IOC 5009 Accelerator Architectures for Machine Learning

Lab I

October, 2021

RISC-V, pronounced as Risk-Five, is an open specification of an ISA, or Instruction Set Architecture. It's an open source instruction set architecture. The RISC-V project came from UC Berkeley and RISC-V has been widely adopted in commercial processor architectures.

RISC-V ISA (Instruction Set Architecture) is designed in a modular way. It means that the ISA has several groups of instructions (ISA extensions) that can be enabled or disabled as needed. This allows implementing precisely the instruction groups that the application needs, without having to pay for area or power that will not be used. One of the groups is special; it has no predefined instructions. Designers can add any instruction they need for the application that they want to accelerate.

In this lab you require to use RISC-V V(vector) extension, which is called RVV, to do your project. The tutorial link of this lab: https://hackmd.io/@KyleLiu-NCTU/HkhSlNeEF

1 Benchmarking RISC -V Instruction Set Architecture

The saxpy is to calculate "y = a * x + y", where "y" and "x" contain multiple elements represented as the vector data type, and "a" is a scalar value. This lab asks you to write a "saxpy" function by using RISC-V and RVV assembly codes. Before you start this lab, you should complete the building of RISC-V toolchains, LLVM and the spike simulator. You should run your assembly programs by using spike simulator and get the number of executed instructions. Please see the above tutorial to build up entire RISC-V toolchains.

In this lab, you should get the following results:

1. The number of executed instructions in the saxpy program written by RISC-V assembly codes. (15%)
2. The number of executed instructions in the saxpy program written by RISC-V V extension assembly codes. (15%)

You should make sure your assembly programs that can work correctly by using the testing input data sets in the lab 1 folder.

RVV benchmark	Description	# of Instr(rvv/no rvv)
saxpy	y[i] = a * x[i] + y[i]	

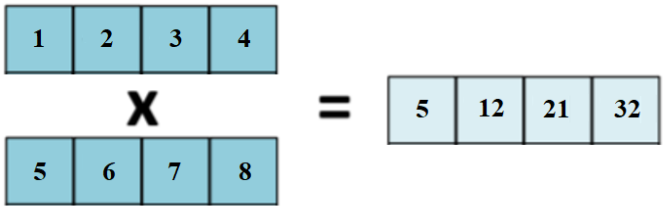
You have to print the result on the terminal, paste the screenshot and code you write in the report.

2 RISC-V ISA Extension

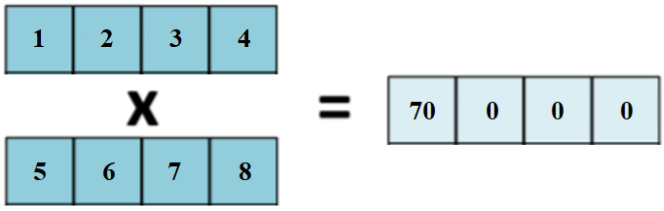
Multiplying two vectors and the dot product are often used in DNN models. However, the RISC-V V extension ISA does not provide a specific instruction to perform these two operations. Therefore, in this lab, you require to create two new RVV extension instructions that can multiply two vectors and the dot product. You should read the above tutorial to figure out the approach to create a new RVV instruction by using the spike simulator.

Note that you can refer files in "mtmac" folder, which contains prototype of two new instructions. You should modify the spike simulator and the assembly codes of mtmac.s to complete the implementation of new instructions. (40%)

a. Vector multiplication (You will get 20% score if you only implement this.)



b. Vector dot product (You don't have to write the report about **a.** if you implement this.)

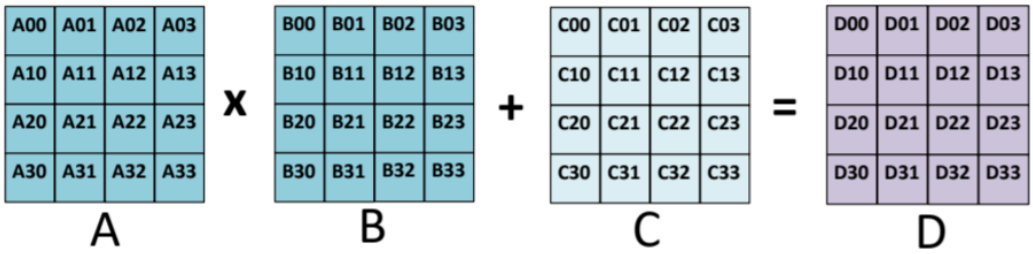


3 RISC-V ISA Extension and Architecture Implementation

Matrix multiplication is often used in DNN operations. This lab asks you to create a specific instruction for matrix multiplication. You should continue the

implementation of the second lab and complete the implementation of the instruction that is specific for the matrix multiplication.

a. Matrix Multiply Accumulate (30%)



You are required to **upload your report to new E3 before the deadline**. Your report should **contain the implementation and result, including the screenshot of code that you modify, the register value in each stage** (print by “spike –d”), **and the output in the terminal**.