test was skipped due to an overshoot of the temperature and it was decided to proceed to -40°C and continue with the temperature profile from that step. Another overshoot happened while going to the first 60°C step but the rate of change was not as great and it was possible to recover and bring the temperature down to 60°C.

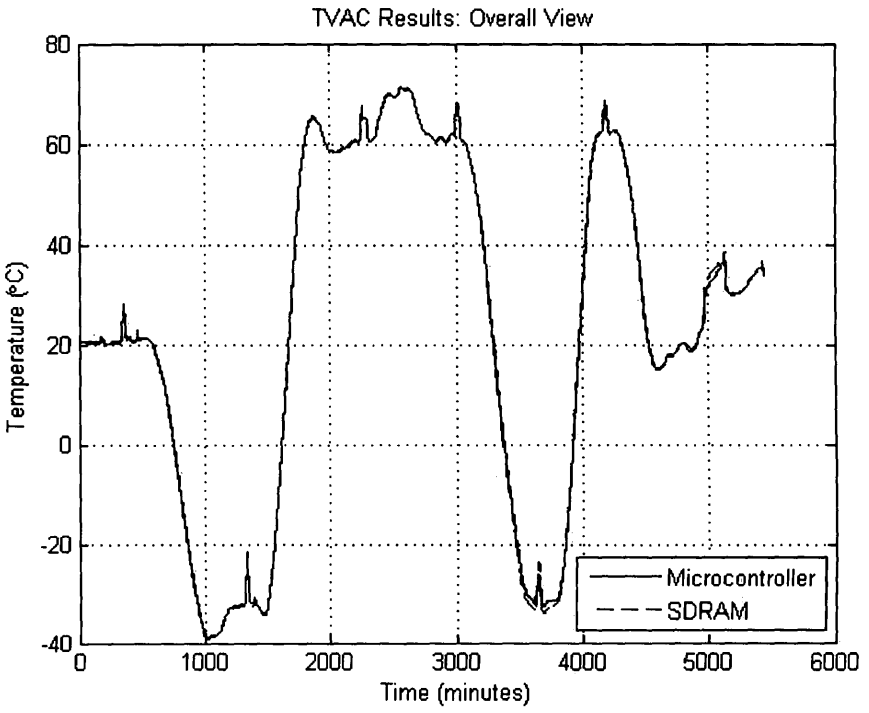

Figure 26 - Temperature of OBC Microcontroller and SDRAM throughout the Test

A close up of the second -30°C test is shown in Figure 27. At this temperature in both instances the camera did not respond. The interface to the camera is a serial interface like the one used to connect the computer to the OBC and at the moment; the camera experienced problems after the TVAC test in cold temperatures when connected to other devices and it is assumed that the camera and not the OBC failed at the cold temperature.

It can be seen in Figure 27 that the SDRAM reaches a higher temperature than the OBC, this is due to the SDRAM's larger power consumption which occurred during the memory test where the SDRAM was constantly being written to and read from for a period of 10 minutes. The 2 small spikes seen in Figure 27 on the left side are during attempts at getting a response from the camera where the equipment was shut off for

several minutes in order to power cycle the camera and this resulted in the equipment cooling down.

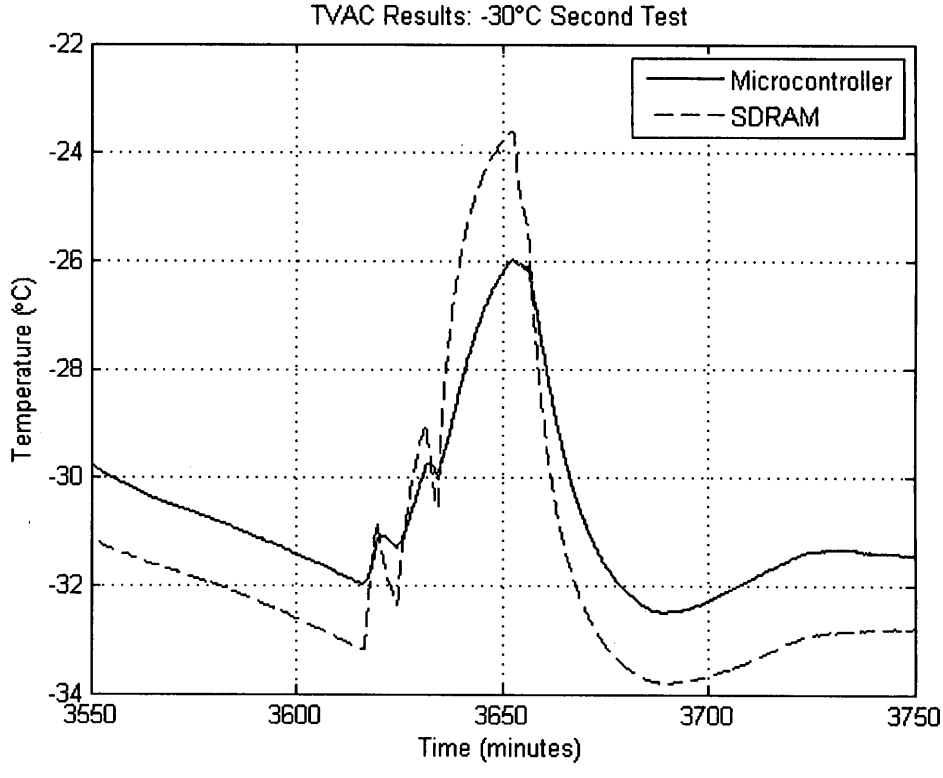

Figure 27 - Second -30°C Test

The second 60°C test is shown in Figure 28 and the resulting curve is similar to Figure 27, in this test as well as all other hot tests, the camera responded to the OBC and pictures were obtained. By comparing the temperature profiles at hot and cold temperatures, the temperature gain of the SDRAM is greater at cold temperatures than hot temperatures for tests running for a comparable amount of time.

The final test is performed at ambient temperature and the temperature profile is shown in Figure 29. The tests were started when the OBC temperature was approximately 24°C and the memory test was left to run for over 2 hours to see the how high the temperature could go. The SDRAM hit a peak of approximately 38°C and the microcontroller reached 36°C this indicated a maximum rise of 14°C after a 2 hour period with maximum possible workload on the SDRAM.

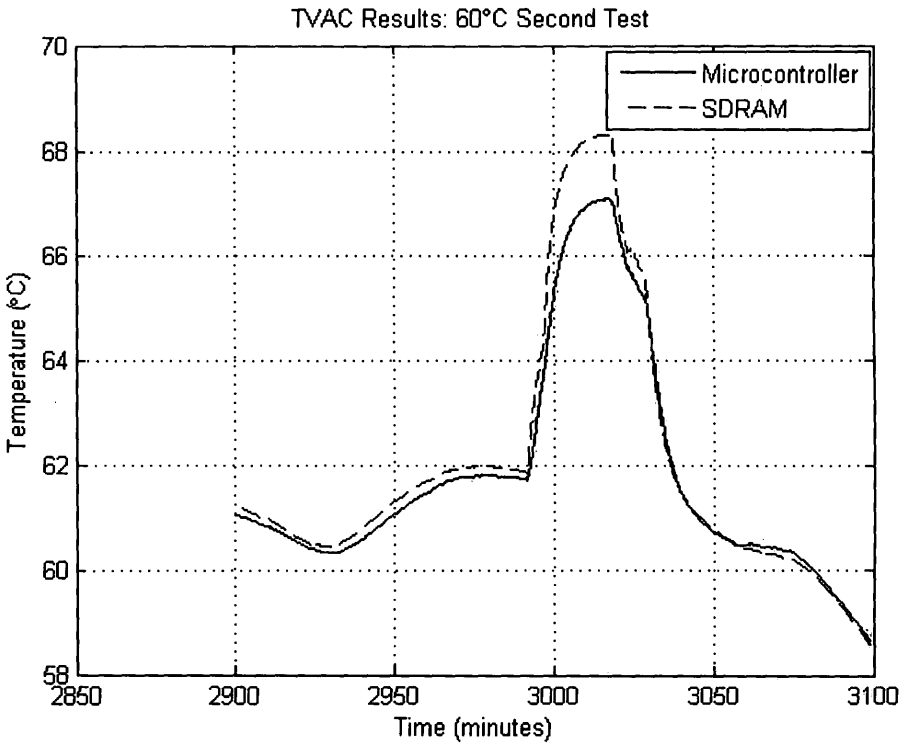

Figure 28 - Second 60°C Test

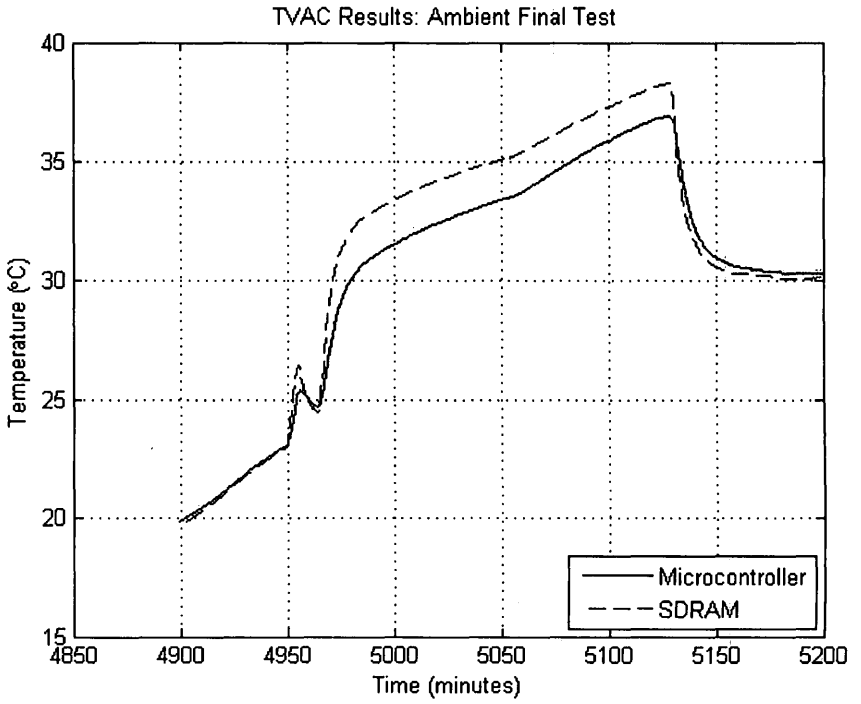

Figure 29 - Final Test at Ambient Temperature

The present OBC design has survived all stages of the TVAC test and has completed all tasks at each testing stage. There is concern about the temperature increase of the microcontroller and SDRAM during the prolonged operation at the final ambient stage. A temperature increase of 17°C at the high temperature operating range of 60°C puts the components close to their operational limit of 85°C. Monitoring the temperature and putting the present OBC design in idle mode to allow cooling down is required to prevent overheating.

Chapter 5 – Final Remarks

A design for a generic nanosatellite OBC is presented. The system is developed based on similar designs of commercial systems such as the Linuxstamp II by Paul Thomas. The functionality included with the new design is to accommodate a wide range of standard interfaces as well as to provide a means to create custom interfaces for generic nanosatellite missions. In particular, the OBC design is suitable for typical CubeSat missions in low Earth orbit for Earth observation or technology demonstration missions.

Characterization of the present OBC design is accomplished by evaluating the power consumption, functionality and the performance of the system through comparison to existing embedded systems as well as a series of environmental tests.

The power requirements of the system during operation vary between 0.5W and 0.8W. Comparison with selected commercial units shows comparable performance. The present OBC design consumes 0.2W more than the Linuxstamp II with the AT91SAM9G20.

Performance tests of the present OBC design are accomplished by comparing computation power with similar systems. The Linuxstamp series embedded systems are used as benchmarks as they use microcontrollers from the same series as the present OBC design. An additional embedded system, the TS7800, is used to compare the performance as it is a commercial embedded system in a PC/104 form factor.

Performance of the OBC is on par with the Linuxstamp II system with the AT91SAM9G20 microcontroller. The other systems that are evaluated for computational performance performed below the level of the present OBC design.

The present OBC design is tested in a thermal vacuum chamber to examine its survivability in the space environment. The OBC performs well under vacuum and extreme temperature conditions in the range of -30°C – 60°C though the heating of the OBC when under load is substantial. Prolonged operation and exposure to high temperatures may cause the present OBC design to reach temperature above its operational limit.

In the space environment, radiation is a concern as it corrupts computer memory and may cause power irregularities in the system. Radiation effects are not examined experimentally in this study as the cost of performing a test at a radiation facility exceeds the scope of the current study. Though radiation effects were not tested they were considered in the selection of the components for the system.

Radiation mitigation effects are considered and a strategy to employ them in software through the use of TMR. TMR is implemented in the first level bootloader to read multiple copies of itself and corrections are made through a voting system. Current implementation of TMR does not check whether differences exist and instead overwrites every memory location.

There is limited power available in space missions, in particular onboard a nanosatellite. It is desired that the power consumption be as low as possible when operating in a space

environment. Lower power consumption reduces the load on the satellite batteries and power subsystem as well as the power being dissipated as heat. High power dissipation causes increase in temperature of the satellite and other subsystems. Electronic devices may work in reduced capacity or fail completely in a temperature outside their rated operation range.

Means to reduce power consumption of the present OBC design are to be investigated. Suggested methods include removing the MRAM from board and disconnecting the 5V output on power regulator. The removal of the MRAM is likely to conserve power as the MRAM consumes power while idle. Quantifying the amount of power used when MRAM is not connected is necessary. If the MRAM requires an amount of power deemed substantial for a nanosatellite mission and it is desired to have the MRAM as part of the OBC, a control circuit that can turn the MRAM fully off may be integrated.

Investigation of the amount of power used up by the voltage regulator for the 5V output is also recommended for future consideration. In a nanosatellite it is common place to have a power subsystem that is in charge of providing power at the necessary voltages and current limits. It is recommended that the power regulator in the present OBC design remains, though the 5V output, if needed, may be supplied by the power subsystem instead.

Future work on the TMR system requires modification of the voting system to only overwrite memory locations that are different to reduce amount of read/write operations. Further work to improve the TMR is also recommended to perform ongoing comparisons

of data in memory. When the Linux kernel is launched, a process can be created that accesses the memory and perform memory comparisons while the system is operating.

References

1. *Small is Beautiful - 1st Report from the Canadian Nanosatellite Workshop.* **1st Canadian Nanosatellite Workshop.** Québec City : s.n., 2012.

2. **California Polytechnic State University.** *CubeSat Design Specification Rev. 12.* San Luis Obispo : s.n., 2009.

3. **California Polytechnic State University**. *Poly Picosatellite Orbital Deployer Mk III ICD.* San Luis Obispo : s.n., 2007.

4. *As ENIAC Turns 50: Perspectives on Computer Science Support for Science and Engineering.* **Eigenmann, Rudolf and Cybenko, George.** 2, s.l. : Computational Science & Engineering, IEEE, 1996, Computational Science and Engineering, IEEE, Vol. 3, pp. 16-17. 1070-9924.

5. **Parlante, Nick.** How Computers Work - Hardware. *Stanford - Introduction to Computer Principles.* [Online] 2012. [Cited: October 22, 2012.] http://www.stanford.edu/class/cs101/how-computers-work-hardware.html.

6. **Xu, Jia.** Operating System Design. Toronto : s.n., 2011.

7. **NASA Orbital Debris Program Office.** NASA Safety Standard - Guidelines and Assessment Procedures for Limiting Orbital Debris. *NASA Orbital Debris Program Office.* [Online] August 1995. [Cited: October 16, 2012.] http://www.orbitaldebris.jsc.nasa.gov/library/NSS1740_14/nss1740_14-1995.pdf.

8. **Gloutnary, Eric and Cusson, Jean-Francois.** FPGA/Radiation Effects 101. Saint-Hubert, Quebec, Canada : CSA FPGA Workshop, 2011.

9. **UTIAS/SFL.** Nanosatellite Launch Services. *UTIAS Space Flight Laboratory.* [Online] [Cited: November 20, 2012.] http://www.utias-sfl.net/Products/NLS-Handout-20090729-press.pdf.

10. **Brand, Christiaan Johannes Petrus.** The Development of an ARM-based OBC for a Nanosatellite. Stellenbosch, South Africa : University of Stellenbosch, December 2007.

11. **Dwyer, Mark Matthew Paul Chamption.** Embedded Software Design for the Canadian Advanced Nanospace eXperiment Generic Nanosatellite Bus. Toronto : University of Toronto, 2009.

12. *Architecture of a Small Low-Cost Satellite.* **Del Corso, D., et al.** Lubeck : Digital System Design Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference on, 2007. Digital System Design Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference on. pp. 428-431. 978-0-7695-2978-3.

13. **ÅAC Microtec.** TAMU On Sprite-Sat. *ÅAC Microtec Website.* [Online] January 23, 2008. [Cited: October 22, 2012.] http://www.aacmicrotec.com/index.php/tamu-on-sprite-sat.html.

14. **Mok, Cecilia Ka Vai.** Design and Implementation of the Flight Application Software and NanoSatellite Protocol for the CanX-2 Nanosatellite. Toronto : University of Toronto, 2005.

15. *Deep Impact, the mission.* **Muirhead, B.K.** s.l. : Aerospace Conference Proceedings, 2002. IEEE, 2002. Aerospace Conference Proceedings, 2002.IEEE. Vol. 1, pp. 1-147 - 1-155. 0-7803-7231-X.

16. **UTIAS/SFL.** The AISSat-1 Mission. *UTIAS-SFL Website.* [Online] 2011. [Cited: October 22, 2012.] http://www.utias-sfl.net/nanosatellites/AISSat-1/.

17. **PC/104 Embedded Consortium.** *PC/104 Plus Specifications - Version 2.3.* s.l. : PC/104 Specifications, 2008.

18. **Pumpkin Corporation.** In Space. *Pumpkin CubeSat Kit.* [Online] 2011. [Cited: October 22, 2012.] http://www.cubesatkit.com/content/space.html.

19. **Gumstix.** Overo FE COM. *Gumstix Web Site.* [Online] [Cited: October 22, 2012.] https://www.gumstix.com/store/product_info.php?products_id=256.

20. **Thomas, Paul.** Linuxstamp II 9260. *Open Circuits.* [Online] 2010. [Cited: October 22, 2012.] http://opencircuits.com/Linuxstamp_II_9260.

21. **LittleChips.** LittleChips. *LittleChips Web Site.* [Online] 2008. [Cited: October 22, 2012.] http://littlechips.com/LN2440SBC.htm.

22. **Hardent Inc.** FPGA for Managers. Saint-Hubert, Quebec, Canada : CSA FPGA Workshop, 2011.

23. **Berg, Melanie.** New Developments in Field Programmable Gate Array (FPGA) Single Event Upsets (SEUs) and Fail-Safe Strategies. Saint-Hubert : CSA FPGA Workshop, 2011.

24. **UK Space Agency.** UKube-1 Mission. *Department for Business Innovations and Skills: UK Space Agency.* [Online] September 19, 2012. [Cited: October 22, 2012.] http://www.bis.gov.uk/ukspaceagency/missions/ukube-pilot-programme.

25. **Xilinx Inc.** Xilinx Press Release #03135. *Xilinx Inc.* [Online] September 25, 2003. [Cited: October 22, 2012.] http://www.xilinx.com/prs_rls/design_win/03135optus.htm.

26. **Despong, Bastien.** *System Engineering and Development and Test of the ADCS Breadboard for SwissCube.* Lausanne : s.n., 2007.

27. **Atmel Corporation.** AT91SAM9G20 Complete. *Atmel ARM Solutions.* [Online] February 2010. [Cited: October 22, 2012.] http://www.atmel.com/devices/SAM9G20.aspx?tab=documents.

28. **Micron Technology.** MT29F2G08ABAEAWP-IT Information. *Micron Technology Website.* [Online] February 2012. [Cited: October 22, 2012.] http://www.micron.com/parts/nand-flash/mass-storage/mt29f2g08abaeawp-it?source=ps.

29. **Macronix International Co.** MX29GL256F H/L. *Macronix Parallel Flash Products.* [Online] October 2011. [Cited: October 22, 2012.] http://www.macronix.com/QuickPlace/hq/PageLibrary4825740B00298A3B.nsf/h_Index/

6F878CF760C559BD482576E00022E6CC/?OpenDocument&EPN=MX29GL256F%20H/L.

30. **Ramtron.** Product Detail: FM23MLD16. *Ramtrom.* [Online] June 2012. [Cited: October 22, 2012.] http://www.ramtron.com/products/nonvolatile-memory/parallel-product.aspx?id=116.

31. **IBM.** Understanding DRAM Operation. *Memory System Architecture.* [Online] 1996. [Cited: October 22, 2012.] www.ece.cmu.edu/~ece548/localcpy/dramop.pdf.

32. *Trends in bipolar static random access memory (SRAM) design* . **Herndon, W.H.** Minneapolis : s.n., 1989.

33. **IBM.** Understanding Static RAM Operation. *Memory System Architecture.* [Online] 1997. [Cited: October 22, 2012.] http://www.ece.cmu.edu/~ece548/localcpy/dramop.pdf.

34. *Embedded dynamic random access memory.* **Kirihata, T.** Hsinchu : International Symposium on VLSI Technology, Systems, and Applications, 2003. 1524-766X.

35. **Micron Technology.** MT48LC16M16A2P-75 IT Information. *Micron Technology Website.* [Online] January 2010. [Cited: October 22, 2012.] http://www.micron.com/parts/dram/sdram/mt48lc16m16a2p-75-it?pc={428A5CC9-2A78-447E-939B-6F3A40D538C6}.

36. *Design of Spin-Torque Transfer Magnetoresistive RAM and CAM/TCAM with High Sensing and Search Speed.* **Xu, Wei.** 1, s.l. : IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2010, Vol. 18. ISSN 1063-8210.

37. **Elghefari, Mohamed and McClure, Steve.** *Radiation Effects Assessment of MRAM Devices.* Passadena : JPL Publication, 2008.

38. **Simpson, Chester.** Linear and Switching Regulator Fundamentals. *National Semiconductor.* [Online] [Cited: October 22, 2012.] http://www.national.com/assets/en/appnotes/f4.pdf.

39. **Zant, Peter Van.** *Microchip Fabrication.* New York : McGraw-Hill Companies, Inc., 2004. ISBN 0-07-143241-8.

40. **Vishay Precision Group.** Why Choose Bulk Metal Foil Resistors For Your Circuit? *Digi-Key Corporation.* [Online] July 1, 2010. [Cited: October 22, 2012.] http://www.digikey.com/Web%20Export/Supplier%20Content/VishayPrecisionGroup_804/PDF/VishayPG_Why_Chosse_Foil_Resistors.pdf?redirected=1.

41. *Thin-Film Thermo-Resistor Radiation Hardness Experimental Results.* **Nikiforov, Alexander Y. and Telets, Vitally A., Figurov, Valerie S.** Showmass Village : Radiation Effects Data Workshop, IEEE, 1997. 0-7803-4061-2.

42. **Analog Devices.** Decoupling Techniques. *Analog Devices.* [Online] 2009. [Cited: October 22, 2012.] http://www.analog.com/static/imported-files/tutorials/MT-101.pdf.

43. **AVX Corporation.** Glass Capacitors Introduction and Performance Curves. *AVX Corporation Website.* [Online] [Cited: October 22, 2012.] http://www.avx.com/docs/Catalogs/gl-cyintro.pdf.

44. **AVX Corporation.** AVX Glass Capacitor Critical To Cassini-Huygens Mission. *AVX Corporation Website.* [Online] [Cited: October 22, 2012.] http://www.avx.com/wsnw_PressReleaseDetail.asp?id=301&s=1.

45. **AVNET Electronics Marketing.** AVX CYR15C471GS. *AVNET Electronics Marketing.* [Online] [Cited: October 16, 2012.] http://avnetexpress.avnet.com/store/em/EMController/Capacitor-Ceramic-Multilayer/AVX/M23269-02-7026/_/R-1438299/A-1438299/An-0?action=part&catalogId=500201&langId=-1&storeId=500201&listIndex=-1&page=1&rank=0.

46. **Nichicon Corporation.** General Description of Aluminum Electrolytic Capacitors. *Nichicon Corporation.* [Online] [Cited: October 22, 2012.] http://www.nichicon.co.jp/english/products/pdf/aluminum.pdf.

47. *A brief introduction to ceramic capacitors.* **Pan, Ming-Jen.** 3, s.l. : Electrical Insulation Magazine, IEEE, 2010, Vol. 26. 0883-7554.

48. *New tantalum capacitors in power supply applications.* **Prymak, J.D.** St. Louis : Industry Applications Conference. Thirty-Third IAS Annual Meeting. IEEE, 1998. 0-7803-4943-1.

49. **Altium Limited.** Altium Designer. *Altium Limited.* [Online] [Cited: October 16, 2012.] http://www.altium.com/.

50. **JPL NASA.** Command & Data Handling Systems. *JPL Mars Reconnaissance Orbiter.* [Online] [Cited: August 22, 2012.] mars.jpl.nasa.gov/mro/mission/spacecraft/parts/command/.

51. **Aho, Alfred V., et al.** *Compilers - Principles, Techniques and Tools 2nd Edition.* Boston : Pearson Education Inc., 2007. 0-321-48681-1.

52. **Yaghmour, Karim.** *Building Embedded Linux System.* Sebastopol : O'Reilly Media, 2009. 978-0-596-10327-9.

53. *Use of JTAG Boundary-Scan for Testing Electronic Circuit Boards and Systems.* **Ngo, Be Van, Law, Peter and Sparks, Anthony.** Salt Lake City : Autotestcon, IEEE, 2008. 1088-7725.

54. *How Fast is Fast Enough? Choosing Between Xenomai and Linux for Real Time Applications.* **Brown, Jeremy H. and Martin, Brad.** Nairobi : OSADL Project: Real Time Linux Workshops, 2010.

55. **Manning, Charles.** Comparison between Yaffs (Yaffs2) and JFFS2. *YAFFS Website.* [Online] 2007. [Cited: 20 October, 2012.] http://www.yaffs.net/comparison-yaffs-vs-jffs.

56. **Linux-MTD.** UBI - Unsorted Block Images. *Linux-MTD.* [Online] January 13, 2009. [Cited: October 20, 2012.] http://www.linux-mtd.infradead.org/doc/ubi.html.

57. *Update on Filesystems for Flash Storage.* **Opdenacker, Michael.** Sophia-Antipolis : Journées Méditerranéennes du Logiciel Libre, 2008.

58. **Keppel, David, et al.** Linux Man Pages - Time. *Linux Man Pages.* [Online] December 11, 2000. [Cited: October 18, 2012.] http://www.linuxmanpages.com/man1/time.1.php.

59. *Contact Temperature Measurement System Based on Tungsten-Rhenium Thermocouple.* **Wang, Daihua, Song, Linli and Zhang, Zhijie.** Taiyuan : Computer Application and System Modeling (ICCASM), 2010 International Conference on, 2010. Computer Application and System Modeling (ICCASM), 2010 International Conference on. pp. V12-660 - V12-663. 978-1-4244-7237-6.

60. **Association Connecting Electronic Industries.** *IPC-2221A.* 2003.

61. **UTIAS/SFL.** CanX-2 System. *Space Flight Laboratory Missions.* [Online] 2011. [Cited: October 22, 2012.] http://www.utias-sfl.net/nanosatellites/CanX2/system.html.

Appendix A – OBC Schematics

The appendix includes all schematics representing the present OBC design.

A. Microcontroller – AT91SAM9G20

Microcontroller - Part 1

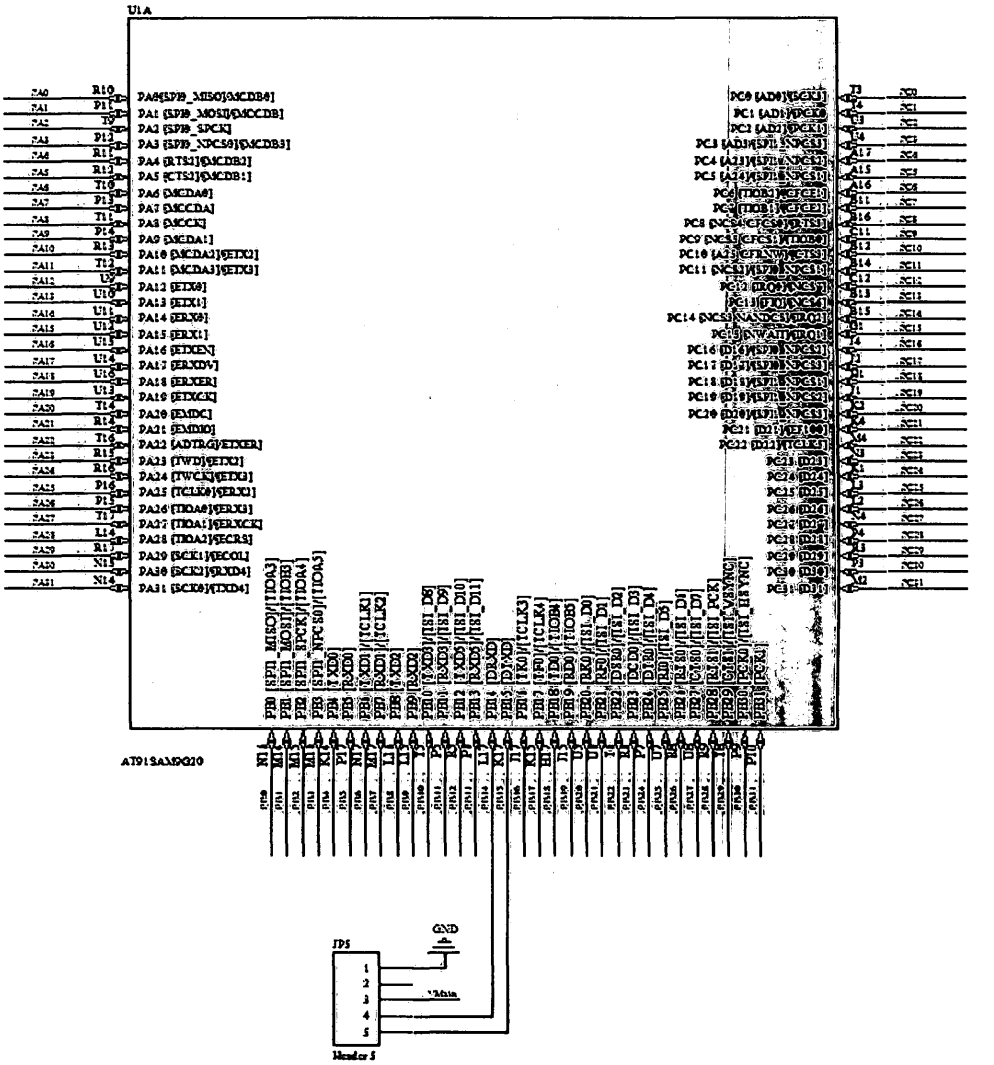

Figure 30 - Microcontroller Schematic - Part 1

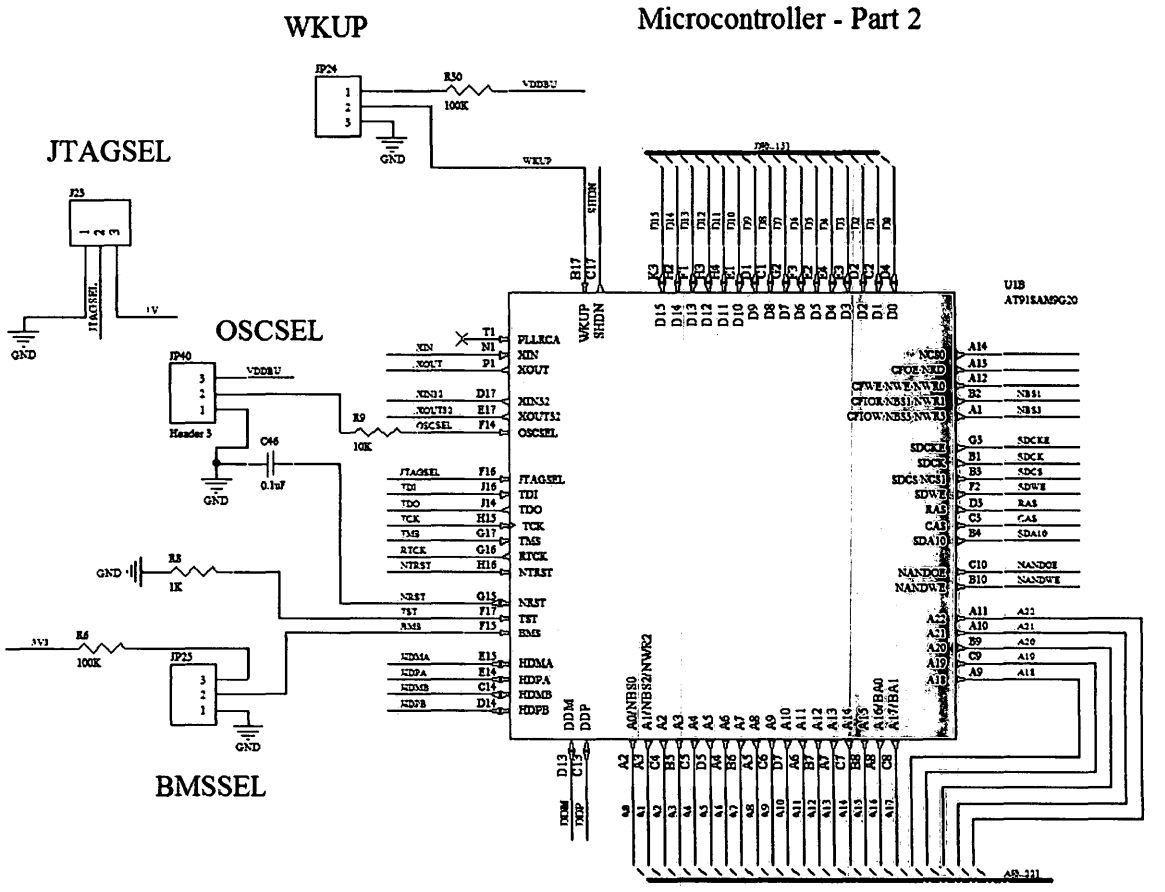

Figure 31 - Microcontroller Schematic - Part 2

Microcontroller - Part 3

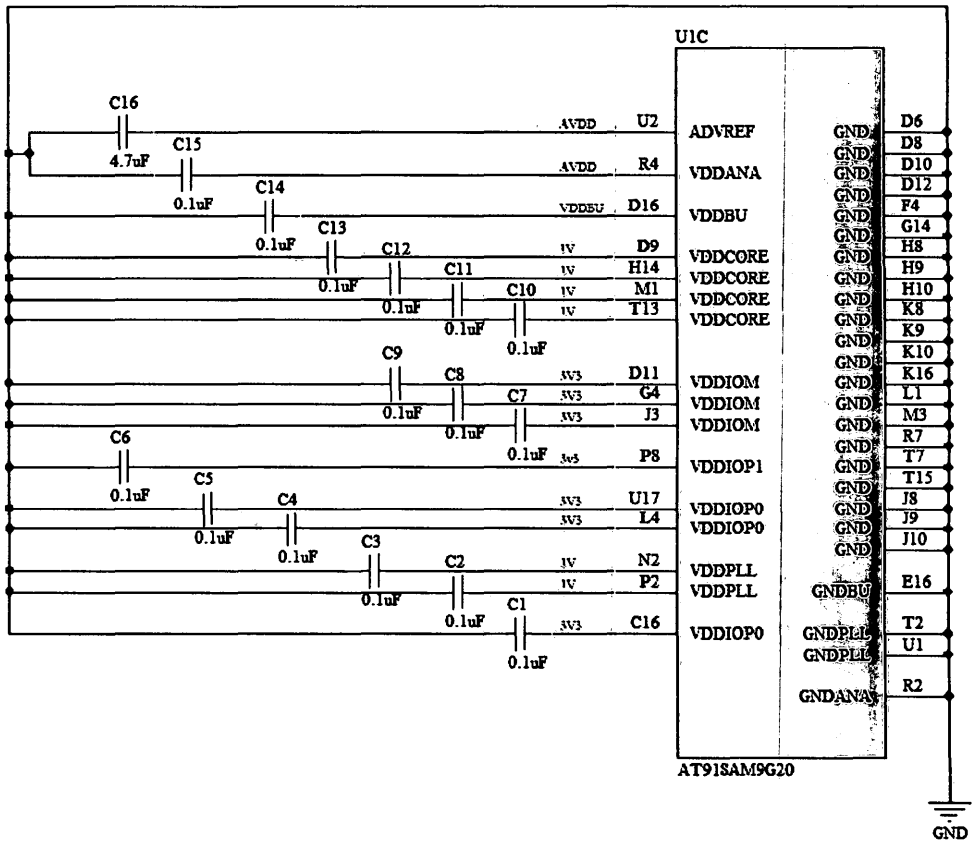

Figure 32 - Microcontroller Schematic - Part 3

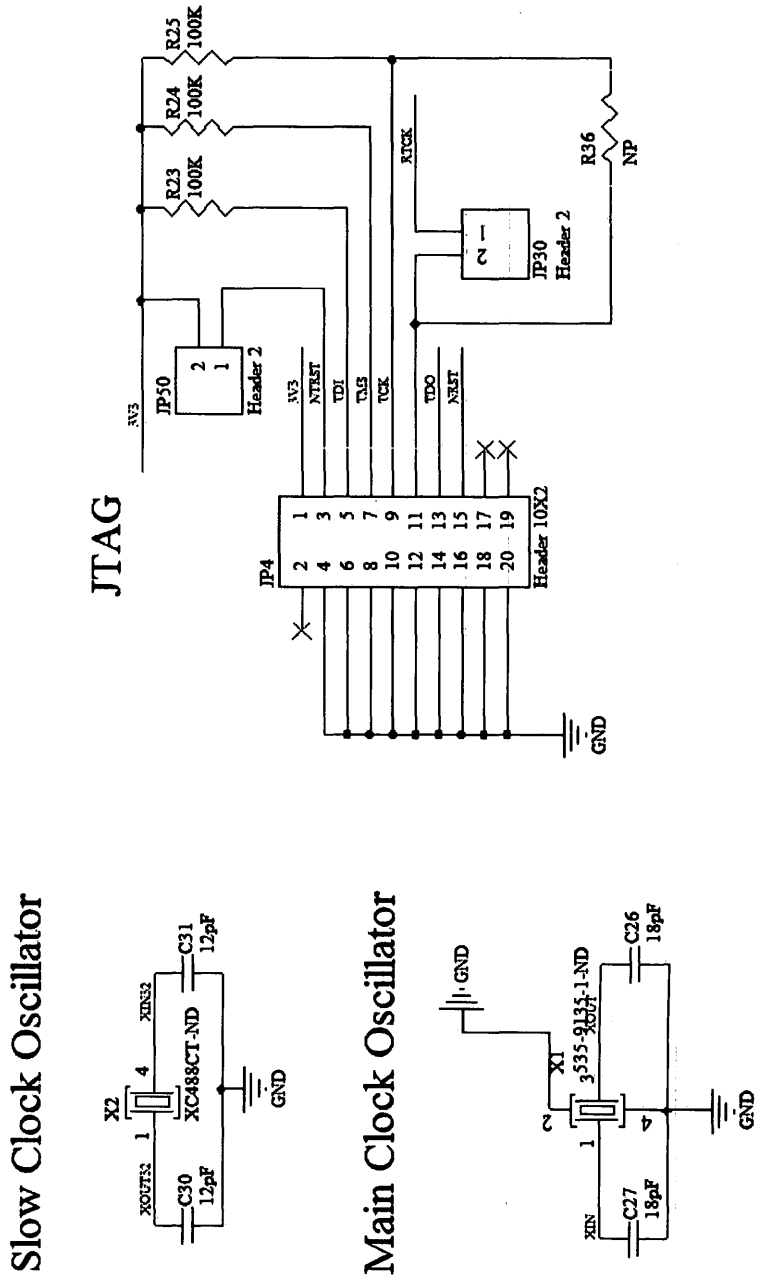

Figure 33 - Microcontroller Schematic - Part 4

B. Power Regulator Schematic

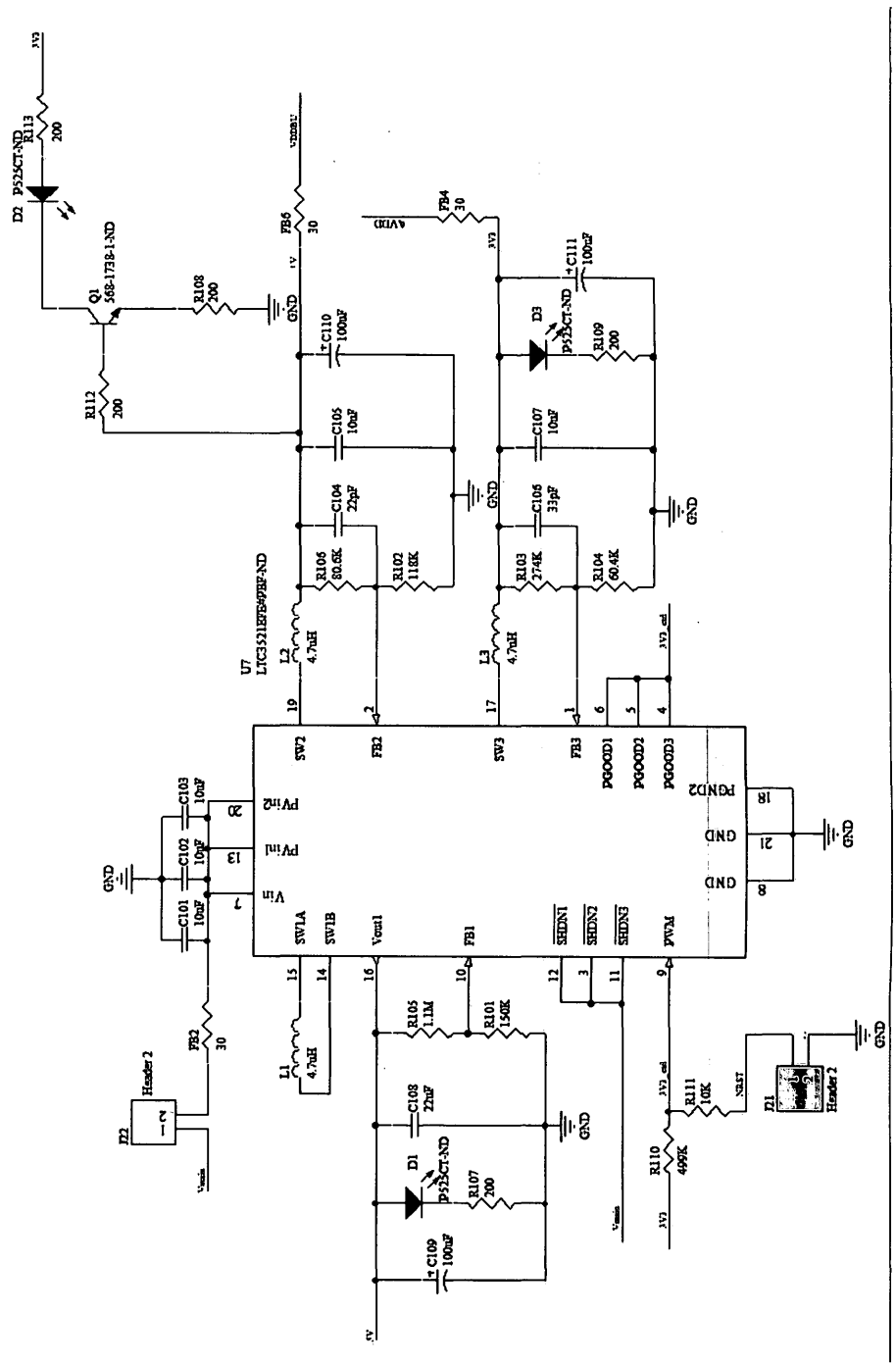

Figure 34 - Power Regulator Schematic

C. SDRAM Schematic

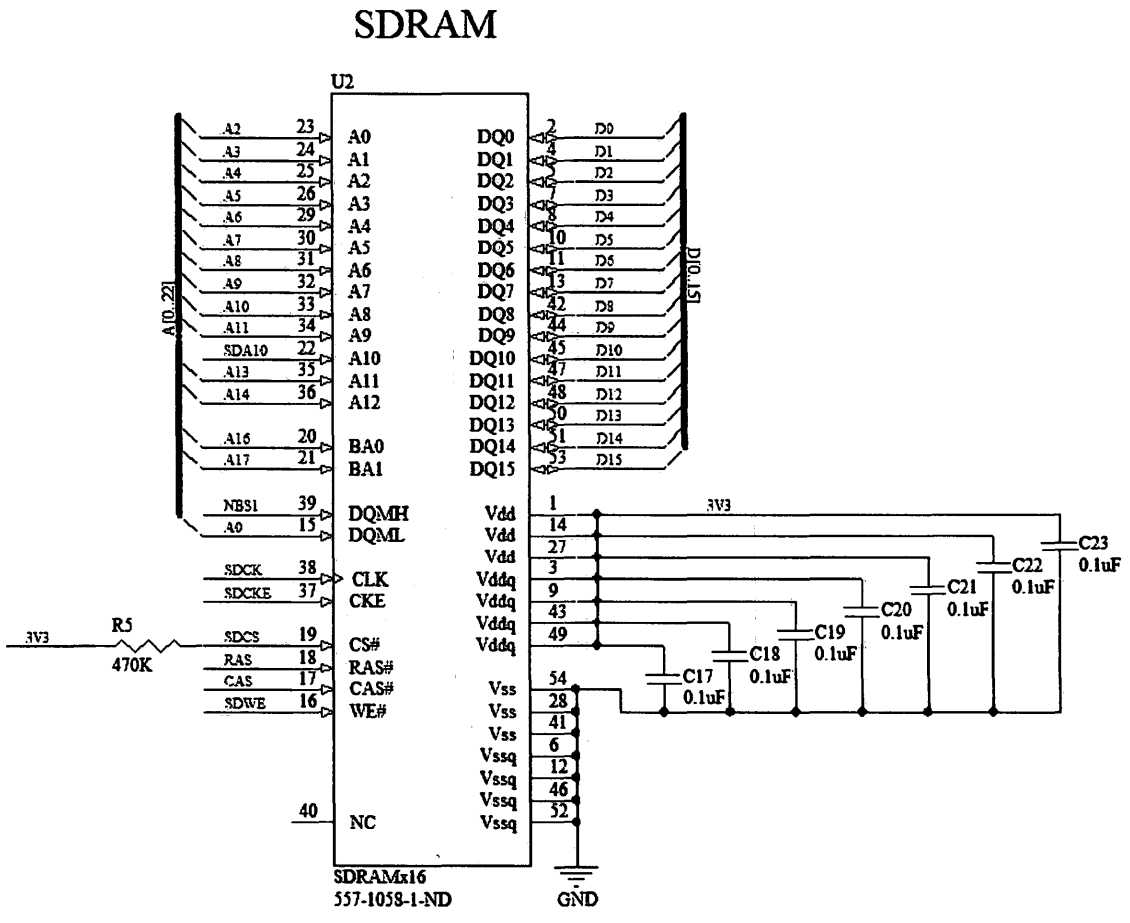

Figure 35 - SDRAM Schematic

D. NAND Flash Schematic

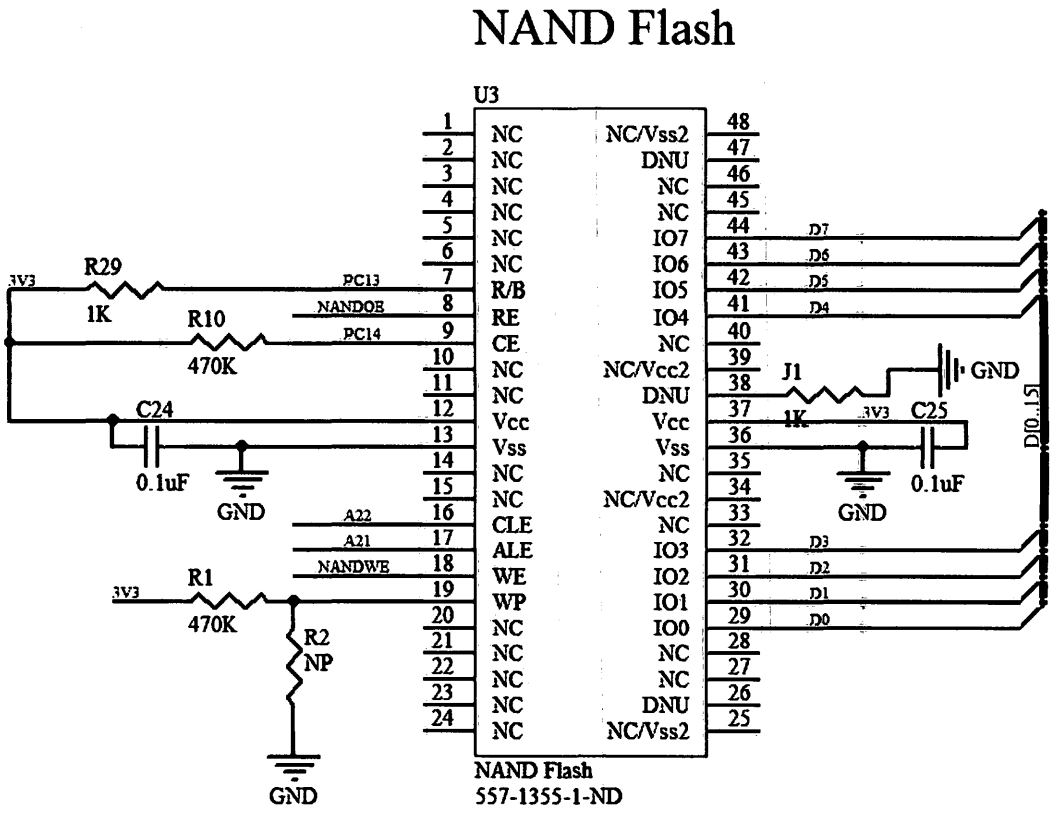

Figure 36 - Nand Flash Schematic

E. MRAM Schematic

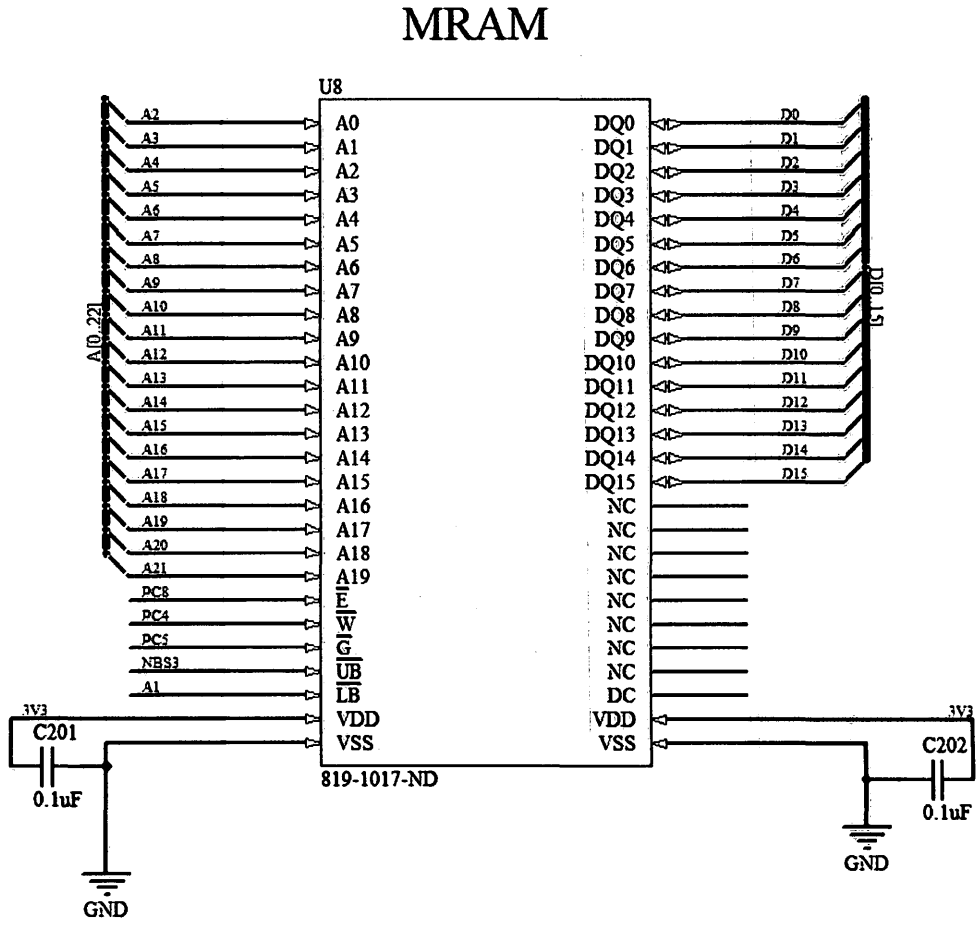

Figure 37 - MRAM Schematic

F. Headers Schematic

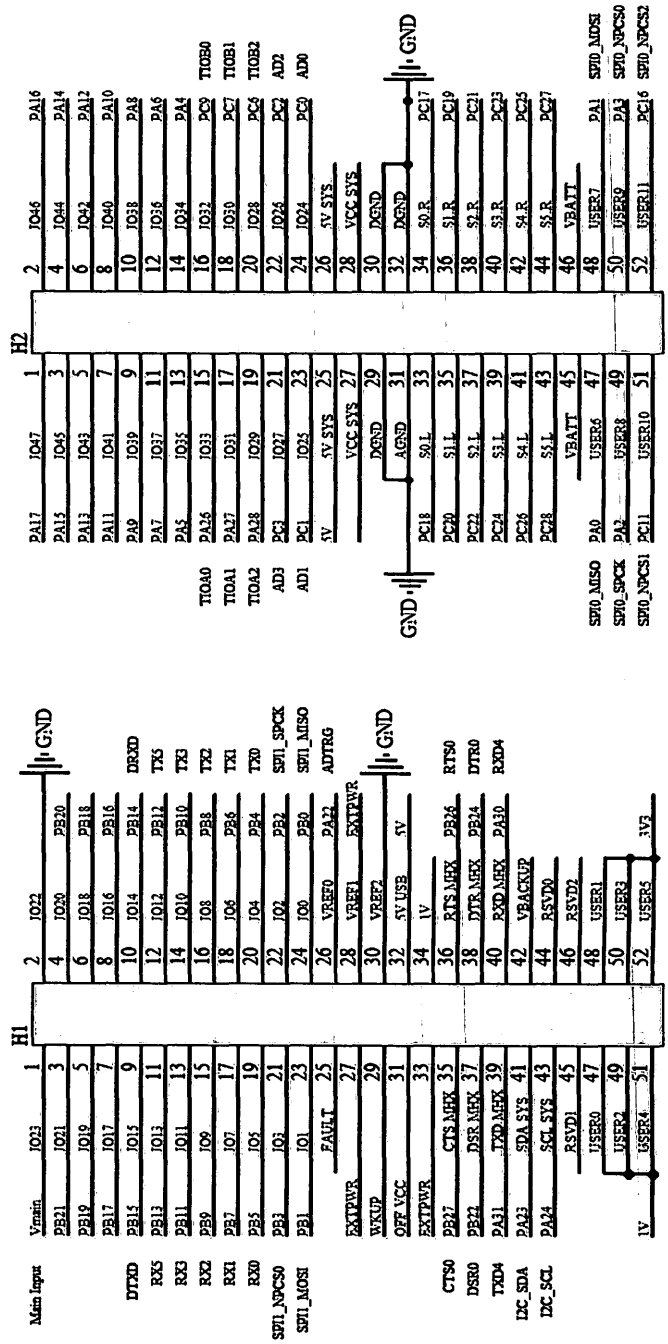

Figure 38 - Headers Schematic

Appendix B – TMR Sample Code

Presented is a sample code implementing the triple modular redundancy algorithms developed for use as part of the AT91Bootstrap bootloader.

```
#define SIZE 0x1508 /* Size of bootstrap file - Important
for TMR, boot may fail if incorrect value used */

volatile unsigned char tmr_voter (unsigned int a, unsigned
int b, unsigned int c)
{
     unsigned int cmp1, cmp2;
     unsigned int temp, result;
     cmp1 = (a)^(b); /* XOR */
     cmp2 = (a)^(c); /* XOR */
     temp = ~(cmp1&cmp2);
     result = temp^(~(a));
     return result;
}
```

Below is an example of a call to the TMR function that write the results to the memory locations.

```
#define writel(value, address) \
     (*(volatile unsigned int *)(address)) = (value)
#define readl(address) \
     (*(volatile unsigned int *)(address))

int limit = 0;
int start = 0;
unsigned int a, b, c, result;
while (limit < SIZE) /* traverse the memory locations */
{
     /* Obtain current memory data */
     a = readl(start);
     b = readl(start+SIZE);
     c = readl(start+SIZE+SIZE);
     /* Compare memory data using TMR */
     result = tmr_voter(a, b, c);
```

```
        /* Overwrite old data with results from TMR */
        writel(result, start);
        writel(result, start+SIZE);
        writel(result, start+SIZE+SIZE);
        limit = limit + 1;
}
```