A field programmable gate array based on-board computer for small satellites.

by

DIRK JACOBUS VAN WYK

Thesis submitted in fulfilment of the requirements for the degree

Master of Engineering: Electrical Engineering

in the Faculty of Engineering and the Built Environment

at the Cape Peninsula University of Technology

Supervisor: Dr. Vipin Balyan

Bellville

September 2020

DECLARATION

I, DIRK JACOBUS VAN WYK, declare that the contents of this dissertation/thesis represent my own unaided work, and that the dissertation/thesis has not previously been submitted for academic examination towards any qualification. Furthermore, it represents my own opinions and not necessarily those of the Cape Peninsula University of Technology.

25 SEPTEMBER 2020

Signed **Date**

ABSTRACT

This thesis will focus on the detailed development and design of an embedded on-board computer (OBC) for small satellites based on high density digital Field Programmable Gate Array (FPGA) technology. The OBC Processor-Board is intended to be designed and developed as a Single Board Computer (SBC) that may be utilised for a variety of aerospace applications. The architecture of the OBC will be developed around the FPGA as the main processor. Investigating with comparison of pervious OBC processors alongside the commonly used microcontroller as the main processor. This will include the overview of the hardware implementation of complex algorithms to the advantage of the OBC.

It is anticipated that the Single Board Computer can be used as a configurable on-board computer with high flexibility allowing in-orbit reconfiguration. Modern FPGAs are designed with embedded processing systems integrated inside the core allowing monotonous performance task to execute more easily with flexibility. One single printed circuit board with the computing power that will handle all the challenging requirements for performance and functionality of the payload data handling.

ACKNOWLEDGEMENTS

The author would like to extend his gratefulness and thank the following people for their contributions which led to the success of this thesis:

- **God**, for giving me guidance throughout this journey and His endless love.
- **Dr. Vipin Balyan** (Supervisor), All the assistance and his patience, support, with guidance during the course of this thesis.
- **Prof. Robert van Zyl**, by providing me with the opportunity to pursue a master’s degree and his support.
- **Ms. Marissa Massyn**, for her patience, selfless support, and love throughout this challenging journey.

Table of Contents

ACRONYMS

1U	One unit CubeSat; 100mm x 100mm x 100mm
ADCS	Attitude determination and control system
ASIC	Application Specific Integrated Circuit
C&DH	Control and Data Handling
CLB	Configurable Logic Block
CMOS	Complementary Metal-Oxide Semiconductor
COTS	Commercially-Off-The-Shelf
CPU	Central Processing Unit
DCM	Digital Clock Manager
DDR	Double Data-Rate
DSP	Digital Signal Processor
EMI	Electromagnetic Interference
EO	Earth Observation
EPS	Electrical power supply
FIFO	First In First Out
FIFO	First In First Out
FIL	FPGA-in-the-Loop
FLP	Flying Laptop
FPGA	Field Programmable Gate Array
GCR	Galactic Cosmic Ray
GND	Ground
GPS	Global Positioning System
HIL	Hardware in the loop
IEEE	The Institute of Electrical and Electronics Engineers Standards Association
IO	Input/Output
IOB	Input/Output Block
JTAG	Joint Test Action Group
LC	Logic Cell
LE	Logic Element
LEO	Low Earth Orbit

LUT	Look-Up Table
LVDS	Low Voltage Differential Signal
MCU	Microcontroller Unit
NASA	National Aeronautics and Space Administration
OBC	On-Board Computer
PCB	Printed Circuit Board
PCC	Power control circuit
PCI	Peripheral Component Interconnect
PCU	Power Control Unit
PLL	Phase Locked Loop
POL	Point of Load
PRN	Pseudo Random Number
RAM	Random Access Memory
SBC	Single Board Computer
SEE	Single Event Effects
SEL	Single Event Latch-up
SET	Single Event Transient
SEU	Single Event Upsets
SoC	System on Chip
SRAM	Static Random-Access Memory
TC	Thermal Cycle
TID	Total Ionization Dose
TMR	Triple Module Redundancy
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
VHDL	Verilog Hardware Descriptive Language
WBS	Work Breakdown Structure

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

1.1 General background

The small satellite industry has grown rapidly over the past few years with designs for a variety of applications and missions. The satellite system structure can be considered as the core structure and a varying one. The payload of the satellite will be the variable component specific to the mission at hand. The development of small satellites can be of great advantage for Low Earth Orbit (LEO) space applications at a lower cost and less development time. The typical small satellite structure consists of several subsystems with ground stations namely the OBC as the main component of the satellite for Control and Data Handling (CDH). Additional structures include the communication system used amid the ground station and with the OBC, making it probable to retrieve remote control as well as sending commands to the satellite. The satellite will have a Power Control Unit (PCU) for all the power demands and the mission/application specific payload firmware with the actual hardware.

This thesis will focus on the development of the on-board computer (OBC) based on a Field Programmable Gate Array (FPGA) as the main processor for small satellites. The recent on-board computer that have been used, and still are in use, within the aerospace field are based on typical space heritage processors. The comparison between the standard microcontroller as the main component of the OBC to that of a FPGA based OBC will be investigated with the development process in this thesis.

1.2 Literature review

Small satellites, nanosatellites have been a trending industry towards the Space 4.0 paradigm. This architype is driven by commercially minded aerospace partners and companies, enterprises working separately to acquire more reliable access to space on the faster and cost-effective side. Nanosatellites are losing defined as any small satellite weighing less than 10kg. Most notable applications and missions for small satellites are for Earth observation analysis the impact of humans on agriculture, forests, geology, and the environment.

The use of Commercially-Off-The-Shelf (COTS) components to build small satellites, CubeSats, at lower costs with endless mission possibilities. The standard CubeSat is 100mm x 100mm x 100mm in structure known as a 1U Satellite. These satellites can be stacked for greater flexibility of applications with structures as 2U or 3U.

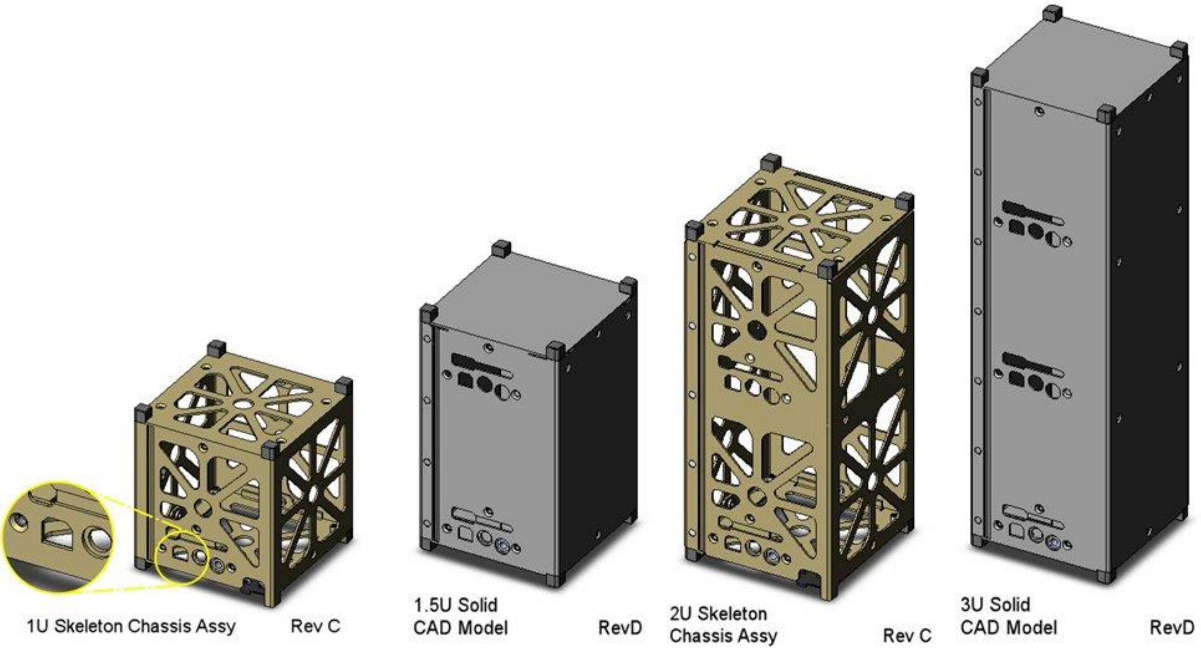


Figure 1.1: CubeSat structure unit sizes [1]

The internal electronic hardware is based on the PC/104 form specifications. The OBC's printed circuit board will have a PC/104 format with all the peripheral support and functionality. The OBC includes the memory subsystem while being the centre of all communication through the serial bus interface.

Nanosatellites or picosatellites are extremely small in physical size and consist of lightweight structures.(Antunes Sandy, 2015) Standard CubeSats are typically 100 mm H x 100 mm x 100 mm in structure size known as 1U. The PocketQube is measured at 50 mm H x 50 mm x 50 mm that is known as 1P. The PocketQube concept was first introduced in 2009, by Professor Robert Twiggs. Satellites have many applications in the new space age, with Earth observation (EO) being one of the most popular missions and primary applications. EO satellites use smart image sensors to monitor and capture data from the Earth's surface, with infrared for in depth data collection. This data may then be utilized to monitor urban development, vegetation growth and natural disaster, etc. The use of satellite imaging technology constantly evolving and improving, the capture information being of higher detail with much improved resolution of the images captured. It is all achieved through the use of a wide range of spectral band cameras (Hillier, 2018). Earth observation has been performed since the mid 1900's by the United States meteorological satellite. The purpose of earth observation is to measure the earth's characteristic to better understand weather patterns, natural disasters, water behaviour and pollutions.(Lokman *et al.*, 2017)

The PocketQube is a cube shaped platform with a mass of less than 250 gram. The first PocketQube was launched using the Morehead Rome Femto Orbital Deployer, installed inside the UniSat-5 microsatellite.

The small satellite industry has grown rapidly the past few years. The same counts for the research and development in space miniaturized technology, small satellite constellations for various missions with new payloads. Building small satellites can become exorbitantly expensive. This can be overcome by using COTS components and open source hardware to build a PocketQube or CubeSat at a reasonable price with endless possibilities.(van Wyk, 2018)

1.3 Single Board Computer

The Single Board Computer (SBC) is the well-known term used for the intended computer which will integrate the necessary elements for the processor the perform its main functions. All the communication interfaces and memory banks among others are included in the SBC. The on-board computer for a satellite is developed of an SBC, which may be used for various functions other than the control of a satellite.

The OBC is the core component of a spacecraft or small satellite structure which houses the hardware and firmware required for Control and Data Handling (CDH). The basic architecture of a small satellite is illustration in Figure 1.2.

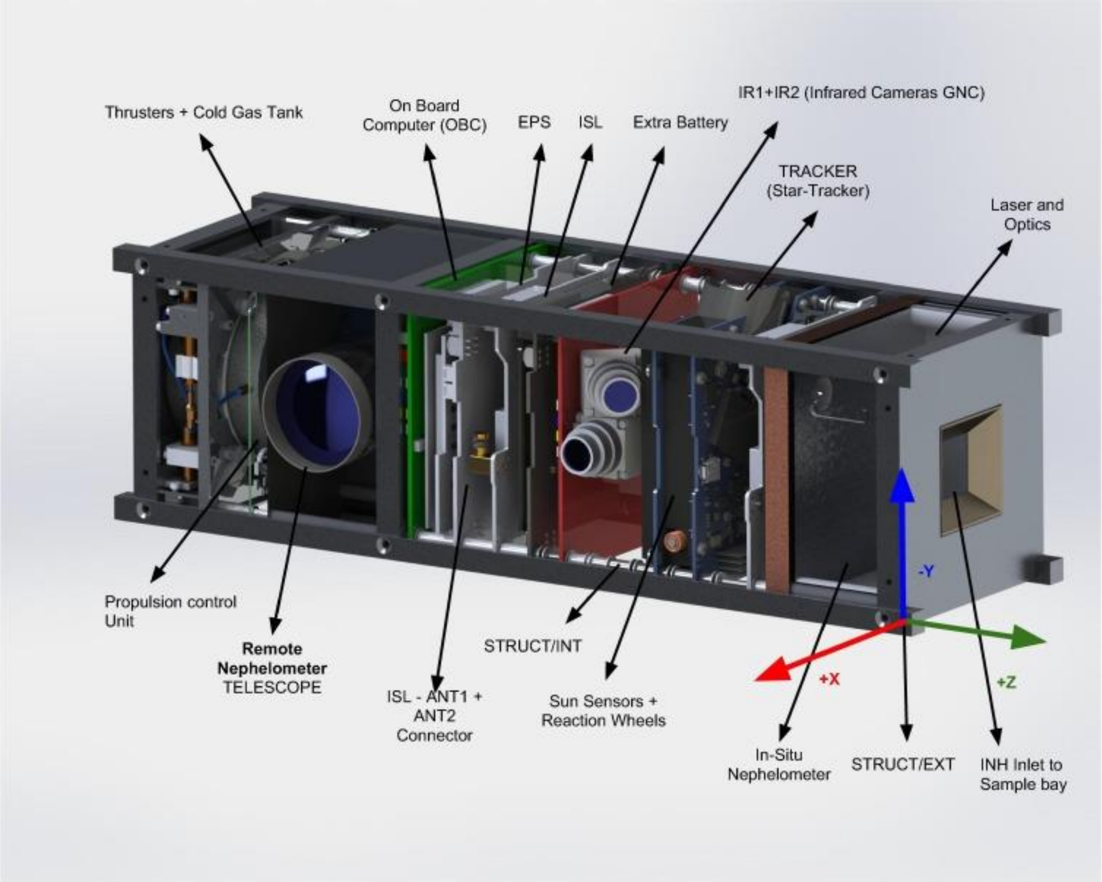


Figure 1.2: Basic structural layout of a CubeSat (Franco, 2017)

The common OBC is based on a microcontroller such as the 32-bit ARM Cortex-M3 based MCU. The microcontroller (MCU) is selected based on the performance features it provides as the whole OBC is based on this main processor for interfacing of all the major elements.

Most on-board computers that have be used in the space environment are based on the microcontroller architecture. Although these MCUs provide robust and predictable performance outcomes in the harsh space environment. Factors that will affect electronics in space include single event radiation effects and charged particles, which need to be considered when choosing the main components of the SBC.(Uryu, 2013)

1.4 FPGA Based On-Board Computer

The Flying Laptop (FLP) micro-satellite developed by the University of Stuttgart has a mission object to validate and quantify new small satellite technologies for near future missions and projects.(Huber *et al.*, 2007)

The overwhelming demand for performance and functionality of new on-board computers creating more complex challenges. The FPGA technology allows for comprehensive functionality of the OBC on a single PCB, with highly integrated and configurable system-on-chip approach. The whole concept is to have a fully reconfigurable system within the OBC in-orbit due to the high flexibility of the FPGA technology.

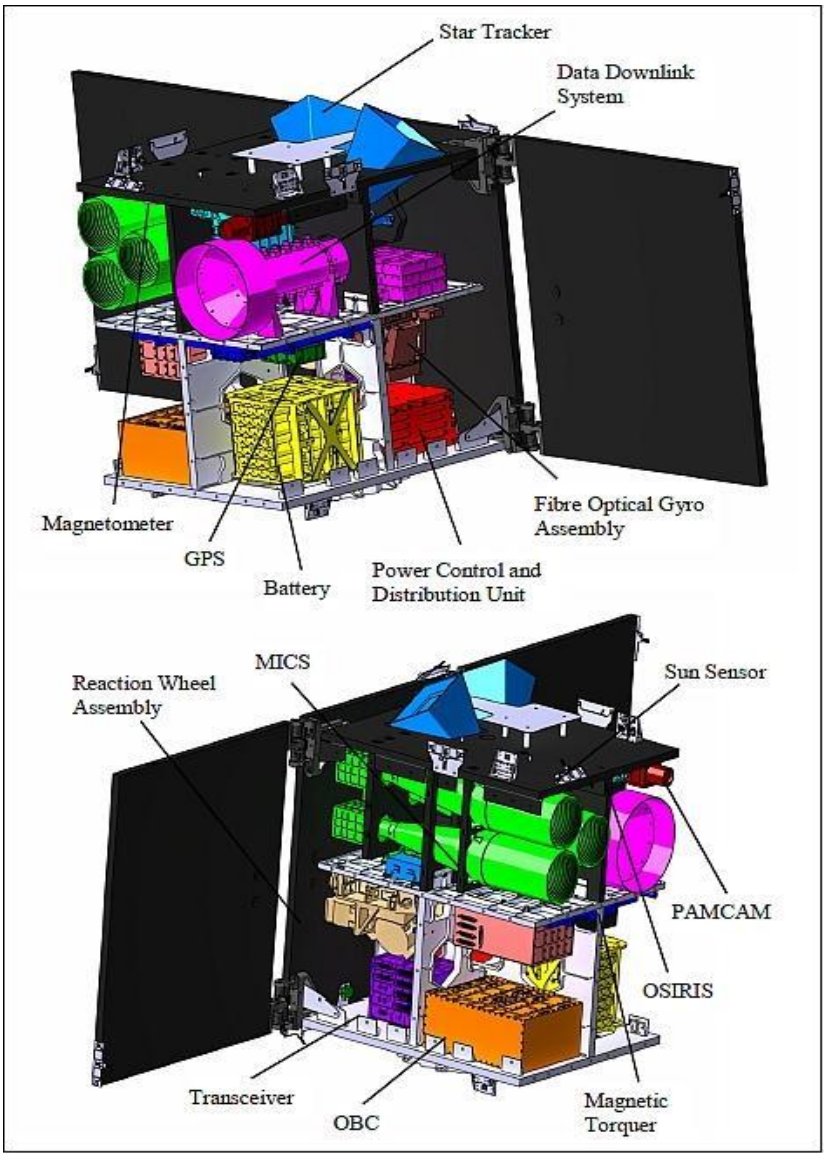


Figure 1.3: The Flying Laptop satellite configuration by IRS Stuttgart.

By utilising digital FPGAs as the main processor for the OBC that is developed to compute rigorous tasks and commands that can be done in parallel, with a pipelined fashion and more effectively using every clock cycle with a low power demand.(Varnavas, Sims and Casas, 2015)

The newer technology available for the modern FPGAs have embedded processing systems integrated inside the FPGA allowing tedious tasks to occur more easily and flexibly.

The feature of the FPGA for comparison when choosing the ideal FPGA are provided in the Table 2.1, based on CubeSat designs.

Table 1.1 FPGA Feature Comparison

Manufacturer	FPGA	Features
Actel	SmartFusion	Up to 500 000 system gates Analogue Compute engine 8 direct analogue inputs 135 user I/0’s
	IGLOO/e	Up to 3 x 10 6 system gates Flash*Freeze mode as low as 5 micro watts. No direct analogue inputs 135 user I/0’s
	ProASIC3 nano	Up to 250 000 system gates No direct analogue inputs 71 user I/O
Xilinx	Spartan-6	Up to 150 000 Logic Cells No analogue inputs available 576 user I/O
	Virtex-6	Up to 760 000 Logic Cells Analogue inputs available 576 user I/O
	Virtex-7	Up to 2 x 10 6 Logic Cells Analogue inputs available 1 200 user I/O

Altera	Cyclone	Up to 20 060 Logic Elements No analogue inputs available 301 user I/O
	Cyclone II	Up to 68 416 Logic Elements Analogue inputs available 622 user I/O
	Cyclone III	Up to 119 088 Logic Elements Analogue inputs available 622 user I/O

This thesis will go into more depth of how the FPGA is implemented through the design and development of a proposed OBC. From the findings in this literature review with further investigation and research, the need for more detailed engineering, and development of a FPGA based on-board computers for small satellites.

2. CONCEPTUAL FRAMEWORK

This thesis will conclude with adequate research the following problem questions:

1. How does the use of FPGA technology compare to the ARM cortex MCU?
2. What is the design and development process for a FPGA based OBC?
3. How will a FPGA based OBC be advantageous in the space satellite industry?
4. Can the OBC be implemented through simulation software Matlab and Labview?
5. Can the SBC be fully tested with basic VHDL programming to the desired outcome?
6. What is the possible error to be encountered and how to overcome these errors?

3. OBJECTIVES

3.1 Primary objective

The primary object of this thesis is to design, develop and test the use of a FPGA based OBC.

3.2 Secondary objective

- Performance features should be tested.
- The implementation of programming the OBC.
- The memory bank of the proposed design.
- How the communication between the OBC and other elements of a satellite will be done.

4. RESEARCH DESIGN AND METHODOLOGY

The methodology for this thesis will be directed towards the procedure of structuring and completing.

The proposed aim is to develop an on-board computer based on the digital FPGA technology compared to that of the norm use of the ARM based MCU. The developed OBC will be more flexible and reconfigurable for the use in small satellite systems as the main OBC. Further research and study in the technology aspects of the OBC platforms used in small satellites. Focus on the architecture and design on the OBC with the chosen FPGA. Simulation and testing of the proposed structure with the FPGA will be conducted as well as hardware implementation for the various features presented by the FPGA.

4.1 Thesis constraints

Possible constraints that will be encountered:

A. The software implementation test procedures being satisfied.
B. Making use of a FPGA evaluation/dev board for the testing and simulation of the designs.
C. Implementing the design as a hardware component.
D. Available resources for further study in this narrow field.
E. Time constraints.

4.2 Thesis outline

Chapter 1 – Introduction: Brief introduction with background to the mission and related topics.

Chapter 2 – FPGA Selection: Research presented on the selection of the hardware for the OBC based on the desired FPGA.

Chapter 3 – OBC architecture and design: Design and development of the OBC structure, elements and architecture.

Chapter 4 – Software development: Discussion and relevant support to the development of the software interfaces through VHDL.

Chapter 5 – Hardware design: Detailing all the hardware components for the OBC around the FPGA.

Chapter 6 – Tests and implementation: Comprising all the test procedures and how it is implemented with the respective test results.

Chapter 7 – Conclusion and summary: A full summary of the project with a conclusion about the findings and recommendations.

5. PROJECT MANAGEMENT

This thesis will follow the work breakdown structure as detailed below.

Table 1.2 Work Breakdown Structure (WBS)

Task Name
A Field Programmable Gate Array based On-Board Computer for small satellites.
1. Project Tracking Database
1.1. Initiating
1.1.1. Define project/research objectives and constraints
1.1.2. Approximate quantifiable mission needs
1.1.3. Define deliverables
1.1.4. Kick Off Meeting

1.2.	Research planning
1.2.1.	Developing the Project Plan
1.2.2.	Scope baseline
1.2.3.	Cost Baseline
1.2.4.	Schedule
1.2.5.	Quality
1.2.6.	Review Project Plans
1.2.7.	Project Plans Approval
1.3.	Executing
1.3.1.	Assessment
1.3.2.	Design
1.3.3.	Realization
1.3.4.	Testing
1.3.5.	System Implementation
1.4.	Monitoring and Controlling
1.4.1.	Tracking
1.4.2.	Progress report
1.5.	Closure
1.5.1.	Prepare Final Documentation
1.5.2.	Draft submit
1.5.3.	Present Final Project
1.5.4.	Project Completed

CHAPTER TWO

FIELD PROGRAMMABLE GATE ARRAY

2.1 Field Programmable Gate Array

The Field Programmable Gate Array (FPGA) is a programmable digital logic device based of a matrix of configurable logic blocks (CLBs). End-users can realize desired logic functions through programming and reconfigure multiples times. The FPGA architecture is built around an array of logic blocks and interconnections that can be customised through programming switches. FPGAs consist of hard blocks such as Random-Access Memory (RAM) or Digital Signal Processors (DSP), which are widely used devices. Contemporary FPGAs are System-on-Chip (SoC) devices composed of a vast amount of soft cores and hard processors, with embedded DSP and memory, with complex configurable logic blocks. Making Field Programmable Gate Array devices are exceptionally alluring within the aerospace industry.

2.2 Variations of Field Programmable Gate Arrays

FPGAs are classified in primarily three different variations: SRAM-FPGA, Flash-FPGA and Antifuse-FPGA. Static random-access memory is a type of semiconductor memory which makes use of bistable latching circuitry to store each bit of information.

2.2.1 SRAM-FPGA

SRAM is volatile in the conventional sense that the information is inevitably dropped when the memory is no longer being fed from a power source. Each bit of information is stored on a SRAM cell which consists of four transistors which forms two cross-coupled inverters as represented in Figure 2.1. The SRAM storage cell makes use of two steady states which are being denoted by Zero (0) and one (1). SRAM-FPGAs can be incorporated into bigger scale logic and has more flexibility. The bit information in the cross-coupled inverters can be altered by radiation effects arising within the circuitry. The static memory based FPGAs among the three types are the most vulnerable to radiation (Kuwahara, 2010).

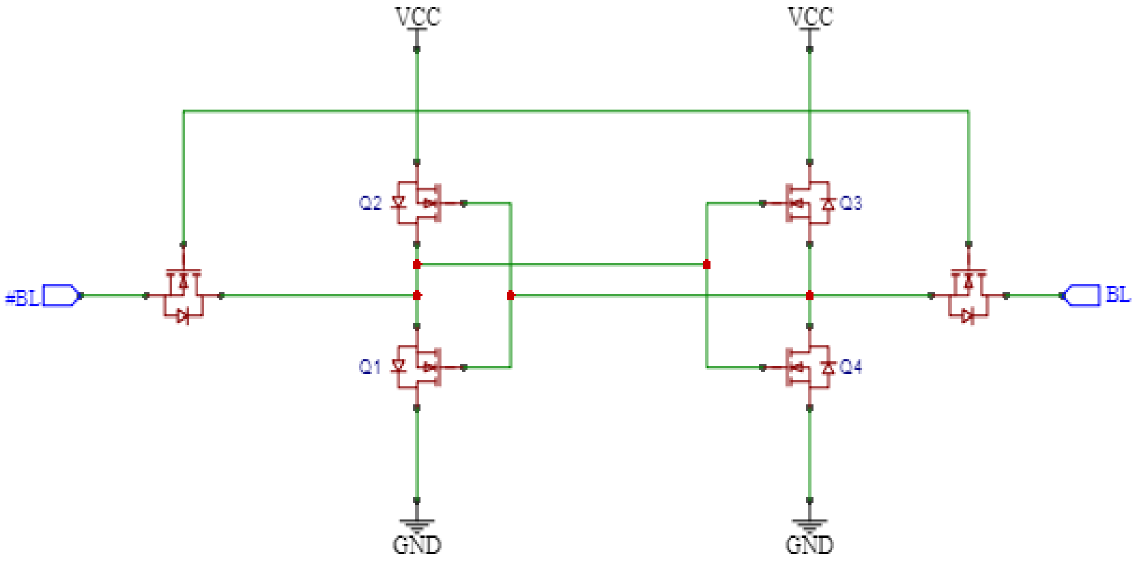


Figure 2.1: A six transistor CMOS SRAM cell structure

The second type of FPGA classified as Flash-FPGA makes use of two major components from SRAM and Antifuse-FPGA namely reconfigurability and being non-volatile. With flash-based FPGAs non-volatile memory cells store the logic pattern directly on the chip, therefore when power is removed from the logic device the logic program will remain on the flash cells. This enables significantly reduced power consumption requirements and offers faster response times making it more reliable and secure. Flash-based FPGAs are more tolerant to radiation effects.

2.2.2 Flash-FPGA

Flash memory utilizes the charge that is stored on a floating gate which makes non-volatile data storage possible. The transistor will conduct depending on the amount of charge applied. Figure 2.2 illustrates the switching element consisting of two floating gate transistors which turns the data path on or off. The program/sense transistor is used during configuration programming to perform Write/Read. The control gate and floating gate is shared by the two transistors. The sum of charge stored in the floating gate can be negatively affected by radiation effects that may cause conversion of the stored information. (1-CORE Technologies, no date)

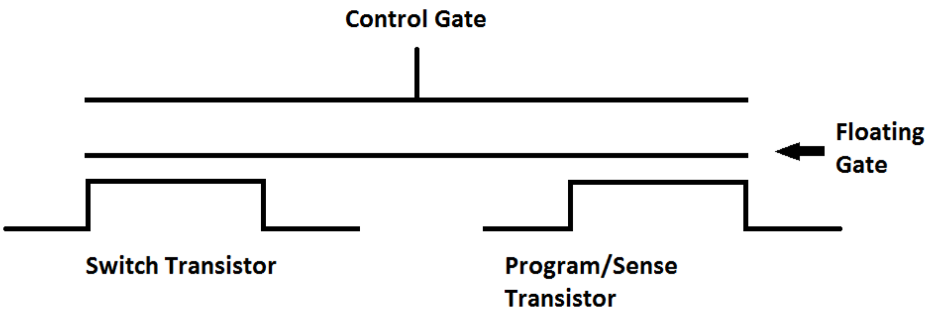


Figure 2.2: A floating-gate transistor used in flash memory (Kastensmidt and Rech, 2015)

2.2.3 Antifuse-FPGA

Antifuse based FPGAs are different from the previous two types in the sense that it can only be programmed once. The antifuse based FPGA is burned to conduct current to store the information hence the FPGA is not reprogrammable. The antifuse is highly resistive and is engineered to create a permanent electrically conductive circuit through excessive voltage. An antifuse-FPGA can be appreciated through the use of a slim layer of non-conducting amorphous silicon between the two metal conductors.

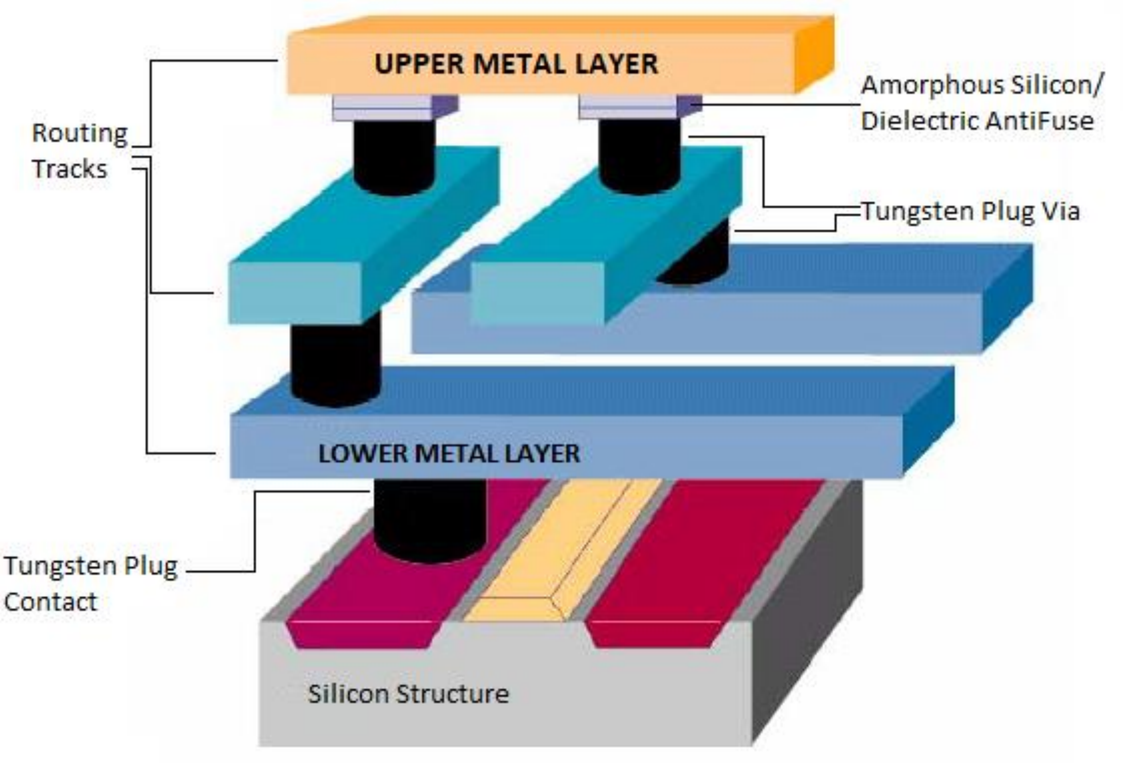


Figure 2.3: Interconnection elements of the Antifuse-FPGA Adapted (Rockett *et al.*, 2007)**.**

2.3 Radiation effects and FPGAs

Electronic components and devices are sensitive to radiation whether it may be in the outer space environment or at ground level, both will have an effect. Integrated circuits functioning within the radiation environment of space are susceptible to ionizing radiation which generates failures as the charge will alter a transistor or bit state. These ionizing fragments interacting with transistors may trigger transient and/or perpetual effects depending on the extent of charge being transferred.

2.3.1 Radiation effects

Radiation induces numerous types of effects in the space environment with the most challenging being ionizing radiation. These effects which are produced by a single event interaction which are known as Single Event Effects (SEE), that may be momentary or even permanent. This transient behaviour from a SEE is known as a Soft Error where the device is not perpetually damaged. The Soft errors include Single Event Upsets (SEU), Single Event Latch-up (SEL) and Single Event Transient (SET). An event occurs based on the amount of deposited energy. An SEU arises when the ionizing particles colliding with a transistor from a memory cell introducing sufficient charge to regress the state of the cell causing a flip in the state of the data bit. The stored information on the memory is then corrupted even though the cell can still perform the read and write commands. Diminution methods such as Triple Module Redundancy (TMR) and preferential modernization can minimize the risk of further data loss or damage. The SEL is a radiation-induced event which will inject charge that activates the feedback loop and connects V_{cc} to GND, crossed by a high current. It all leads to a disproportionate supply power which consequently causes latent failure of device functionality. SEL can be mitigated through monitoring the power consumption of the device, built in during the design when manufactured for a radiation tolerant device. Flash-FPGAs have a high probability of a SEL to occur as the reprogramming sequence requires a high voltage. SET causes transient pulses by temporarily switching driver transistors to a high state when they are supposed to be at a low state. A SET may be serious when complicated computer systems with autonomous but coordinated calculating nodes are used due to the SET having an impact on the processing speed of the internal clock.

2.3.2 Space Radiation

During the design of on-board computers for the space environment, radiation effects is a high-risk problem. Space radiation mainly comprises of different particles, namely electrons, protons, heavy ions and helium nuclei. All originating from solar particle events, novas and supernovas or cosmic rays. These particles can move freely within the vacuum of the space environment. The Earths' atmosphere is a protection barrier for these particles as they come in contact with other atoms and molecules which breaks them down causing them to loose their initial energy state. Space radiation can provoke faults and cause damage in electronic devices with a few main kinds of radiation namely cosmic rays, mesons, and alpha particles.

Galactic cosmic rays (GCR) are particles which arrive at the region of the Earth from outside the nearby solar system. Another radiation type that could lead to faults are mesons which are mainly shaped by the interaction of high-energy particles of cosmic rays with the terrestrial atmosphere. The decay of radioactive elements produces another kind of radiation known as alpha particles. These have a low energy compared to those of heavy ions.

The Van Allen radiation belts are a constant threat for orbiting spacecrafts or satellites. They comprise of electrons and ions consisting of energies in the 10-100 MeV range for the inner radiation belt, which readily penetrate spacecraft and are scattered nonuniformly within the magnetosphere. The Van Allen radiation belts occupies a compact region above the equator and is a by-product of cosmic radiation. Figure 2.4; illustrates the Van Allen radiation belts made up of two Van Allen belts which are donut-shaped regions of trapped high energy charged particles with properties similar to cosmic rays.

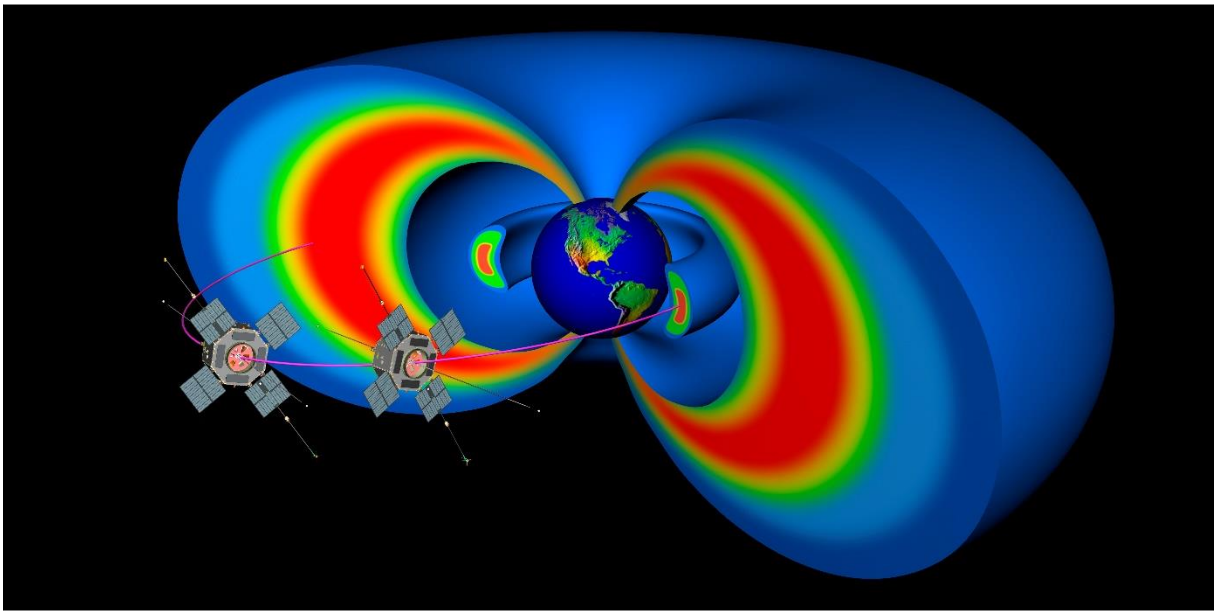


Figure 2.4: Van Allen belts illustrating the inner and outer radiation belts (VanAllen Probes 2020)

Satellite components may be damaged by the penetrating power of the high energy protons in the radiation belts. Electronic components and devices on satellites must be hardened against radiation to operate steadfastly. An object on a satellite enclosed by 3 mm of aluminium in an elliptic orbit passing through the radiation belt will receive about 25 Sievert (Sv) per year.

2.3.3 Cosmic rays

Galactic cosmic rays (GCR) are particles which arrive at the region of the Earth from outside the nearby solar system. The number and sort of cores in these particles are corresponding to those in close planetary system material. Figure 2.5 shows the energy spectrum for numerous elements that may be encountered. The total number of the curves data point in the lower energy portion of the Figure indicate that cosmic rays experience solar-cycle modulation.

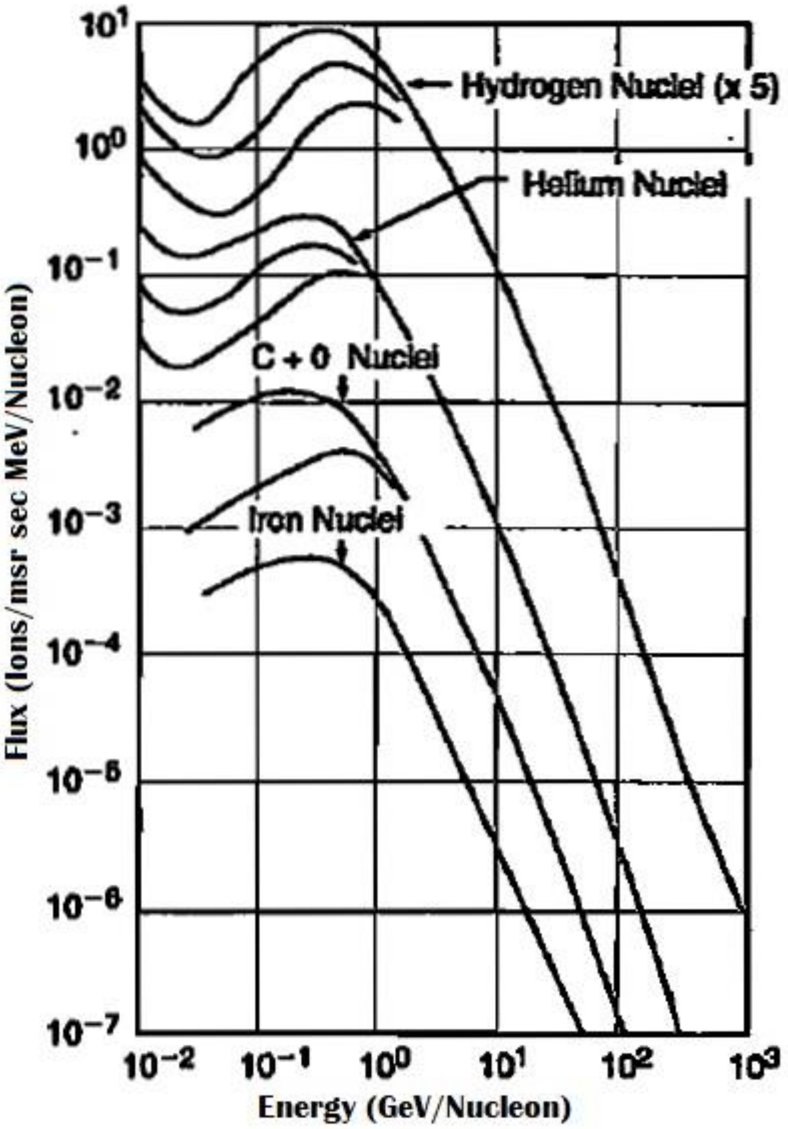


Figure 2.5: Energy spectra for numerous elements of cosmic rays(*Larson and Wertz, 1992)*

Cosmic rays represent a genuine danger on the grounds that a solitary particle can cause a glitch in like manner electronic segments, for example, random access memory, microchips, and power semiconductors.

2.3.4 Space qualified FPGAs

The space environment is an incredibly challenging environment for Field Programmable Gate Arrays. Commercially available FPGAs are more susceptible to radiation effects and other factors within the space environment than most types of electronics due to SRAM FPGAs being mainly memory devices. The most frequent error caused by radiation is bit flipping in memory devices where the configuration logic itself is susceptible to radiation and not only the bit information. Remediation methods can be applied to most types of electronic devices, such as triple-module-redundancy (TMR) on memory elements to brush the effects of radiation on registers and memories. During the past there has only been two options for the use of FPGAs in space applications. Re-programmable/configurable, high density and conventional SRAM-FPGAs which were repurposed and profoundly adapted for the space environment, and lower-density, anti-fuse, non-volatile devices designed specifically for space applications.

Xilinx space grade products are built and specified to have a recognized performance in a space radiation environment. Xilinx have the radiation-hardened space grade families of FPGAs available, namely the Virtes-5QV Family and the Virtex-4QV family.

Irrespective of the technology, all traditional devices have a substantial growth in complexity. This changes the applications of the FPGA from a dedicated function solution to an intricate system-on-chip (SoC) with impacts on the total development approach. The FPGA description is more of a system specification and is distributed in module specifications with more clear interfaces in order which decreases the complexity.

2.4 FPGA Space Applications

The use of Field Programmable Gate Arrays in the outer space environment has many advantages which include the integration, performance, efficiency and cost thereof. The FPGA allows for the integration of various digital unit functions into one FPGA. Reprogrammable SRAM-FPGAs are highly flexible, combined with high performance and complexity are progressively more important for space applications. The applications for FPGAs have evolved from simple logic into more complete subsystem platforms which combined numerous real time system functions on a single chip. The FPGAs are being more frequently used in crucial applications, missions and regularly substituting the use of ASICs.

Reconfigurable processing technology is still a comparatively new field of study for the space environment and applications. With outer space environment being diverse from earthbound techniques, the radiation will produce bit-flipping in memory devices or semiconductors can experience ionisation malfunction. These types of errors and damage to hardware systems cannot be debugged or repair, they require high reliability manufacture, assembly, and operating techniques. The utilization of run time reconfiguration in orbit will allow to alter on-board components by replacing defective designs through different phases of a mission. Large reprogrammable FPGA devices eradicate the need for functions such as phase locked loops, voltage regulators and memory as the on-board memory is adequate. Integration through this high level approach allows engineers to reduce the general system on-board power supply requirements, the cost of materials and space of the printed circuit boards (PCBs). (Sandi Habinc, 2002)

FPGAs have space legacy and coordinate effectively with payloads, on-chip memory and improved power execution as well as peripherals. Every one of these components that impact the decision of on-board computing at present. Table 2.1 shows the present status for profoundly incorporated on-board computing systems for small spacecrafts.(Staff, 2013)

Table 2.1: Examples of the highly integrated on-board computing systems adapted(Staff, 2013)

Products	Manufacturers	Processors used
Nanomind A712D	GomSpace	ARM7
ISIS OBC	ISIS	ARM9
Pluggable Socketed Processor Module	Pumpkin	C8051F120, PIC24F256110, PIC 24F256GB210, MSP430F1612, MSP430F1611, MSP4302618
MODAS	Utah State University	TI320C6713DSP
Intrepid	Tyvak	ATMEL AT91SAM9G20
Q5, Q6, Q7	Xiphos	PowerPC 405, Xilinx Spartan-6, Actel ProASIC3 Control FPGA , Xilinx Zynq 7020 ARM dual core Cortex A9, Actel ProASIC3 Control FPGA
ArduSat	NanoSatisfi	ATMEL ATmega328P
Medusa, HPD, RCC5, Athena-2 SBC, FMC Gen2/3, AIP	SEAKR	2450 DMIPS, PowerPC e500 core, Xilinx Virtex 5 FPGA, Virtex-5 FX-130T FPGAs, CCSDS, Leon Processor card, RCC4-LX160 & RCC4-LX200

2.4.1 FPGA based GPS Process

A satellite with a GPS receiver as a payload is used to determine the position of the spacecraft and used for scientific data collection. GPS satellites use a Pseudo Random Number (PRN) generator which will produce a different arrangement of bits, the Coarse Acquisition (CA) code. These resulting signals are modulated on a carrier signal before being transmitted.

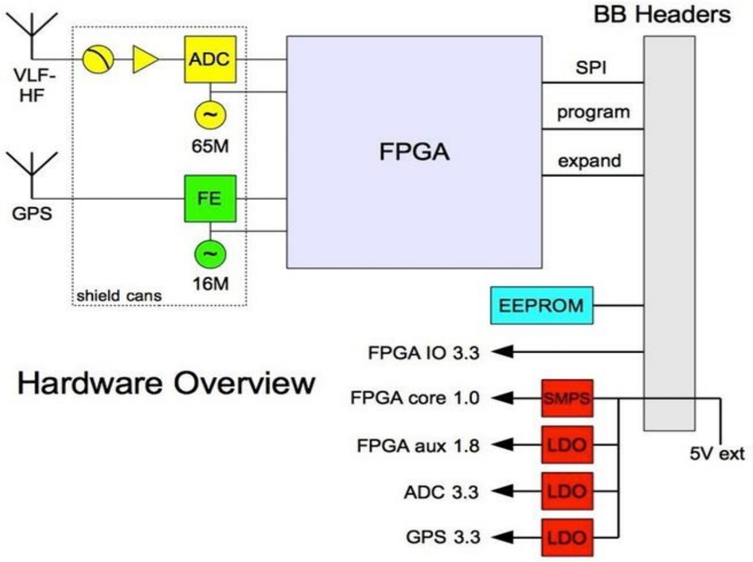


Figure 2.6: FPGA based process for GPS design (Xilinx 09 June 2020)

2.4.2 Image processing with FPGAs

Various image sensor payloads are available for the space satellite industry. The image sensor may be used as a star tracker to capture images of stars and may be used to determine the accurate orientation of the small satellite through the relative position of the stars.

The camera interface uses a Low Voltage Differential Signal (LVDS) interface that makes use of synchronized clock to that of the data packet. There are two variations of the embedded clock signal from the LVDS interface standard, namely isochronous and anti-isochronous. An anti-isochronous LVDS signal uses the start and stop bits to imply the clock.

As seen in Figure 2.7 below, the starting/stopping bits lay in the centre of the period when the pixel clock is high (Thai, 2014).

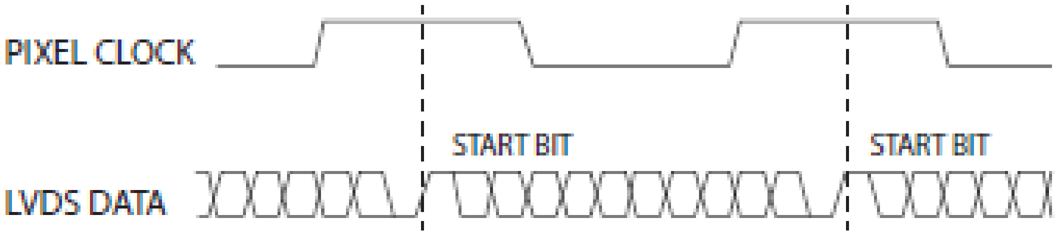


Figure 2.7: Low Voltage Differential Signal Clock Timing(Thai, 2014)

Satellites have many applications in the new space age, with Earth observation (EO) being one of the most popular missions and primary applications. EO satellites use smart image sensors to monitor and capture data from the Earth's surface, with infrared for in depth data collection. This data may then be utilized to monitor urban development, vegetation growth and natural disaster, etc. The use of satellite imaging technology constantly evolving and improving, the capture information being of higher detail with much improved resolution of the images captured. It is all achieved through the use of a wide range of spectral band cameras.(C. Hillier and Balyan, 2019)

2.5 FPGA through space

The FPGA has been utilized in space for more than a couple of decades with an assorted degree of accomplishment. Numerous issues have been experienced because of innovative reasons just as because of plan philosophy lacks. With the limit of the FPGAs consistently expanding the last issue is developing, which has been shown in a portion of the ongoing satellite developments.

With an expansion in the capability and performance of the FPGAs reasonable for space exploration are as of now slanting towards innovation for space flight including dose tolerance, SEU sensitive register with built-in TMR and low voltage requirements. The applications of FPGAs have moved from basic paste logic to complete subsystems. The utilization of FPGAs within aerospace applications is consistently expanding and creating new mission specific sectors. Where FPGAs are all the more usually utilized in critical missions and payloads, succeeding ASIC technology all the time.

Flight commendable FPGAs offer the designer critical points of interest over discrete logic, just as decrease of weight and physical board space, expanded unwavering quality and adaptability and lower cost of proprietorship.(S Habinc, 2002)

2.6 Space environment

The near-Earth space and atmospheric atmospheres strongly impact the performance and lifespan of the functioning spacecraft systems by affecting their physical size, weight, intricacy, and cost. Some natural associations additionally limit the specialized capability of these frameworks. They can prompt exorbitant glitches or even the loss of parts or subsystems.

Independently, working under vacuum-like conditions can present noteworthy issues for some spacecraft and satellite systems. When in a vacuum, most natural materials will outgas the age of misleading atoms which may go about as contaminants to different surfaces. Indeed, prior to arriving in orbit, particles and elements from the air may fall onto optical lenses and corrupt the exhibition of electro-optical instrumentation. Since there is not yet a practical method to clean shuttle surfaces as soon as the vehicle arrives in orbit, maintaining efficient contamination control during the design and production is a significant issue for most spacecrafts.(Larson and Wertz, 1992)

The electrical potential of a satellite or electronic component is measured with respect to the plasma in which it is submerged where the nett current flow is zero. That is, the following currents must cancel each other:

- The electron current from the plasma to the satellite/spacecraft.
- The ion current from the plasma to the satellite/spacecraft.
- The consequent electron current.
- The reflected electron current.
- The photoelectron current.

The voltage at the part's surface additionally relies upon the material's capacitance as for the encompassing materials, particularly the vehicle ground.

Since materials have shifting optional outflow constants and photoelectron flows, the equilibrium possibilities additionally contrast. An electrostatic arc happens if the electric field surpasses the interruption field along the outside of the material, through the material, or between neighbouring materials. Electromagnetic interference (EMI) from such arcs can make satellites to work inconsistently.(Larson and Wertz, 1992)

CHAPTER THREE

OBC ARCHITECTURE AND DESIGN

3.1 Overview

The third Chapter discusses the details around the architecture of on-board computer systems used in the space satellite industry. Focus will be on the hardware used for the main processor of the OBC as well as the design structure. The architecture of the FPGA is discussed briefly in this Chapter.

3.2 Basic On-Board Computer systems

The OBC is the main component of a small satellite, essentially being the heart of the whole system. A space satellite is comprised of a variety of various subsystems which all need to work in conjunction with each other as an integrated system. Aerospace architectures has a major part in creating this conceivable by associating the different modules together and devices or components from different manufactures. All these modules with various functionality can be integrated on the entire satellite which is managed and operated effectively. The capabilities of the on-board computers dictate the configuration of the rest of the satellite to an exceptionally large extent.

The on-board computer performs functions in satellites like those of the central nervous system in any living organism. They interface with various other subsystems and sensors to ensure the satellite's survival. In space idiom, the on-board computers are at the very core of what is called the `Satellite Bus', which is essentially a minimal system which can be reused in multiple spacecrafts with various kinds of payloads. The capability of selectively reconfiguring the on-board computers in subsequent spacecraft without adversely affecting the performance of the satellite bus is one of the most important factors defining the viability of a given satellite bus.(Desai, 2010)

The OBC control all the major functions for effective operation therefor being a major role in the satellite which controls all the functions, namely:

- Attitude and orbit control
- Telecommunication data management
- Telemetry data management
- On Board time management
- Autonomous Reconfiguration

- System housekeeping
- Interfacing with other subsystems

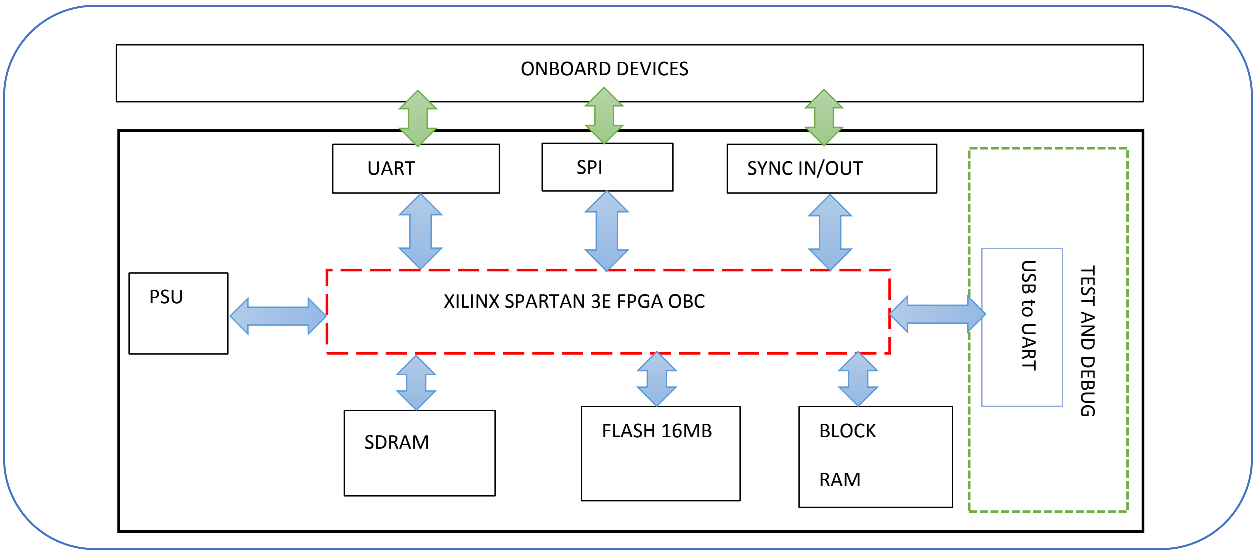


Figure 3.1: Proposed OBC interface layout

The On-board computer mainly consist of a microprocessor, memory banks and the interfacing system. Most reliable OBC modules readily available on the market for small satellites or CubeSats are microcontroller based. The use of an ARM based microcontroller for the main processor of the OBC is a very favourable within the OBC sector. Other manufacturers make use of Atmel's LEON3FT microprocessor which is used in Low-Earth Orbit missions and deep space exploration.

Spaceteq manufactures a high-dependable, low-powered flight computer for Low Earth orbit satellites and spacecrafts using the radiation-tolerant Cobham Gaisler GR712RC processor. The CR71RC is a dual-core LEON3FT processor with advanced protocols. Figure 3.2 illustrates the basic interface structure of a nanosatellite/small satellite to the OBC. The layout is based on the mission application for the satellite and the various payloads.

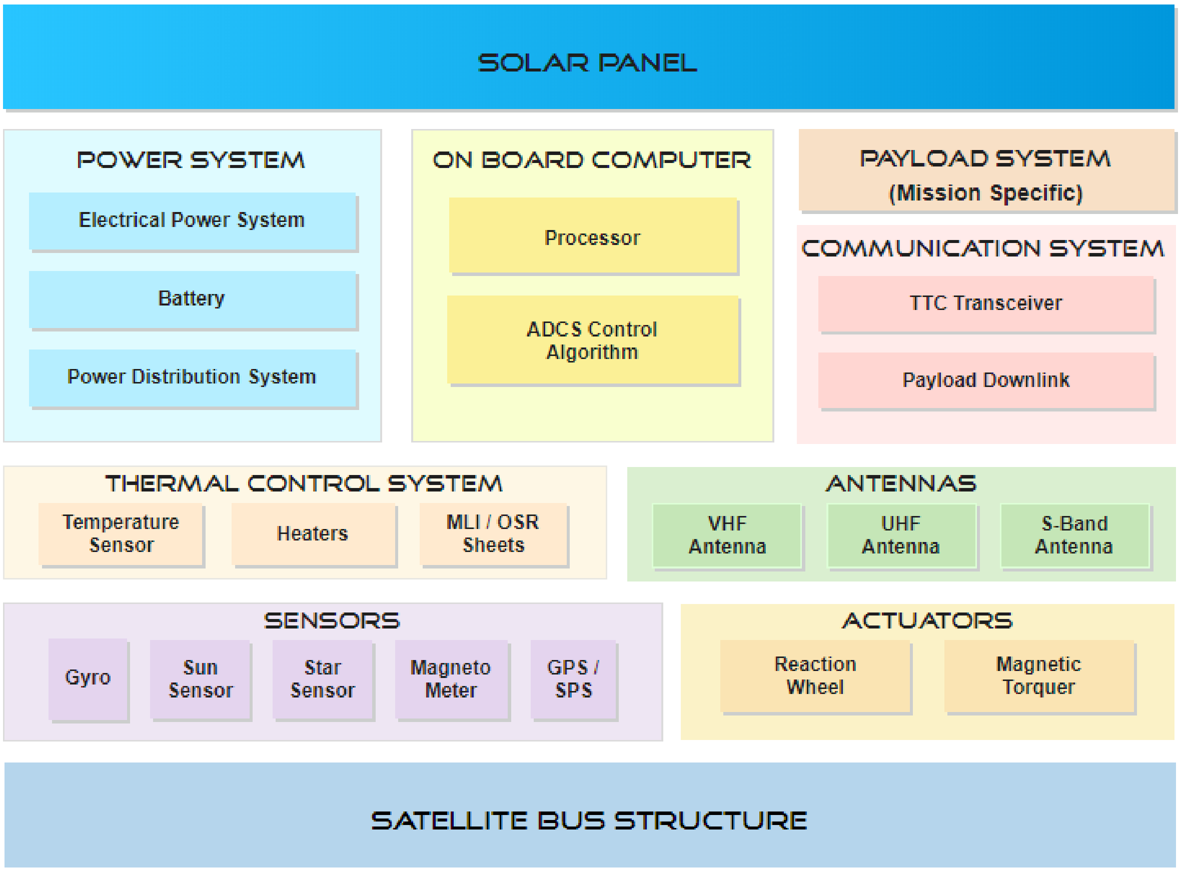


Figure 3.2: NanoSatellite Structure

FPGAs are programmable and reprogrammable electronic devices that depend on a network of configurable logic blocks (CLBs) associated through programmable interrelates to Application-specific integrated circuits (ASICs). ASICs has a permanent design dependent relative to the application, which can't be altered. FPGAs are reconfigured to adjust or enhance the structure as per the intended design or application prescribes.

The FPGA is an electronic component that is assembled complete, yet the remaining parts structuring subordinate permitting adaptability in actualizing designs concepts. Various FPGA manufacturers design a reconfigurable architecture, for instance dependent on CLBs, logic cells (LCs) or logic elements (LEs), provided by Xilinx and Altera, respectively. Additionally, these FPGAs incorporate hard blocks such as random access memory (RAM) or digital signal processor (DSP), while their amalgamation enhances the asset usage and maximum operating frequency.(Tlelo-Cuautle, Rangel-Magdaleno and de la Fraga, 2016)

Figure 3.3 illustrates a streamlined structure of a Xilinx FPGA. It is clear that in general there are three major types of structures that are required as follows:

- Programmable/re-programmable Interconnect
- Input/Output blocks
- Logic blocks

Currently, depending on the FPGA's family, its architecture may include a variety of hardware components that are directly integrated into the FPGA material.
These range from embedded multipliers and/or DSP blocks, RAM blocks, digital clock managers, phase-locked loops, soft-core processors, and intellectual property cores.

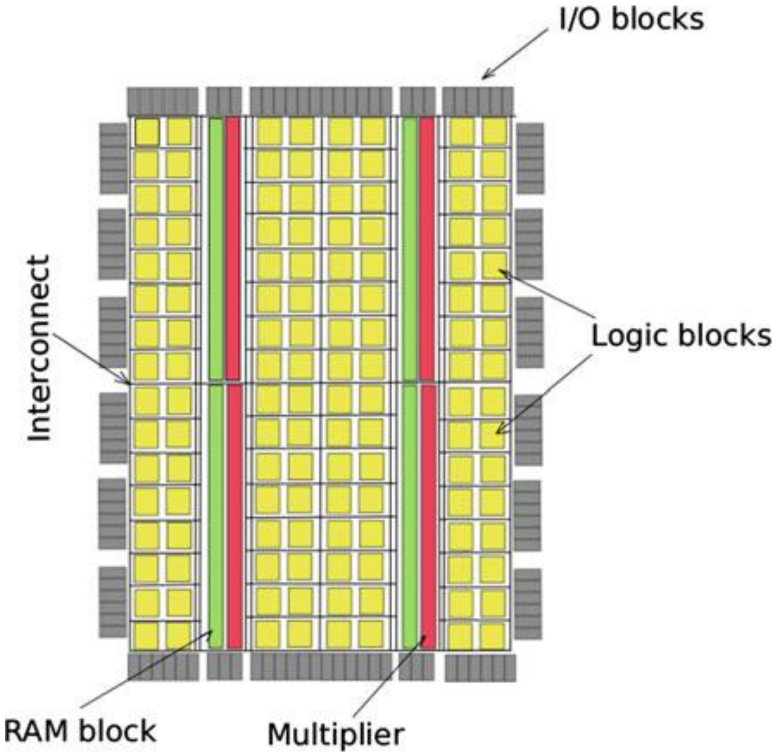


Figure 3.3: Simplified Xilinx FPGA structure

3.2.1 Selection criteria for FPGA

The arrangement of necessities for the ideal equipment is of high essentialness in the design process and final operational requirements a small satellite. These requirements that are to be set by the end user will define the OBC architecture. The process of choosing an FPGA is like any other decision-making process. It’s also necessary to ensure correct data processing and flow in a real-time basis.

Requirements for the OBC is defined for a specified mission or application of the satellite. As there is no intended mission at hand, the requirements for this OBC design is to be flexible to common requirements of industry used OBC platforms. It is possible to set common attributes apart from the current available OBC hardware available for small satellites within the space industry which are more widely being used and implemented based on the tabulated information presented in Table 3.1

Table 3.1: FPGA Package type operating comparison

Series	Operating Voltage	Operating Temprature	I/O Pins	Package Type	PRICE USD
MAX II	1.8 V to 3.3 V	-40°C to +100°C	80	TQFP	-
CYCLONE II	1.8 V to 3.3 V	-40°C to +100°C	89	TQFP	-
SPARTAN 6	1.8 V to 3.3 V	-40°C to +85°C	89	TQFP	-
SPARTAN 6	1.8 V to 3.3 V	-40°C to +100°C	200	TQFP	7,39
SPARTAN 3E	1.8 V to 3.3 V	-40°C to +100°C	68	TQFP	22,68
MAX 10	3.3 V	-40°C to +100°C	101	BGA	9,47
CYCLONE 10	1.2 V	-40°C to +125°C	176	BGA	11,22
Microsemi	1.5 V	-40°C to +85°C	49	QFN	8,35
Microsemi	1.5 V	-40°C to +85°C	77	QFP	10,06
Microsemi	1.5 v	-40°C to +85°C	71	QFP	11,49

The use of commercial grade electronics requires the on-board computers to tolerate errors caused by SEUs (Single Event Upsets) and other forms of radiation damage, power brownout, and malfunctioning peripherals.

The prerequisites of the OBC are characterized as follow:

- Reconfigurable computing through optimisation of SRAM_FPGAs on small satellites.
- The centralized OBC will function as an SBC with abroad range of elements.
- Compatible with most or all payload systems and peripherals.
- Capacity to direct all the necessary subsystem functions.
- Capable of mathematical calculation dependent on fixed point figures.
- Rapid data communication for Ka-Band interchange up to 500 Mbps.

3.3 Architecture for OBC systems

The global hardware architecture of a spacecraft has a standard architecture as shown in Figure 3.3 below, detailing all the required sections and blocks that interact through the satellite system. The reason for following this architecture configuration is based on less complexity with regard to wiring among the space division where it will not necessitate a large amount of communication ports devoted to the communication between satellite subsystems. Malfunction of the OBC or within its subsystems will not result necessarily in the malfunction of the complete system, and failure of communication with the satellite's main systems. The seclusion at the degree of equipment and programming of the framework structure, which is reasonable for the association of the undertaking of the project. Whereas the subsystems will be built more or less autonomously by the project collaborators. Unbiased simulation and testing of subsystems is possible.(Hanafi and Latachi, 2017)

The FPGA electronic device being utilised within the payload subsystem has the advantage of high density, performance with additional flexibility and minimal fees in the design and development execution. Keeping in mind the design strategy must take into reason the effects of the ionizing radiation exposure within the space environment. Ideally the developing system using COTS FPGA devices and technologies available to the commercial market, researchers have developed SEE mitigation techniques based on hardware redundancy.

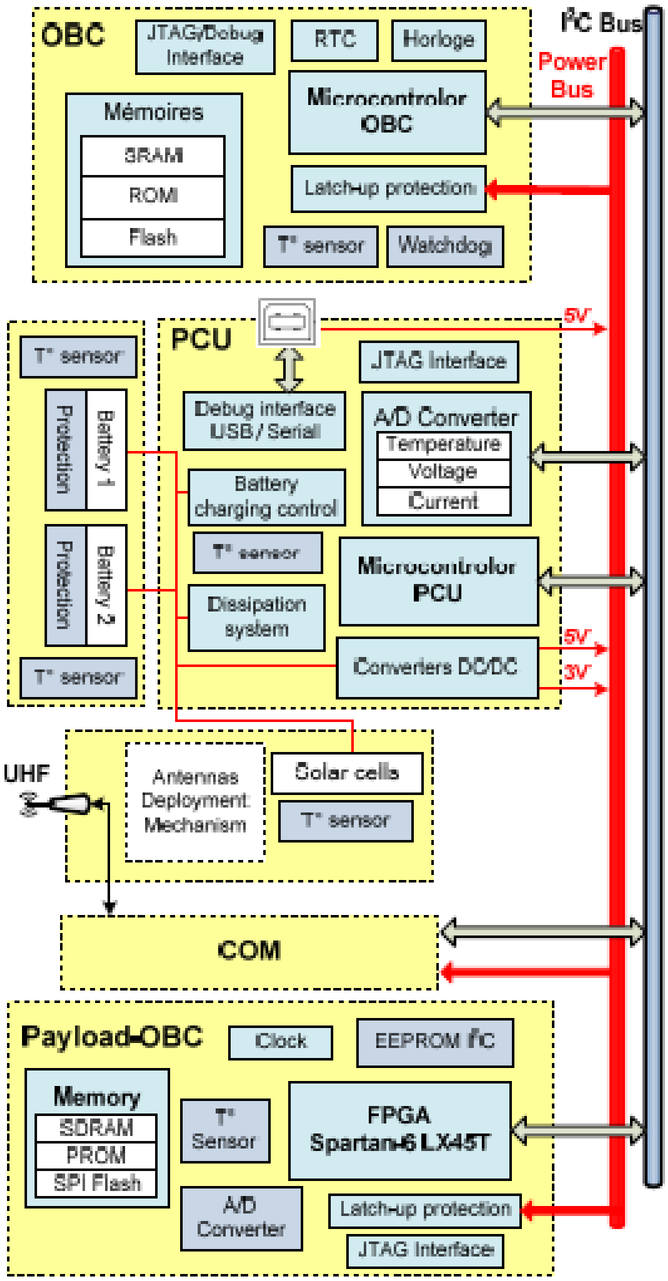


Figure 3.4: Basic architecture of an OBC system

3.4 OBC FPGA Selection

Defining which specific requirements are applicable to a space grade OBC compared to that of a standard computer system on Earth. The OBC will have to fulfil a specific criterion to make it usable for space applications. In the section bellow it gives a brief overview of the essential requirements for the FPGA used as the main processor of the OBC.

An on-board computer can transmit commands and supervise telemetry over a single interface with the command and data handling system, permitting the control of numerous satellite subsystems and payloads. The C&DH system's size is directly proportionate to the intricacy of a small satellite. As the systems increase for a satellite mission or it's payloads, the additional monitoring, control and configuration capacity is necessary. Consistency concerns alone may expand the hardware's size if we require superfluous C&DH subsystems.(Larson and Wertz, 1992)

A development board is used to do a hardware-in-the-loop (HIL) simulation with the onboard Xilinx Spartan 3E FPGA which is detailed and presented in Chapter 5 of this thesis.

3.4.1 Power Requirements

The electrical power subsystem (EPS) provides, distributes, and controls the satellite electrical power as illustrated in Figure 3.5 bellow.

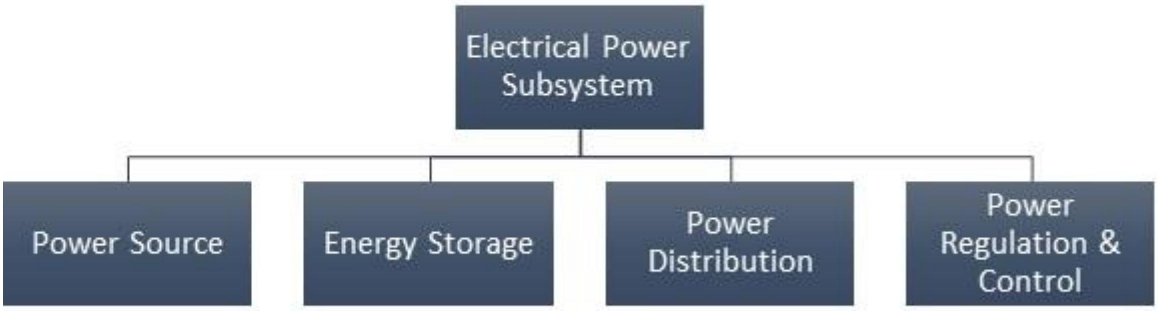


Figure 3.5: Function block diagram of an typical EPS (Adapted SMAD)

A sufficient power structure is a major necessity in satellite systems since the electrical power is tremendously limited, predominantly in CubeSats/Small Satellites. With regard to a low power consumption statistic, the OBC ought to oblige an element whereby power can be spared by crippling peripherals which are not being used during a specific period of the mission.

Comparing the power consumption between FPGAs may be challenging due to the various designs and payload requirements. The main power consumption comes from the Input/Output (I/O) pins, clock speeds and the active components of the FPGA. The idle power requirements are stated on the datasheets of the various FPGAs, which is mostly average overall.

Flash-FPGAs use less power compared to that of the SRAM-FPGAs which are orders of magnitude and require more power in the range of milli-watts as to micro-watts. However, with lower clock frequencies required for operation as to conventional processors, the FPGAs are able to offer power savings. Less power is consumed due to less transistor switching, hence lower clock speeds.

3.4.2 Environment and operating temperature

The discrepancies in temperature at Low Earth Orbit (LEO) are overly critical. At the point where a CubeSat/small satellite orbits the sun side of the Earth, the outside of the small satellite/spacecraft gets extremely warm and comes to temperatures of up to 150°C. Correspondingly, when encompassing the zone of overshadowing, the temperature can plunge to an extreme low temperature of -150°C.

These extraordinary fluctuations in temperature can produce harmful outcomes to electronic hardware components inside a satellite systems or spacecraft when COTS components are used and may cause failure in various aspects of the mission.

Therefor the electronic hardware components that are used for the OBC of a satellite should be radiation hardened and/or have an functional temperature range of between -40°C and +85 to +100°C which is the standard for modern industrial electronic hardware and commercial equipment.

3.4.3 Operational Voltage

The Electrical Power Supply (EPS) of a satellite is composed of three modules which are the Power Control Circuit (PCC), the photovoltaic panel (PV) and the battery. An unregulated bus voltage varying in voltage is generated by the EPS. Without various voltage regulars for the electronic components, it saves weight and cost on the overall hardware of the satellite. The best practise will be to select components that operate in the same voltage ranges.

Space graded IC components such as the SPACE IC POL (Point-of-Load) convertor have been treated to mitigate Single Event Effects (SEE) radiation as well as have been Total Ionization Dose (TID). This synchronous buck regulator is radiation hardened and can operate with input voltages up to 24 V. Figure 3.6 below exemplifies the typical application of the POL convertor.

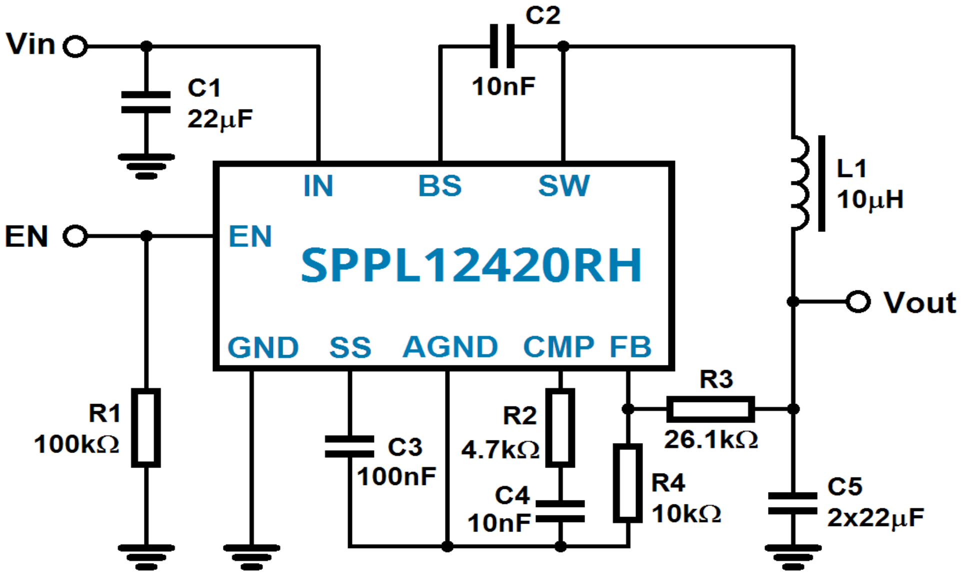


Figure 3.6: Typical Schematic circuit of an application

3.4.4 Component packing

By taking into account the physical size and weight of the proposed electronic components. The physical packaging of components such as integrated circuits (IC) and all other various parts, will have an influence on the general size and the OBC subsystem weight. These packages also indicate which type of industry the FPGA package is designed for, and how it is identified.

As appeared in Figure 3.7, there are three variations of the Xilinx Spartan 3E family packages are available, contingent upon the application for the OBC designed during this thesis. Inclination is given to surface mount technology compared to dual in-line packages because of volume. Besides, on the physical packaging for ICs, small-outline integrated circuit and quad flat package (QFP) are favoured over ball grid array (BGA) and land grid array (LGA) for simplicity of joining through the soldering/reflow procedure.(LWABANJI TONY LUMBWE, 2013)

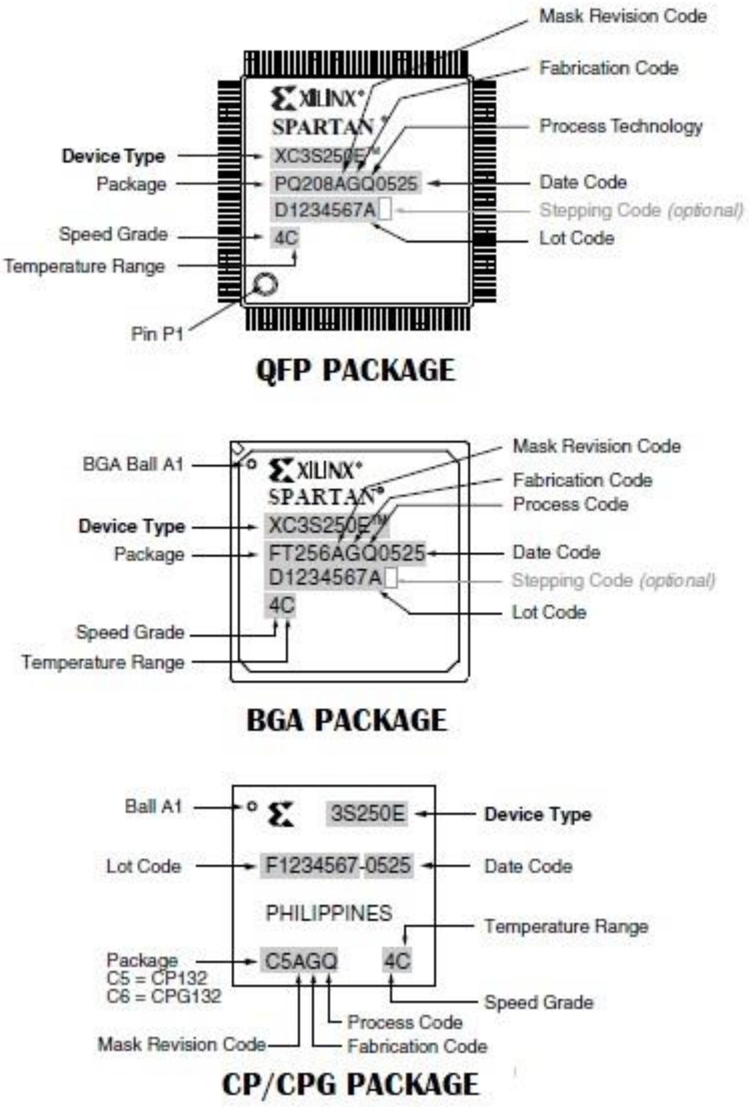


Figure 3.7: Xilinx package types adapted

Based on the availability and feasibility for this thesis, the Xilinx Spartan 3E family is used to implement the FPGA as the main processor on the proposed OBC design. The Spartan 3E FPGA is available in the TQFP package with temperature ranges of -40°C to +100°C, which is suitable for low Earth orbit satellite applications. Based on the small form factor, the FPGA device does not require much physical board space, allowing to incorporate an on-board ESP and external flash memory. The board is designed to comply with the PC/104 form factor used in the small satellite industry and is compatible with other satellite components and payloads.

The table below, Table 3.2, shows the different FPGAs to was considered for this thesis as an experimental OBC board at low cost with most support and readily available.

Table 3.2: FPGA selection table for the proposed OBC design

Series	Operating Voltage	Operating Temperature	I/O Pins	Package Type
MAX II	1.8 V to 3.3 V	-40°C to +100°C	80	TQFP
CYCLONE II	1.8 V to 3.3 V	-40°C to +100°C	89	TQFP
SPARTAN 6	1.8 V to 3.3 V	-40°C to +85°C	89	TQFP
SPARTAN 6	1.8 V to 3.3 V	-40°C to +100°C	200	TQFP
SPARTAN 3E	1.8 V to 3.3 V	-40°C to +100°C	68	TQFP
MAX 10	3.3 V	-40°C to +100°C	101	BGA
CYCLONE 10	1.2 V	-40°C to +125°C	176	BGA
Microsemi	1.5 V	-40°C to +85°C	49	QFN
Microsemi	1.5 V	-40°C to +85°C	77	QFP
Microsemi	1.5 v	-40°C to +85°C	71	QFP

3.5 I/O peripherals and serial bus compatibility

The assortment of digital and analogue I/O pins accessible on the FPGA are essential as far as connectivity and integration. Having the option to interface the FPGA OBC system with all associated subsystems inside a small satellite is compulsory and is legitimately identified with the FPGA picked to execute the design for this OBC. The data transfer between subsystems is made conceivable by the OBC's FPGA’s incorporated serial interfaces through UART and I^{2}C. Choosing a FPGA package which has at least one of every one of the serial interfaces is enthusiastically suggested since these channels are hampered by their data transfer rates with the address bus sizes.

3.5.1 Memory

SRAM memory gives exceptionally quick random admittance in either writing and/or reading modes. In a run of the conventional small satellite environment, the operating system root and some various procedures will be duplicated from flash memory into the static memory where it will be stored during typical operation. Random access writing and/or reading commands are possible at higher speeds. The Atmel AT45DB161E flash memory is used on the FPGA OBC for program storage and re-programmable. The algorithm for programming the entire array randomly as displayed in Figure 3.8. For safeguarding data integrity, each bank of the Atmel

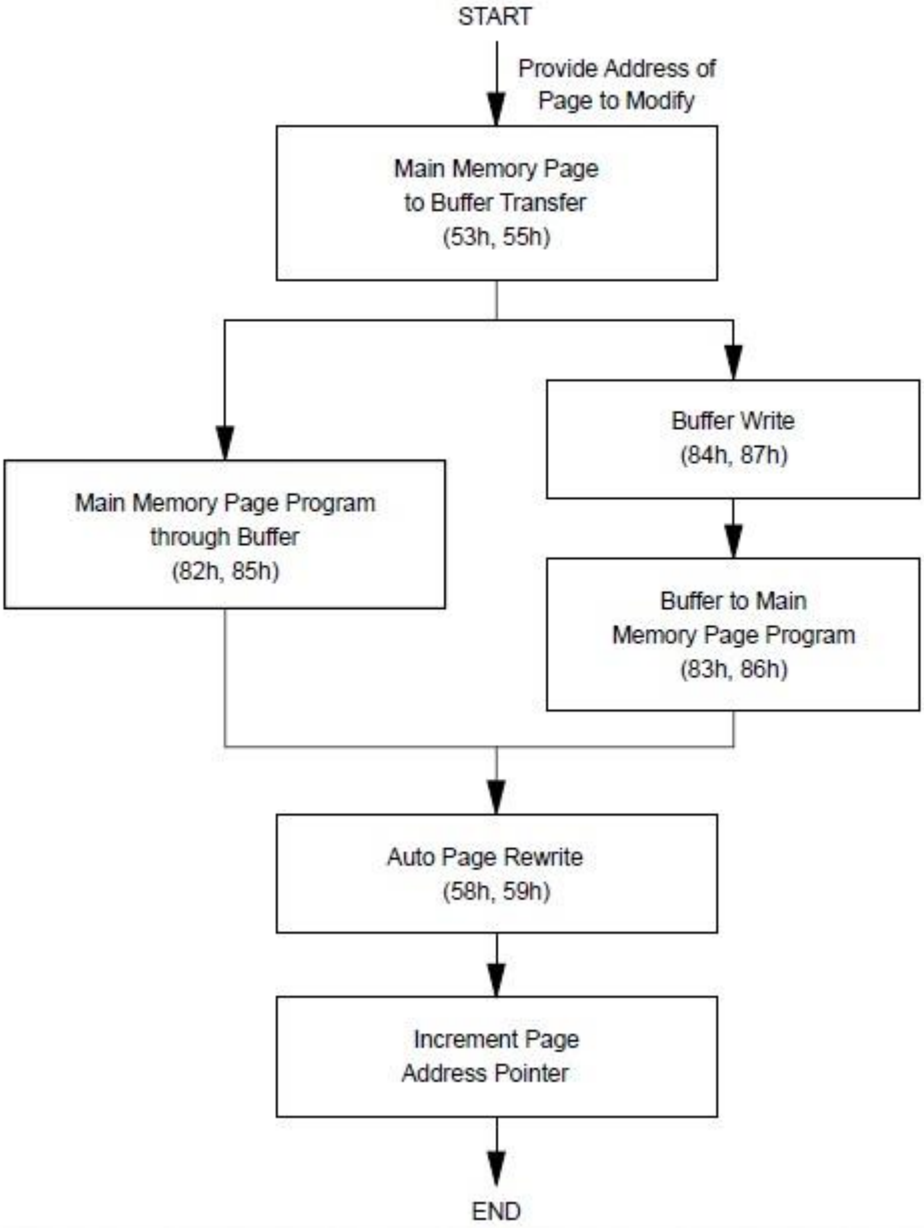

START
Provide Address of Page to Modify
Main Memory Page to Buffer Transfer (53h, 55h)
Buffer Write (84h, 87h)
Main Memory Page Program through Buffer (82h, 85h)
Buffer to Main Memory Page Program (83h, 86h)
Auto Page Rewrite (58h, 59h)
Increment Page Address Pointer
END

Figure 3.8: Atmel program or re-program algorithm

DataFlash sector must be revised at least once within every 20 k collective bank erases and program operations.

The type of storage is dependent on the application at hand. It is during storage that errors are most likely to arise. Errors usually occur when radiated particles penetrate the memory cells contained within the RAM. These type of errors are defined as bit flips in the memory.(C. P. Hillier and Balyan, 2019)

3.5.2 Previous implementations

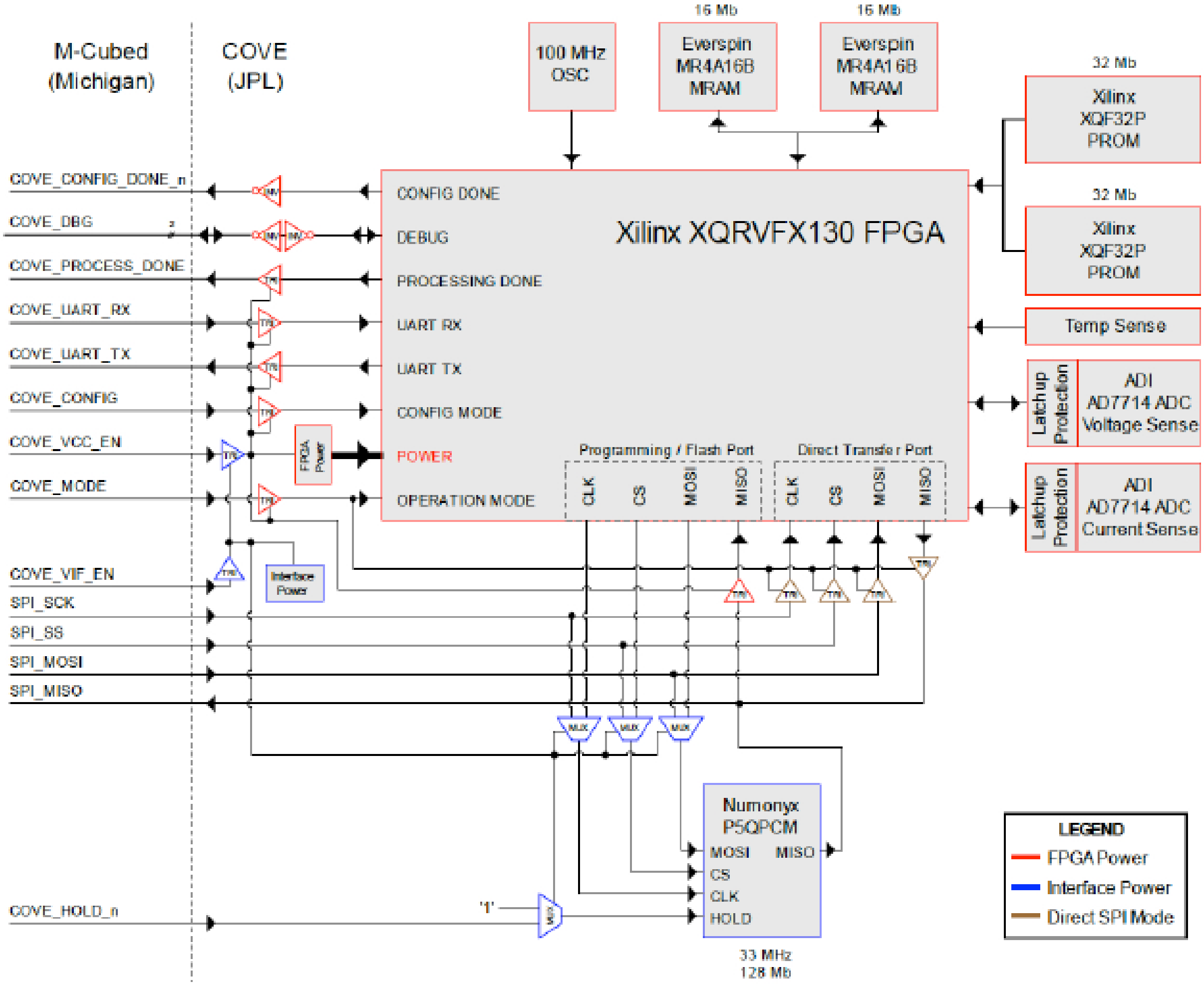


Figure 3.9: Block diagram of the COVE by NASA-JPL

The CubeSat On-board processing Validation Experiment (COVE), board is structured with the aim to limit power utilization while offering the full information handling ability of the Xilinx V5QV FPGA. This is accomplished by specifically driving the interface on/off on the FPGA.

The COVE is a payload experiment that will prove that an image processing algorithm designed for MSPI, Multiangle Spectro polarimetric Imager, utilising the first in-space application of a radiation hardened Virtex-5QV FPGA manufactured by Xilinx. The Figure above shows the general outline of the COVE with the Xilinx FPGA as the main processor.

CHAPTER FOUR

SATELLITE AND SYSTEM MONITORING

4.1 Overview

This Chapter will discuss the details in the simulation and verification concepts of the developed algorithms being controlled. The hardware development of the OBC is hardly done without providing any software and/or firmware support as well. Software and hardware tests are intended to verify the successful operation of the OBC design. It is required to have proper debugging tools to support the hardware when it comes to complex process.

The OBC architecture is based on the general flexible structure from Xilinx as there is no specified mission at hand during this thesis. General operations of an OBC for small satellites will be simulated on MATLAB and Simulink.

A development board is used to do a hardware-in-the-loop (HIL) simulation with the onboard Xilinx Spartan 3E FPGA which is detailed and presented in Chapter 5 of this thesis.

4.2 Monitoring the satellite

During orbit in the rough space environment, it is essential to monitor the state of the systems and sub-systems incessantly. With any abnormalities being detected by the OBC caused by faulty electronic components, can be noted to be isolated before any damage is caused to the mission or it's sub-systems. These sub-systems of the small satellite, including the OBC, are sensitive elements of the satellite. Therefor the monitoring process must be responsive and accurate to pick any minor changes that may cause future errors.

4.2.1 Temperature monitoring

An on board temperature sensor used to monitor the OBC boards' temperature during orbit. Other critical boards and payload may also have their own on board temperature sensors. With the use of space grade components, it is possible to mitigate some of the problems and error that may be encountered in the space environment.

Monitoring the temperature of the satellite is important to determine the thermal state for all the subsystems. Thresholds are set with the software to command the attitude determination and control system (ADCS), to turn the satellite toward the sun or away from the sun's heat.

Same with the switching of the power supply to boards are used to prevent any thermal damage that may occur during orbit due to external elements.

4.2.2 Power system monitoring

The power monitoring system is implemented with the use of voltage sensors and current sensors.

Mostly used for telemetry purposes and general monitoring. The total power consumption can be calculated with the data received from the voltage measurements to determine the total power consumption of the OBC and other subsystems/payloads.

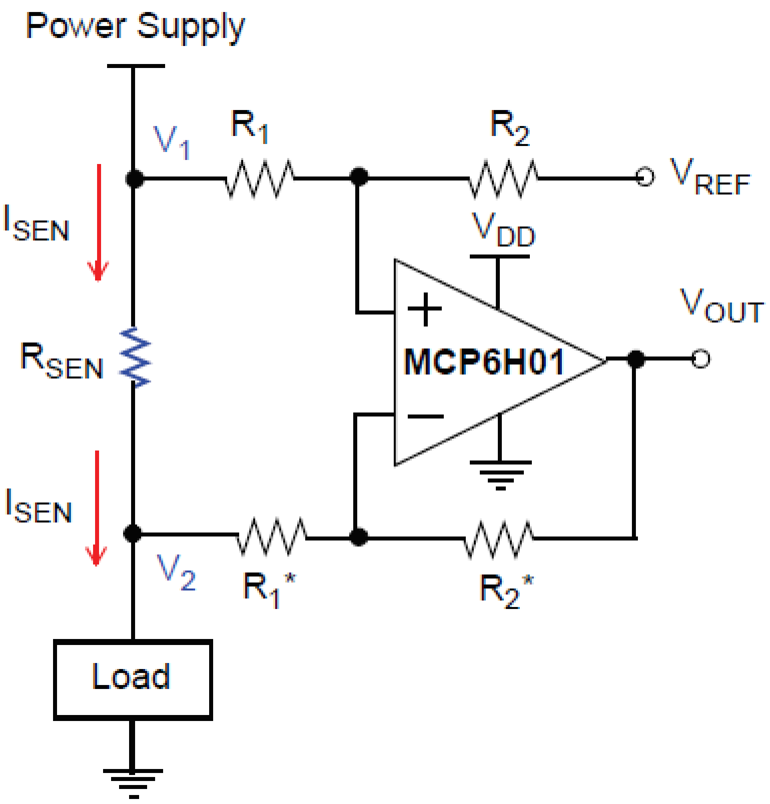


Figure 4.1: High side current sensing application (Sensing and Concepts, 2011)

4.2.3 Performance monitoring

The successful mission operation of a satellite depends on the performance of the OBC. In most small satellites there are independent platforms for command and data handling, telemetry, and payload operation. All these operational requirements can be done with the use of a single on-board computer.

4.3 Architecture of OBC FPGA

The Xilinx Spartan 3E family offers the lowest cost-per-logic for any of their FPGA families which eases system costing. Through establishing the most feasible configuration solutions including product serial and flash memory parallel integrating the functions of various chips into a single FPGA.

As for the architecture of the Spartan 3E family it is made up of five essential programmable functional elements. The Configurable Logic Blocks (CLB) contain flexible Look-Up Tables (LUT) which implement logic with storage elements being used as flip-flops or latches. CLBs are capable of performing a range of logical functions as well as the storage of information. The second fundamental programmable element is the Input and Output Blocks (IOB). These regulate the flow of bits of data from the I/O pins and the core logic of the FPGA chip. For every IOB, they support bidirectional data flow with three-state operation. These IOBs support a number of Double Data-Rate (DDR) registers as well as signal standards.

Block RAM is the third element to the Spartan 3E family's architecture, which provides data storage in the form of 18 kb blocks. Blocks that are multipliers is an addition to the Spartan 3E which accepts two 18-bit binary numbers as inputs and calculates the product. The last element of the family's architecture is the Digital Clock Manager (DCM) that provides the autonomous calibrating, delaying, multiplication and phase-shifting clock signals. Figure 4.2 bellow shows these elements as seen in the architecture with a circle of IOBs surrounding a usual array of CLBs (Information *et al.*, 2009).

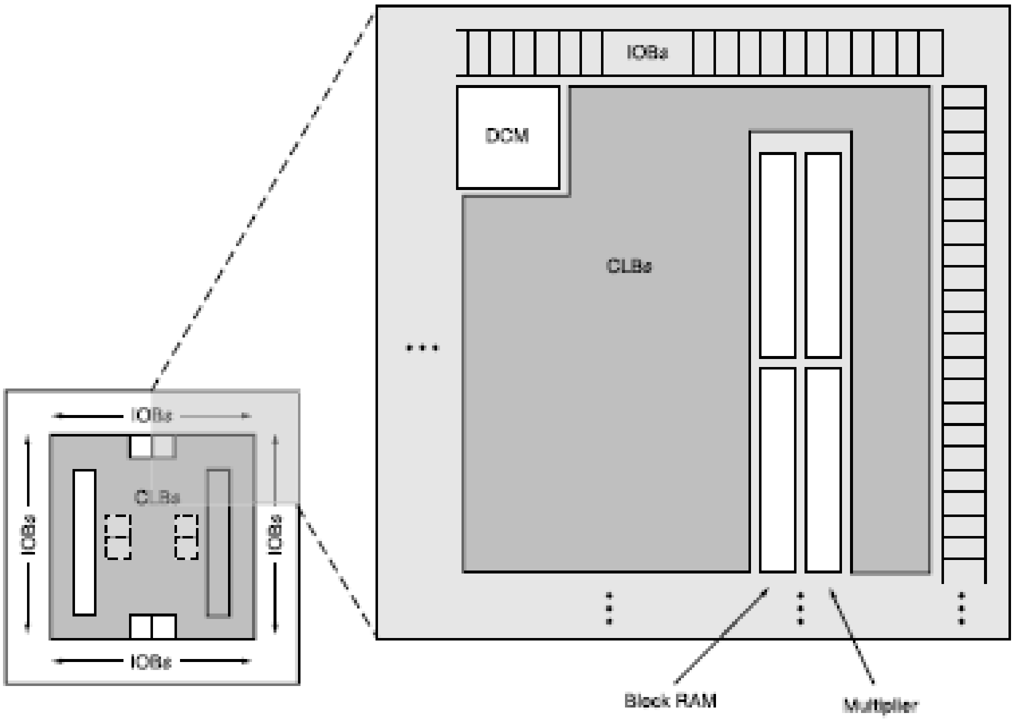


Figure 4.2: Xilinx FPGA Spartan 3E family architecture with IOBs (Information *et al.*, 2009)

Each CLB contains four slices which is made up of two LUTs. Each slice can implement logic functions and may be configured as 16x1 of RAM memory or 16-bit shift registers. In these slices there are two storage elements to implement flip-flops or latches.

The IOB providing a programmable bidirectional interface between a package pin and the FPGA's internal logic. With a number of main signal paths inside the IOB namely the output/ input paths and the 3-state paths. The input path carries the data from the I/O pin, through an voluntary programmable delay element directly to the input line. The output path, O_1 / O_2, carries data from the FPGA's internal logic through a multiplexer and then a 3-state driver to the IOB pad. Figure 4.3 shows the streamlined diagram of the internal structure of the IOB.

High impedance is determined by the 3-state path. Lines that transfer data from the internal logic through a multiplexer to an output driver as labelled as T1 and T2 in Figure 4.3. An inverter option is available on all the signal paths entering the IOB.

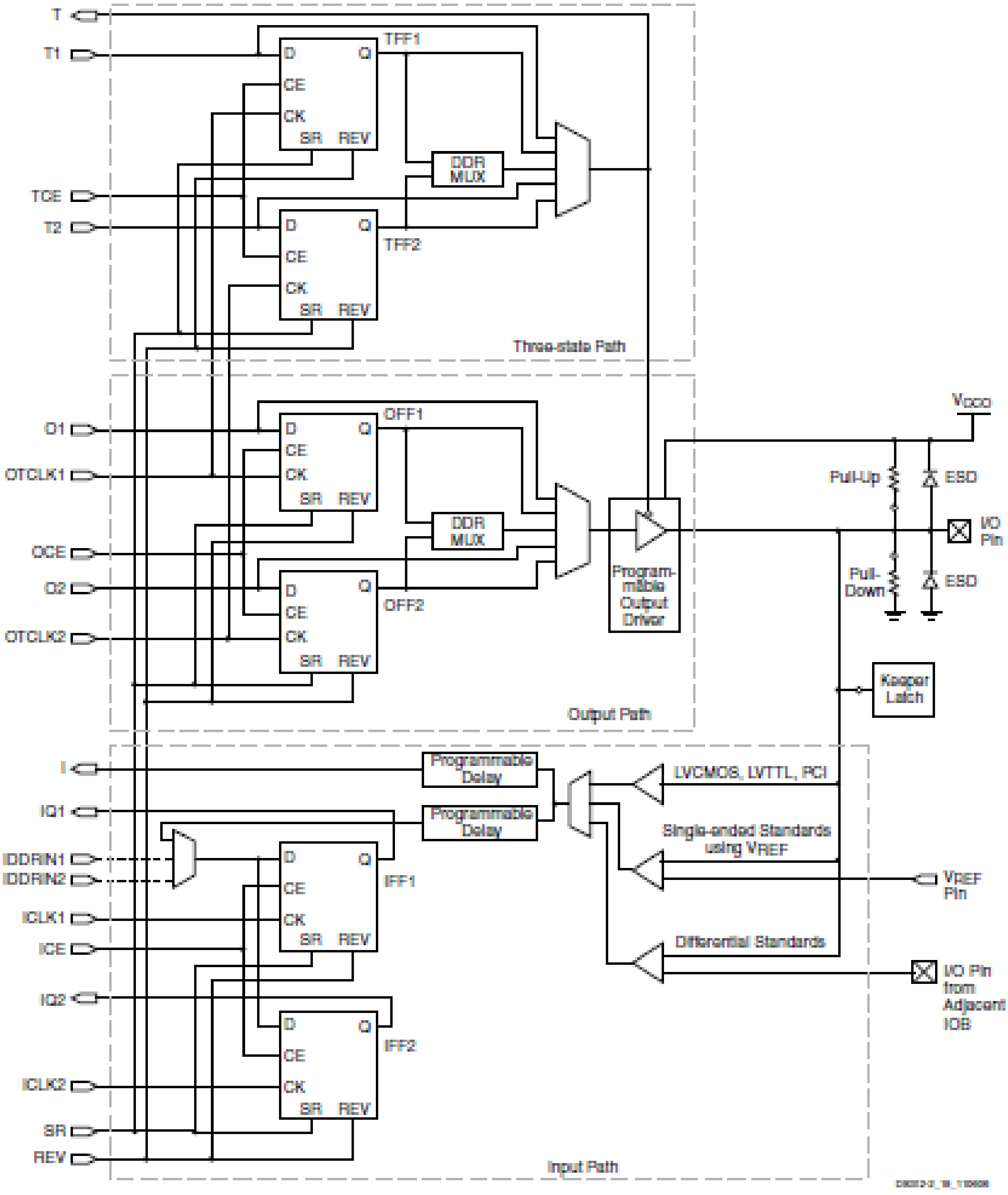


Figure 4.3: Xilinx FPGA simplified IOB internal structure (Information *et al.*, 2009)

4.4 FPGA Hardware-In-The-Loop Simulation

The Field Programmable Gate Array based onboard computer for small satellites is tested and verified through a FPGA-in-the-Loop (FIL) simulation. Simulations such as Hardware-in-the-Loop (HIL) and the FIL is done using the MATLAB and Simulink environment.

For the simulations, a development board from Digilent Basys 2 FPGA board will be used with the same Xilinx FPGA family as the OBC designed in Chapter 5 to follow. This board is designed around the Xilinx FPGA Spartan-3E family. The Xilinx FPGA structures a 500 K gate Spartan 3E FPGA with a 32 bit processor and DDR interfaces, the XC3S500E FPGA.

This board is used for the HIL simulations to validate the operation of the tests for the FPGA, in conjunction with the designed OBC based on the same family of FPGAs will also be tested in the same setup. The FPGA used to design the OBC is the Xilinx XC3S250E-VQ100I FPGA from the industrial temperature range at -40°C to +100°C. The basic HIL simulation will follow as illustrated in Figure 4.4 below used for FPGA data capture with HDL code read from FPGA internal signals.

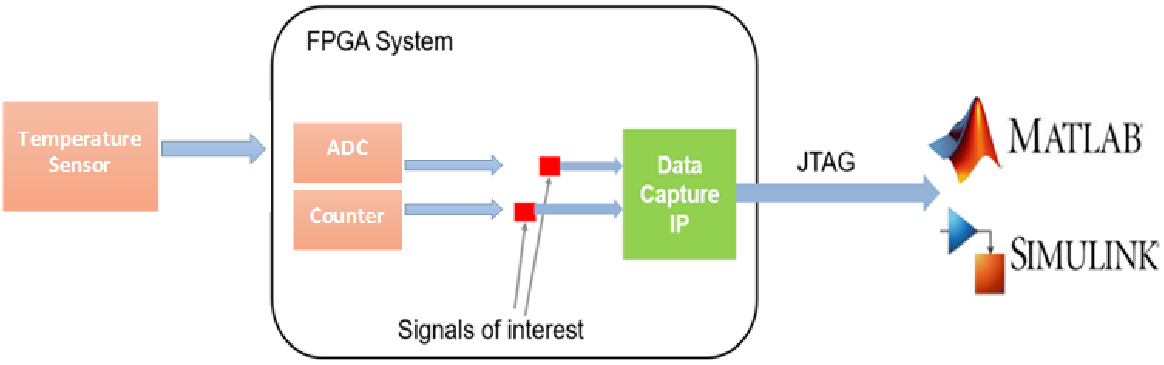


Figure 4.4: Basic HIL simulation through MATLAB

CHAPTER FIVE

HARDWARE DESIGN AND RESULTS

5.1 Overview

The chosen hardware must be able to comply with standard operational function expected from an OBC for small satellites. As there is no intended mission application or payload at hand for the duration of this thesis, the FPGA based OBC will be simulated in MATLAB and Simulink to justify real world applications. These tests and results are discussed in detail in this Chapter, with the conclusion and recommendations to follow in Chapter 6 of this thesis.

5.2 Hardware for HIL simulation

The hardware that is being used during this HIL simulation is done with the Digilent Basys 2 FPGA development board as the primary hardware for verification. The secondary hardware used is the FPGA based OBC designed for this thesis with the Xilinx Spartan 3E FPGA.

The FPGA based OBC is designed around the Xilinx XC3S250E-VQ100I FPGA. This FPGA is used as it has an industrial temperature rating of -40°C to +100°C compared to that of the standard commercial range as found on the Basys 2 board with temperature ranges of 0°C to +85°C.

The HIL simulation for the FPGA is done with MATLAB through data capture from the FPGA internal signals. These signals being captured are from an onboard temperature sensor used to simulate the temperature monitoring of the OBC board when in orbit. This temperature determines the operation efficiency of the FPGA and the circuit board performance in real world applications.

The thermal cycling (TC) effect that spacecrafts and satellites undergo in the severe temperature fluctuations in the space environment is some of the major dependability threats. With a LEO small satellite, such as the SwissCube, exterior temperatures are reported to change from 31°C through to -29°C as being illustrated in Figure 5.1 below. To prevent the shuttle from damage in severe external temperature fluctuations with outer surface protection, including thermal management, layer insulation, and sun shields, heat piping and heat radiators are being utilized in high-end small satellites. Small satellites like CubeSats, it is challenging to have these types of physical protections mostly due to the time involved, finances and physical restrictions (Kim and Yang, 2019).

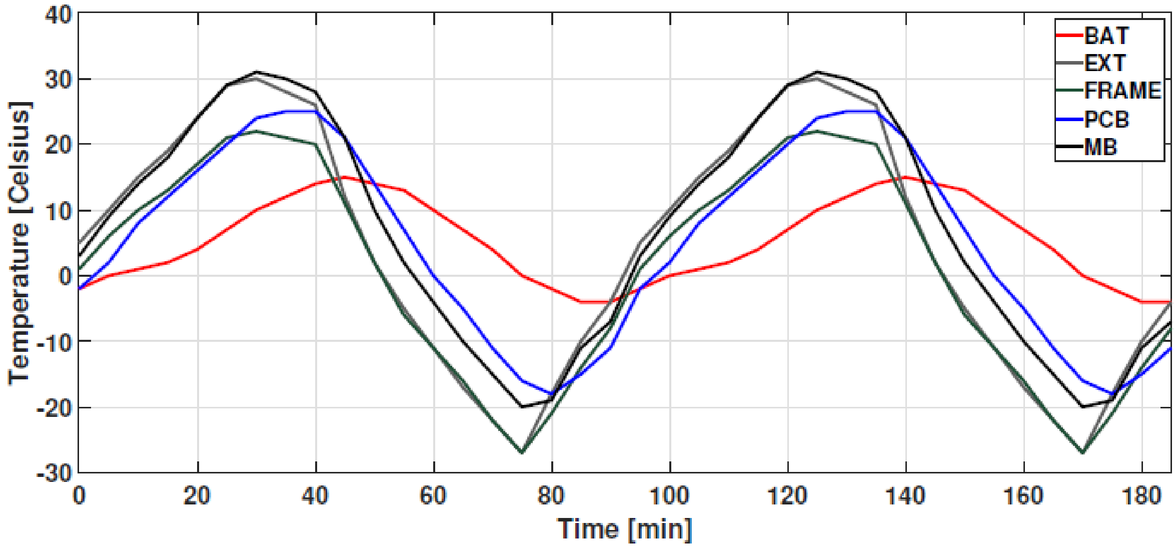


Figure 5.1: Temperature range over time for the SwissCube adapted (Kim and Yang, 2019)

With these fluctuations in temperature being experience in LEO for CubeSats, the electronic components onboard the OBC and other platforms may cause extensive damage. Modern CubeSats use COTS components that are not radiation hardened treated or does not have a TID rating and operate on the standard commercial temperature range as mentioned before. The components used to design the FPGA based OBC for this thesis are mostly in the same industrial temperature range and operating temperature ranges. Although none of the components are space grade electronics, it can be used within the space environment with an expected lifetime in orbit. The most important electronic components being the FPGA chip, flash memory and oscillators. These components are able to operate in temperature ranges of -40°C to +100°C.

Within the aerospace industry it must be considered that both the natural and aggressive ecosystems, natural environments for the designation of hardness and survivability. The latest preferred designed technologies, developed over the past few decades, make it viable to design spacecrafts and small satellites which will withstand natural and moderate levels of tough space environments.

Despite the fact that advances for solidifying against threatening military dangers and for normal endurance of satellites overlap, they are unmistakable and are typically treated independently aside from in the territories of survivability to add up to portion because of the Van Allen belts, SEE brought about by galactic cosmic rays and high vitality protons, and space charging because of normally happening space plasmas. The concluding singularities must be handled synergistically in the development and design of satellites.(Larson and Wertz, 1992)

5.3 Hardware design for Spartan OBC

Detailing the hardware components used to design the FPGA based OBC platform. The design and development of the satellite board is outlined within this Chapter and is aimed at providing support for most of the standard devices used onboard a small satellite. This board is designed to be used as a development board as there are no specified requirements for the board during the project period, therefore it is customisable and can be integrated into commercial small satellite components to control and read/send data. The board is designed on the PC104 standard for small satellites. Each section of the OBC is discussed in detail in this Chapter and the proposed HIL simulation. Refer to Annexure A.

5.3.1 OBC Memory

For the OBC designed board to hold a program on the FPGA a 16-Mb flash memory chip is implemented with the design. The memory chip used is one from Atmel, the AT45DB161E chip. The Atmel flash memory operates at 2.5V and is ideally suited for data storage applications including image storage and program code. The AT45DB161E has provision for RapidS serial interface for various projects that requires a higher speed operation. With various bits ranging from 17 to 504 bits of memory, organised as 4096 pages of 512 bytes each. Additional static RAM buffers of 512 bytes each as part of the main memory. The memory is partitioned into three levels of granularity comprising of sectors, blocks, and pages for optimal flexibility. Figure 5.2 shows a breakdown of the levels and details the number of pages per sector and block in the memory architecture (Extra, 2012).

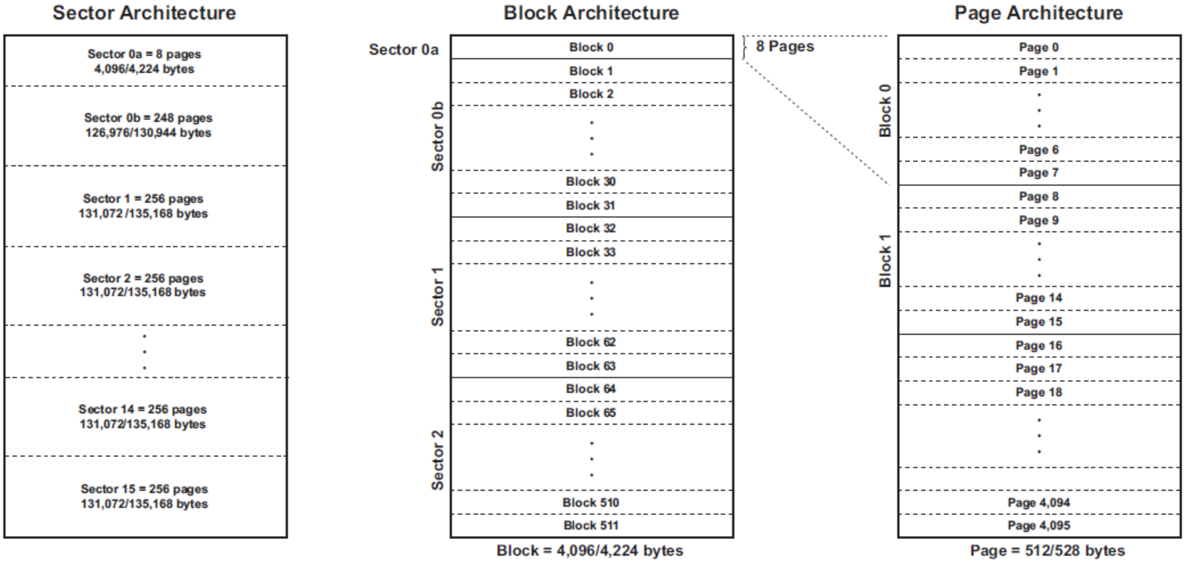


Figure 5.2: Memory architecture diagram illustrating breakdown of levels (Extra, 2012)

The memory array from the AT45DB161E, is divided into three levels of granularity consist of sectors, blocks and pages. Figure 5.2 exemplifies the rundown of each level and details the number of pages per sector and per block.

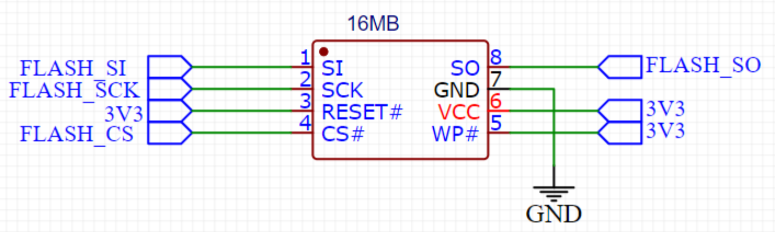


Figure 5.3: Flash memory circuit diagram used on the FPGA OBC design

5.3.2 Oscillator

The crystal oscillator used in the design of the OBC is in the same operating temperature range as the Xilinx FPGA and Atmel flash memory. The crystal oscillator is running at 32MHz for this board. Operating voltage of 3.3V and maximum current at Icc = 20mA. The Figure below shows the in-circuit connection of the crystal oscillator. A bypass 0.1uF capacitor is used to stabilise the oscillator.

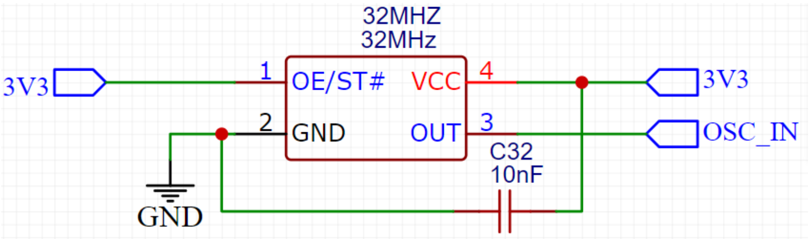


Figure 5.4: 32MHz SG5032CCN Crystal Oscillator

5.3.3 Programming Interface

The FPGA OBC needs to be programmable and configurable to suite an end user’s requirements. Therefore, it has been designed to be able to connect to the Xilinx XC3S250E FPGA through the onboard JTAG connection or via the micro-USB port.

The FT2232D is used in for the USB interface which incorporates the functionality of two second generation BM chips into a single device. The standalone downstream USB port is converted to tow IO channels which can be individually configured as a UART interface or as a FIFO interface. This makes it more convenient to program the FPGA through an onboard USB port. This USB port is also used to connect to the HIL simulation environment setup in MATLAB.

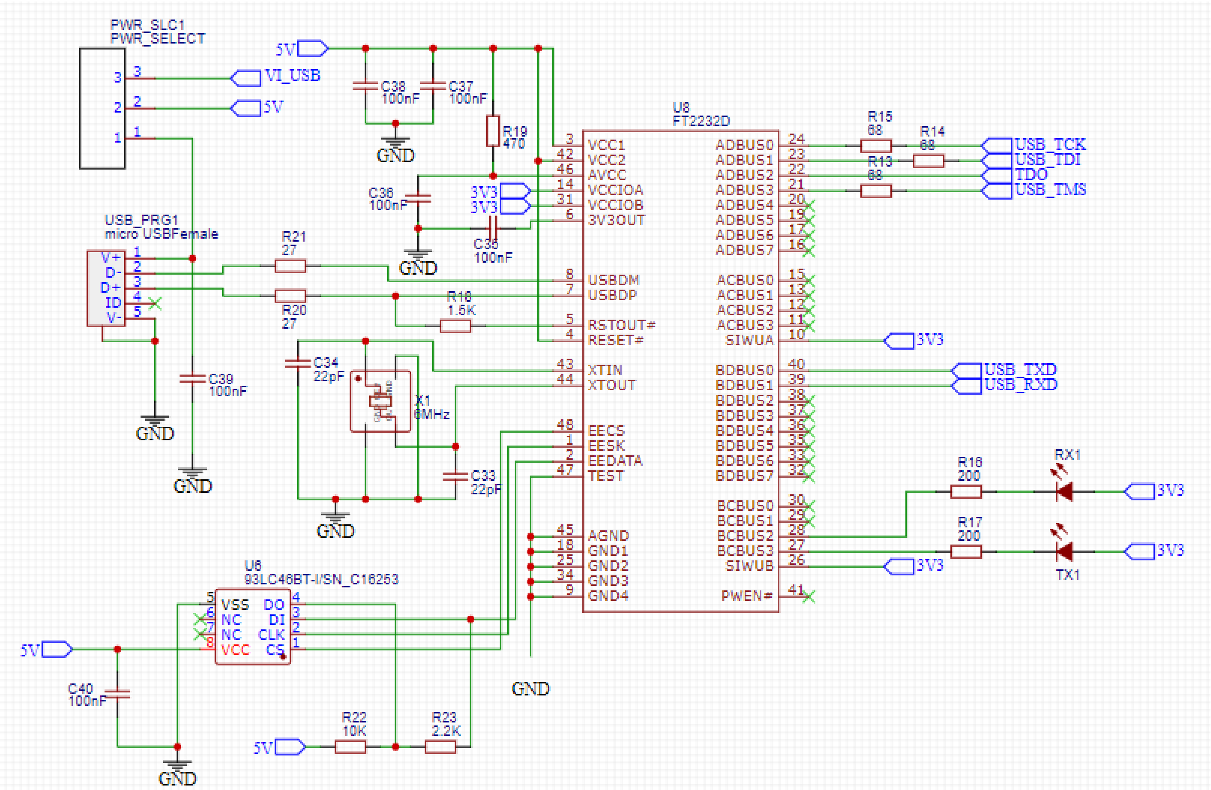


Figure 5.5: FT2232D USB Interface schematic used for FPGA OBC

For additional support, the FPGA may be programmed through the JTAG configuration as an alternative. To program the FPGA using the JTAG, a USB blaster module is required. The Xilinx Spartan 3E family FPGAs and the Platform Flash PROMs all make use of a four-wire JTAG port compliant with the international IEEE 1149.1/1532 standard. They divide the JTAG TCK, USB_TCK, clock input and the select input function through TMS mode. These components can connect in any order with the TDO output on the JTAG string of one device transmitting the TDI/USB_TDI input from the other device in the sequence. The TDO output from the last device in the JTAG string leads the JTAG connector. This programming interface on the FPGA is powered through the VCC_AUX supply. Subsequently, the PROM's VCC supply must additionally be 2.5 V. (UG332, 2015)

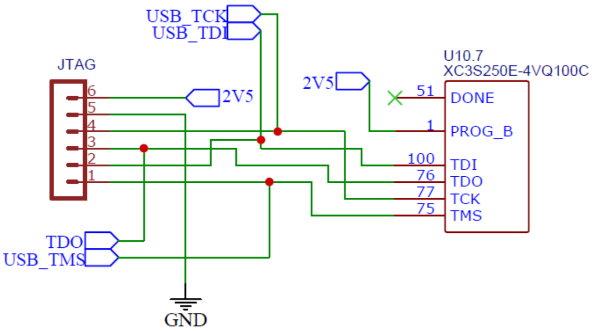


Figure 5.6: JTAG Connection interface for the FPGA OBC

5.4 PC/104 form factor

The FPGA based OBC is design around the PC/104 form factor. The PC/104 modules are either 8-bit or 16-bit bus type. The PCB complies to the mechanical specifications of the PC/104 16-bit module. The PC/104 standard for embedded applications have the following restrictions:

- PCB boards are to be sized to 90.17 mm X 95.89 mm
- Board spacing with standoffs should not be less than 15 mm
- Bus connectors may be used for terminations with “stack-through” or “non-stack-through” headers
- ISA Bus must be configurable for 8-bit or 16-bit modules

The possible stack configuration for multiple boards are shown in the image below with 8-bit and 16-bit bus modules.

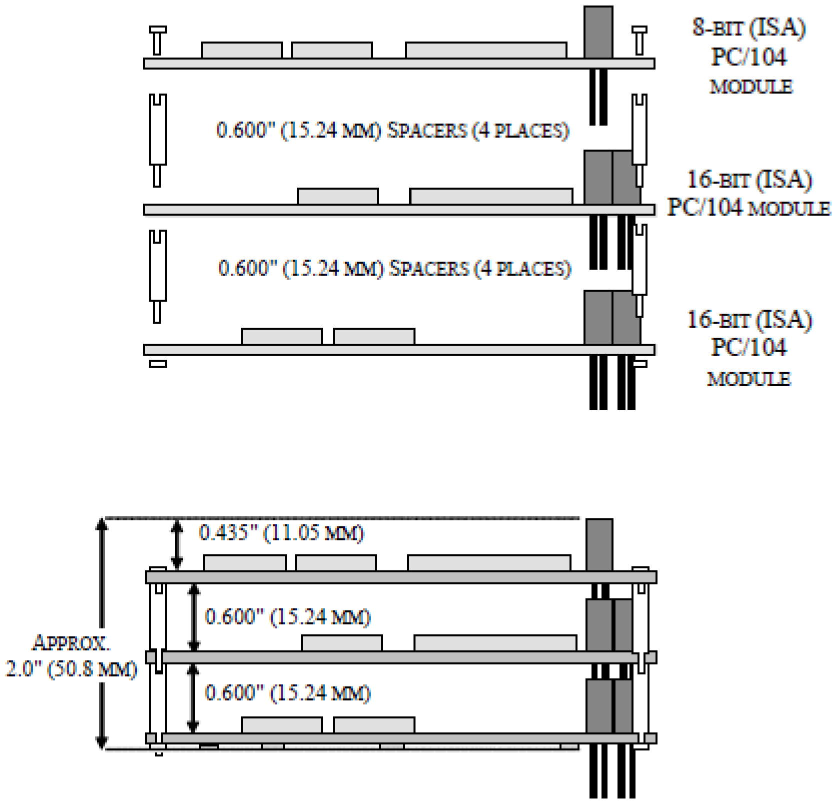

8-BIT (ISA) PC/104 MODULE
0.600" (15.24 MM) SPACERS (4 PLACES)
16-BIT (ISA) PC/104 MODULE
0.600" (15.24 MM) SPACERS (4 PLACES)
16-BIT (ISA) PC/104 MODULE
0.435" (11.05 MM)
0.600" (15.24 MM)
APPROX. 2.0" (50.8 MM)
0.600" (15.24 MM)

Figure 5.7: PC/104 board stack configuration (Note, 2003)

5.5 Printed Circuit Board

The physical PCB is designed in CAD software with reference to the PC/104 requirements in mind. For all the components to be compact and seamlessly integrated onto one board, a four-layer PCB is design. All the electronic components are surface mount apart from the ISA header, JTAG header and micro-USB ports for programming and external power supply. The materials under PC/104 are tabulated below in comparison to the actual materials used in manufacturing the PCB.

Table 5.1: PCB material comparison

Materials	PC/104	FPGA OBC
Layers	N/A	4 layer
PCB thickness	>=1.6 mm	1.6mm
Housing	Thermoplastic, UL Rated 94-V0	FR4-Standard Tg 130-140C
Contact	Phosphor	Phosphor
Solder	Tin-Lead (63-37)	HASL(with Lead)
Solder Clip	Aluminium Alloy	N/A
Female interface	15 µm hard gold	TG155 Cu
Male interface	Gold flash	Gold Flash
Solder tail	100 µm solder	1 oz
Underplate	50 µm nickel	N/A

For the design of this OBC board, a four-layer double sided board was designed due to the ease of integration of the ground and power tracks on different layers. By using a four-layer board, the use of through hole components are eliminated making it more compact. All components are placed on the top layer to make direct connections easier to the FPGA with all signal tracks. The Spartan 3E XC3S250E-4VQ100I FPGA is placed in roughly the centre of the board and at an angle of 45°, to access all pins of the chip with relative equal length tracks.

5.5.1 Power Supply

The Xilinx XC3S250E FPGA requires various voltages inputs for optimal operation along the flash memory chip. Therefor a very stable and reliable power supply and control circuity is required. Since the different voltages that are required to provide 1.2 V to 5 V on electronic components, the use of the AMS1117 voltage regulator is introduced. The use of linear voltage regulators is the most compatible for this OBC build in the since of their efficiency and readily available.

The circuit design for the AMS1117 series regulators requires the use of an output stabilising capacitor as part of the device frequency compensation. This additional capacitor will ensure stabile operation in most operational conditions with temperature range of 0°C to +150°C.

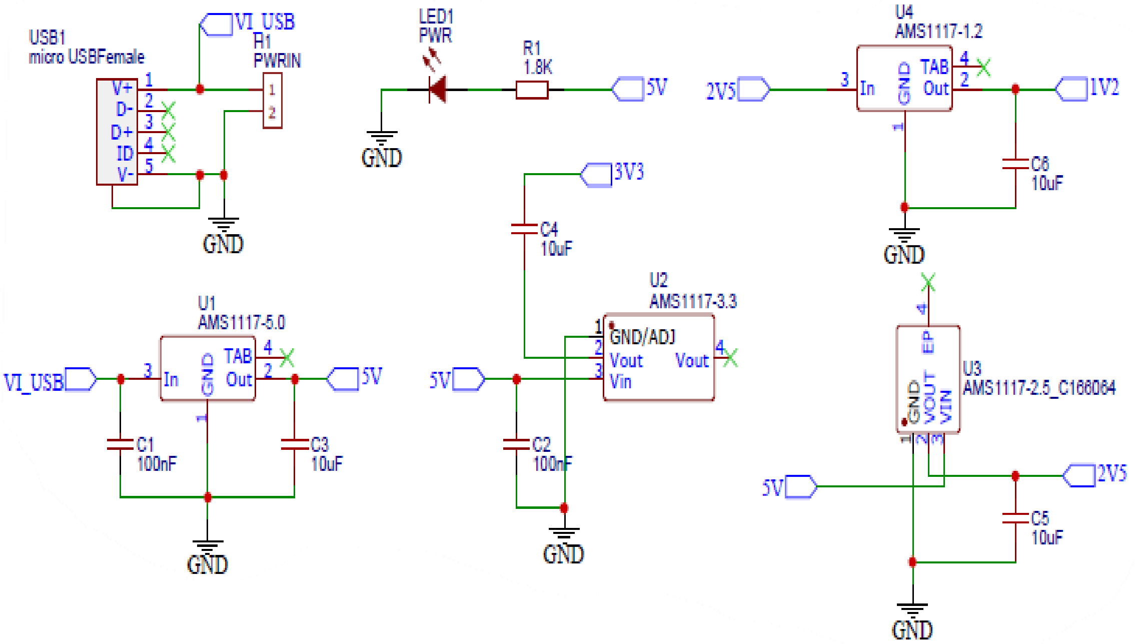


Figure 5.8: AMS1117 Voltage Regulators as the ESP for the OBC

The AMS1117 voltage regulator series have internal power and thermal restrictive circuitry designed to safeguard the device under overload conditions such as the space environment for LEO small satellites. Such as the SwissCube that experienced temperature range of 30°C to -30°C. The design of the voltage regulators used on the OBC are illustrated in the Figure below.

The power supply subsystem was submitted and considered for implementation. The AMS1117 is designated as the preferred component for this subsystem of the OBC and its operations were confirmed through a prototype build. Peripherals for I/Os and communication and programming interfaces of the FPGA to be implemented on the OBC are presented in terms of operation, performance, and feasible applications on a small satellite platform. The concluding PCB implementation of the proposed design was incorporated and the orientation of parts are based on the PC/104 form factor for the whole board.

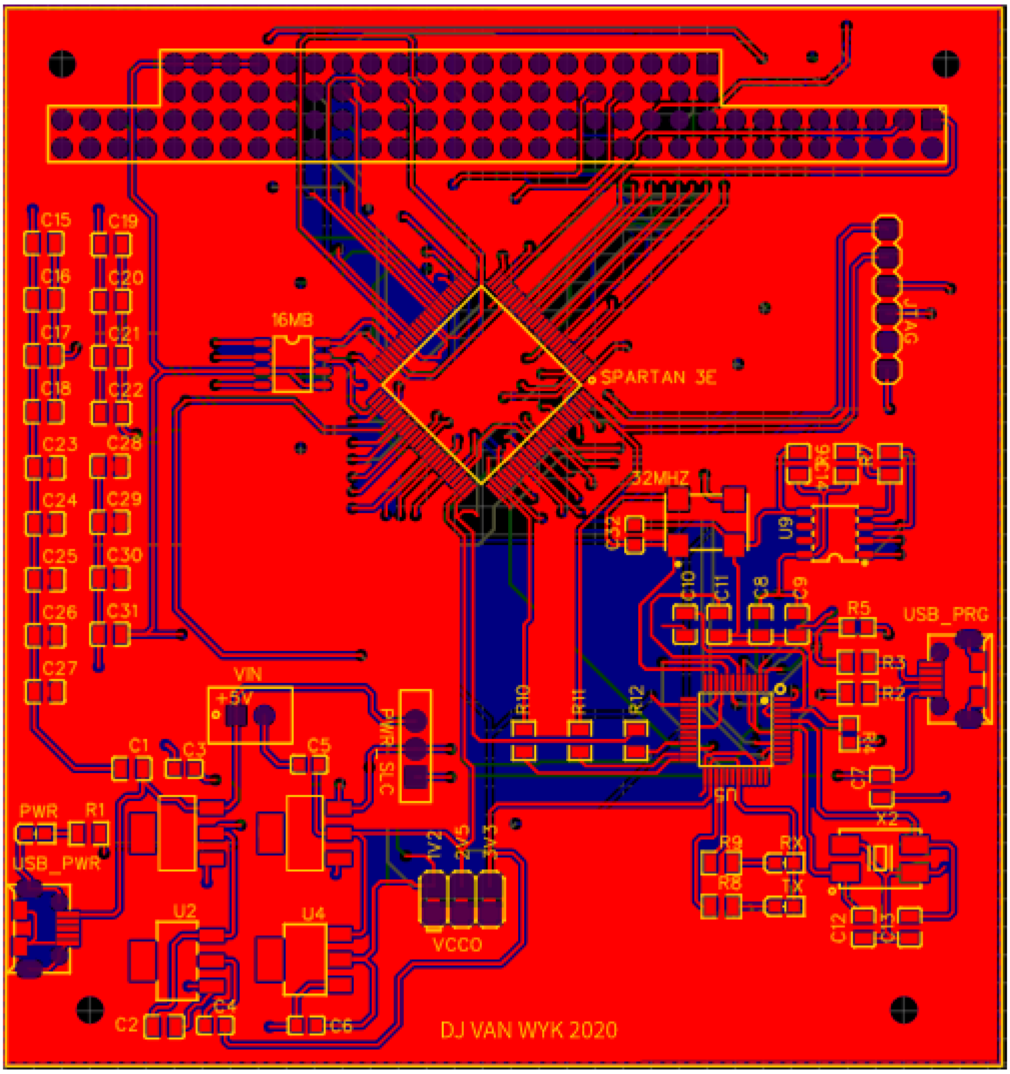


Figure 5.9: Top silk layer of proposed FPGA OBC (DJVW, 2020)

5.5.2 Communication

I^{2}C buses supports bidirectional data transfer between master and slave buses, multiple master bus, intercession between instantaneously communicating masters without distortion of serial data on the bus. Serial clock synchronization gives devices with different bit rates the ability to communicate through a common serial bus and can be utilized as a grip mechanism to halt and resume serial data exchange. The payload outline below describes the internal communication structure used in the ZACUBE-2 satellite with its various payload subsystems.

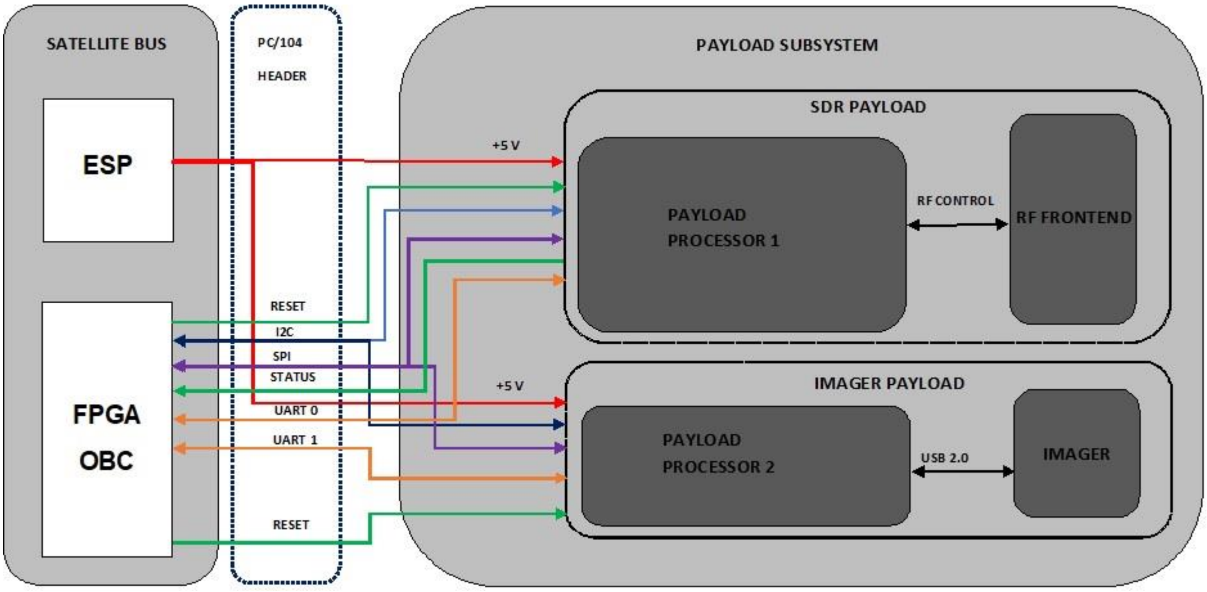


Figure 5.9.1: ZACUBE-2 payload layout and interfaces (Adapted from CPUT)

The payload processor performs all the processing related to the specific mission payload. With regard to ZACUBE-2, it performs SDR processing for AIS communication with all the procedures used for image processing.

The SPI Interface is a universal serial bus that can function in both master and slave modes. This interface contains the Data Input, and Data output as well as the clock with SPI configure lines. The Xilinx Spartan 3E FPGA used in the design of the proposed OBC has the industry standard SPI serial flash interface which makes use of the device’s internal oscillator. This SPI configuration interface can be used as stand-alone FPGA applications.

SPI serial flash mode is a standard four-wire interface can be defined by what means the FPGA communicates and interfaces with the SPI flash, as well as which flash command the FPGA executes to start the (DATA_IN) read procedure and the amount of imitation bytes added before the FPGA expects to retrieve/receive useful information that is valid. The Figure below details the interface for PROMs supportive of read and fast read commands through the SPI flash PROM.(Information *et al.*, 2009)

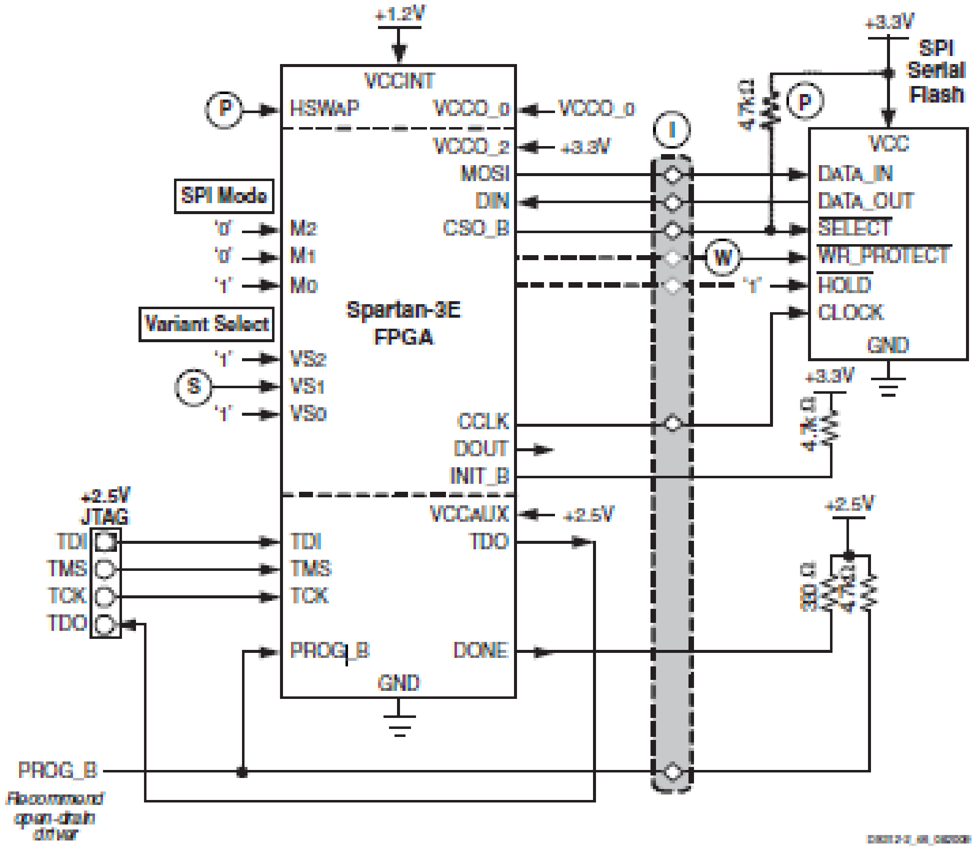


Figure 6: Xilinx FPGA SPI Flash PROM interface (Information *et al.*, 2009)

CHAPTER SIX

RESULTS AND SIMULATION

6.1 Overview

This final Chapter will discuss the details around the architecture of on-board computer systems used in the space satellite industry. Focus will be on the hardware used for the main processor of the OBC as well as the design structure. The simulation and verification of the memory system and communication of the FPGA is discussed briefly in this Chapter.

6.2 Simulations and verification

The designed and manufactured OBC based on the Xilinx Spartan 3E FPGA is tested through simulation of the memory integration and communication interface using UART. To verify the simulations and VHDL test bench script for the OBC, the Digilent Basys 2 development board is used to compare the results. The OBC is designed to be programmable through the JTAG pin header and the USB interface.

6.2.1 FIFO

First In First Out (FIFO) data transfer and cross clock domains. FIFOs are used within FPGAs mostly throughout and ASIC designs as basic building blocks. FIFOs have various purposes of which include crossing clock domains, storing data and buffering data prior to sending it to the RAM modules. A FIFO is a one-way channel for data to transfer through.

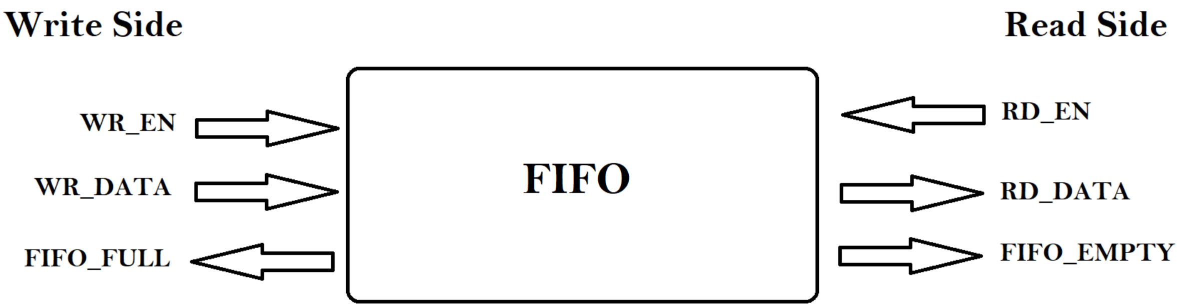


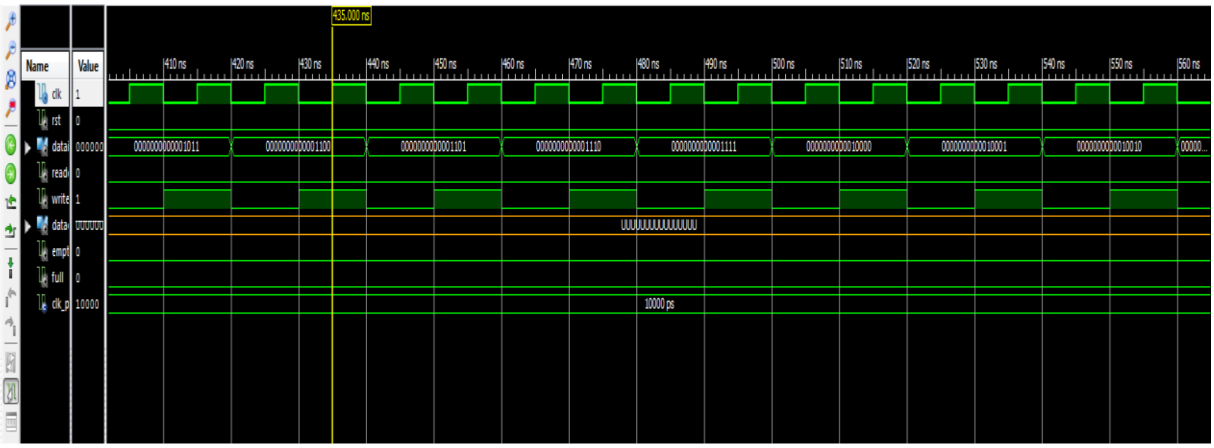
Figure 6.1: Basic interface signals of a FIFO module

FIFOs are basically memory buffers that are used to momentarily store bits of data until another process is set to accept/read it.

From the Figure 6.1 above; the WR_EN (Write Enable) is used to write data into the FIFO with the WR_DATA (Write Data) holding the data that is sent to the FIFO. The FIFO model has a clock and reset trigger points. The clock is used for every action to execution upon trigger of the clock on the rising (HIGH) edge or falling (LOW) edge of the clock frequency based on your application. As the name states, reset is the signal used to reset the FIFO memory when a high signal is transmitted.

From the read side, there is the RD_EN (Read_Enable) which allows the reading of the current data from the FIFO memory. To read data, the RD_EN input must receive a high input during one clock cycle and the data the is stored first will be read first from the RD_DATA (Read_Data) pin. When reading data constantly or at a high rate, the RD_EN pin's state may be set high for a longer period than one clock cycle.

The write and read function are implemented on the designed FPGA Spartan 3E board. This verifies the memory operation of the OBC with 16 MB flash memory onboard. A basic write sequence is done in VHDL using the Xilinx IDE platform to configure the OBC.



Name	Value
clk	1
rst	0
datai	000000
read	0
write	1
data	UUUUUU
empt	0
full	0
clk_p	10000

Figure 6.2: Read and Write sequence of Spartan 3E

The screen snippet above details the read and write sequence on the flash memory through a FIFO model. The is done through every clock pulse, the write of data is enabled.

The onboard memory is from Atmel, model AT45DB161E. This module allows for simple in-system programmability and re-programmability which does not necessitate high input voltages. Where the flash memory operates off a single 2.5 V power source for the erase, write and read commands. This flash memory is ideal for the use within various applications to store data, images and program code. Rapid serial interface through the 3-wire interface is done using the Serial Input (SI), Serial Output (SO) and the Serial Clock (SCK) pins (Extra, 2012).

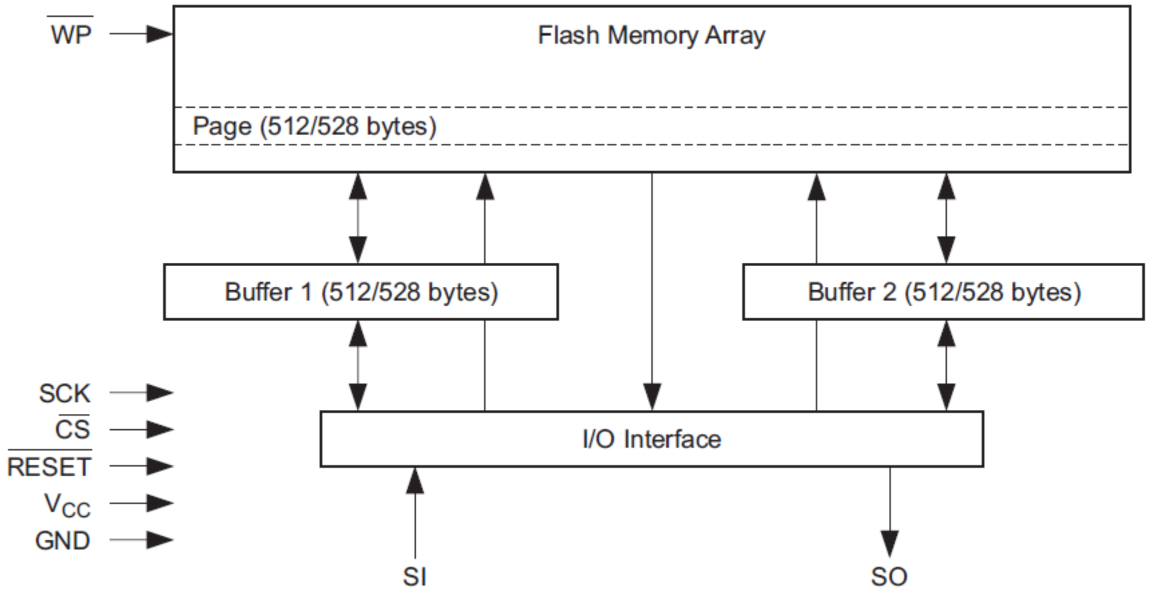


Figure 6.3: Block diagram of the Atmel AT45DB161E memory device

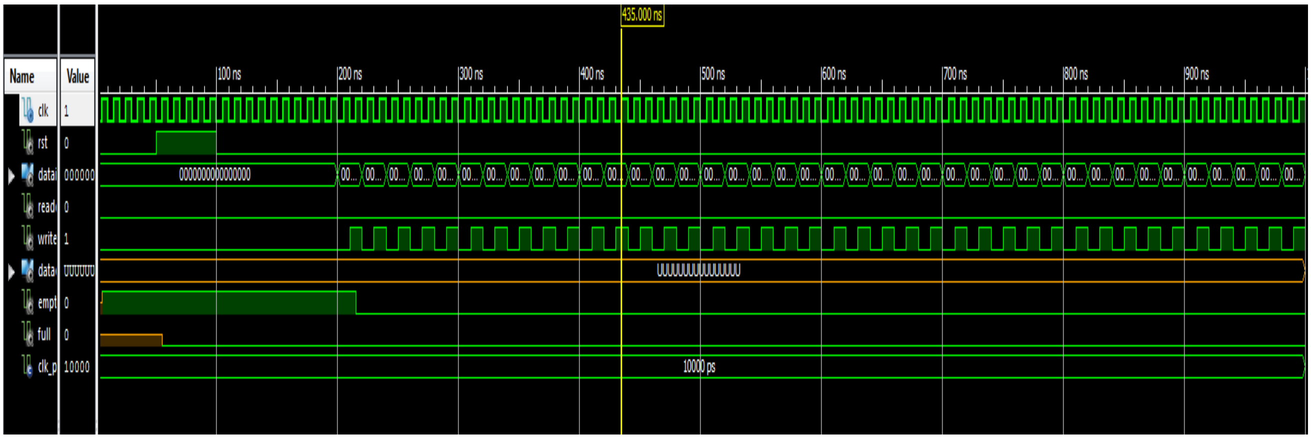


Figure 6.4: Full Clock cycle FIFO simulation

The Figure above shows the outcomes of the write/read operation for a full cycle where the FIFO buffer is full. In simultaneous read and write FIFOs, there is no dependency between the write or read operations of the data. Instantaneous reading and writing are possible in an overlapping manner or continuously. In other words, the two systems with different frequencies can be connected to the FIFO buffer. When designing a platform, you do not need to concern regarding harmonising these two systems as this is taken into consideration by the FIFO. Parallel read and write operations of the FIFOs depend on the command signals for reading and writing which are based in two groups, synchronous and asynchronous FIFO banks. For this simulation the synchronous standard FIFO was used. The DATA_IN signal is the input to the FIFO with the DATA_OUTPUT as the output. WR_EN is the signal used to write data into the FIFO memory when the CLK is triggered on the positive edge. RN_EN is used for reading data from the FIFO buffer and can be reset through RS as high. The FIFO buffer will indicate when it is full with the FULL signal at high and the same for the EMPTY signal when the buffer is empty, the signal will be high. The Figure below illustrates the basic FIFO write and read procedure.

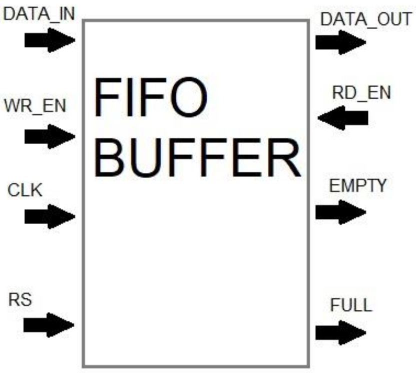
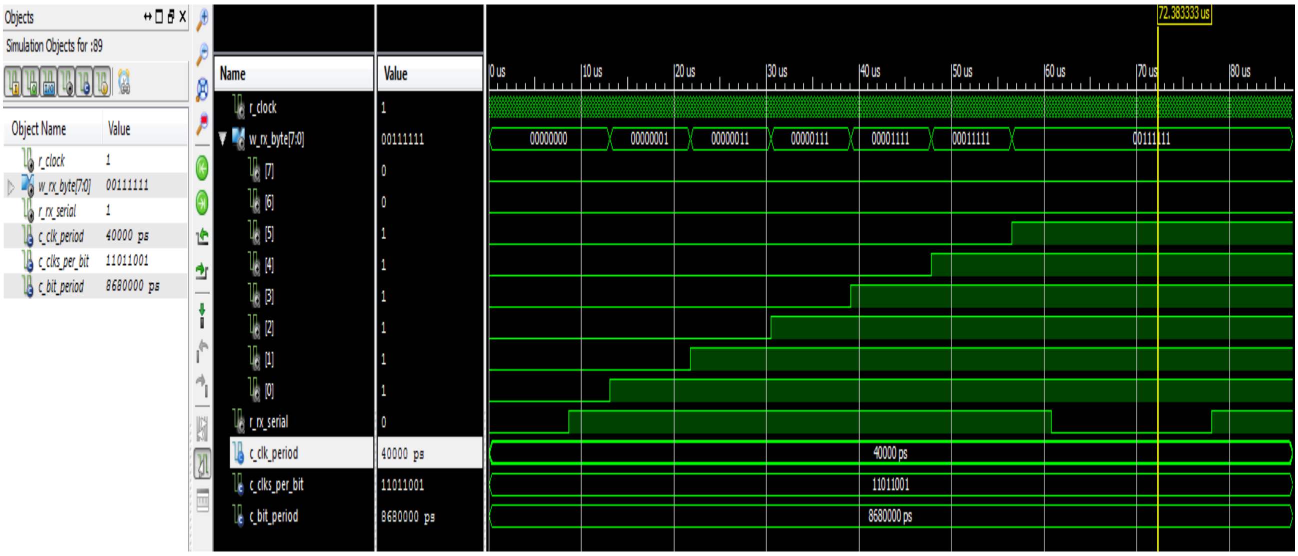


Figure 6.5: FIFO memory write/read sequence

6.2.2 UART

The requirement for serial exchange of data for small satellites represents a significant prerequisite. Serial communication allows transmission of one bit or receive one bit at a given time between two or more devices and involves the transfer of data over a single wire, channel, in either direction.

The UART transmission is of a serial type communication. Nonconcurrent transmission suggests that both the receiver and the transmitter have singular local clock signals which are established before the proceeding of the communication instead of simultaneous, where the communication transmission is synchronised on both ends through a common clock.



Objects

Simulation Objects for :89

Object Name	Value
r_clock	1
w_rx_byte[7:0]	00111111
r_rx_serial	1
c_clk_period	40000 ps
c_clks_per_bit	11011001
c_bit_period	8680000 ps

Name	Value
r_clock	1
w_rx_byte[7:0]	00111111
[7]	0
[6]	0
[5]	1
[4]	1
[3]	1
[2]	1
[1]	1
[0]	1
r_rx_serial	0
c_clk_period	40000 ps
c_clks_per_bit	11011001
c_bit_period	8680000 ps

Figure 6.6: FPGA OBC simulated UART communication

The UART mode of communication transfers data in blocks of 8 bits, also known as frames, which are user-configurable in terms of contents. The USART can function in either synchronous and asynchronous modes, which are application dependant (LWABANJI TONY LUMBWE, 2013).

Figure 6.6 above shows the output of the UART communication between the FPGA based OBC designed with successful WRITE and READ commands of 8-bits.

CHAPTER SEVEN

7.1 Conclusion

This thesis presents an investigative design approach for a FPGA based on-board computer using the Xilinx Spartan FPGA series. This OBC is proposed for integration with standard payload platforms based on the PC/104 form factor for small satellites. The proposition is innovative since the FPGA chip being used is of older technology and based on the processing power of FPGAs.

The first steps present for this research thesis was to establish the various parameters governing an on-board computer for small satellites within low Earth orbit and the space environment. Emphasis was put on the main components that are proposed for the OBC by using commercially available electronic components. The FPGA device had to be of commercial grade for the operating temperature and durability that will most likely survive in the space environment. As well as the current systems and electronic components being used in the small satellites industry that are of space grade and/or radiation hardened components. Research was done on various FPGA manufacturers and current commercial OBC platforms available for satellite systems. A comparison is done on the most feasible and readily available FPGA packages as to which one will be used for the design implementation and verification of the OBC.

The final FPGA that was used is one from Xilinx's Spartan 3E family manufactured in 2018 which has been available by Xilinx for a number of years now. The Spartan chip has 250K system gates with a distributed RAM 38 K and 68 differential I/O pairs. This processor was designed to operate in temperature ranges of -40°C to +100°C which is suitable for low Earth orbit missions and applications, at low voltage. The FPGA can be programmed through the USB interface and the JTAG header pins. Programming can be stored on the on-board external 16MB flash memory. Generic programming through Verilog or VHDL to configure payloads or subsystems to control or read data from. The designed OBC has an on-board power supply unit to power the various electronic components from the FPGA to the flash memory.

This FPGA OBC was tested under a semi-controlled environment and not simulated under conditions similar to the space environment. The reliability will only be confirmed upon vacuum testing and vibration testing under heat. The electronic components are susceptible to radiation damage or SEE from the cosmic rays, as the electronic component used are COTS components. Performance may not be as tested during this research proposed thesis.

7.2 Recommendations

The simulations done during this thesis to verify and test the codes was done using a smaller grade evaluation board with limited integration with software packages such as MATLAB and Microblaze. The FPGA OBC therefor could also only be verified through the Vivado Studio environment with basic functionality.

By using the latest available evaluation boards for such simulations, testing and verification can be done through a proper HIL simulation and implementation to the space industry standards satisfaction.

Further research into the field of the use of FPGAs as the main processor on a given OBC with the Xilinx or Atmel Space Grade components will produce well designed products. These are costly components and not readily available. The overall OBC's PCB can be made smaller to fit within a PocketQube or integrate other subsystems of a CubeSat onto one board alongside the OBC. Integration can be made to incorporate other Xilinx devices with minor changes or provision for an Altera device to compare against the Xilinx family.

The process and workings done throughout this thesis shows how COTS components can be utilized for space applications and at low cost for use in small spacecrafts, satellites, and space applications with the use of FPGAs.

7.3 Future work

This thesis is a research-based approach to the development and design of a FPGA based OBC where COTS components are used during the thesis. To scale this research and development into a commercial product as the final outcome would be considered for the use to build small satellites in universities and school to learn about the space industry. Building their own functional satellite for small scale missions with an optical payload or weather station for data capturing. As this has been a huge learn curve for myself and keeps on piquing my interest within the space industry.

References

1-CORE Technologies (no date) 'SRAM Based FPGA Architectures Overview'. Available at: https://www.pdx.edu/nanogroup/sites/www.pdx.edu.nanogroup/files/FPGA-architecture.pdf.

Antunes Sandy (2015) *DIY Comms and Control for Amateur Space*. Maker Media, Inc.

Desai, A. P. (2010) 'Design of On Board Computers for a Nanosatellite'.

Extra, D. (2012) 'Atmel AT45DB161E Pin Configurations and Pinouts', pp. 1–70.

Habinc, S (2002) 'Lessons learned from FPGA developments', *Gaisler Research*, pp. 1–36. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Lessons+Learned+from+FPGA+Developments#0%5Cnhttp://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Lessons+learned+from+FPGA+developments#0.

Habinc, Sandi (2002) 'Suitability of reprogrammable FPGAs in space applications', *Feasibility Report*, pp. 1–44.

Hanafi, A. and Latachi, I. (2017) 'FPGA-based Secondary On-Board Computer System for Low-Earth-Orbit Nano-satellite', pp. 1–6.

Hillier, C. and Balyan, V. (2019) 'Error Detection and Correction On-Board Nanosatellites Using Hamming Codes', *Journal of Electrical and Computer Engineering*, 2019. doi: 10.1155/2019/3905094.

Hillier, C. P. (2018) 'A system on chip based error detection and correction implementation for nanosatellites by', (October).

Hillier, C. P. and Balyan, V. (2019) 'Effect of Space Radiation on LEO Nanosatellites', Journal of Engineering and Applied Sciences 14 (18), 6843 - 6857.

Huber, F. *et al.* (2007) 'FPGA based on-board computer system for the "Flying Laptop" micro-satellite', *European Space Agency, (Special Publication) ESA SP*, (SP-638).

Information, O. *et al.* (2009) 'Spartan-3E FPGA Family : Module 1 : Spartan-3E FPGA Family : Introduction', 312, pp. 1–233.

Kastensmidt, F. and Rech, P. (2015) *Radiation effects and fault tolerance techniques for FPGAs and GPUs*, *FPGAs and Parallel Architectures for Aerospace Applications: Soft Errors and Fault-Tolerant Design*. doi: 10.1007/978-3-319-14352-1_1.

Kim, B. and Yang, H. (2019) 'Temperature sensor assisted lifetime enhancement of satellite embedded systems via multi-core task mapping and DVFS', *Sensors (Switzerland)*, 19(22), pp. 1–17. doi: 10.3390/s19224902.

Kuwahara, T. (2010) 'FPGA-based reconfigurable on-board computing systems for space applications', *Engineering Sciences*. Available at: http://elib.uni-stuttgart.de/opus/volltexte/2010/5030/.

Larson, W. J. and Wertz, J. R. (1992) *Space mission analysis and design*, *Choice Reviews Online*. doi: 10.5860/choice.29-5149.

Lokman, A. H. *et al.* (2017) 'A Review of Antennas for Picosatellite Applications', *International Journal of Antennas and Propagation*, 2017. doi: 10.1155/2017/4940656.

LWABANJI TONY LUMBWE (2013) 'Development of an onboard computer (OBC) for a CubeSat', (May), pp. 1–178.

Note, P. (2003) 'PCI-104 Specification', *Race*, (November).

Rockett, L. *et al.* (2007) 'Radiation hardened FPGA technology for space applications', *IEEE Aerospace Conference Proceedings*. doi: 10.1109/AERO.2007.353098.

Sensing, C. and Concepts, C. (2011) '[1]電流検出回路のコンセプト、存在意義(重要！).Pdf', pp. 1–12.

Staff, M. D. D. (2013) 'Small Spacecraft Technology State of the Art Mission', (NASA/TP--2014--216648/REV1), p. 43. Available at: http://www.nasa-usa.de/sites/default/files/files/Small_Spacecraft_Technology_State_of_the_Art_2014.pdf.

Thai, T. (2014) 'Applications for FPGAs on Nanosatellites', (April).

Tlelo-Cuautle, E., Rangel-Magdaleno, J. de J. and de la Fraga, L. G. (2016) *Engineering applications of FPGAs: Chaotic systems, artificial neural networks, random number generators, and secure communication systems*, *Engineering Applications of FPGAs: Chaotic Systems, Artificial Neural Networks, Random Number Generators, and Secure Communication Systems*. doi: 10.1007/978-3-319-34115-6.

UG332 (2015) 'Spartan-3 Generation Configuration', *Xilinx User Guide*, 332.

Uryu, A. N. (2013) *A Combined Data and Power Management Infrastructure*. doi: 10.1007/978-3-642-35557-8.

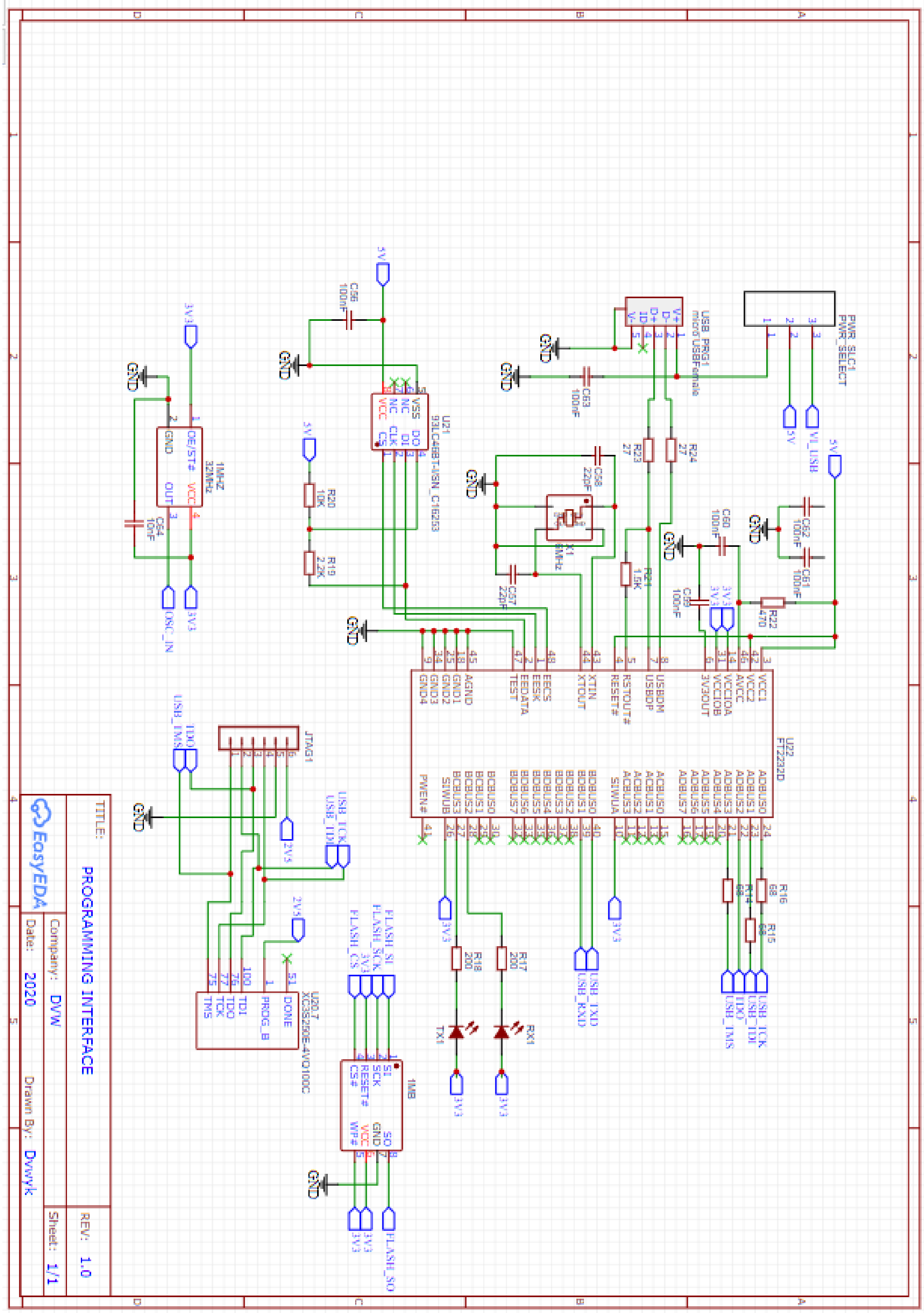
Varnavas, K., Sims, W. H. and Casas, J. (2015) 'The Use of Field Programmable Gate Arrays (FPGA) in

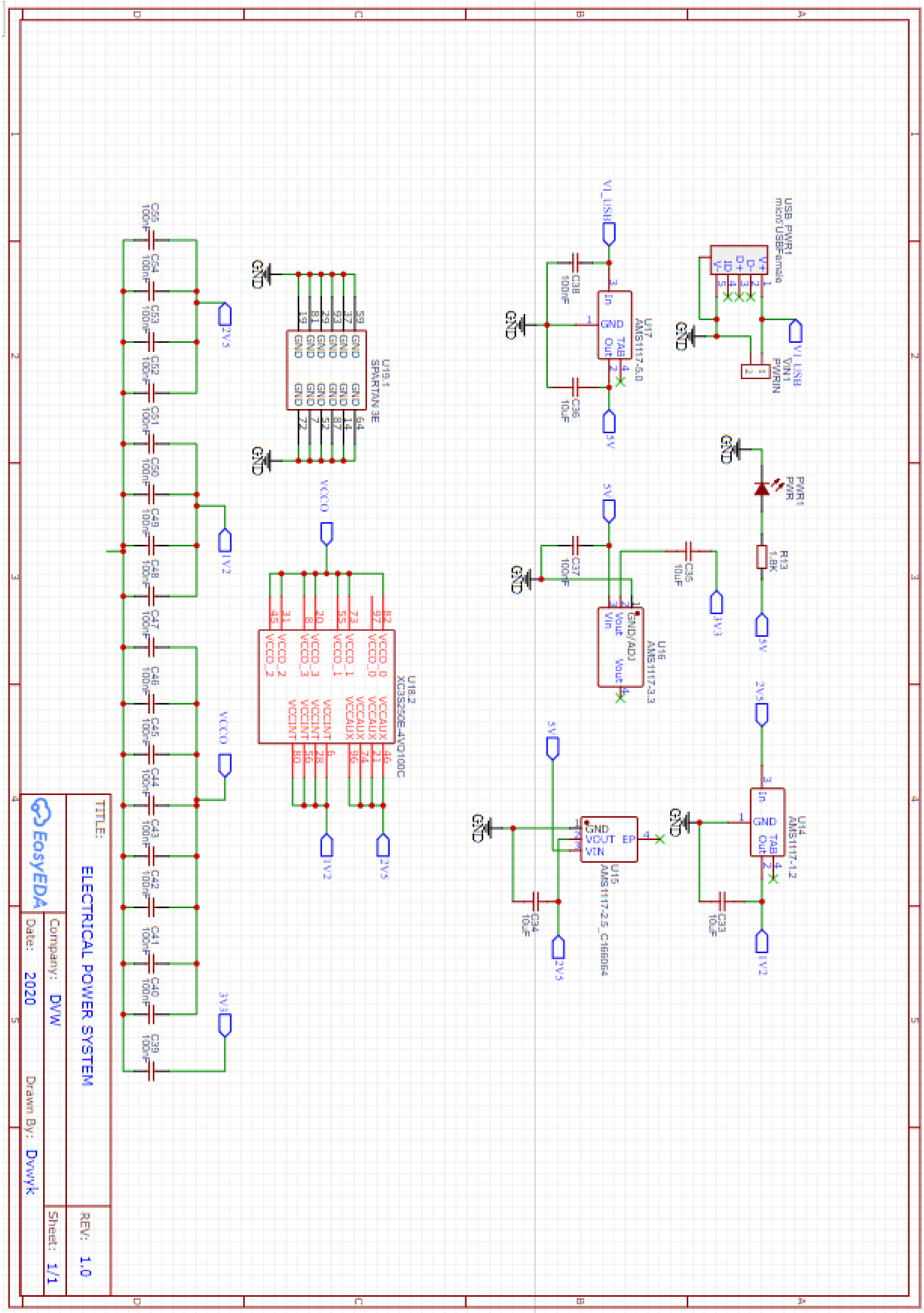
Small Satellite Communication Systems’, *SPACOMM 2015: The Seventh International Conference on Advances in Satellite and Space Communications*, pp. 86–89. Available at: https://www.thinkmind.org/download.php?articleid=spacomm_2015_4_30_20063.

van Wyk, D. J. (2018) ‘Optical PocketQube Mission Implementation’, (November).

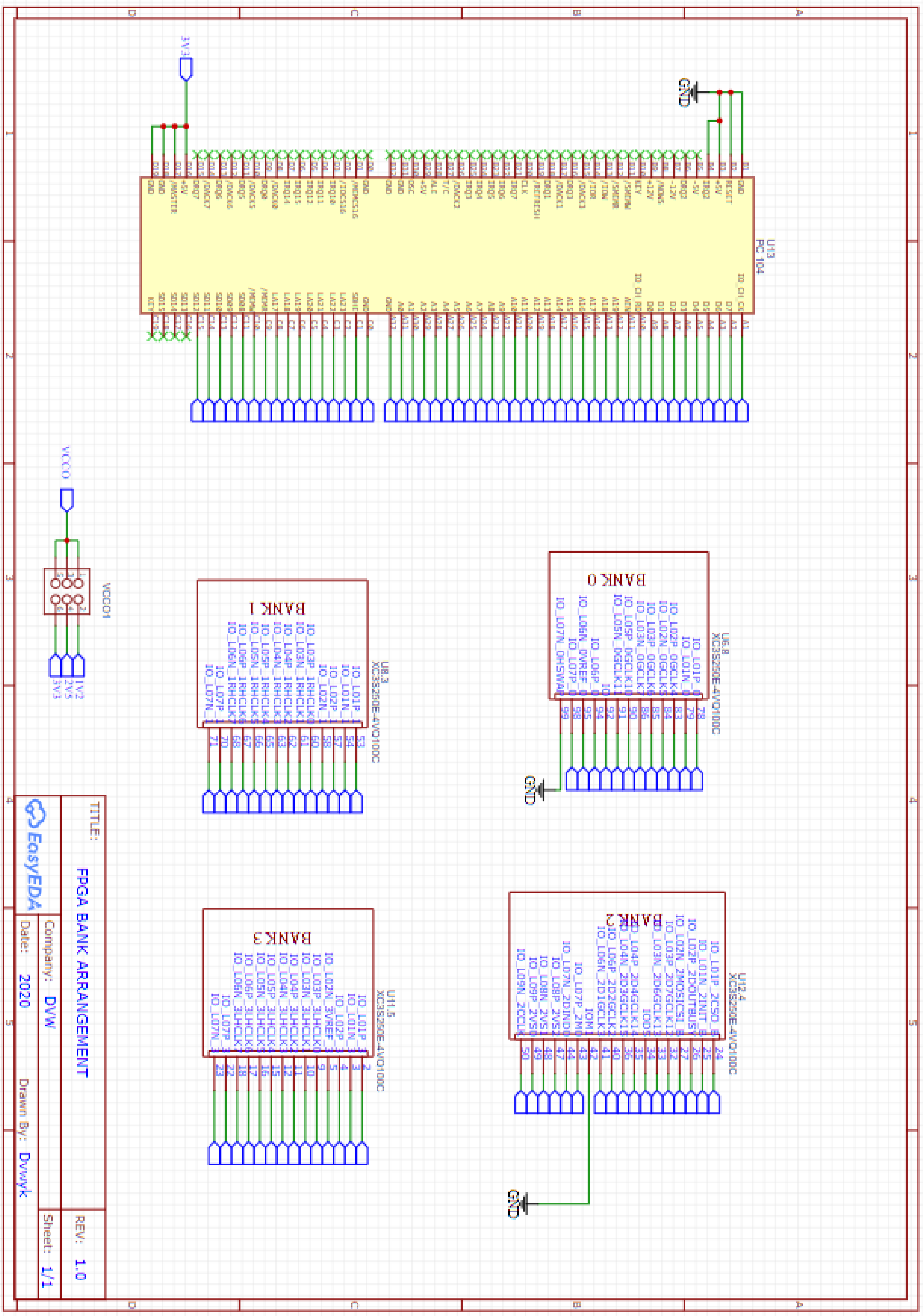
Annexure A

FPGA OBC Schematic diagram

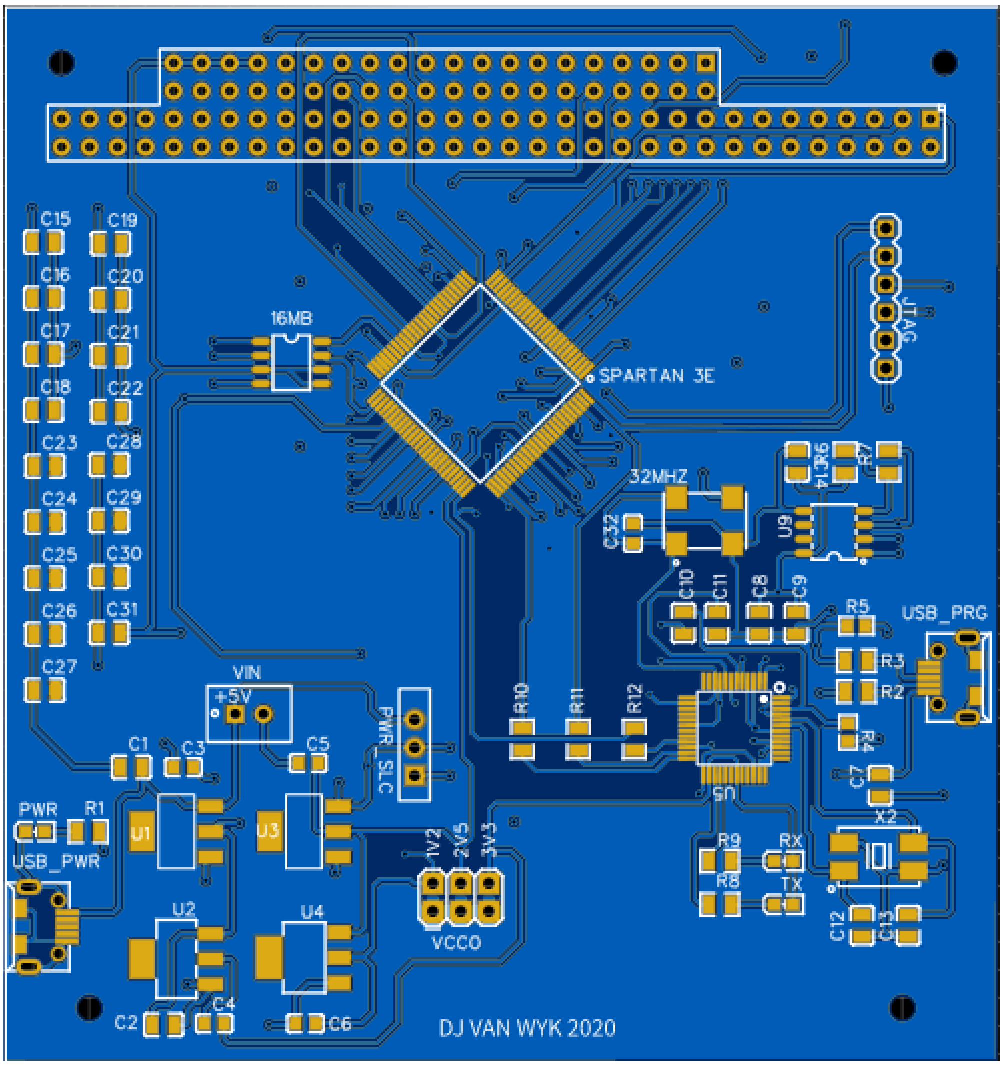



TITLE:
ELECTRICAL POWER SYSTEM
REV: 1.0
Company: DVW
Date: 2020
Drawn By: Dvwyk
Sheet: 1/1
EasyEDA
USB_PWR1
microUSBFemale
V+
D-
D+
ID
V-
VIN1
PWRIN
V1_USB
GND
U17
AMS1117-5.0
In
GND
TAB
Out
C38
100nF
C36
10uF
5V
PWR1
PWR
R13
1.8K
U16
AMS1117-3.3
GND/ADJ
Vout
Vin
C37
100nF
C35
10uF
3V3
U14
AMS1117-1.2
C33
10uF
1V2
2V5
U15
AMS1117-2.5_C166064
VOUT
VIN
EP
C34
10uF
U19.1
SPARTAN 3E
U18.2
XC3S250E-4VQ100C
VCCO_0
VCCO_1
VCCO_2
VCCO_3
VCCAUX
VCCINT
VCCO
C39
C40
C41
C42
C43
C44
C45
C46
C47
C48
C49
C50
C51
C52
C53
C54
C55
100nF
A
B
C
D
1
2
3
4
5

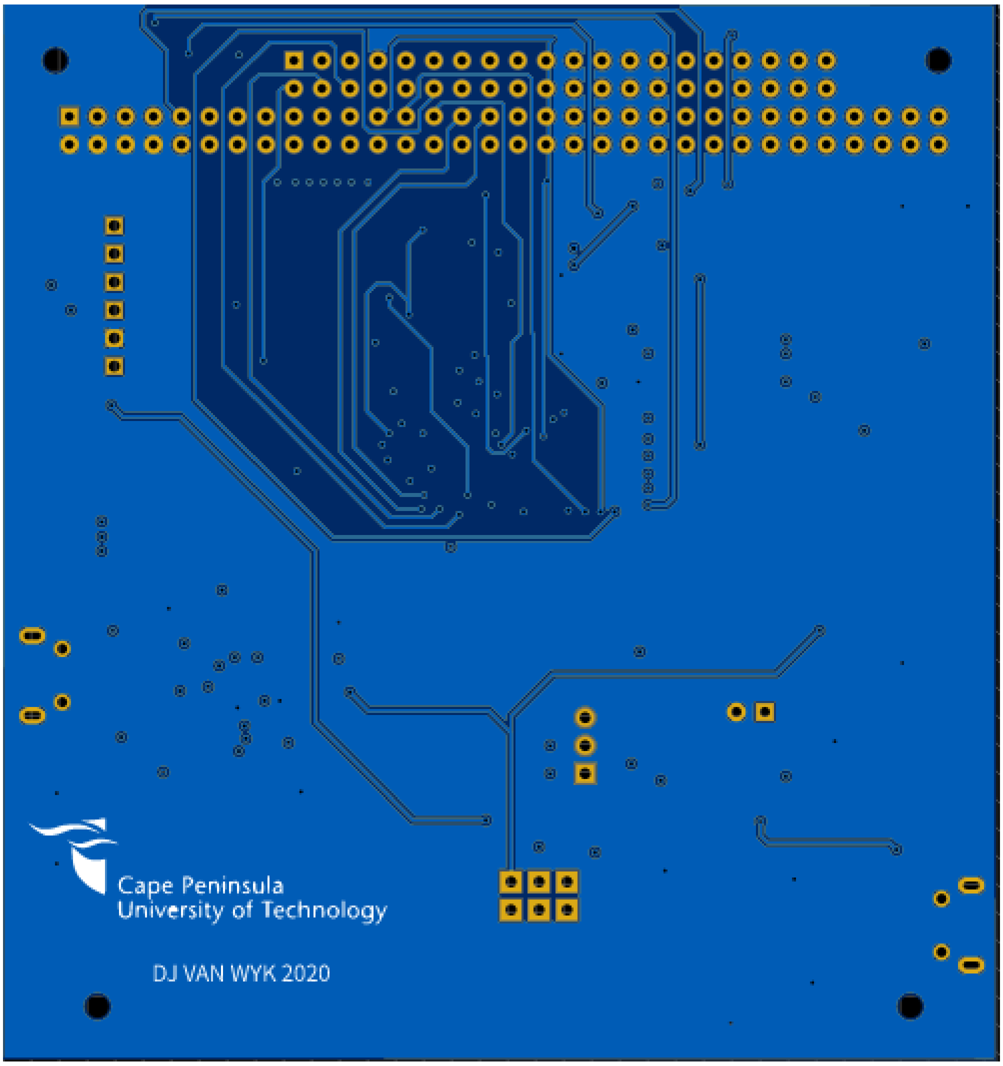
71


TITLE: FPGA BANK ARRANGEMENT
Company: DVW
Date: 2020
Drawn By: Dvwyk
REV: 1.0
Sheet: 1/1
EasyEDA
U13
PC 104
3V3
GND
VCCO
VCCO1
1V2
2V5
3V3
U6.8
XC3S250E-4VQ100C
BANK 0
U8.3
XC3S250E-4VQ100C
BANK 1
U12.4
XC3S250E-4VQ100C
BANK 2
U11.5
XC3S250E-4VQ100C
BANK 3

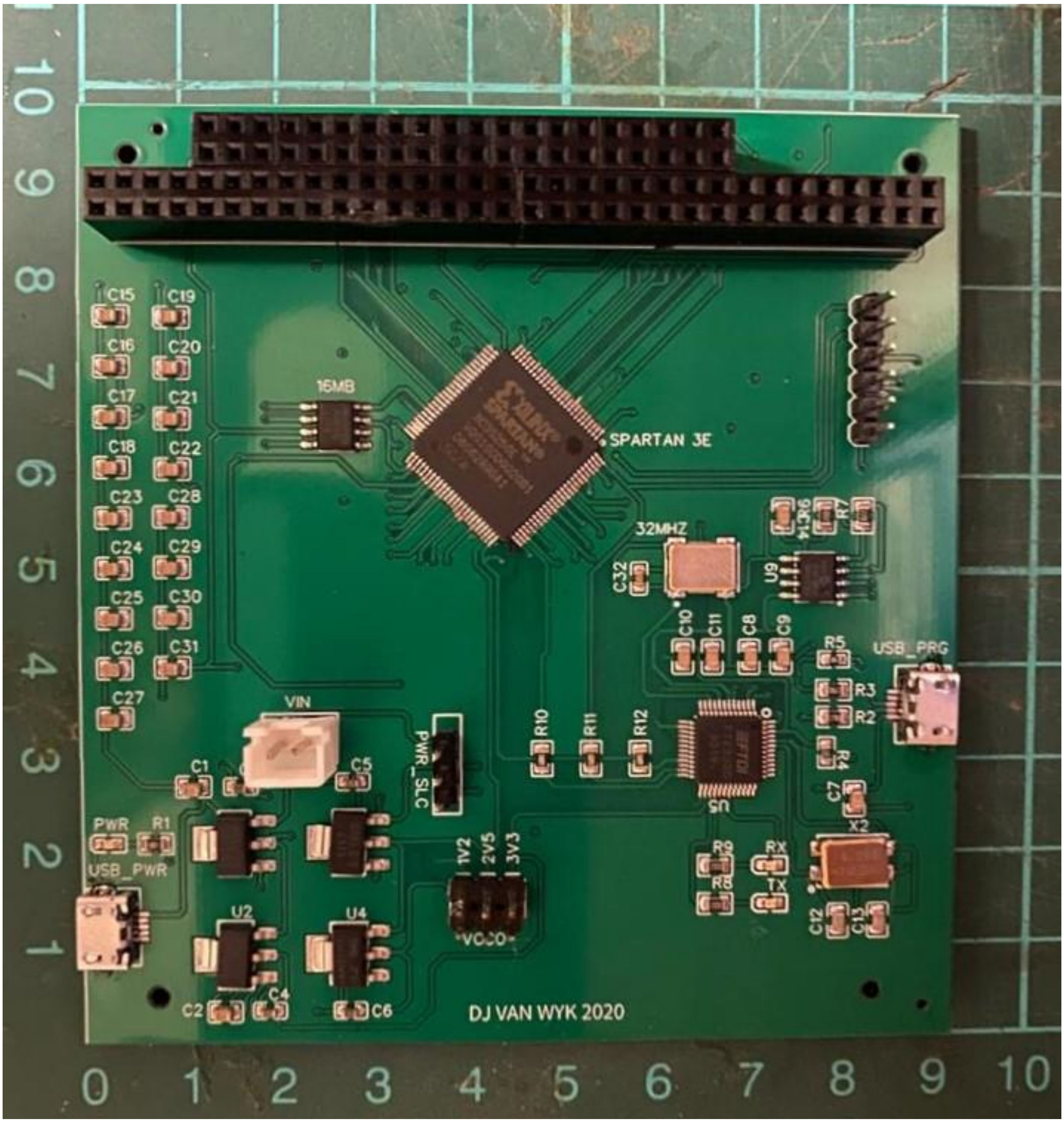
Annexure B



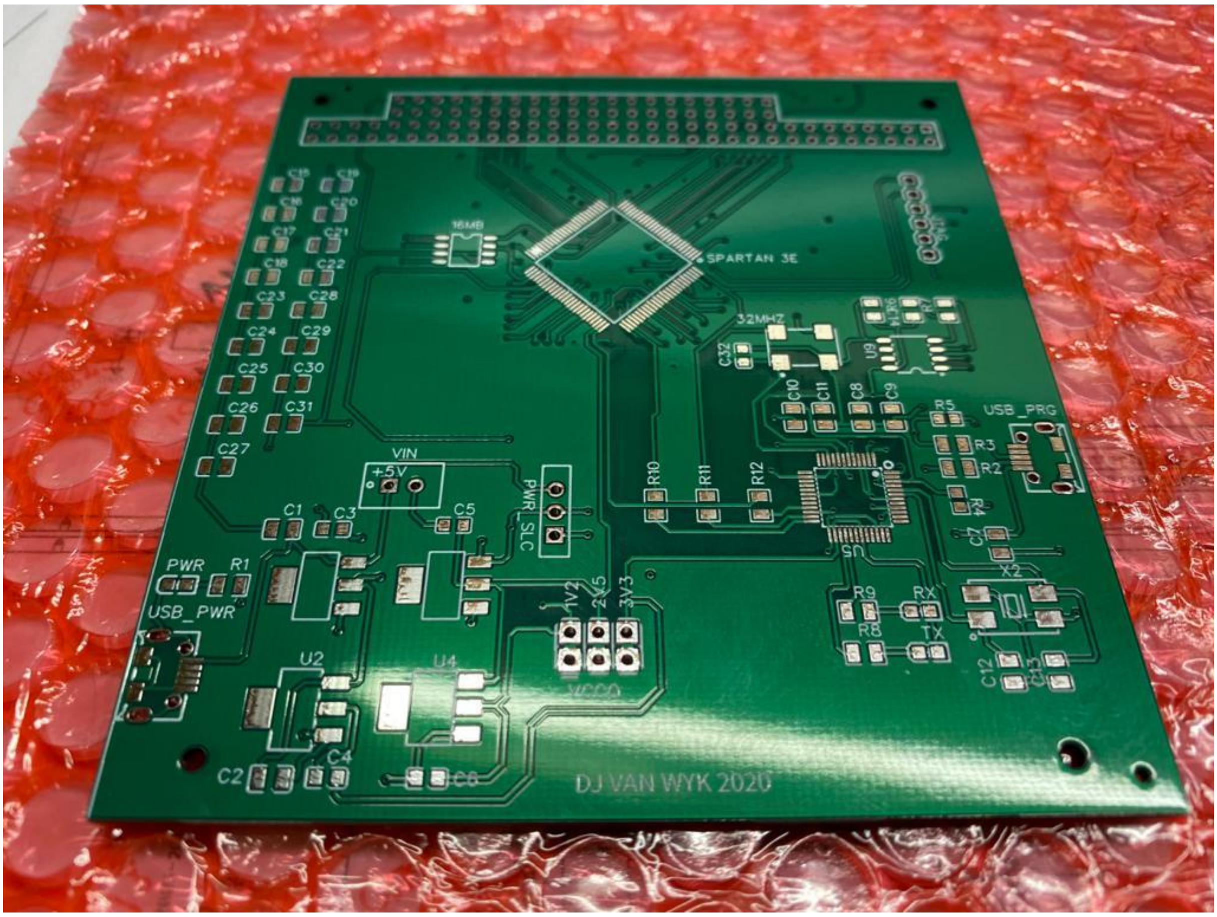
FPGA OBC Top layer of PCB



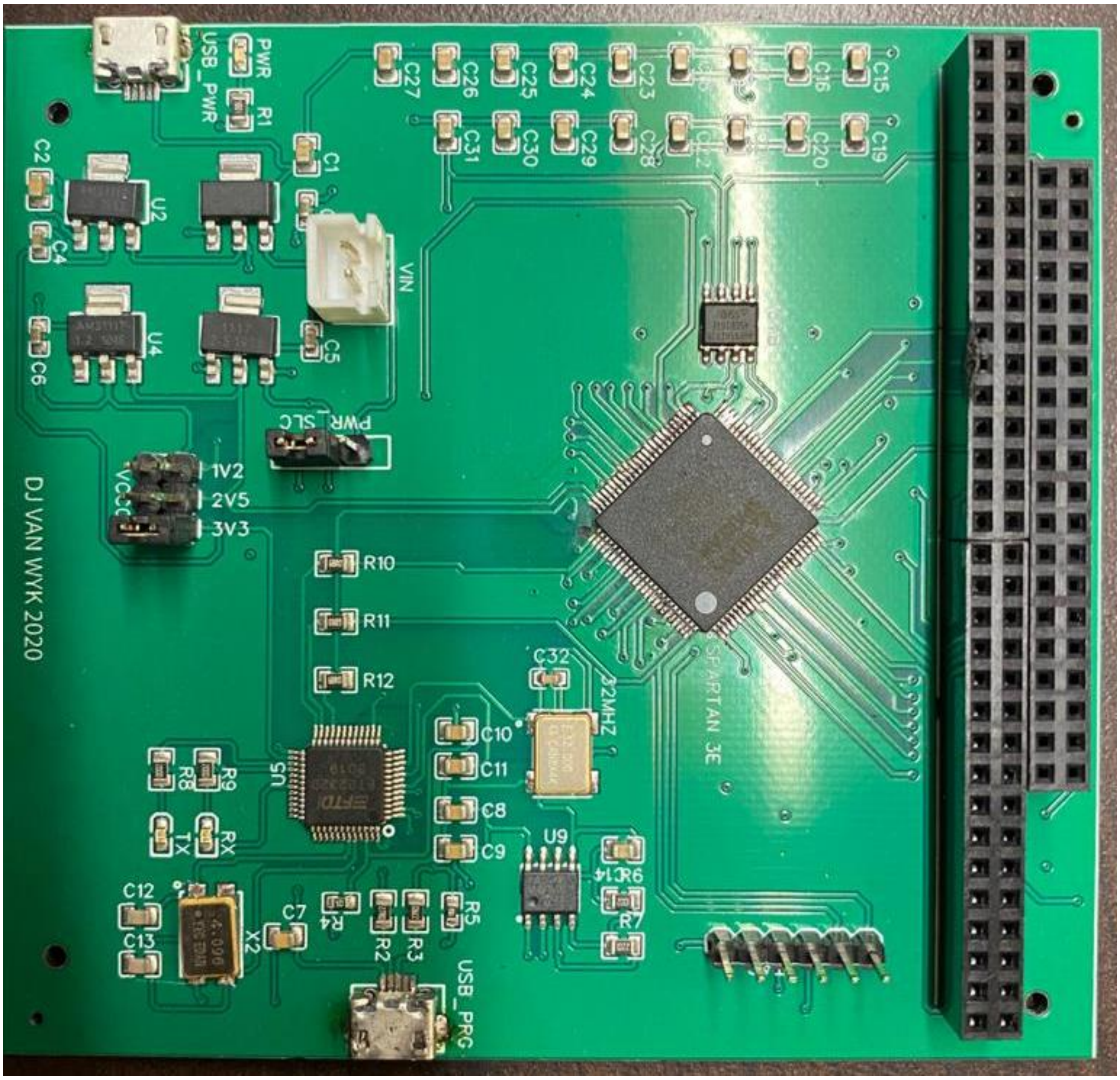
FPGA OBC Bottom layer of PCB



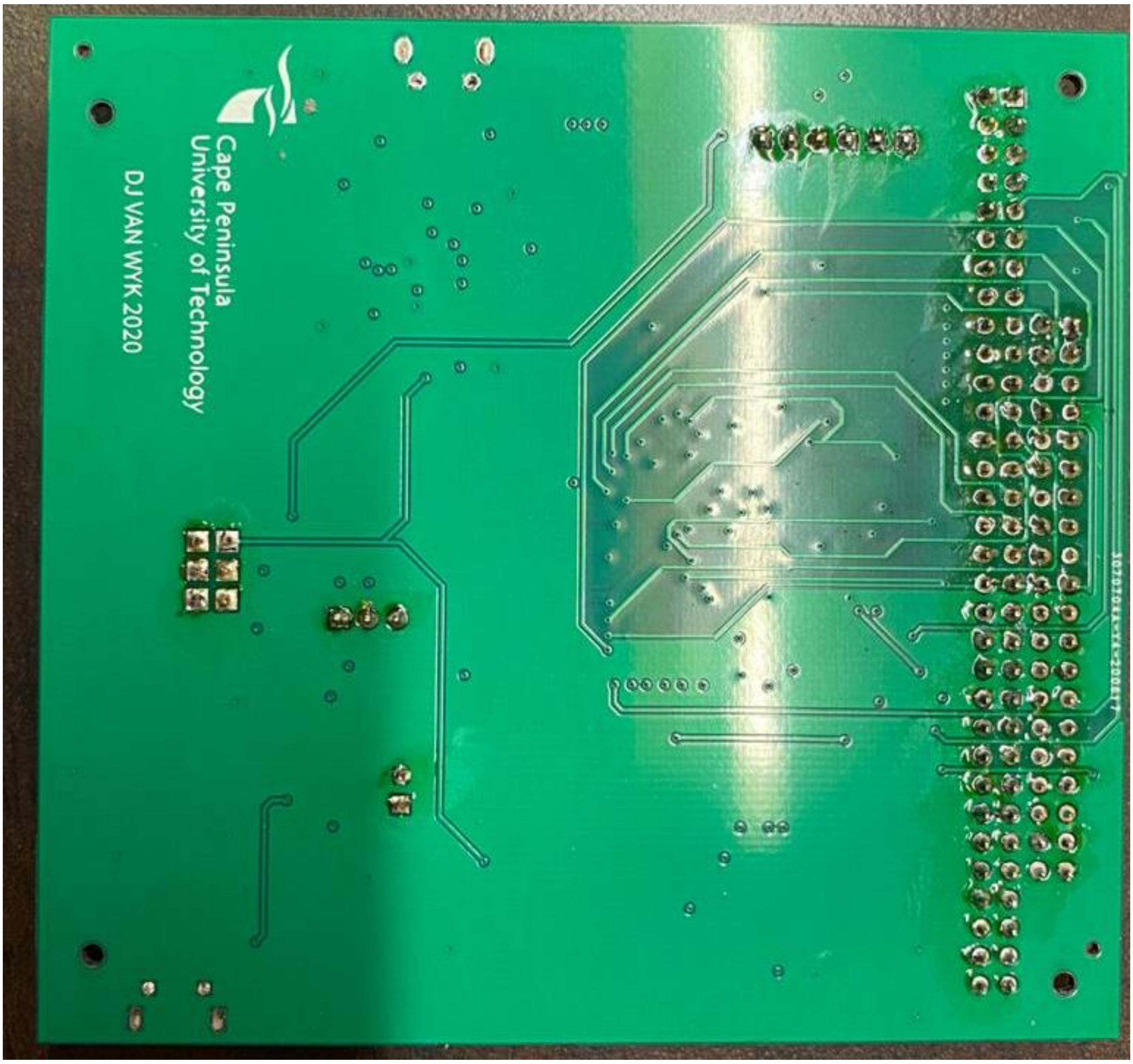
Complete FPGA based OBC for small satellites, PC/104 form factor.



Proposed FPGA OBC clean PCB manufactured into a 4-layer PCB.



PCB Soldered with all components and PC/104 form factor header pin.



Bottom side of the finish PCB for the FPGA based OBC.

Annexure C

```
--------------------------------------------------------------------------------
-- Company: DVW 217222854
-- Engineer: Dirk Jacobus van Wyk
--
-- Create Date:   12:54:36 07/25/2020
-- Design Name:
-- Module Name:   E:/Thesis Books/BASYS 2/FIFO/TB_STD_FIFO.vhd
-- Project Name:  FIFO
-- Target Device:  Spartn-3E OBC
-- Tool versions:
-- Description:   Write and Read Memory Test, varified with Basys2 dev board
--
-- VHDL Test Bench Created by ISE for module: STD_FIFO


--------------------------------------------------------------------------------
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.NUMERIC_STD.ALL;

ENTITY TB_STD_MYFIFO IS
END TB_STD_MYFIFO;

ARCHITECTURE behavior OF TB_STD_FIFO IS


        -- Module Statement used by the Unit Under Test
        component STD_MYFIFO
                Generic (
                        constant MYDATA_WIDTH  : positive := 16;
                        constant MYFIFO_DEPTH : positive := 1024
                );
                port (
            CCLK            : in std_logic;
            RSET            : in std_logic;
            MyDataIn        : in std_logic_vector(15 downto 0);
            Write_EN        : in std_logic;
            Read_EN         : in std_logic;
            MyDataOut       : out std_logic_vector(15 downto 0);
            MemFull         : out std_logic;
            MemEmpty        : out std_logic
        );
    end component;

    --Input Declaration
    signal CCLK                 : std_logic := '0';
    signal RSET                 : std_logic := '0';
    signal MyDataIn: std_logic_vector(15  downto 0) := (others => '0');
    signal Read_EN : std_logic := '0';
    signal Write_EN: std_logic := '0';

    --Output Declaration
    signal MyDataOut            : std_logic_vector(15 downto 0);
    signal MemEmpty             : std_logic;
    signal MemFull              : std_logic;

    -- Clock sample time definition
    constant CCLK_PD : time := 10 ns;

BEGIN

    -- Starting the Unit Under Test
    uut: STD_MYFIFO
        PORT MAP (
            CCLK            => CCLK,
                    RSET            => RSET,
                    MyDataIn        => MyDataIn,
                    Write_EN        => Write_EN,
                    Read_EN         => Read_EN,
                    MyDataOut       => MyDataOut,
                    MemFull         => MemFull,
                    MemEmpty        => MemEmpty
        );

-- Clock procedure designations
CCLK_PS : process
begin
        CCLK <= '0';
        wait for CCLK_PD/2;
        CCLK <= '1';
        wait for CCLK_PD/2;
end process;

-- Reset sequence
REST_Pro : process
begin
wait for CCLK_PD * 5;

        RST <= '1';

        wait for CCLK_PD * 5;

        RSET <= '0';

        wait;
end process;
```

```
-- Write sequence
WR_Pro : process
        variable counter : unsigned (15 downto 0) := (others => '0');
begin
        wait for CCLK_PD * 20;

        for i in 1 to 1024 loop
                counter := counter + 1;

                MyDataIn <= std_logic_vector(counter);
                        wait for CCLK_PD * 1;

                Write_EN <= '1';

                wait for CCLK_PD * 1;

                Write_EN <= '0';
        end loop;

        wait for CCLK_PD * 20;



        wait;
end process;

-- Read sequence
RE_Pro : process
begin

        wait for CCLK_PD * 1000;
```

```
                Read_EN <= '1';

                wait for CCLK_PD * 60;

                Read_EN <= '0';



                wait;
        end process;

END;

----------------------------------------------------------------------------------
-- Company: 217222854
-- Engineer: Dirk Jacobus van Wyk
--
-- Create Date:    12:17:13 07/25/2020
-- Design Name:                    SPARTAN 3E - OBC
-- Module Name:    STD_FIFO - BehavioralTest
-- Project Name:
-- Target Devices: BASYS2 DEV BOARD VERIFIED ON CUSTOM FPGA OBC
-- Tool versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
```

```
library IEEE;

USE IEEE.STD_LOGIC_1164.ALL;

USE IEEE.NUMERIC_STD.ALL;


entity STD_MYFIFO is

        Generic (

                constant Data_WID  : positive := 16;

                constant MYFIFO_DEP   : positive := 1024

        );

        Port (

                CCLK            : in  STD_LOGIC;

                RSET            : in  STD_LOGIC;

                Write_EN        : in  STD_LOGIC;

                MyDataIn        : in  STD_LOGIC_VECTOR (Data_WID - 1 downto 0);

                Read_EN         : in  STD_LOGIC;

                MyDataOut       : out STD_LOGIC_VECTOR (Data_WID - 1 downto 0);

                MemEmpty        : out STD_LOGIC;

                MemFull         : out STD_LOGIC

        );

end STD_MYFIFO;


architecture Behavioral of STD_MYFIFO is


begin


        MyFIFO_Proc : process (CCLK)

        --data type statement

        --4-bit memory FIFO test

                type MyFIFO_Mem is array (0 to MYFIFO_DEP - 1) of STD_LOGIC_VECTOR (
Data_WID - 1 downto 0);

                variable F_Memory : MyFIFO_Mem;
                variable F_Head : natural range 0 to MYFIFO_DEP - 1;
                variable F_Tail : natural range 0 to MYFIFO_DEP - 1;

                variable FLooped : boolean; --Loop Full when True
        begin
                if rising_edge(cCLK) then
                        if RSeT = '1' then
                                F_Head := 0;
                                F_Tail := 0;

                                FLooped := false;

                                MemFull  <= '0';
                                MemEmpty <= '1';
                        else
                        --Read sequence
                                if (Read_EN = '1') then
                                        if ((FLooped = true) or (F_Head /= F_Tail)) then
                                                -- Correct data out memory
                                                MyDataOut <= F_Memory(F_Tail);

                                                -- Correct FIFO tail indicator as needed
                                                if (F_Tail = MYFIFO_DEP - 1) then
                                                        F_Tail := 0;

                                                        FLooped := false;
                                                else
                                                        F_Tail := F_Tail + 1;
                                                end if;


                                        end if;
```

```
            end if;
            -- Write sequence
            if (Write_EN = '1') then
                if ((FLooped = false) or (F_Head /= F_Tail)) then
                    -- Writing data to memory
                    F_Memory(F_Head) := MyDataIn;

                    -- Increase F_Head indicator as needed
                    if (F_Head = MYFIFO_DEP - 1) then
                        F_Head := 0;

                        FLooped := true;
                    else
                        F_Head := F_Head + 1;
                    end if;
                end if;
            end if;

            -- Update MemEmpty and MemFull flags
            if (F_Head = F_Tail) then
                if FLooped then
                    MemFull <= '1';
                else
                    MemEmpty <= '1';
                end if;
            else
                MemEmpty        <= '0';
                MemFull         <= '0';
            end if;
        end if;
    end if;
end process;
end Behavioral;
```

Annexure D

```
--------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date:   17:51:09 07/25/2020
-- Design Name:
-- Module Name:   E:/Thesis Books/BASYS 2/UART/UART_RX_TB.vhd
-- Project Name:  UART
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: UART_RX
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
-- Notes:
-- The testbench has been repeatedly generated using types std_logic and
-- std_logic_vector for the ports of the unit under test.  Xilinx recommends
-- that these types to always be used for the top-level I/O of a design in order
-- to assure that the testbench will bind correctly to the post-implementation
-- simulation model.
--------------------------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.ALL;
use ieee.numeric_std.all;
```

```
entity UART_RX_TB is
end UART_RX_TB;

architecture U_Behave of UART_RX_TB is

  -- Test Bench uses an 25 MHz Clock
  constant c_CLK_PER : time := 40 ns;

  -- Change the boundary to 115200 baud for UART
  -- 25000000 / 115200 = 217 Clocks Per Bit.
  constant U_CLKS_BIT : integer := 217;

  -- 1/115200:
  constant U_BIT_PER : time := 8680 ns;

  signal RE_Clock     : std_logic                    := '0';
  signal WR_RX   : std_logic_vector(7 downto 0);
  signal RE_RX : std_logic := '1';



  procedure for UART_WR_BYTE (
    My_DataIn       : in  std_logic_vector(7 downto 0);
    signal U_Serial : out std_logic) is
  begin

    -- Sending Start Bit
    U_Serial <= '0';
    wait for U_BIT_PER;
    -- Transferring Data Byte
    for ii in 0 to 7 loop
      U_Serial <= My_DataIn(ii);
      wait for U_BIT_PER;
    end loop;

    -- Transferring Stop Bit
    U_Serial <= '1';
    wait for U_BIT_PER;
  end UART_WR_BYTE;

begin

  -- Executing UART Receiver
  UART_RX : entity UART_RX
    generic map (
      GE_CLKS => U_CLKS_BIT
      )
    port map (
      IN_Clk       => RX_Clock,
      IN_RX_Serial => RE_RX_Serial,
      OP_RX_DV     => open,
      OP_RX_Byte   => WR_RX
      );

  RX_Clock <= not RX_Clock after CCLK_PD/2;

  process is
  begin
    -- Transmitt a command to the UART
    wait until rising_edge(RX_Clock);
    UART_WR_BYTE(X"3F", RE_RX_Serial);
    wait until rising_edge(RX_Clock);

    -- Check transmitted correct signal was received
    if OP_RX_Byte = X"3F" then
      report "Correct Byte Transmitted, well received" severity note;
    else
      report "Incorrect Byte tranmitted" severity note;
    end if;

    assert false report "Transmitting test concluded" severity failure;

  end process;

end Behave;

----------------------------------------------------------------------
-- This file holds the UART Receiver code.  This UART receiver is able to
-- receive 8-bits of serial data, one start bit, one stop bit,
-- with no parity bit.  When receive is complete OP_RX will be
-- directed high for one clock cycle.
--
-- Set Generic GE_CLKS as follows:
-- GE_CLKS = (Frequency of IN_Clk)/(Freq_UART)
-- 25 MHz frequency clock at 115200 baud rate for UART
-- (25000000)/(115200) = 217
--
library ieee;
use ieee.std_logic_1164.ALL;
use ieee.numeric_std.all;
```

```
entity myUART_RX is
  generic (
    GE_CLKS : integer := 217
    );
  port (
    IN_Clk       : in  std_logic;
    IN_RX : in  std_logic;
    OP_RX     : out std_logic;
    OP_RX_Byte   : out std_logic_vector(7 downto 0)
    );
end myUART_RX;


architecture Behave_RTL of myUART_RX is

  type SM_Main is (F_Idle, F_RX_Sr_Bit, F_RX_Data,
                   F_RX_Stp_Bit, F_Cleanup);
  signal SM_Main : F_SM_Main := F_Idle;

  signal RE_Clk_Count : integer range 0 to g_CLKS_PER_BIT-1 := 0;
  signal RE_Bit_Index : integer range 0 to 7 := 0;  -- 8 Bits Total
  signal RE_RX_Byte   : std_logic_vector(7 downto 0) := (others => '0');
  signal RE_RX_DV     : std_logic := '0';

begin

  -- Controlling the RX state-machine
  p_RX : process (IN_Clk)
  begin
    if rising_edge(IN_Clk) then
```

```
case F_SM_Main is

 when F_Idle =>
  RE_RX_DV     <= '0';
  RE_Clk_Count <= 0;
  RE_Bit_Index <= 0;

  if IN_RX = '0' then
   F_SM_Main <= F_RX_Sr_Bit;
  else
   F_SM_Main <= F_Idle;
  end if;


 -- Verifing the mid of start bit making sure it's still at 'low'
 when F_RX_Sr_Bit =>
  if RE_Clk_Count = (GE_CLKS -1)/2 then
   if IN_RX = '0' then
    RE_Clk_Count <= 0;  -- reseting counter at the middle of the state machine
    F_SM_Main   <= F_RX_Bits;
   else
    F_SM_Main   <= F_Idle;
   end if;
  else
   RE_Clk_Count <= RE_Clk_Count + 1;
   F_SM_Main   <= F_RX_Sr_Bit;
  end if;


 -- Wait for GE_CLKS -1 clock cycles to sample the serial data
 when F_RX_Bits =>
```

```
  if RE_Clk_Count < GE_CLKS -1 then
    RE_Clk_Count <= RE_Clk_Count + 1;
    F_SM_Main   <= F_RX_Bits;
  else
    RE_Clk_Count           <= 0;
    RE_RX_Byte(RE_Bit_Index) <= IN_RX;

    -- Are all bits transmitted out?
    if RE_Bit_Index < 7 then
      RE_Bit_Index <= RE_Bit_Index + 1;
      F_SM_Main   <= F_RX_Bits;
    else
      RE_Bit_Index <= 0;
      F_SM_Main   <= F_RX_Stp_Bit;
    end if;
  end if;



-- Stop bit received = 1
when F_RX_Stp_Bit =>
  -- Wait for GE_CLKS -1 clock cycles for Stop bit to complete cycle
  if RE_Clk_Count < GE_CLKS -1 then
    RE_Clk_Count <= RE_Clk_Count + 1;
    F_SM_Main   <= F_RX_Stp_Bit;
  else
    RE_RX_DV     <= '1';
    RE_Clk_Count <= 0;
    F_SM_Main   <= F_Cleanup;
  end if;
```

```
    -- Stay at 1 clock cycle
    when F_Cleanup =>
     F_SM_Main <= F_Idle;
     RE_RX_DV   <= '0';


    when others =>
     F_SM_Main <= F_Idle;

   end case;
  end if;
 end process p_RX;

 OP_RX_DV   <= RE_RX_DV;
 OP_RX_Byte <= RE_RX_Byte;

end RTL;
```

--End of Document---