EE-508: Hardware Foundations for Machine Learning

Modeling Accelerators

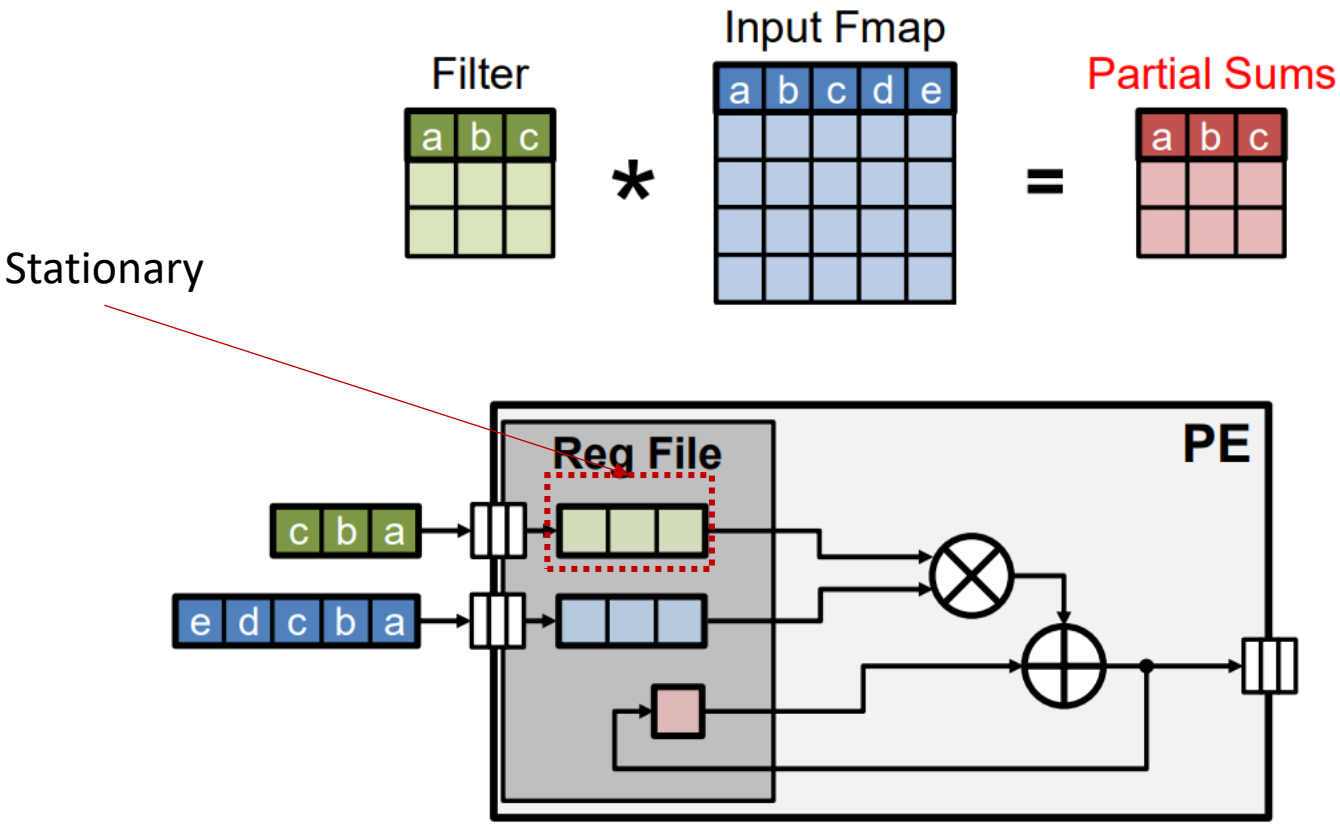
University of Southern California

Ming Hsieh Department of Electrical and Computer Engineering

Instructors:
Arash Saifhashemi

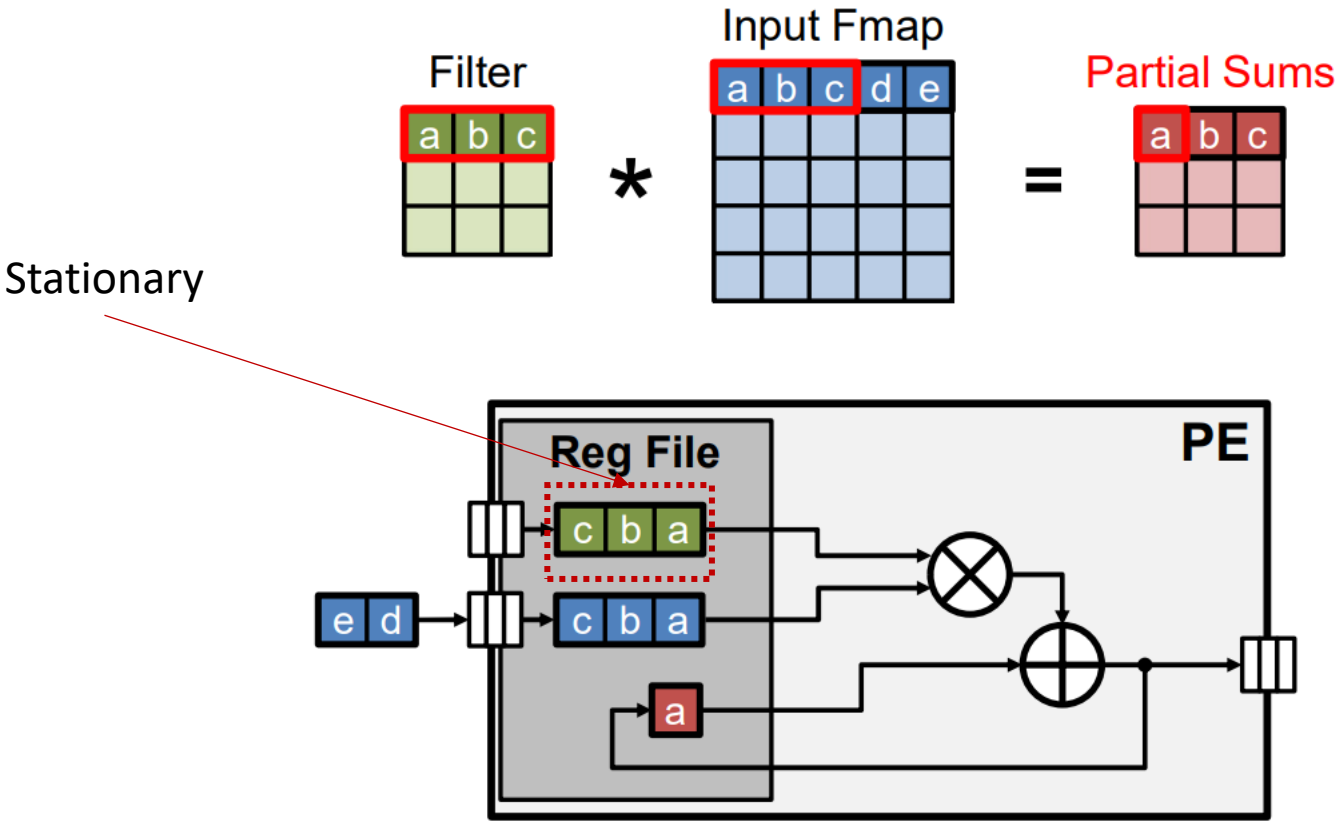
Row Stationary (Eyeriss)

Row Stationary



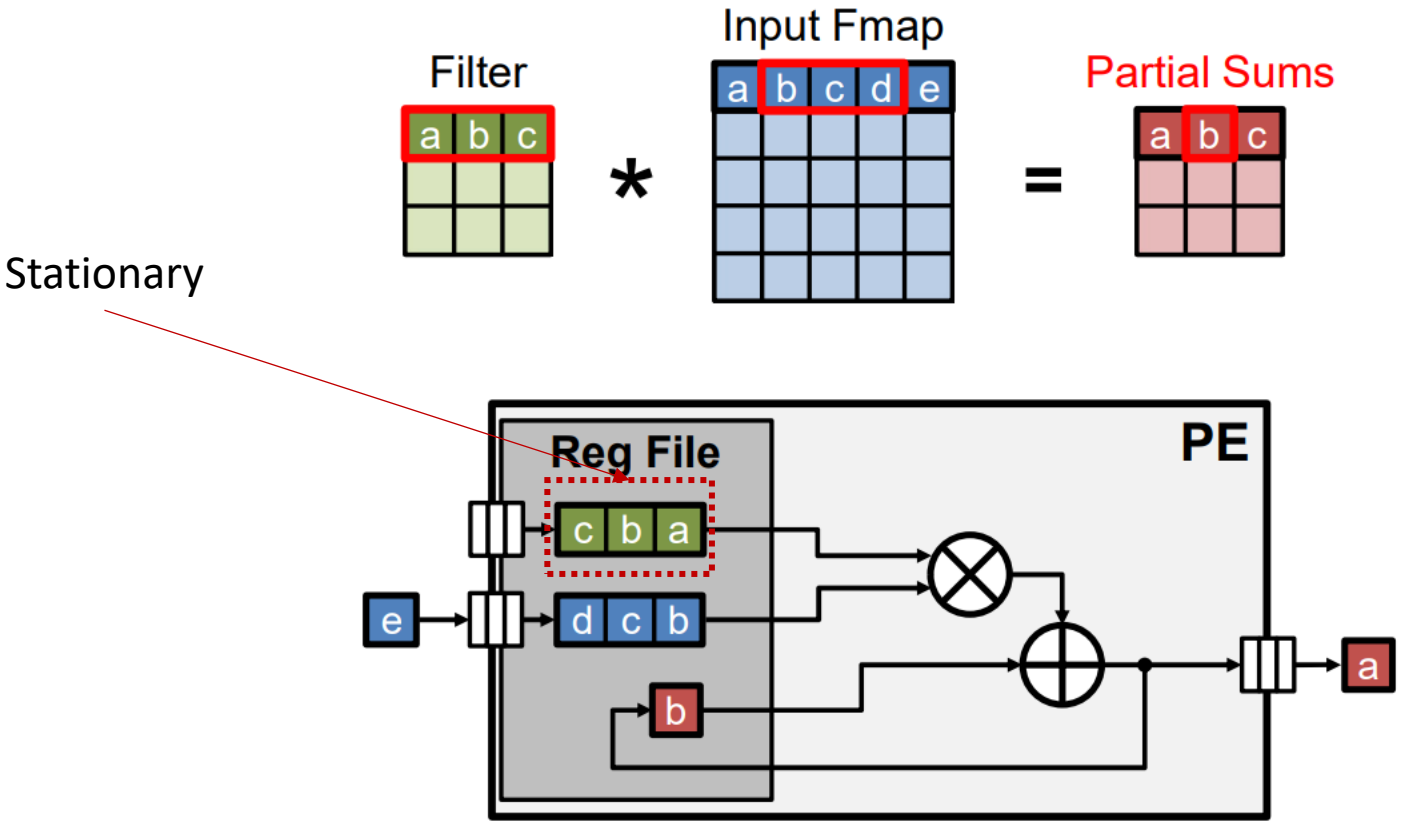
- Keep Filter row stationary
- Stream the IFMAP into PE

Row Stationary



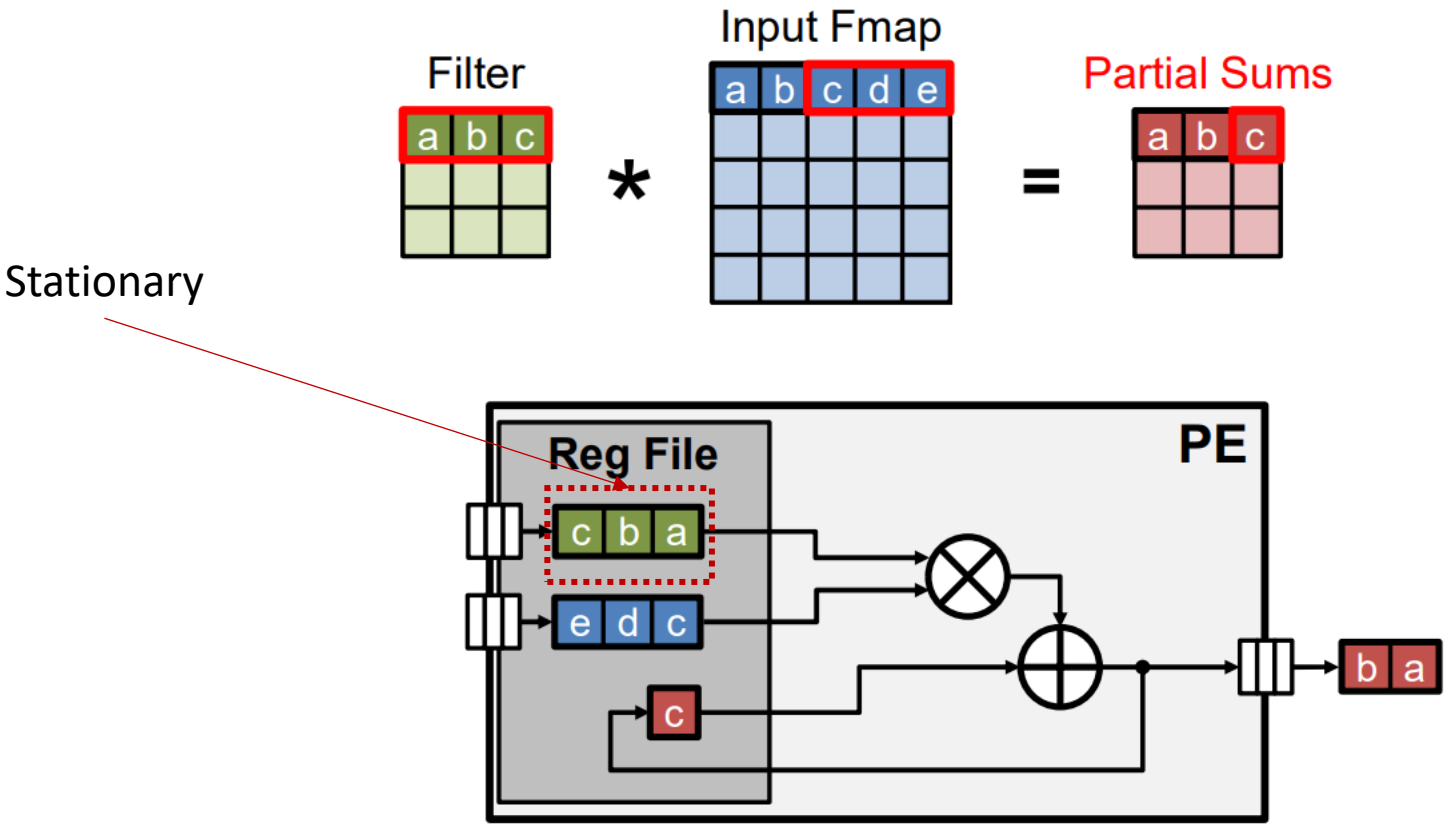
- Keep Filter row stationary
- Stream the IFMAP into PE

Row Stationary



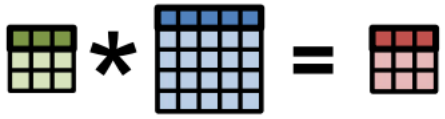
- Keep Filter row stationary
- Stream the IFMAP into PE

Row Stationary

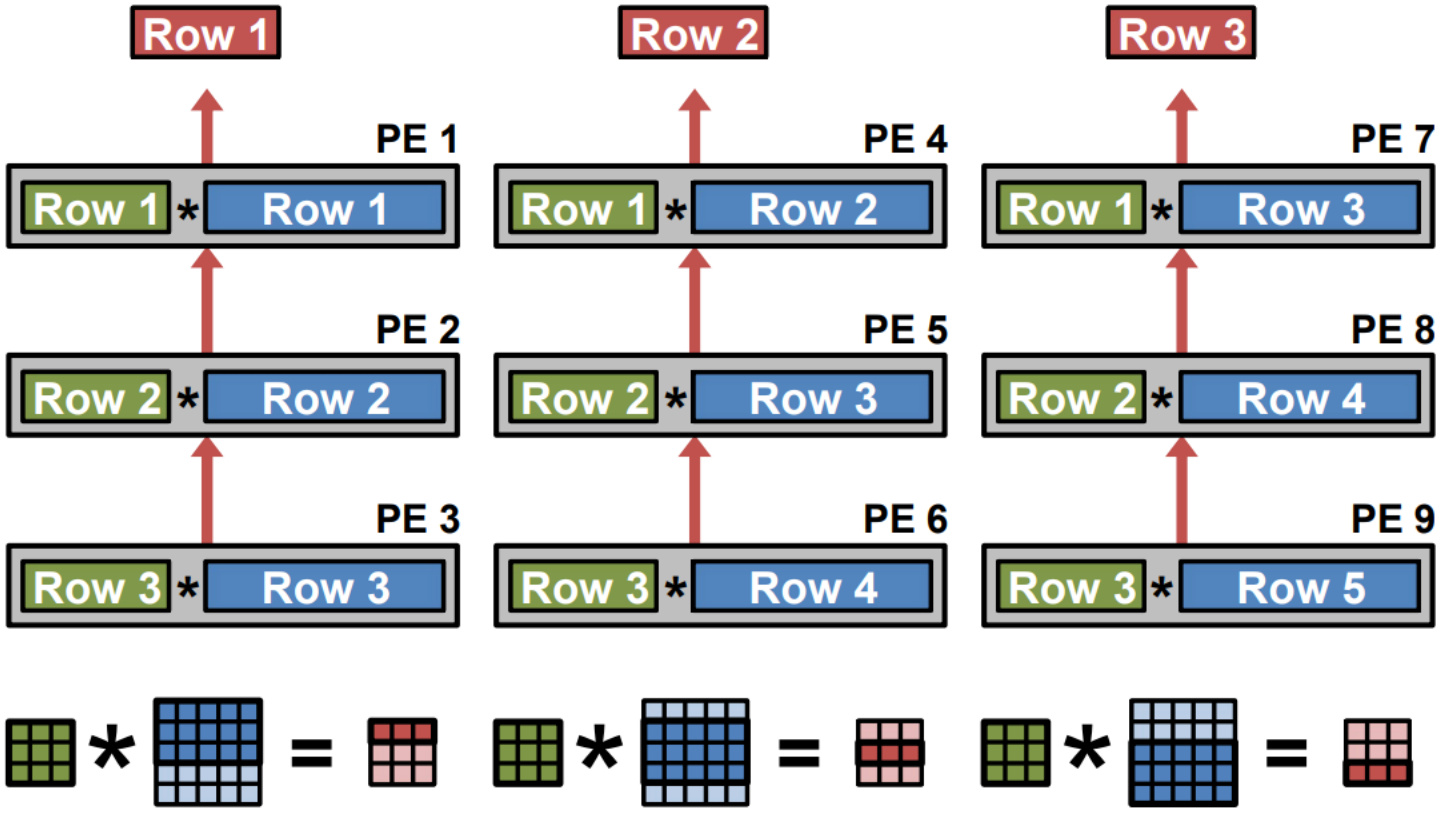


- Keep Filter row stationary
- Stream the IFMAP into PE

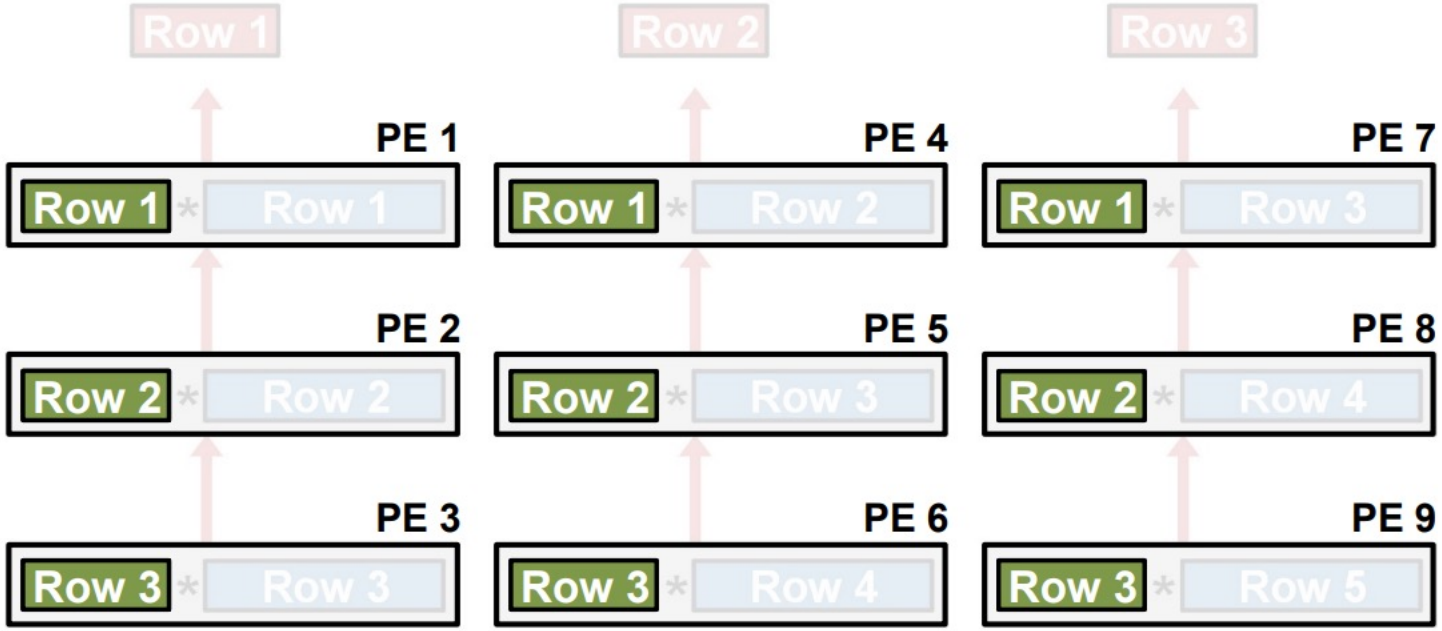
Row Stationary



Row Stationary – Multiple Rows

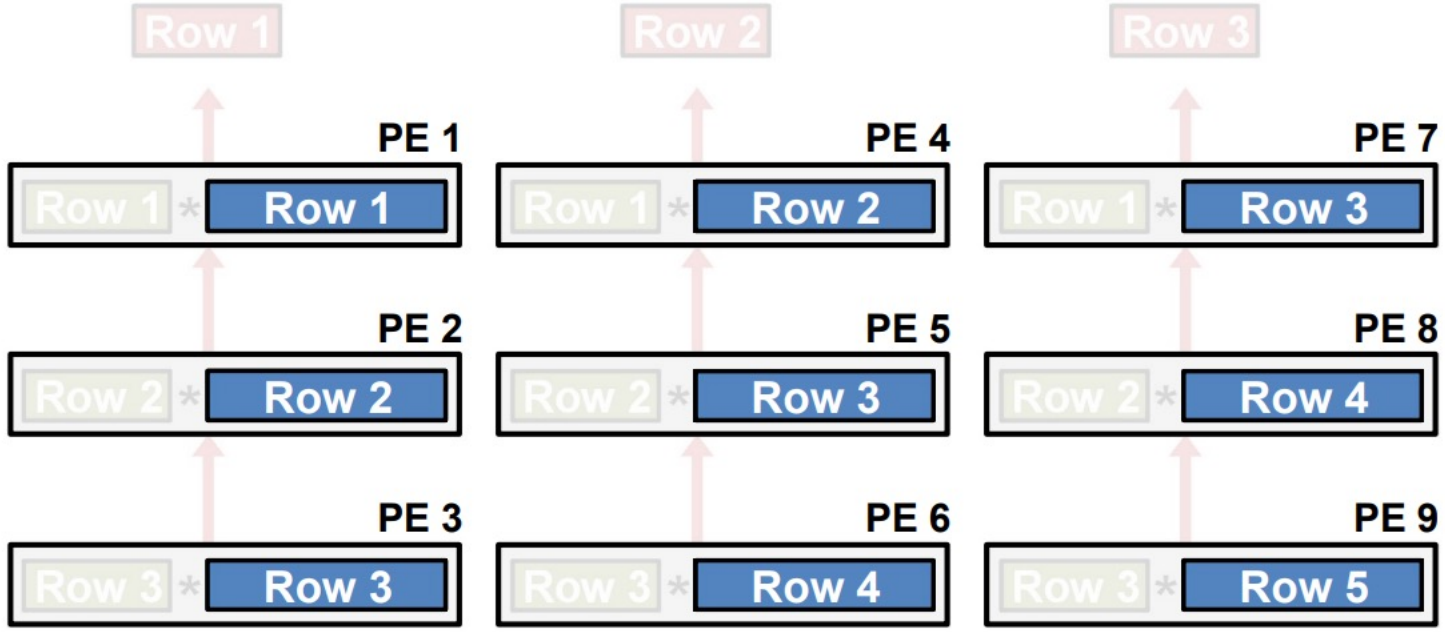


Row Stationary – Multiple Rows



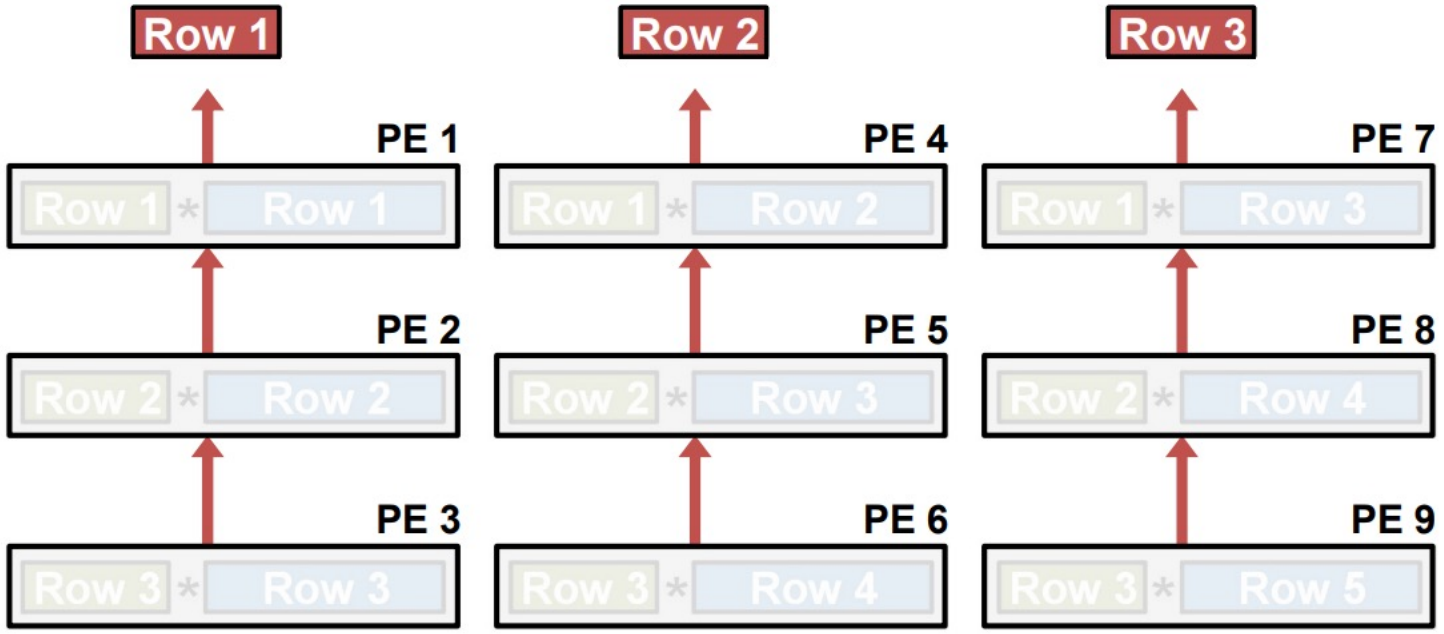
Filter rows are reused across PEs **horizontally**

Row Stationary – Multiple Rows



Fmap rows are reused across PEs **diagonally**

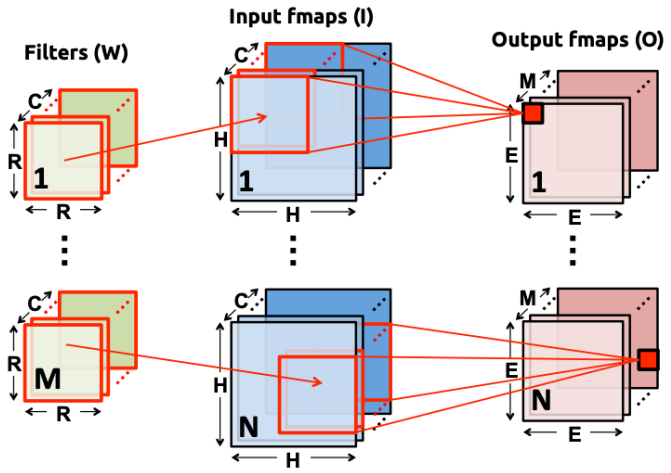
Row Stationary – Multiple Rows



Partial sums accumulate across PEs **vertically**

Comparison of Reuse in Different Dataflows

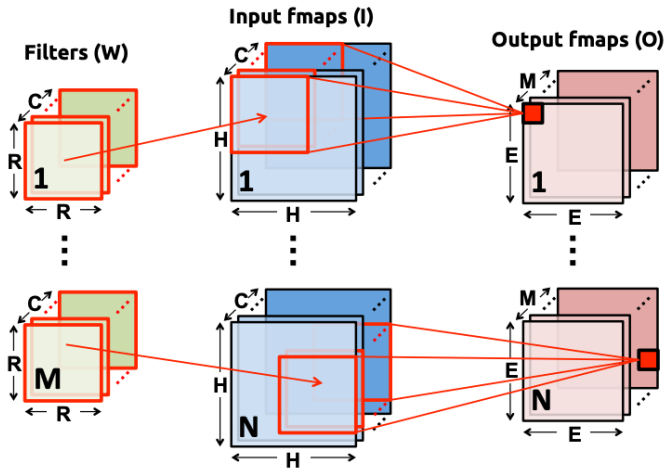
- **Convolutional reuse** (**Only CONV** layers):
  - A small amount of unique **input data** can be shared across many operations.
    - Each **filter weight** is reused E^2 times in the same ifmap plane.
    - Each **ifmap pixel**, i.e., **activation**, is usually reused R^2 times in the same filter plane.



Source: Chen, Sze and Emer, ISCA 2016

Comparison of Reuse in Different Dataflows

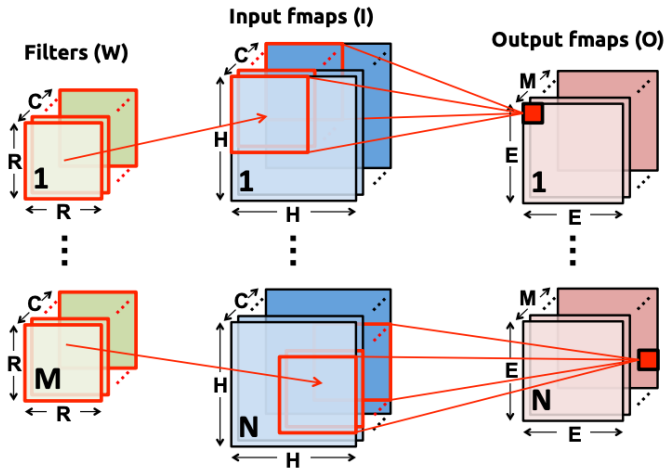
- **Convolutional reuse** (**Only CONV** layers):
  - A small amount of unique **input data** can be shared across many operations.
    - Each **filter weight** is reused E^2 times in the same ifmap plane.
    - Each **ifmap pixel**, i.e., **activation**, is usually reused R^2 times in the same filter plane.
- Filter reuse (both **CONV** and **FC** layers):
  - Each **filter weight** is further reused across the batch of **N ifmaps** in both CONV and FC layers.



Source: Chen, Sze and Emer, ISCA 2016

Comparison of Reuse in Different Dataflows

- **Convolutional reuse** (**Only CONV** layers):
  - A small amount of unique **input data** can be shared across many operations.
    - Each **filter weight** is reused E^2 times in the same ifmap plane.
    - Each **ifmap pixel**, i.e., **activation**, is usually reused R^2 times in the same filter plane.
- Filter reuse (both **CONV** and **FC** layers):
  - Each **filter weight** is further reused across the batch of **N ifmaps** in both CONV and FC layers.
- ifmap reuse (both **CONV** and **FC** layers):
  - Each **ifmap** pixel is further reused across **M** filters (to generate the **M** output channels).

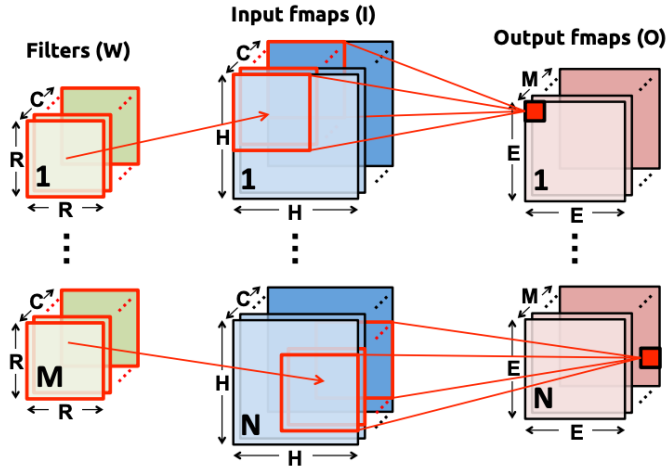


Source: Chen, Sze and Emer, ISCA 2016

Comparison of Reuse in Different Dataflows

- Weight Stationary (WS)
  - Each filter weight remains stationary in the RF to maximize **convolutional** and **filter** reuse.
  - Once a weight is fetched from DRAM to the RF, the PE runs all NE^2 operations that use the same filter weight.

$$O_{n,m,p,q} = \sum_{c,r,s} I_{n,c,Up+r,Uq+s} \times F_{m,c,r,s}$$

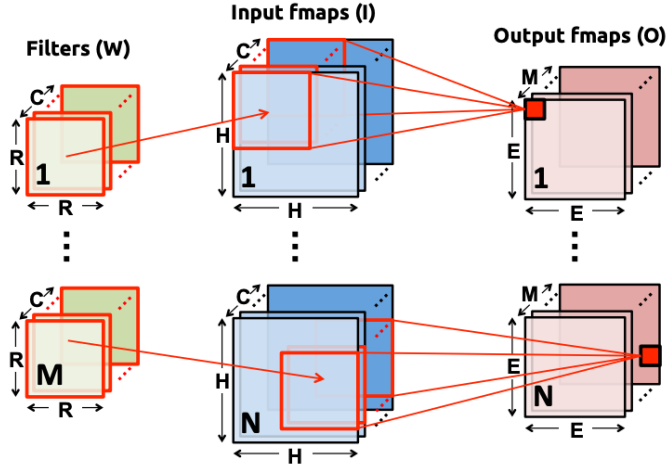


Source: Chen, Sze and Emer, ISCA 2016

Comparison of Reuse in Different Dataflows

- Output Stationary (OS)
  - The accumulation of each ofmap pixel stays stationary in a PE.
    - Multiple/Single ofmap channels (MOC) vs (SOC)
    - Multiple/Single ofmap-plane pixels (MOP) vs. (SOP)
  - There are three practical variants.

$$O_{n,m,p,q} = \sum_{c,r,s} I_{n,c,Up+r,Uq+s} \times F_{m,c,r,s}$$



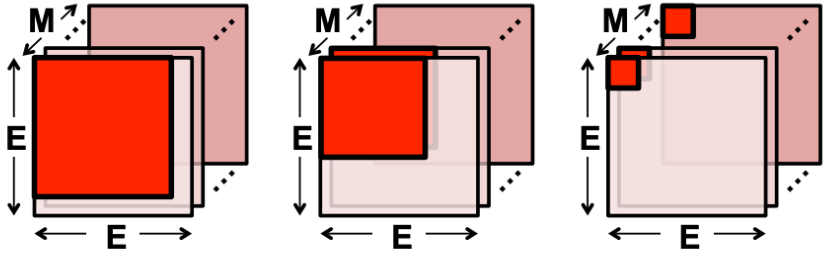


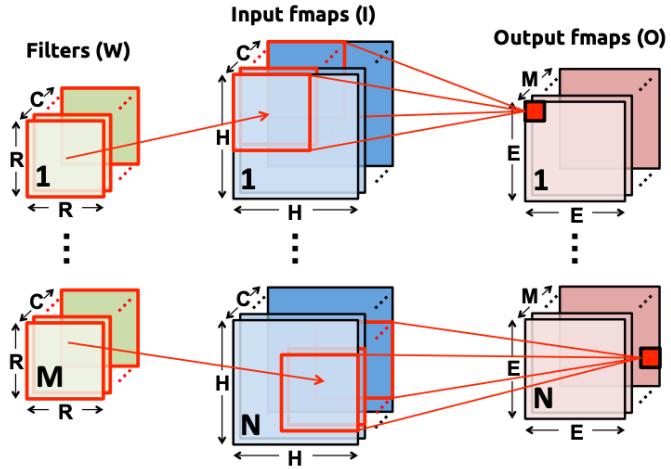
Figure 3. Comparison of the three different OS dataflow variants: (a) SOC-MOP, (b) MOC-MOP, and (c) MOC-SOP. The red blocks depict the ofmap region that the OS dataflow variants process at once.

Source: Chen, Sze and Emer, ISCA 2016

Comparison of Reuse in Different Dataflows

- No Local Reuse (NLR)
  - Does not exploit data reuse at the RF level
  - Uses inter-PE communication for ifmap reuse and psum accumulation.

$$O_{n,m,p,q} = \sum_{c,r,s} I_{n,c,Up+r,Uq+s} \times F_{m,c,r,s}$$

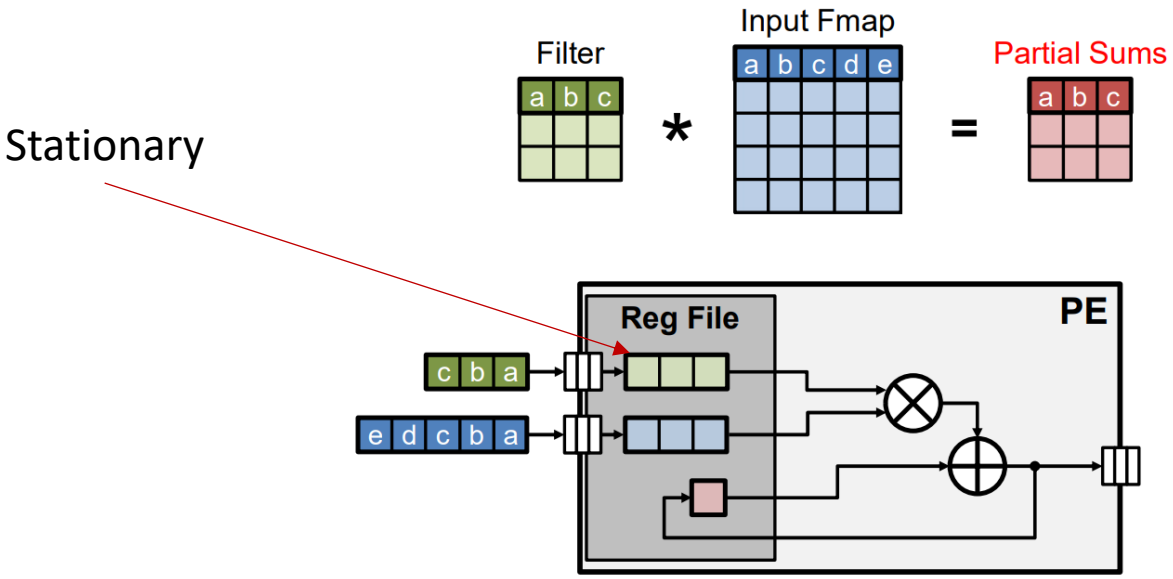


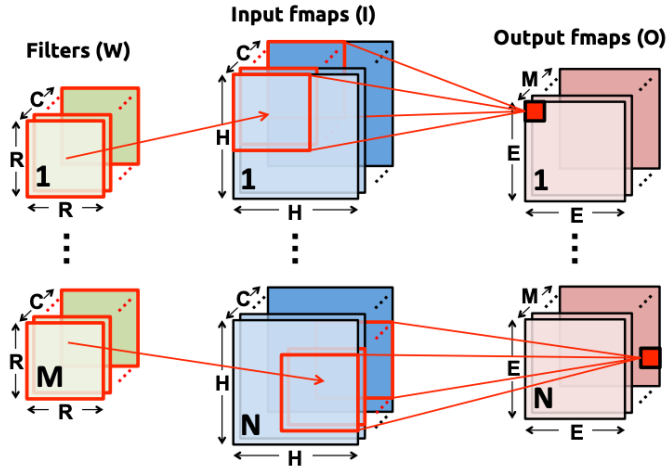
Source: Chen, Sze and Emer, ISCA 2016

Comparison of Reuse in Different Dataflows

- Row Stationary (RS)

$$O_{n,m,p,q} = \sum_{c,r,s} I_{n,c,Up+r,Uq+s} \times F_{m,c,r,s}$$

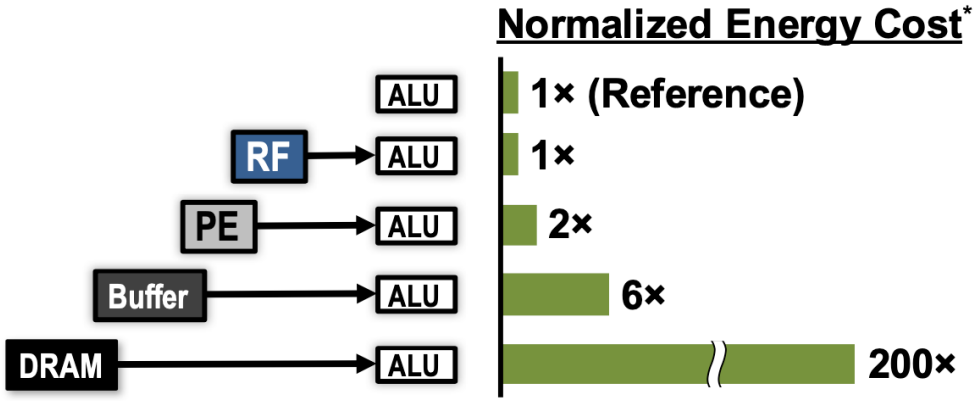




Source: Chen, Sze and Emer, ISCA 2016

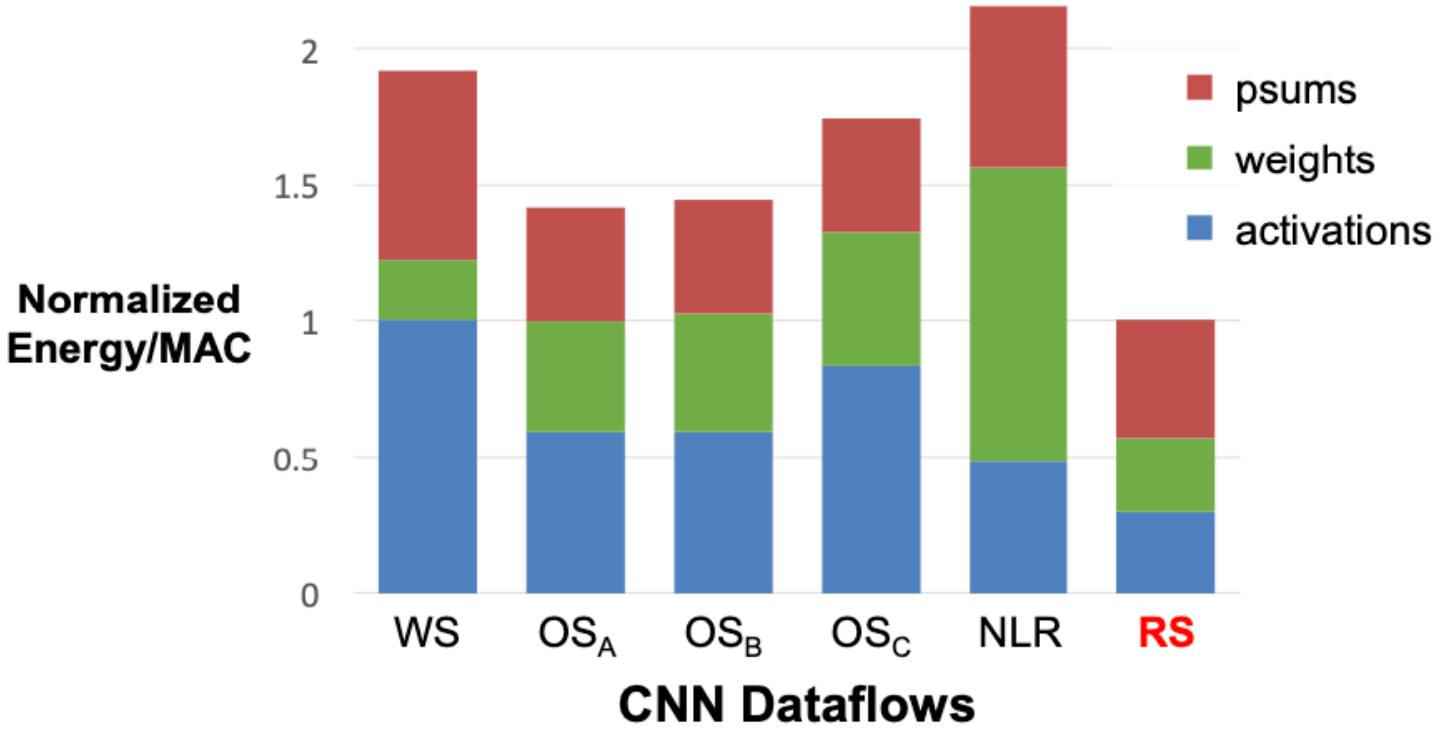
Comparison of Reuse in Different Dataflows

- Evaluation Setup
  - Same total area
  - 256 PEs
  - AlexNet
  - Batch size = 16



Source: Chen, Sze and Emer, ISCA 2016

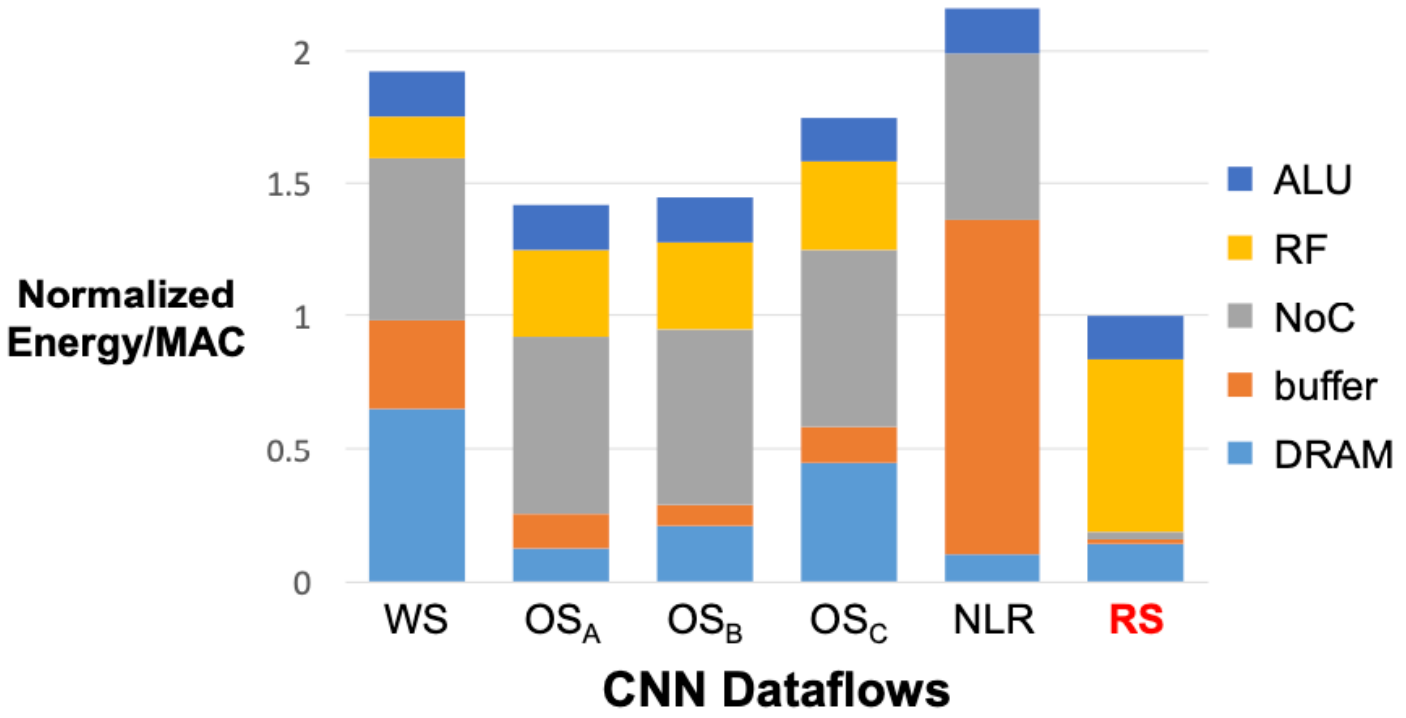
Comparison of Reuse in Different Dataflows (CONV Layers)



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

Source: Chen, Sze and Emer, ISCA 2016

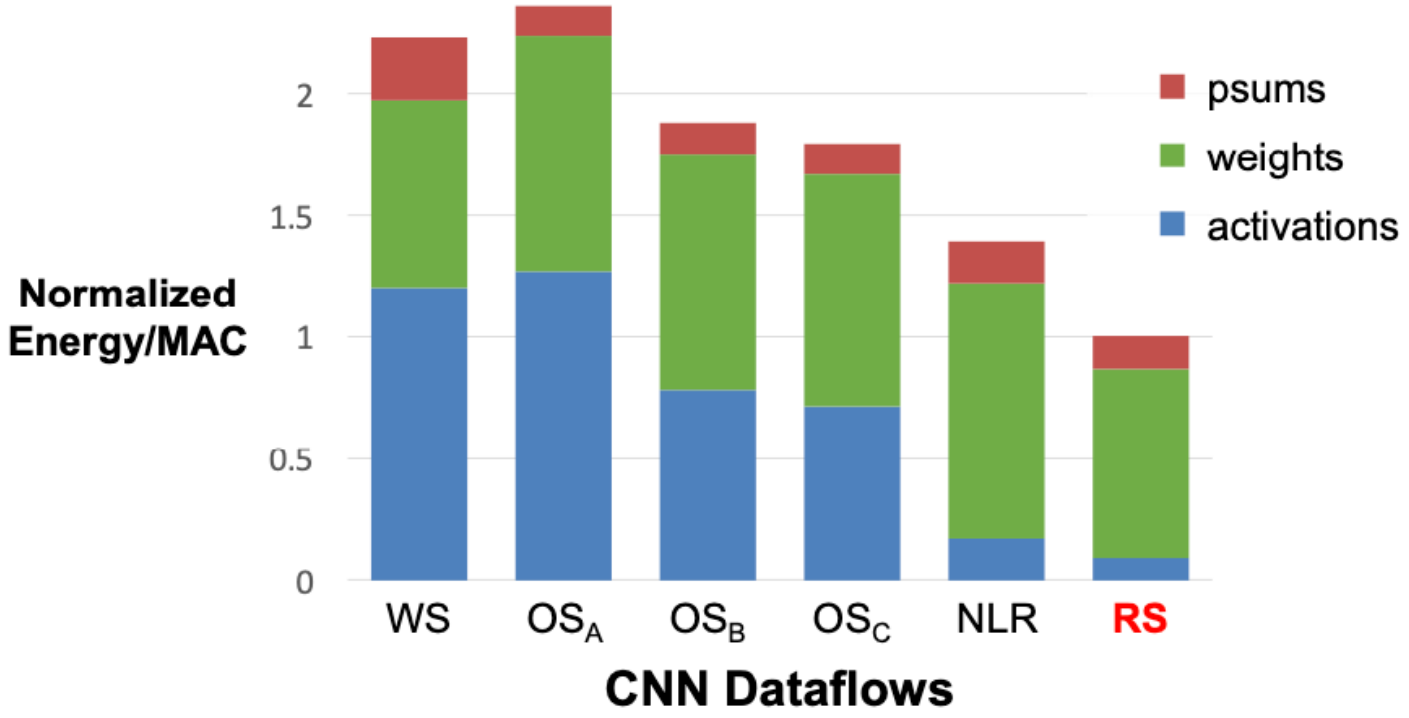
Comparison of Reuse in Different Dataflows (CONV Layers)



RS uses **1.4× – 2.5× lower** energy than other dataflows

Source: Chen, Sze and Emer, ISCA 2016

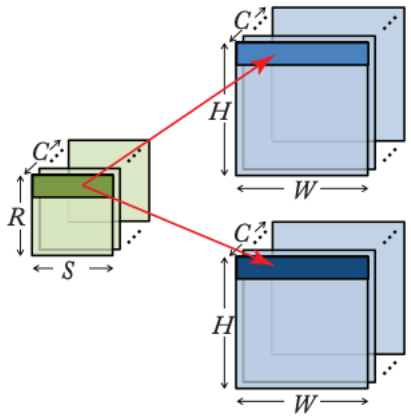
Comparison of Reuse in Different Dataflows (FC Layers)

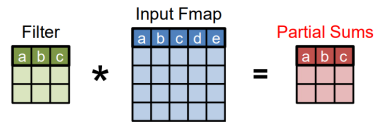


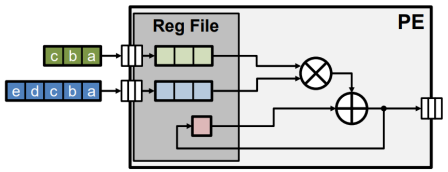
RS uses at least **1.3× lower** energy than other dataflows

Source: Chen, Sze and Emer, ISCA 2016

Row Stationary Higher than 2d Convolution

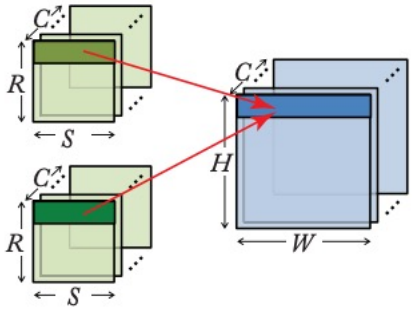


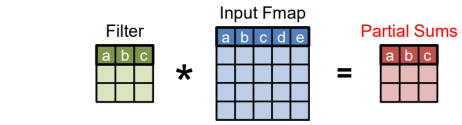


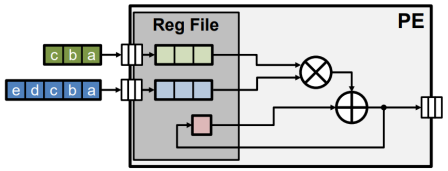


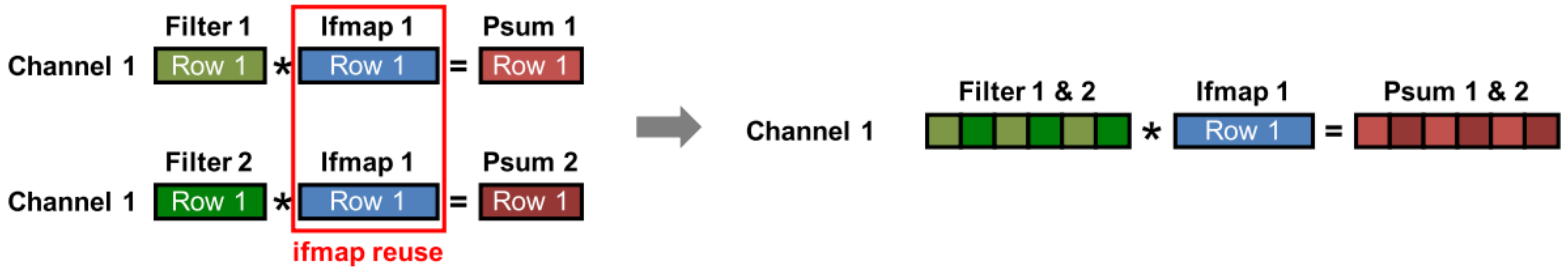
Multiple fmaps

Row Stationary Higher than 2d Convolution



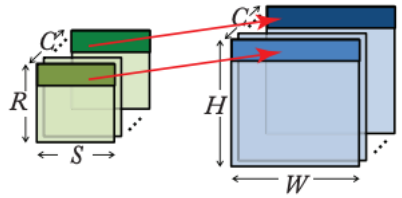


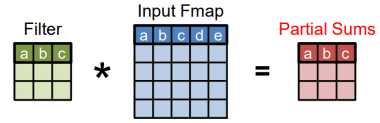


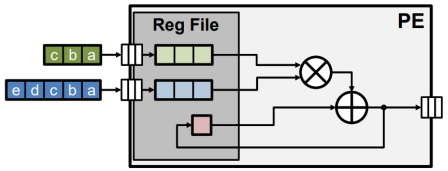


Multiple filters

Row Stationary Higher than 2d Convolution

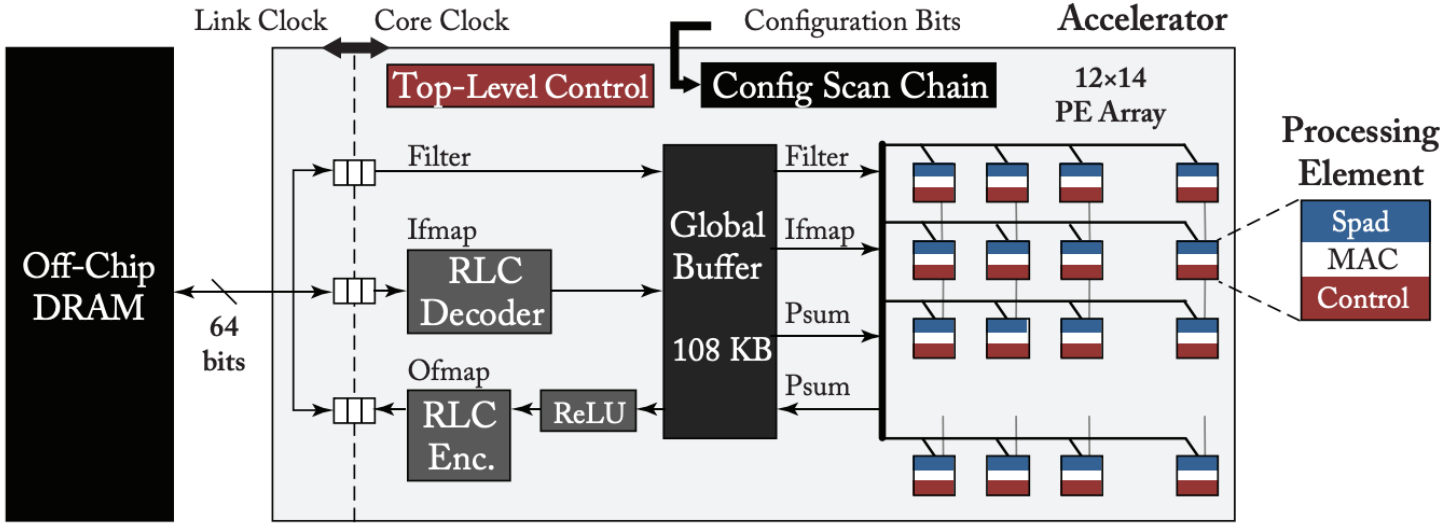


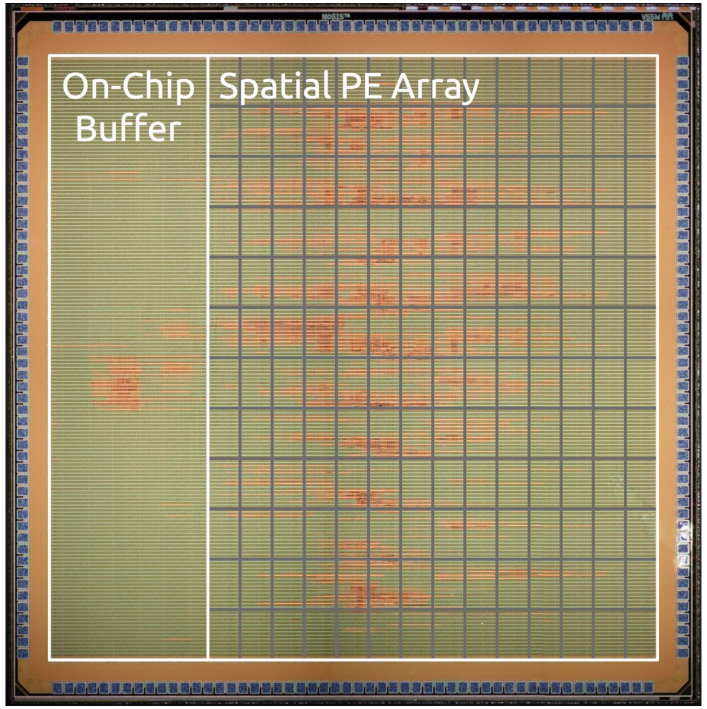




Multiple channels

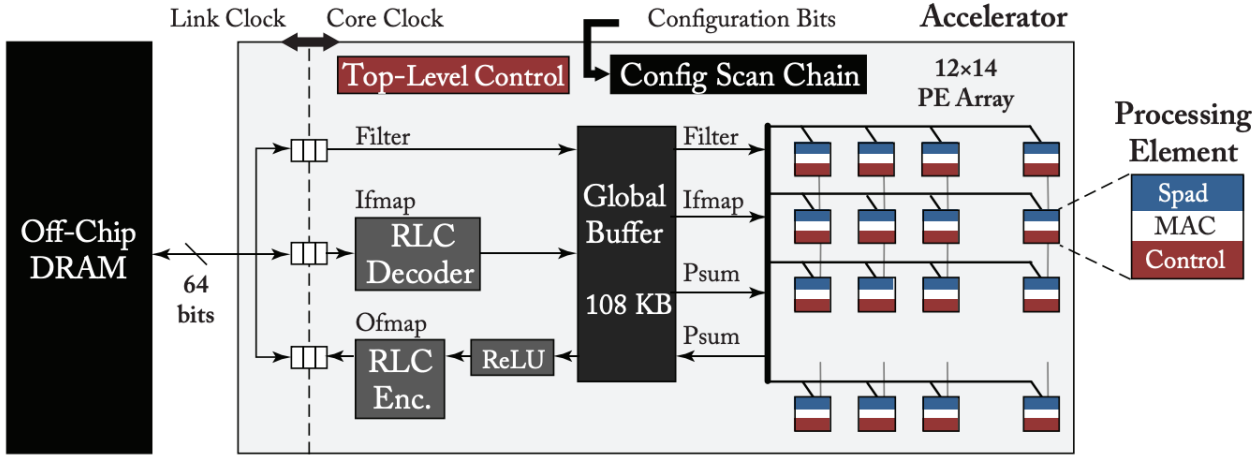
Row Stationary Example: Eyeriss

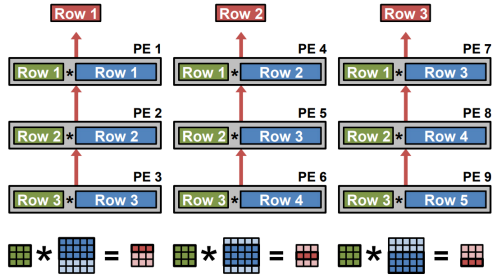


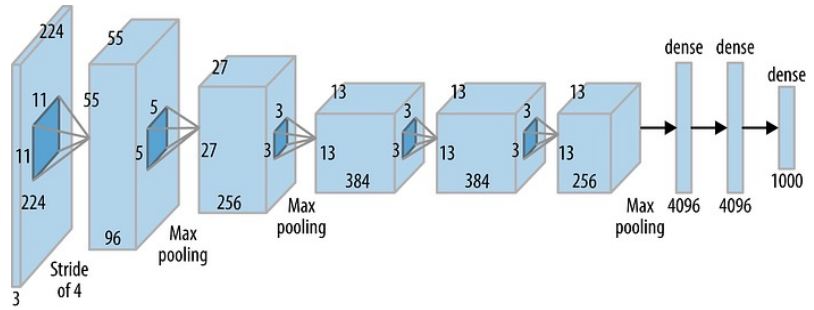


- An energy-efficient deep convolutional neural network (CNN) accelerator.
- **Row-Stationary (RS) Dataflow**: Designed to minimize data movement, optimizing energy efficiency.
- **Spatial Architecture**: Parallel processing with an array of PEs.
- **On-chip Global Buffer**: Reduces energy-consuming off-chip memory access.
- **Dynamic-Configurability**: Each PE can adapt to different layer parameters in a CNN.

Row Stationary Example: Eyeriss







- The third to fifth layers of AlexNet, each 2-D convolution only uses a 13x3 PE array
- The second layer of AlexNet, it requires a 27x5 PE array to complete the 2-D convolution