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

This section should introduce the subject area and explain its significance, providing a broad motivation for the project. Briefly outline the aims, which give a general overview of the project's purpose, followed by specific objectives. Objectives should detail actionable tasks that can be assessed at the end of the project (e.g., simulate…, test…, compare…). These objectives will also align with the work programme discussed later in the report.

This section should be a thorough examination of relevant theories and existing literature, supporting the project’s basis. Begin with an introduction to the theoretical framework underlying the project, followed by a critical appraisal of literature that is directly relevant. A strong literature review will demonstrate academic insight through structured groupings and logical connections between sources, providing motivation and context for the project’s work.

This section should outline the planned work for the remainder of the project. It will often link back to the project’s original objectives, detailing specific tasks to be completed. Include a reference to a detailed Gantt chart (provided as an appendix), which should represent the project timeline, updated progress, and adjustments against the initial plan

**Background, Aims and Objectives – (1 page)**

Approximate computing has become an emerging technique that reduces execution time, or power consumption by allowing a small degree of error in computations [1]. It has gained significant attention in digital circuit design, particularly as the demand for energy-efficient computing continues to rise. This approach is finding applications in areas such as machine learning, computer vision, web search, and data analysis. Many signal processing, image processing, and multimedia tasks are inherently error-tolerant and can produce results that appear indistinguishable to the human eye, even without the need for exact computations. By leveraging this error tolerance, approximate computing can be applied in such error-tolerant operations by providing meaningful results faster and/or with lower power consumption at the cost of reducing accuracy [2].

In today’s digital era, the demand for high-performance computing has grown exponentially, driven by data-intensive applications such as machine learning, big data analytics, computer vision, and multimedia processing. Historically, this demand has been met by advancements in semiconductor technology, guided by Moore’s Law, which predicts that the number of transistors on a chip doubles approximately every two years. This trend has enabled continuous improvements in computational power, energy efficiency and cost.

However, as transistor sizes approach their physical and economic limits, the rate of progress predicted by Moore’s Law is slowing. With traditional scaling facing significant challenges, it is becoming increasingly difficult to achieve the necessary performance and energy efficiency gains through conventional means alone. This has created a need for new approaches that can complement traditional scaling.

One such approach is approximate computing, as discussed above. It offers a model shift in how computations are performed. In contrast to focusing solely on precision, approximate computing introduces a controlled level of errors in computation to reduce execution time and/or power consumption. This technique is particularly well-suited for applications in signal processing, image processing, and multimedia where exact results are often unnecessary, and error-tolerant outputs can be perceived as correct by humans.

By manipulating this inherent error tolerance, approximate computing provides an opportunity to overcome the limitations of traditional scaling and continue to deliver improvements in computational efficiency. This project aims to explore the potential of approximate computing in digital circuit design by integrating existing methods of approximate computing to provide a hardware solution that balances performance, power consumption, and accuracy. As the limitations of Moore’s Law become more apparent, the importance of innovating techniques like approximate computing continues to grow. By addressing these challenges, this project aims to contribute to the development of sustainable, high-performance digital systems that are capable to meet demands of future technologies.

The primary aim of this project is to design and implement a 32-bit Multiply-Accumulate (MAC) unit utilising approximate computing techniques, specifically tailored for integration into a RISC-V System on Chip (SoC). To achieve this, the project will follow a series of specific objectives. The first objective involves conducting a comprehensive literature review to identify suitable approximate computing techniques, followed by the selection of the most appropriate methods based on performance and characteristics. A detailed design of the MAC will be developed, incorporating the chosen approximate computing methods. The third objective is to implement the MAC unit using Hardware Description Language (Verilog), and to perform initial testing through FPGA simulations to verify functionality. Once the MAC unit has been tested in isolation it will be integrated into a RISC-V SoC, ensuring seamless compatibility and functionality within the broader system architecture. After integration, the final objective is to conduct thorough testing, comparing the performance, power consumption and accuracy of the new design. These tests will focus on evaluating how the approximations affect the system’s overall performance. Each of these objectives will be simulated, tested, and compared against current industry standards to assess the success of the project in achieving its goals.

**Theory and Literature Review – (4 pages)**

This section should be a thorough examination of relevant theories and existing literature, supporting the project’s basis. Begin with an introduction to the theoretical framework underlying the project, followed by a critical appraisal of literature that is directly relevant. A strong literature review will demonstrate academic insight through structured groupings and logical connections between sources, providing motivation and context for the project’s work.

**Technical Progress to Date – (2 pages)**

In this section, detail any initial work done, including any preliminary simulations, experiments, or findings. Discuss the methodologies and approaches taken so far, and any insights that have emerged. Provide a summary of the progress and the technical milestones reached to date.

**Project Management – (1 page)**

The primary objective of this project is to design and implement an Arithmetic Unit using Approximate Computing techniques, specifically a 32-bit Multiply-Accumulate (MAC) unit, for integration into a RISC-V System on Chip (SoC). This will involve utilizing the Artix Nexus 7 FPGA and the Xilinx Vivado Design Suite for design, simulation, and testing. The planned work is organized into several phases, each corresponding to specific tasks and milestones, which will be carried out according to the updated project timeline outlined in the accompanying Gantt chart, Figure 1 and Figure 2.

The project will begin with an extensive literature review and method selection phase, where various Approximate Computing techniques will be researched and evaluated. The pros and cons of each technique will be documented, and the allowable error tolerances will be determined for each method. This review will culminate in the selection of the most suitable Approximate Computing methods, followed by the development of an initial hardware accelerator plan. This phase will conclude with the completion of **Milestone 1**.

Following the literature review, the next phase will focus on the design of the 32-bit MAC unit. A detailed block diagram will be developed, taking into account the selected Approximate Computing techniques and the target performance metrics, including power consumption, speed, and accuracy. The expected performance of the design will be calculated, and peer reviews will be conducted to refine the design. Once the design is finalized, the implementation will proceed using Hardware Description Languages (Verilog/VHDL), followed by initial debugging and testing. This phase will mark the completion of **Milestone 2**.

In the subsequent phase, simulation and verification will take place. Test benches will be developed using the Xilinx Vivado Design Suite to conduct a series of simulations. These simulations will evaluate the functionality and performance of the MAC unit, including power consumption and accuracy, under the conditions specified by the design. The design will undergo further debugging and refinement based on simulation results, with the final performance metrics documented. This will lead to the completion of **Milestone 3**.

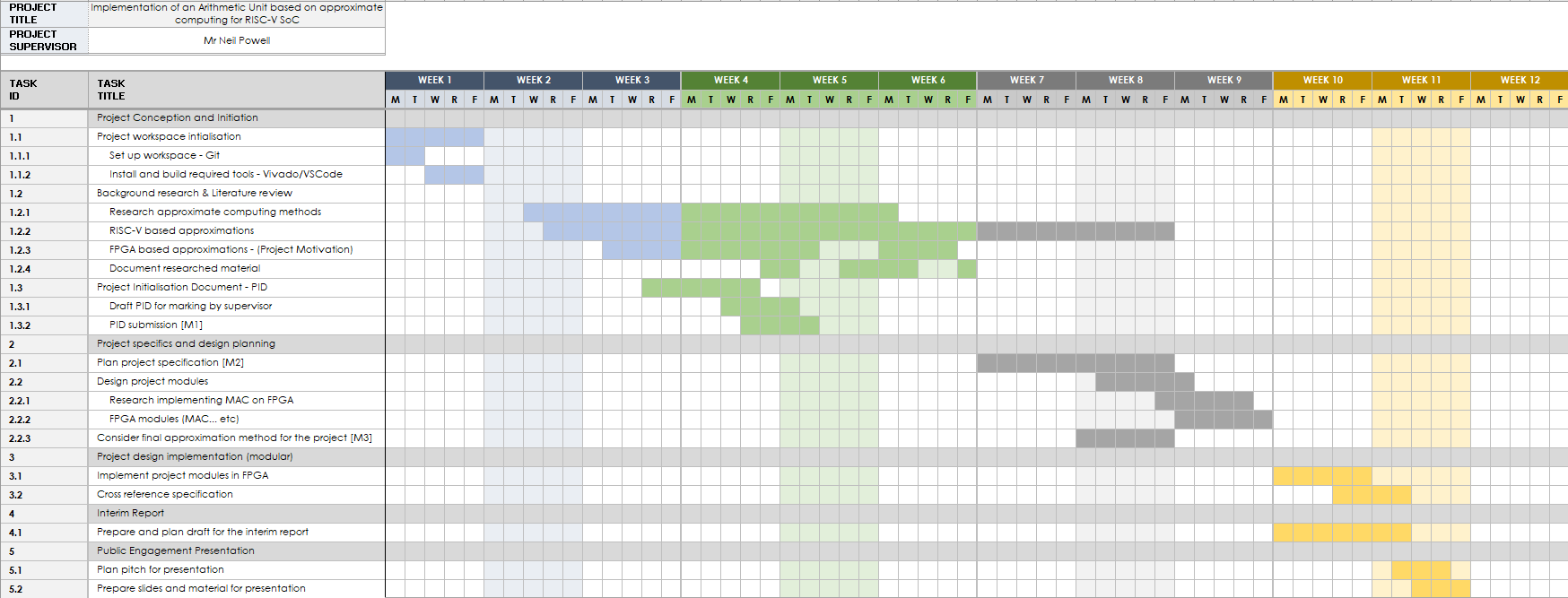
The next major phase will involve integrating the MAC unit into the RISC-V architecture. This phase will include participation in relevant labs to better understand the implementation strategies for embedding the MAC unit into the RISC-V processor. Once integrated, the system will undergo debugging and initial testing to verify that the MAC unit functions correctly within the RISC-V SoC framework. Successful integration and testing will result in the completion of **Milestone 4**.

Finally, a thorough evaluation and analysis phase will be conducted to assess the final design. This will involve creating a series of test cases to evaluate power consumption, processing speed, and accuracy, as well as conducting error testing to evaluate the impact of Approximate Computing on the system’s output. The results will be compared to existing models to assess the advantages and trade-offs of using Approximate Computing techniques.

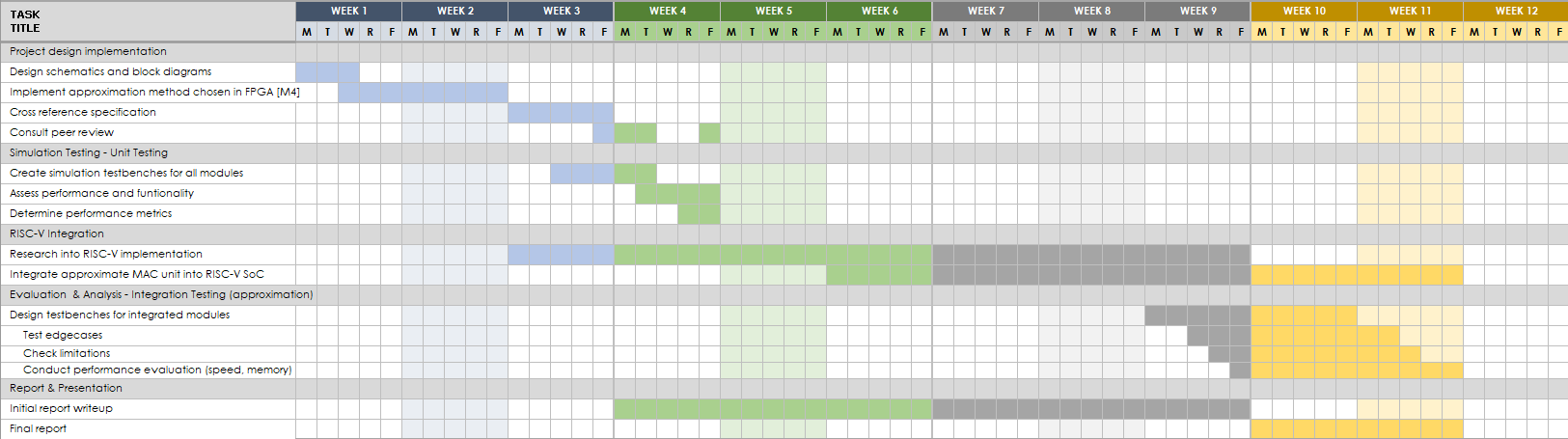
Throughout the entire project, comprehensive documentation will be maintained, detailing the design process, test results, and performance evaluations. This will culminate in the preparation of a final report, which will reflect the design and implementation process, key findings, and recommendations for future work in Approximate Computing.

A detailed Gantt chart outlining the project timeline, milestones, and progress will be included as an appendix to track the project's completion. The chart will serve as a tool to monitor progress and make necessary adjustments against the initial plan to ensure that the project remains on schedule.

**Appendix**



*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. Gupta and K. Suneja, "Hardware Design of Approximate Matrix Multiplier based on FPGA in Verilog," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2020, pp. 496-498, doi: 10.1109/ICICCS48265.2020.9121004. keywords: {Performance evaluation;Computational modeling;Approximate computing;Hardware;Software;Table lookup;Hardware design languages;approximate computing;matrix multiplier;Verilog;Field Programmable Array (FPGA);Look Up Tables (LUTs)},
2. Y. Guo, X. Chen, Q. Zhou and H. Sun, "Power-Efficient and Small-Area Approximate Multiplier Design with FPGA-Based Compressors," 2024 IEEE International Symposium on Circuits and Systems (ISCAS), Singapore, Singapore, 2024, pp. 1-5, doi: 10.1109/ISCAS58744.2024.10558590. keywords: {Image coding;Accuracy;Power demand;System performance;Circuits;Approximate computing;Hardware;Approximate computing;FPGA-based compressor;Low-power circuit},