TrIM: An Efficient Systolic Array for Convolutional Neural Networks

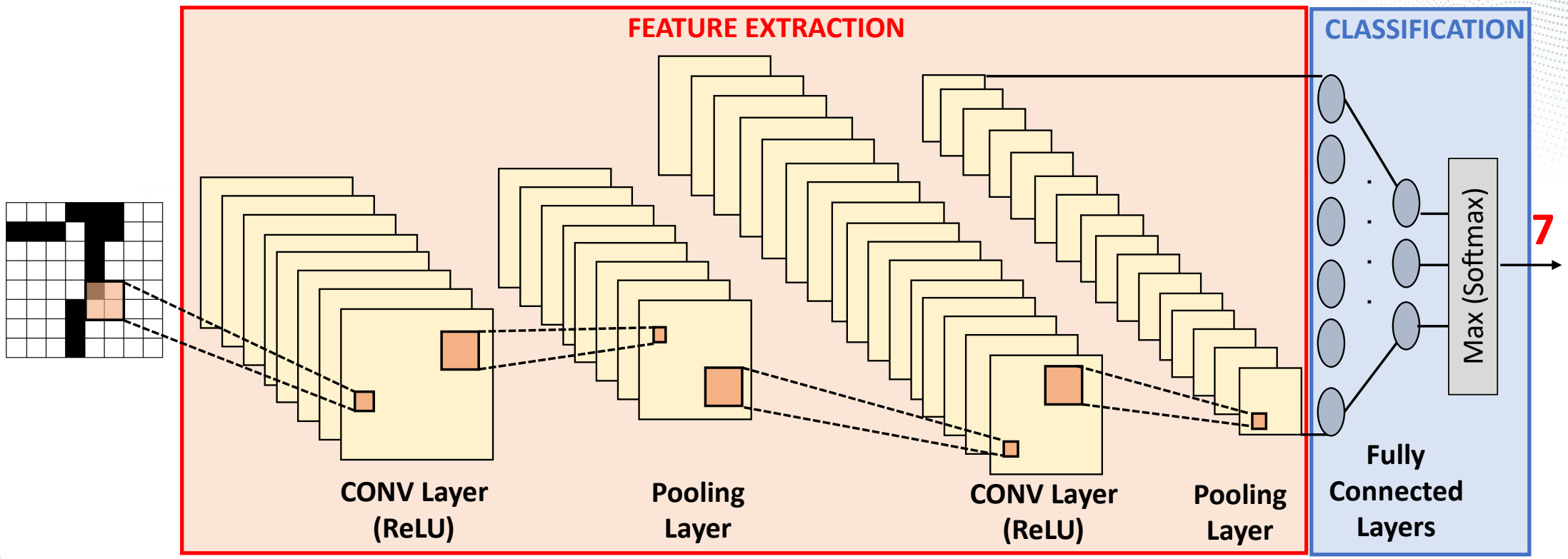
Dr Cristian Sestito
Research Fellow, APRIL AI Hub
School of Engineering, The University of Edinburgh
csestito@ed.ac.uk

Verification Futures Conference 2025
Track: Breakthrough Technologies
1st July 2025, University of Reading, UK

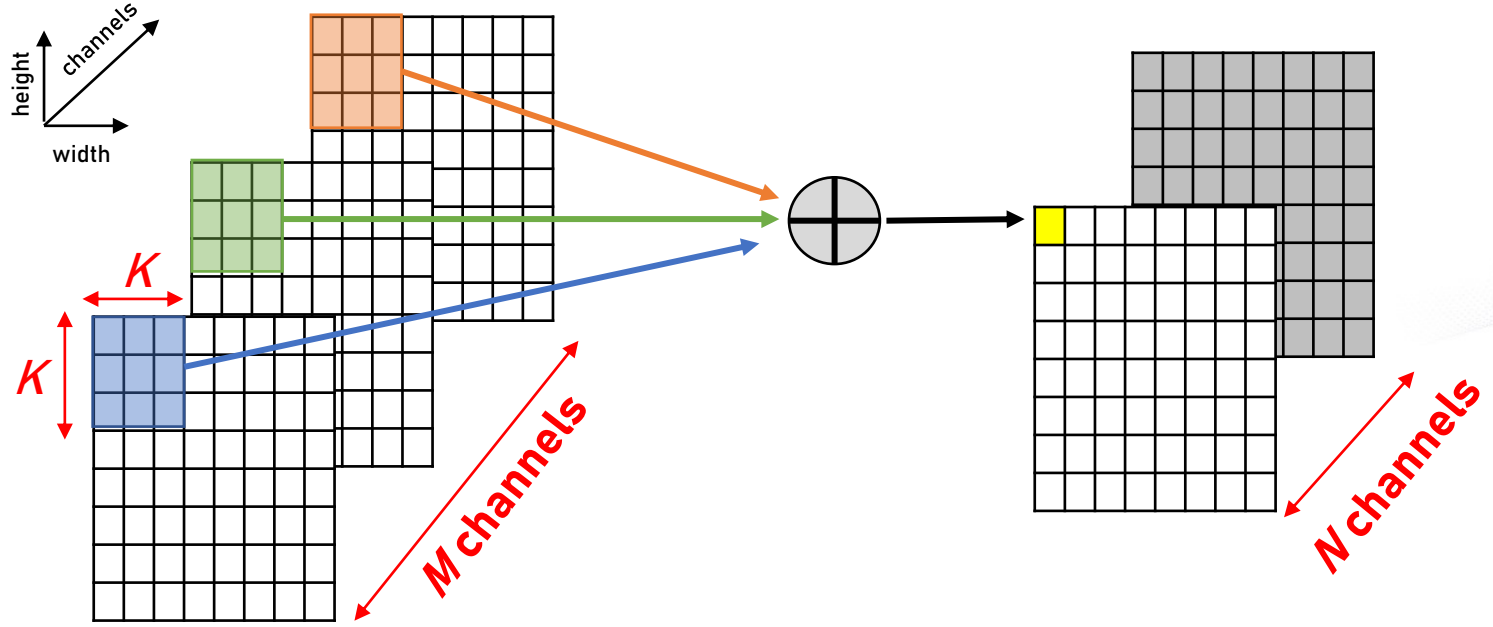
Agenda

Convolutional Neural Networks (CNNs)	• Basics • Applications
Hardware Architectures for CNNs	• Compute Architectures • Systolic Arrays
TrIM: Triangular Input Movement Systolic Array	• Dataflow • Architecture • Results

Convolutional Neural Networks (CNNs)



Convolutional Layers



Input feature maps (*ifmaps*)

Output feature maps (*ofmaps*)

Convolutional layers use **3D filters** (made of **multiple kernels**) to generate ofmaps

Applications

Computer Vision

- Image classification
- Image segmentation
- Face recognition
- Medical imaging
- Satellite imaging

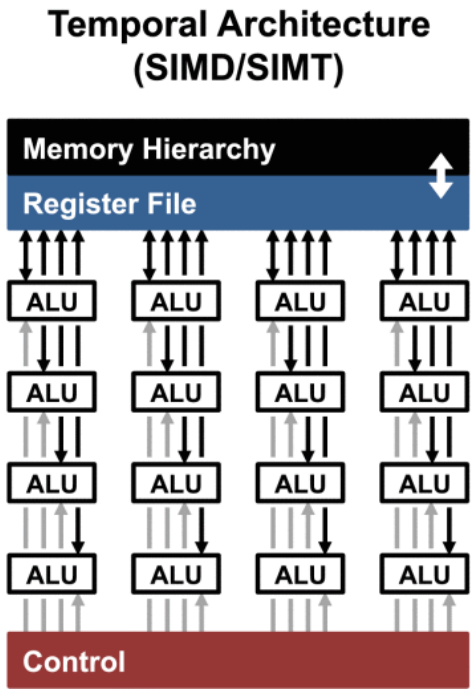
Autonomous Systems

- Vehicles
  - Lane detection
  - Traffic sign recognition
- Robots
  - Path planning
  - Object detection

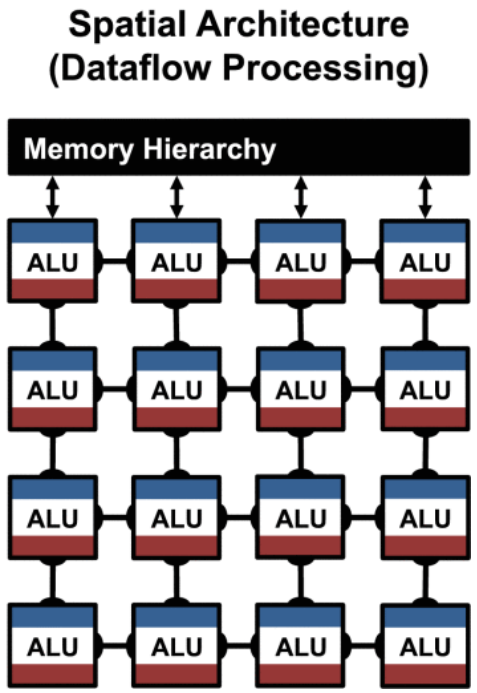
Others

- Time series analysis
- Recommendation systems
- Drug discovery
- Style transfer
- Defect detection

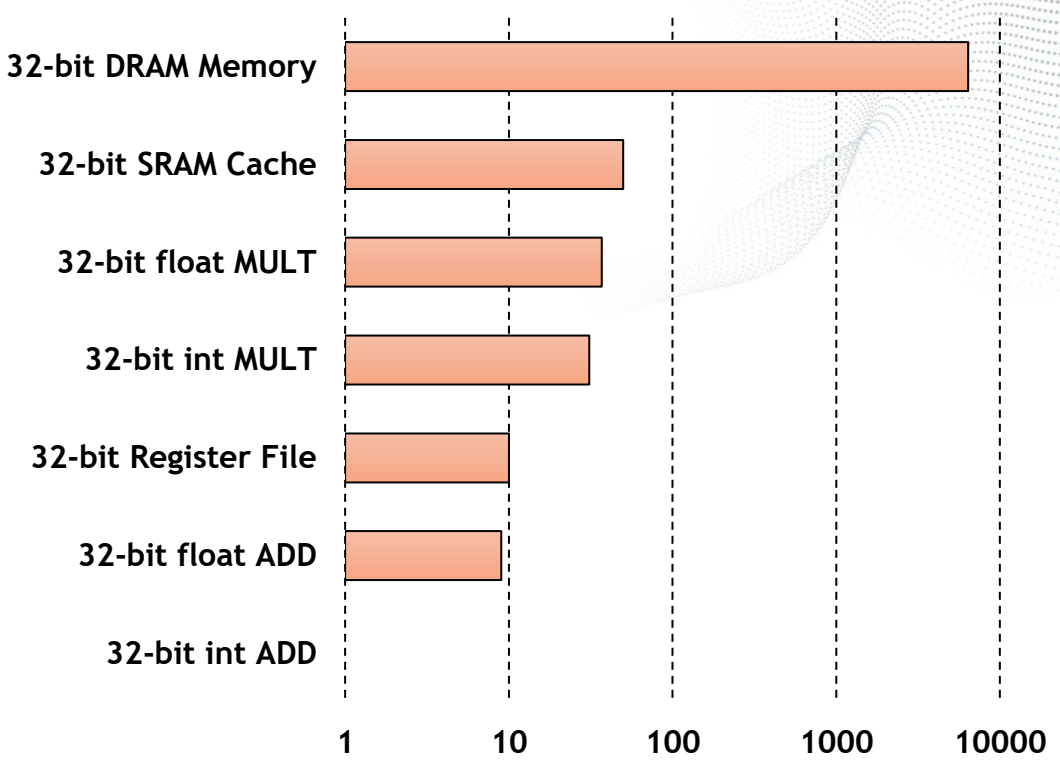
Compute Architectures



CPUs, GPUs



Systolic arrays

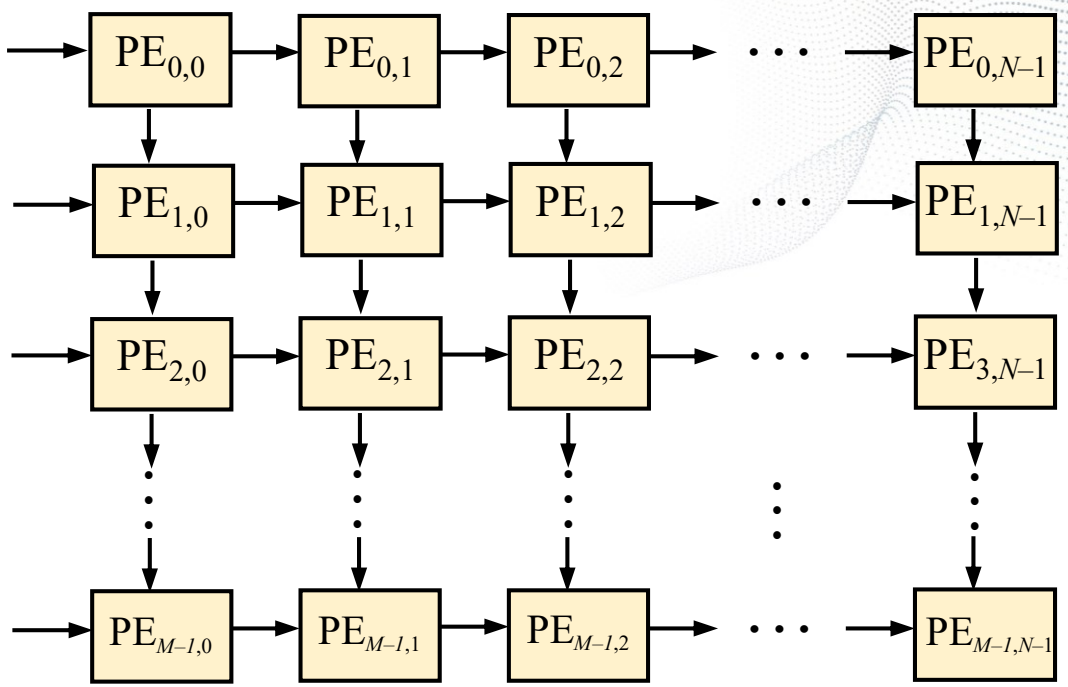


Relative Energy Cost (45nm, 0.9 V). From Horowitz, ISSCC 2014: https://doi.org/10.1109/ISSCC.2014.6757323

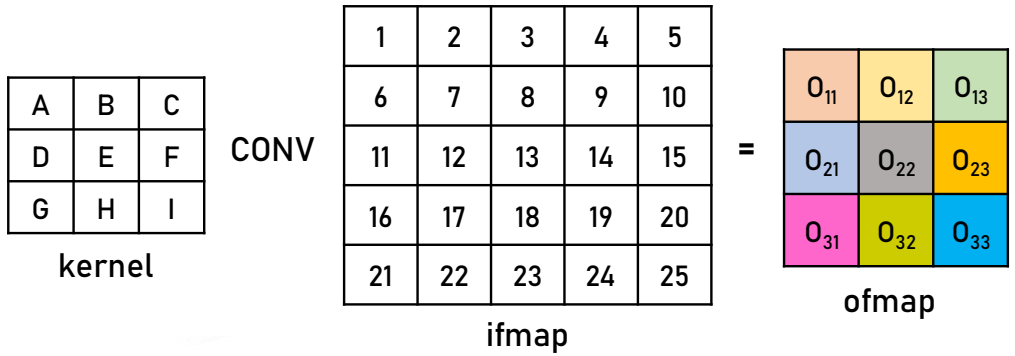
Sze et al., Proc. IEEE 2017: https://doi.org/10.1109/JPROC.2017.2761740

Systolic Arrays

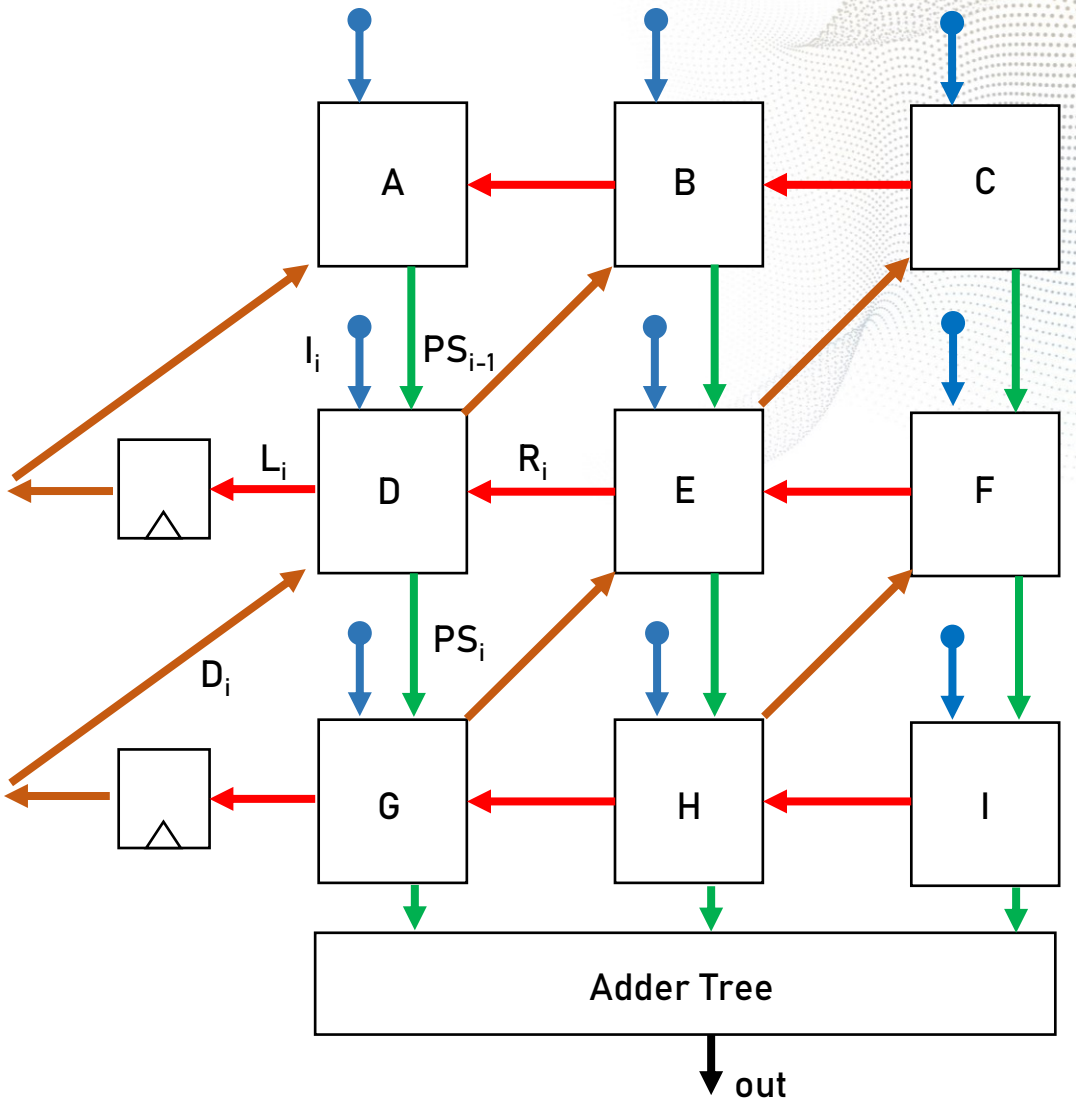
- **1978**: **H. T. Kung** coined the term systolic array
  - Class of **parallel computing architectures**
  - Data moves **synchronously** by using interconnected **processing elements**
- **2015**: **Google** introduces the **Tensor Processing Unit** (TPU) that uses systolic arrays to **accelerate neural network computations**
- **Dataflows:** Weight stationary, output stationary, input stationary, row stationary



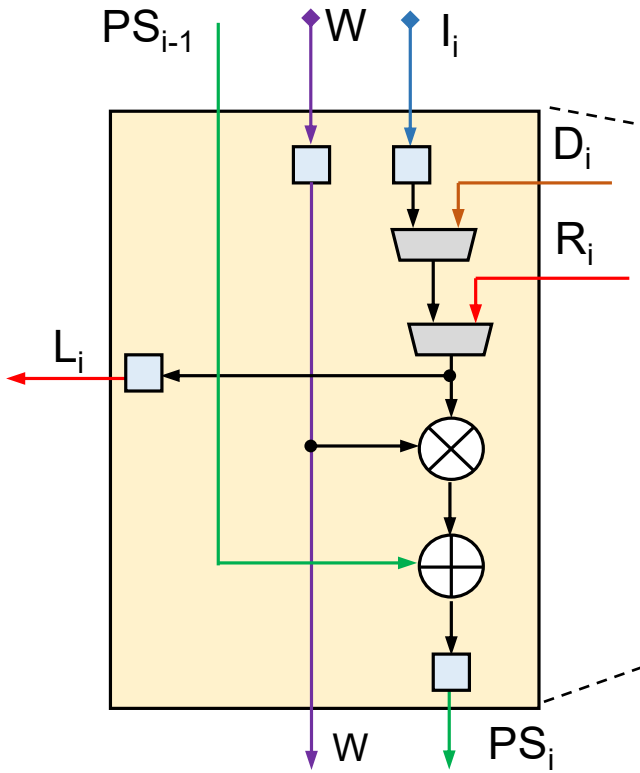
TrIM: The Dataflow

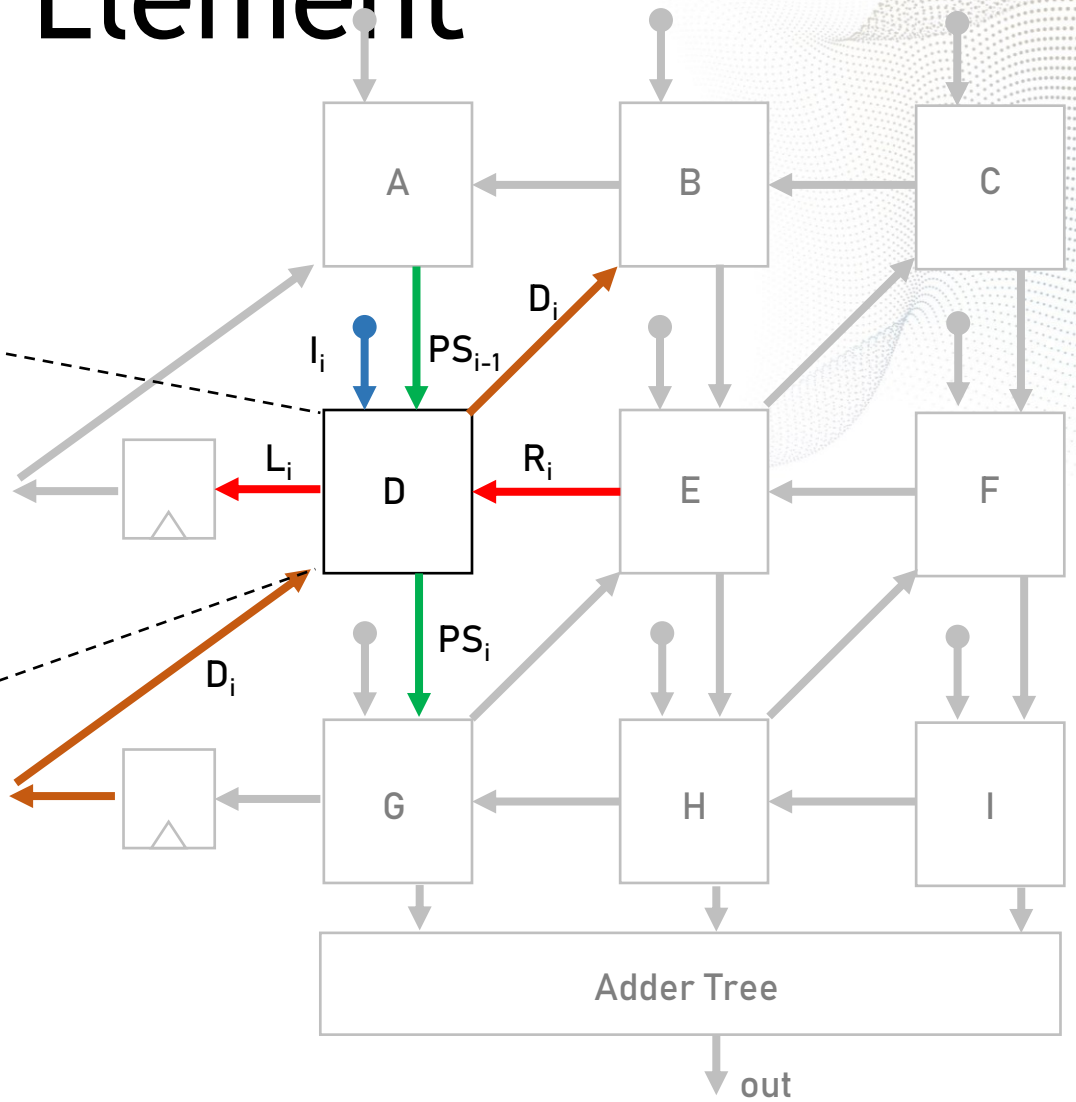


1. Kernel weights are loaded and hold vertically
2. Ifmap activations are loaded vertically from the memory
3. Ifmap activations move from right to left, and from left to up (diagonally) to form a triangle
4. Partial sums move vertically



TrIM: The Processing Element





TrIM: Example

A	B	C
D	E	F
G	H	I

kernel

CONV

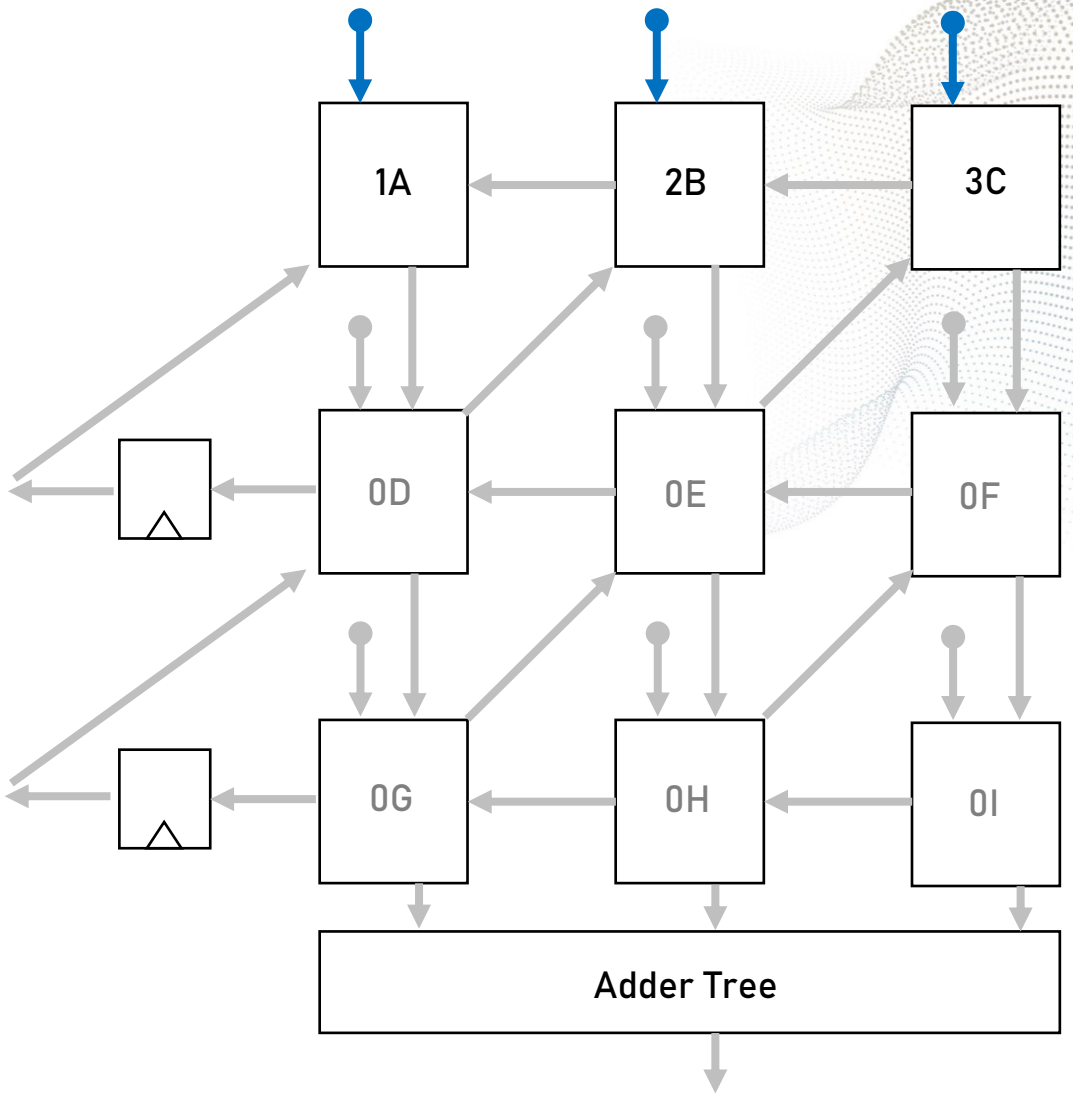
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

ifmap

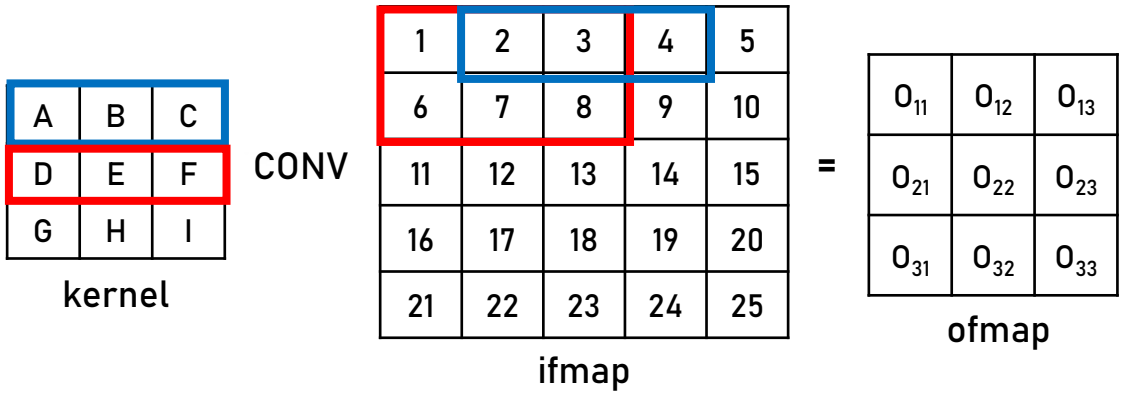
=

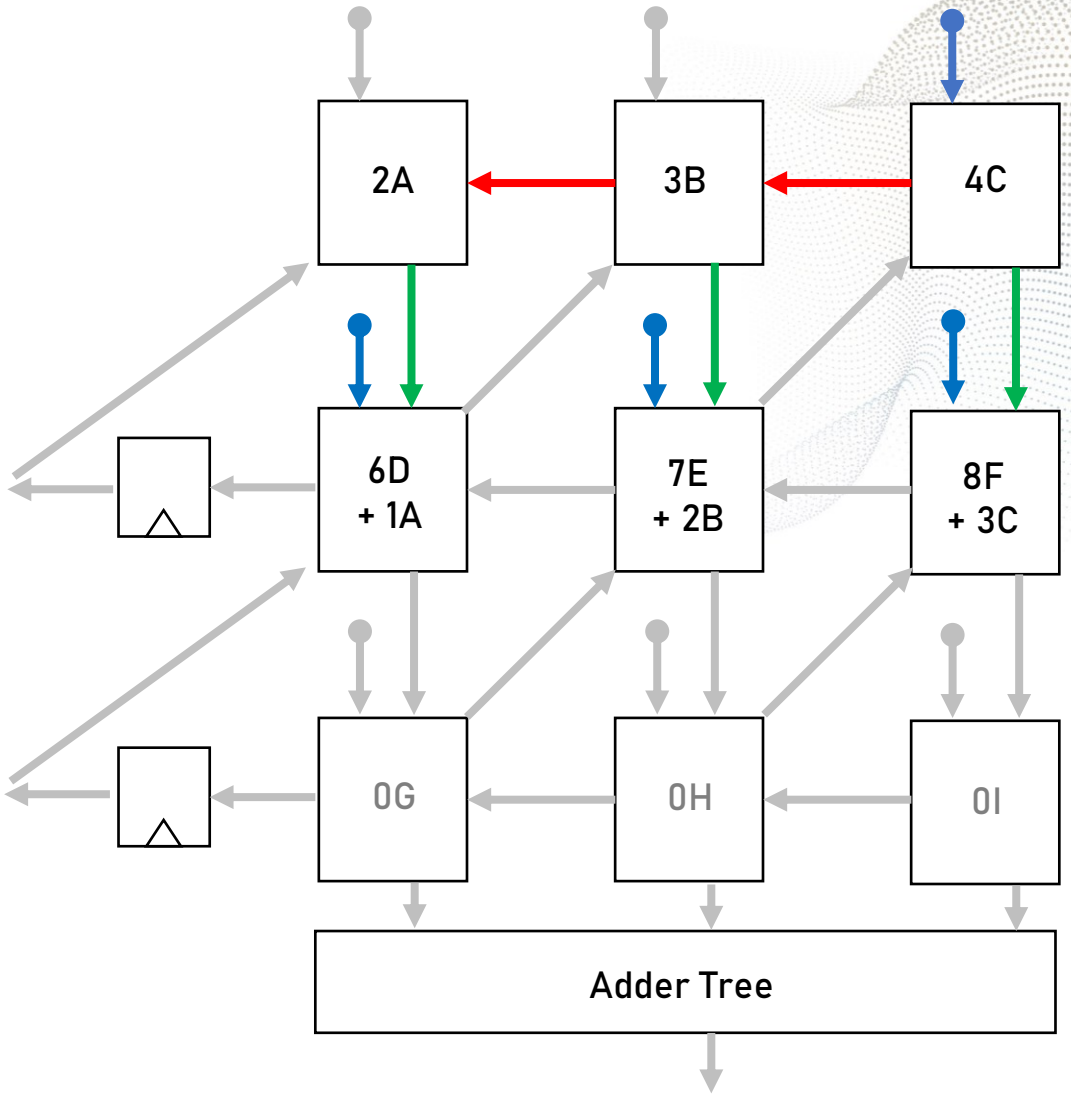
O_{11}	O_{12}	O_{13}
O_{21}	O_{22}	O_{23}
O_{31}	O_{32}	O_{33}

ofmap

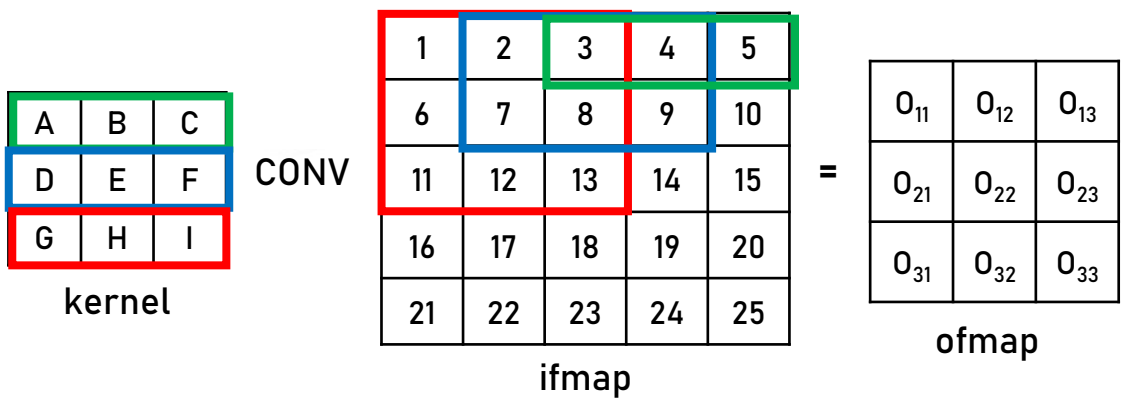


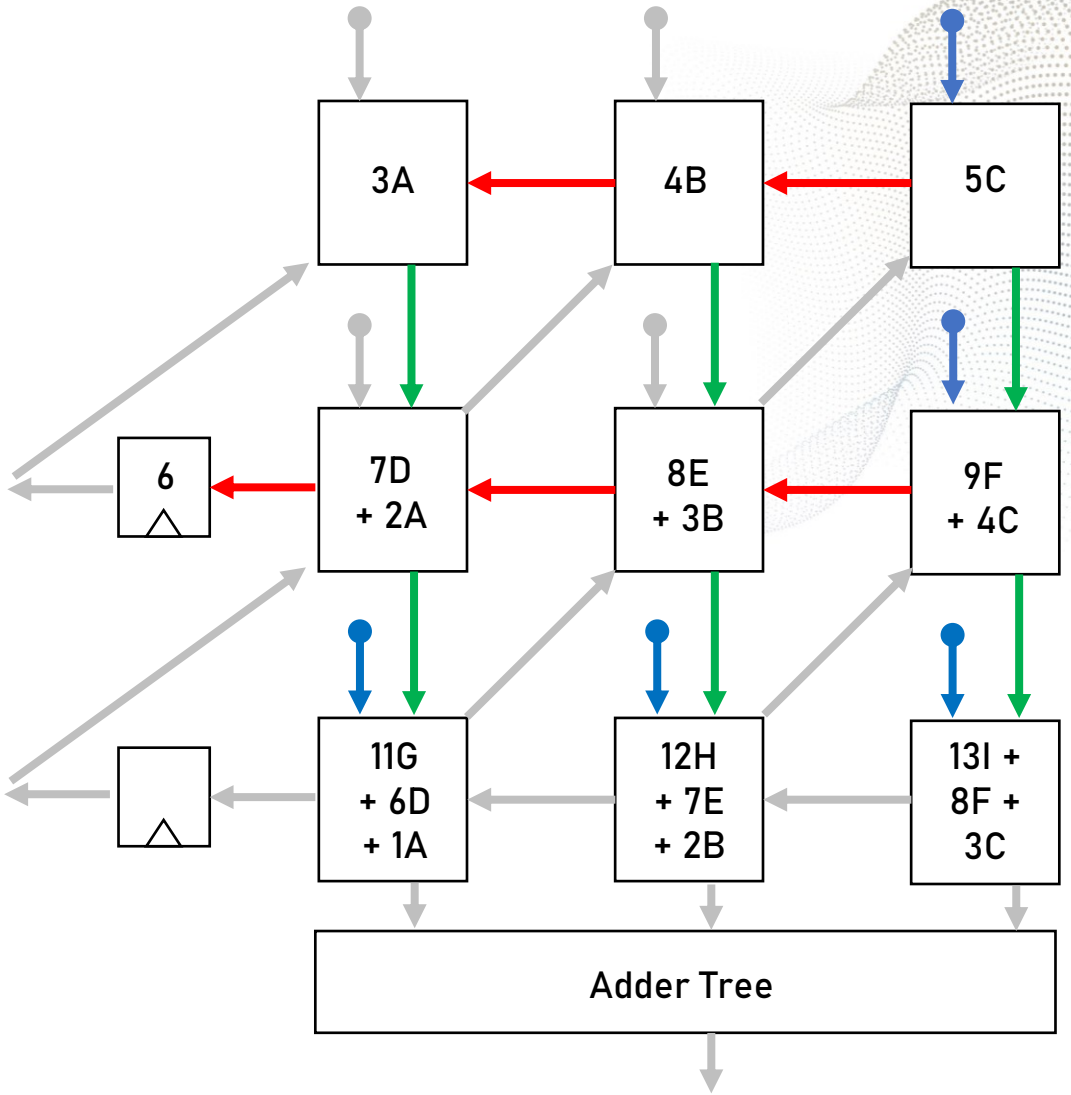
TrIM: Example



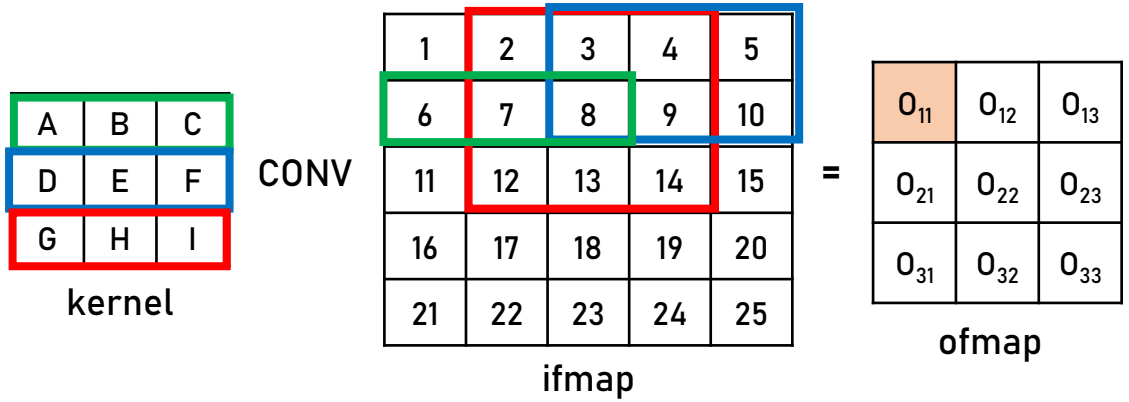


TrIM: Example

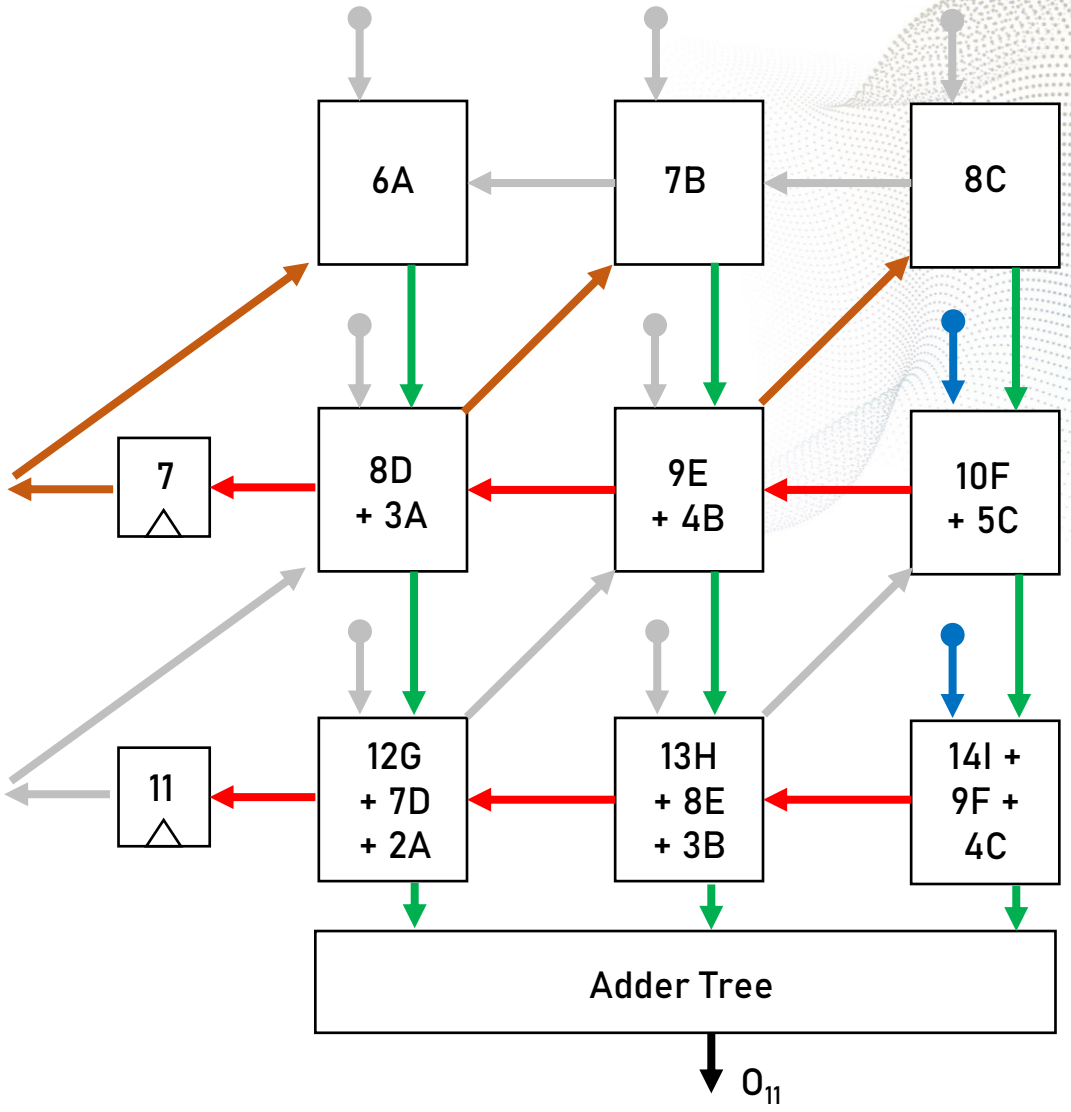




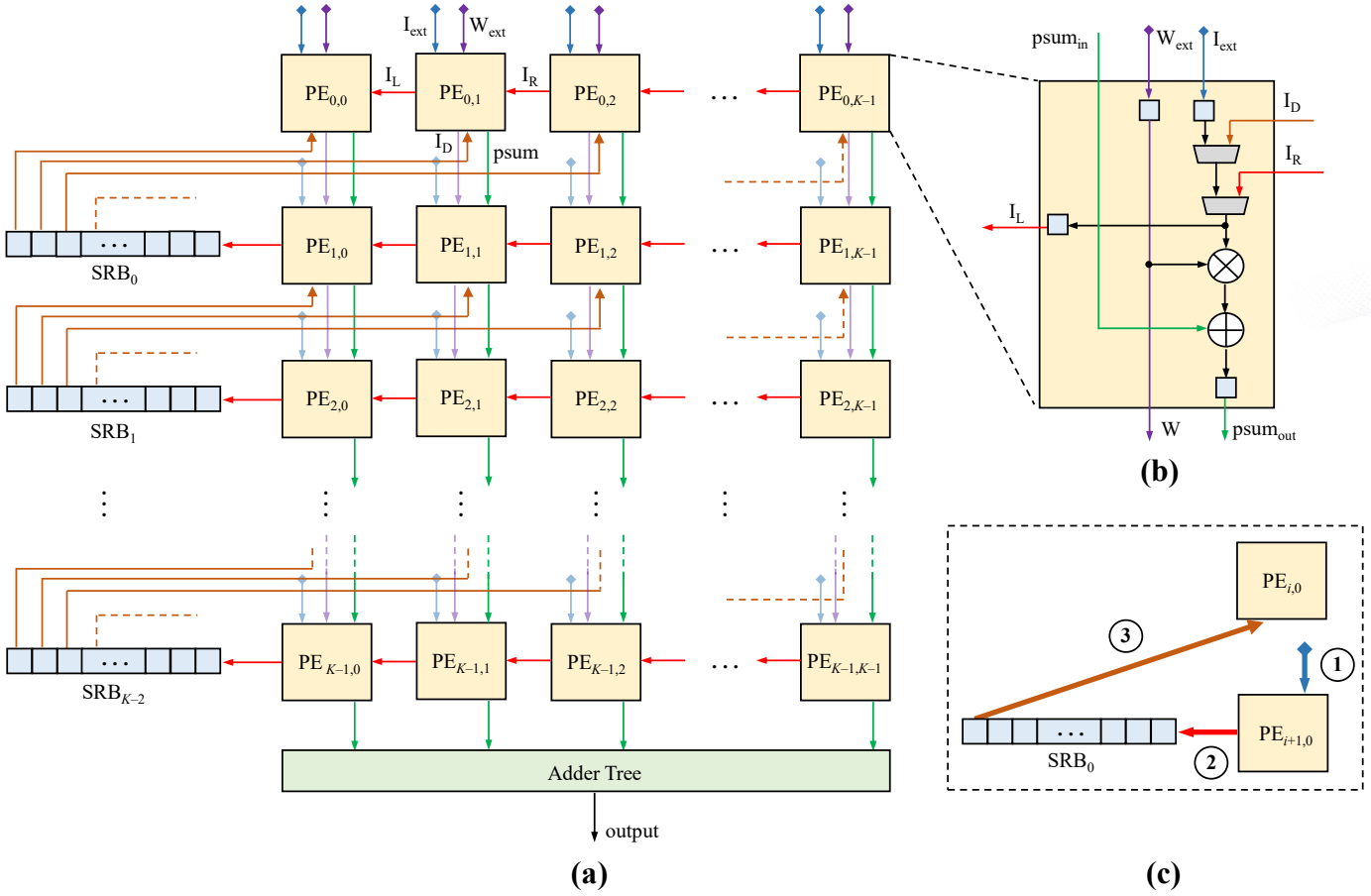
TrIM: Example



and so on...

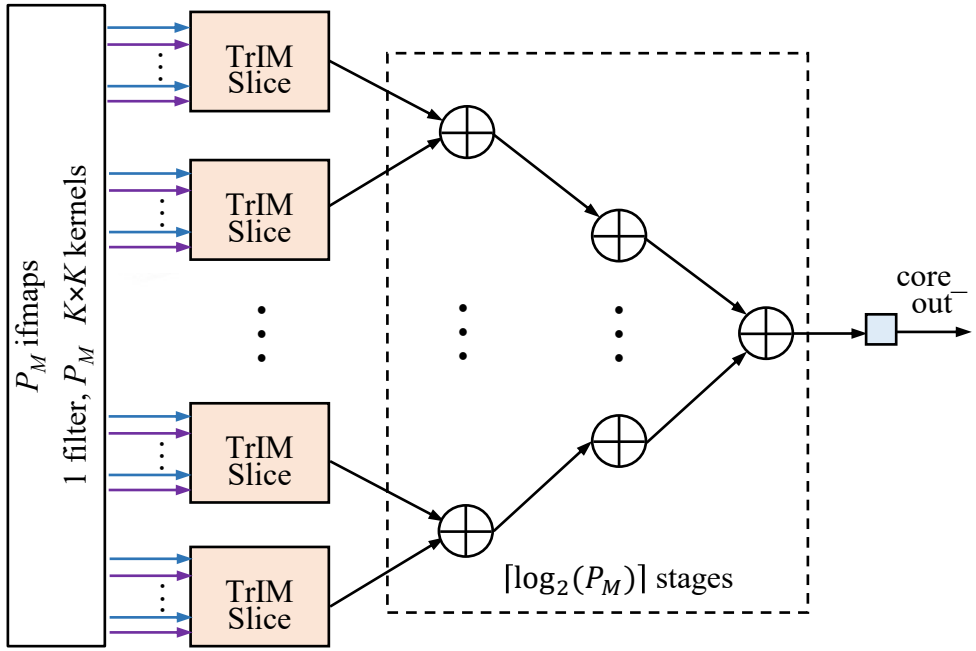


TrIM: The Generalised Architecture

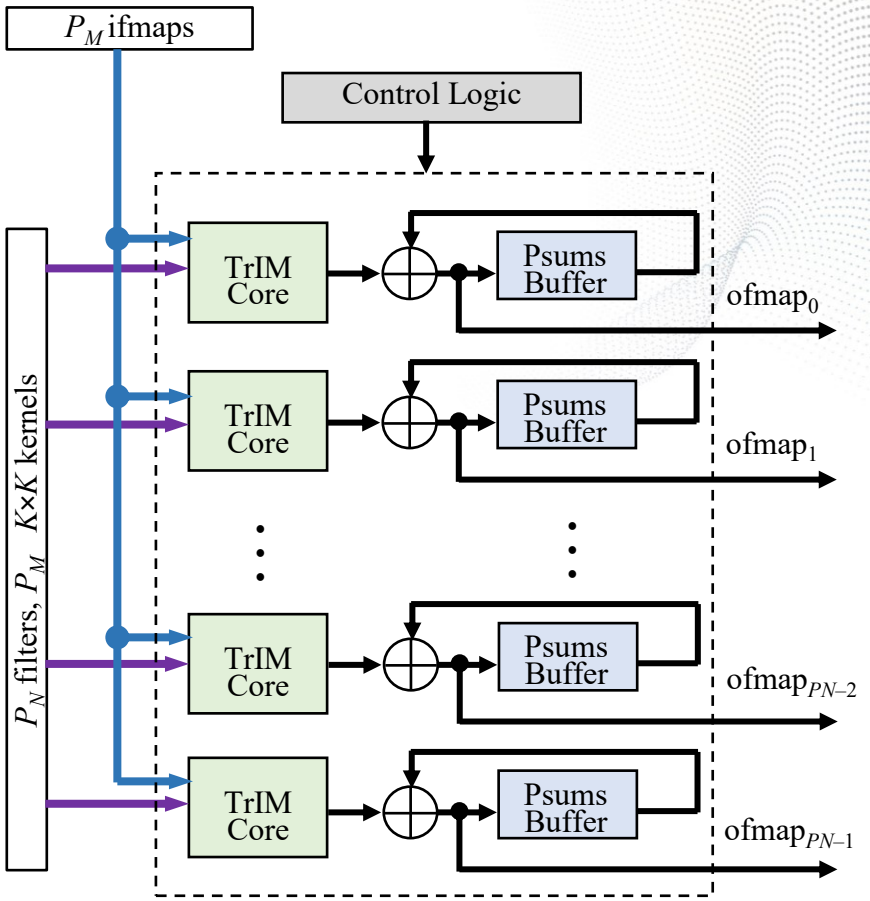


Sestito et al., ArXiv 2024: https://doi.org/10.48550/arXiv.2408.01254

TrIM: Managing Convolutional Layers



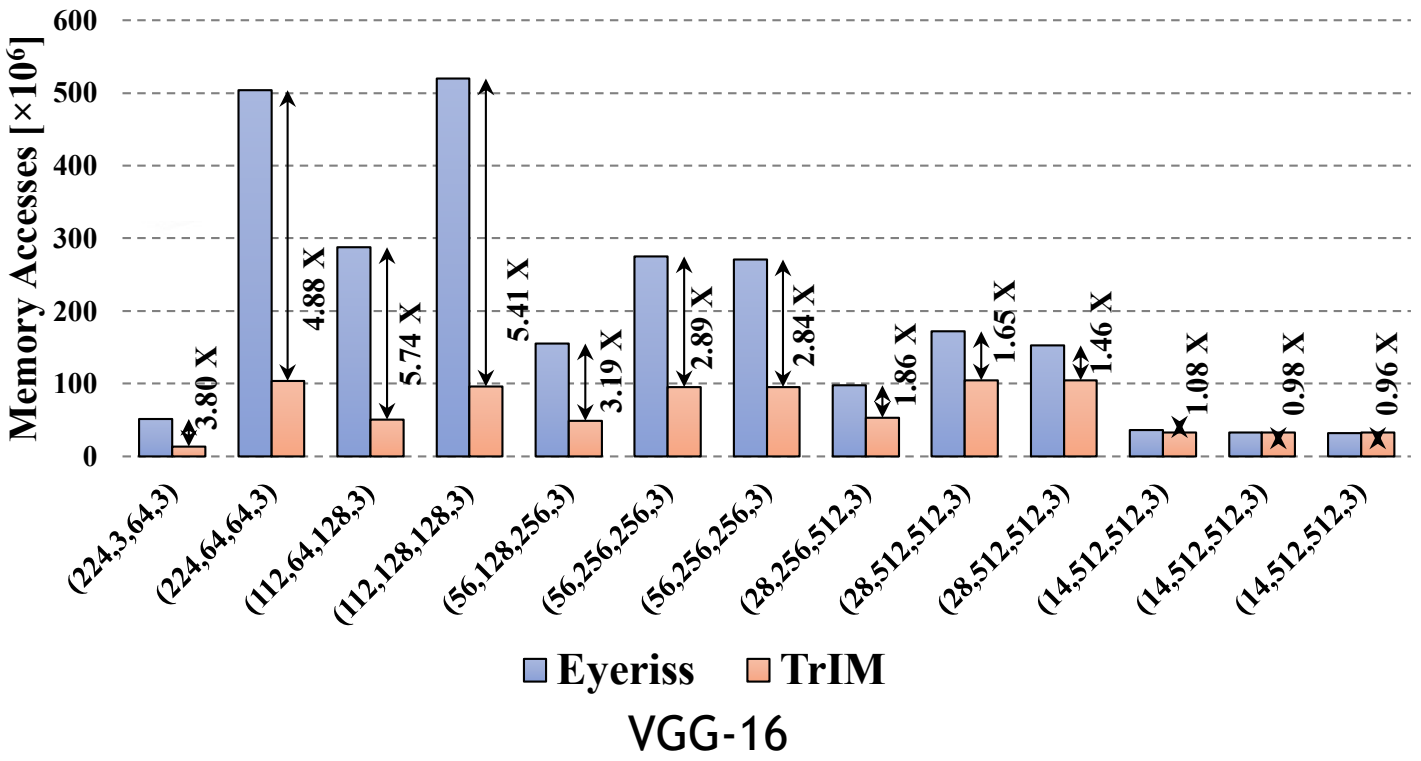
Different **slices** relate to different **kernels** of the **same 3D filter**



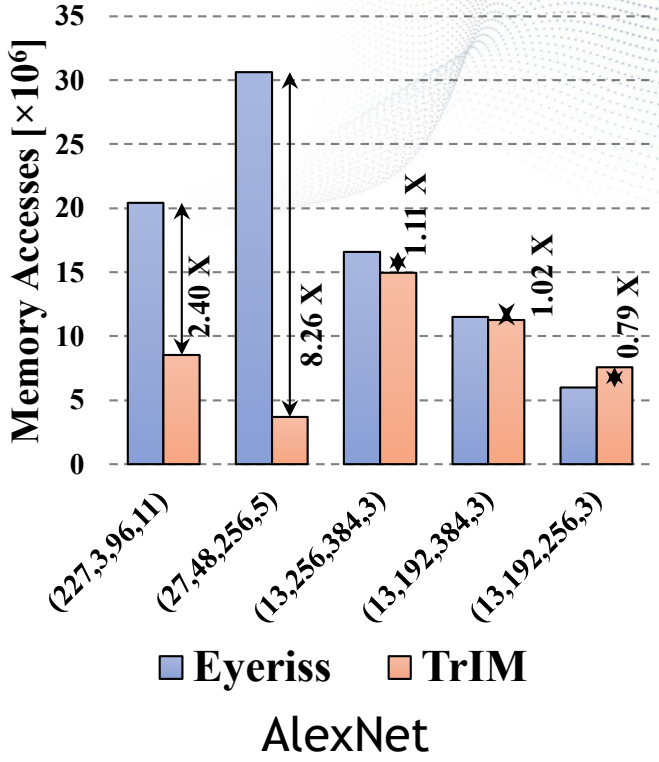
Different **cores** relate to different **3D filters** (ofmaps)

Sestito et al., IEEE TCAS-I 2025: https://doi.org/10.1109/TCSI.2024.3522351

TrIM/Eyeriss: Memory Accesses



VGG-16



AlexNet

Sestito et al., IEEE TCAS-I 2025: https://doi.org/10.1109/TCSI.2024.3522351

Comparisons: FPGA Designs

	[1]	[2]	[3]	**TrIM Arch.**
Device	XCZU9EG	XCZU3EG	XCVX690T	**XCZU7EV**
Precision [bits]	16	8	16	**8**
PEs	1024	256	243	**1512**
Dataflow	OS + WS	WS	RS	**TrIM**
LUTs	348K	40.78K	107.17K	**194.35K**
FFs	N.A.	45.25K	34.45K	**89.72K**
DSPs	1061	257	7	**0**
BRAMs [Mb]	8.82	4.15	N.A.	**10.21**
Freq. [MHz]	200	150	150	**150**
Peak GOPs/s	409.6	76.8	72.9	**453.6**
Power [W]	11	1.398	8.25	**4.329**
GOPs/s/W	37.24	54.94	8.84	**104.78**

[1] Sun et al., IEEE TVLSI 2023: https://doi.org/10.1109/TVLSI.2023.3241933
[2] Wu et al., IEEE TCAS-I 2024: https://doi.org/10.1109/TCSI.2023.3347417
[3] Zhang et al., IEEE TCAS-II 2024: https://doi.org/10.1109/TCSII.2023.3326489

Take-Aways

- CNNs extract **features** from images at **different abstraction levels**, mimicking the human **visual system**
- **Accuracy-Operations-Memory** trade-off
- **Systolic arrays** suitable to **tackle** the **Von-Neumann bottleneck**
- **TrIM**:
  - **Ifmap utilisation** is **maximised** through **local triangular movement**
  - **Reduced memory accesses** when compared to the art

Acknowledgements

- EPSRC FORTE Programme Grant (EP/R024642/2)
- RAEng Chair in Emerging Technologies (CiET1819/2/93)

- APRIL AI Hub (EP/Y029763/1)

The APRIL AI Hub aspires to unite the electronics and artificial intelligence (AI) communities for developing and bringing to market AI-based tools for boosting productivity across the entire electronics industry supply chain. https://linktr.ee/aprilhubuk