Project Title

INTELLERA,
A Hardware based Accelerated Matrix MAC Processor.

Dated: Sep 30, 2023

Supervisor: Dr. Fahad Bin Muslim

Co-Supervisor: Dr. Taj Muhammad

Group Members: Saad Khan – Team Lead

Syed Zaeem Shakir

Mahnoor Maleeka

Revision History:

Revision History	*Date*	*Comments*
1.00		
2.00		

Document Approval:
The following document has been accepted and approved by the following:

Signature	***Date***	***Name***

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

1.1 PURPOSE

In the field of machine learning, matrix multiplication is a fundamental operation that plays a crucial role in various algorithms and computations. However, traditional processors often struggle to efficiently execute matrix multiplication tasks due to their general-purpose nature. This inefficiency results in increased time consumption and hampers the overall performance of machine learning applications. Therefore, there is a pressing need for a specialized processor design that incorporates matrix multiplication as a core instruction and aims to decrease time consumption in machine learning.

1.2 PRODUCT SCOPE

Intellera is an innovative project aimed at designing a RISC-V based processor with a customized instruction set architecture (ISA) that includes matrix multiplication instructions. The goal of Intellera is to address the performance bottleneck caused by conventional processors when executing matrix operations in machine learning algorithms. By enhancing the processor's capabilities and providing dedicated hardware acceleration for matrix Multiply-Accumulate (MAC) operations, Intellera aims to significantly reduce the time consumed in machine learning tasks.

Intellera proposes a novel approach to address the time consumption challenge in machine learning. The project focuses on designing a RISC-V based processor that features an extended instruction set architecture specifically tailored for matrix multiplication. By integrating dedicated hardware support for matrix MAC operations, Intellera can harness the parallel(pipelining) processing capabilities of the processor to accelerate matrix computations and reduce the overall execution time.

The project utilizes the knowledge of computer hardware and RISC-V Instruction Set for creating a customized instruction set and concept of systolic arrays implementation for accelerating the multiplication of matrices.

Table 1: Terms used in this document and their description.

Name	Description
RISC	Reduced Instruction Set Computer
FPGA	Field Programable Gate Array
ISA	Instruction Set Architecture
MAC	Multiplication Accumulation
ALU	Arithmetic Logic Unit

1.3 OVERVIEW

The project represents a cutting-edge venture in the realm of digital systems design and high-performance computing. This project combines the versatility of the RISC-V instruction set architecture (ISA) with the computational power of specialized Matrix Multiply-Accumulate (MAC) operations. The overarching goal is to create a flexible and high-performance computing platform that can efficiently handle a wide range of computational tasks, including those heavily reliant on matrix operations, such as machine learning, signal processing, and scientific computing.

1.3.1 Key Objectives

1. RISC-V Processor Integration: The project's foundation rests on the integration of a 32-bit RISC-V processor, adhering to the RV32I standard. This serves as the backbone for executing standard RISC-V instructions.

2. Custom ISA for Matrix Operations: A custom instruction set architecture (ISA) optimized for matrix operations is developed and integrated into the processor. This custom ISA includes Matrix MAC instructions, enabling precise and efficient matrix multiplication and addition.

3. High-Performance Computing: The processor is designed with a keen focus on performance. It aims to achieve high clock frequencies and low-latency execution to support real-time and computationally intensive tasks.

4. Hardware Pipelining: To improve instruction throughput, the processor employs a pipelining architecture with at least five stages, optimizing the execution of instructions.

5. FPGA Implementation: The project leverages Field-Programmable Gate Arrays (FPGAs) as the hardware platform for development and testing. FPGA compatibility allows for rapid prototyping and real-time testing of the processor's capabilities.

6. Software Toolchain Integration: The project integrates with software tools like Vivado for hardware description and synthesis, Venus for assembly code development, and DigitalJS for schematic design.

1.3.2 Expected Outcomes

The successful realization of this project will yield a versatile FPGA-based computing platform capable of executing both standard RISC-V instructions and custom Matrix MAC operations. This system's adaptability, high-performance characteristics, and FPGA compatibility position it as an ideal choice for various applications across domains, ranging from embedded systems to scientific research.

2. THE OVERALL DESCRIPTION

The project encompasses the development of a versatile RISC-V processor with the primary objective of accelerating matrix multiplication operations, a pivotal task within the domain of machine learning and scientific computing. The project's evolution began with the implementation of a single-cycle RISC-V processor using Vivado, establishing a solid foundation for subsequent enhancements. The immediate plan involves transitioning this processor into a highly efficient, pipelined architecture with five stages. This pipeline design aims to mitigate hazards and optimize the execution of RISC-V instructions, delivering improved throughput and performance. Beyond the processor core, a key innovation lies in the integration of a dedicated systolic array hardware architecture, meticulously designed to expedite matrix multiplication. This addition introduces a specialized matrix computation engine, effectively harnessing parallelism to dramatically reduce execution times for large-scale matrix operations.

2.1 PRODUCT PERSPECTIVE

In the realm of modern computing, the demand for high-performance processors tailored to specific computational tasks has intensified. The customized RISC-V processor represents an innovative response to this demand, offering a scalable and efficient solution for accelerating matrix multiplication operations. It stands at the intersection of hardware and software, where traditional RISC-V architecture is enhanced with a carefully crafted module inspired by systolic array for matrix operations. This project is a testament to the adaptability and extensibility of the RISC-V ISA, providing a dedicated solution for a critical computation within machine learning workflows. It complements existing processors by offering superior matrix computation capabilities, making it an invaluable tool for a wide range of applications, including deep learning, data analytics, and scientific simulations.

3. WORK BREAKDOWN STRUCTURE

1. **Project Initiation**
   - 1.1 Define Project Objectives and Scope
   - 1.2 Identify Key Stakeholders
   - 1.3 Develop Project Plan
   - 1.4 Obtain Necessary Resources
2. **Requirement Analysis and Design**
   - 2.1 Requirements Gathering
   - 2.2 Custom ISA Design
   - 2.3 Matrix MAC Instruction Specification
   - 2.4 Processor Architecture Design
   - 2.5 Memory Hierarchy Design
   - 2.6 Control Unit Design
   - 2.7 Hazard Unit Design
   - 2.8 ALU Design
   - 2.9 Register File Design
3. **Hardware Implementation**
   - 3.1 FPGA Selection and Setup
   - 3.2 HDL (Verilog) Coding for Processor
   - 3.3 Integration of Custom ISA
   - 3.4 Testing Individual Hardware Components
4. **Software Integration**
   - 4.1 Development of Custom Assembly Code for Matrix MAC
   - 4.2 Software-Hardware Integration
   - 4.3 Debugging and Testing Software-Hardware Interaction
5. **Performance Optimization**
   - 5.1 Clock Frequency Optimization
   - 5.2 Pipeline Optimization
   - 5.3 Memory Access Optimization
   - 5.4 Custom ISA Performance Enhancement
6. **Testing and Validation**
   - 6.1 Unit Testing of Hardware Components
   - 6.2 Integration Testing of the Processor
   - 6.3 Validation of Custom Matrix MAC Instructions
7. **Documentation and Reporting**
   - 7.1 Project Documentation
   - 7.2 Schematic Diagrams
   - 7.3 User Manuals
   - 7.4 Project Reports
8. **Project Management and Coordination**
   - 8.1 Project Meetings
   - 8.2 Progress Tracking and Reporting
   - 8.3 Issue Resolution
   - 8.4 Risk Management
   - 8.5 Resource Allocation
9. **Quality Assurance**
   - 9.1 Quality Control Checks
   - 9.2 Code Reviews
   - 9.3 Testing and Validation Audits
10. **Project Completion and Delivery**
    - 10.1 Final System Integration
    - 10.2 Final Testing and Validation
    - 10.3 Handover to End-Users or Stakeholders
    - 10.4 Project Closure and Documentation Archive
11. **Post-Project Evaluation**
    - 11.1 Post-Implementation Review

11.2 Lessons Learned

11.3 Future Enhancement Recommendations

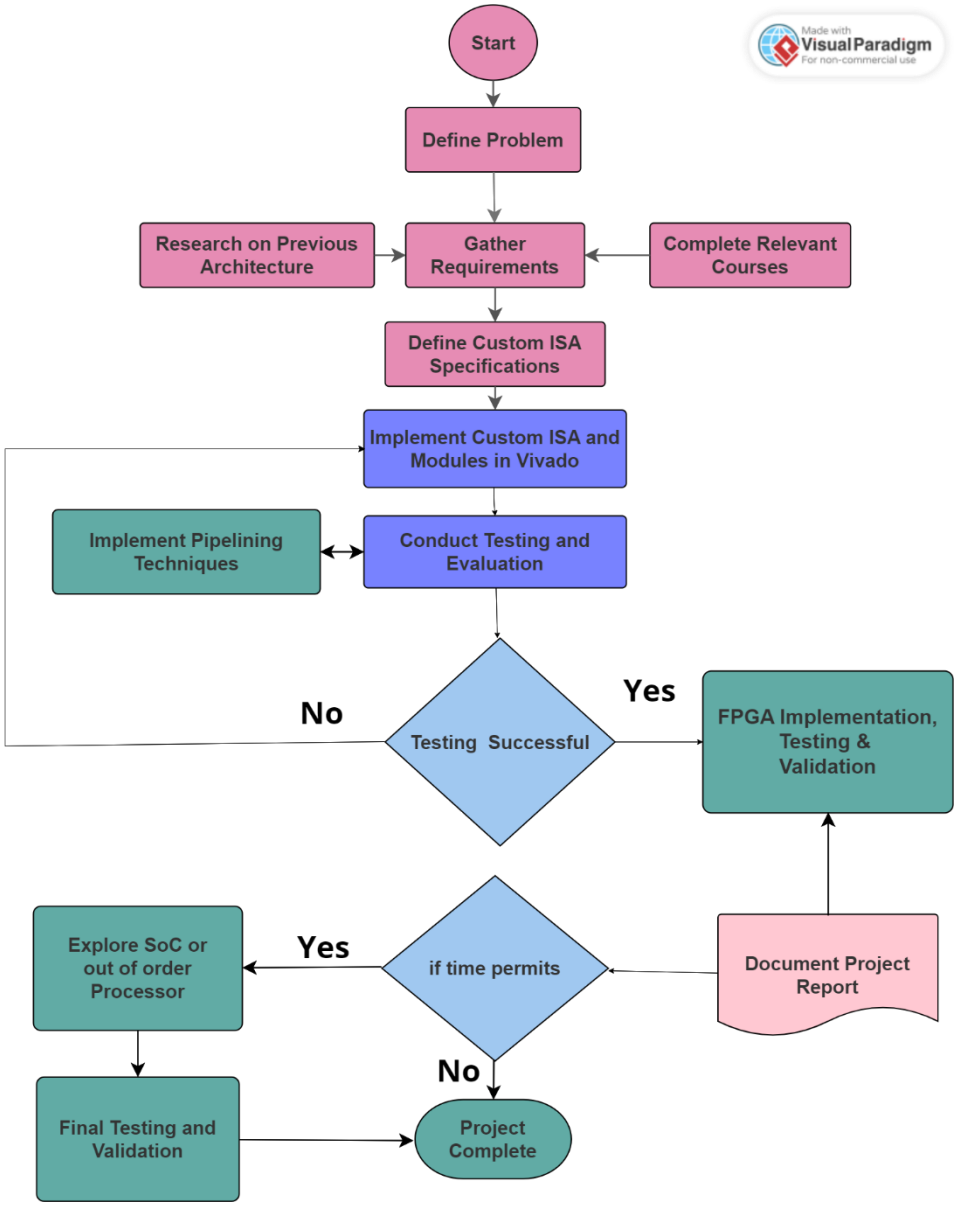

Figure 1 Workflow Chart

4. Design

4.1 ARCHITECTURAL DESIGN

4.1.1 Single Cycle Processor

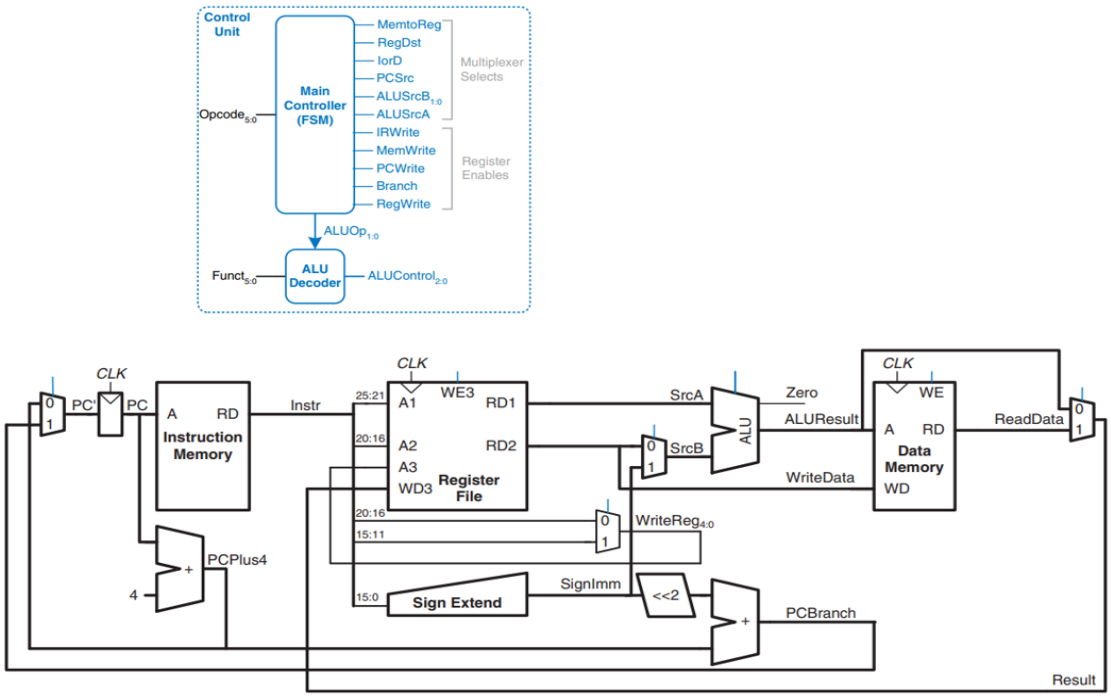

Figure 2 Single Cycle Processor

4.1.2 Staged Pipelined Processor

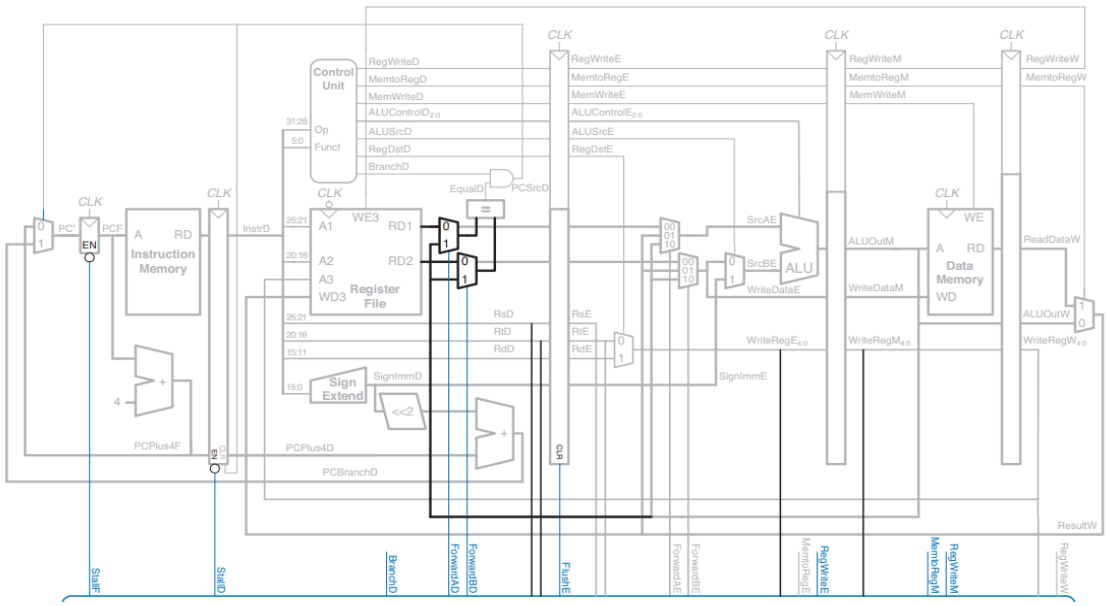

Figure 3 5-Staged Pipelined Processor

4.1.3 Staged Pipelined Matrix MAC Processor

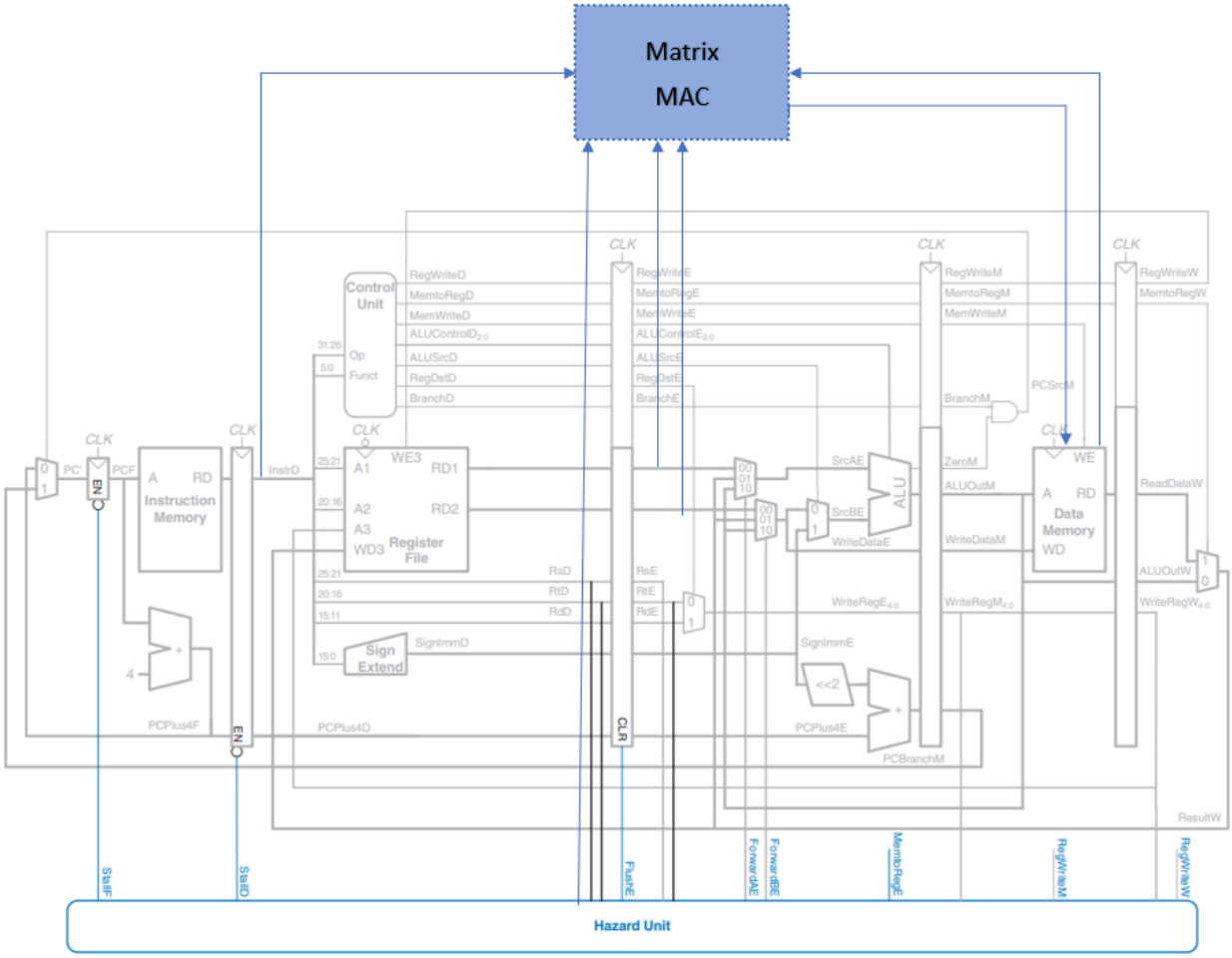

Figure 4 5-Staged Pipelined Matrix MAC Processor

The "Matrix MAC" module will be added to the 5-stage pipelined processor. It will have 4 inputs and 1 output. The current instruction will be one of the inputs to figure out which kind of instruction needs to be executed. "RD1" and "RD2" Values will also be fetched from the Register File to the MAC Module. The data memory will also be available to the MAC Module to figure out the inputs as well as the outputs to store them later Onn when the execution is completed.

4.2 Why did we choose Architecture Design?

The selection of our FPGA-based RISC-V processor architecture with an integrated Matrix MAC module is a strategic choice rooted in the pursuit of performance, versatility, and adaptability. This architecture stands out for several compelling reasons:

1. High-Performance Computing: Our design prioritizes high-performance computing by incorporating a 32-bit RISC-V processor as its foundation. This standard RISC-V core provides a robust and well-established framework for executing general-purpose instructions efficiently.

2. Custom Matrix MAC Instructions: By introducing a custom instruction set architecture (ISA) tailored for matrix operations, we optimize the processor's capability to perform complex matrix calculations. This customization is particularly valuable for tasks that require matrix multiplication and addition, such as machine learning algorithms and scientific simulations.

3. FPGA Compatibility: Leveraging Field-Programmable Gate Arrays (FPGAs) as our hardware platform aligns with our goal of rapid prototyping and hardware acceleration. FPGAs offer the flexibility to configure and reconfigure hardware components swiftly, making them an ideal choice for iterative development and real-time testing.

4. Pipelining for Throughput: The adoption of a pipelining architecture with multiple stages, including fetch, decode, execute, memory, and write-back, ensures efficient instruction throughput. Pipelining enhances the processor's ability to execute instructions in parallel, thus improving overall performance.

5. Software Integration: Our architecture seamlessly integrates software development tools like Vivado for hardware description and synthesis, Venus for assembly code development, and DigitalJS for schematic design. This integration streamlines the development process and allows for effective software-hardware co-design.

6. Versatility and Compatibility: We ensure compatibility with the RISC-V standard (RV32I) while enhancing the processor with custom Matrix MAC instructions. This versatility enables our processor to execute both standard RISC-V instructions and specialized matrix operations, making it adaptable to a wide range of applications.

4.3 MODULE IDENTIFICATION

1. RISC-V Processor Core Module: Responsible for executing basic RISC-V instructions adhering to the RV32I standard. Contains the core components like the ALU, Register File, Control Unit, and Hazard Unit.

2. Custom ISA Module: Defines and implements a custom instruction set architecture optimized for matrix operations, including Matrix MAC instructions.

3. Matrix MAC Execution Module: Efficiently executes Matrix Multiply-Accumulate (MAC) instructions for matrix operations, including matrix multiplication and addition.

4. Pipeline Stage Modules: Comprising fetch, decode, execute, memory, and write-back stages, these modules enable hardware pipelining for instruction throughput optimization.

5. Memory Hierarchy Module: Manages data memory access and includes mechanisms for conflict resolution and hazard handling.

6. Performance Optimization Module: Focuses on optimizing clock frequency, pipeline efficiency, memory access, and overall system performance.

7. FPGA Interface Module: Ensures compatibility with the selected FPGA platform, handling FPGA-specific configurations and interactions.

8. Software Integration Module: Integrates software tools like Vivado for hardware description and synthesis, Venus for assembly code development, and DigitalJS for schematic design.

9. Testing and Validation Module: Encompasses unit testing of hardware components, integration testing of the processor, and validation of custom Matrix MAC instructions.

10. Documentation and Reporting Module: In charge of project documentation, schematic diagrams, user manuals, project reports, and knowledge transfer materials.

11. Project Management and Coordination Module: Manages project meetings, progress tracking and reporting, issue resolution, risk management, and resource allocation.

5. ARCHITECTURE VIEW

5.1 Development View:

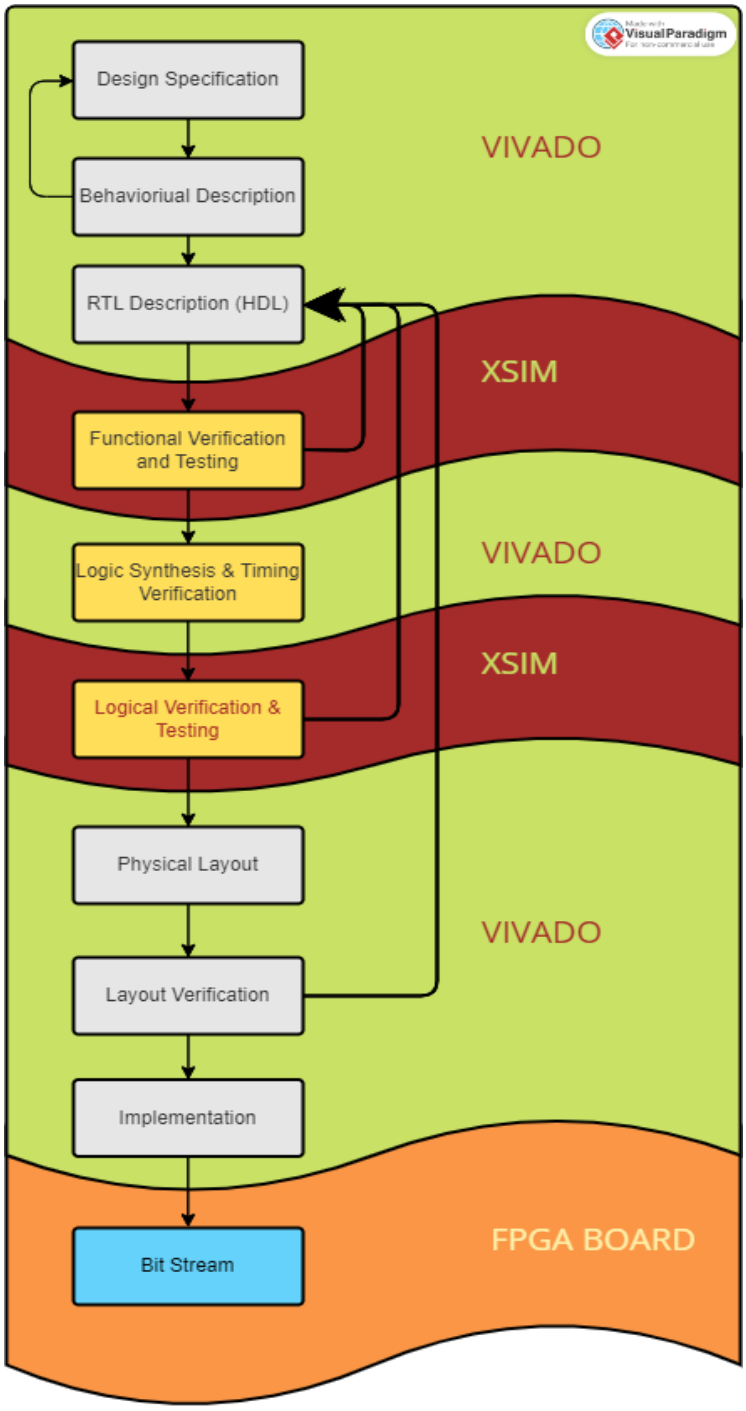

Figure 5 Development Workflow View

5.2 Physical View:

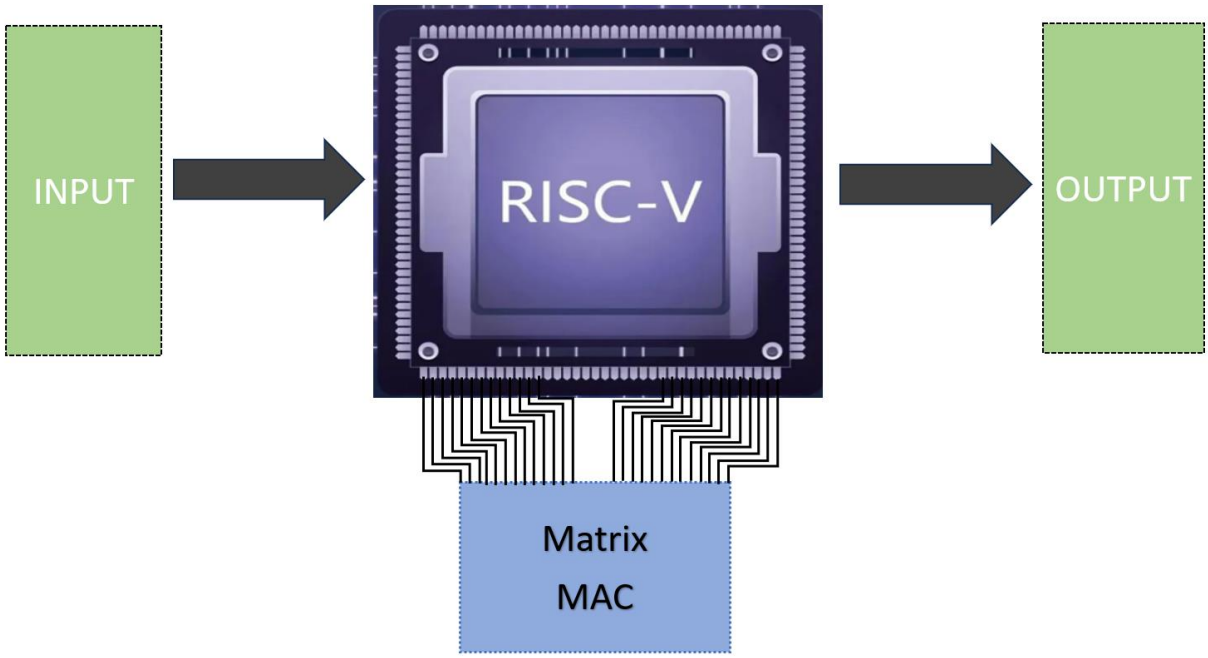

Figure 6 Physical Architectural View