

# Project 4

**Due: 11/26/2025 10:30 AM**

## 서론

- (1) 각 프로젝트의 파일은 Project<번호> -> Problem<번호> 폴더 구조로 구성한다. 예를 들어, Project4->Problem1 의 폴더에 Problem1에서 요구하는 모든 파일을 저장한다. Project 폴더를 압축(zip)하고 압축 파일 이름은 학번으로 하여 제출한다. 명시된 폴더 구조를 따르지 않아 발생하는 불이익에 대한 책임은 본인에게 있음.
- (2) 보고서가 필요한 경우는 pdf 형식으로 제출하며, 언어는 영어/한글 모두 무방하다.
- (3) 프로젝트의 deadline 까지 LMS의 <과제 및 평가>란에 제출한다.
- (4) 프로젝트의 마감기한을 지키지 못한 경우, 감점 없이 0 점 처리한다.
- (5) 프로젝트는 개별로 수행하며, 다른 사람 또는 자료에서 도움을 받거나 이를 참고로 하였을 경우 관련 소스코드와 보고서에 출처를 명확히 제시하여야 한다. 또한, 다른 학생에게 도움을 주었을 경우에도 소스코드와 보고서에 이에 관련하여 서술한다. 그렇지 않을 경우 모두 0 점 처리한다.

## Overview

LLM(Large Language Model), CNN (Convolutional Neural Network)을 비롯한 다양한 application에서 MAD(Multiply-add) operation이 가장 많은 실행시간을 차지하게 된다.

Project 4에서는 LLVM, GCC, 그리고 RISC-V core를 이용하여 IMAD(Integer Multiply and Addition/Accumulation) 연산을 추가하고, IMAD 적용 이전과의 성능을 비교 분석한다. 제공된 application code(conv.c)는 1-D convolution을 수행하며, 그림 1과 같은 IMAD 연산이 포함되어 있고, baseline architecture에서는 이를 MUL, ADD 두 개의 instruction을 통해 수행한다. Compiler가 IMAD instruction을 지원한다면, 두 개의 instruction이 아닌 IMAD instruction 하나로 수행할 수 있고, code size를 줄일 수 있다.

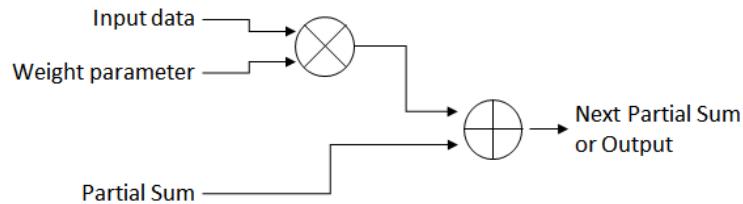
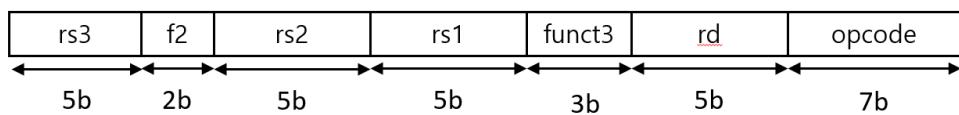


그림 1. IMAD(Integer Multiply and Accumulation) 연산

## 문제 1(150 점): LLVM & GCC Extension for IMAD Instruction

% 문제에서 수정할 것을 명시한 파일만 수정한다. 문제 1의 폴더 구조는 다음과 같이 구성한다. Problem1/{LLVM, GCC}

Application code (conv.c)의 IMAD operation 을 RISC-V architecture 에서 수행할 IMAD instruction 으로 생성하기 위해서 LLVM 및 GCC 를 수정해야 한다. 3 source operand 를 이용하는 IMAD instruction 의 format 은 아래와 같다. IMAD instruction extension 을 위해 Project 4 overview 를 참고하여 LLVM 과 GCC 를 수정하고, 다음을 제출한다. 함께 제공된 "imad.ll" file 을 통해 확장된 instruction set 을 test 할 수 있다.



(1) (50 점) IMAD instruction 이 확장 및 검증된 다음의 LLVM compiler source file.

- ISDOpcodes.h, RISCVInfo.td

(2) (50 점) IMAD instruction 이 확장 및 검증된 다음의 GCC assembler source file.

- riscv-opc.c, riscv-opc.h, riscv-dis.c

(3) (50 점) 컴파일러 수정에 대한 내용을 정리한 보고서(project4.pdf)

## 문제 2(100점): RISC-V Core Simulation for IMAD Instruction

제공된 Makefile에서, compile option중 optimization level을 "-O0"로 설정하고 다음을 진행한다. Makefile을 이용하여 "conv.c" file을 compile하면, 문제 1에서 수정한 compiler가 IMAD를 포함하는 instruction을 생성한다. Project 4 overview를 참고하여 IMAD instruction 확장 전후에 대한 conv(binary file)을 제공된 RISC-V core에서 실행하고 성능분석한 내용을 project4.pdf에 작성한다.

(1) (50 점) 실행코드 크기에 대한 비교

(2) (50 점) Retired instruction 수 측정 및 비교

- conv.c 의 conv 함수를 실행할 때의 retired instruction 수를 측정할 것