FEDERAL UNIVERSITY OF SANTA CATARINA
TECHNOLOGICAL CENTER
ELECTRICAL AND ELECTRONICS ENGINEERING DEPARTMENT

Development of a satellite radiation experiment for evaluating SDRAM memories in harsh environments

Undergraduate Dissertation presented to the Federal University of Santa Catarina as a requisite for the bachelor degree of Electronics Engineering

André Martins Pio de Mattos

Supervisor: Eduardo Augusto Bezerra, Ph.D.
Co-Supervisor: Luigi Dilillo, Dr.

Florianópolis, 2021

ANDRÉ MARTINS PIO DE MATTOS

DEVELOPMENT OF A SATELLITE RADIATION EXPERIMENT FOR EVALUATING SDRAM MEMORIES IN HARSH ENVIRONMENTS

Undergraduate Dissertation presented to the Federal University of Santa Catarina as a requisite for the bachelor degree of Electronics Engineering. Supervisor: Eduardo Augusto Bezerra. Co-Supervisor: Luigi Dilillo.

FLORIANÓPOLIS
2021

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Mattos, André Martins Pio de
Development of a satellite radiation experiment for evaluating SDRAM memories in harsh environments / André Martins Pio de Mattos ; orientador, Eduardo Augusto Bezerra, coorientador, Luigi Dilillo, 2021.
100 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia Eletrônica, Florianópolis, 2021.

Inclui referências.

1. Engenharia Eletrônica. 2. Sistemas embarcados. 3. Experimentos espaciais. 4. Ensaios de radiação. 5. Satélites. I. Bezerra, Eduardo Augusto. II. Dilillo, Luigi. III. Universidade Federal de Santa Catarina. Graduação em Engenharia Eletrônica. IV. Título.

André Martins Pio de Mattos

DEVELOPMENT OF A SATELLITE RADIATION EXPERIMENT FOR EVALUATING SDRAM MEMORIES IN HARSH ENVIRONMENTS

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção do título de Bacharel em Engenharia Eletrônica e aprovado em sua forma final pela Banca Examinadora.

Florianópolis, 24 de setembro de 2021

ICP-Edu
Documento assinado digitalmente
Fernando Rangel de Sousa
Data: 05/10/2021 14:30:12-0300
CPF: 884.649.114-91
Verifique as assinaturas em https://v.ufsc.br

Prof. Fernando Rangel de Sousa, Dr.
Coordenador do Curso

Banca Examinadora:

ICP-Edu
Documento assinado digitalmente
Eduardo Augusto Bezerra
Data: 30/09/2021 08:47:50-0300
CPF: 830.851.577-00
Verifique as assinaturas em https://v.ufsc.br

Prof. Eduardo Augusto Bezerra, Ph.D.
Orientador
Universidade Federal de Santa Catarina

Vargas

Prof. Fabian Luis Vargas, Dr.
Pontifícia Universidade Católica do Rio Grande do Sul

Documento assinado digitalmente
gov.br
FABIO BATAGIN ARMELIN
Data: 05/10/2021 07:24:28-0300
Verifique em https://verificador.iti.br

Fabio Batagin Armelin, Dr.
Instituto Nacional de Pesquisas Espaciais

Acknowledgments

This work is dedicated to everyone that allowed me to accomplish my journey up to this point in life. I am grateful for all the support from my family, my girlfriend, and friends. Also, for the opportunity that my supervisor gave me to develop this work in a great international partnership environment. I wish to extend a huge thank you to my laboratory colleagues for the insightful discussions and learning experiences, the institutions that I have developed my research, and my internship supervisor that supported this work.

Abstract

Electronics in the context of space applications require attention to performance and reliability under harsh radiation conditions. This work proposes the development of a platform capable of evaluating the radiation effects in three different generations of a same synchronous dynamic random-access memory in test facilities and in a real space mission. The results are interesting to characterize this devices for future space applications and the platform could be used as a radiation sensor in future missions.

Keywords: satellite, CubeSat, radiation, payload, SDRAM, critical applications, harsh environments, space, experiment, embedded systems.

Resumo

Sistemas eletrônicos no contexto de aplicações espaciais requerem esforços adicionais quanto ao desempenho e confiabilidade em ambientes com alta radiação. Este trabalho propõe o desenvolvimento de uma plataforma capaz de avaliar os efeitos da radiação em três gerações diferentes de uma mesma memória dinâmica síncrona de acesso aleatório em laboratórios de teste e em missões espaciais reais. Os resultados são interessantes para caracterizar esses dispositivos para futuras aplicações espaciais e também a plataforma pode ser reutilizada como um sensor de radiação em futuras missões.

Palavras-chave: satélite, CubeSat, radiação, carga útil, SDRAM, aplicações críticas, espaço, experimento, sistemas embarcados.

Resumo Estendido

O crescente mercado de pequenos satélites [1] está liderando uma rápida evolução da tecnologia e empurrando os limites de tamanho, consumo de energia, confiabilidade e recursos dos sistemas embarcados. O CubeSat, um padrão de satélite, e suas variantes estão surgindo como modelos promissores para diversas aplicações em sensoriamento remoto, telecomunicações, observação da Terra e até exploração espacial.

Comparando com os outros padrões e satélites de grande porte, que usam componentes customizados para o uso espacial e requerem um esforço considerável no desenvolvimento e lançamento, o CubeSats tem uma vantagem devido ao uso de componentes comercialmente disponíveis para outros setores (COTS) e a reduzida logística envolvida que impacta diretamente no custo da missão.

Apesar destas vantagens, as missões com CubeSats ainda não atingem todas as aplicações do setor espacial, devido ao escopo ou mesmo a uma inviabilidade tecnológica, reforçando que as missões convencionais ainda têm relevância e importância no assunto. Este cenário traz um novo horizonte para os avanços tecnológicos e incentiva o uso generalizado de pequenos satélites para resolver problemas desafiadores e melhorar a compreensão do espaço.

O ambiente enfrentado nessas missões espaciais é hostil devido às dramáticas variações de temperatura, vácuo, doses de radiação e estresse mecânico durante o lançamento. Esses parâmetros atingem níveis críticos ao considerar missões de longo prazo ou órbitas de altitude mais elevada. A maioria das missões de CubeSat são lançadas em perfis de órbita de baixa altitude, onde as condições são menos exigentes, mas ainda assim perigosas e onerosas para o desenvolvimento. Além disso, a utilização de componentes COTS sem cautela e análise prévia pode levar a falhas catastróficas como consequência da criação de pontos de falha e imprevisibilidade no sistema. Por esse motivo, diversos estudos estão sendo realizados para identificar e quantificar as vulnerabilidades desses componentes no ambiente espacial.

A condição mais desafiadora a superar é a quantidade significativa de radiação na qual esses sistemas são expostos fora da proteção do campo magnético terrestre. Desde o lançamento, o satélite está exposto a um amplo espectro de interações de radiação aleatórias com diversas magnitudes e consequências [2]. Para minimizar esses efeitos, diferentes abordagens são adotadas em várias camadas, desde a própria arquitetura de hardware até complexas técnicas de correção de erros e redundâncias de software.

Assim, avaliar a vulnerabilidade desses componentes devido à exposição à radiação e caracterizar os limiares de falha crítica são essenciais para determinar os esforços que devem ser aplicados durante o desenvolvimento. As memórias de circuito integrado, em uma ampla variedade de tecnologias e características operacionais, são um dos componentes críticos mais afetados pelas altas doses de radiação [3].

Os esforços deste trabalho se concentram no desenvolvimento de uma plataforma capaz de realizar a caracterização da radiação de diferentes memórias SDRAM em condições espaciais. Esta plataforma testará estes chips no ambiente espacial, participando da missão FloripaSat-2. As memórias utilizadas foram previamente caracterizadas em experimentos de laboratório, então, ao expor esses dispositivos a ambientes reais e executar as mesmas rotinas de teste, não apenas será possível gerar mais resultados para análise, mas também fornecerá uma oportunidade para avaliar as próprias metodologias de teste. Além disso, após coletar dados suficientes para serem analisados, essa carga útil poderá ser usada para fornecer um status da missão, no que diz respeito às doses de radiação a que o satélite foi exposto.

A arquitetura de carga útil consiste nos seguintes módulos: um subsistema de controle e gerenciamento do experimento, operado por um System-On-a-Chip (SoC); conversores de energia para fornecimento dos níveis de tensão adequados; monitores de curto-circuito; barramentos e interface de comunicação; e os chips de memória SDRAM em análise.

O trabalho está estruturado nas seguintes seções: contextualização (background), fornecendo uma explicação simplificada dos principais tópicos abordados neste trabalho; análise da missão (mission analysis), descrevendo o contexto em que o experimento é realizado e definindo os requisitos para o desenvolvimento deste trabalho; desenvolvimento (development), apresentando todos os sistemas desenvolvidos e integrados para criar a plataforma e discutindo suas motivações, estratégias e diretrizes; resultados (results), descrevendo os testes executados e seus resultados para validar o funcionamento da plataforma; e conclusão (conclusion), apresentando as considerações finais. O corpo do trabalho será descrito ao longo das seções do texto original no idioma inglês.

Acronyms

General Nomenclature

CAD	*Computer-Aided Design.*
COTS	*Commercial Of-The-Shelf.*
ESA	*European Space Agency.*
FPGA	*Field-Programmable Gate Array.*
HDL	*Hardware Description Language.*
IC	*Integrated Circuit.*
LEO	*Low-Earth Orbit.*
LIRMM	*Laboratory of Computer Science, Robotics and Microelectronics of Montpellier.*
OBC	*On-Board Computer.*
OBDH	*On-Board Data Handling.*
PCB	*Printed Circuit Board.*
RADEF	*RADiation Effects Facility.*
RTOS	*Real-Time Operating System.*
SAA	*South Atlantic Anomaly.*
SoC	*System-On-A-Chip.*
SpaceLab	*Space Technology Research Laboratory.*
VESPER	*Very energetic Electron facility for Space Planetary Exploration missions in harsh Radiative environments.*

Memory Technologies

DDR	*Double Data Rate.*
eNVM	*Embedded Non-Volatile Memory.*
eSRAM	*Embedded Static Random-Access Memory.*
Flash	*Flash is a non-volatile memory.*
FRAM	*Ferroelectric Random-Access Memory.*
MRAM	*Magnetoresistive Random-Access Memory.*
NOR Flash	*Flash based on NOR logic gates.*
SDRAM	*Synchronous Dynamic Random-Access Memory.*
SDR	*Single Data Rate.*

Radiation Effects

DD	*Displacement Damage.*
MBU	*Multiple-Bit Upset.*
MCU	*Multiple-Cell Upset.*
SBU	*Single-Bit Upset.*
SEB	*Single Event Burnout.*

SEE	*Single Event Effect.*
SEFI	*Single Event Functional Interruption.*
SEGR	*Single Event Gate Rupture.*
SEL	*Single Event Latch-up.*
SET	*Single Event Transient.*
TID	*Total Ionizing Dose.*

Communication Protocols

AHB	*Advanced High-performance Bus.*
AMBA	*ARM Advanced Microcontroller Bus Architecture.*
AXI	*Advanced eXtensible Interface.*
FSP	*FloripaSat Protocol.*
I2C	*Inter-Integrated Circuit.*
JTAG	*Joint Test Action Group.*
SPI	*Serial Peripheral Interface.*
UART	*Universal Asynchronous Receiver-Transmitter.*

Error Mitigation Techniques

CRC	*Cyclic Redundancy Check.*
SECDED	*Single Error Correction, Double Error Detection.*

Specific Nomenclature

BFM	*Bus Functional Model.*
BGA	*Ball Grid Array.*
GPIO	*General Purpose Input and Output.*
MSS	*Microcontroller SubSystem.*
SMC	*Soft Memory Controller.*
TSOP	*Thin Small Outline Package.*

List of Figures

3.1 Conceptual exploded view of a CubeSat (FloripaSat-2). . . . 7
3.2 Simplified FPGA internal architecture. 10
3.3 Conceptual PCB board. 12
4.1 A preview of the satellite simulated groundtrack. 14
4.2 Schematic overview of the Van Allen radiation belts. 14
4.3 Simplified preview of the radiation vulnerable regions. 15
4.4 Inside view of the SDRAM memory chip (IS42S16320F). . . 16
5.1 Preview of the developed payload. 19
5.2 Development board used during the prototype phase. 21
5.3 Simplified engineering model architecture. 22
5.4 Conceptual payload model integrated with the FloripaSat-2 OBDH. 23
5.5 Payload model integrated with FloripaSat-2 OBDH. 23
5.6 FPGA die overview. 24
5.7 Internal architecture diagram overview. 26
5.8 Memory controller interfaces overview. 27
5.9 Core SDR wrapped for an AXI interface architecture. 28
5.10 Core SDR Address scheme overview. 28
5.11 Core SDR architecture diagram overview. 29
5.12 eSRAM memory scheme. 30
5.13 System abstraction layers diagram. 34
5.14 Simplified system execution flow diagram. 35
5.15 System architecture diagram. 36
5.16 Experiment architecture diagram. 38
5.17 Communication interrupt and task architecture diagram. . . 39
5.18 Board stackup planning. 46
5.19 FPGA pinout usage. 47
5.20 Top Mid-layer routing with focus on the memory address signals. 48
5.21 Bottom Mid-layer routing with focus on the memory address signals. 48
5.22 Hardware architecture overview. 49
5.23 Power architecture overview. 50
5.24 PCB 3D rendering overview. 50
5.25 PCB manufactured overview. 51
5.26 FPGA fanout dog-bone pattern and decoupling capacitors implementation. 51

6.1 Memory controller simulation flow. 55
6.2 Memory controller simulation results: log output. 56
6.3 Memory controller simulation results: signal waveforms. . . . 57
6.4 Hardware test results: fully operational board. 58
6.5 Hardware test results: average power consumption. 58
6.6 Example of a firmware log message. 59
6.7 Experiment radiation parameters. 60
6.8 Radiation experiment setup diagram. 61
6.9 Payload board in red fixed inside the beam area in blue. . . 61
6.10 Radiation cross section results according with the SDRAM memories technology scaling (models B, D, and F). 62
6.11 Integration setup with FloripaSat-1 OBDH engineering model 63
6.12 Engineering model payload integrated with the FloripaSat-2 OBDH. 64
6.13 Example SPI command communication signals from the OBC to the payload using the FSP protocol. 64
6.14 Example SPI acknowledgement answer communication signals from the OBC to the payload using the FSP protocol. . 65
6.15 Flight model payload model integrated with FloripaSat-2 platform in the FlatSat. 66
6.16 Flight model payload model integrated with FloripaSat-2 platform (payload highlighted). 67

List of Tables

5.1 Project phase progress according to the specifications. 20
5.2 FSPv0.2 package fields. 39
5.3 FSPv0.2 adresses. 39
5.4 FSPv0.2 package types. 40
5.5 FSPv0.2 commands. 40
5.6 Summary of ESA guidelines. 42

Contents

Acronyms **i**

List of Figures **iii**

List of Tables **v**

1 Introduction **1**
1.1 Objectives 2
1.2 Work Outline 3

2 Methodology **4**
2.1 Experimental setup 4
2.2 Hardware development 5
2.3 Firmware development 5
2.4 SoC FPGA development 5
2.5 Project management 6

3 Background **7**
3.1 CubeSat Standard 7
3.2 Radiation Effects 8
3.3 SDRAM Memories 9
3.4 FPGA Devices 10
3.5 PCB Principles 11

4 Mission Analysis **13**
4.1 Environmental Parameters 13
4.2 Memory Experiment Analysis 16
4.3 Payload Specification 17

5 Development **19**
5.1 System Overview 19
    5.1.1 Prototype 20
    5.1.2 Engineering Model 21
    5.1.3 Flight Model 23
5.2 SoC FPGA Architecture 24
    5.2.1 Design Overview 25
5.3 Firmware Architecture 32
    5.3.1 Design Guidelines 32

5.3.2 Design Overview 34
5.3.3 Memory Fault Detection Algorithms 40
5.4 Hardware Architecture 42
5.4.1 Design Guidelines 42
5.4.2 Preliminary Design Analysis 46
5.4.3 Design Overview 49

6 Results 55
6.1 Simulations 55
6.2 Hardware Tests 57
6.3 Firmware Tests 58
6.4 Radiation Tests 60
6.5 Engineering Model Integration 63
6.6 Flight Model Integration 65

7 Conclusion 68

References 72

A Payload hardware schematics 73

1 Introduction

The increasingly market of small spacecrafts [1] is leading a rapid evolution of the technology and standards employed in these missions and pushing the limits of size, power consumption, reliability and capabilities of the embedded systems. The CubeSat, a satellite standard, and its variants are emerging as promising models for several applications in remote sensing, telecommunications, earth observation and even space exploration.

Comparing with the other standards and bulky satellites, which use high-end and tailored components and require considerable effort in development and launch, the CubeSats have an advantage due to the use of Commercial Of-The-Shelf (COTS) components and the reduced logistics involved that impacts directly in the mission cost.

Despite this advantage, the CubeSat missions do not reach all the space sector applications yet, due to the size scope or even a technological impracticability, reinforcing that the bulky missions still have relevance and importance in the subject. This scenario brings a new horizon for technological advances and encourage the widespread usage of small satellites to solve challenging problems and enhance the understanding of the space.

The environment faced in these space missions is harsh due to dramatic temperature variations, vacuum, radiation doses, and mechanical stress during the launch. These parameters achieve critical levels when considering long-term missions or higher altitude orbits. The majority of CubeSat missions are deployed in low altitude orbit profiles, where the conditions are less demanding, nevertheless still dangerous and onerous for development. Also, the utilization of COTS components without caution and previous analysis could lead to catastrophic failures as consequence of the creation of bottlenecks and unpredictability in the system. For this reason, several studies are being carried out to identify and quantify the vulnerabilities of these components in the space environment.

The most challenging condition to overcome is the significant amount of radiation in which these systems are exposed outside the protection of the earth magnetic field. Since the deployment, the satellite is exposed to a wide spectrum of random radiation interactions with diverse magnitudes and consequences [2]. In order to minimize these effects, different approaches are adopted in various layers, since the hardware architecture itself to complex software correction error techniques and redundancies. Then, evaluating the vulnerability of those components due to the radiation exposure and characterizing the critical failure thresholds are essential to determine the efforts that should be applied during the development. The Integrated Circuit (IC) memories, in a wide variety of technologies, system integration, manufacturing nodes, and operational characteristics, are one of the most critical components affected by the high doses of radiation [3].

The capacity of these devices, the importance of the information stored, and the strict requirements imposed make unfeasible, in several applications, the employment of intensive failure mitigation techniques due to the performance trade-offs or even the substantial amounts of data. For this reason, other approaches are investigated, which generally leads for high-cost custom hardened devices for specific niches. Therefore, it is essential to evaluate the real vulnerability of these components and determine which radiation mitigation strategy is required in each application.

In this context, two types of memory could be enumerated: non-volatile and volatile. The first group offers technologies with significant robustness within radiation exposure conditions without technological enhancements, such as: NOR Flash, MRAM, and FRAM. Also, due to less performance intensive requirements, some memory management techniques can significantly reduce errors and degradation. However, the pressure to fulfill the performance needs of demanding applications with the available volatile memory technologies introduce challenges that are under the analysis of many studies. The most performance capable and widespread technologies are the Synchronous Dynamic Random-Access Memory (SDRAM) chips, which are being improved and tailored for decades in the personal computers, workstations and servers industries.

1.1 Objectives

The efforts of this work focus on the development of a platform capable of performing radiation characterization of different SDRAM memories in space conditions. This payload will test these chips in the real harsh space environment by participating of the FloripaSat-2 mission. These particular SDRAM memories were previous characterized on laboratory experiments, then by exposing them to the real environments and executing the same tests routines will not only generate more results for analysis, but also provide an opportunity to assess the test methodologies themselves. Also, after collecting sufficient data to be analysed, this payload could be used to provide a meaningful health status, concerning the radiation doses which the satellite was exposed.

In order to accomplish these objectives, the payload is designed to follow the FloripaSat-2 OBDH DaughterBoard standard [4] [5], which defines the connectors, shape and size of the board. This standard allows the utilization of the module throughout future FloripaSat [6] core[1] missions in reason of its low space occupation inside the CubeSat, being considered in further uses as an expansion module instead of a payload experiment. Also, due to the mission limited power budget, the developed board consider reduced power consumption and define clever power management strategies. In addition, methods for monitoring latch-up, a type of short circuit which can occur inside an IC, are considered in the design.

Therefore, combining all these requirements, the payload architecture should consists of the following modules: a control and management subsystem, operated by a System-On-a-Chip (SoC) solution Field-Programmable Gate Array (FPGA) [7]; power converters for properly voltage level supply; latch-up monitors; communication and interface buses; debug module; and the SDRAM memory chips.

1.2 Work Outline

The work is structured in the following sections: background, providing a detailed explanation of the main topics approached in this work; mission analysis, describing the context in which the experiment is performed and defining the requirements for the development of this work; development, introducing all the systems developed and integrated to create the experiment platform and presenting the motivations, strategies, and guidelines for them; results, describing the executed tests and their results to validate the operation of the platform; and conclusion, providing a summary of the outcomes of the work.

---

[1] The FloripaSat-2 mission reuses the FloripaSat-1 core modules.

2 Methodology

Regarding the methodology applied for the development of this payload, a significant background knowledge came from both laboratory members and professors during years of experience working with dependable embedded systems, radiation tolerant devices, experimental tests, and extensive data analysis in the subject. The work was mainly developed as part of an internship in the Laboratory of Computer Science, Robotics and Microelectronics of Montpellier (LIRMM), with later support of the group for testing in radiation facilities. Then, the conclusion of the project and the integration with the FloripaSat-2 mission was performed in the Space Technology Research Laboratory (SpaceLab).

Some important architectural decisions and considerations were result of discussions based on previous publications and experiments. Moreover, during the development, different design guidelines were used to improve the payload architecture and overall reliability. Some of this standards are managed by respected institutions and incorporate a wide variety of systems, including high complexity missions and critical applications. Then, despite being based on these standards, the scope is scaled and simplified to attend the demands of a CubeSat project and the payload impacts on the success criteria of the satellite mission. The following topics describe the methodology used for the main aspects of this work.

2.1 Experimental setup

This payload uses previous works as a base for the experimental setup, particularly, the analysis of a wide variety of memories in harsh radiation conditions of the doctorate student Viyas Gupta [3] and the experiments performed at the RADEF (Jyväskylä, Finland) and at VESPER (CERN, Switzerland) facilities evaluating the same SDRAMs chips used in this project [8]. Then, this work uses this background knowledge intending to expand characterization of the memories behavior and fault models in harsh radiation conditions.

Following the same approach, the memory tests are implemented using standardized methods, which are subdivided in two categories: static and dynamic [9]. The static tests consist in evaluating the memory in absence of commands or single operation bursts and the dynamic covering the fault events in case of multiple operation bursts, in which each operation is a read, write, and cell refresh execution. In order to fulfill a significant part of the fault cases, different static and dynamic tests are executed continuously seeking for their condition specific deviations. Since the same algorithms were used in the laboratory experiments, similar results are expected, unless unpredictable rare events occurs, and a correlation could be traced between the ground and space test segments.

2.2 Hardware development

The hardware development methodology consisted of following key practices and guidelines to guarantee the proper board operation from the first time manufacturing it. For instance, the Printed Circuit Board (PCB) follows a simplified version of the European Space Agency (ESA) [10] guidelines, which were adapted for another CubeSat payload in a similar mission, regarding objectives and radiation concerns [11].

The design process consists of several repeated steps until the PCB project completion: requirement analysis, components selection, schematics preparation, layout design, design rules verification, and manufacturability assessment. Note that this process is iterative, in other words, many of these steps are repeated until a solid PCB design is achieved. For that, several independent reviews are performed intending to mitigate the risks before the actual board production.

2.3 Firmware development

Similar to the hardware strategy, the flight firmware is a iterative process requiring different assessments and tests. Also, previous experience and guidelines form the base of the code, where the FloripaSat-2 OBDH design and development strategy are a key source of inspiration. In section 5 this characteristics are extensively debated and the main ideas for increasing the payload reliability and code quality are detailed.

The project started with a simple firmware prototype developed for a development kit. Then, a firmware with a robust architecture was introduced after the addition of several functional and safety features and the execution of tests. This final version is able to flight alongside the FloripaSat-2 OBDH, following its requirements and standards.

2.4 SoC FPGA development

In order to reduce the development time, the SoC FPGA design is based on implementations and tools provided by the device manufacturer. This approach is possible since the experiment requirements require only standard interfaces with the memories chips. Then, the development consists of integrating different available blocks (e.g., memory controller, microprocessor, and peripherals), enabling safety features (e.g., watchdog, and SECDED), and testing these elements by simulations and in the target hardware platforms.

2.5 Project management

The project execution was divided in different phases, including requirements definition, documentation and design reviews. Generally, a standard used for space application is the systems engineering methodology, which defines the space mission from its initial specification until the operations finalization [12]. This standard require long-term planning and development periods, support multidisciplinary teams and produce high reliability systems. Then, for the scope of this work only key aspects and good practice patterns were followed. Also, regarding this project premises, agile standards are incorporated into the processes in order to increase the overall payload quality despite the rapidly development campaign.

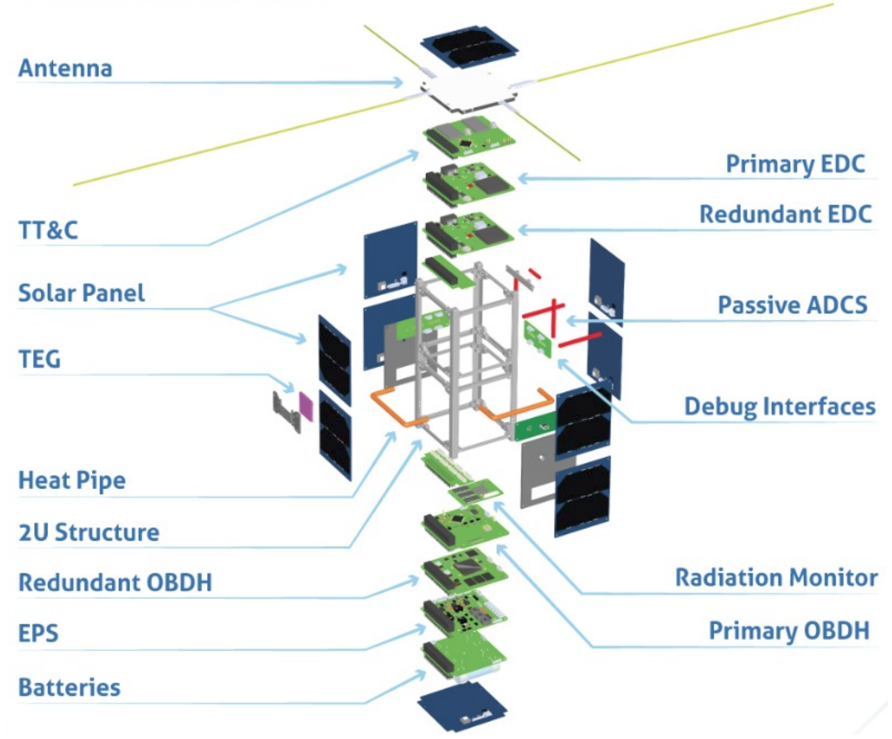
3 Background

3.1 CubeSat Standard

This work proposes a scientific experiment payload for CubeSats [13], a nanosatellite [1] standard. This standard of satellites consists of specific criteria for its shape, size, and weight, which help reduce costs due to the higher feasibility for mass-production and availability of off-the-shelf parts. Besides these benefits, the standardization reduces the development time and ease the access of transportation and deployment into space.

Essentially, CubeSats are satellites composed by small cube units, which might be combined to form larger spacecrafts and to attend the mission requirements and goals. This ”unit” is referred to as a 1U, then a 1U CubeSat is a 10 cm cube with a mass of approximately 1 to 1.33 kg. Figure 3.1 shows an illustrative diagram of the internal architecture of a 2U CubeSat, containing the core modules (power management, data handling and communication), payloads[2] (primary and secondary experiments), solar panels, antennas and others auxiliary modules.

Figure 3.1: Conceptual exploded view of a CubeSat (FloripaSat-2).



Source: SpaceLab, 2020.

---

[2] The ”Radiation Monitor” illustrated in the diagram is the payload proposed in this work. It also presents its preliminary location inside the satellite.

3.2 Radiation Effects

The space is full of engineering challenges related to the harsh environment conditions, including: mechanical stress, drastic temperature variations and severe radiation doses. This work focus on the last aspect, analysing its influence in electronic devices and evaluating the temporary and cumulative damaging phenomena. These effects are characterized by the Single Event Effects (SEEs), caused by transient interactions, and the Total Ionizing Dose (TID) and Displacement Damage (DD), caused by accumulative interactions. The sensitive circuit nodes are affected or permanently damaged with the penetration of ionizing particles, which are representative outside the Earth's magnetic field protection [14] [15] [16].

The interaction with ionizing particles can induce the occurrence of instantaneous damaging events, which are characterized by the single event effects in various categories depending on the errors produced and the device components affected: Single-Bit Upset (SBU), soft-error; Multiple-Bit Upset (MBU), soft-error; Multiple-Cell Upset (MCU), soft-error; Single Event Transient (SET), soft-error; Single Event Functional Interruption (SEFI), soft-error; Single Event Latch-up (SEL), hard/soft-error; Single Event Gate Rupture (SEGR), hard-error; and Single Event Burnout (SEB), hard-error [14]. These effects might lead to soft-errors, errors that are reversible since can be mitigated by power cycles, resets, or other erase operations. In other cases, the effects induce hard-errors, leading to permanent device failure.

Both SBUs and MBUs are transient events that causes bit-flips (upsets) within a memory cell or latch, differing in the quantity of elements affected. Similarly, the MCUs are caused by the generation of two or more bit-flips within a cell array. The SETs are characterized by the collection of free carriers generated that induce a current or voltage transient that may lead to an error. The SEFIs are caused due to a perturbation in control signals, causing a loss of functionality and entering in a undefined state, a test mode, or a halt. The SELs occurs when a parasitic thyristor is triggered in the device structure, which generates high-current consumption and may lead to a thermal overheat in the affected region. The SEGRs occurs when a particle hit breaks the gate dielectric of a MOS transistor and the SEBs when a strike causes a destructive burnout due to high-currents.

The cumulative effects depends on the continuous exposure to radiation conditions that degrades the function of the device until its operation is definitely compromised [15]. The TID is a long-term failure mechanism that causes shift in the transistors threshold voltage, increase of leakage currents (power consumption), and also the degradation of the gain in bipolar devices. The DD effect is the result of collisions between high-energy protons and the device that, when a minimum required energy is achieved, creates deformities in the lattice by removing the hit atom which results in different local electrical characteristics. The accumulation of this effect with energy above the displacement threshold can create a large fault cluster. For instance, it causes the reduction in bipolar transistor gain, reduction of the efficiency in solar cells, light emitting diodes and photodetectors, charge transfer inefficiency in charge coupled devices and resolution degradation in solid-state detectors.

3.3 SDRAM Memories

Synchronous dynamic random-access memories (SDRAM) are dynamic random-access memories (DRAM) whose operation is coordinated by an external supplied clock signal. The DRAM devices are a type of random access semiconductor memory that stores each bit of data in a cell structure based on a capacitor and a transistor. The capacitor retains or releases the energy during charge and discharge events controlled through the transistor. These two states, charged and discharged, represent the two values of a bit. However, due to parasitic coupling loads, the capacitor slowly leaks off, which leads to data loss after a certain period. Then, to prevent this, the device requires an external controller to perform periodic refresh cycles that rewrites the data in the capacitors, restoring them to their original initial charge.

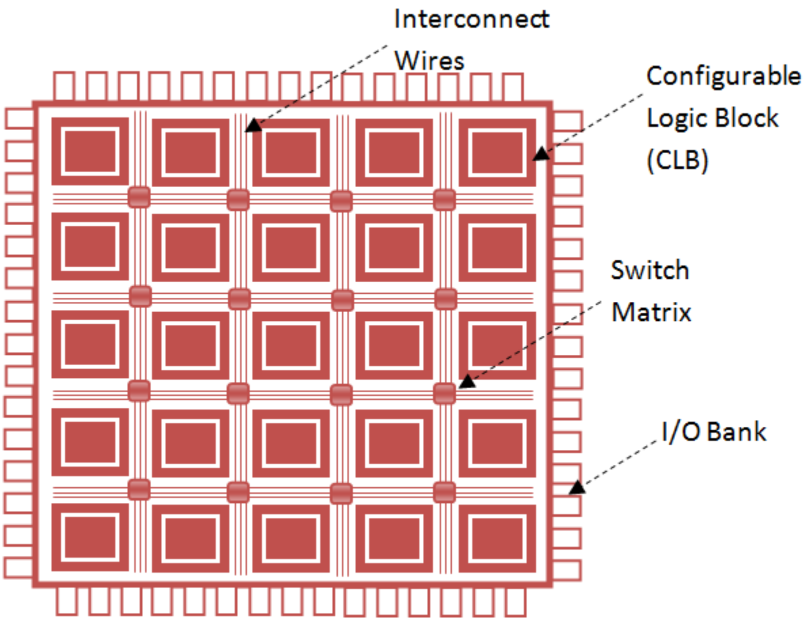
The DRAM memories have the disadvantage of direct influence of input signals in the internal functions since each combination of control signals determine the execution of one memory operation (e.g, read, write, and refresh). Using a synchronous interface, it is possible to use pipeline techniques that improve the memory performance in comparison with asynchronous interfaces at the same frequency of operation. Pipelining means that the device can handle new commands before the termination a previous operation processing. Then, write and read commands can be followed by other commands after a fixed number of clock cycles (latency). Also, to improve even more the access operation speed, the memory is divided into equally sized sections called banks, which allows independent operations occurring simultaneously. This architecture enables SDRAMs to achieve greater concurrency and higher data throughput than the asynchronous memories.

These devices are susceptible to harsh radiation environments and many studies are carried out to evaluate their behavior in different conditions using ground-based experiments [17] [18] [19].

3.4 FPGA Devices

A Field-Programmable Gate Array (FPGA) is an integrated circuit designed to be reconfigurable after manufacturing, differing from its counterparts that are purpose specific and unchangeable. Then, it can be configured by the developer to perform almost any digital operations in various levels of complexity using a Hardware Description Languages (HDL) and automation tools to translate the high-level commands into the physical structure combinations. The device contains an array of programmable logic blocks and a hierarchy of reconfigurable interconnects that enables the blocks to be combined for creating complex structures or merely simple logic gates (e.g., AND, OR, XOR). Also, in the majority of the devices, these blocks include memory elements that might be from simple flip-flops to more complete blocks of memory. Figure 3.2 illustrates a simplified internal architecture of these devices.

Figure 3.2: Simplified FPGA internal architecture.



Source: allaboutfpga, 2021.

The FPGAs core structure can be combined with other technologies and structures to form even more flexible and ready to use devices. This mixed chips are generally referred to as System-On-a-Chip (SoC) devices

[7], which internal architecture combine several elements to attend the different elements required by an application in an unique and compact solution. For instance, in this work is used a SoC solution that includes a FPGA array, an ARM-core processor, volatile and non-volatile memories, communication interfaces, clock sources and conditioners, independent memory controllers, and a bus interface. In subsection 5.2 the device and its subsystems are described with more depth.

Besides the architectural aspect of FPGA devices, the technology employed is important to determine its characteristics under radiation conditions. Despite the chosen chip not being a rad-hard device, it presents robust attributes in comparison with regular systems due to its Flash based implementation [3] and radiation characterization reports provided by the manufacturer [20] [21].

3.5 PCB Principles

The Printed Circuit Boards (PCB) are a combination of interspersed layers of conductive and non-conductive materials to form electrical circuits. These layers and electrical circuits are connected using structures called ”vias” and ”tracks”, respectively. The electronic components are attached, electrically and mechanically, in the external layers through a soldering process in their corresponding ”pads” (i.e., the designated electrical interface location). Figure 3.3 presents a simplified conceptual diagram of a multi-layer PCB.

Using electronic Computer-Aided Design (CAD) tools, the PCBs are deigned for production in specialized companies that use automated machinery from the board manufacturing and assembly to optical and electrical testing. Due to this production chain, the PCBs design can incorporate high precision layouts, which allowed their evolution for complex circuits and structures. However, this progress exploiting the limit of the physical implementation transformed the development of these boards a challenging task. Some unusual and counter intuitive effects start to change the expected characteristics of the circuits, which might lead to defective designs. For instance, when working with high-speed [3] signals, their integrity might be compromised by mismatching in length, impedance, reflections, crosstalking, and return paths.

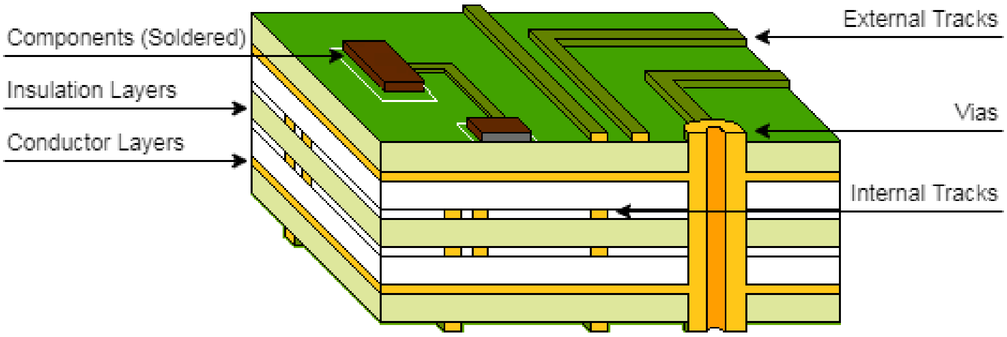
In order to avoid these phenomena, the PCB designers use mitigation techniques and follow validated standards. In this work, the developed

---

[3] Signals with short periods of time during edge-to-edge changes, which generally appears in higher frequencies.

board required some measures to guarantee the proper operation and meet the requirements, since it is designed for space applications [11]. These techniques and patters are described in further sections and the proper justifications provided.

Figure 3.3: Conceptual PCB board.



Source: PCBWay, 2020.

4 Mission Analysis

Before defining the payload specification and the subsystems architecture, it is important to estimate and preview the environmental conditions in which the system will be exposed. Also, since the payload objective is to evaluate the SDRAM memories, it is important to use previous laboratory experiments to predict and estimate the faults associated with radiation exposure, besides the required information to define timing and operation parameters during the in-orbit experiments. The following sections present the effects and concerns for the critical environment parameters in which the payload is exposed and provide an analysis of memory experiments and results.

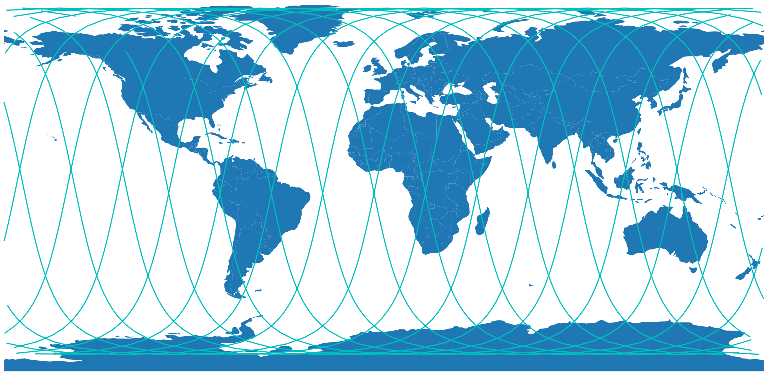
4.1 Environmental Parameters

In order to predict the environmental conditions, the orbit parameters are required and since the payload is designed for CubeSats, generally in Low-Earth Orbits (LEO), the parameters should be analysed in a mean orbit in this range. Since the payload experiment is part of the FloripaSat-2 mission, the estimations presented in the work are performed for this mission parameters: heliosynchronous orbit; 98^o inclination; 500 km apogee and perigee; approximately 100 minutes orbit period; approximately 0 eccentricity. Since the orbit is heliosynchronous and circular, the satellite follows a circumference in a pole to pole trajectory, which traces almost a perpendicular line with the equator plane. Then, due to the Earth rotation around its own axis, the satellite describes an orbit that periodically covers the entire sphere area relative to the Earth surface. Figure 4.1 presents a preview of this orbit using the parameters herein described and demonstrate the relative surface coverage in orbital passages.

Radiation Analysis

In this orbit (LEO), the satellite is protected against the major part of the high energy particles and radiation, due to the Earth magnetic field shielding. However, there are two problematic regions, in terms of high exposure: the South Atlantic Anomaly (SAA) and in high latitudes (around the magnetic axis) [15]. Figure 4.2 shows schematically the magnetic belts, known as Van Allen radiation belts, and the region where the anomaly is located. The high energy particles are entirely deflected from the Earth trajectory or experience severe deviation hitting the Earth poles, as con-

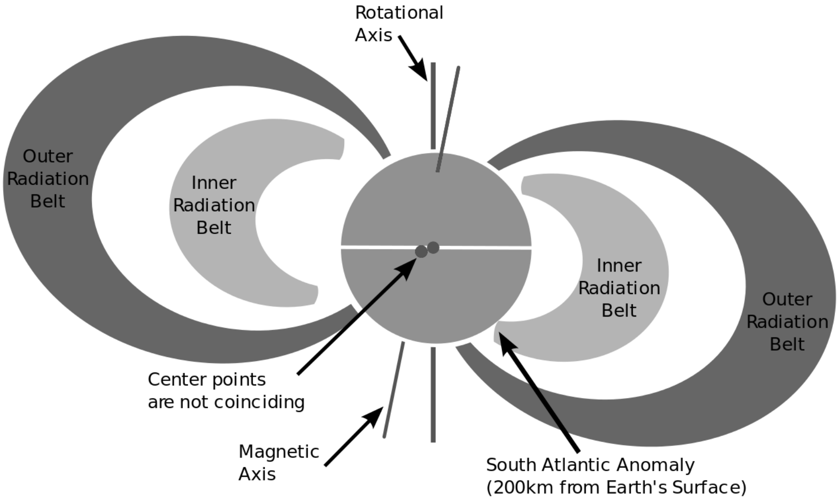
Figure 4.1: A preview of the satellite simulated groundtrack.



Source: SpaceLab, 2021

sequence of the magnetic interactions. The lower energy, although still ionizing and harmful, particles achieve the lower orbits and might cause damages in the most sensible electronic devices. Moreover, the SAA region, where the inner radiation belt is closest to the Earth surface, increase the amount of exposure of higher energy particles in a considerable area, expanding the harmful effects in the satellites subsystems.

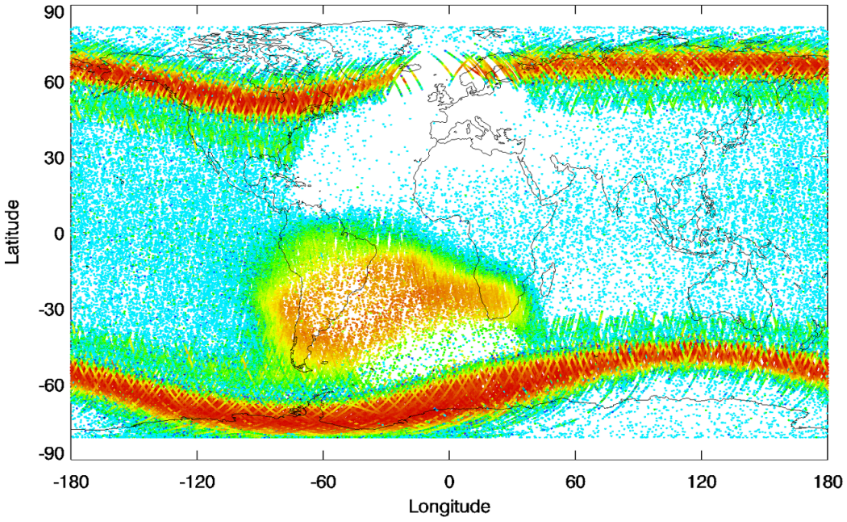
Figure 4.2: Schematic overview of the Van Allen radiation belts.



Source: Wikipedia, 2021

In order to characterize these belts and evaluate the ionizing particle penetration, the PROBA-V mission [22], using advanced equipment, performed measurements of these particles in different energy intervals. The measures include: Electrons, from 0.5MeV to 20MeV; Protons, from 9.5MeV to 248MeV, and after the 248MeV; and Helium particles, from 38MeV to 980MeV and further. Also, a penetration intensity relative to the Earth surface position was provided. The results suggest critical radiation levels for LEO missions in these particular regions and provide a reference value for estimations in this work. Figure 4.3 is a sample of the measurements, for low energy electrons in logarithmic scale, performed by the PROBA-V mission and shows qualitatively the regions herein described, intending to only provide a simplified visualization.

Figure 4.3: Simplified preview of the radiation vulnerable regions.



Source: V. Pierrard et al., 2016

Thermal and Mechanical Analysis

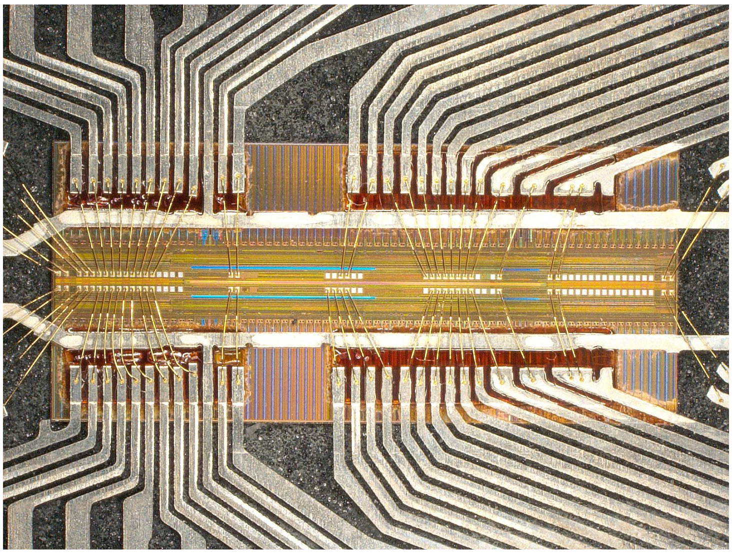
A comprehensive thermal analysis for FloripaSat-1 mission was carried out by the development members, which characterize the environmental temperatures for the satellite during its orbit [23]. In summary, the results lead to the estimated range of -25 to 60 degrees Celsius as maximum variation in several orbital conditions. These values are predicted in the surface of the solar panels, then it is expected less variation inside the satellite. For the payload, the component must be operational in these ranges, which COTS commercial or industrial grades are capable to meet the requirements, avoiding the use of specific components ratings, such as space or even military.

Moreover, concerning mechanical stresses, especially during the launch and deployment, the payload must maintain its physical integrity for the entire mission operation. Then, the design, assembly and integration adopt the FloripaSat-2 guidelines that are based in the lessons learned from the predecessor mission (FloripaSat-1). These procedures and design considerations are summarized throughout the document, in the design robustness considerations or in qualification tests planning. Therefore, using the FloripaSat platform methodology, the developed payload should work properly, since mechanically the board and attachment scheme is similar to the other satellite modules. Also, to ensure this statement, the payload will pass through the entire FloripaSat-2 integration and qualification process performed by their experienced members.

4.2 Memory Experiment Analysis

Regarding the memory experiments, these three SDRAM chips were characterized in test facilities by the laboratory group [8] and one of the memories by other groups at the California Institute of Technology in different studies [17] and [18]. The SDRAM memory chips used are produced by the company Integrated Silicon Solution Inc. (ISSI) and use different manufacture nodes: 110 nm for IS42S16320B [24], 72 nm for IS42S16320D [25], and 63 nm for IS42S16320F [26]. The memories are Single Data Rate (SDR) SDRAMs, with 536,870,912 bits organized in four banks with 8192 rows and 1024 columns of 16 bits. The maximum operating frequency is 143 MHz, with a 3.3 V input supply voltage, and packaged in 54-pin TSOP-II packages. Figure 4.4 shows the memory internal structure.

Figure 4.4: Inside view of the SDRAM memory chip (IS42S16320F).



Source: D. Söderström et al., RADECS, 2019

4.3 Payload Specification

The payload specification is subdivided in four categories: payload, general payload requirements; environment, reliability requirements related to environmental parameters; performance, capability and features requirements; and operation, behavioral requirements.

Payload

REQ-PAY-000 The payload shall be compliant with the CubeSat standard and general restrictions.
REQ-PAY-010 The payload must be compatible with the FloriapaSat-II mission.
REQ-PAY-020 The system shall be compatible with the FloriapaSat-II OBDH DaughterBoard standard.
REQ-PAY-030 The system must have a communication interface with the OBDH module through the DaughterBoard connector.
REQ-PAY-040 The payload must include three different SDR SDRAM memories.
REQ-PAY-050 The payload must convert specific voltage levels for its subsystem from the provided OBDH power supply.
REQ-PAY-060 The OBDH module must be able to shutdown the payload without turning down the power supply channel.
REQ-PAY-070 The system must adopt strategies to reduce power consumption.

Environment

REQ-ENV-000 The payload components shall be selected in order to reduce the harsh space environment effects.
REQ-ENV-010 The payload shall be designed to minimize the radiation influence over the components other than the memories.
REQ-ENV-020 The payload must be resilient to the temperature variations and vacuum faced on the mission orbit.
REQ-ENV-030 The board mechanical attachment shall be resilient to the vibrations and stresses during the launch and deployment.
REQ-ENV-040 The PCB solder mask and components soldering must be in compliance with space applications.

Performance

REQ-PER-000 The memory controller must be able to operate in the memories frequency ranges.
REQ-PER-010 The memory refresh rates should attend the specifications.
REQ-PER-020 The system controller must operate in frequencies similar to the memories.
REQ-PER-030 The communication interfaces shall attend the OBDH performance requirements.

Operation

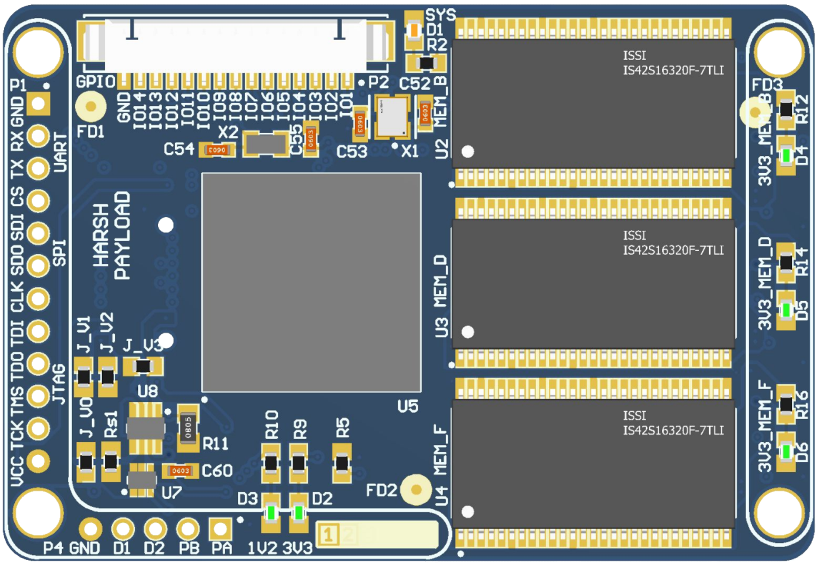
REQ-OPE-000 The system must have two operation modes: experiment and health monitor.
REQ-OPE-010 When in experiment mode, the system must perform memory test operations and send results continuously.
REQ-OPE-020 When in monitor mode, the system must execute essential memory tests, send status and reduce the power consumption.
REQ-OPE-030 The OBDH module must be able to manage the operation modes.
REQ-OPE-040 The OBDH module must be able to request and schedule commands through a communication interface.
REQ-OPE-050 The OBDH module must be able to receive data and logs through a communication interface.
REQ-OPE-060 The payload must have a memory controller module to manage the SDRAM memories.
REQ-OPE-070 The system shall perform different tests in the SDRAM memories.
REQ-OPE-080 The system shall synchronize execution time with the OBDH.
REQ-OPE-090 The system must count the elapsed time to provide reference for the radiation events data.
REQ-OPE-100 The payload must use a watchdog timer to prevent execution faults.

These requirements were important to guide the architecture definition and the elaboration of tests. Considering the scope of this work, the payload has a lean list of requirements, which provide the minimal conditions to execute the project. In section 6, the main requirements are cited for the tests that exercise their validation.

5 Development

The following topics will cover the entire payload development process in details, focusing on the architecture, methods, and technologies, besides the actual development steps and results. Some organizational and work activities aspects are shortly mentioned or placed as annexes, unless essential for the comprehension of the measures and approaches adopted throughout the development, since the focus of this work is the technological challenges and analysis, and the architectural proposals. Figure 5.1 shows the payload board (Flight Model) developed during this work. Also, since the source code, source FPGA project, scripts, and hardware files are very extensive to include as annexes, the entire project is available in a licensed open-source format in a repository at GitHub [27][4].

Figure 5.1: Preview of the developed payload.



Source: Author

5.1 System Overview

In order to accomplish the payload requirements, the development process reached different stages of completion: prototype phase, engineering model and flight model. These categories are a simplification of the system engineering phases definition, which generally define a deliverable and a associated review. For this reason, an overview of each stage of the system

---

[4] All parts of the project were developed by the author with guidance of supervisors and colleagues, except by the preliminary radiation experiment in the engineering model, which was performed by the LIRMM colleagues.

is provided to ease the understanding of the project evolution and an iterative architectural conception. Table 5.1 summarizes the delivered features in each phase, the development period, and overall standards compliance.

Table 5.1: Project phase progress according to the specifications.

Features and standards	*Prototype (1 month)*	*Engineering Model (5 months)*	*Flight Model (6 months)*
System control and management	x	x	x
Reliable radiation controller	x	x	x
Radiation hardened board		x	x
Single SDRAM memory access	x	x	x
Multiple SDRAM memory access		x	x
Static memory test algorithms		x	x
Dynamic memory test algorithms			x
Frequency test algorithms			
Refresh rate test algorithms			
UART communication interface	x	x	x
SPI communication interface		x	x
I2C communication interface			
CAN communication interface			
Watchdog timer protection		x	x
Low power mode operation		x	x
FSP protocol implemented		x	x
Debug logging routines	x	x	x
Experiment routines		x	x
Communication routines		x	x
System housekeeping routines		x	x
Reliable FreeRTOS instance		x	x
Firmware reliability review		x	x
Hardware reliability review			x

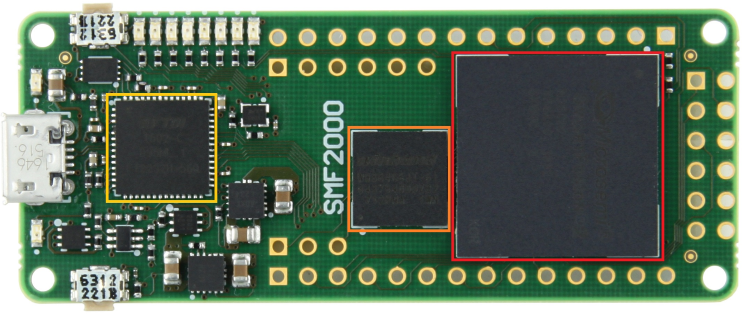
Source: Author.

5.1.1 Prototype

During the prototyping phase, the base architecture and components of the design were selected to fulfill the payload requirements. The SDRAM memories controller was the first aspect to be defined due to its importance and objective inside the payload. This definition lead for a FPGA oriented architecture, which could provide the necessary design flexibility and port connections, high frequencies support, and parallel execution during memory operations. Then, among different technologies and vendor options, the SmartFusion2 family from Microsemi emerged as a good trade-off between performance, power consumption, and reliability. Also, these devices have an embedded ARM Cortex-M3 processor, several useful on-chip peripherals and the advantage of being flash based, which provides a better radiation resilience [20] [21] [28].

Then, in order to test and validate the first design concepts, the next stage was selecting a suitable development board. Among the different options provided for this device family, the SMF2000 development kit (from Trenz Electronic) were the best solution for the prototype conception. This board standout due to its low price, simple architecture, and all the necessary features, including a SDR SDRAM memory for the first design assessments. Figure 5.2 presents the board and its components highlighted: in yellow, a serial to USB converter used to format the FPGA bitstream, debug, and send log messages; in orange, the 64Mb (or 8MB) SDR SDRAM; and in red, the M2S010 SoC FPGA chip.

Figure 5.2: Development board used during the prototype phase.



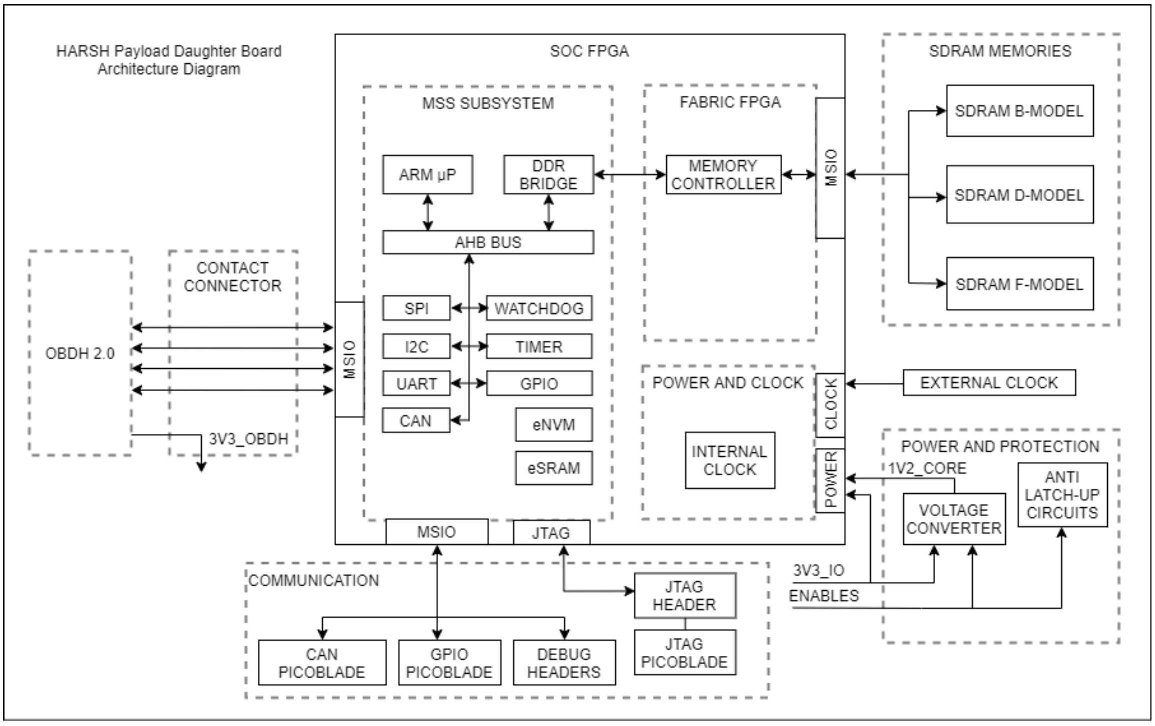
Source: Trenz Electronic, 2021

Regarding the delivered features of the prototype, shown in Table 5.1, the base SoC design solution and the minimal system firmware were developed to attend the most critical aspects of payload, which includes the memories control and tests, simple communication routines, and usage of the internal non-volatile memory to store the firmware sources. Since the FPGA is flash based, a non-volatile memory, the bitstream is retained indefinitely and any formatting routine is necessary after system resets.

5.1.2 Engineering Model

After the prototyping phase, the engineering model started to be developed considering the previous phase review outputs. Since the first approach was using a development kit, the major design revisions focus on creating the hardware platform, defining the FPGA SoC architecture and developing the embedded software. Then, the first iteration of hardware development was based on the SMF2000 and to reduce further changes[5], this version intended to be as close as possible of the flight model. Interfaces, connectors and the memories were the major changes from the development kit, besides the FPGA assignments and the board shape itself. Figure 5.3 presents a simplified architecture diagram of the engineering model, which includes the hardware and SoC FPGA modules.

Figure 5.3: Simplified engineering model architecture.



Source: Author

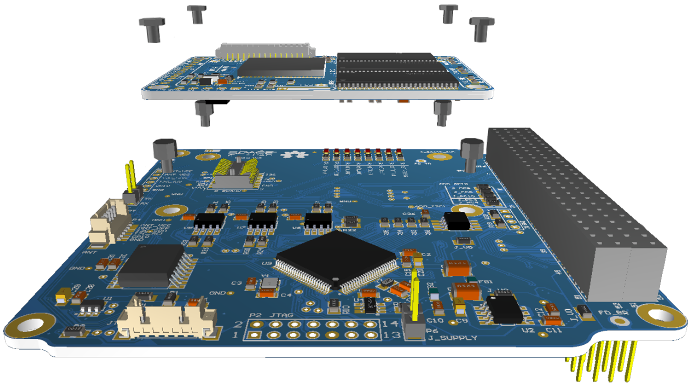
In order to visualize the integration with the FloripaSat-2 OBDH, some conceptual assemblies were performed during the development. Figure 5.4 presents one of these integration setups and provide a visual understanding of both modules, their mechanical connection and electrical interface through the contact connector.

In summary, the engineering model review revealed that the hardware design and the SoC FPGA solution are adequate to the payload mission and requirements as a flight version. The firmware, showing the basic operation and experiments, presented satisfactory overall stability, but it was decided to refactor some routines, perform more specific tests, and improve the reliability and redundancy of some modules.

---

[5] Despite the engineering model providing a good opportunity for a board focused on ground experiments, with more resources and isolation for the experiment memories section, the platform was developed to minimize changes for the flight model due to budget restrictions.

Figure 5.4: Conceptual payload model integrated with the FloripaSat-2 OBDH.



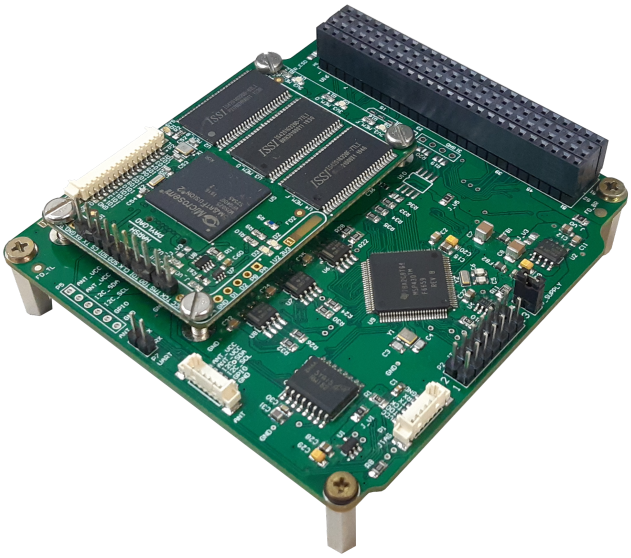
Source: Author

5.1.3 Flight Model

Regarding the flight qualified model, the major changes from the engineering version were the system firmware and the experiment algorithms. These modifications were important to improve the experiment execution and provide better results. Also, the integration tests revealed some weak points and non-reliable management routine implementations that might led to critical failures. Then, in order to solve these issues, the firmware was updated, tested and the design reviewed.

Figure 5.5 presents the payload attached to the FloripaSat-2 OBDH during the integration. More details about the integration setup and results are described in further topics. Also, the final design architecture is presented in details in the following sections.

Figure 5.5: Payload model integrated with FloripaSat-2 OBDH.

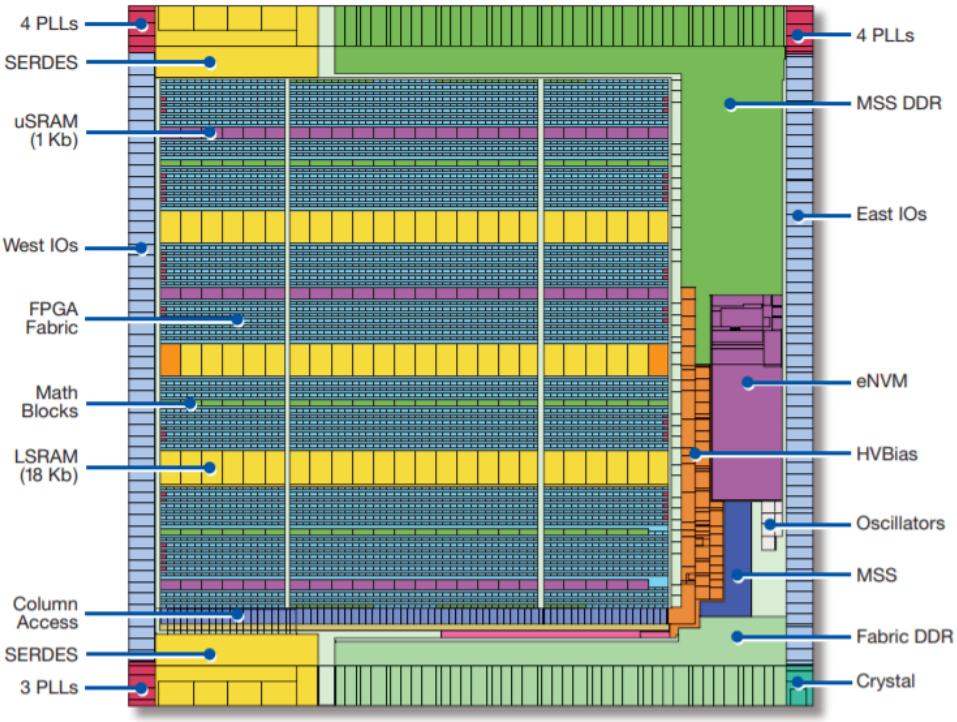


Source: Author

5.2 SoC FPGA Architecture

The payload core subsystem is the SoC FPGA, which provides the system control and management due to the embedded microprocessor, memory controller, communication interfaces and main peripherals. The device comprehends the main functionalities of the payload system, then using a radiation resilient device[6] improves the overall dependability and permits the analysis of the memory faults within the absence of controller errors [20] [21] [28]. Also, the SmartFusion2 family is suitable for critical applications due to its intrinsic robustness, high density die, high-performance, and availability of a wide variety of validated peripherals. These devices have a heritage of utilization in space applications and demonstrates satisfactory overall performance and reliability [29]. Figure 5.6 provide a internal physical architecture overview of the SmartFusion2 devices. The following sections describe the SoC system implemented inside the M2S010, part of the SmartFusion2 family.

Figure 5.6: FPGA die overview.



Source: Microsemi, 2020.

---

[6] The device is not radiation tolerant, but offers some reliability features and the manufacturer provides some valuable radiation performance reports for SEE (mainly SEL, SEU, and SEFI [21]).

5.2.1 Design Overview

The SoC architecture embedded in the chip has several features and allows each subsystem to be enabled or disabled in the design, in order to reduce the power consumption. The device is subdivided in different functional sectors, where the main part for this project are the Microcontroller Subsystem (MSS), with a fixed implementation, and the FPGA Fabric, where the user modules and configurations are implemented. Both parts are employed to create a flexible system attending the required capabilities and define the core design: controller, using the ARM Cortex-M3 processor; Soft Memory Controller(SMC) for the SDRAM management, in the FPGA Fabric section; serial communication (UART_0, SPI_0, I2C_0); Watchdog timer, for system protection; timers, for timing and synchronization; GPIO ports; Embedded Non-Volatile Memory (eNVM_0) for code sources storage; Embedded Volatile Memory (eSRAM_0) for code execution and temporary data storage; and debug, using the System Controller for JTAG interface. The following topics details each subsystem characteristics and operation. Also, Figure 5.7 presents a simplified diagram of the built-in internal available modules.

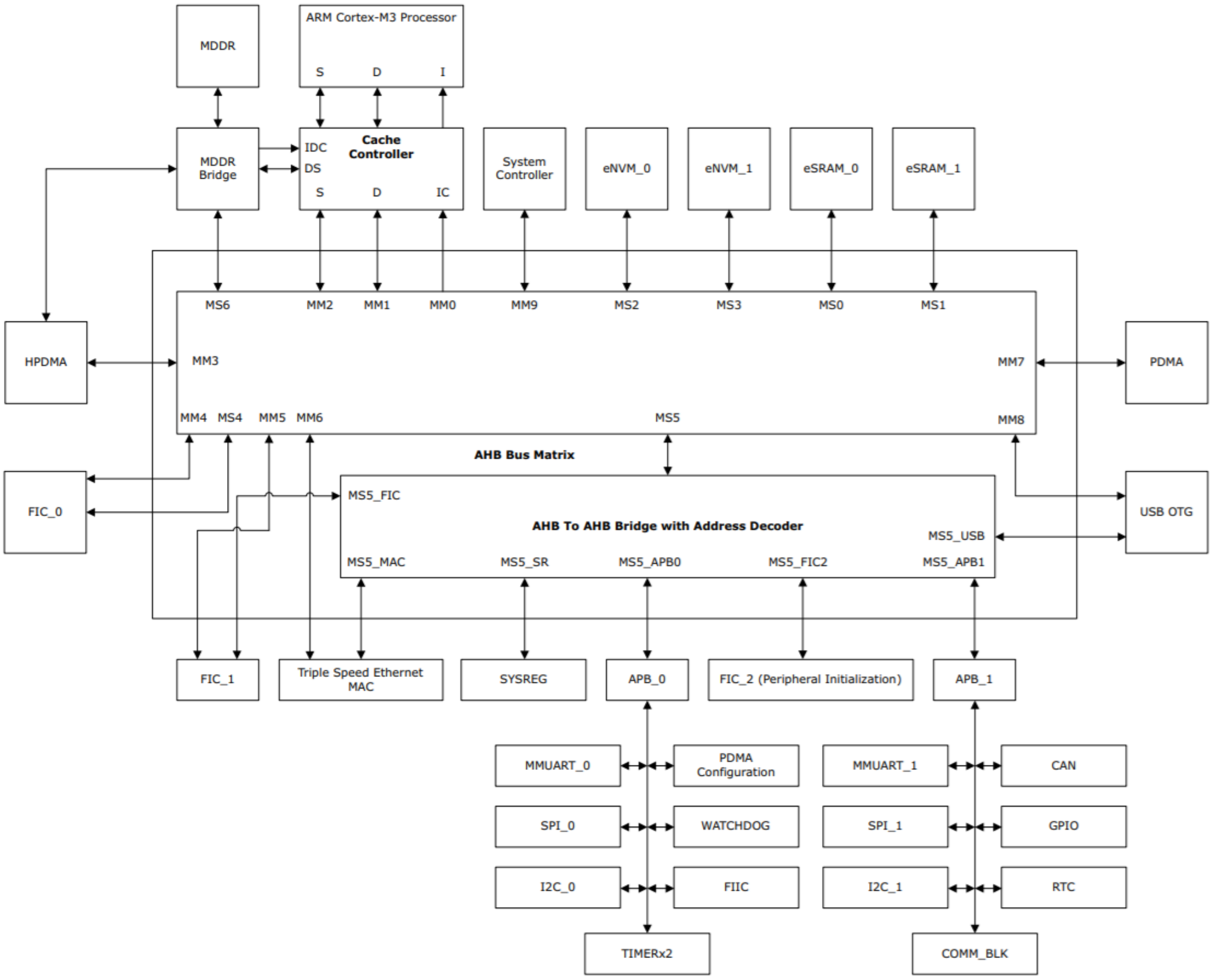
Microprocessor

In the SoC architecture, the processing unit is a hard core instance of an ARM Cortex-M3 microprocessor. The unit has a low-power consumption design and intend to accomplish deeply embedded applications requirements. This processor consists in a 32-bit RISC architecture with 3-stage pipeline that provide enough processing performance for the payload routines and algorithms. In order to provide the utility features for this device, the SoC design includes a set of peripherals, which combine communication, timers, system memories, watchdog and debug modules. When disabled, these subsystems are hold in low power mode and provide customization during the development that contribute for the feasibility in power constrained applications. Also, the system uses the ARM Advanced Microcontroller Bus Architecture (AMBA) as interface for the peripherals and the FPGA fabric devices, employing the Advanced High-performance Bus (AHB) protocol in the communications.

Memory Controller

In order to manage the SDRAM memories, a controller instantiated in

Figure 5.7: Internal architecture diagram overview.

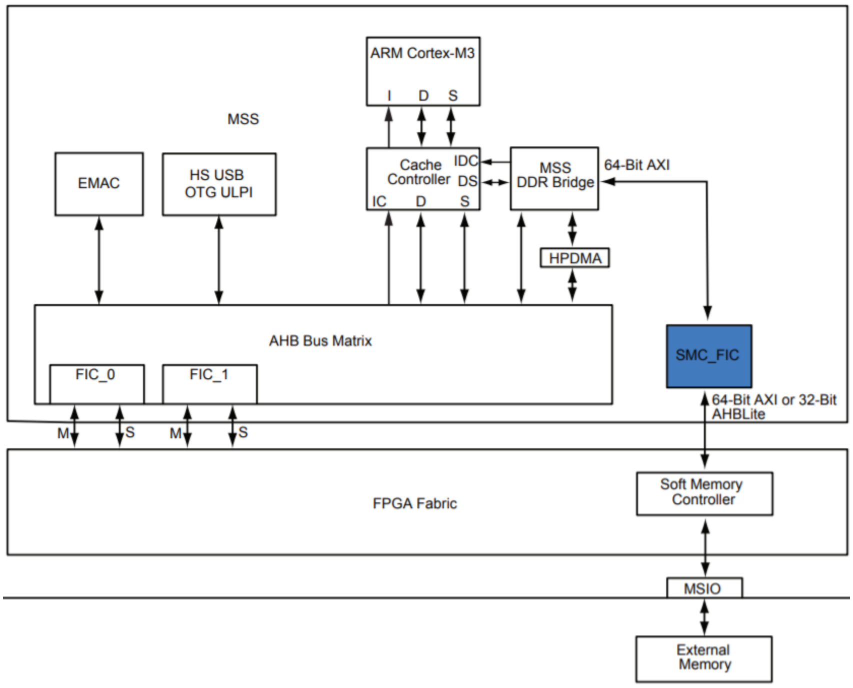


Source: Microsemi, 2020.

the FPGA fabric is used for the external interface with these devices. In this architecture, the controller is shared among the memories to simplify the hardware design (only one shared interface instead of three times the number of signals) and provide similar operational characteristics to isolate the memories from other effects. A strategy was adopted to optimize the experiment execution that consists of a round Robin cyclic algorithm. The method implementation is described in further sections, but the main objective is to keep experiments in the memories running simultaneously with this shared controller scheme. Also, due to similar internal logical architecture, despite the manufacturing differences, the memories are compatible in operation and constraints. Figure 5.8 describes the interfaces used for the controller from the processor until the external interface. The processor, a master in the AHB bus, requests data operations for the DDR Bridge that redirect these commands for the memory controller, using a

AHB to AXI interface. Then, the controller, besides the regular management operations, handles the communication and control signals through the MSIO ports to the actual memory. Since the SDRAM chips are connected in parallel in the same MSIO ports, the controller uses chip select signals to enable the correct device.

Figure 5.8: Memory controller interfaces overview.

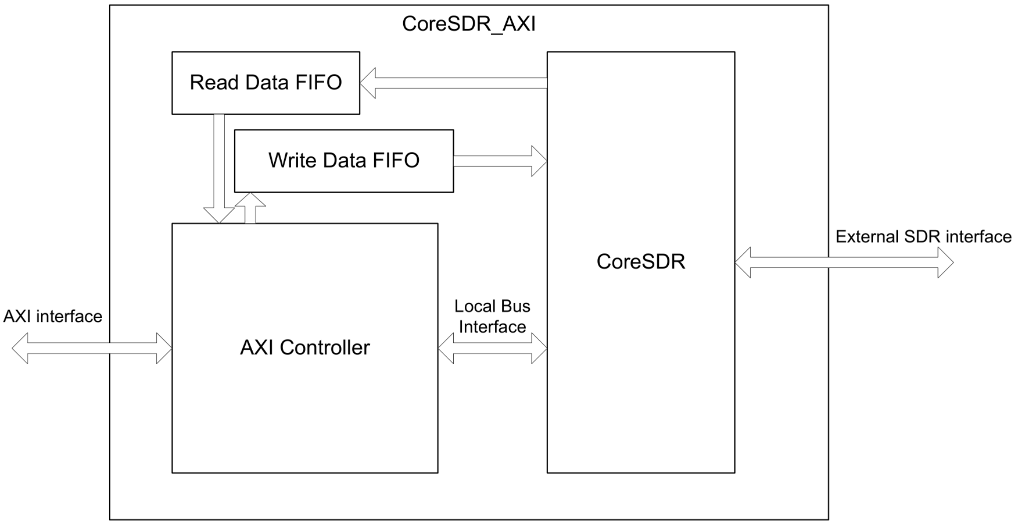


Source: Microsemi, 2020.

This controller is part of the catalog collection provided by the SoC FPGA vendor, which provides several ported IP Cores for the fabric implementation. The used core is designed for SDR SDRAM memories and adapted to be connected to an Advanced extensible Interface (AXI). Figure 5.9 shows a simplified diagram of this IP core. In summary, this module receive operation requests through the AXI bus and the internal controller handles this demands, generating the formatted operation frame for the actual memory controller and handling its responses for the requester. However, since the MSS subsystem (the requester) architecture provide an AHB interface, a converter is required to properly handle the protocol exchange for an AXI bus. Also, this module is provided in the vendor catalog and do not require external control, besides the reset signal.

The CoreSDR is the controller that manage the memory communication, generating the control signal within the time constraints. It receives the operation requests, data length and addresses from the AXI controller and the data from the write buffer in case of a write operation. The input

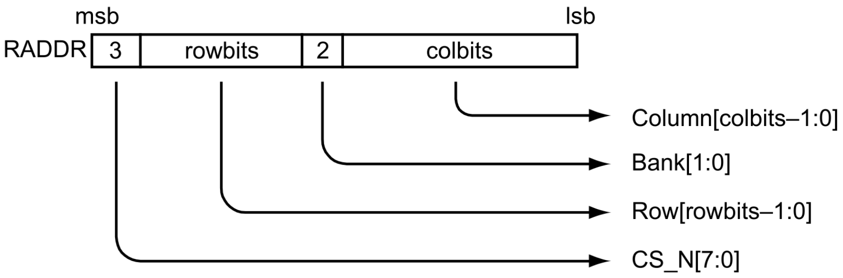
Figure 5.9: Core SDR wrapped for an AXI interface architecture.



Source: Microsemi, 2020.

address is structure as described in Figure 5.10. This scheme is important to determine which memory should be accessed in relation to the input address, where the three most significant bits are decoded in up to eight chip selects and the others bits to set the column, row and bank. When an encoded chip select bit changes, it means that the address space of that memory ends and the next starts. This scheme allows the usage of up to eight memories without the need of additional controllers or external interfaces.

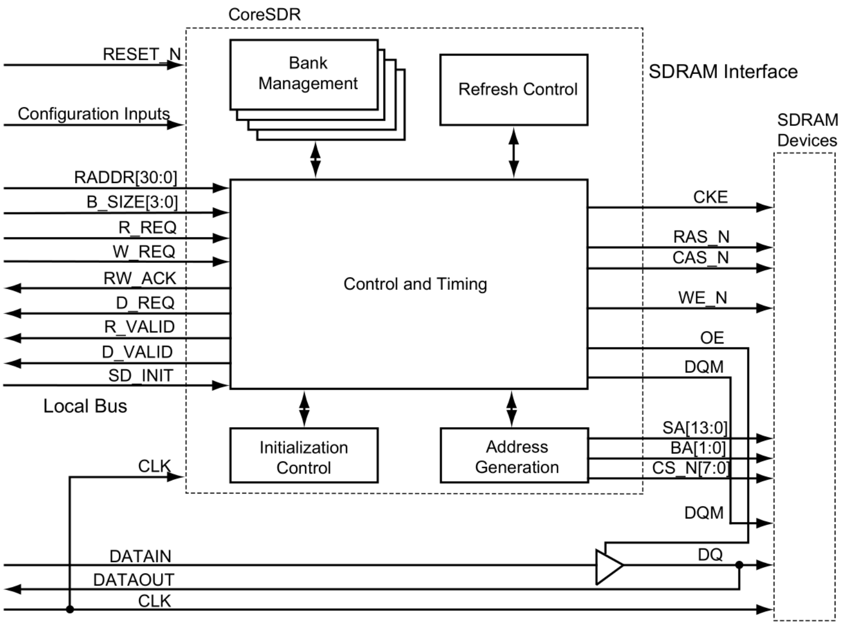
Figure 5.10: Core SDR Address scheme overview.



Source: Microsemi, 2020.

Figure 5.11 shows the internal modules of the controller and the corresponding signals. The output signals are directly connected to the SDRAM memories: RAS, CAS and WE determine the requested commands; SA and BA set the target addresses; DQM is a mask for bytes within a word; CKE is used to enable the clock in the memory; CS activate the selected memory; DQ is used for input and output data; and OE is used to enable the tri-state in DQ signals. The refresh control unit is used to send periodic refresh commands, due to the SDRAM technology requirements, and the address generation module convert the input addresses to the memory access format. The rest of the subsystems handles other functionalities: initialization, resets and timing constraints. Regarding the last item, the three memories have compatible time requirements and latency characteristics at the payload operation frequency, which were manually configured to attend the datasheet specifications.

Figure 5.11: Core SDR architecture diagram overview.



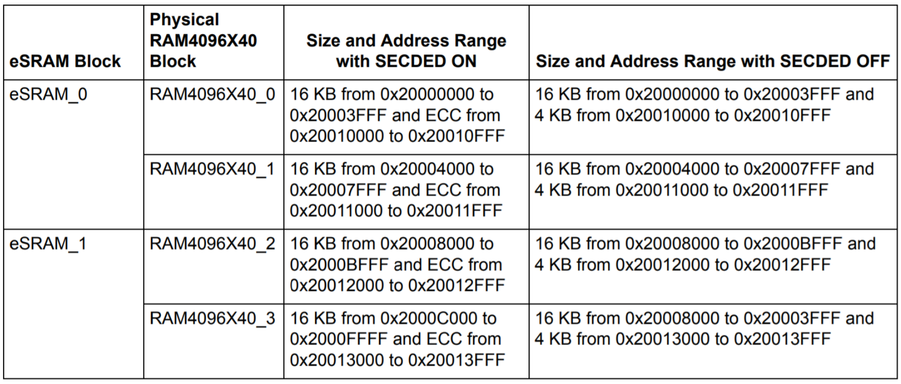
Source: Microsemi, 2020.

Embedded Memories

In order to store the code and data, the SoC solution includes two different types of memories: embedded Non-Volatile Memory (eNVM), using the flash technology; and the embedded Static RAM (eSRAM). For the payload design, the eNVM is dedicated for code storage, constants and user data if correctly allocated, which provides to the system all required instructions during execution and reboots. Also, since the eNVM is a flash memory, the radiation effects are reduced due to its intrinsic resilience. The eSRAM is responsible for the runtime environment, providing a faster memory for variables, stack, heap, and other system memory abstractions. Since the experiment memories (SDRAM) could generate many bytes of errors, in case of block failures or high radiation exposures, there is a limit for the experiment data. This approach guarantee that the system tasks will be properly executed without memory issues due to capacity, since this data is stored in a bounded manner. In this scenario, the worst case is achieving the eSRAM memory capacity and overwriting the older results.

The eNVM has a capacity of 256kBytes that provides a secure size for all the system applications without limitation issues. The eSRAM has two modules with 32kBytes each when SECDED-ON and 40kBytes each when SECDED-OFF. The Single Error Correction, Double Error Detection (SECDED), a extension of the Hamming code, is a technique to improve the data storage dependability, in this case on the eSRAM that is very susceptible to the radiation effects. Since this algorithm is a built-in feature of the SoC FPGA and there is a reasonable storage capacity margin, the eSRAM is used with the SECDED enabled. Figure 5.12 describe the eSRAM scheme in both cases. Then, using this approach, it is possible to avoid potential problems with the system memories, allowing the payload to properly evaluate the radiation effects on the target devices, the SDRAMs.

Figure 5.12: eSRAM memory scheme.



eSRAM Block	Physical RAM4096X40 Block	Size and Address Range with SECDED ON	Size and Address Range with SECDED OFF
eSRAM_0	RAM4096X40_0	16 KB from 0x20000000 to 0x20003FFF and ECC from 0x20010000 to 0x20010FFF	16 KB from 0x20000000 to 0x20003FFF and 4 KB from 0x20010000 to 0x20010FFF
	RAM4096X40_1	16 KB from 0x20004000 to 0x20007FFF and ECC from 0x20011000 to 0x20011FFF	16 KB from 0x20004000 to 0x20007FFF and 4 KB from 0x20011000 to 0x20011FFF
eSRAM_1	RAM4096X40_2	16 KB from 0x20008000 to 0x2000BFFF and ECC from 0x20012000 to 0x20012FFF	16 KB from 0x20008000 to 0x2000BFFF and 4 KB from 0x20012000 to 0x20012FFF
	RAM4096X40_3	16 KB from 0x2000C000 to 0x2000FFFF and ECC from 0x20013000 to 0x20013FFF	16 KB from 0x20008000 to 0x20003FFF and 4 KB from 0x20013000 to 0x20013FFF

Source: Microsemi, 2020.

Serial Interfaces and IO Ports

The Microsemi FPGA SoC solution provides several peripherals in the MSS subsystem. For this payload, it is used different serial communication protocols (SPI, I2C, CAN and UART) and GPIOs to provide flexibility and redundancy. The SPI protocol, as slave, is used exclusively for communication with the FloripaSat-2 OBDH. In this interface is used the FloripaSat Protocol (FSP), in order to attend the OBDH requirements and increase the communication reliability. Also, as redundant channels, the I2C (as slave) and CAN could be used for the same interface. This design allows compatibility with further missions of the FloripaSat core, since the payload could be used as a health monitor for radiation damage. Moreover, a UART interface is provided for system logging, which ease the development and debug processes, and a JTAG for the FPGA programming. Also, 20 GPIOs are available for parallel protocols, hardware tests or debugging, 6 IOs for the latch-up monitors and two pins for the FPGA live probes, used for system monitoring during debug.

These interfaces are available in different connectors, models and locations depending on priority and utilization: SPI, I2C and 4 GPIOs on the contact connector (main interface with the OBDH); CAN, JTAG and 14 GPIOs on dedicated connectors; UART, JTAG, SPI, live probes and 2 GPIOs on dedicated connectors only available in the engineering model for testing purposes due to space constraints; and the 6 IOs, directly connected to the latch-up connectors.

Watchdog Timer, Interruptions and Timers

The internal watchdog timer provide a failure recovery feature for the payload system, without the necessity of an external device. This mechanism protects the payload against software errors by resetting the device and restoring the system execution from boot. Moreover, the payload uses the available timers to synchronize the peripherals execution and trigger system actions, alongside the interruption handler that manages these events.

5.3 Firmware Architecture

The CubeSat standard, although restrictive concerning definitions of external mechanical and electrical interfaces, integration guidelines and materials utilization, it does not restrain the implementation of the internal subsystems. Regarding the firmware architecture a considerable freedom allows the developers to propose their own standards using approaches and patterns from different application niches. For instance, companies tend to introduce a proprietary architecture in their platforms, creating a tailored environment of modules and development tools [30] [31] [32]. Moreover, since the CubeSat missions have a substantial contribution from the academic environment, a wide range of techniques and models are employed intending to attend the application requirements and explore novel strategies [33] [6].

Despite these differences, the projects share principles and patterns inherent of critical applications, and more specifically, space niche that imply focusing on the same aspects: reliability, harsh environment mitigation techniques, redundancy, and predictability. Then, regarding the software development, using these concerns to design failure-aware systems generally lead to sophisticated and robust architectures, which increase the mission success probability. Considering the scope of this project, the following sections detail the aspects that guided the payload firmware development, describe the behavioral operation of each module and include the subsystems design.

5.3.1 Design Guidelines

The payload software follows guidelines presented in [34], where the flight software is proposed following good design patters and practices, modularity, and reliability. Despite that, the framework considers a long-term reusability and presents an approach for the satellite OBC, with many more tasks and responsibilities. Then, these principles are applied for this work in a simplified context and key aspects are detailed in the following topics.

Failure Protection Strategies

The payload firmware is designed to decrease failures due to system malfunctions and environmental conditions. Some of these strategies are periodic system resets, watchdog timer, hardware check routines, default parameter values and error correction algorithms in the system memories and communication interfaces. For instance, the embedded SRAM memories, designated to store the experiment results, use the Single Error Correction, Double Error Detection (SECDED), a extension of the Hamming code. Also, in the communication with the FloripaSat-2 OBDH, a Cyclic Redundancy Check (CRC) algorithm, an error-detecting code used to detect accidental changes to raw data, is applied inside the communication interface protocol.

Moreover, the payload firmware in based on an operating system targeted to embedded systems, which improves the overall reliability since the platform has inheritance of several projects in different applications, including the space sector. Then, using this framework to develop the payload firmware, the system failures are mitigated and the architecture more qualified to the application.

Internal and External Synchronization

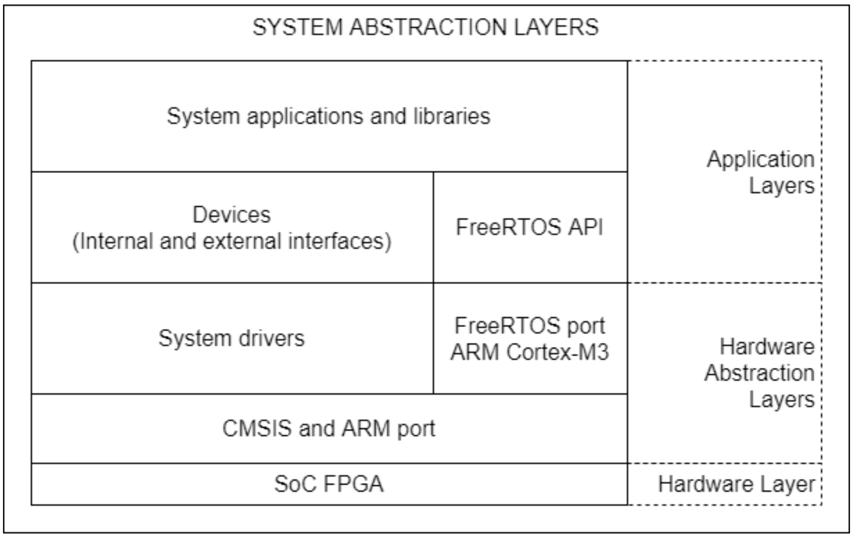
The system synchronization within internal constraints and with external devices is a challenge since the payload is intended to operate controlled by the OBDH and, meanwhile, perform several functions within certain time and priority limitations. Then, using the operating system framework, this requirement is accomplished using queues, tasks, semaphores, and interruption handlers, which provides tools to accommodate the different constraints in a deterministic fashion. In summary, each task has an execution rate, priority and optional parameters (semaphores, initial delays and group events) that are predictably handled by the system scheduler. Besides this execution scheme, data should be transferred between these task without the necessity to be immediately handled, which is the fundamental strategy of the data synchronization. Also, since the system operation is controlled by the OBDH (slave mode), it is necessary an interruption handler for communications, which is non-deterministic and could cause issues. Then, to avoid this problem, a binary semaphore is used to create a link between the interruption occurrence and a handler task, which transforms the operation deterministic in terms of execution alongside the rest of the system tasks.

Moreover, the communication with the OBDH has synchronization strategies due to the FloripaSat Protocol (FSP), which defines a communication standard between both modules. This protocol uses handshake operations, acknowledge responses and set a command list, besides error-detecting routines to ensure correct communication. These strategies and features are detailed in depth throughout the following sections, reveling the actual architecture, operation mechanisms, and consequences.

Abstraction Layers

The firmware structure is based on abstraction layers, which creates a separation between layers and allow a development oriented to the application itself instead of the hardware details in that feature. Also, this strategy increase the firmware readability, flexibility, organization, and maintenance, due to the implementation independence from distant layers and abstraction of unnecessary information. Figure 5.13 presents this layers and their hierarchy. At the bottom, it is represented the lowest level, the SoC implementation, then the first abstractions, called hardware abstraction layers due to handling with specific operation related with the processor and peripherals. At the upper levels, there are the application layers that use a high-level description approach for the implementation of the payload system functionalities and operations.

Figure 5.13: System abstraction layers diagram.



Source: Author.

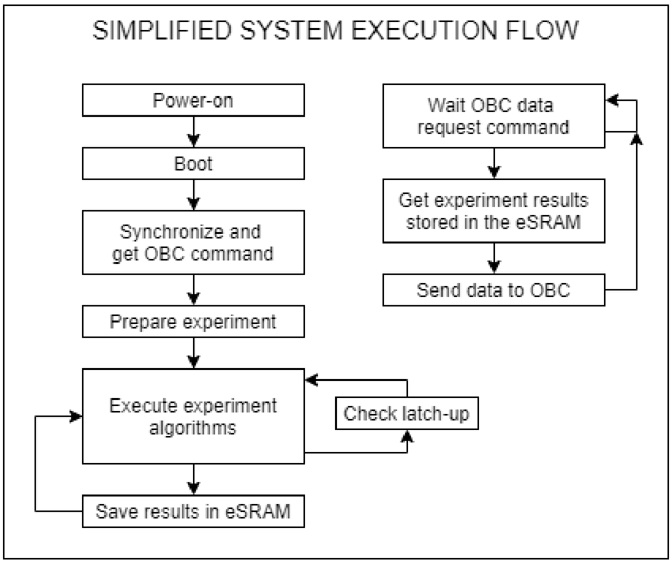
5.3.2 Design Overview

The firmware architecture, as herein described, use an embedded operating system that is widely used in commercial products. The FreeRTOS is a Real-Time Operating System (RTOS) for microcontrollers and small microprocessors with proven robustness, tiny footprint, and wide device support. Then, in order to use a similar architecture as the FloripaSat-2

OBDH, share libraries, and ease the development process, the FreeRTOS was the most feasible platform for this project, besides an official port for the ARM Cortex-M3 microprocessor inside the SoC FPGA. This firmware development approach lead to the conception of functionalities translated into operating system tasks and synchronization schemes in queues and semaphores. In summary, the firmware is designed using the operating system features and structure.

Regarding the system functionality, Figure 5.14 describes a simplified execution diagram, presenting the main experiment functions and their sequence. The housekeeping and system functions are omitted to evidence the major payload purpose, the experiment execution itself. After power-on and internal setup, the payload receives a command from the OBDH, prepares the experiment parameters and enables the experiment execution. Then, this data is stored in the embedded SRAM for later retrieve when a OBDH data request occurs. This loop summarizes the experiment application and the payload purpose, evaluate the SDRAM memories in the harsh environment.

Figure 5.14: Simplified system execution flow diagram.

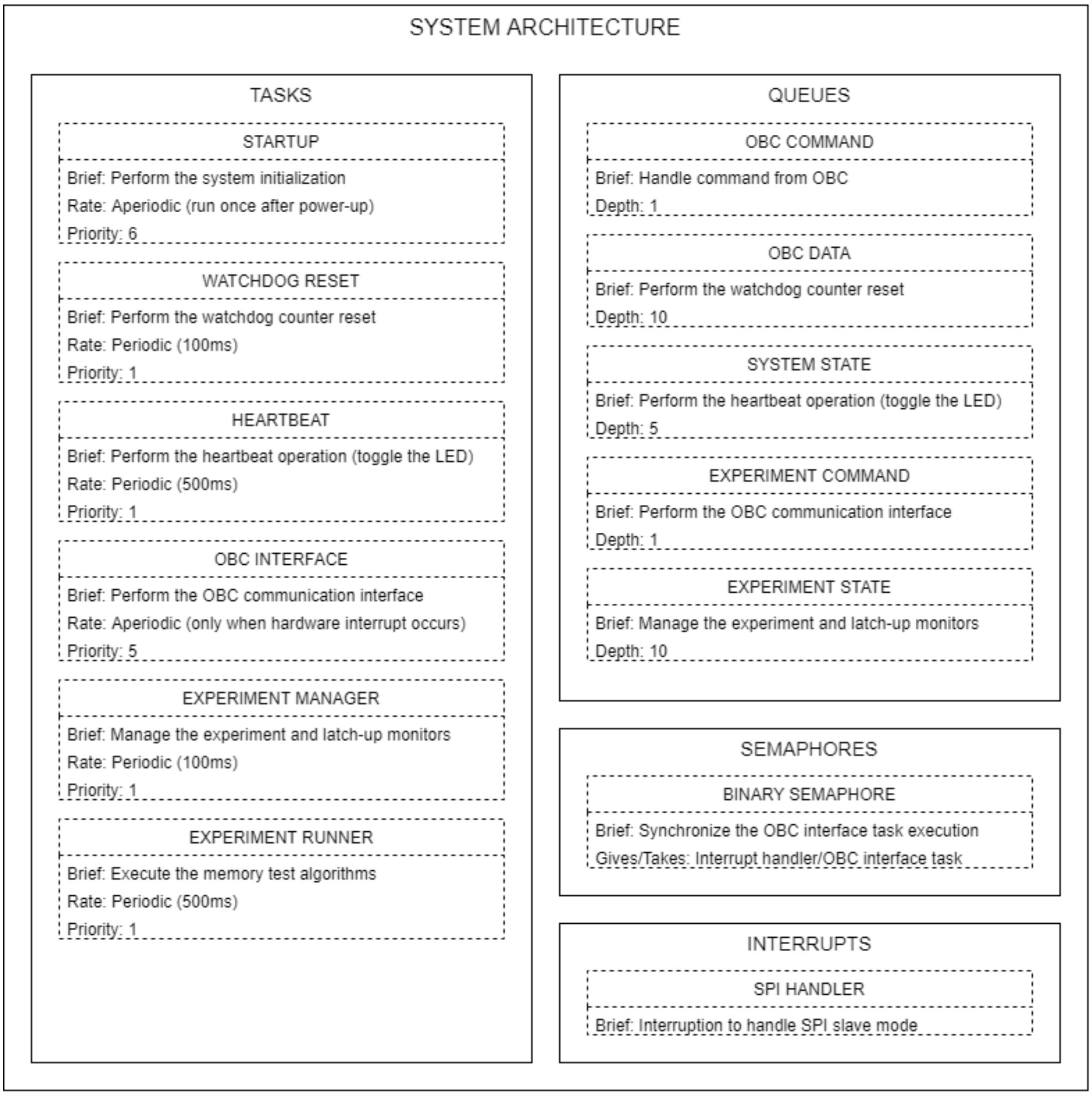

SIMPLIFIED SYSTEM EXECUTION FLOW
Power-on
Boot
Synchronize and get OBC command
Prepare experiment
Execute experiment algorithms
Check latch-up
Save results in eSRAM
Wait OBC data request command
Get experiment results stored in the eSRAM
Send data to OBC

Source: Author.

These functionalities are translated in tasks, queues, semaphores, and interrupt handlers, which are represented in Figure 5.15. Also, some parameters are presented (priority, rate, periodicity and depth) that are detailed and justified in the following topics.

Experiment Routines

Figure 5.15: System architecture diagram.



SYSTEM ARCHITECTURE

TASKS

STARTUP
Brief: Perform the system initialization
Rate: Aperiodic (run once after power-up)
Priority: 6

WATCHDOG RESET
Brief: Perform the watchdog counter reset
Rate: Periodic (100ms)
Priority: 1

HEARTBEAT
Brief: Perform the heartbeat operation (toggle the LED)
Rate: Periodic (500ms)
Priority: 1

OBC INTERFACE
Brief: Perform the OBC communication interface
Rate: Aperiodic (only when hardware interrupt occurs)
Priority: 5

EXPERIMENT MANAGER
Brief: Manage the experiment and latch-up monitors
Rate: Periodic (100ms)
Priority: 1

EXPERIMENT RUNNER
Brief: Execute the memory test algorithms
Rate: Periodic (500ms)
Priority: 1

QUEUES

OBC COMMAND
Brief: Handle command from OBC
Depth: 1

OBC DATA
Brief: Perform the watchdog counter reset
Depth: 10

SYSTEM STATE
Brief: Perform the heartbeat operation (toggle the LED)
Depth: 5

EXPERIMENT COMMAND
Brief: Perform the OBC communication interface
Depth: 1

EXPERIMENT STATE
Brief: Manage the experiment and latch-up monitors
Depth: 10

SEMAPHORES

BINARY SEMAPHORE
Brief: Synchronize the OBC interface task execution
Gives/Takes: Interrupt handler/OBC interface task

INTERRUPTS

SPI HANDLER
Brief: Interruption to handle SPI slave mode

Source: Author.

The experiment is executed through two dependent and periodic tasks: ”experiment manager task” and ”experiment runner task”. The first is responsible to handle the commands received through the communication, constantly check the latch-up monitors, and coordinate the system queues. The second task coordinate the experiment execution and save the generated experiment data results. Concerning the execution flow, the manager task has a higher priority than the runner one, thereby executing despite the experiment cycle accomplished the end. This design would lead to issues if a synchronization mechanism did not coordinate the execution of one task accordingly to the other. Then, in order to fulfill this need, a queue system was designed to trigger some states that are executed in key moments, for instance: reading the memory to restore the experiment results, changing the execution configuration parameters, and many other functionalities.

There are two queues directly used in the experiment routine and three indirectly, which transfer data, configuration parameters, and status. The two direct queues are used for receiving the configuration from the manager to the runner task, and sending the position and size of the experiment data stored in the dedicated memory sector to the manager task. The other queues are used by the manager and communication tasks for inputting the commands from the OBC, and outputting the data and status information to the OBC.

Figure 5.16 summarizes the experiment execution flow, presenting from the communication with On-Board Computer (OBC) to the execution of the memory test algorithms. It starts with the command received from OBC, process the execution parameters, run the test algorithms, store data in the dedicated memory, send data status for further retrieving, read this data, and finishes sending it to the OBC. This sequence is a simplified overview and support the visualization of the routine, but it should be analysed as two tasks not necessarily contiguous in time and using queues to trigger sequential events.

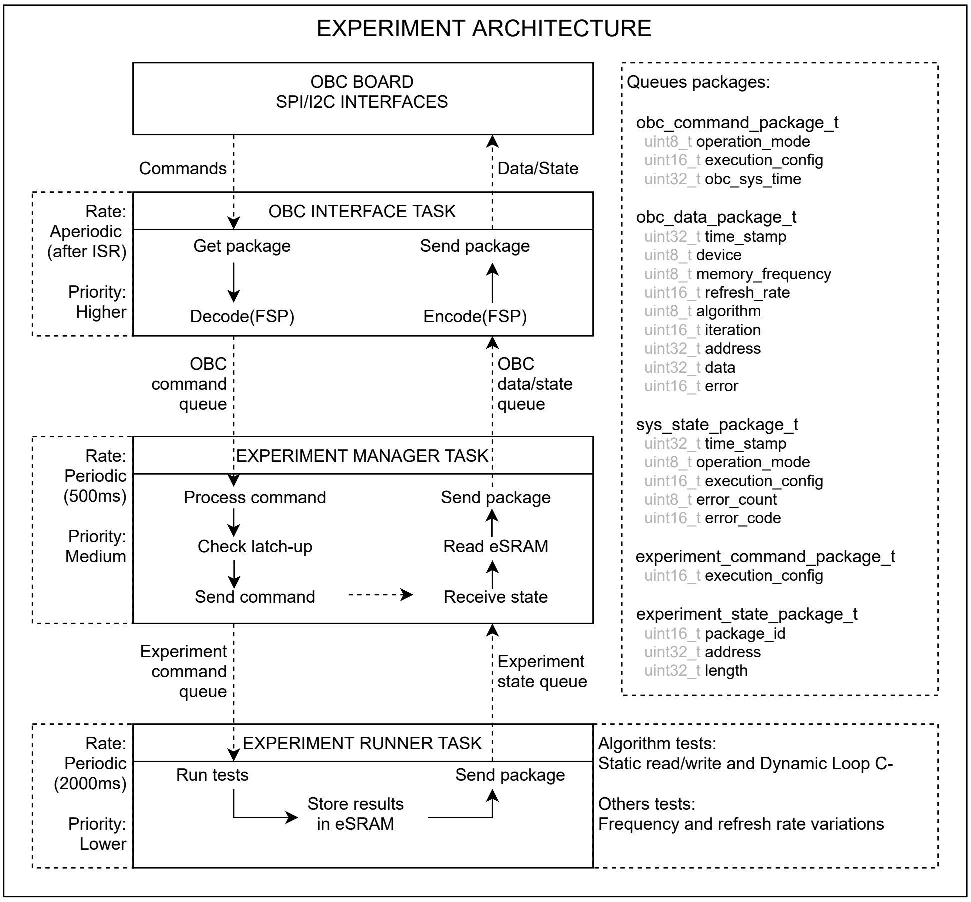
Housekeeping Routines

The housekeeping routines are responsible to initialize and maintain the system properly configured and operational. There is an aperiodic task referred to as "startup" that performs the system boot once after power up, in other words, the initialization of all peripherals, interfaces, and devices. Also, there is a dedicated periodic task that ensures constant notification of properly operation for the watchdog timer and another one with the lowest priority for other periodic non essential functions (e.g., blink LED).

Communication Routines

The communication routines are handled using an aperiodic task and interruptions (one for each physical protocol). The task execution trigger is associated with the interruption since the payload is a slave in the communication from the perspective of the OBC. Then, to ensure that the communication is correctly attended, the task has the highest priority. This strategy is adopted to accomplish both requirements, attend communication requests with low latency and use the payload firmware formal

Figure 5.16: Experiment architecture diagram.



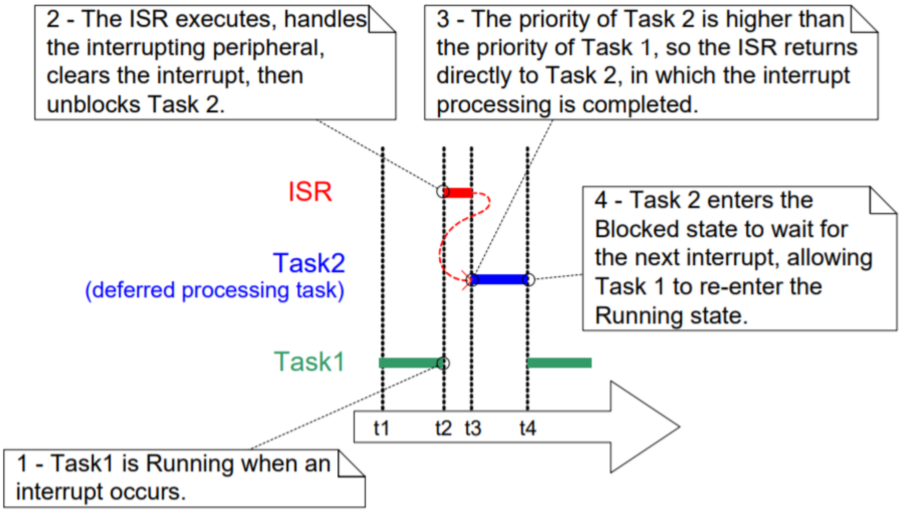
Source: Author.

application structure (i.e., the operating system tasks). Also, this method prevents conflicts between the scheduler and manually configured interruptions, and allows the use of the operating system features safely. Figure 5.17 presents the scheme adopted to receive and process the communication interactions. Also, Figure 5.16 describes the interface of the communication task with the experiments routines using queues for synchronization.

Also, the communication uses the FloripaSat Protocol (FSP), developed with the purpose of standardizing the communication between the FloripaSat service modules. This protocol does not specify the physical layer, so it can be used, for instance, over SPI, I2C, or UART. For the payload, the SPI interface is used as the physical layer and a FSP address was provided.

The package fields defined by the FSP are shown in Table 5.2. The **n** value is defined by the field ”Length” and can assume values between 0 and 248. Therefore, the package total size can reach from 7 to 255 bytes. In order to grant the correctness of the transmitted data, a Cyclic Redundancy Check is made. The standard used is CRC-16-CCITT (initial value = 0

Figure 5.17: Communication interrupt and task architecture diagram.


2 - The ISR executes, handles the interrupting peripheral, clears the interrupt, then unblocks Task 2.
3 - The priority of Task 2 is higher than the priority of Task 1, so the ISR returns directly to Task 2, in which the interrupt processing is completed.
ISR
Task2
(deferred processing task)
4 - Task 2 enters the Blocked state to wait for the next interrupt, allowing Task 1 to re-enter the Running state.
Task1
t1
t2
t3
t4
1 - Task1 is Running when an interrupt occurs.

Source: FreeRTOS, 2020

and polynomial = 0x1021). All the FSP's package fields, except the SOD, are used to generate the CRC value.

Field	*Description*	*Length (Bytes)*
SOD	Start of data (always 0x7E)	1
SRC	Source address	1
DST	Destination address	1
Length	Length of the package payload	1
Type	Type of package	1
Payload	Data of the package	**n**
CRC	CRC-16-CCITT bytes	2

Table 5.2: FSPv0.2 package fields.

The FloripaSat team defined module addresses for each board in the system, including the payload referred to as ”HARSH”. Table 5.3 presents this list.

Module	*Address*
EPS	1
TTC	2
OBDH	3
HARSH	4

Table 5.3: FSPv0.2 adresses.

Also, the protocol defines package types, as expressed in Table 5.4, and allowed commands, defined in Table 5.5

Package Type	*Value*
Data without ACK	1
Data with ACK	2
Command without ACK	3
Command with ACK	4
ACK	5
NACK	6

Table 5.4: FSPv0.2 package types.

Command	*Value*
NOP	1
Send Data	2
Request RF Mutex	3
Shutdown	4

Table 5.5: FSPv0.2 commands.

Debug and Logging Routines

In order to perform debug sessions and execute integration tests, there is a logging system to provide useful feedback during the development. The routines use a serial interface (UART) to output relevant steps and notify critical failures, which are easily read through a serial monitor (e.g., PuTTY or equivalent). These logs are inserted in key positions, using status flags to select between appropriate messages.

5.3.3 Memory Fault Detection Algorithms

Memory fault detection algorithms are used to evaluate the three memories under the harsh radiation conditions. The implemented algorithms use static, based on writes and reads executed once per address, and dynamic analysis, in which the operations are continuously executed several times for each memory address.

The static analysis consists of write operations with a fixed data pattern (e.g., solid ‘0’, solid ‘1’, or checkerboard patterns). After a specified time, defined by the OBC or a default 10-minute period, the memory is read to detect any difference from the previously written pattern. In (1), it is presented a simplified schematic view of the algorithm [35], where the arrow indicates the addressing order (‘↑’ up or ‘↓’ down), ‘w’ (write) and ‘r’ (read) indicates the operation, and the following number indicates the data background.

$$\begin{aligned} &\downarrow (w0);\\ &\downarrow (r0); \end{aligned} \tag{1}$$

The dynamic analysis performs continuous read and write operations. This algorithm allows the detection of functional faults and emulates a more realistic behaviour [36, 37]. For this purpose, the March C- algorithm (2) was applied. Note that each element enclosed by the parenthesis is applied to each memory address.

$$\begin{gathered} \uparrow (w0);\\ \{\uparrow (r0, w1); \uparrow (r1, w0); \downarrow (r0, w1); \downarrow (r1, w0); \uparrow (r0); \} \end{gathered} \tag{2}$$

5.4 Hardware Architecture

The payload hardware consists of a 6-layer PCB, using the FloripaSat-2 OBDH DaughterBoard standard and following simplified space application design guidelines. In order to enumerate this patterns and constraints, the following section describes the applied techniques and particularities of the payload design. Then, each developed modules receives a brief architectural and functional description.

5.4.1 Design Guidelines

During the board development, regarding the design guidelines, the rules specified in the ESA normative for space qualified printed circuit boards, ECSS-Q-ST-70-12C [10], were used as a reference. However, since this standard embraces the space environment characteristics and high reliability, following these rules would require an extensive study and analysis, besides the higher manufacturing prices due to specific procedures and materials. Then, in order to conciliate the board requirements for dependability and feasibility in terms of development time and funding, this payload follows rules applied in the master student Cezar Rigo thesis [11]. These rules are summarized in Table 5.6 and some observations are noted concerning the compliance of these rules in the payload project design.

Table 5.6: Summary of ESA guidelines.

Rule	*Compliance*
The distribution of layers and their thickness must be symmetrical. This is done to prevent the board from deforming or distorting, especially when under intense vibration.	Full
The copper thickness on both sides of a laminate must be the same, except in the external board layers.	Full
The outer and inner layers must use basic copper thickness of 70 μm, 35 μm or 17 μm. This is necessary for the manufacturing process.	Full

Table 5.6 – *Continued from previous page*

Rule	*Compliance*
The distribution of copper within a layer must be homogeneous. This ensures homogeneous pressure during lamination, avoiding empty spaces, cracks, or compression of the glass. Mainly needed when using 70 μm thickness.	Full
Tracks should not be routed in layers of copper plane. This is done to avoid obstacles for the high-frequency return currents.	Full
The projected insulation distance between two layers must be at least 100 μm. This avoids risk of reduced electrical insulation.	Full
The fabricated thickness of a rigid epoxy PCB must be $\leq$ 2.4 mm.	Full
The number of copper layers for an epoxy PCB must be $\leq$ 20.	Full
The relationship between the thickness of the PCB and the diameter of the via hole on a rigid PCB must be $\leq$ 7.	Full
For copper thickness 17 μm, no width of track or track spacing and/or components must be $\leq$ 96 μm, already including manufacturing tolerances.	Full
Tracks should not be routed in external layers, given a higher vulnerability to disruption.	Partial [7]
BGA component pads must be circular and with teardrop.	Full
Minimum distance between BGA pads should be 1mm.	None [8]
BGA component pads must be designed with via on the pad, being micro vias or filled vias.	None [9]

---

[7] Since no vias on pad were used due to the high cost, there are some tracks on outer layers.

[8] The FPGA family used in this project only offer 0.8mm BGA pads, which is slightly lower than this rule, but large enough to avoid common problems of very small BGA packages.

[9] This technique was not used due to its high cost.

Table 5.6 – *Continued from previous page*

Rule	*Compliance*
BGA footprint trails must be equidistant between the pads. This increases the distance isolation.	Full
Pads must be designed with teardrop reinforcement.	Full
Non-functional pads should be kept on footprint.	Partial [10]
Annular ring must be at least 50 μm wide in internal layers and 200 μm in external ones.	Full
Copper planes larger than 10 cm^2 must be made in grid format, with 45 degree rotation. This creates ventilation openings and moisture evaporation of the inner layers.	Full
The fabricated thickness of a rigid epoxy high-frequency PCB must be ≤ 3 mm.	Full
The number of copper layers for a rigid epoxy high-frequency PCB must be ≤ 14.	Full
Dielectric strength of the material must be greater than 1000 Vrms/mm	Full
For power distribution, a thickness of copper of 35 μm or more should be used.	Partial [11]
Any conducting elements and copper islands should be left isolated.	Full
The temperature of the tracks must be less than 85 degree Celsius.	Full
Critical tracks should all be routed on one single layer.	Partial [12]
Conductors in outer layers must be covered with polymeric film and not the solder mask.	Partial [13]

---

[10] Due to size limitations, some unused pads were removed, but restricted to the BGA components.

[11] The power density of the payload is very low and have a distribution over planes, then thinner internal planes were used.

[12] Again due to size and cost (and consequently layers) limitations, critical tracks had to use 2 layers, in which at least the care to route them in protected and referenced internal layers was taken.

[13] The board was designed to use solder mask due to the difficulty of assembling components without it. It is planned to use polymeric film before the flight integration.

Table 5.6 – *Continued from previous page*

Rule	*Compliance*
Tracks with impedance control must be made in specific layers.	Full
Vias should not be used on impedance controlled tracks, except for the connection with the component pads.	Partial [14]
Low-speed digital circuits must be separate from high-speed digital circuits.	Full
Grounding planes must be used in high-speed digital applications.	Full
High-speed signals must be routed above of uninterrupted power plans.	Full
The location of the components must be such that each solder connection can be visually inspected.	Full
The designed test points must be separated component pads.	Full
The footprint fanout must be symmetrical. This is done to prevent the component from moving in specific direction during the soldering process.	Full
There must be a minimum distance of 0.25 mm between the pads and the vias. This is done to prevent the solder being removed from the pad by the via, during the soldering process.	Full
The width of a track must be less than 1/3 of the width of a pad. This prevents the track from removing heat from the pad during the soldering process.	Full
The solder mask should not be used as it has problems related to outgassing.	None [15]

---

[14] In the memories routing, mentioned in further sections, more than one layer was required due to the quantity of signals and parallel connections.

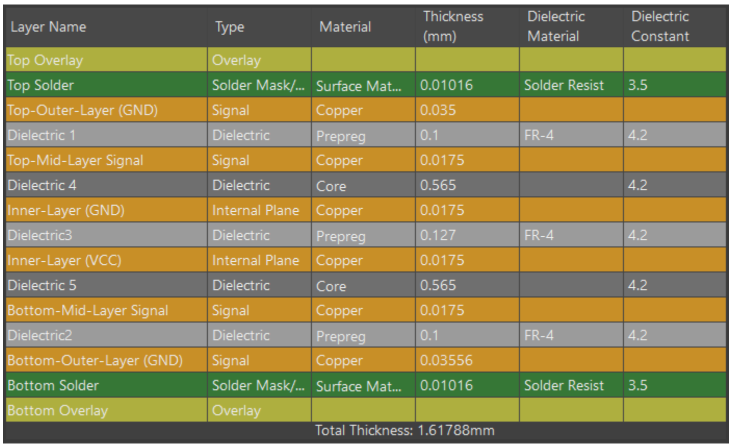
[15] As mentioned before, the board was designed for easier assembly due to size limitations. A bakeout process is intended to be executed, mitigating the outgassing effects.

5.4.2 Preliminary Design Analysis

Layer Stackup Planning

The payload layers scheme was specified to attend the rules herein described and to increase the feasibility of production, which uses convenient dimensions for the target manufacturer. Since the engineering model do not require space qualified boards, this scheme allowed to reduce the costs and time by using a less controlled manufacturing processes. Figure 5.18 describe the payload stackup, including: layers, dimensions and characteristics. In order to avoid tracks in the external layers, the strategy was to use the middle ones for routing, permitting the shield properties of these external layers, and suitable power planes in the internal ones. Moreover, to reduce crosstalk between the signal layers, they are orthogonal concerning the routing preferential directions. Despite the external layers assigned for ground reference, some parts are used to handle the external supply voltages to prevent splitting the power plane since this leads to a degradation of the expected current return paths. In summary, to correctly employ this stackup, the internal layers must have the lowest impedance (short return paths) and the external providing electromagnetic interference shielding to internal signals.

Figure 5.18: Board stackup planning.



Layer Name	Type	Material	Thickness (mm)	Dielectric Material	Dielectric Constant
Top Overlay	Overlay				
Top Solder	Solder Mask/...	Surface Mat...	0.01016	Solder Resist	3.5
Top-Outer-Layer (GND)	Signal	Copper	0.035		
Dielectric 1	Dielectric	Prepreg	0.1	FR-4	4.2
Top-Mid-Layer Signal	Signal	Copper	0.0175		
Dielectric 4	Dielectric	Core	0.565		4.2
Inner-Layer (GND)	Internal Plane	Copper	0.0175		
Dielectric3	Dielectric	Prepreg	0.127	FR-4	4.2
Inner-Layer (VCC)	Internal Plane	Copper	0.0175		
Dielectric 5	Dielectric	Core	0.565		4.2
Bottom-Mid-Layer Signal	Signal	Copper	0.0175		
Dielectric2	Dielectric	Prepreg	0.1	FR-4	4.2
Bottom-Outer-Layer (GND)	Signal	Copper	0.03556		
Bottom Solder	Solder Mask/...	Surface Mat...	0.01016	Solder Resist	3.5
Bottom Overlay	Overlay				

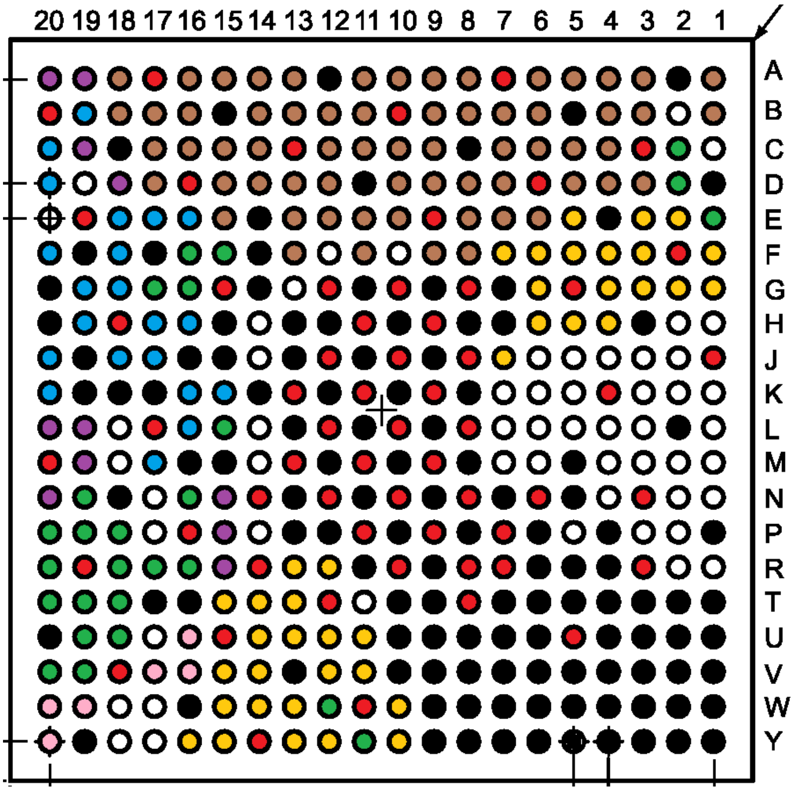
Total Thickness: 1.61788mm

Source: Author.

Placement Planning

The components placement were defined using the external connectors accessibility, considering the position in relation to the motherboard (the FloripaSat-2 OBDH), and internal requirements. However, the positioning of the FPGA device and memory chips were the major factor considered, since the most critical routing is between these components. Also, to ease the layout development process, the memories were symmetrically placed in relation to the FPGA and uniformly distributed in the board. Figure 5.19 present the FPGA pin assignment to attend these requirements. It is important to note that the DDRIO and MSIOD banks were not used due to the maximum voltage levels allowed, 2.5 Volts. This assignment strategy provides three groups of signals in different directions, which ease the BGA fanout and tracks escape routes: memory control and address signals; memory data signals; and general purpose inputs and outputs.

Figure 5.19: FPGA pinout usage.



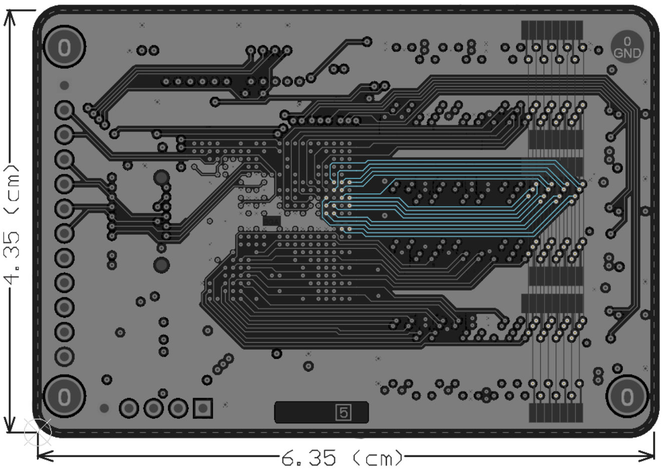
Source: Author.

High-Speed Design Planning

The board component interfaces are in the edge (frequencies of 100MHz and sharp signal transitions) of being considered as high-speed signals due to the SDRAM memories. Then, even if a more complex approach is not required for this work, some measures were considered in the payload design to avoid issues and bad performance. The most important strategy is the properly grounding and power planes for return signals, which were described in the stackup planning topic. Also, orthogonal and similar length routing strategies were adopted to avoid crosstalking and latency issues in the memory signals. The memories are placed symmetrically to provide an easier and organized pattern for routing.

Figure 5.20 presents the signal layer used for tracing from the FPGA to the memory chips. The highlighted tracks are the address signals that are routed together to mitigate the length mismatch. The other signals are grouped together depending on their purpose, control or data signals.

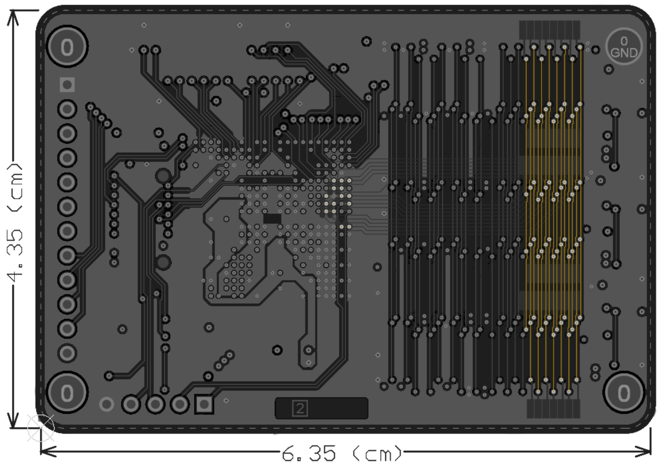
Figure 5.20: Top Mid-layer routing with focus on the memory address signals.



Source: Author.

Figure 5.21 shows a similar overview of the same address signals, but presenting them in the other signal layer. The tracks are traced in the shortest routes as feasible and provide the required parallel connections, since the three memories share the same controller pins interface.

Figure 5.21: Bottom Mid-layer routing with focus on the memory address signals.

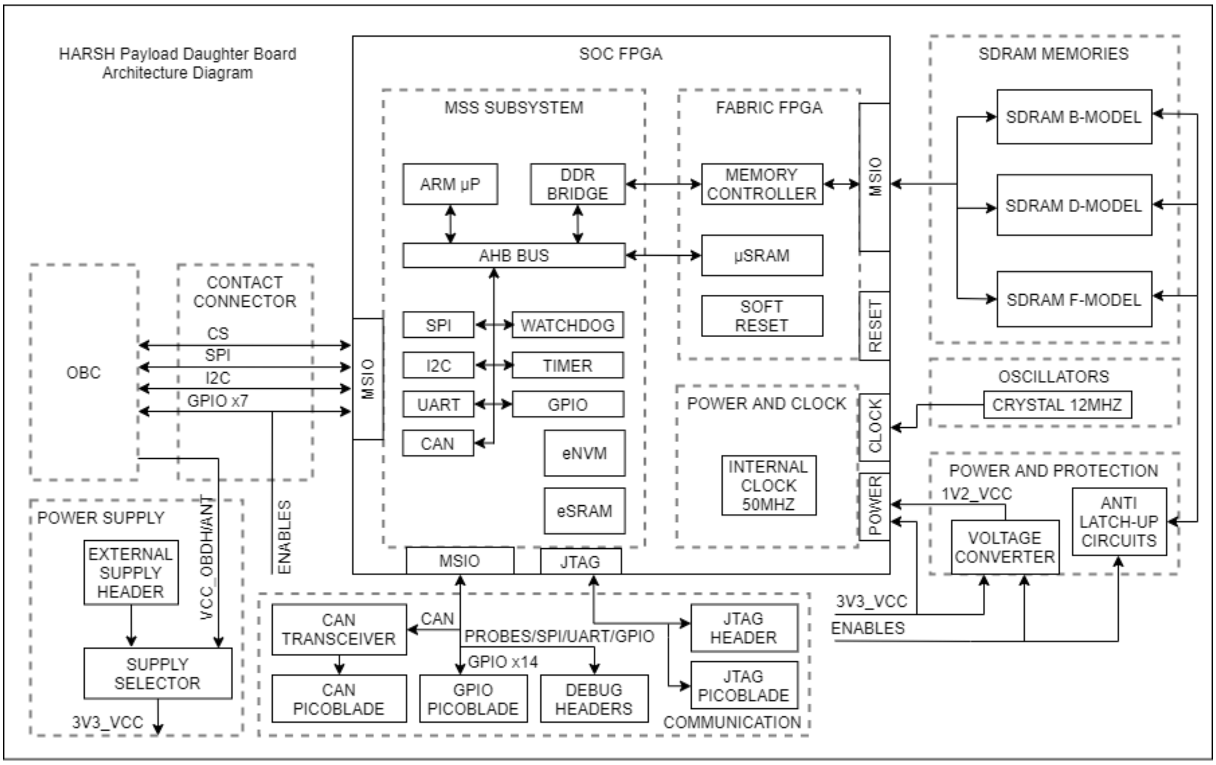


Source: Author.

5.4.3 Design Overview

The payload hardware essentially consists of a controller, to manage communication and memories, and the SDRAM devices under experiment. However, some auxiliary components and modules are necessary to handle other tasks and fulfill the requirements, including: power converter, to supply the correct FPGA voltage; latch-up monitors, to prevent catastrophic system failures and provide additional data for the experiment; debug support for power supply, logging, and programming; additional communication modules and connectors; and the motherboard interface connector. Figure 5.22 presents the payload subsystems, internal connections and external interfaces, besides an internal overview of the SoC FPGA device implemented architecture.

Figure 5.22: Hardware architecture overview.

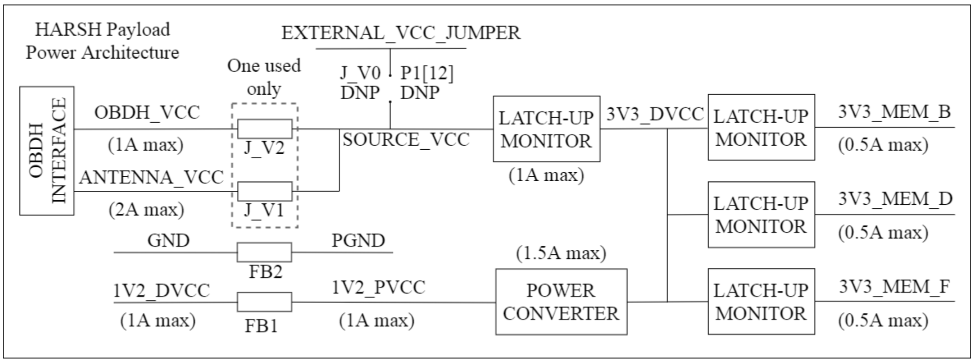


Source: Author.

In addition to the summarized power supply architecture presented, Figure 5.23 shows in more details the power line names, voltages, and currents, besides their relation with the latch-up monitors.

Figure 5.24 presents a rendered view of the board top and bottom layers and Figure 5.25 shows the manufactured board. The following topics describe the subsystems functionality and characteristics. Also, the

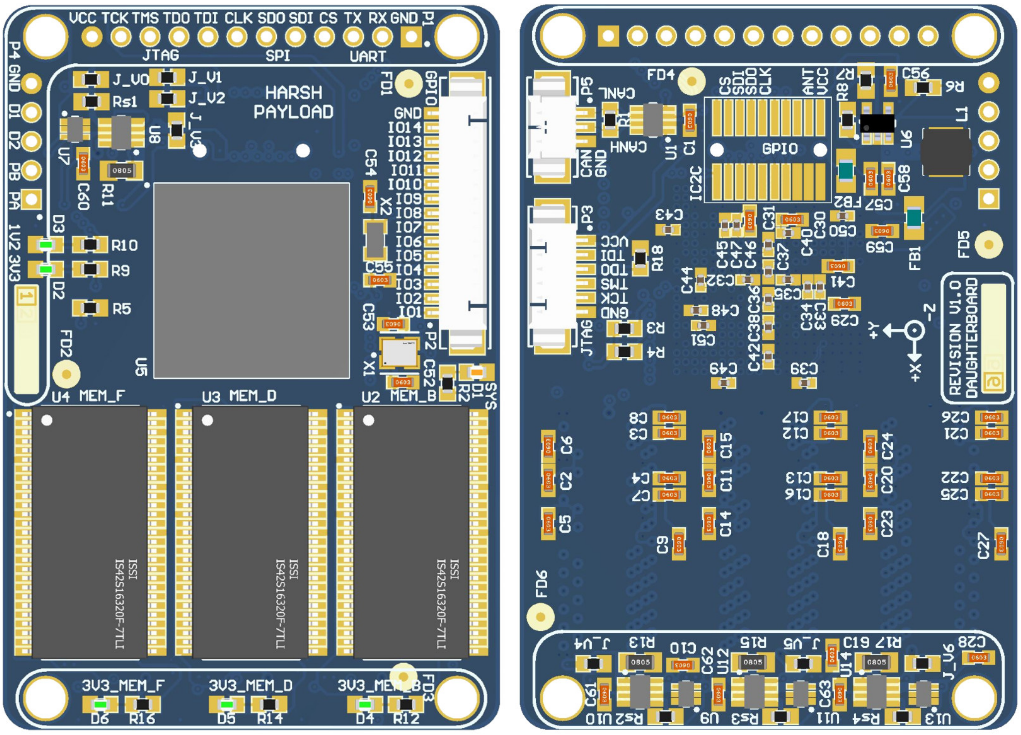
Figure 5.23: Power architecture overview.



Source: Author.

schematics are presented in Appendix A, alongside layers prints and other information.

Figure 5.24: PCB 3D rendering overview.

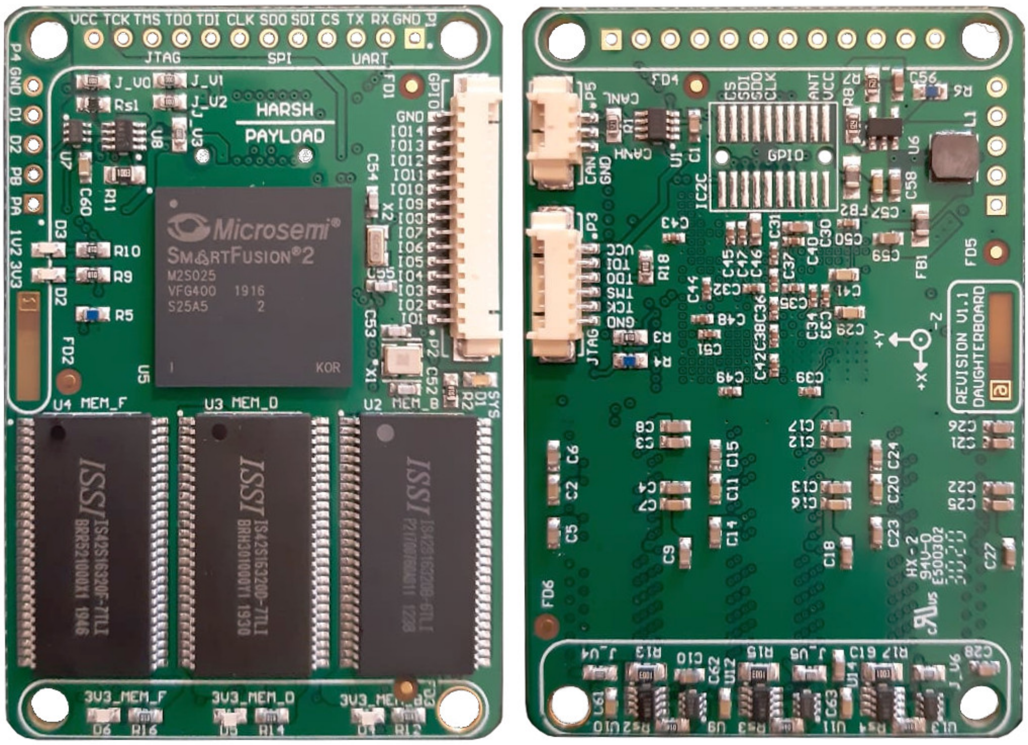


Source: Author.

SoC FPGA Module

The FPGA device is a Ball Grid Array (BGA) component that require a fanout implementation and escape routes for the internal pins. The adopted strategy for this work is the dog-bone pattern, which is a widespread technique and attend the project necessities. Figure 5.26

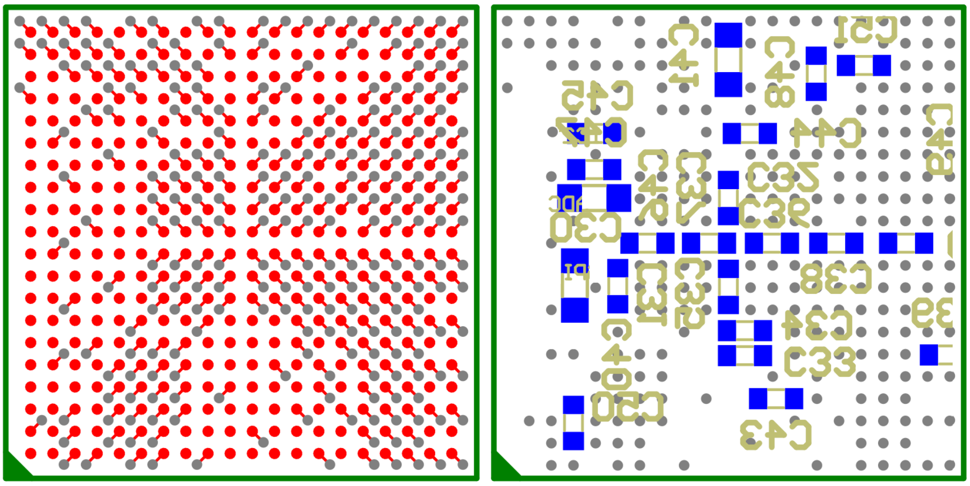
Figure 5.25: PCB manufactured overview.



Source: Author.

presents the implemented pattern, escape route, and decoupling capacitors placement using a top view of the board. Since several FPGA pins are not used and the board placement favors the routing, this implementation only required two signal layers and two power planes, which attend the payload specification.

Figure 5.26: FPGA fanout dog-bone pattern and decoupling capacitors implementation.



Source: Author.

Memories Module

The memories module is composed of the three experiment memory chips: IS42S16320B, IS42S16320D, and IS42S16320F. These devices are 512Mb (or 64MB) Single Data Rate (SDR) SDRAMs operating at 3.3V and up to 143MHz in a 54-pin TSOP-II packaging. A SDR device means that it can only read/write one time in a clock cycle, differing from its successors Double-Data Rate (DDR) memories that perform two operations per cycle. The selected parts configuration has a 16-bit data width and the industrial grade qualification.

The target operation frequency is 100MHz using the other parameters in the nominal indications, except for the refresh frequency (8K cycles every 64ms) that might be changed during some experiment tests. The electrical topology is a parallel connection that shares the same controller interface. This strategy was adopted since it saves the usage of several FPGA pins and the used configuration is the same.

During the board design, the placement was selected to avoid any component unrelated to the memories be located on their opposite side, with exception of their decoupling capacitors, ensuring that they are as close as possible of their power pins. Regarding the routing, the strategies employed were described in subsubsection 5.4.2.

Communications and Debug Interfaces

The payload uses several communication interfaces distributed in different connectors according to the requirements or usage. There are six available interfaces for different purposes: SPI, JTAG, UART, I2C, GPIOs, and CAN. The SPI, JTAG, and UART are the main used communication protocols and the rest were added for redundancy or additional non-critical features. In the same scheme, there are six different connectors that share some interfaces: 3 picoblade connectors, 2 debug headers, and 1 contact connector. The debug headers are not used in the flight configuration due to size limitations.

The picoblades are separately utilized for a CAN channel, a JTAG interface, and the application GPIOs. The debug headers provide an easier access for the UART logger, the application SPI, a JTAG interface, debug GPIOs, and a external power supply input for debug. The contact connector is the main interface with the OBC, which provides the application SPI, an I2C for redundant communication, OBC power inputs, and the control GPIOs.

This approach was adopted to improve the design flexibility and allow the adaptation of the payload for different scenarios. For instance, it is planned to perform ground experiments before the payload launch and also the possibility of utilization in different satellite missions.

Power Module

The power subsystem is composed of a DC converter and a simple noise filtering network. In the payload specification is stipulated that a 3.3V power line is reserved for supplying the board, which should ensure current limitation in case of a latch-up. This requirement is accomplished using a anti latch-up circuitry and through the power converter protection as a redundancy.

In order to supply the 1.2V for the FPGA chip, there is a DC switching power converter (TLV62565DBVR). This step-down converter operates with high efficiency (above 85%), supplying up to 1.5A with overcurrent and thermal protections. The selected part is a 5-Pin SOT-23 chip and features an enable input.

The noise filtering network is the combination of four passive components: two ferrite inductors and two ceramic capacitors. This circuit filters the high frequency noises generated by the switching converter and received from the OBC. Then, there are 1.2V power lines and power grounds before and after the filter, which allows better power performance for the rest of the circuits.

Latch-up Monitors

The latch-up circuitry consists of four monitors: one for the entire board and three for individually controlling the experiment memories. The circuit is based on the LTC4361ITS8-1 device that provides the required overcurrent protection through the management of a MOSFET transistor (SI1416EDH-T1-GE3). The monitor uses a SOT-8 package and the transistor chip a SOT-363-6 package, which offers a compact and efficient solution. This system is completely independent of a controller for opening the circuit in case of non nominal condition, but dependent for closing again. The memory monitors are controlled by the SoC FPGA in case of an event and the general monitor is only controlled by the OBC, since the payload is completely turned off during this condition. The system also provides a status pin that is used for detecting these events.

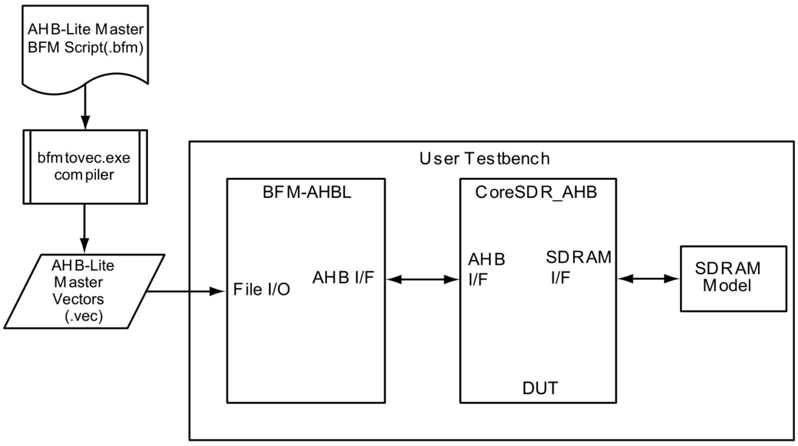
6 Results

The payload subsystems in hardware, firmware, and SoC FPGA were tested during the development for later be integrated with the FloripaSat-2 OBDH. Since there were several simulations and tests to achieve enough reliability for integration, this section focus on presenting the most critical tests and their results. The test strategy focus on evaluating the lower level elements and then validating the payload operation as a system. Before the actual testing, some simulations were performed. The final assessments consisted of a radiation test on the board and the actual integration with the FloripaSat-2 platform. The requirements were used to assess the readiness of the payload.

6.1 Simulations

In order to validate the first concepts, a simulation before the design synthesis was performed intending to verify mainly the SDRAM controller. The Microsemi development tool (Libero) has integration with ModelSim (a simulation software) and provide the Bus Functional Model (BFM) solution, which enables the emulation of the communication between MSS and fabric logic through the AHB. Figure 6.1 demonstrates the simulation flow using the BFM scripts. The user defines the commands in the BFM script, then a compiler converts them to a vector file that is further used as an input for the testbench system, which consists of an AHB master interface, the memory controller, and the emulated memory. Since the memory controller is part of the Microsemi catalog, the sources of this testbench scheme are provided to support the early design validation.

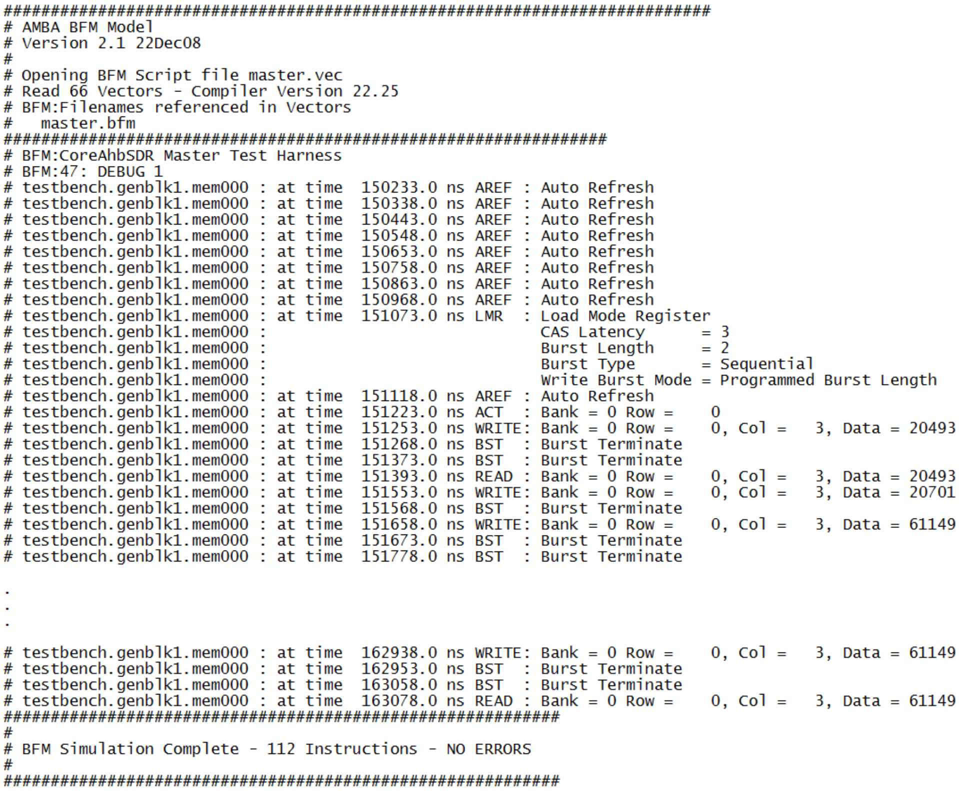
Figure 6.1: Memory controller simulation flow.



Source: Microsemi, 2020.

For this simulation, the script performed explicit write and read operations successively, leaving the task of setup and refresh commands to the memory controller, since it must independently execute this operations. The simulation outputs are logged in a text format detailing events, operations, and their timestamps. Figure 6.2 presents a summarized log output.

Figure 6.2: Memory controller simulation results: log output.



```
###########################################################################
# AMBA BFM Model
# Version 2.1 22Dec08
#
# Opening BFM Script file master.vec
# Read 66 Vectors - Compiler Version 22.25
# BFM:Filenames referenced in Vectors
#   master.bfm
################################################################
# BFM:CoreAhbSDR Master Test Harness
# BFM:47: DEBUG 1
# testbench.genblk1.mem000 : at time  150233.0 ns AREF : Auto Refresh
# testbench.genblk1.mem000 : at time  150338.0 ns AREF : Auto Refresh
# testbench.genblk1.mem000 : at time  150443.0 ns AREF : Auto Refresh
# testbench.genblk1.mem000 : at time  150548.0 ns AREF : Auto Refresh
# testbench.genblk1.mem000 : at time  150653.0 ns AREF : Auto Refresh
# testbench.genblk1.mem000 : at time  150758.0 ns AREF : Auto Refresh
# testbench.genblk1.mem000 : at time  150863.0 ns AREF : Auto Refresh
# testbench.genblk1.mem000 : at time  150968.0 ns AREF : Auto Refresh
# testbench.genblk1.mem000 : at time  151073.0 ns LMR  : Load Mode Register
# testbench.genblk1.mem000 :                             CAS Latency      = 3
# testbench.genblk1.mem000 :                             Burst Length     = 2
# testbench.genblk1.mem000 :                             Burst Type       = Sequential
# testbench.genblk1.mem000 :                             Write Burst Mode = Programmed Burst Length
# testbench.genblk1.mem000 : at time  151118.0 ns AREF : Auto Refresh
# testbench.genblk1.mem000 : at time  151223.0 ns ACT  : Bank = 0 Row =     0
# testbench.genblk1.mem000 : at time  151253.0 ns WRITE: Bank = 0 Row =     0, Col =   3, Data = 20493
# testbench.genblk1.mem000 : at time  151268.0 ns BST  : Burst Terminate
# testbench.genblk1.mem000 : at time  151373.0 ns BST  : Burst Terminate
# testbench.genblk1.mem000 : at time  151393.0 ns READ : Bank = 0 Row =     0, Col =   3, Data = 20493
# testbench.genblk1.mem000 : at time  151553.0 ns WRITE: Bank = 0 Row =     0, Col =   3, Data = 20701
# testbench.genblk1.mem000 : at time  151568.0 ns BST  : Burst Terminate
# testbench.genblk1.mem000 : at time  151658.0 ns WRITE: Bank = 0 Row =     0, Col =   3, Data = 61149
# testbench.genblk1.mem000 : at time  151673.0 ns BST  : Burst Terminate
# testbench.genblk1.mem000 : at time  151778.0 ns BST  : Burst Terminate

.
.
.

# testbench.genblk1.mem000 : at time  162938.0 ns WRITE: Bank = 0 Row =     0, Col =   3, Data = 61149
# testbench.genblk1.mem000 : at time  162953.0 ns BST  : Burst Terminate
# testbench.genblk1.mem000 : at time  163058.0 ns BST  : Burst Terminate
# testbench.genblk1.mem000 : at time  163078.0 ns READ : Bank = 0 Row =     0, Col =   3, Data = 61149
###########################################################
#
# BFM Simulation Complete - 112 Instructions - NO ERRORS
#
###########################################################
```

Source: Author.

Another output is the waveform of the controller signals. Figure 6.3 shows a part of the simulation execution, regarding refresh, load mode, read and write commands. The parameters were selected in conformity with the datasheet [24] [25] [26] nominal values.

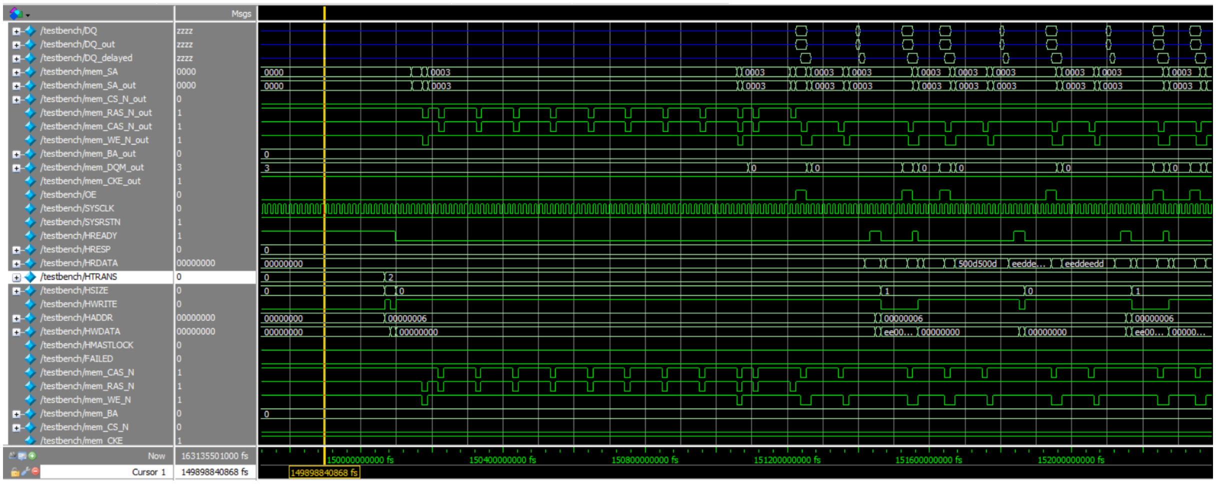
During this design evaluation, some misunderstanding of memory parameters were detected and corrected, such as the refresh rate that was smaller than the minimal recommended. Also, this test ensured that the memories could operate in parallel without losing the timing constraints and functional requirements.

Requirements validated with these simulations: REQ-PAY-040, REQ-PER-000 [16], REQ-PER-010, REQ-PER-020, REQ-OPE-060.

---

[16] As further explained, the operation frequency in the board had limitations not detectable during the simulations.

Figure 6.3: Memory controller simulation results: signal waveforms.



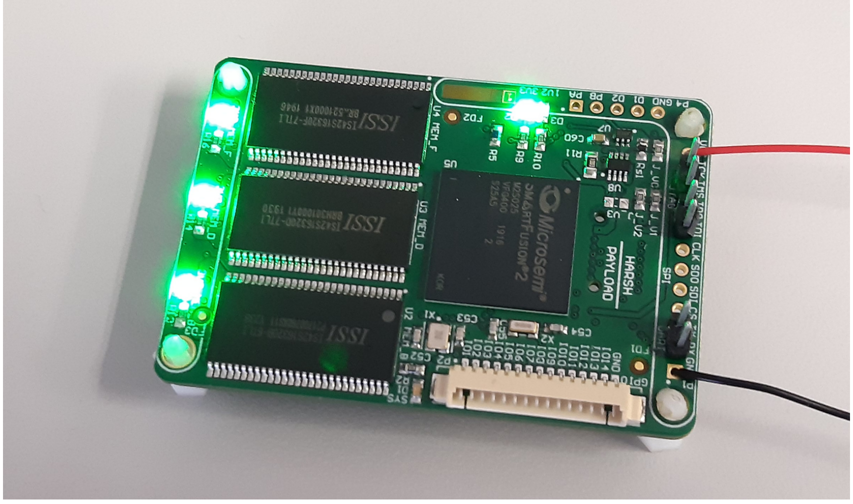
Source: Author.

6.2 Hardware Tests

The board hardware evaluation was performed in steps: visual inspection, mechanical inspection, electrical inspection and electrical testing. During the first inspection, the packaging was assessed to ensure integrity of the board, the 3D renders were compared with the received board, the layer markers checked, and the reports from the manufacturer revised. The mechanical inspection was executed to ensure that the board dimensions and mounting holes were correct. The electrical inspection was important to detect any potential problem concerning the proper electrical operation, such as: solder short circuits, missing components, lifted pins, poor soldering in any pin, swapped components, and device partnumbers. The final step was to perform electrical testing, starting with a continuity evaluation of several points and at the end a power-up procedure. Figure 6.4 and Figure 6.5 presents the final steps of the hardware testing.

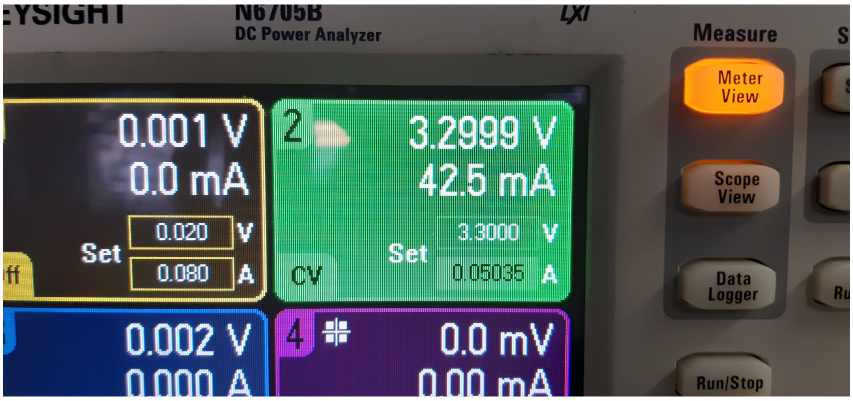
After these procedures, any problem was detected in the hardware design, despite some limitations in the maximum memories operation frequency. It was detected that in the maximum frequency (approximately 143MHz) the memories presented some corrupted data addresses and control signal glitches, attributed to signal interference and integrity. This was expected since the project focused on an 100MHz memory operation and with three devices in parallel is hard to precisely match lengths and return paths. Despite that, this limitation not affect the The debug interfaces supported easy debugging sessions and the status LEDs provided a good visual feedback for the board operation. Further functional testing was performed in the scope of the firmware evaluation.

Figure 6.4: Hardware test results: fully operational board.



Source: Author.

Figure 6.5: Hardware test results: average power consumption.



Source: Author.

Requirements validated with these tests: REQ-PAY-000, REQ-PAY-030, REQ-PAY-040, REQ-PAY-050, REQ-PAY-060, REQ-PAY-070, REQ-ENV-000, REQ-ENV-010, REQ-ENV-020, REQ-ENV-030, REQ-PER-020. Requirements that failed with these tests: REQ-ENV-040. This last requirement was not followed in the board design due to the size limitations and the increased assembly difficulty. Requirements that partially failed with these tests: REQ-PER-000 [17].

6.3 Firmware Tests

The firmware was tested across all the project phases with different approaches. During the prototype, more flexible and specific tests were exe-

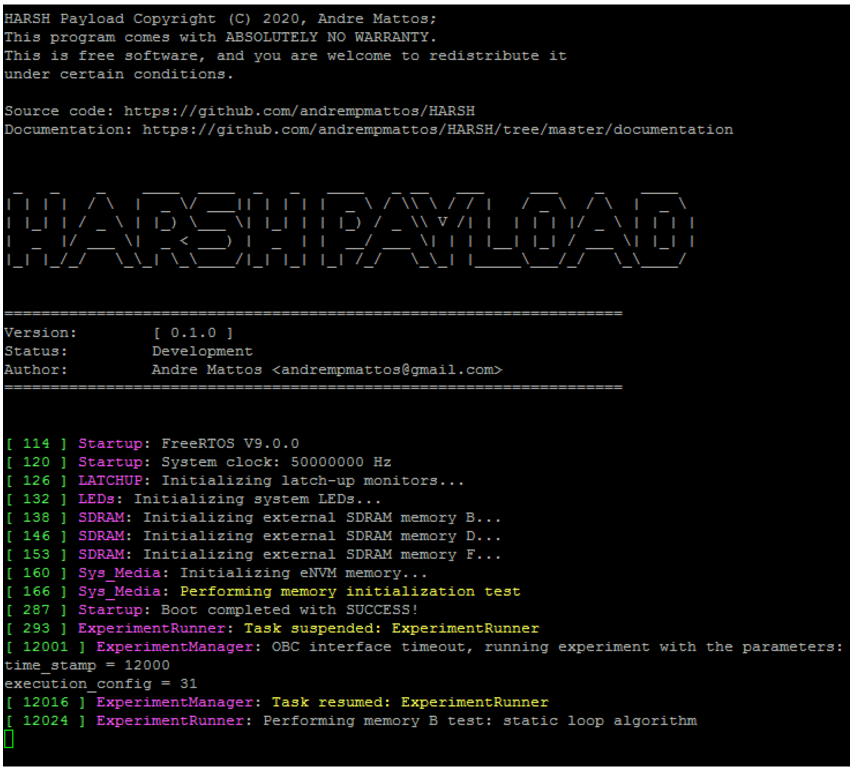
---

[17] As previously explained in this section.

cuted to validate the behavior of peripherals and external devices. Then, starting with the engineering model and the first board manufactured, the firmware was already in a more mature state with several log messages, self-test routines, and the major structure implemented. This lead to functional testing and the requirements were confronted until achieving the desired behavior. When the first integration tests started, the real application in the full state could be excited, providing enough confidence of the payload performance and reliability. Also, a full radiation experiment was performed, which produced more insights about the platform under a simulated harsh environment, as explained in the next section.

In the flight model, the firmware was already tested and integrated with the FloripaSat-2 OBDH. Then, the efforts were applied to the firmware documentation. Figure 6.6 presents an example of log message showing the last version of the firmware.

Figure 6.6: Example of a firmware log message.



```
HARSH Payload Copyright (C) 2020, Andre Mattos;
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions.

Source code: https://github.com/andrempmattos/HARSH
Documentation: https://github.com/andrempmattos/HARSH/tree/master/documentation


 _   _    _    ____  ____  _   _   ____   _ __   ___     ___    _    ____  
| | | |  / \  |  _ \/ ___|| | | | |  _ \ / \\ \ / / |   / _ \  / \  |  _ \ 
| |_| | / _ \ | |_) \___ \| |_| | | |_) / _ \\ V /| |  | | | |/ _ \ | | | |
|  _  |/ ___ \|  _ < ___) |  _  | |  __/ ___ \| | | |__| |_| / ___ \| |_| |
|_| |_/_/   \_\_| \_\____/|_| |_| |_| /_/   \_\_| |_____\___/_/   \_\____/ 


===================================================================
Version:        [ 0.1.0 ]
Status:         Development
Author:         Andre Mattos <andrempmattos@gmail.com>
===================================================================


[ 114 ] Startup: FreeRTOS V9.0.0
[ 120 ] Startup: System clock: 50000000 Hz
[ 126 ] LATCHUP: Initializing latch-up monitors...
[ 132 ] LEDs: Initializing system LEDs...
[ 138 ] SDRAM: Initializing external SDRAM memory B...
[ 146 ] SDRAM: Initializing external SDRAM memory D...
[ 153 ] SDRAM: Initializing external SDRAM memory F...
[ 160 ] Sys_Media: Initializing eNVM memory...
[ 166 ] Sys_Media: Performing memory initialization test
[ 287 ] Startup: Boot completed with SUCCESS!
[ 293 ] ExperimentRunner: Task suspended: ExperimentRunner
[ 12001 ] ExperimentManager: OBC interface timeout, running experiment with the parameters:
time_stamp = 12000
execution_config = 31
[ 12016 ] ExperimentManager: Task resumed: ExperimentRunner
[ 12024 ] ExperimentRunner: Performing memory B test: static loop algorithm
```

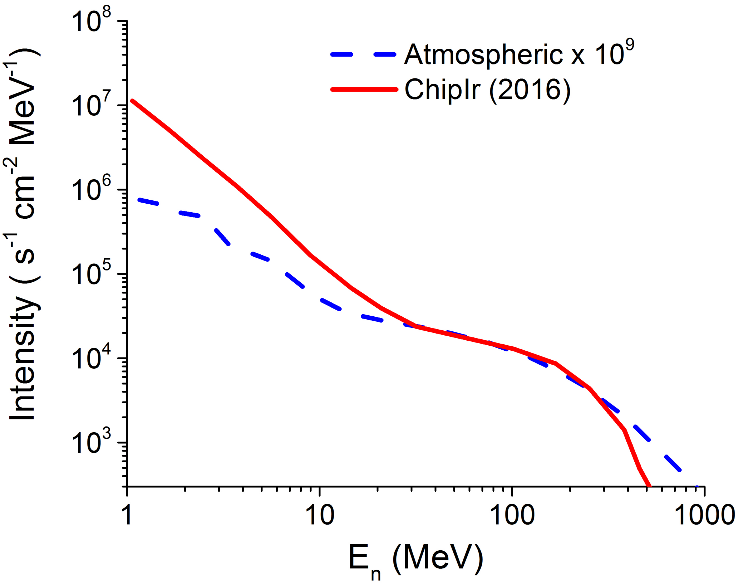
Source: Author.

Requirements validated with these tests: REQ-PAY-070, REQ-PER-030, REQ-OPE-000, REQ-OPE-010, REQ-OPE-020, REQ-OPE-030, REQ-OPE-040, REQ-OPE-050, REQ-OPE-070, REQ-OPE-080, REQ-OPE-090, REQ-OPE-100.

6.4 Radiation Tests

During the development and improvements for the flight model, it was possible to evaluate the payload under experimental radiation conditions in the ChipIr beamline [38] [39]. The experiment was intended to provide only a preliminary overview of the platform under radiation environments due to schedule and budget restrictions for more extensive and broad assessments. Two identical boards were exposed to high energy neutrons during continuous hours before presenting critical damage and reported several events. This experiment represented a great evaluation opportunity for the payload during the critical transition of the engineering model to the flight system, providing precious data to enhance the testing routines and identify potential fault mechanisms. Despite that, the experiment presented some limitations since the type of radiation used was not the main target of the experiment (atmospheric-like neutrons instead of more energetic and diverse particles found in LEO). As presented in Figure 6.7, the payload was exposed to a irradiation beam that produces neutron particles in a controlled intensity, spectrum, and area. Note that the beam intensity is several orders of magnitude higher than the atmospheric conditions, which turns feasible only few hours of experimental exposure representing months in the real environment.

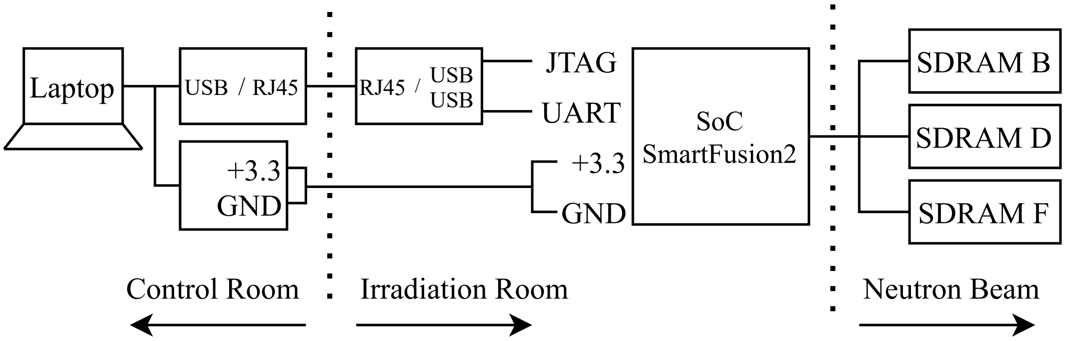
Figure 6.7: Experiment radiation parameters.



Source: ChipIr, 2020.

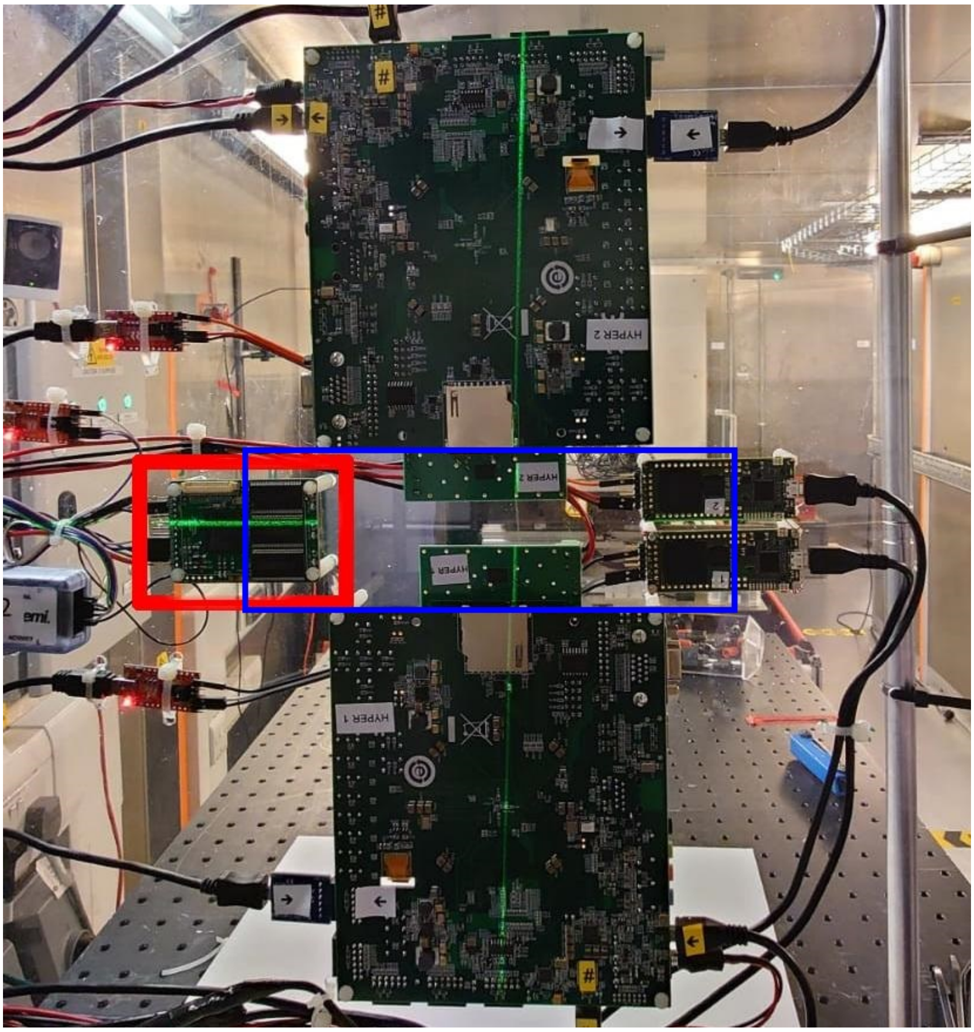
Figure 6.8 describes a simplified diagram of the test setup used during the experiment. Figure 6.9 presents the actual setup mounted inside the experiment facilities.

Figure 6.8: Radiation experiment setup diagram.



Source: L. Luza et al., 2021

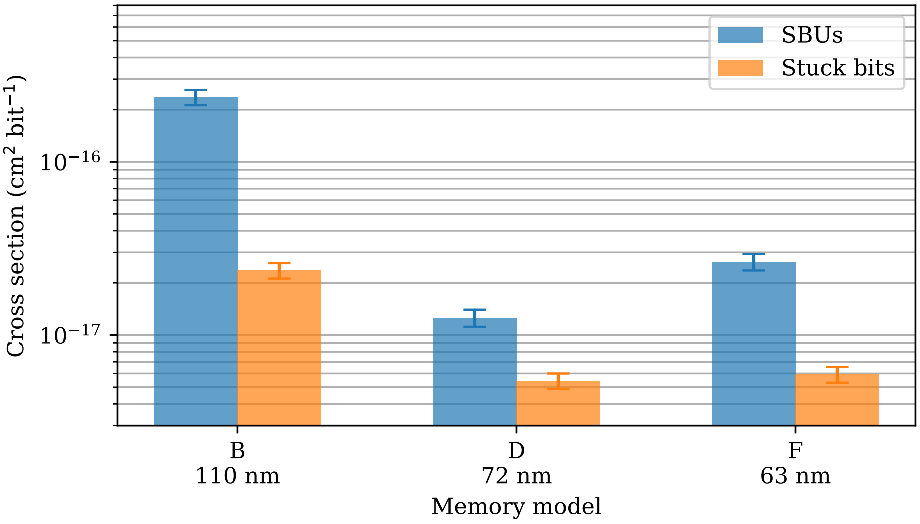
Figure 6.9: Payload board in red fixed inside the beam area in blue.



Source: L. Luza et al., 2021

The scope of this work focus on the development aspects of the platform, then the results acquired from radiation experiments, in ground facilities or in orbit, are used for the production of future scientific studies [40]. Despite that, a preliminary result achieved was the comparison between the technological differences of the memory devices. Figure 6.10 shows a summary of the analysis of the experiment, which is better described in a work under review by the author and other colleagues. The experiment exposed Single Bit Upsets (SBU) and *stuck bits*[18]. The results point that the technological node scaling is not the only factor changing the error cross sections, but also the manufacturing enhancements and capabilities (materials, foundry equipment, physical implementation enhancements, and other processes) interfere in the analysis. Despite still preliminary, this result is interesting since it is expected that reducing the node size tend to increase the error cross sections, which is not seen in the results transition from 110nm to 72nm. The scaling from 72nm to 63nm follows the expected trend and support the evidence that the experiment was properly performed.

Figure 6.10: Radiation cross section results according with the SDRAM memories technology scaling (models B, D, and F).



Source: L. Luza et al., 2021

Requirements validated with these tests: REQ-ENV-010, REQ-OPE-010, REQ-OPE-100, REQ-PER-010, REQ-PER-020. Requirements that partially failed with these tests: REQ-PER-000 [19].

---

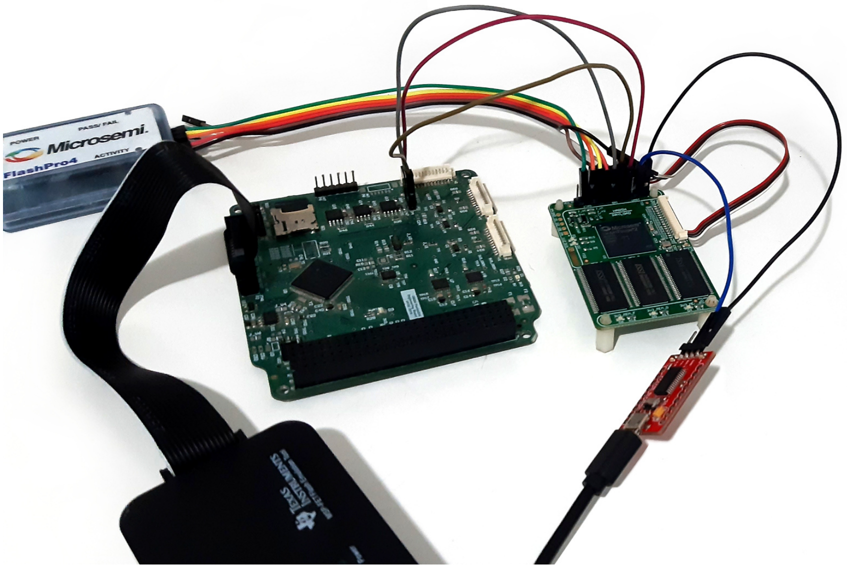
[18] A effect similar to a SBU, but with an extended failure period, resisting to multiple read and writes and then restoring to the expected cell behavior

[19] As previously explained.

6.5 Engineering Model Integration

The integration with the engineering model occurred in three different steps: the first using a Raspberry Pi as OBC, the second with an older version of the FloripaSat-2 OBDH (the FloripaSat-1 OBDH), and the last using the FloripaSat-2 OBDH engineering model. In order to perform the first tests, a Raspberry Pi 3 was used to emulate the OBC behavior and communication. This approach was necessary due to limited access to the actual OBC. Despite the hardware differences, this test setup allowed the debug of the most critical communication issues and the improvement of the interface protocol. The integration with an older FloripaSat-1 OBDH was necessary due to similar limitations. Figure 6.11 present this test setup.

Figure 6.11: Integration setup with FloripaSat-1 OBDH engineering model



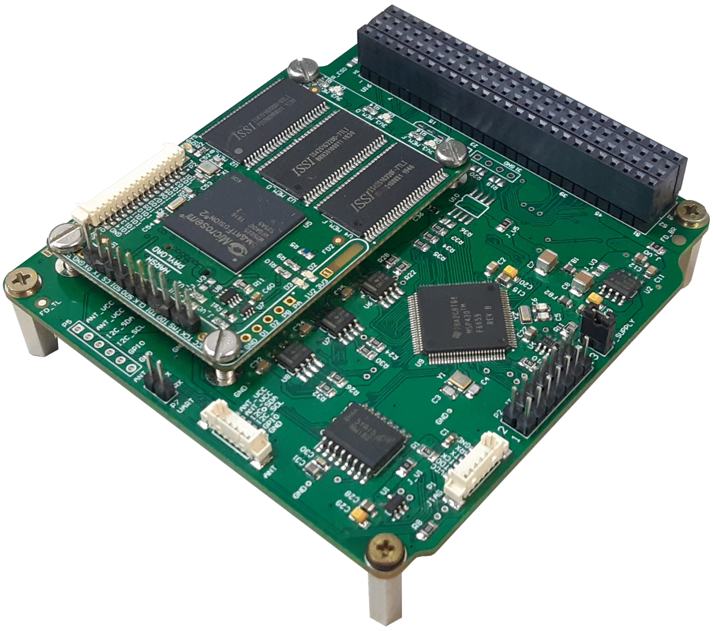
Source: Author.

Finally, with sufficient access to a proper FloripaSat-2 OBDH board, the final engineering model testing could be done. Figure 6.12

Figure 6.13 presents a command request from the OBC during tests with the FloripaSat-2 OBDH. It is important to note that the communication between the payload and the OBC uses the FloripaSat Protocol (FSP). In the figure is shown 8 bytes: header, destination address, source address, package type, number of payload packages, payload (command in this case) and the last two for CRC.

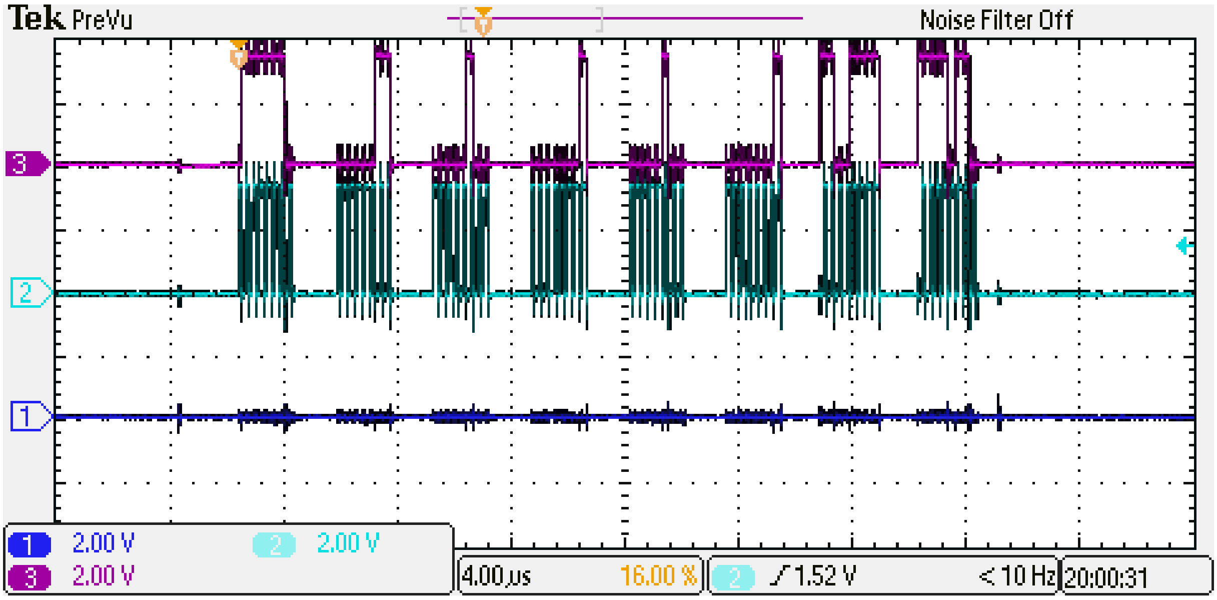
Figure 6.14 presents the payload acknowledgement answer due to the master request. The same package structure is observed, differing just for the type, payload content, address positions, and CRC check.

Figure 6.12: Engineering model payload integrated with the FloripaSat-2 OBDH.



Source: Author.

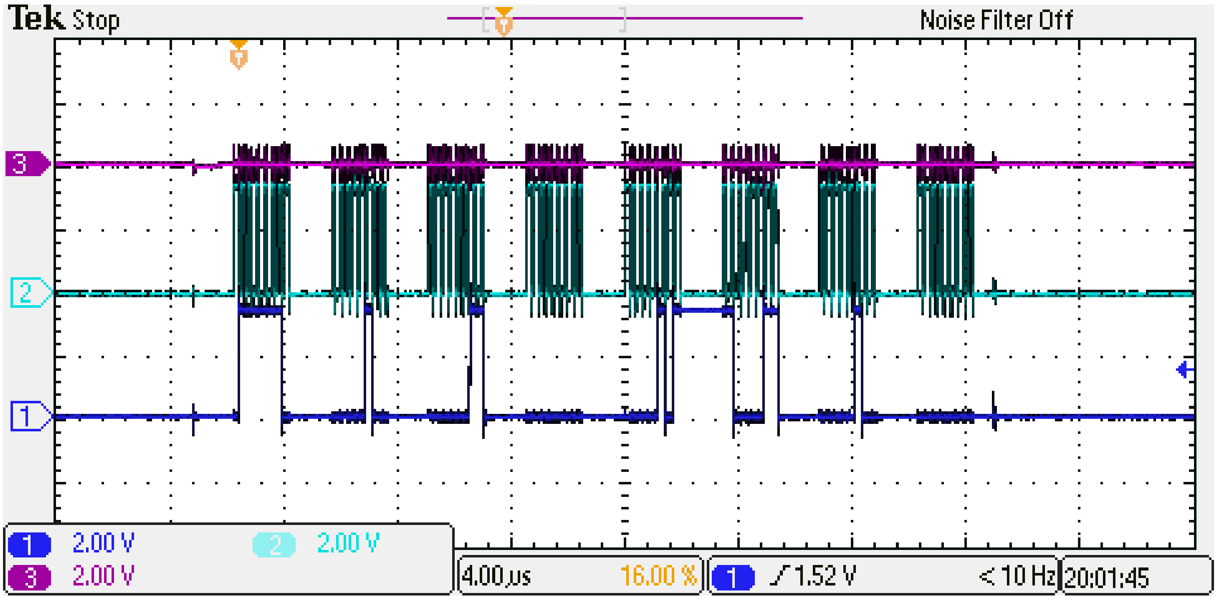
Figure 6.13: Example SPI command communication signals from the OBC to the payload using the FSP protocol.



Source: Author.

All requirements, with few exceptions, were exercised with the integration since all features of the payload are assessed. However, the most important requirements validated with this integration: REQ-PAY-000, REQ-PAY-010, REQ-PAY-020, REQ-PAY-030, REQ-PAY-060, REQ-PER-030, REQ-OPE-030, REQ-OPE-040, REQ-OPE-050, REQ-OPE-080.

Figure 6.14: Example SPI acknowledgement answer communication signals from the OBC to the payload using the FSP protocol.

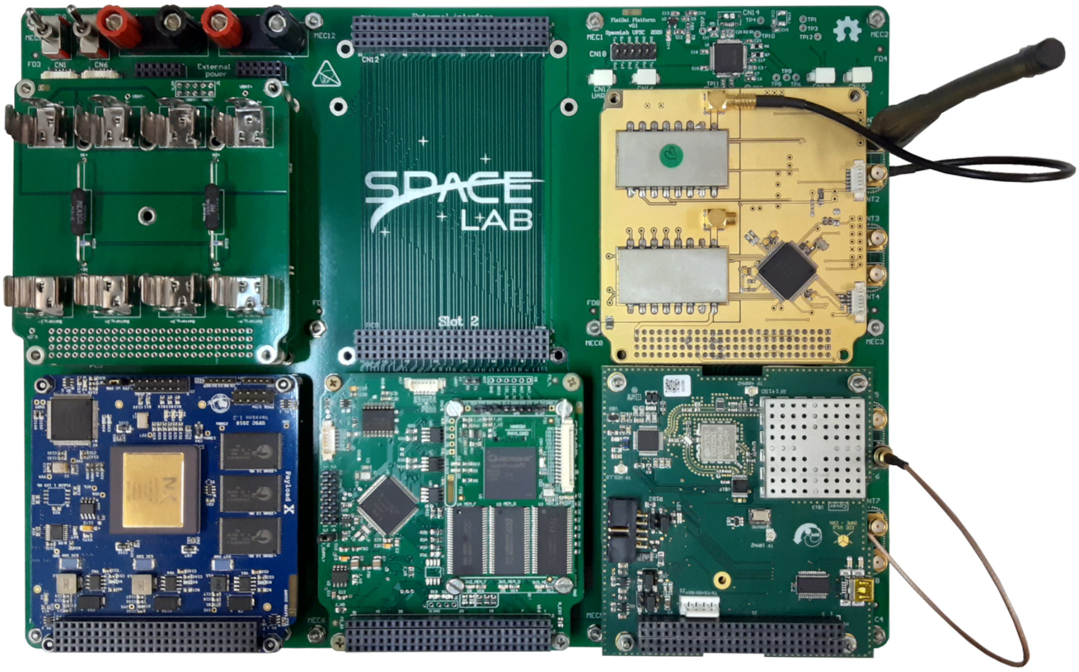


Source: Author.

6.6 Flight Model Integration

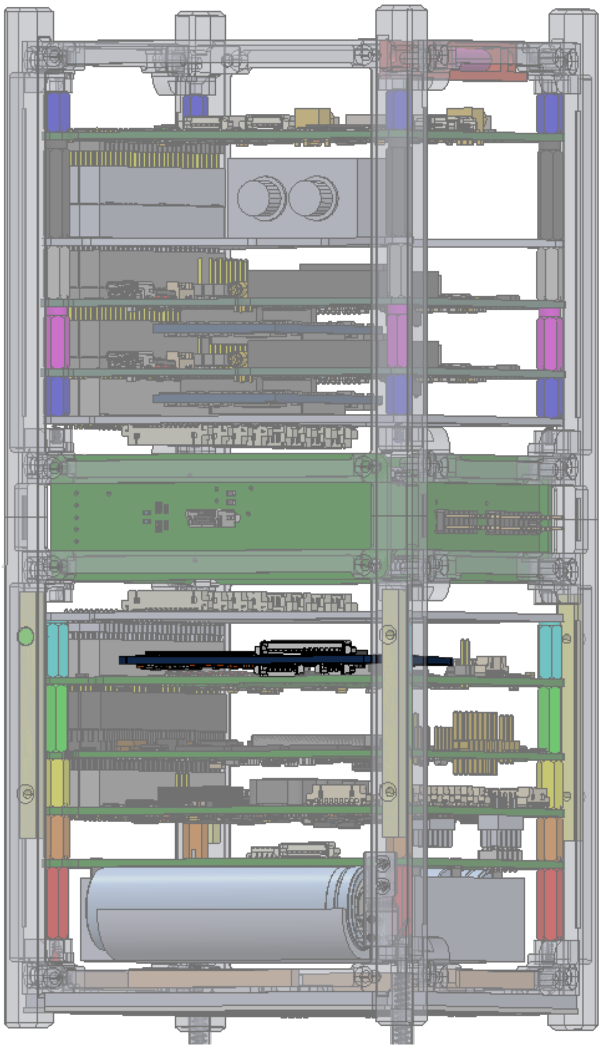
The payload flight model was integrated with the FloripaSat-2 platform in the FlatSat phase as presented in Figure 6.15. The system performed as intended during these tests since the actual interface between the payload and satellite continued the same (i.e., a communication and power bus provided by the FloripaSat-2 OBDH). Due to schedule limitations, the real integration with the FloripaSat-2 satellite in the final form factor was not possible, but all the required steps were planned and the payload is waiting for the satellite launch. Figure 6.16 shows a schematic view of the satellite with the payload highlighted.

Figure 6.15: Flight model payload model integrated with FloripaSat-2 platform in the FlatSat.



Source: SpaceLab, 2021.

Figure 6.16: Flight model payload model integrated with FloripaSat-2 platform (payload highlighted).



Source: SpaceLab, 2021.

7 Conclusion

The development of a complete payload experiment was presented, achieving the desired mission objectives. Also, this work lead to relevant outcomes for both research laboratories, since the payload is currently being used for different radiation assessments in the memory devices and is planned to be launched alongside the FloripaSat-2 satellite in a complete space mission. Concerning the technologies involved in the payload, the approach and design presented some improvements in comparison with the previous similar missions developed in both laboratories since some new concepts were employed, such as the use of a RTOS for a more robust firmware system and the creation of a multipurpose device in relatively controlled budget.

The project management techniques facilitated and structured the development, from defined phases and base requirements. The final verification and integration tests presented satisfactory overall performance and reliability, but revealed some bottlenecks. In firmware, the payload allowed a robust framework for future utilization and good maintenance. In hardware, the board work properly from the first production, but presented limitations in the maximum memory frequency operation (in comparison with the maximum operation allowed). In the FPGA design, the system performed properly, but more investigation to improve the radiation resilience could increase the reliability of the experiments and its duration before presenting critical failures.

For future work, the payload might be exposed to more broad radiation sources and energies, providing a better assessment of the system under radiation. Also, this could be used to augment and corroborate the results found in the experiments performed in the space missions.

In summary, the payload accomplished its requirements and purpose, brought some enhancements to the previous employed methods, reported a complete CubeSat payload development campaign, and provided the necessary tools for personal development and learning experience. Future works are planned since after the FloripaSat-2 successful launch, the experiment data will be provided and other ground experiments are intended, allowing data analysis and further investigation in the memory devices.

References

[1] National Aeronautics and Space Administration, Mission Design Division. Small Spacecraft Technology State of the Art, NASA/TP–2015–216648/REV1, 2015.

[2] Wiley J. Larson; James R. Wertz. Space Mission Analysis and Design, 1999.

[3] Viyas Gupta. Analysis of single event radiation effects and fault mechanisms in SRAM, FRAM and NAND Flash : application to the MTCube nanosatellite project., 2017.

[4] SpaceLab. *FloripaSat-2 Documentation*. Space Technology Research Laboratory, march 2021. Accessed on: April 19, 2021.

[5] SpaceLab. FloripaSat core on-board computer, Acesso em: Dezembro de 2019. Disponível em: <github.com/spacelab-ufsc/obdh2>.

[6] SpaceLab. FloripaSat-I Mission, Acesso em: Dezembro de 2019. Disponível em: <floripasat.ufsc.br/>.

[7] Altera. What is an SoC FPGA?, 2014.

[8] D. Söderström, L. M. Luza, H. Kettunen, A. Javanainen, W. Farabolini, A. Gilardi, A. Coronetti, C. Poivey, and L. Dilillo. Electron-induced upsets and stuck bits in SDRAMs in the jovian environment. *IEEE Transactions on Nuclear Science*, pages 1–1, 2021. DOI: 10.1109/TNS.2021.3068186. [Early access].

[9] Said Hamdioui. Testing static and dynamic faults in random access memories, 2002.

[10] European Space Agency. ECSS-Q-ST-70-12C - Design rules for printed circuit boards, 2014.

[11] C. Rigo, L. Seman, M. Berejuck, and E. Bezerra. Printed Circuit Board Design Methodology for Embedded Systems Targeting Space Applications, 2020.

[12] National Aeronautics and Space Administration. NASA Systems Engineering Handbook - NASA SP-2016-6105 Rev2, 2016.

[13] NASA CubeSat Launch Initiative. Basic Concepts and Processes for First-Time CubeSat Developers, 2017.

[14] E. Petersen. *Single Event Effects in Aerospace.* John Wiley & Sons, Inc., Hoboken, USA, October 2011.

[15] Dyer, C. Radiation effects on spacecraft & aircraft. In *Proceedings of the Second Solar Cycle and Space Weather Euroconference*, 2002.

[16] R. Baumann and Kirby Kruckmeyer. *Radiation handbook for electronics.* Texas Instruments, January 2019.

[17] Steven M. Guertin and Mehran Amrbar. Single event testing of sdram, ddr2 and ddr3 memories. In *2016 IEEE Radiation Effects Data Workshop (REDW)*, pages 1–7, 2016.

[18] Farokh Irom and Mehran Amrbar. Heavy ion single event effects measurements of 512mb issi sdram. In *2015 IEEE Radiation Effects Data Workshop (REDW)*, pages 1–6, 07 2015.

[19] L. Luza, D. Soderstrom, H. Puchner, R. Alía, M. Letiche, A. Bosio and L. Dilillo. Effects of Thermal Neutron Irradiation on a Self-Refresh DRAM, 2020.

[20] Microsemi. *IGLOO2 and SmartFusion2 65nm Commercial Flash FPGAs: Interim Summary of Radiation Test Results.* Microsemi, October 2014.

[21] Durwyn Dsilva, Jih-Jong Wang, Nadia Rezzak, and Narayan Jat. Neutron see testing of the 65nm smartfusion2 flash-based fpga. In *2015 IEEE Radiation Effects Data Workshop (REDW)*, pages 1–5, 2015.

[22] Viviane Pierrard, G. Rosson, Kris Borremans, Joseph Lemaire, Jeroen Maes, S. Bonnewijn, E. Ransbeeck, E. Neefs, M. Cyamukungu, Sylvie Benck, L. Bonnet, S. Borisov, J. Cabrera, Ghislain Gregoire, C. Semaille, G. Creve, J. Saedeleer, Bart Desoete, F. Preud'homme, and P. Nieminen. The energetic particle telescope: First results. *Space Science Reviews*, 184:87–106, 11 2014.

[23] Edemar Morsch Filho, Vicente Nicolau, Kleber Paiva, and Talita Possamai. A comprehensive attitude formulation with spin for numerical model of irradiance for cubesats and picosats. *Applied Thermal Engineering*, 168:114859, 12 2019.

[24] ISSI. *IS42S86400B, IS42S16320B, IS45S16320B - 64M x 8, 32M x 16, 512Mb SYNCHRONOUS DRAM, Rev. H.* Integrated Silicon Solution, Inc., December 2011. Accessed on: April 14, 2021.

[25] ISSI. *IS42/45R86400D/16320D/32160D IS42/45S86400D/16320D/32160D - 16Mx32, 32Mx16, 64Mx8 512Mb SDRAM, Rev. B.* Integrated Silicon Solution, Inc., May 2015. Accessed on: April 14, 2021.

[26] ISSI. *IS42R86400F/16320F, IS45R86400F/16320F, IS42S86400F/16320F, IS45S86400F/16320F - 32Mx16, 64Mx8 512Mb SDRAM, Rev. B1.* Integrated Silicon Solution, Inc., July 2017. Accessed on: April 14, 2021.

[27] André M. P. de Mattos. HARSH Payload repository, Acesso em: Agosto de 2021. Disponível em: <https://github.com/andrempmattos/harsh-payload>.

[28] C. Urbina-Ortega, G. Furano, G. Magistrati, K. Marinis, A. Menicucci, and D. Merodio-Codinachs. Flash-based fpgas in space, design guidelines and trade-off for critical applications. In *2013 14th European Conference on Radiation and Its Effects on Components and Systems (RADECS)*, pages 1–8, 2013.

[29] Christopher Wilson, James MacKinnon, Patrick Gauvin, Sebastian Sabogal, and Alan D. George. uCSP: A Diminutive, Hybrid, Space Processor for Smart Modules and CubeSats. In *30th Annual AIAA/USU Conference on Small Satellites*, 2016.

[30] GomSpace. Command and Data handling, Acesso em: Dezembro de 2019. Disponível em: <gomspace.com/shop/subsystems/command-and-data-handling/default.aspx>.

[31] ISIS. Command and Data handling, Acesso em: Dezembro de 2019. Disponível em: <www.isispace.nl/products/command-data-handling-systems/>.

[32] EnduroSat. OnBoard Computer (OBC), Acesso em: Dezembro de 2019. Disponível em: <www.endurosat.com/cubesat-store/cubesat-obc/onboard-computer-obc/>.

[33] PolySat. PolySat Research Laboratory, Acesso em: Dezembro de 2019. Disponível em: <www.polysat.org/>.

[34] Carlos E. Gonzalez, Camilo J. Rojas, Alexandre Bergel, and Marcos A. Diaz. An architecture-tracking approach to evaluate a modular and extensible flight software for CubeSat nanosatellites. *IEEE Access*, 7:126409–126429, 2019.

[35] Said Hamdioui. Test for single-port and two-port SRAMs. In *Testing Static Random Access Memories*, volume 26 of *Frontiers in Electronic Testing*, chapter 6, pages 106–107. Springer, Boston, MA, US, 1 edition, 2004. DOI: 10.1007/978-1-4757-6706-3.

[36] Alberto Bosio, Luigi Dilillo, Patrick Girard, Serge Pravossoudovitch, and Arnaud Virazel. Basics on SRAM testing. In *Advanced Test Methods for SRAMs*, chapter 1, pages 1–19. Springer, Boston, MA, 2010. DOI: 10.1007/978-1-4419-0938-1.

[37] D. Niggemeyer, M. Redeker, and J. Otterstedt. Integration of non-classical faults in standard March tests. In *Proceedings. International Workshop on Memory Technology, Design and Testing (Cat. No.98TB100236)*, pages 91–96, San Jose, CA, USA, Aug. 1998. DOI: 10.1109/MTDT.1998.705953.

[38] Carlo Cazzaniga and Christopher D. Frost. Progress of the Scientific Commissioning of a fast neutron beamline for Chip Irradiation. *Journal of Physics: Conference Series*, 1021:012037, May. 2018. DOI: 10.1088/1742-6596/1021/1/012037.

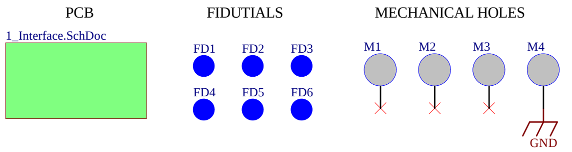
[39] Carlo Cazzaniga, Marta Bagatin, Simone Gerardin, Alessandra Costantino, and Christopher David Frost. First tests of a new facility for device-level, board-level and system-level neutron irradiation of microelectronics. *IEEE Transactions on Emerging Topics in Computing*, Nov. 2018. DOI: 10.1109/TETC.2018.2879027.

[40] Lucas Matana Luza, Daniel Söderström, André Martins Pio de Mattos, Eduardo Augusto Bezerra, Carlo Cazzaniga, Maria Kastriotou, Christian Poivey, and Luigi Dilillo. Technology impact on neutron-induced effects in sdrams: A comparative study. In *2021 16th International Conference on Design Technology of Integrated Systems in Nanoscale Era (DTIS)*, pages 1–6, 2021.

A Payload hardware schematics

Rev	Description	Date	Author
1.0	Initial release.	30-Apr-2020	Andre M. P. Mattos
1.1	Review updates.	28-May-2020	Andre M. P. Mattos

Revision History



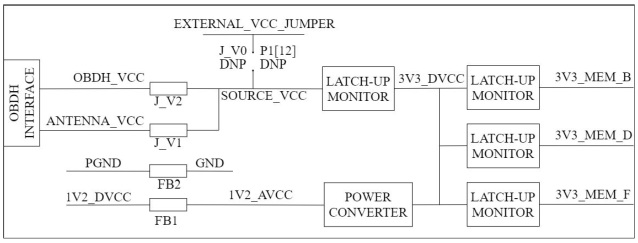
PCB Elements

HARSH Payload Daughter Board
(HARSH_Payload) for OBDH 2.0 Hardware

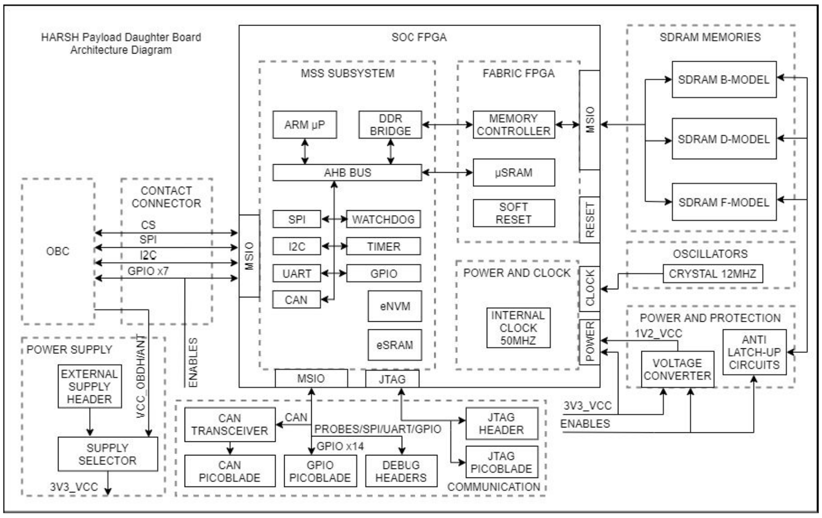
Copyright © 2020
by SpaceLab and LIRMM.

- Drawn by: André Martins Pio de Mattos.
- Based on OBDH 2.0 designed by: André Martins Pio de Mattos.
- Based on DB_Memory designed by: Yan Castro de Azeredo.
- Support: Kleber Reis Gouveia Júnior.

Project Information



Power Diagram

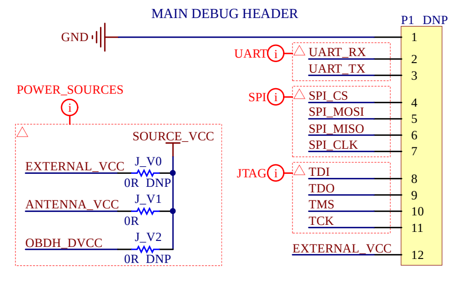


Block Diagram

Title: ***HARSH Payload Achitecture***
Size: A4 | Project: HARSH_Payload.PrjPCB | Revision: 1.1
Date: 6/1/2020 | Time: 3:23:15 PM | Sheet 1 of 8
Drawn By: Andre Martins Pio de Mattos | Model: HARSH_DB

LIRMM - SPACERADGroup
UFSC - SpaceLab

Institutional partnership project:
research in memory radiation hardness



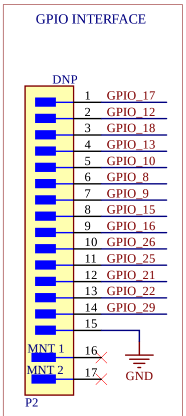
FLIGHT MODEL VERSION: (P1 header not placed)

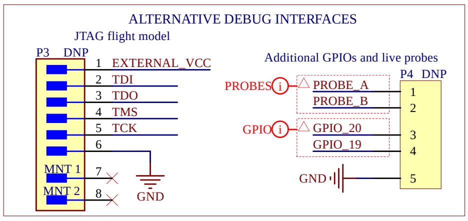
- Power supply: from FloripaSat EPS Antenna regulator place J_V1 or from FSat-II EPS OBDH regulator place J_V2 (ONLY one J_Vx placed)
- Programming: use P3 picoblade connector

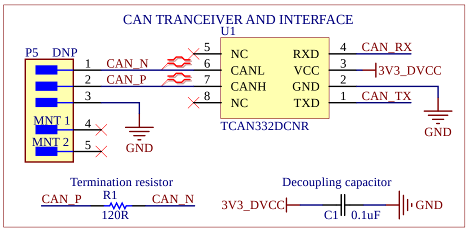
OTHER VERSIONS:

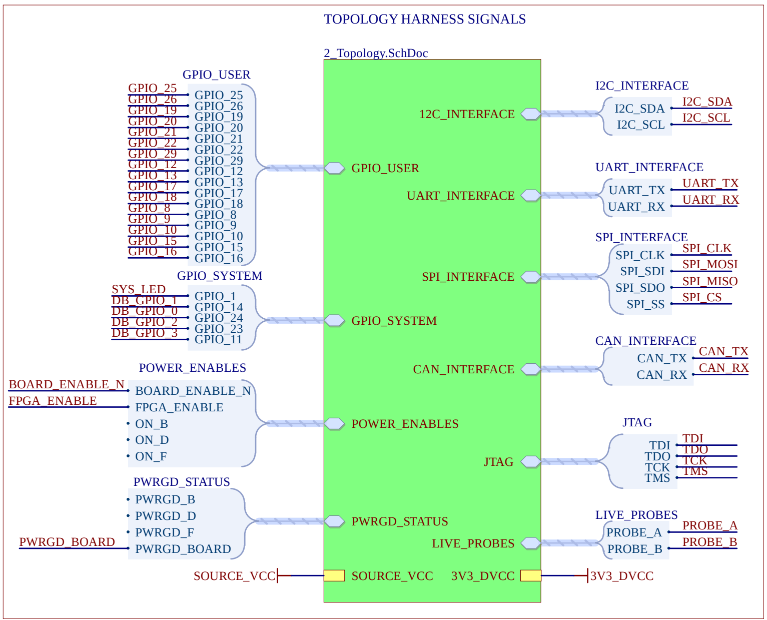
- Power supply: from FloripaSat EPS Antenna regulator place J_V1, from FSat-II EPS OBDH regulator place J_V2 or from external supply place J_V0 (ONLY one J_Vx placed or connected)
- Programming: use P1 header pins (8, 9, 10 and 11)

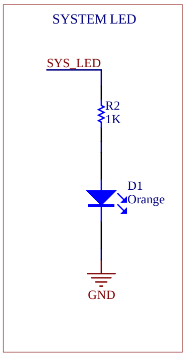
OBSERVATIONS:
- JTAG programming requires power supply from external supply or OBDH sources

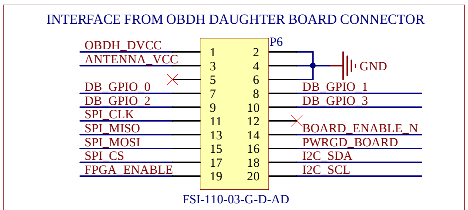












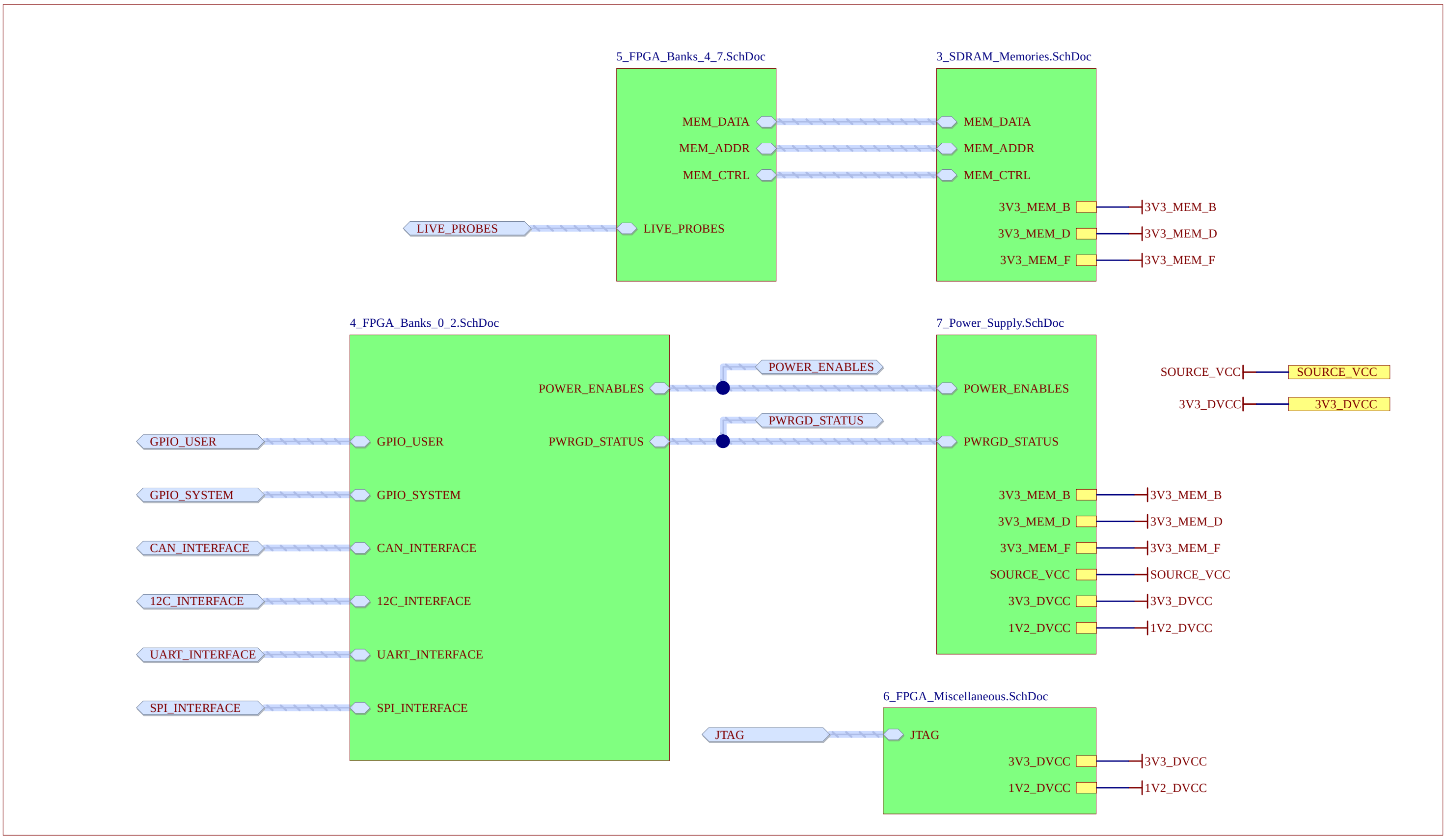
Features not used from the generic daughter board interface connector:
-BATTERY VCC
Features used differently, but compatible, in the generic daughter board interface connector:
- DB_CS2 as GPIO for FPGA enable (Active High)
- DB_ADC0 as GPIO for board "power good" status (feedback of payload power status)
- DB_ADC1 as GPIO for board enable (Active Low)

According to SAMTEC product specification, the maximum body height of the connector when pressed should be 3.28 mm.

Title: ***HARSH Payload Interface***
Size: A4 | Project: HARSH_Payload.PrjPCB | Revision: 1.1
Date: 6/1/2020 | Time: 3:23:15 PM | Sheet 2 of 8
Drawn By: Andre Martins Pio de Mattos | Model: HARSH_DB

LIRMM - SPACERADGroup
UFSC - SpaceLab

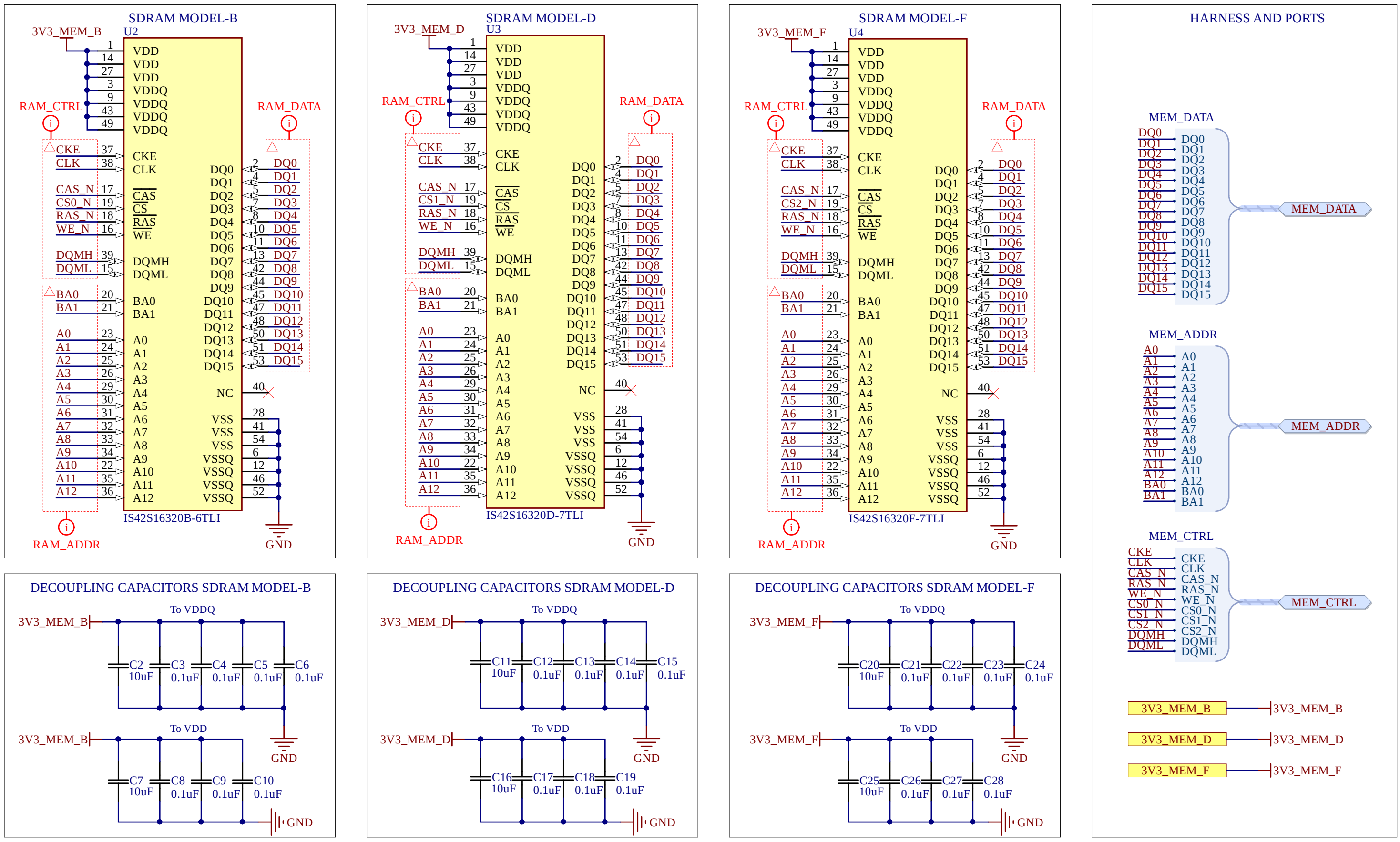
Institutional partnership project:
research in memory radiation hardness



Title: ***HARSH Payload Topology***
Size: A4 Project: HARSH_Payload.PrjPCB Revision: 1.1
Date: 6/1/2020 Time: 3:23:16 PM Sheet 3 of 8
Drawn By: Andre Martins Pio de Mattos Model: HARSH_DB

LIRMM - SPACERADGroup
UFSC - SpaceLab

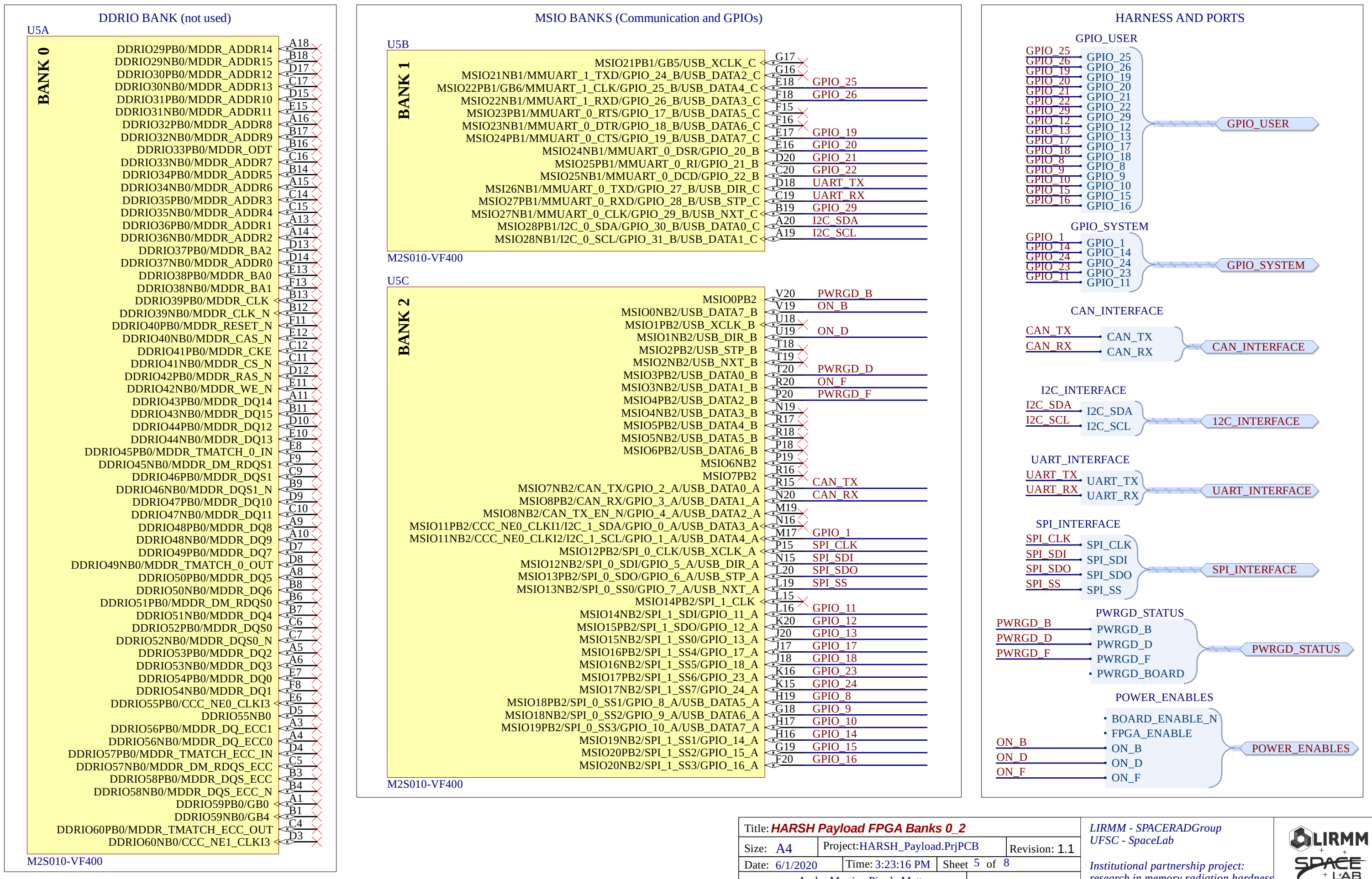
Institutional partnership project:
research in memory radiation hardness



Title: ***HARSH Payload SDRAM memories***
Size: A4 Project: HARSH_Payload.PrjPCB Revision: 1.1
Date: 6/1/2020 Time: 3:23:16 PM Sheet 4 of 8
Drawn By: Andre Martins Pio de Mattos Model: HARSH_DB

LIRMM - SPACERADGroup
UFSC - SpaceLab

Institutional partnership project:
research in memory radiation hardness


DDRIO BANK (not used)
U5A
BANK 0
M2S010-VF400
MSIO BANKS (Communication and GPIOs)
U5B
BANK 1
MSIO21PB1/GB5/USB_XCLK_C G17
MSIO21NB1/MMUART_1_TXD/GPIO_24_B/USB_DATA1_C G16
MSIO22PB1/GB6/MMUART_1_CLK/GPIO_25_B/USB_DATA4_C E18 GPIO_25
MSIO22NB1/MMUART_1_RXD/GPIO_26_B/USB_DATA3_C F18 GPIO_26
MSIO23PB1/MMUART_0_RTS/GPIO_17_B/USB_DATA5_C F15
MSIO23NB1/MMUART_0_DTR/GPIO_18_B/USB_DATA6_C F16
MSIO24PB1/MMUART_0_CTS/GPIO_19_B/USB_DATA7_C E17 GPIO_19
MSIO24NB1/MMUART_0_DSR/GPIO_20_B E16 GPIO_20
MSIO25PB1/MMUART_0_RI/GPIO_21_B D20 GPIO_21
MSIO25NB1/MMUART_0_DCD/GPIO_22_B C20 GPIO_22
MSI26NB1/MMUART_0_TXD/GPIO_27_B/USB_DIR_C D18 UART_TX
MSIO27PB1/MMUART_0_RXD/GPIO_28_B/USB_STP_C C19 UART_RX
MSIO27NB1/MMUART_0_CLK/GPIO_29_B/USB_NXT_C B19 GPIO_29
MSIO28PB1/I2C_0_SDA/GPIO_30_B/USB_DATA0_C A20 I2C_SDA
MSIO28NB1/I2C_0_SCL/GPIO_31_B/USB_DATA1_C A19 I2C_SCL
U5C
BANK 2
MSIO0PB2 V20 PWRGD_B
MSIO0NB2/USB_DATA7_B V19 ON_B
MSIO1PB2/USB_XCLK_B U18
MSIO1NB2/USB_DIR_B U19 ON_D
MSIO2PB2/USB_STP_B T18
MSIO2NB2/USB_NXT_B T19
MSIO3PB2/USB_DATA0_B T20 PWRGD_D
MSIO3NB2/USB_DATA1_B R20 ON_F
MSIO4PB2/USB_DATA2_B P20 PWRGD_F
MSIO7NB2/CAN_TX/GPIO_2_A/USB_DATA0_A R15 CAN_TX
MSIO8PB2/CAN_RX/GPIO_3_A/USB_DATA1_A N20 CAN_RX
MSIO11NB2/CCC_NE0_CLKI2/I2C_1_SCL/GPIO_1_A/USB_DATA4_A M17 GPIO_1
MSIO12PB2/SPI_0_CLK/USB_XCLK_A P15 SPI_CLK
MSIO12NB2/SPI_0_SDI/GPIO_5_A/USB_DIR_A N15 SPI_SDI
MSIO13PB2/SPI_0_SDO/GPIO_6_A/USB_STP_A L20 SPI_SDO
MSIO13NB2/SPI_0_SS0/GPIO_7_A/USB_NXT_A L19 SPI_SS
MSIO14NB2/SPI_1_SDI/GPIO_11_A L16 GPIO_11
MSIO15PB2/SPI_1_SDO/GPIO_12_A K20 GPIO_12
MSIO15NB2/SPI_1_SS0/GPIO_13_A J20 GPIO_13
MSIO16PB2/SPI_1_SS4/GPIO_17_A J17 GPIO_17
MSIO16NB2/SPI_1_SS5/GPIO_18_A J18 GPIO_18
MSIO17PB2/SPI_1_SS6/GPIO_23_A K16 GPIO_23
MSIO17NB2/SPI_1_SS7/GPIO_24_A K15 GPIO_24
MSIO18PB2/SPI_0_SS1/GPIO_8_A/USB_DATA5_A H19 GPIO_8
MSIO18NB2/SPI_0_SS2/GPIO_9_A/USB_DATA6_A G18 GPIO_9
MSIO19PB2/SPI_0_SS3/GPIO_10_A/USB_DATA7_A H17 GPIO_10
MSIO19NB2/SPI_1_SS1/GPIO_14_A H16 GPIO_14
MSIO20PB2/SPI_1_SS2/GPIO_15_A G19 GPIO_15
MSIO20NB2/SPI_1_SS3/GPIO_16_A F20 GPIO_16
HARNESS AND PORTS
GPIO_USER
GPIO_SYSTEM
CAN_INTERFACE
I2C_INTERFACE
UART_INTERFACE
SPI_INTERFACE
PWRGD_STATUS
POWER_ENABLES
BOARD_ENABLE_N
FPGA_ENABLE
ON_B
ON_D
ON_F

Title: ***HARSH Payload FPGA Banks 0_2***

Size: A4 | Project: HARSH_Payload.PrjPCB | Revision: 1.1

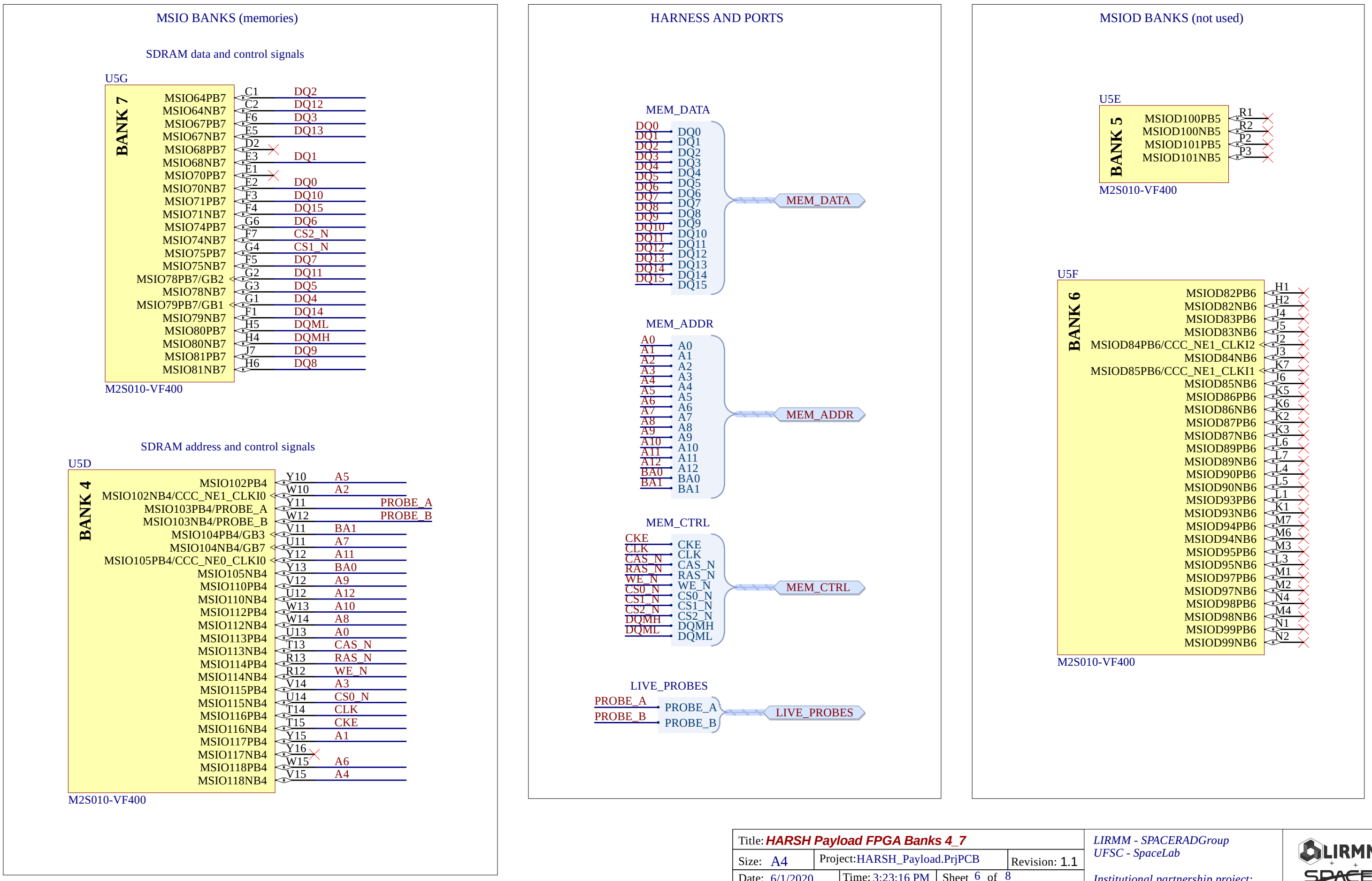
Date: 6/1/2020 | Time: 3:23:16 PM | Sheet 5 of 8

Drawn By: Andre Martins Pio de Mattos | Model: HARSH_DB

LIRMM - SPACERADGroup
UFSC - SpaceLab

Institutional partnership project: research in memory radiation hardness

---



Title: ***HARSH Payload FPGA Banks 4_7***

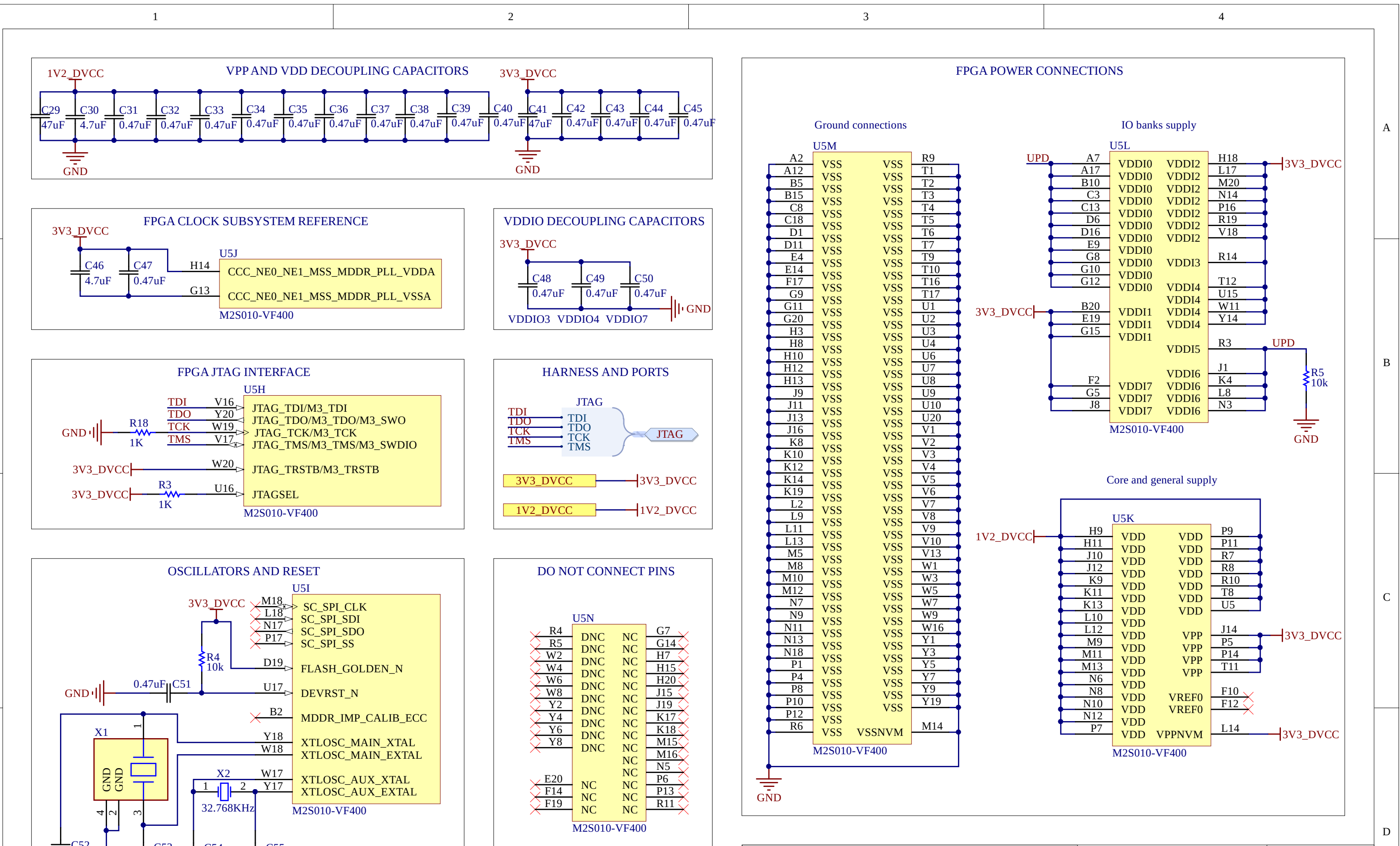
Size: A4 | Project: HARSH_Payload.PrjPCB | Revision: 1.1

Date: 6/1/2020 | Time: 3:23:16 PM | Sheet 6 of 8

Drawn By: Andre Martins Pio de Mattos | Model: HARSH_DB

LIRMM - SPACERADGroup
UFSC - SpaceLab

Institutional partnership project: research in memory radiation hardness

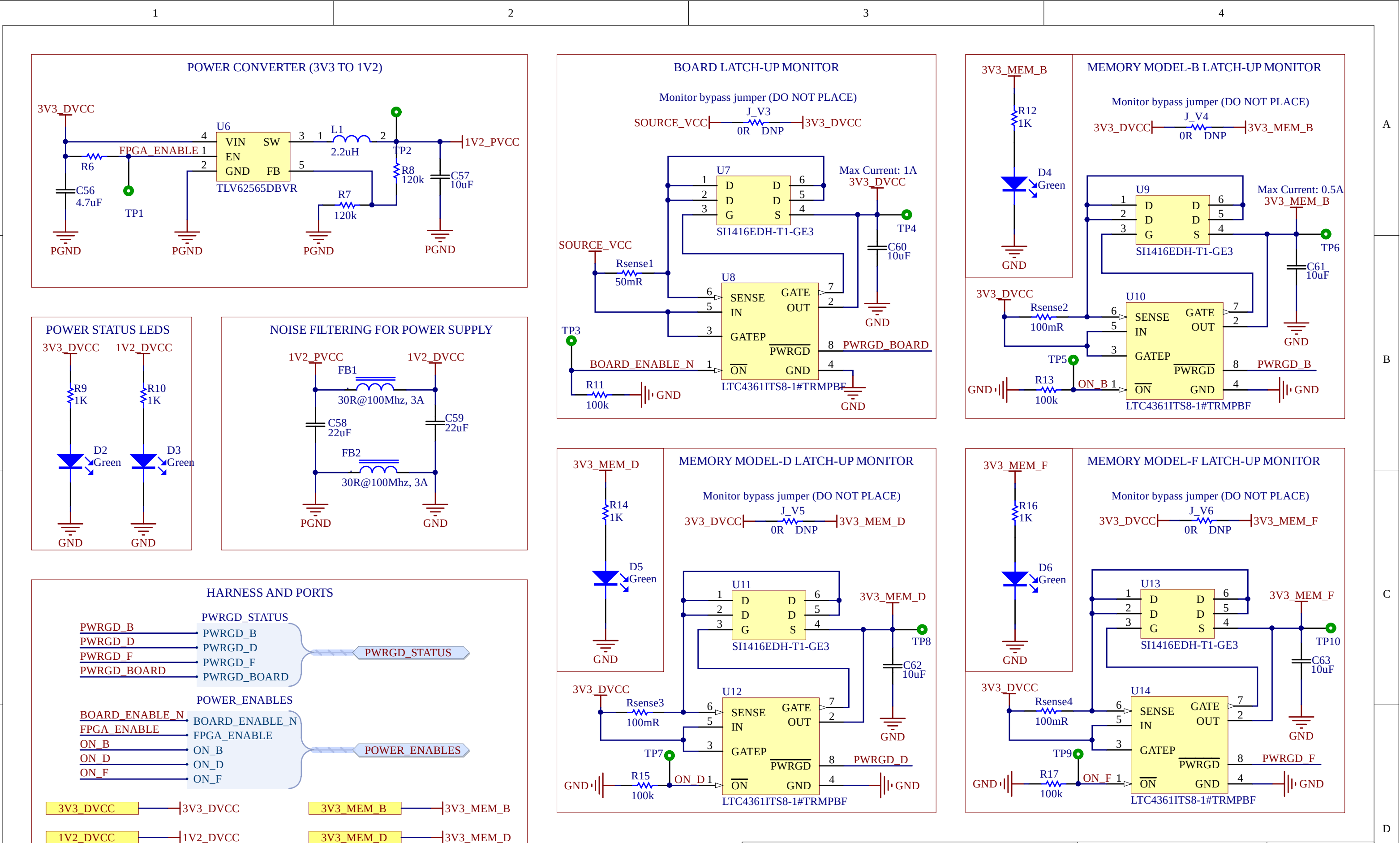


Title: ***HARSH Payload FPGA Miscellaneous***
Size: A4 Project: HARSH_Payload.PrjPCB Revision: 1.1
Date: 6/1/2020 Time: 3:23:17 PM Sheet 7 of 8
Drawn By: Andre Martins Pio de Mattos Model: HARSH_DB

LIRMM - SPACERADGroup
UFSC - SpaceLab

Institutional partnership project:
research in memory radiation hardness

LIRMM SPACE LAB

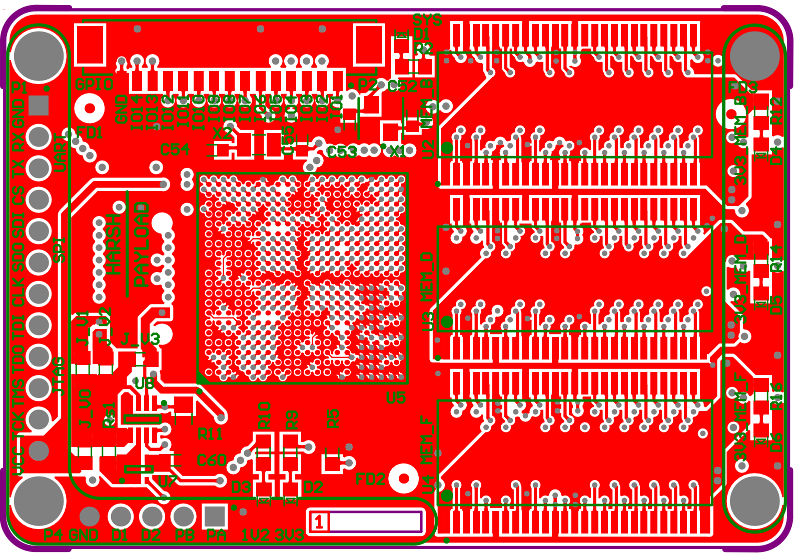


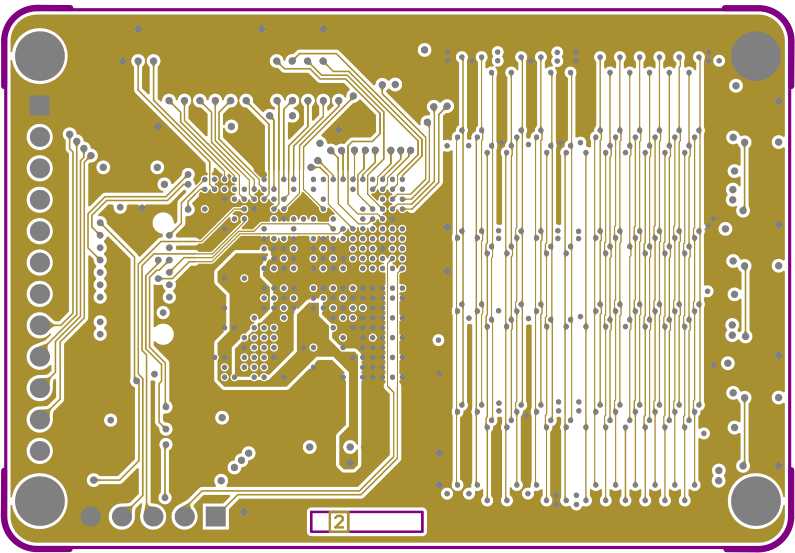
Title: ***HARSH Payload Power Supply***
Size: A4 Project: HARSH_Payload.PrjPCB Revision: 1.1
Date: 6/1/2020 Time: 3:23:17 PM Sheet 8 of 8
Drawn By: Andre Martins Pio de Mattos Model: HARSH_DB

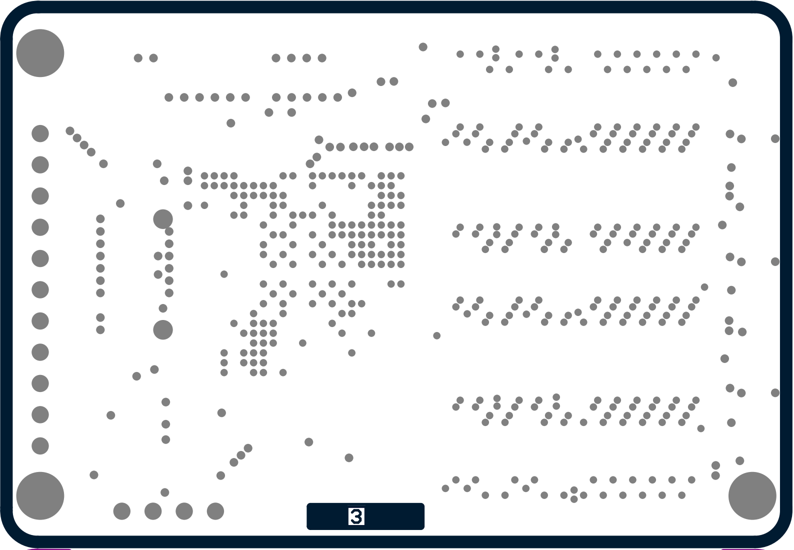
LIRMM - SPACERADGroup
UFSC - SpaceLab

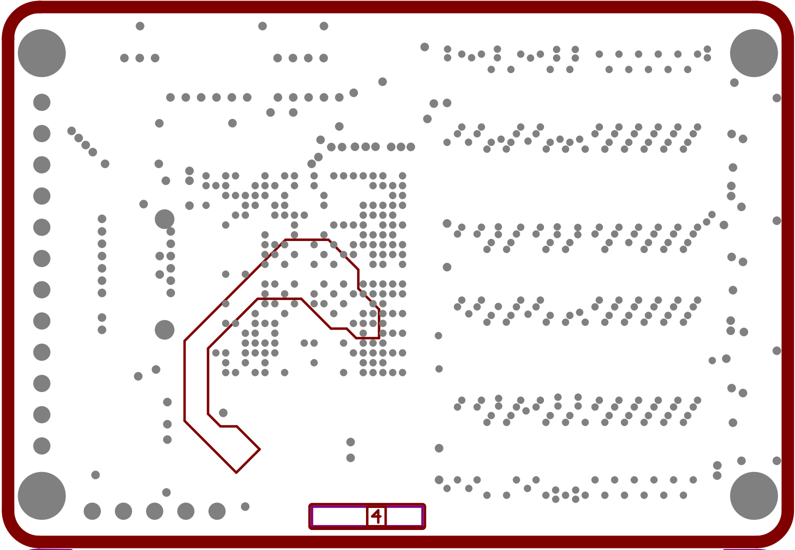
Institutional partnership project:
research in memory radiation hardness

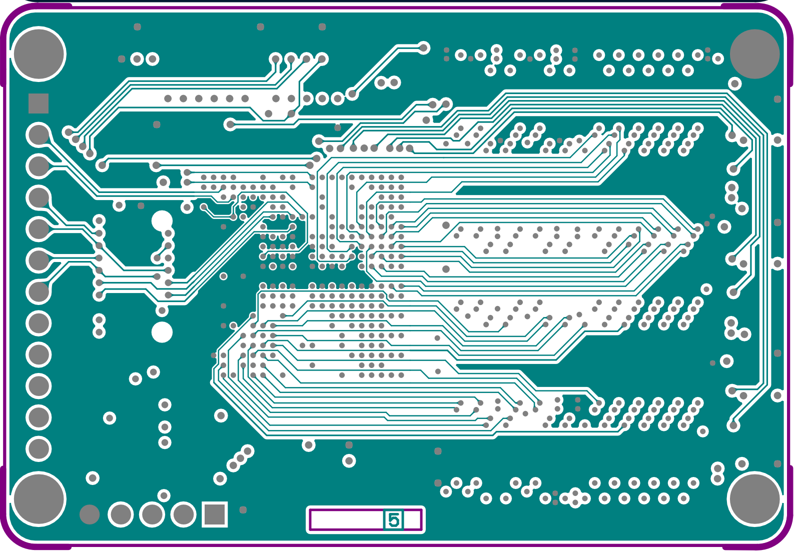
LIRMM SPACE LAB

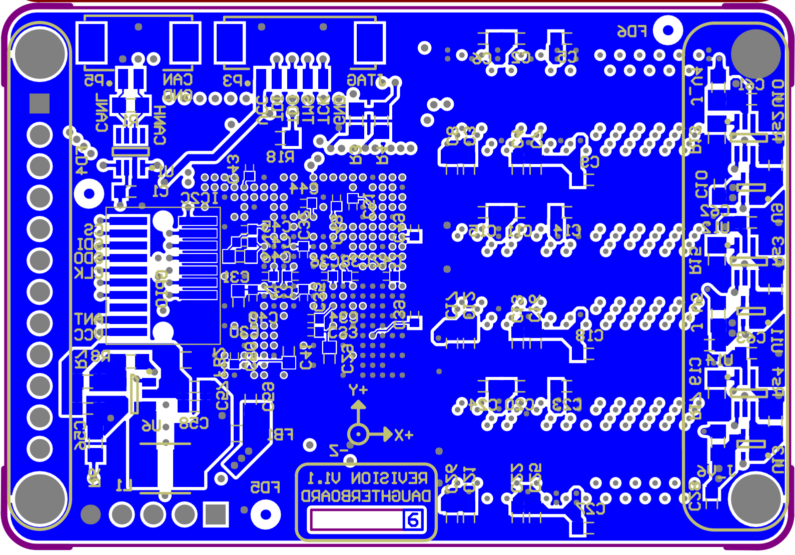

1
GPIO
GND IO14 IO13 IO12 IO11 IO10 IO9 IO8 IO7 IO6 IO5 IO4 IO3 IO2 IO1
P1 GND RX TX CS SDI SDO CLK TDI TDO TMS TCK VCC
UART
SPI
JTAG
HARSH PAYLOAD
FD1
FD2
FD3
P2
C52
C53
C54
C55
C60
X1
X2
SYS D1 R2
MEM_B
MEM_D
MEM_F
3V3_MEM_B
3V3_MEM_D
3V3_MEM_F
R12 R14 R15
D4 D5 D6
U2 U3 U4 U5 U7 U8
J_V0 J_V1 J_V2 J_V3
Rs1
R5 R9 R10 R11
D2 D3
P4 GND D1 D2 PB PA 1V2 3V3


2


3


4


5


6
REVISION V1.1
DAUGHTERBOARD
P5
CAN GND
P3
JTAG
CANL
CANH
R1
U1
C1
FD4
FD5
FD6
I2C
CS SDI SDO CLK
ANT VCC
U10
U6
L1
FB1
C56
C58
C59
R8
Y+
+X
-Z
C2 C5 C6
C8 C9 C15 C11 C14
C18 C20 C23 C24 C27
C10 C61 C63
C19
R18 R19 R23 R4