EYERISS – PAPER READING REPORT

- Shreyas S (EP22B040)

1. **Provide a short (100-word) summary of the paper.**

This paper presents a deep neural network accelerator architecture optimized for performance, energy efficiency, and flexibility. Its key contribution is a widely adopted dataflow taxonomy for spatial architectures, emphasizing "stationary" data to reduce memory access. It introduces a "row stationary" dataflow that breaks dense convolutions into parallel 1D primitives, processed row-wise with repeated accumulation. The work also notes that energy efficiency isn't dictated solely by DRAM bandwidth. High data movement to on-chip global buffers also incur substantial energy cost.

2. **What did you learn from the paper?**
   - The different spatial architectures can be classified according to their dataflows as:
     - Weight Stationary, Input Stationary, Output Stationary and No Local Reuse.
   - The paper's row stationary dataflow strategically exploits spatial and temporal reuse by partitioning convolution operations into 1D primitives, enabling parallel processing and reducing on-chip buffer bandwidth
   - This minimizes energy by exploiting reuse across multiple levels of the memory hierarchy (RF, inter-PE communication, global buffer, DRAM).
   - They also developed a systematic framework to analyze and compare dataflows under the same hardware constraints. It quantified energy cost per memory hierarchy access and showed RS achieves 1.4×–2.5× better efficiency in convolutional layers compared to alternatives.
   - The RS dataflow and Eyeriss struck a compromise: simple hardware (no caches, minimal control) but flexible mapping to support different CNN layers.
3. **What did you like about the paper?**
   - The paper's taxonomy of CNN dataflows provides a structured framework to classify and compare designs, which is valuable for systematic analysis.
   - It highlights that energy costs of memory access extend beyond DRAM, showing that significant overhead can arise within the accelerator itself (e.g., high-bandwidth transfers between the global buffer and PE array).
   - This broader perspective emphasizes that optimizing on-chip data movement is as crucial as reducing off-chip accesses for achieving energy-efficient architectures.
4. **What did you dislike about the paper?**
   - The distinction between row-stationary (RS) and output-stationary (OS) dataflows sometimes feels less fundamental than presented.

- It appears that with minor reformulations in computation order and memory access scheduling, an OS dataflow could potentially capture many of the same reuse benefits as RS.
- This raises the concern that the claimed novelty of RS may, in part, reflect how the analysis framework classifies data reuse, rather than a strictly unique capability.

5. **The Eyeriss paper evaluates implementing the AlexNet model on hardware. What pieces are missing from the architecture for successfully running the AlexNet model end-to-end? How would you implement them?**
   - A DMA subsystem seems absent in the accelerator. It might be essential to orchestrate efficient DRAM accesses without stalls.
   - Support for non-linear layers.
     - They may not be as compute intensive as the MAC workloads itself.
     - However, they are essential layers. Post-processing these layers would incur an energy cost.
     - Adding some functional blocks within the accelerator would help in reducing the post-processing cost.
   - To convert model graph into the mapping/folding parameters that Eyeriss expects with essential padding and temporal flow, and to orchestrate end-to-end execution, a software stack is essential.
6. **In the fully connected layers of AlexNet, DRAM accesses still draw the most energy when compared to RF, ALU, etc. Why? How would you reduce this energy cost?**
   - The FC layers lose spatial reusability, that is present in abundance in the convolutional layers.
   - These matrices cannot fit in the global buffer and hence multiple load-store cycles occur for the computation of each layer.
   - When batch size is small, the same weight is reused fewer times, so amortization of the DRAM read energy is poor.
   - Adding a global scratchpad unit, or a larger local register file in the PE might help reduce the data movement cost.
   - Whenever possible, use a large batch-size. This helps in increasing the reuse of data that is otherwise absent in FC layers.