Fiona O’Connell, Kelly Deal, Danni Krauser

Milestones List

(Milestones in bold, with tasks labeled by the member in charge’s initial)

1. **Algorithm Selection** - decide on the specific software algorithm to implement.
    a. K: Research software algorithms to implement.
2. **CPU-Only Implementation** - develop the CPU-only software implementation in C++
    a. K: Develop the CPU-only software implementation in C++
3. **Performance Measurement** - implement the test vectors and datasets for software validation and measure the baseline performance metrics
    a. K: Implement the test vectors and datasets for validation and performance measurement.
    b. K: Measure the baseline latency, throughput, and CPU usage of the C++ software on the ARM processor
    c. K: Analyze the C++ software to identify the bottleneck for hardware migration.

4. **Hardware Architecture -** design the core hardware architecture for the accelerator.
    a. F: Design the core architecture of the hardware accelerator
5. **Interface Design** - define the AXI-Lite interface and its corresponding register map.
    a. F: Define the AXI-Lite interface and register map for sending control signals and retrieving status to/from the accelerator
    b. F: Document the memory and I/O mapping required for the host C++ application to communicate with the hardware IP
    c. D: Research design patterns for integrating the ARM software driver with the generated AXI-Lite hardware IP core registers.
6. **Data Transfer Configuration** - configure the AXI interfaces for data transfer.
    a. F: Configure AXI interfaces for efficient data transfer between memory and the hardware accelerator
7. **IP Core Validation** - validate the generated hardware IP core using Cocotb
    a. F: Validate the generated hardware IP core using a simulation testbench (Python's cocotb)
8. **System Integration** - create the Vivado block diagram to integrate the custom hardware IP core with the ARM processor and memory system.
    a. F: Create the Vivado block diagram to integrate the custom hardware IP core with the ARM processor and memory system.
    b. F: Synthesize, place, and route the complete integrated hardware design onto the Zynq FPGA fabric
    c. F: Verify the generation of the bitstream and load the hardware configuration onto the Zynq board.
9. **Host Application** - develop the host C++ application to manage data transfers and calls the hardware accelerator

    a. F: Develop the host C++ application for the ARM processor that calls the hardware accelerator and handles data buffer transfers
    b. F: Refine the host application's data handling to ensure optimal data transfer rates across the AXI buses.

10. **End-to-End Testing** - test the end-to-end functionality of the hardware-accelerated system.
    a. D: Test the end-to-end functionality of the hardware-accelerated system against the software baseline.
11. **Performance Analysis** - measure the latency and throughput of the accelerated version and perform a comparative analysis with the software baseline.
    a. D: Measure latency and throughput metrics of the hardware-accelerated version running on the Zynq board.
    b. D: Determine power consumption of the Zynq board during the execution of both the baseline and accelerated tests
12. **Results Presentation** - create visuals and a report to present the quantifiable results of the hardware acceleration
    a. D: Create visuals to present the quantifiable results of the hardware acceleration.

Timeline

Semester 1 Deliverables

Week 6	
Week 7	Algorithm Selection - 10/13
Week 8	
Week 9	
Week 10	CPU-Only Implementation - 11/3
Week 11	Performance Measurement - 11/10
Week 12	
Week 13	Hardware Architecture - 11/24
Week 14	
Week 15	Interface Design - 12/8

Semester 2 Deliverables

Week 1	

Week 2	
Week 3	Data Transfer Configuration - 1/26
Week 4	
Week 5	IP Core Validation - 2/9
Week 6	
Week 7	System Integration - 2/23
Week 8	Host Application - 3/2
Week 9	(Finalize and optimize)
Week 10	
Week 11	End-to-End Testing - 3/23
Week 12	Performance Analysis - 3/30
Week 13	Results Presentation - 4/6
Week 14	
Week 15	

Effort Matrix

	Fiona	Kelly	Danni
Algorithm Selection	15%	70%	15%
CPU-Only Implementation		90%	10%
Performance Measurement		25%	75%
Hardware Architecture	100%		
AXI Interface Design	100%		
Data Transfer Configuration	90%	10%	
IP Core Validation	100%		
System Integration	90%	10%	

Host Application	50%	50%	
End-to-End Testing	25%	25%	50%
Performance Analysis		10%	90%
Results Presentation	10%	10%	80%