ADVANCED VLSI SYSTEM DESIGN

Fall 2024
Rev 1

Preliminary Project Planning Form

Due day: 6:00pm 11/08/2024
One per team. Submit to the course website on Moodle.
(Grades of this form is part of final project. **Please answer with cautions!**)

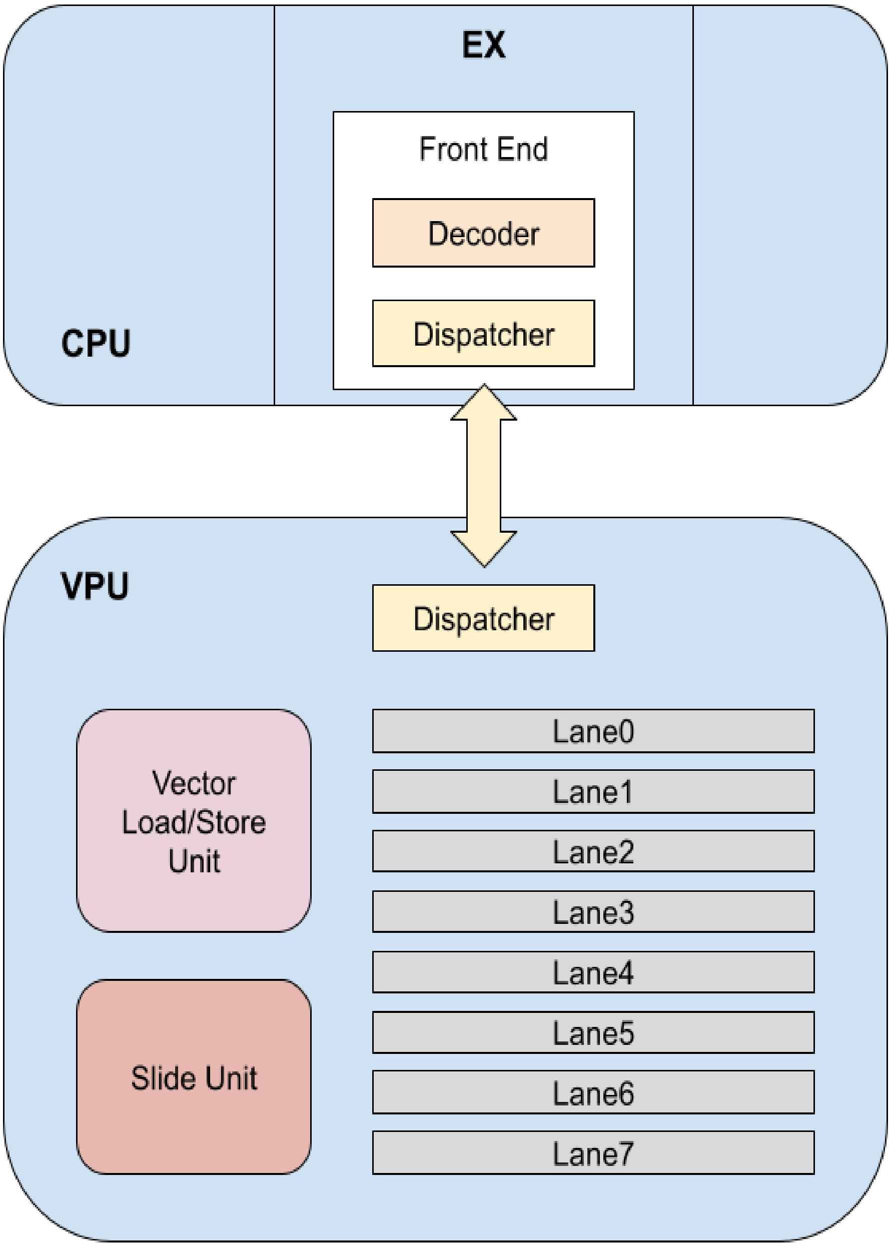
TEAM Name: iLoveAVSD
(If you want to change your team name, please also specify your old team name.)

Team Leader Name: 林承炫

Members Name: 吳承恩、姚磊漢、高揚喻、陳昱中

Target Application for ASPU or Dual-Core	A RISC-V zve64x Vector Co-Processor
Please describe your target application with short motivation and key components that will be related to your application processor (Note that audio is one of applications. ADC/DAC is one of important components, but is not considered as an application.)	**[Overview]** The proposed project focuses on designing a high-performance vector processing unit to enhance computational efficiency in parallel data processing tasks, particularly those involving large-scale arithmetic operations such as vector addition and matrix multiplication. This vector processing unit is intended to provide substantial speedups in data-intensive applications, such as **deep learning inference, computer vision, cryptography, and multimedia processing**. **[key parts with block diagram]** A **Lane** in a vector processor is essentially a smaller, independent processing unit within the vector execution pipeline. Each lane operates in parallel with other lanes, allowing the vector processor to process multiple elements of a vector simultaneously. Lanes are typically replicated within the vector processor, with each one handling one element of the vector at a time.

A **Vector Load-Store Unit (LSU)** in a vector processor is responsible for managing data transfers between memory and the vector processor. It connects to the AXI bus to read or store data in batches. The LSU improves data throughput and ensures efficient handling of vectorized data, especially for large data sets used in applications like image processing or scientific computing. Additionally, it operates on a separate data path from the original CPU, allowing independent and parallel data transfer streams for scalar and vector data.



Please describe your application with targeting specification and how the application processor will work with CPU & memory on both hardware and software sides. NOTES. If you provide a specification that is too easy or too vague, your proposal may be revoked. Therefore, do pay more efforts on this part.	[Overview] Our target application for this project is **machine learning**, specifically accelerating deep learning models such as yolo on edge devices. Machine learning models, especially neural networks, involve a vast number of matrix multiplications, convolutions, and other operations that require processing large datasets with repetitive patterns. These operations are highly suited to vector processing, where a single instruction can operate on multiple data elements in parallel (SIMD). [Specification] • Support RISC-V v-extension spec (zve64x), including vector loads and stores, vector integer arithmetic, vector fixed-point, vector reduction operations, vector mask, vector permutation. • 32 vector registers (VLEN = 64) • Data formats: SEW support for int8, int16, int32, int64 • Support LMUL = 1 • Independent memory access paths for RVV load/store (Has its own LSU connect to AXI bus) • Frequency: Target operating frequency of 1+ GHz **RISC-V v-extension spec:** **zve64x**. This extension is chosen to ensure the capability to run AI models that require extensive matrix operations, as well as to handle fixed-point multiplication computations. The 64-bit VLEN provides the necessary capacity to perform these complex calculations efficiently, allowing for higher precision in arithmetic operations that are critical for AI model accuracy. **Data Formats:** Supports 8-bit, 16-bit, 32-bit, and 64-bit data types. This versatility is essential for accommodating different levels of precision required by various applications, from low-precision deep learning inference (e.x. int8 model) to high-precision scientific calculations.

Frequency:

Target operating frequency of 1 GHz, achieving a theoretical performance of over 20 GFLOPS. The chosen frequency provides a good balance between performance and power consumption, allowing the accelerator to deliver high computational throughput while maintaining energy efficiency.

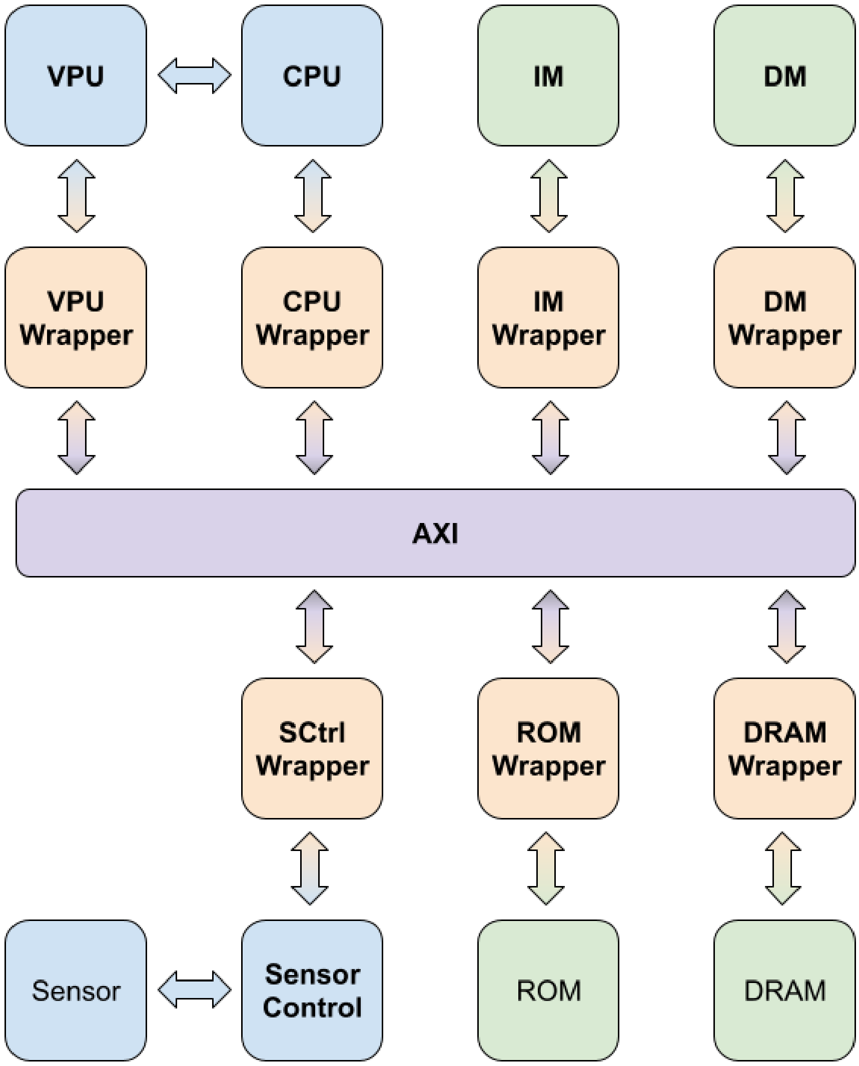
Performance and Power Justification:

The 1 GHz operating frequency is selected based on the available hardware technology, providing sufficient computational power to handle complex vector operations while keeping power consumption within acceptable limits. The target performance of 20 GFLOPS is designed to meet the demands of deep learning inference, which require high computational throughput to process large datasets effectively. The combination of high frequency and efficient vector operations enables the accelerator to deliver significant speedups compared to conventional scalar processing. Power consumption is also a critical consideration in the design. By using a frequency of 1 GHz, we ensure that the accelerator operates within a power-efficient range, balancing performance gains with energy use.

[Work with CPU, illustrated with figures if possible]

The CPU fetches the vector instructions as part of its normal instruction stream. When a vector instruction is identified, the CPU uses the **Dispatcher Unit** in the Execute (EXE) stage to hand off the instruction to the vector co-processor.

Once the instruction reaches the vector co-processor, it performs the necessary vector operations, including vector loads and stores, vector integer arithmetic, vector fixed-point, and so on. Results from the vector co-processor are returned to the CPU (scalar result) or directly stored in memory using its own **Vector Load Stote Unit (VLSU)** , which connects to the AXI bus.

	 VPU CPU IM DM VPU Wrapper CPU Wrapper IM Wrapper DM Wrapper AXI SCtrl Wrapper ROM Wrapper DRAM Wrapper Sensor Sensor Control ROM DRAM
Please provide task assignment for every member. There shall be at least one person dedicate to verification of IPs.	**Team Division:** - Implementation Group：吳承恩、林承炫 - Verification Group：姚磊漢、陳昱中 - Software Group：吳承恩、高揚喻 **Implementation Group:** This team is responsible for the complete system implementation, covering multiple aspects from design to integration. Specifically, their work primarily includes the design and optimization of the Vector Processing Unit (VPU), CPU integration, bus interconnect configuration, DMA controller implementation and management, as well as the integration of related wrappers and connections. Team members need to ensure smooth communication between these

components to achieve stable system operation. Additionally, they must consider resource allocation and power consumption control to meet design requirements.

Verification Group:

This team is dedicated to writing formal verification SVA files and will collaborate closely with the implementation team. When drafting formal verification SVA files, they will frequently consult with the implementation team to verify specifications and circuit behavior, achieving synchronous progress between verification and design. This approach ensures that the system meets our expected specifications and functions as intended.

Software Group:

This team is mainly responsible for designing software that can run smoothly on the completed CPU, focusing on data-intensive tasks like deep learning inference and image processing. They will work closely with the implementation team to ensure compatibility and seamless cooperation between software and hardware. Additionally, the team will optimize applications to fully leverage the accelerator's high performance in handling large data tasks.

Please provide project time schedule by providing all members milestones for their own tasks using a chart. Note that please plan by week and DO check the dates for demo and final presentation.

Section1:

The main goal is to define the core architecture and establish system requirements.

Item	~ 11/11	11/12
Define VPU design, CPU integration, system architecture	Implementation Group	
Establish SVA verification requirements and initial specifications	Verification Group	
Define application scenarios and major software requirements	Software Group	
Team Communication and Coordination		Team Work

Section2:

The team focuses on initial development of the main components.

Item	11/13 ~ 11/25	11/26
Start VPU, DMA controller implementation; design and test bus interconnect	Implementation Group	
Write initial SVA files; verify specifications with implementation group	Verification Group	
Develop basic software framework and golden data generation	Software Group	
Team Communication and Coordination		Team Work

Section3:

The team’s main focus is system integration and initial testing.

Item	11/27 ~ 12/9	12/10
Complete VPU and CPU integration; system simulation	Implementation Group	
Early simulation verification; ensure expected system operation	Verification Group	
Assist with simulation and integration	Software Group	
Team Communication and Coordination		Team Work

Section4:

The team is dedicated to optimization and functional testing.

Item	12/11 ~ 12/23	12/24
Optimize resource allocation and power consumption; adjust for target performance	Implementation Group	
Full functional verification; confirm all functionalities meet specifications	Verification Group	
Application-level performance testing and optimization	Software Group	
Team Communication and Coordination		Team Work

Section5:

The team focuses on final integration, validation, and demonstration preparation.

Item	12/25 ~ 1/7	1/8 ~ 1/15
Complete final system integration	Implementation Group	
Final system check to ensure no errors	Verification Group	
Complete demo application development; prepare demo scenarios	Software Group	
Prepare and conduct system demonstration; submit final report		Implementation Group