SCNN: An Accelerator for Compressed-sparse Convolutional Neural Networks

Original Paper by: Angshuman Parashar et al. (NVIDIA, ISCA '17)

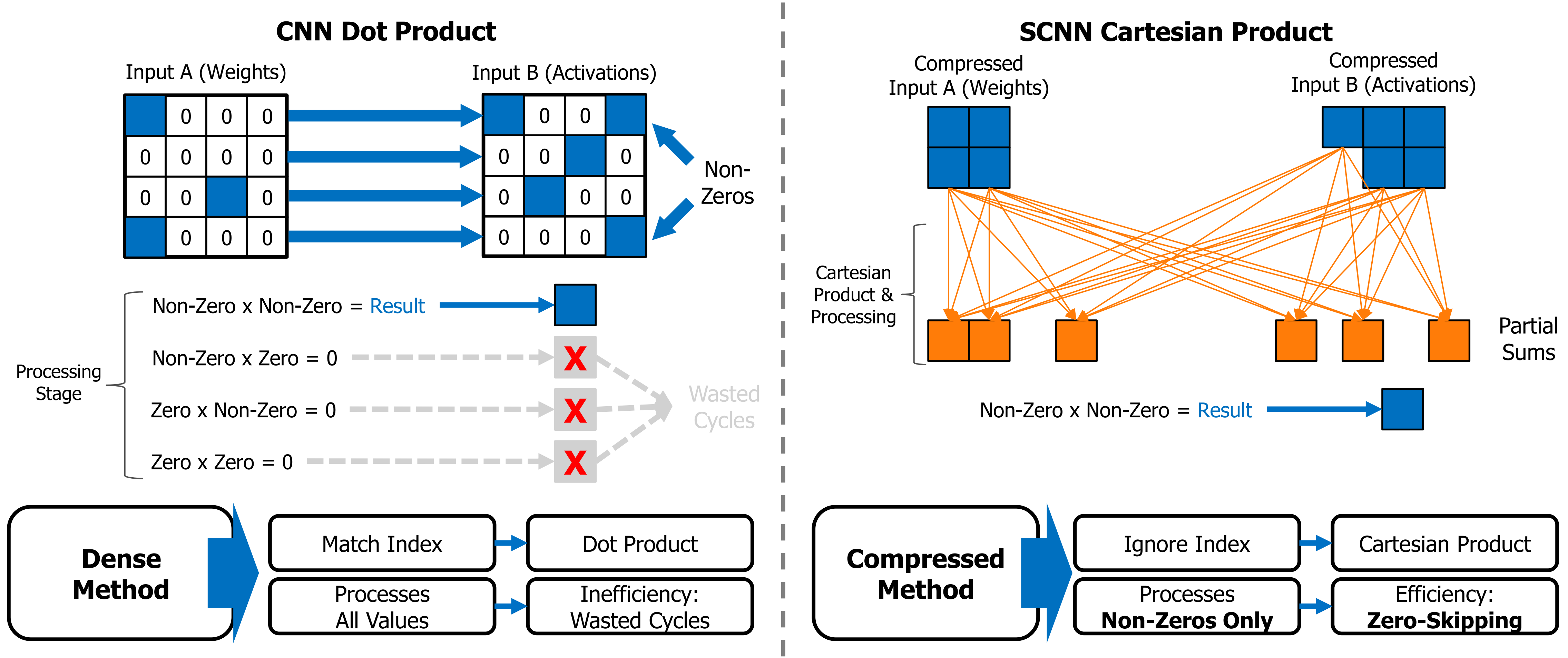
Follow-up Presentation

Presented by Jongyun Hur

Cartesian Product & Zero-Skipping

Cartesian Product & Zero-Skipping

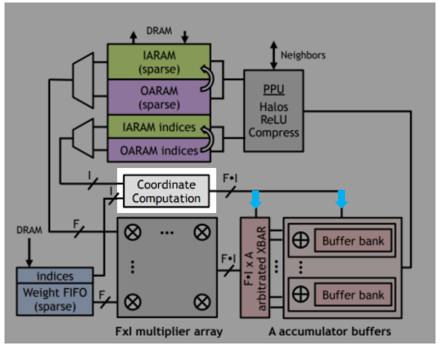
Why cartesian product enables Zero-Skipping

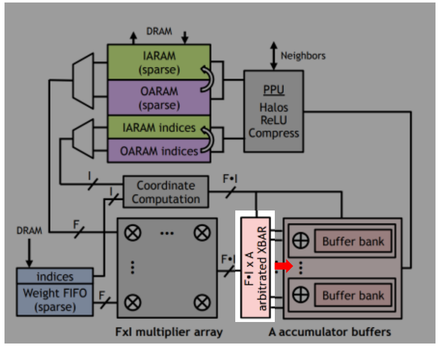


Coordinate Computation vs. XBAR

Coordinate Computation vs. XBAR

Decoupling address calculation from data delivery



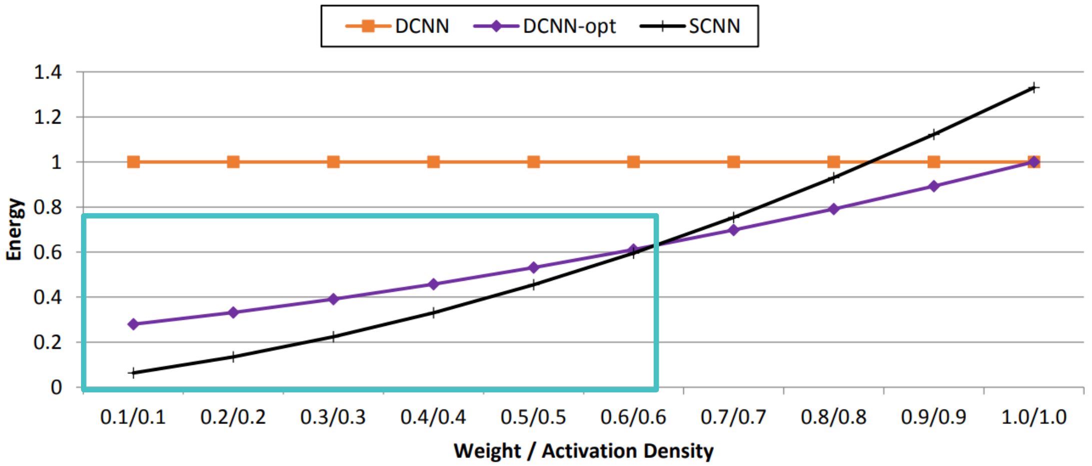


	Coordinate Computation	XBAR
Primary Role	**Computes** where the data should go.	**Delivers** the actual value to the destination.
Destinations & Sent Info	Sends Distinct Info to **Two** Targets: 1. To XBAR: Routing control signal (Bank ID) 2. To Buffer: Storage location (Memory Address)	Sends Data to Single Target: 1. To Buffer: Actual partial sum values generated by multipliers.
Mechanism	Decodes compressed indices to generate output coordinates (w,h,k) and splits them into routing ID and local address.	Routes the partial sums to the specific accumulator bank designated by the routing ID.

DCNN-opt vs. SCNN

DCNN-opt vs. SCNN

Why DCNN-opt consumes more energy than SCNN



(b) Energy

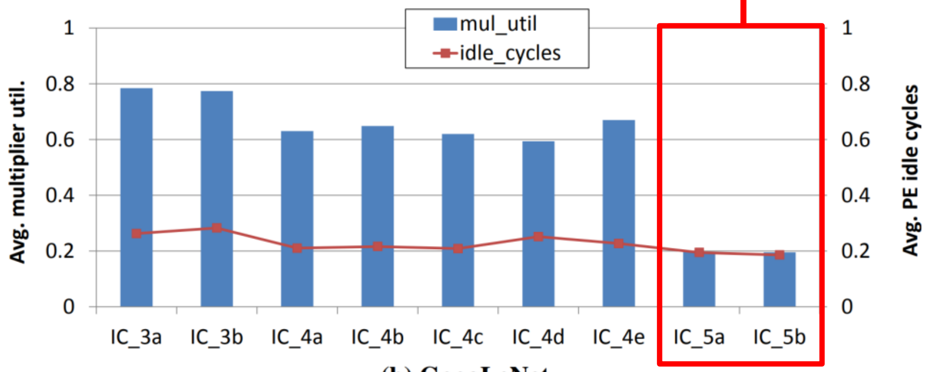
- **DCNN-opt (Zero-Gating)**
  - Reduces dynamic energy via Zero-Gating compared to DCNN.
  - Cannot skip zero operations → Execution time remains unchanged.
  - **Result: Wastes static power during idle cycles.**

- **SCNN (Zero-Skipping)**
  - Eliminates dead cycles via Zero-Skipping.
  - **Reduces both energy consumption and execution time.**
  - Result: Achieves the lowest total energy use by minimizing static power overhead.

GoogLeNet & Impact of Filter Size

GoogLeNet & Impact of Filter Size

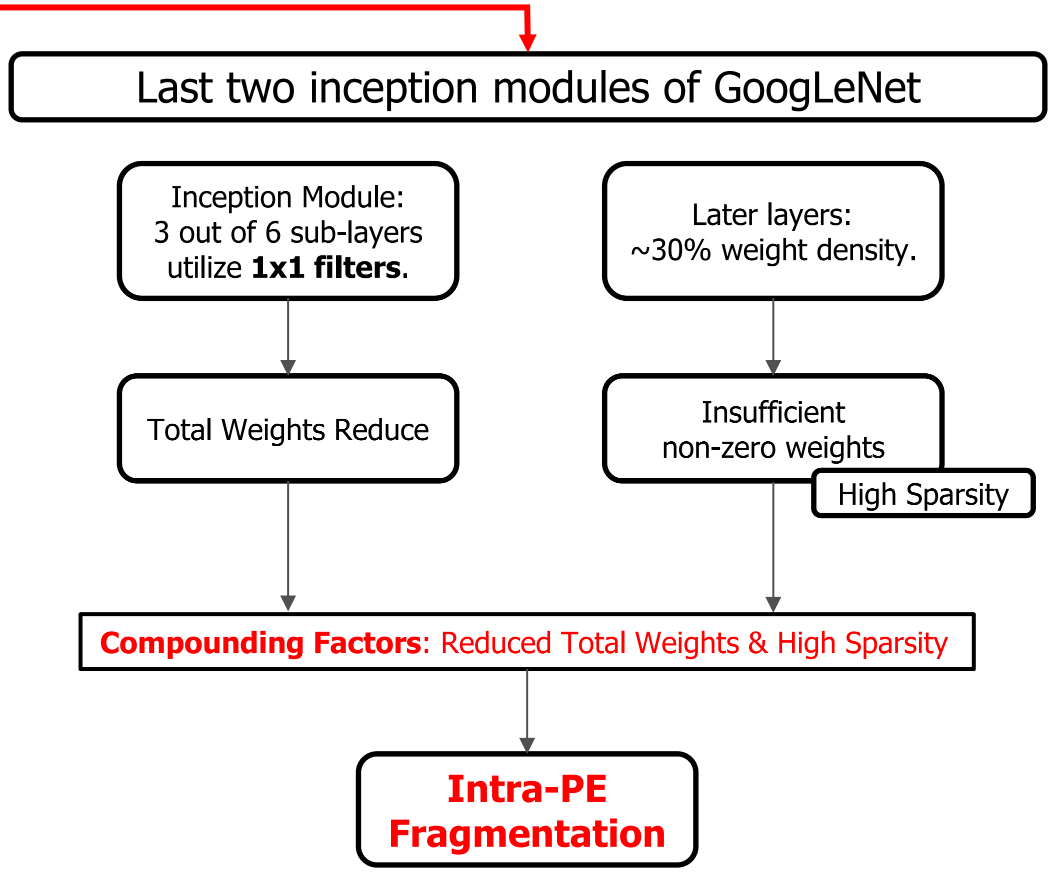
How small filters & high sparsity induce fragmentation



(b) GoogLeNet

$$\boxed{Total\,Weights} = Filter\,Width\,(R) \times Filter\,Height\,(S) \times \boxed{Group\,Size\,(K_c)}$$
$$= Filter\,Width\,(R) \times Filter\,Height\,(S) \times 8$$

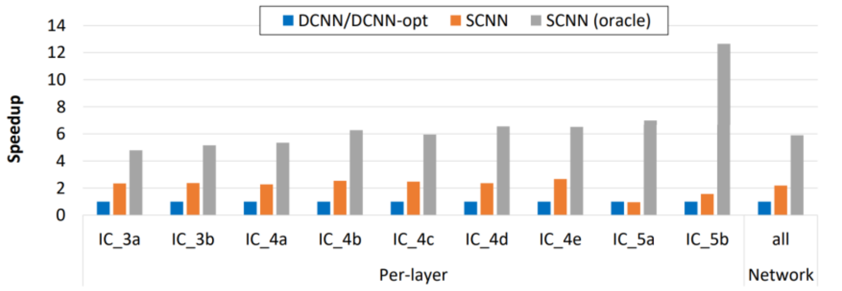
- Group size (K_c) is fixed at 8 in the paper.
- Total Weights: Total learnable parameters per layer



SCNN vs. Oracle

SCNN vs. Oracle

Performance gaps relative to the theoretical limit



(b) GoogLeNet

Performance Gap between SCNN and SCNN (oracle)

$$\underbrace{\text{SCNN(oracle)}=\frac{Total\ Multiplications}{Number\ of\ Multipliers}=\frac{Total\ Multiplications}{1024}}$$

An ideal theoretical abstraction
without physical hardware limitations

SCNN

Distributed Resources:
16 Multipliers per PE → Intra-PE fragmentation

Leads to: Low Utilization

Uneven Workload Distribution across PEs → Inter-PE load imbalance

Leads to: Idle Cycles

SCNN (oracle)

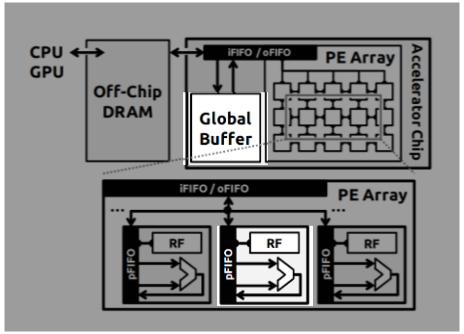
Unified Resources:
1024 Multipliers in Single Pool → No Intra-PE fragmentation

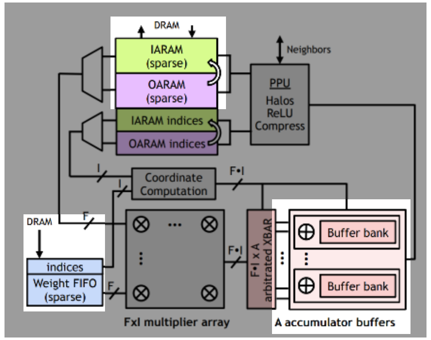
Even Workload Distribution across PEs → No Inter-PE load imbalance

Memory Difference: CNN vs. SCNN

Memory Difference: CNN vs. SCNN

Functional specialization of L0, L1 storage and accumulation logic





	CNN	SCNN
L1 Memory	Global Buffer	IARAM/OARAM (Physically distributed in PEs, but functionally serves as L1)
L0 Memory	Register File (RF)	Weight FIFO, Accumulator Buffer, Vector I/F Register
Architecture	**Unified** All data types (Input, Weight, Psum) share the same RF.	**Distributed** Uses physically separated, dedicated memories optimized for each data role.
Accumulation Strategy	**Deterministic (Fixed).** Output addresses are pre-determined, allowing sequential accumulation in the RF/Buffer.	**Scatter Accumulation.** Output addresses are scattered, requiring Banked Buffers and a Crossbar to handle conflicts.