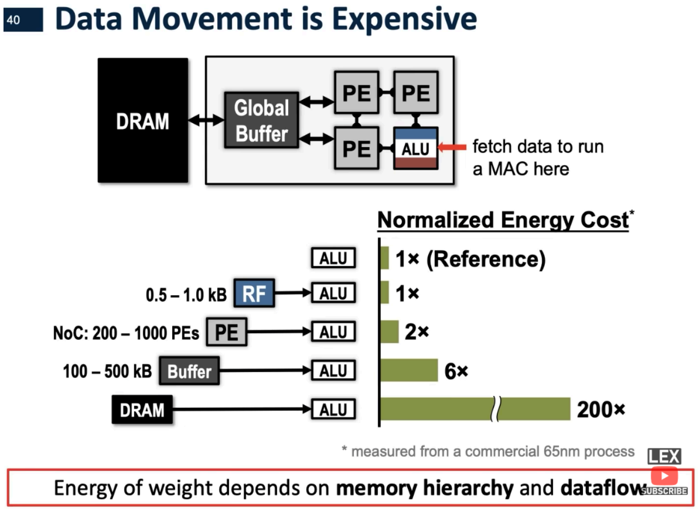
Summary

In Deep Neural Networks, to reduce the data movement from either SRAM or DRAM for the necessary computation introduction of small local memory.



Data Movement is Expensive

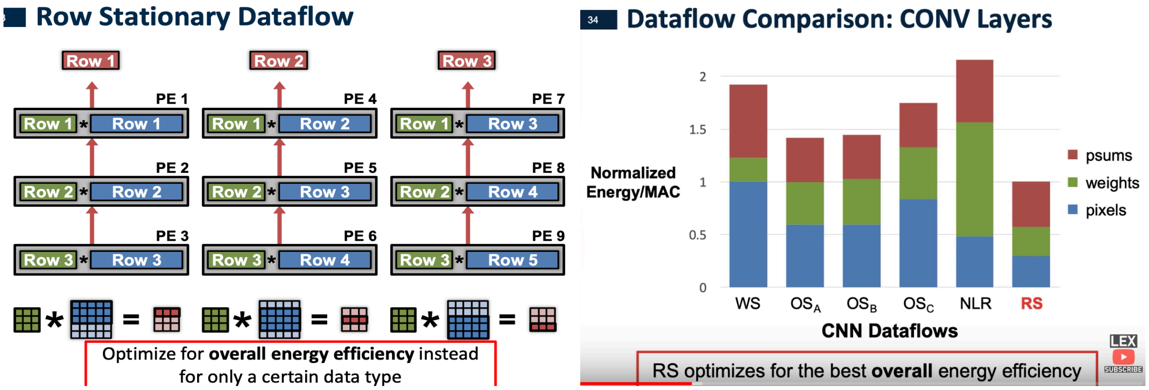
* measured from a commercial 65nm process

Energy of weight depends on **memory hierarchy** and **dataflow**

Figure 1: Data movement and its expenses

Alongside a few of the below techniques were interesting and will be helpful for project.

1. Local memory with Row Stationary (RS) Dataflow mechanism incorporated will be helpful and also optimizes for the best overall energy efficiency.



Row Stationary Dataflow

Optimize for **overall energy efficiency** instead for only a certain data type

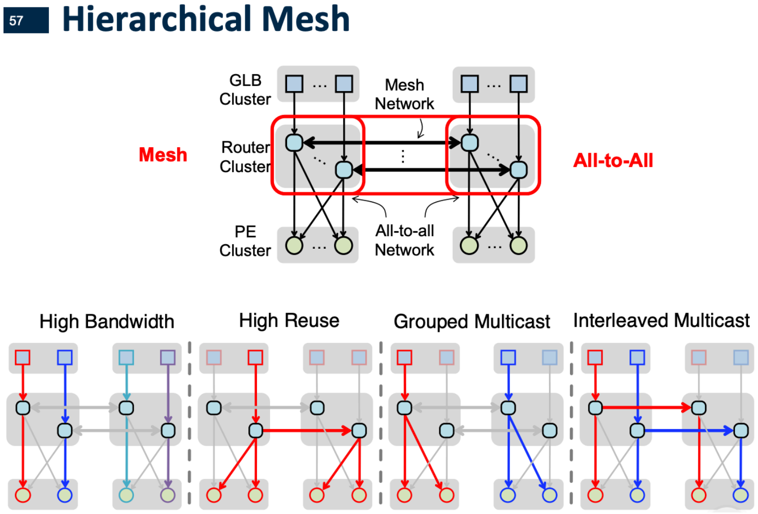
Dataflow Comparison: CONV Layers

RS optimizes for the best **overall** energy efficiency

Figure 2: Row Stationary Dataflow

2. Also by exploiting sparsity either by skip memory access and computation or compress data to reduce storage and data movement will help improve the efficiency of DNN.
3. NetAdapt, a platform-aware DNN adaptation can automatically adapt DNN to a mobile platform to reach a target or energy budget and also uses empirical measurements to guide optimization. Hence overall latency and accuracy both are improved.
   One of the application where NetAdapt can be implemented is Fast Monocular Depth Estimation which comprises of an Auto Encoder DNN architecture with encoding layers (Reduction) and decoding layers (Expansion).

4. Network Pruning and Reducing Precision are other two methods for an efficient DNN design.
5. Finally, an approach called hierarchical mesh with lower level having an all-to-all connection and the higher level with the mesh connection.



57 **Hierarchical Mesh**

Figure 3: Hierarchical Mesh