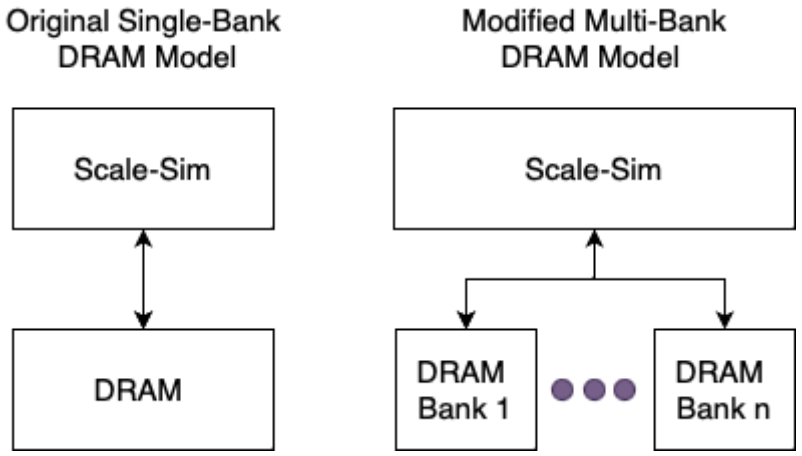
Extend Scale-Sim Accelerator Simulator with Multi-Bank DRAM Model

MOTIVATION

Performance optimization of Neural Network Accelerators on systems with multi-bank DRAM (off-chip memory).

Problem Statement

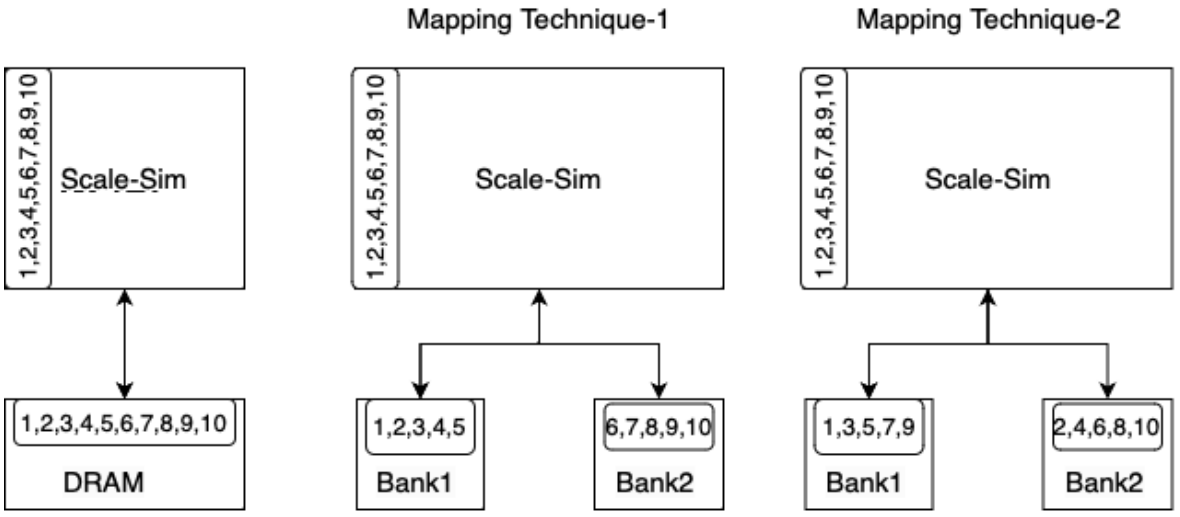
This project will explore multiple memory bank modes in Scale-Sim-v2 (Analytical simulator for NN accelerators). In this mode, the data from the local memory in the accelerator is distributed across different off-chip DRAM banks (instead of one bank as in the current model) with an efficient data mapping strategy. This mode needs several mapping files as input configuration to pre-determine the distribution of inputs in the external memory banks. An efficient memory mapping can harness higher memory bandwidth due to multiple DRAM banks to increase the accelerator performance.



In this project, we will explore different mappings from Accelerator SRAM to off-chip multiple DRAM banks to improve the accelerator performance and memory utilisation. We will also explore if efficient memory mapping to multiple banks varies with different accelerator hardware (PE size, SRAM sizes) and with different dataflows- is, ws, os.

Some of the mapping techniques that you can get started with are as follows:

Mapping Technique-1: This mapping method involves assigning the incoming data blocks to a memory bank ‘i’ as long as it has the space to accommodate them. If the current bank runs out of space, the data blocks get assigned bank i+1; thus, the process continues.
Mapping Technique-2: This mapping method involves cyclically assigning incoming data blocks to memory banks.



To Get Started: Download the source code of Scale-Sim-v2 (the link is mentioned in the Resources section). Try to understand the code file memory_map.py- this is where the number of memory banks is brought into the picture and, therefore, can help understand how changes can be introduced to the Scale-Sim code to support multiple memory bank mapping.

To Explore:
SCALE sim can estimate the following:

- Run time in cycles
- Average utilisation
- On-chip memory requirements
- Off-chip interface bandwidth requirements

Resources:

1. Scale-Sim-v2: https://scalesim-project.github.io – this is the project page with links to the source code (on GitHub), documentation, and tutorials.
2. Scale-Sim-v2 Setup: Instructions are here: https://github.com/scalesim-project/scale-sim-v2
3. CNN: https://cs231n.github.io/convolutional-networks/: This is a good reference for understanding CNNs. The students can explore additional material available in this course.
4. DRAM Memory Module: 15-740/18-740 Computer Architecture Lecture 19: Main Memory
5. SIMD vs. Spatial vs. Systolic Architecture:
   a. Efficient Processing of Deep Neural Networks: A Tutorial and Survey - Good paper to understand reuse – is, ws and os (takes references from Eyeriss); covers SIMD vs Spatial since Eyeriss is a spatial architecture.
   b. https://arxiv.org/pdf/1704.04760.pdf: Google TPU is one of the architectures used in scale-sim. It is used in systolic PEs and could be a good read.