|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **THE UNIVERSITY OF SHEFFIELD School of Electrical and Electronic Engineering**  **3rd Year Individual Project — Project Initialisation Document** | | | |  |
| **Student Name** | | Amaan Mujawar | | | |
| **Project Title** | | Implement an Arithmetic Unit utilising Approximate Computing into RISC-V SoC | | | |
| **Supervisor** | | Mr Neil Powell | **Second Marker** |  | |

**Project motivation:**

The rapid increase in data generated by applications such as image processing, video streaming, and machine learning has created a need for efficient computing solutions. Traditional computing methods often lead to high energy consumption and prolonged processing times, making them inadequate for modern data-intensive tasks.

This project aims to develop an Arithmetic Unit that utilises Approximate Computing, which allows for less precise calculations in scenarios where perfect accuracy is not critical. This approach can significantly enhance processing speed and reduce energy usage, making it particularly suitable for applications that can tolerate minor inaccuracies.

The motivation for this work to integrate Approximate Computing into a Reduced Instruction Set Computer (RISC-V) System on Chip (SoC). By focusing on these objectives, we aim to improve overall system efficiency, enabling better handling of large data volumes while minimising resource consumption [1].

**Project Specification:**

**Scope:**

1. **Functional Requirements:**
   * Design a 32-bit MAC unit that employs Approximate Computing techniques.
   * Ensure the MAC unit integrates seamlessly with the RISC-V Processor architecture.
   * Achieve specified performance metrics:
     + Reduced power consumption
     + Increased processing speed
     + Acceptable levels of accuracy
     + Optimized chip area
2. **Non-Functional Requirements:**
   * Utilize the Xilinx Vivado Design Suite for design and simulation.
   * Ensure the design is scalable.
   * Maintain high standards of documentation throughout the project lifecycle.

**Deliverables:**

* **Design Documentation:**
  + Architectural diagrams and schematic designs for the MAC unit.
* **Hardware Implementation:**
  + HDL code for the MAC unit.
  + Synthesis and simulation results from Vivado.
* **Testing Documentation:**
  + Comprehensive test benches for validating performance metrics.
  + Analysis of results evaluating power consumption, speed, accuracy, and chip area.
* **Final Report:**
  + Summary of findings, insights on Approximate Computing techniques, and recommendations for future work.

**Constraints:**

* The project must be completed within the allocated timeframe and resources.
* Compliance with RISC-V specifications and standards.

**Evaluation Metrics:**

* Performance will be evaluated based on:
  + Power and area consumption measurements
  + Processing speed benchmarks
  + Accuracy of results

**Project Schedule:**

The objective of this project is to design and implement an Arithmetic Unit using Approximate Computing techniques for integration into a RISC-V SoC, utilizing the Artix Nexus 7 FPGA and the Xilinx Vivado Design Suite.

The project will begin with a literature review and method selection, where various Approximate Computing techniques will be researched, documenting their pros and cons while determining allowable tolerances. This will culminate in selecting the most suitable methods and developing an initial hardware accelerator plan, achieving **Milestone 1**.

Next, in the design phase, a detailed block diagram of the 32-bit Multiply-Accumulate (MAC) unit will be created, followed by calculations of expected performance metrics. Peer reviews will be conducted to refine the design, which will then be implemented using Hardware Description Languages (Verilog/VHDL). Initial debugging and testing will ensure the integrity of the design, reaching **Milestone 2** upon completion.

The subsequent phase will involve simulation and verification, where test benches will be developed in Vivado to conduct extensive simulations. Functionality and performance will be assessed, leading to final debugging and documentation of performance metrics, thereby achieving **Milestone 3**.

Integration with the RISC-V architecture will follow, including participation in labs to understand implementation strategies. The Approximate MAC unit will be integrated into the RISC-V CPU, followed by debugging and initial testing, marking **Milestone 4** upon successful integration.

Finally, a thorough evaluation and analysis phase will create test cases to assess the final design, including error testing and performance calculations.

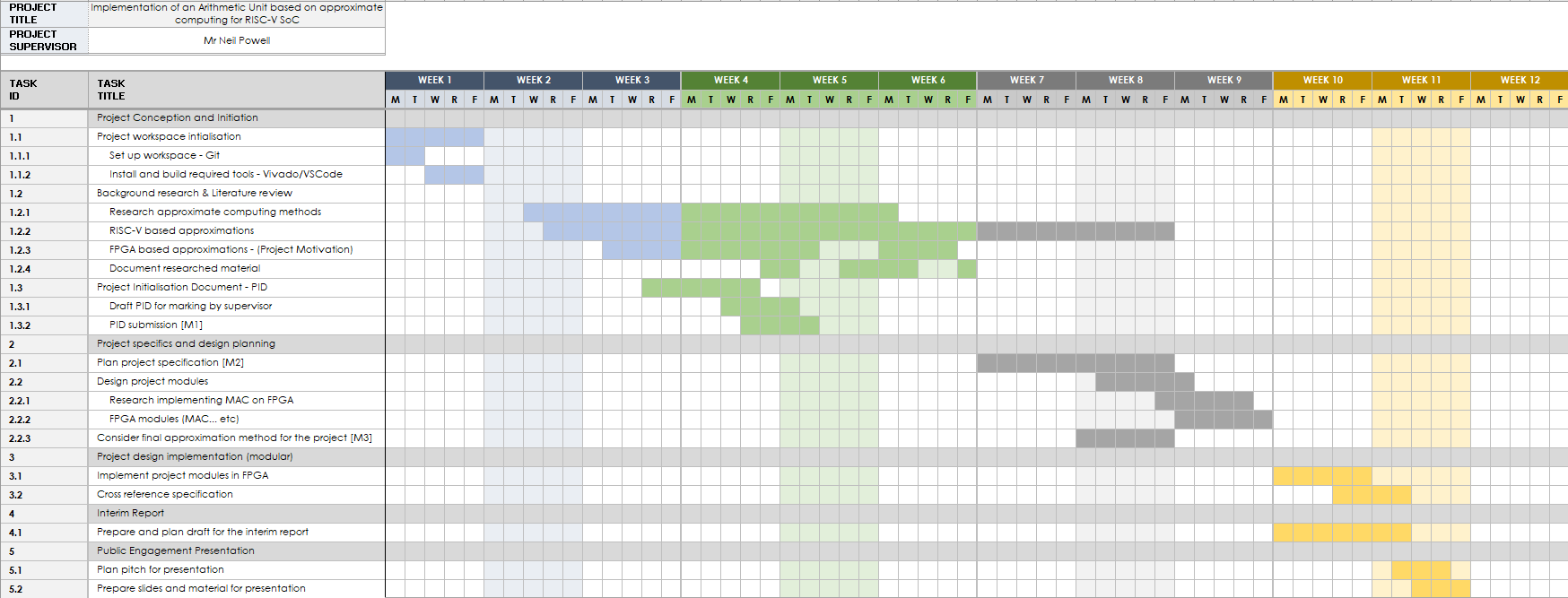
Throughout the project, comprehensive documentation will be maintained, including the introduction, literature review, and initial report, culminating in a finalised report that reflects the design and implementation process.

**Risk Register:**

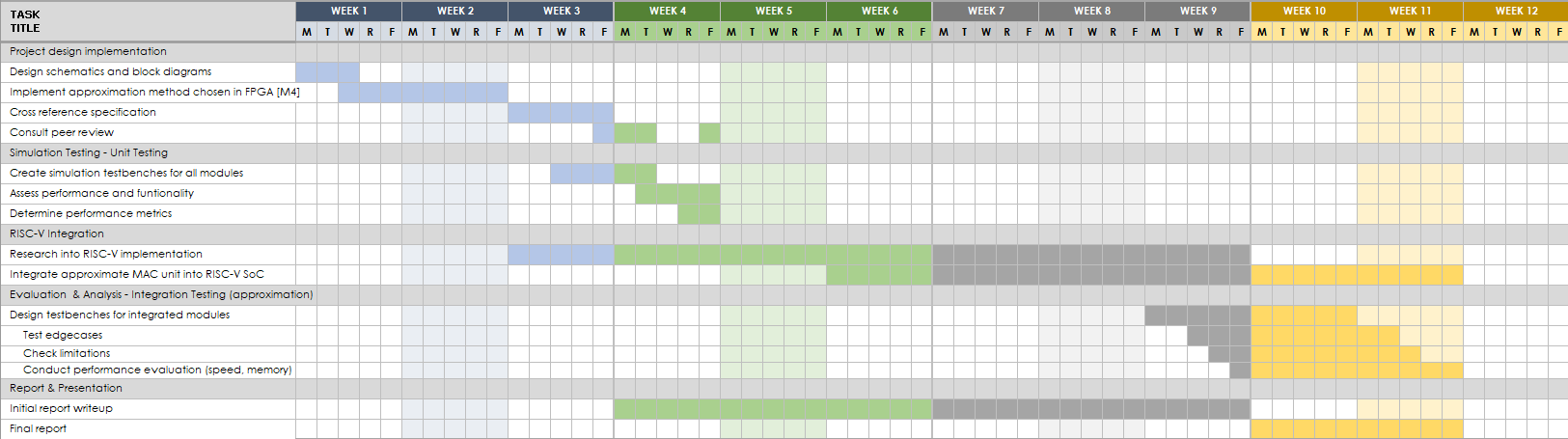
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Description of Risk** | **Risk evaluation (L/M/H)** | **Chance of risk (L/M/H)** | **Mitigation of Risk** |
| **1** | Loss of data (USB key) | M | L | Multiple back-ups, GitHub. |
| **2** | Delays in literature review affecting subsequent phases | M | M | |  | | --- | |  |   Implement peer reviews and consult supervisor regularly. |
| **3** | Incomplete or inaccurate design documentation | M | M | Consult with supervisor to ensure documentation is thorough and accurate. |
| **4** | Technical difficulties with FPGA programming | H | M | Seek support from resources and experienced colleagues. |
| **5** | Issues during testing leading to inaccurate results | H | M | Develop detailed test cases and conduct iterative testing. |
| **6** | Hardware failures (e.g., FPGA malfunctions) | H | L | Keep backup hardware available and conduct regular checks. |
| **7** | Resource availability (e.g., tools, software licenses) | M | L | Secure necessary tools and licenses early in the project. |
| **8** | Changes in project scope or requirements | M | M | Establish clear specifications. |
| **9** | Staff member availability (e.g., illness, other commitments) | M | M | Have tasks that can be carried on in parallel, so workflow is not interrupted. |
| **10** | Time management issues affecting project timeline | M | M | Conduct regular status meetings with supervisor. |

**Appendix**

**Gantt chart:**



*Figure 1: Gantt Chart for Semester 1*



*Figure 1: Gantt Chart for Semester 2*

M1 – Milestone 1 for producing a final process initiation document for submission

M2 – Milestone 2 for planning a detailed project specification

M3 – Milestone 3 for choosing an approximation method to be implemented for the project

M4 – Milestone 4 for implementing the chosen approximation method in FPGA

**References:**

[1] H. Nakahara and T. Sasao, "A deep convolutional neural network based on nested residue number system," 2015 25th International Conference on Field Programmable Logic and Applications (FPL), London, UK, 2015, pp. 1-6, doi: 10.1109/FPL.2015.7293933. keywords: {Table lookup;Field programmable gate arrays;Convolution;Kernel;Neural networks;Clocks;Dynamic range},

[2] S. Ullah, S. S. Murthy and A. Kumar, "SMApproxLib: Library of FPGA-based Approximate Multipliers," 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 2018, pp. 1-6, doi: 10.1109/DAC.2018.8465845. keywords: {Table lookup;Field programmable gate arrays;Libraries;Adders;Performance gain;Delays;Viterbi algorithm;Approximate Computing;Multipliers;Adders;FPGAs;Optimization;Area;Latency;Design Space Exploration},

[3] İ. Taştan, M. Karaca, and A. Yurdakul, “Approximate CPU Design for IoT End-Devices with Learning Capabilities,” *Electronics*, vol. 9, no. 1, p. 125, Jan. 2020, doi: https://doi.org/10.3390/electronics9010125.

‌