

# Project 4

**Due: 11/27/2023 10:30 AM**

## 서론

- (1) 각 프로젝트의 파일은 Project <번호> -> Problem <번호> 폴더 구조로 구성한다. 예를 들어, Project 4->Problem 1 의 폴더에 Problem 1 에서 요구하는 모든 파일을 저장한다. 모든 파일은 Project 4 하에서 압축(zip)하고 압축 파일 이름은 학번으로 한다.
- (2) 보고서가 필요한 경우는 pdf 형식으로 제출한다.
- (3) 프로젝트 제출일의 수업시작(오전 10 시 30 분) 전까지 blackboard 를 통하여 제출한다.

## ■ 모든 프로젝트는 개인별 프로젝트입니다.

## 문제 공통: Overview

CNN (Convolutional Neural Network)은 다양한 application 에서 사용되고 있으며 MAD (Multiply-add) 연산을 기반으로 한다. 따라서, application 의 성능을 높이기 위하여 MAD 연산을 가속하기 위한 다양한 방법들이 제안되고 있다.

Project 4 에서는 LLVM & GCC compiler 와 RISC-V core 를 확장하여, IMAD(Integer Multiply and Addition/Accumulation) 연산을 추가하고, baseline architecture 와의 성능을 비교, 분석한다. 제공된 application code (conv.c)는 1D convolution operation 를 수행하며, 여기에는 그림 1 과 같은 IMAD 연산이 포함되어 있고, baseline architecture 에서는 이를 MUL, ADD 두 개의 instruction 을 통해 수행한다.

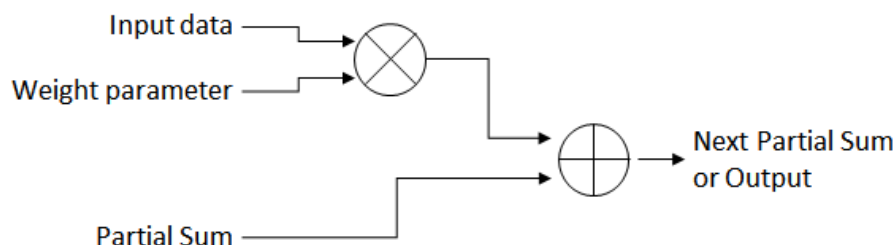


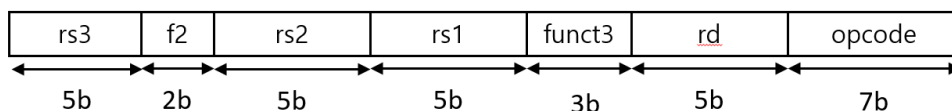
그림 1. IMAD(Integer Multiply and Accumulation) 연산

Compiler 와 RISC-V core 가 IMAD instruction 을 지원한다면, 두 개의 instruction 이 아닌 IMAD instruction 하나로 수행할 수 있고, code size 를 줄일 수 있다.

Project 4 는 기존에 제공된 VMware 내의 LLVM, GCC 와, project4 에서 제공된 RISC-V core 를 이용하여 진행한다.

### 문제 1(80 점): LLVM & GCC Extension for IMAD Instruction (SW)

Application code (conv.c)의 IMAD operation 을 RISC-V architecture 에서 수행할 IMAD instruction 을 생성하기 위해서 제공된 LLVM 및 GCC 를 수정해야한다. IMAD instruction 의 format 은 아래와 같으며 3 개의 source operand 를 이용한다. 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) (40 점) IMAD instruction 이 확장 및 검증된 다음의 LLVM compiler source file.

- ISDOpcodes.h, RISCVInfo.td

(2) (40 점) IMAD instruction 이 확장 및 검증된 다음의 GCC assembler source file.

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

### 문제 2(60점): RISC-V Core Extension for IMAD Instruction (HW)

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 RISC-V core가 IMAD instruction을 수행할 수 있도록 datapath와 control path를 추가한 뒤, 수정된 HW에서 conv (binary file)의 실행을 simulation한다. 이때 추가된 회로에 의한 critical path의 증가는 고려하지 않는다.

(60 점) 추가된 IMAD instruction을 수행할 수 있도록 수정 및 검증된 다음의 RISC-V Verilog source file.

- riscv\_decode.v, riscv\_decoder.v, riscv\_regfile.v, riscv\_def.v, riscv\_core.v, riscv\_issue.v, riscv\_pipe\_ctrl.v

### 문제 3(60점): 분석 보고서

IMAD ISA 지원을 위한 컴파일러와 하드웨어를 구현 내용(문제 1 과 2) 및 다음을 포함한 분석 보고 (problem3.pdf)

- (10 점) 함수 conv 및 코드 전체의 실행코드 크기 비교
- (10 점) IMAD 구현 이전 및 이후의 전체 retired instruction 의 수 비교.
- (40 점) 구현 및 분석 보고서