Mediatek Dual-Core Deep-Learning Accelerator for Versatile AI Applications

Chih-Chung Cheng and Chien-Hung Lin

MEDIATEK

MediaTek, Hsinchu, Taiwan

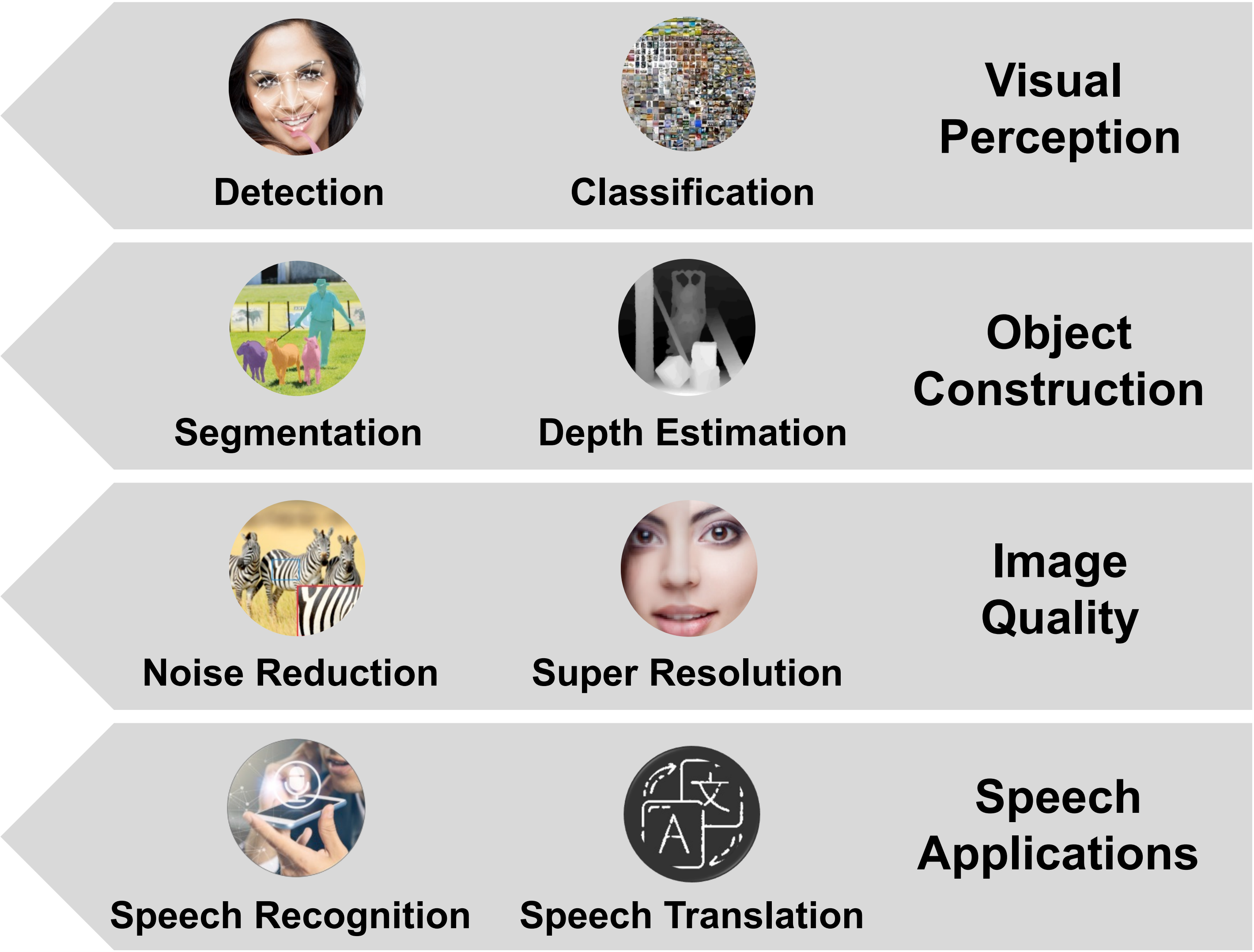
Outline

- Motivation
- Overall Architecture
- Key Features
- Implementation Results
- Conclusion

Outline

- Motivation
- Overall Architecture
- Key Features
- Implementation Results
- Conclusion

AI Applications in Smartphones



Detection

Classification

Visual Perception

Segmentation

Depth Estimation

Object Construction

Noise Reduction

Super Resolution

Image Quality

Speech Recognition

Speech Translation

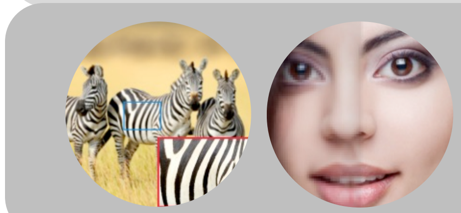
Speech Applications

Versatile Topologies, OPs, and Precisions

Precisions

INT8 – INT16

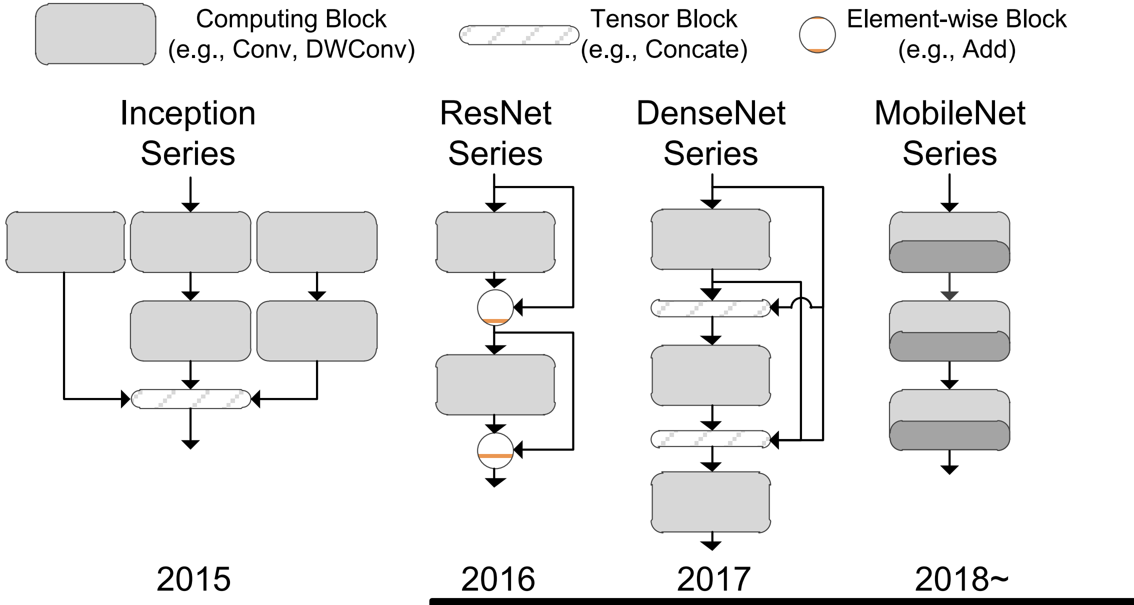
INT8 – FP16



INT16 – FP16

INT8 – FP16

Topology

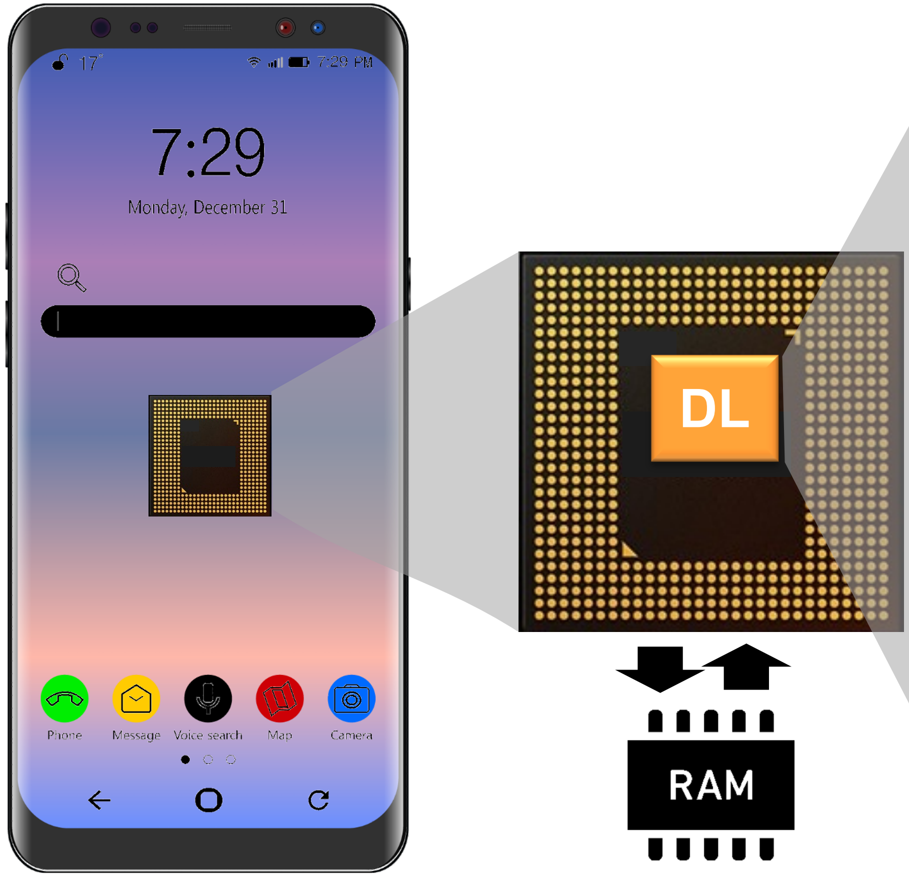


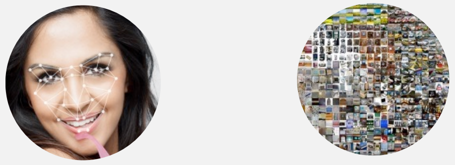
Programmability is necessary

Operations

Convolution	Conv, DWConv, TranposeConv, Dilated Conv, FC, etc.
Pooling	Max-pooling, Avg-pooling, ROI-Align, L2-pooling, etc.
Activation	Relu, Relu6, PRelu, Tanh, Sigmoid, Elu, etc.
Element-wise	Add, Mul, Sub, Neg, Min, Max, Mean, Sum, etc.
Pixel	Resize Bilinear, Resize Nearest Neighbor, etc.
Tensor	Concate, Split, Reshape, Transpose, Slice, Depth-to-Space, etc.
Recurrent	RNN, LSTM, etc.

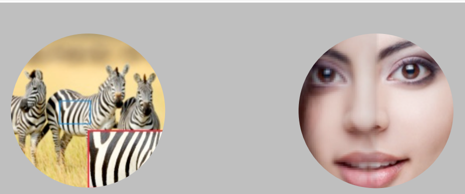
Constrained Power/BW vs. High Performance



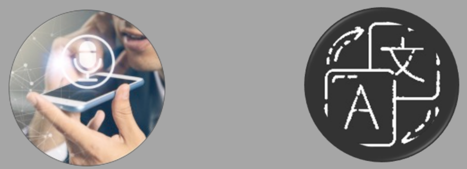


Perception
Compute: 0.01 ~ **8 TOPS**
BW: ~**2 GB/s** (@100FPS)

Construction
Compute: 0.1 ~ **10 TOPS**
BW: ~**8 GB/s** (@30FPS)



Quality
Compute: 0.5 ~ **40 TOPS**
BW: > **15 GB/s** (@2FPS)



Speech
BW: > **20 GB/s**

Phone Constraints
Temperature **< 45 °C**
Power **< 5 Watts**

SoC Constraints
Power: **2 ~ 3 Watts**
BW: **10 ~ 30 GB/s**

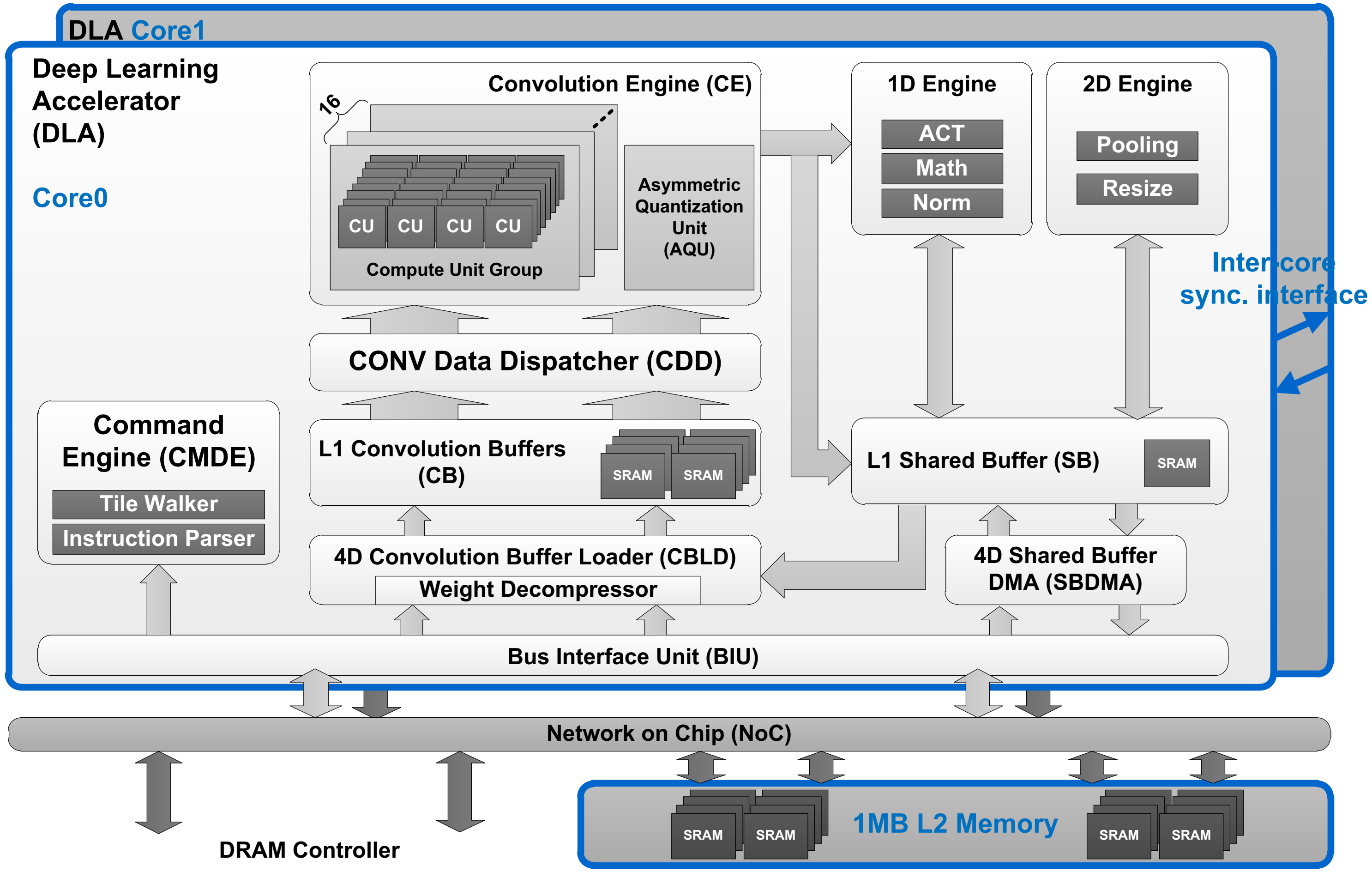
App. Constraints
Power: **< 1 Watt**
BW: **1 ~ 10 GB/s**

***BW: DRAM Bandwidth**

Outline

- Motivation
- Overall Architecture
- Key Features
- Implementation Results
- Conclusion

Overall Architecture



- **Multi-core architecture**
  - Dual-core in this implementation
- **Hardware sync. interface**
  - Direct inter-DLA communication
  - No intervention by AP-MCU
- **L2 memory**
  - Inter-layer data exchange
  - Inter-MDLA data exchange

Convolution Engines

1 Convolution Buffer Loader (CBLD)

- 4-D DMA with stride access capability, for moving weight/activation from ext. to CB
- Support compressed weight

2 Convolution Buffers (CB)

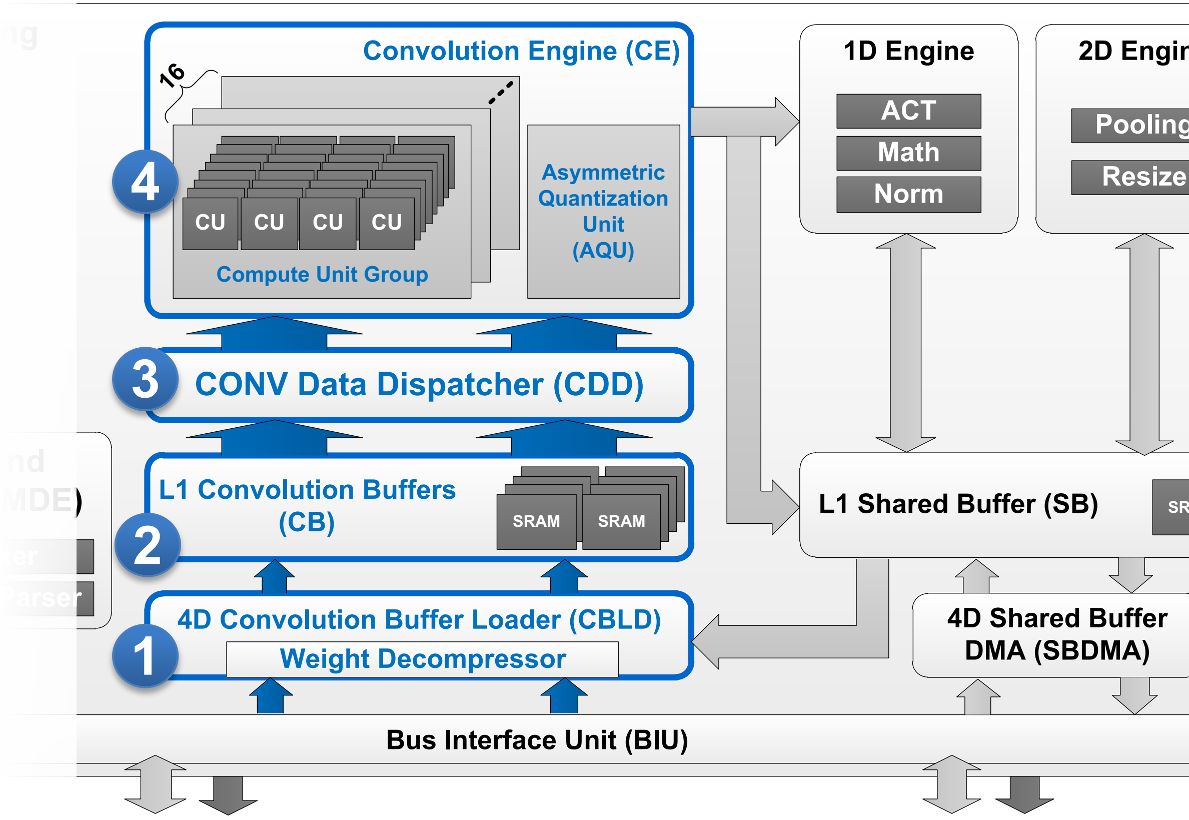
- L1 memory for activation & weight of tiles
- No DRAM access for CONV of this tile

3 CONV Data Dispatcher (CDD)

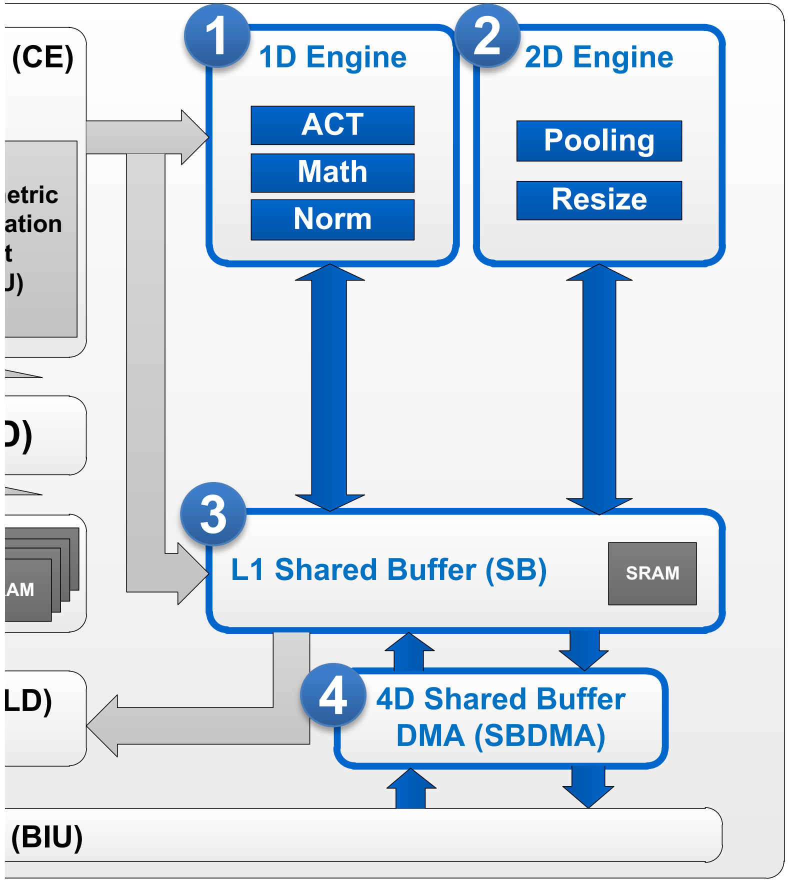
- Read data from CB and dispatch to CE
- Register banks for reduce CB access

4 Convolution Engine (CE)

- Support Multiple kinds of CONV and FC



Non-Convolution Engines



1 1D Engine

- Activation functions (ReLu, PReLu, …)
- Math functions (Add, MUL, ….)

2 2D Engine

- Pooling functions (Avg. , Max)

3 L1 Shared Buffer (SB)

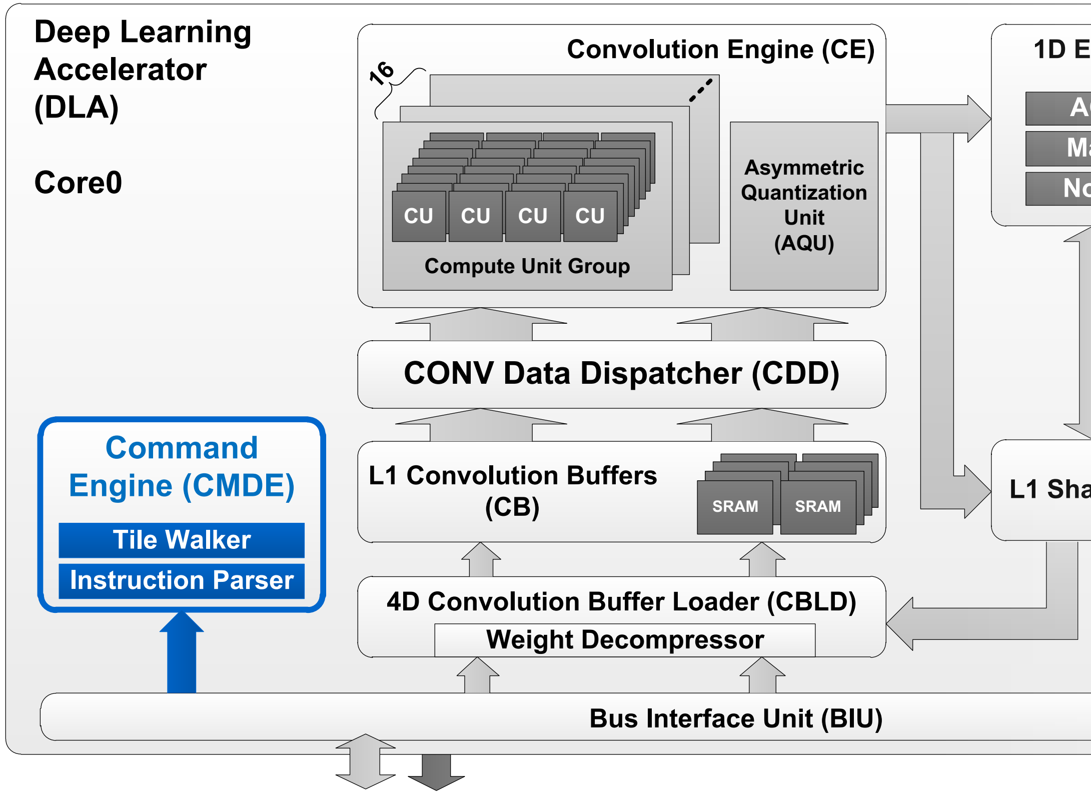
- For engines to exchange data

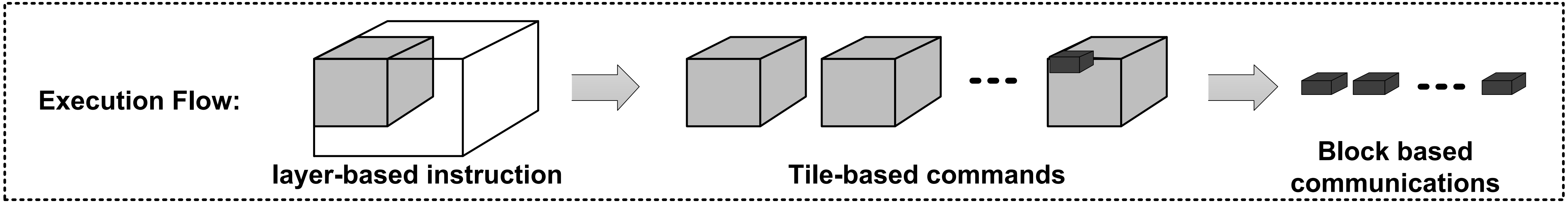
4 Shared Buffer DMA (SBDMA)

- 4D DMA with stride access capability
- Move data from/to external to/from SB

Command Engine for Flow Control

- **Read DLA programs**
  - A DLA program consists of layer-based instructions
- **Decode instruction and issue commands to engines**
  - Multiple tile-based commands for an instruction
- **Control working flow**
  - Handle data dependency
  - Prevent resource conflict





Execution Flow:

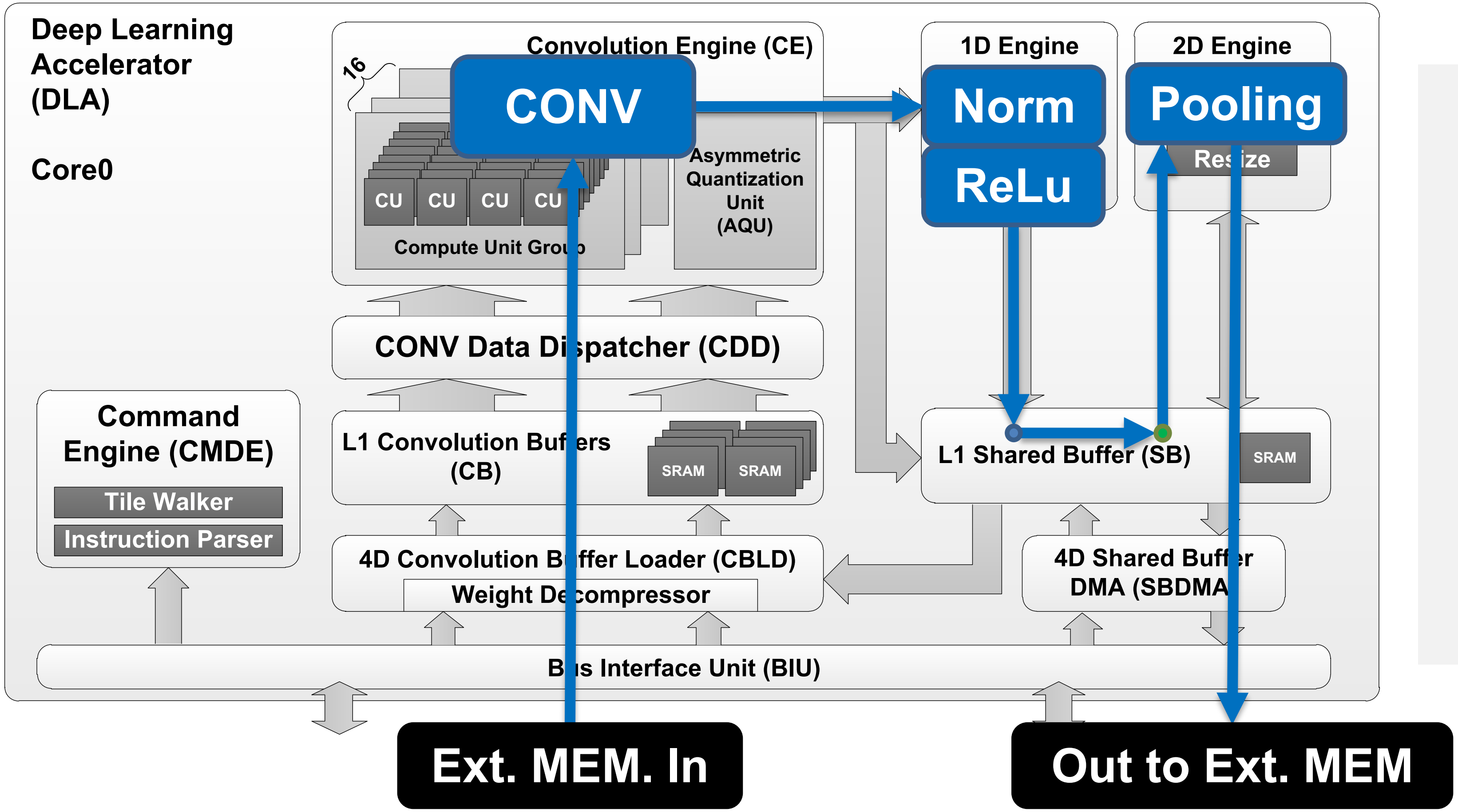
layer-based instruction

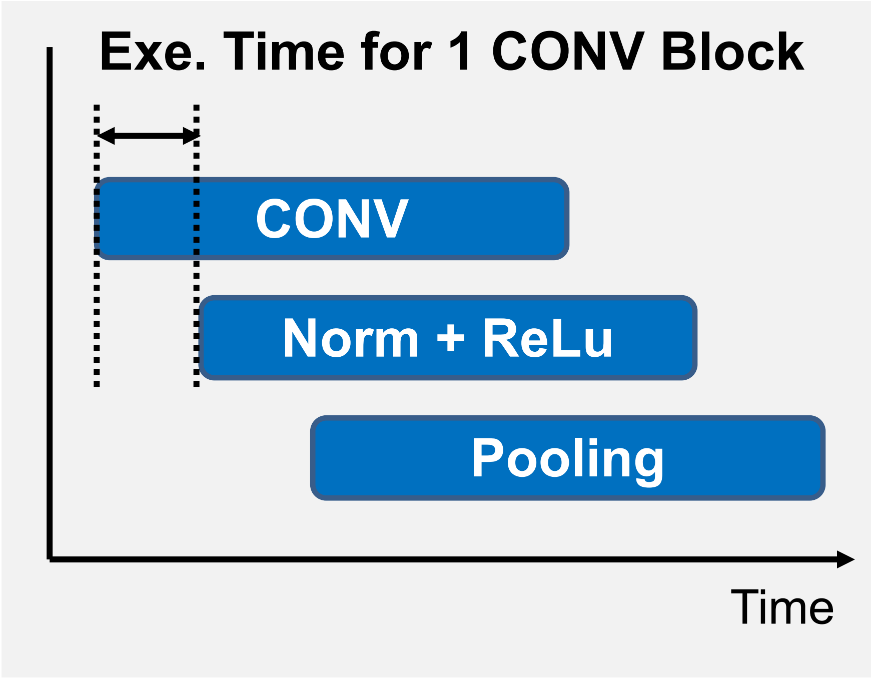
Tile-based commands

Block based communications

Example of Network Execution

- In following example with 4 OPs, **only 2 external memory** access (IN & Out)
- Pipeline working with block unit makes **low operating latency**



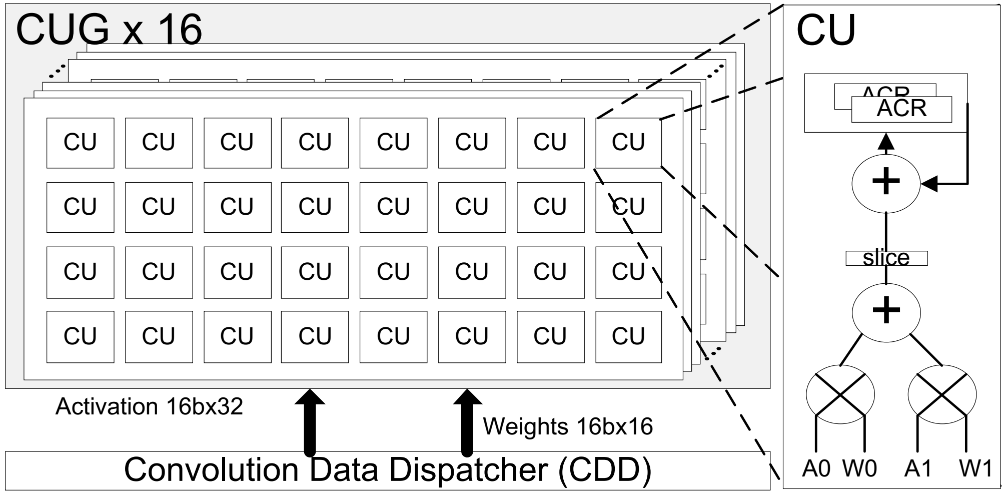


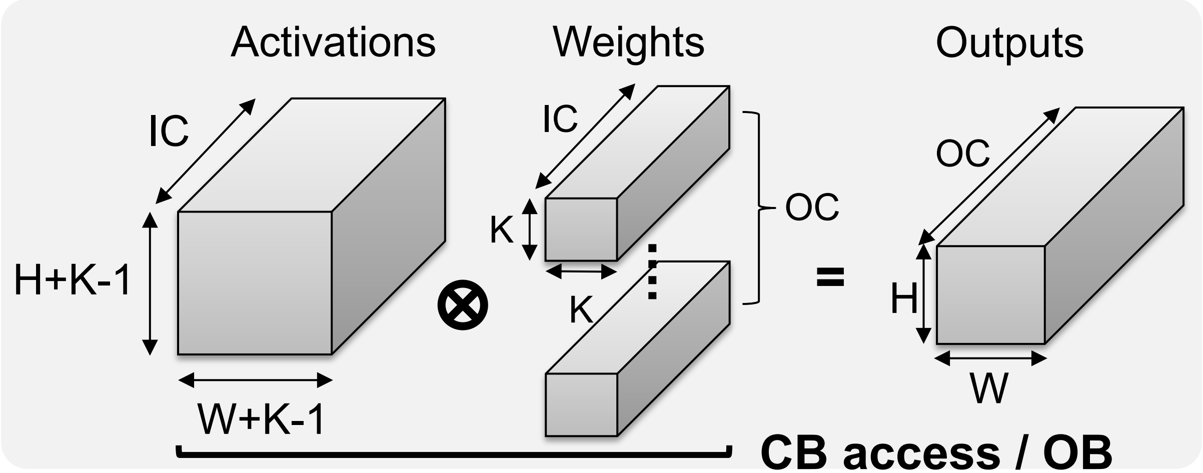
Outline

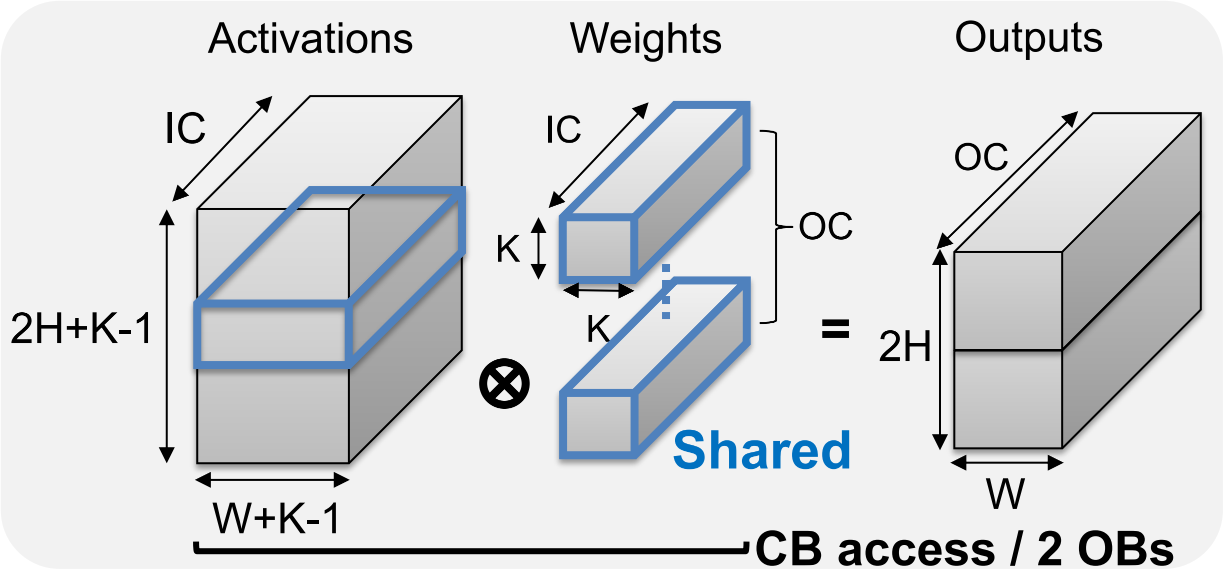
- Motivation
- Overall Architecture
- Key Features
- Implementation Results
- Conclusion

Data Re-Use in Convolution Engine

- **CONV MAC Array with data reuse**
  - 2x8b or 1x16b or 0.5xFP16 MACs / cycle per CU
  - 32 CUs in 1 CUG, and total 16 CUGs
- **Output stationary mechanism for 1 output block**
  - **Reduce SRAM access power** for partial-sum (PS)
    - 1 PS SRAM R/W power is 6x of MULT
  - A 3-D output block (OB) contains 512 CONV results
- **Dual accumulators for 2-OB concurrent mode**
  - **Reduce 50% of weight** and **16% of activation** access to CB

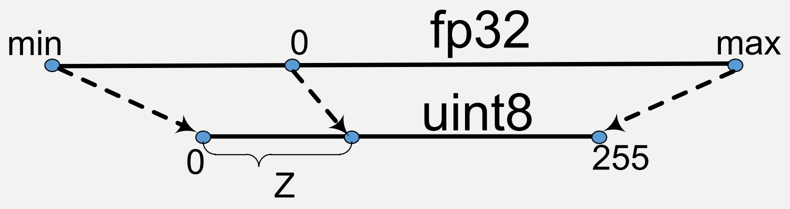






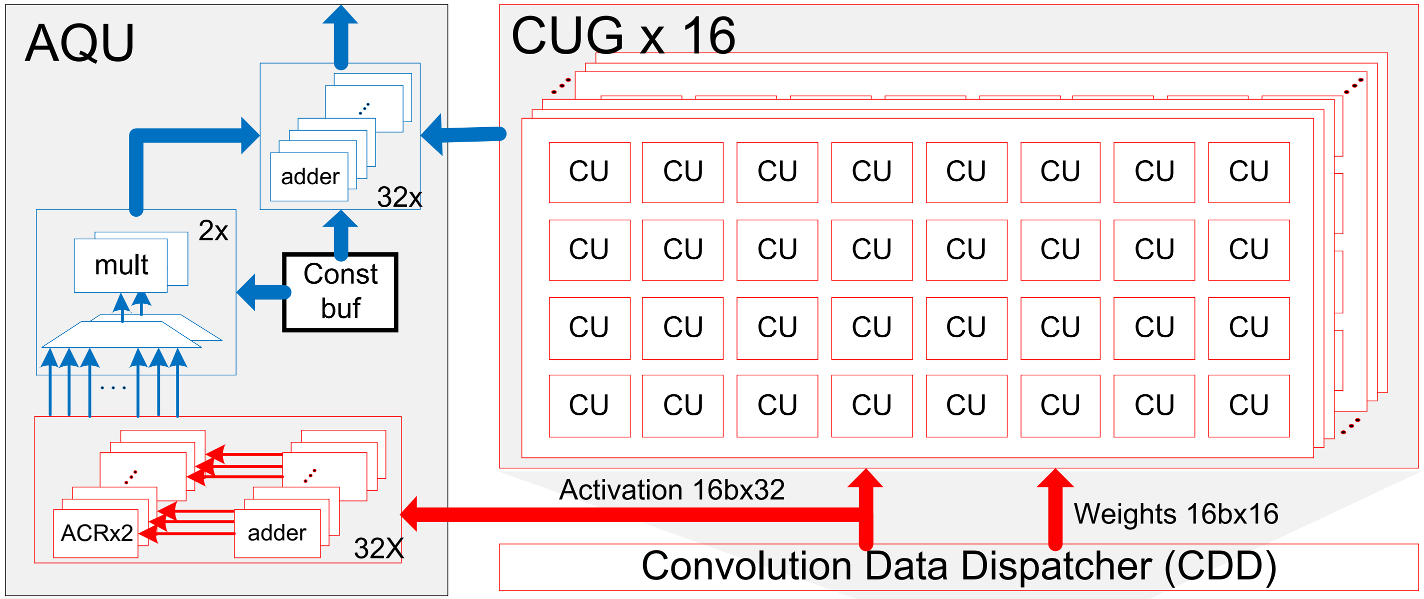
Asymmetric Quantization Unit

- **Asymmetric Quantized (ASYMM-Q) data format**
  - Applied by Android neural network API (NNAPI)
  - Better utilization of rare bits, but mathematical operations become complicated
- **Native support for ASYMM-Q**
  - 32% performance improving
  - Almost no power overhead



`A[x,y,ic] = A_Q[x,y,ic] + Z_A;`

`W[x,y,ic] = W_Q[x,y,ic] + Z_w;`



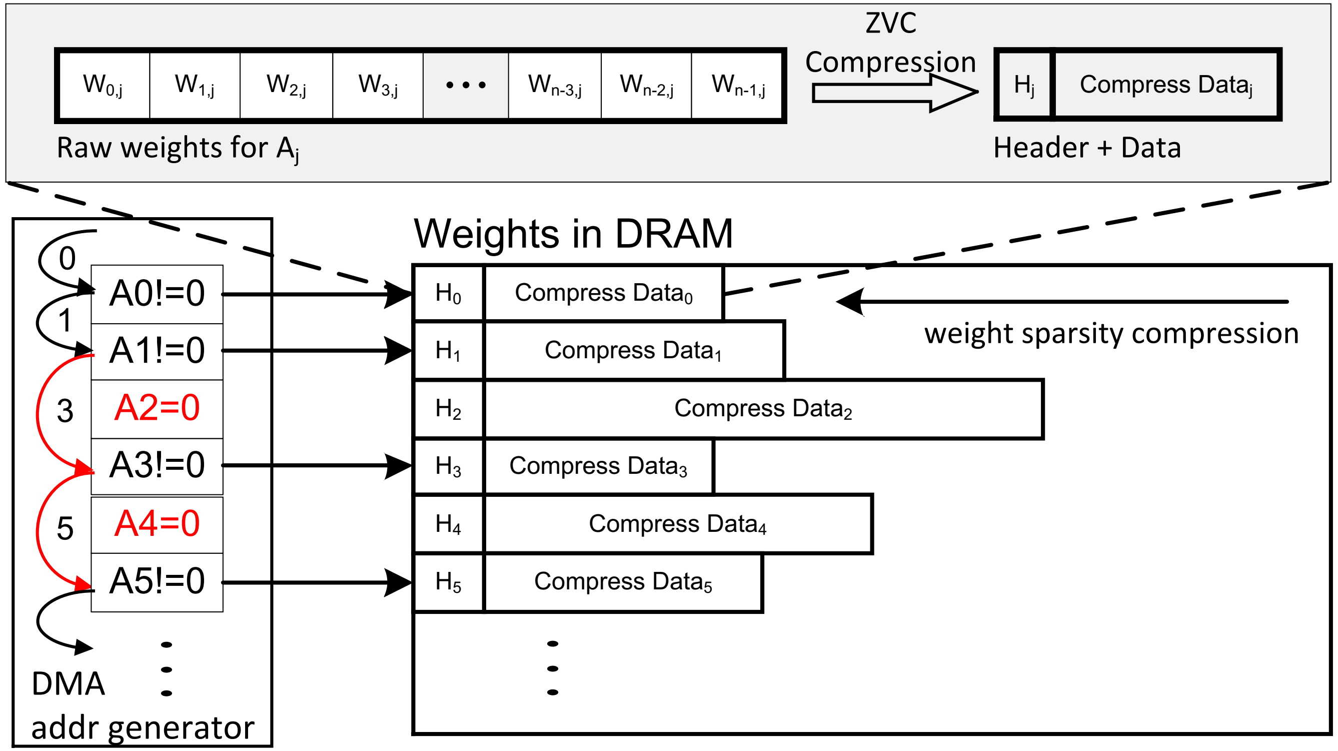
$$OUT[x,y] = \left(Z_A * \sum_{w_x,w_y,ic} W_Q[i]\right) + Z_A * Z_w * K + Z_w * \left(\sum_{w_x,w_y,ic} A_Q[x+w_x, y+w_y, ic]\right) + \sum_{w_x,w_y,ic} A_Q[x+w_x, y+w_y, ic] * W_Q[w_x, w_y, ic]$$

Weight Compression and Zero-Skipping

- **Zero value compression on weights to reduce BW**
- **Activation zero skipping in FC layer**
  – A zero valued activation makes related computation & weight access skipped
  – Reduce 87% of DRAM BW and 71% of computation @ typical LSTM

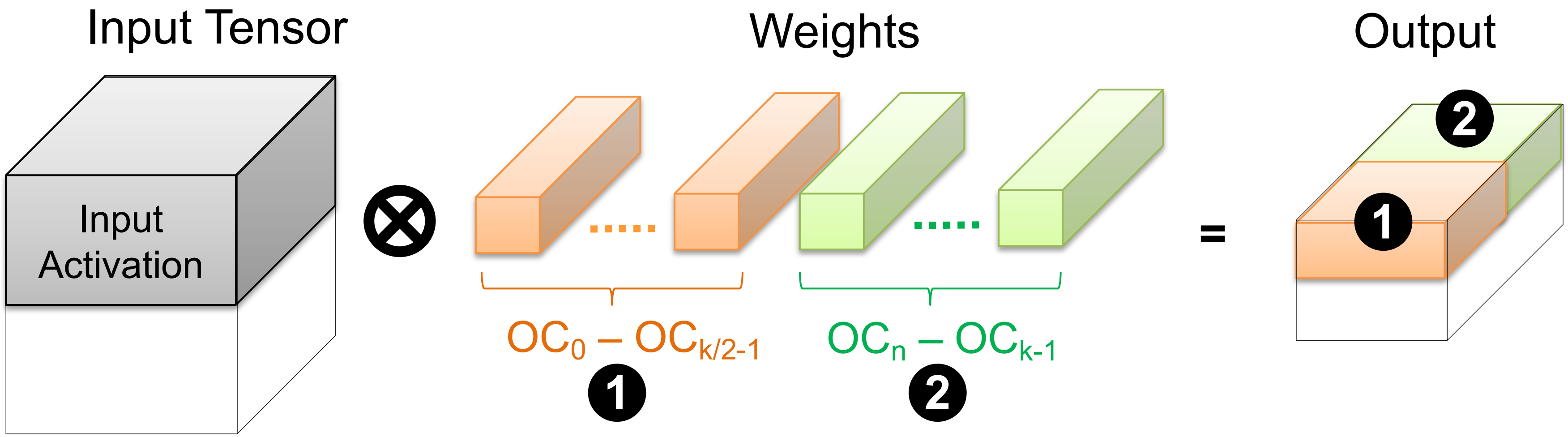
Fully-Connected Layer Operation with zero-skipping

$$[\ A0\ ,\ A1\ ,\ 0,\ \dots\ Ac\ ] \times \begin{bmatrix} W_{00}\ , & W_{10}\ , & \cdots & W_{k0} \\ W_{01}\ , & W_{11}\ , & \dots & W_{k1} \\ \cancel{W_{02}\ ,} & \cancel{W_{12}\ ,} & \cancel{\dots} & \cancel{W_{k2}} \\ \vdots & \vdots & \dots & \vdots \\ W_{0c}\ , & W_{1c}\ , & \dots & W_{kc} \end{bmatrix}$$

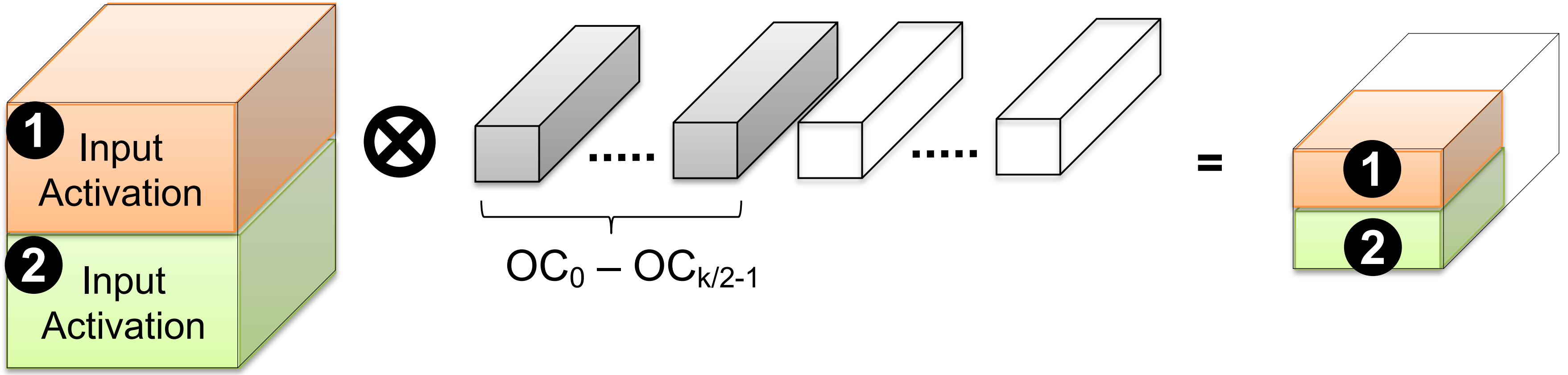


Output Tile Ordering for BW Reduction

C first:
Weight reload
Activation reuse



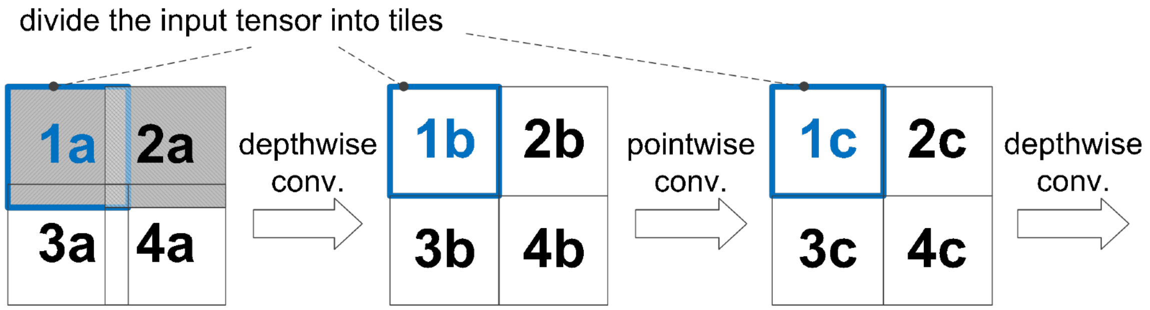
X-Y first:
Weight reuse
Activation reload



- Programmable tile walking order
- Compiler searches for the tile order with minimum BW

Layer-Fusion for BW Reduction

- Traverse layer first to utilize the short-life-time behavior of NN feature maps
- DLA core supports 2-layers fusion
- Utilize L2 SRAM to fuse more layers
- MobileNet v1 result: reduce 63% of DRAM access by combined fusion



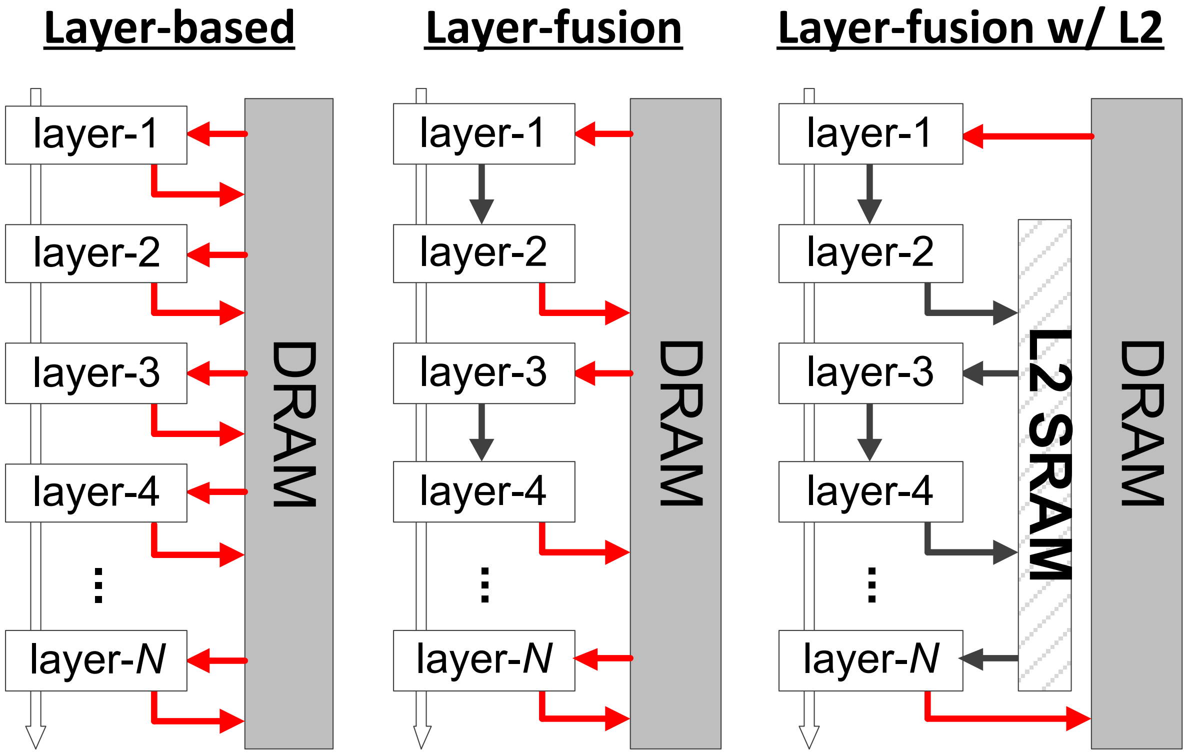
Execution order:

Layer-based:

(1a, 2a, 3a, 4a) → (1b, 2b, 3b, 4b) → (1c, 2c, 3c, 4c)

Tile-based + layer-fusion:

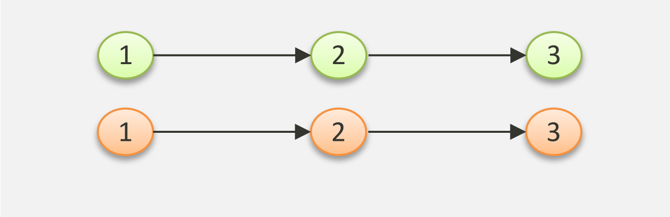
(1a, 1b, 1c) → (2a, 2b, 2c) → (3a, 3b, 3c) → (4a, 4b, 4c)



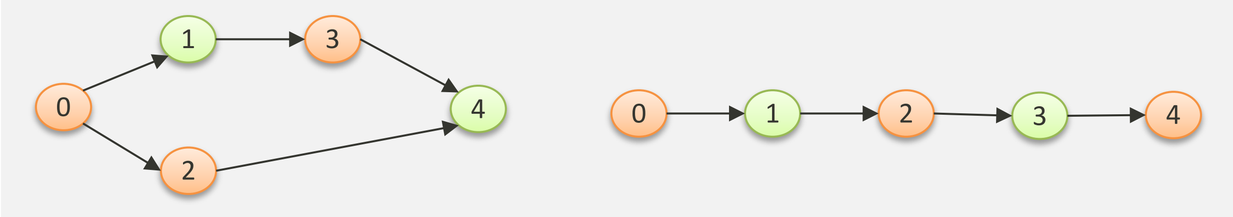
Co-working Mechanism among DLA Cores

- Handle different OPs

Core0 Core1

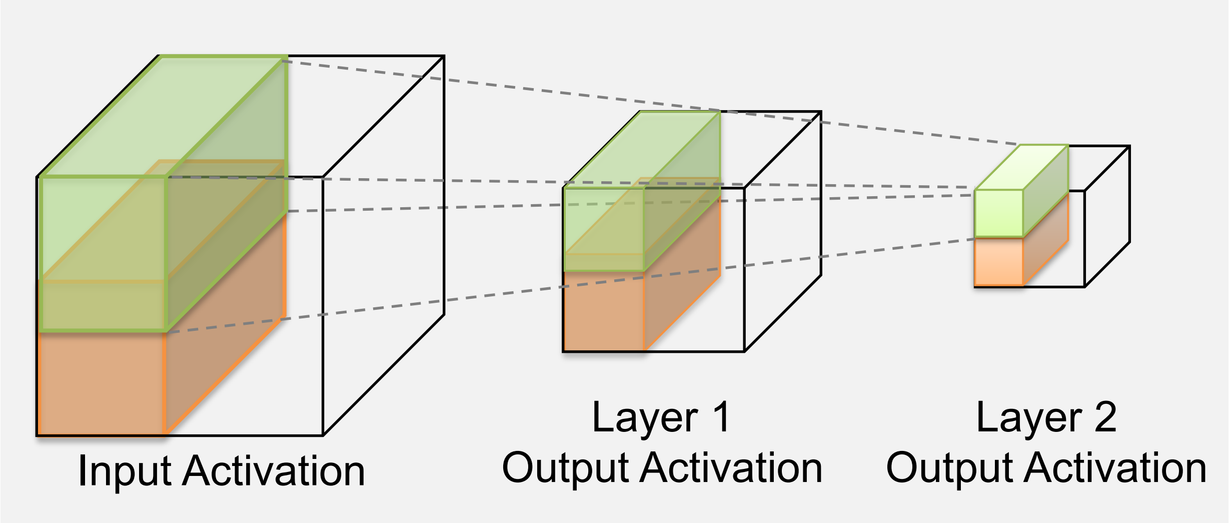


Network/Batch Based

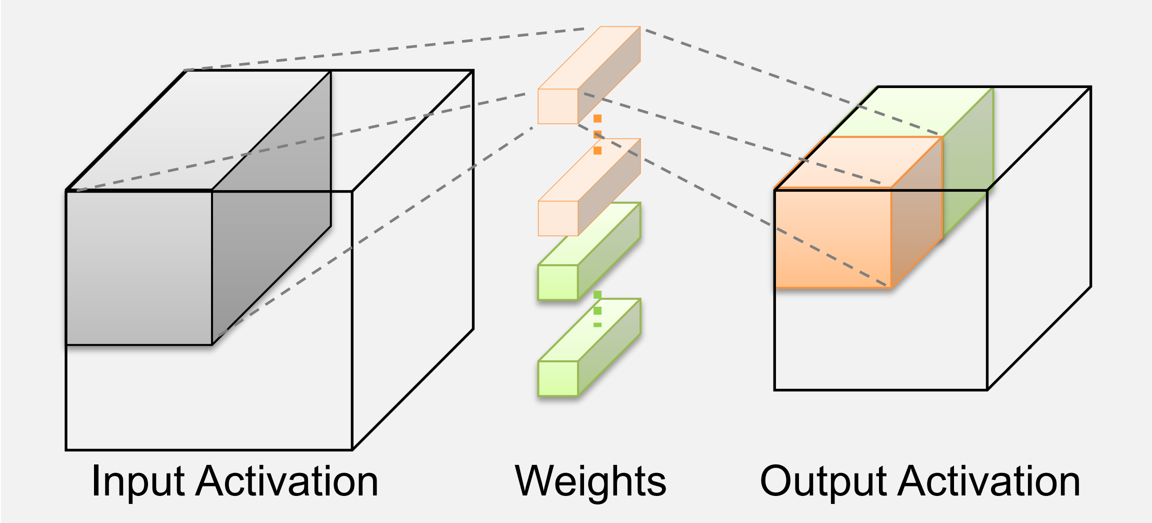


Layer/Tensor Based

- Concurrently work on the same OPs



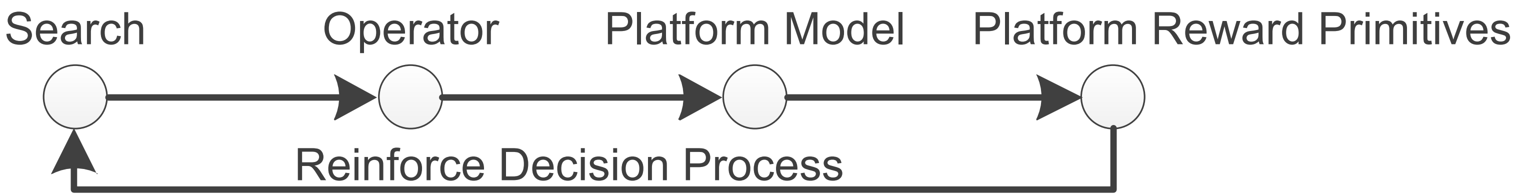
Network Deep Fusion (NDF)



Symmetric Multi-Processing (SMP)

Reinforcement Learning Based Tiling Strategy

- Reinforcement learning (RL) trained searching engine
  - Utilize SoC platform metrics as the reward



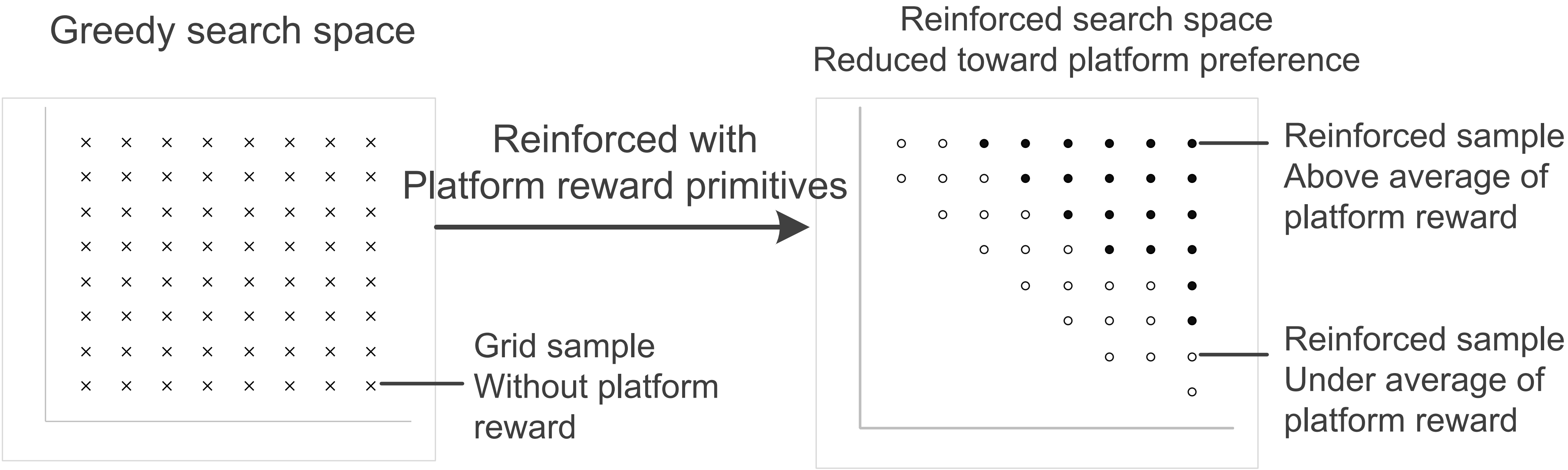
Search

Operator

Platform Model

Platform Reward Primitives

Reinforce Decision Process



Greedy search space

Reinforced search space
Reduced toward platform preference

Reinforced with
Platform reward primitives

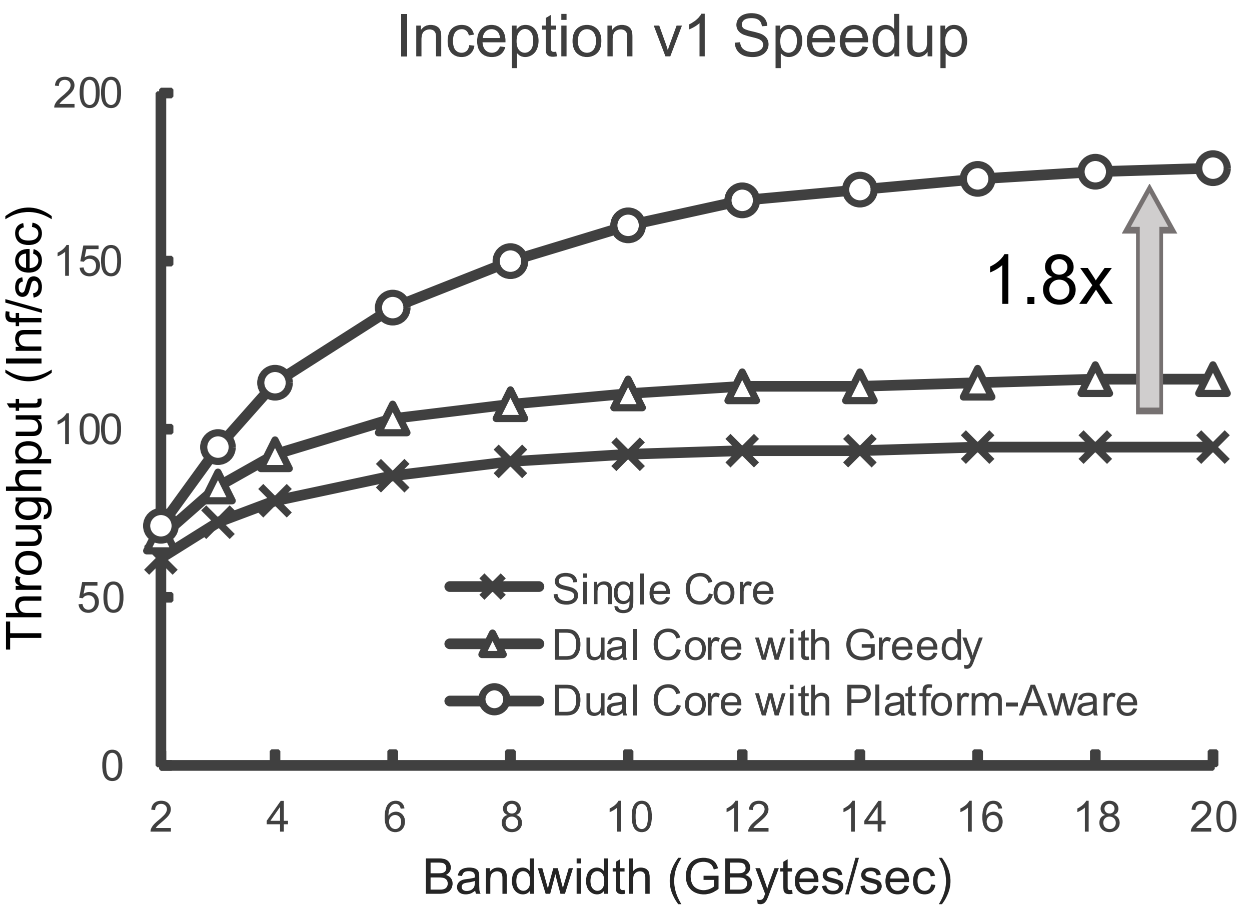
Grid sample
Without platform
reward

Reinforced sample
Above average of
platform reward

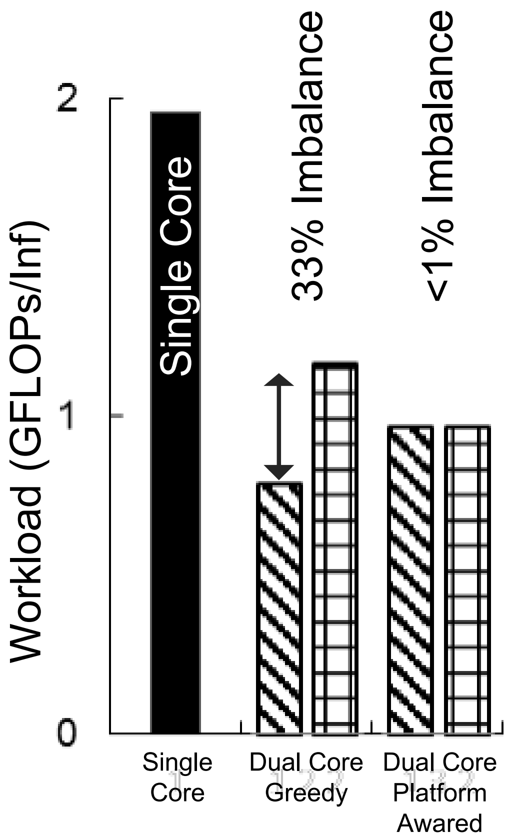
Reinforced sample
Under average of
platform reward

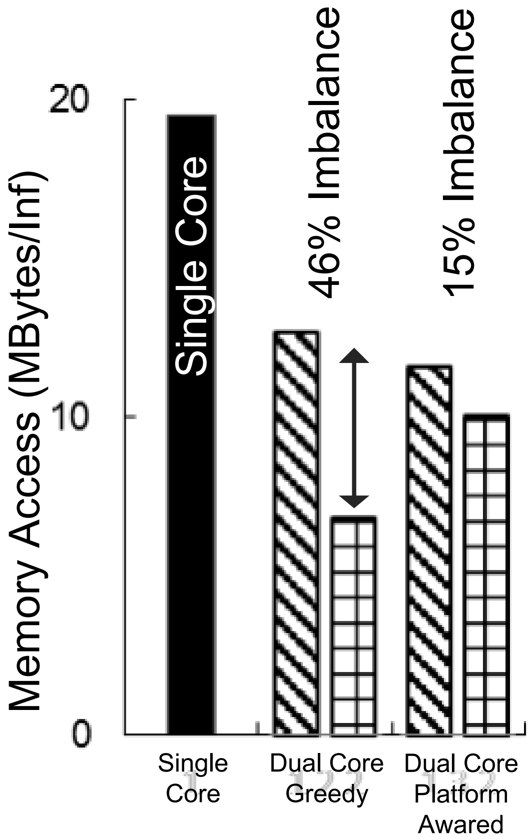
Benefits of RL-Based Tiling Strategy

- **Better dual-core performance**



- **More Balanced computation and BW**

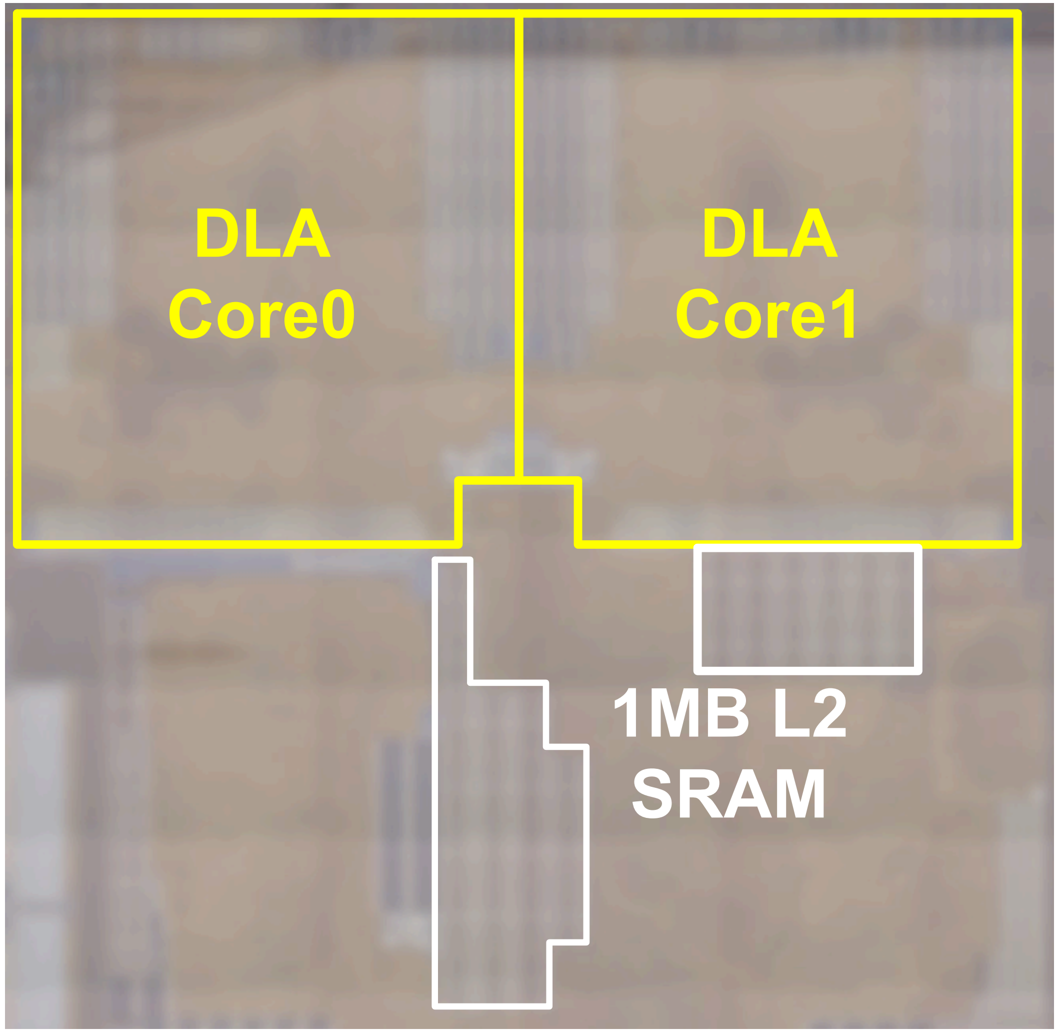




Outline

- Motivation
- Overall Architecture
- Key Features
- Implementation Results
- Conclusion

Implementation Summary



Process	7nm
Area	1.34 mm^2 /core 0.36 mm^2 L2 SRAM
Supply	0.575 V - 0.825 V
Speed	290 MHz - 880 MHz
Precision	ASYMM-Q8, INT8, INT16, FP16
Peak Performance (2 Cores)	3.6 TOPS @ 8b 1.8 TOPS @ 16b 0.9 TFLOPS @ FP16
Energy Efficiency	3.42 - 13.32 TOPS/W
Area Efficiency	1.19 TOPS/mm^2

Comparison with Previous Works

	ISSCC_2016 [2]	ISSCC_2017 [3]	ISSCC_2018 [4]	ISSCC_2019 [5]	This Work (2 Cores)
function	CONV	CONV	CONV, FC	CONV, FC	CONV, FC
Process (nm)	65	28, FD-SOI	65	8	7
Area (mm^2)	12.25	34.9 (chip) 2.2 (CAs)	16	5.5	3.04
Supply Voltage (V)	0.82 - 1.17	0.575 - 1.1	0.63 - 1.1	0.5 - 0.8	0.575 - 0.825
Operating frequency (MHz)	100 - 250	200 - 1175	2 - 200	67 - 933	290 - 880
On-Chip Memory (KB)	181.5	5760	256	1568	2176
Data Type	INT16	INT8, INT16	1 - 16	INT8, INT16	ASYMM-Q8, INT8, INT16, FP16
Peak Performance (GOPS)	84	676 (8b @ CAs) 75 (16b @ DSP)	7372 (1b), 1382 (4b), 691.2 (8b), 345.6 (16b)	1910 (8b)	3604 (8b)
Power (mW)	278 @ 200MHz	39 @ 0.575V	3.2 - 297 @ 0.63V - 1.1V	39 - 1553 @ 0.5V- 0.8V	174 - 1053 @ 0.575V - 0.825V
Energy Efficiency (TOPS/W)	0.17 @ 1V	2.93 (8b) @ 0.575V	50.6 (1b), 11.6 (4b), 5.57 (8b), 3.08 (16b) @ 0.66V	1.23 - 3.52 (Dense) 4.5 - 11.5 (Sparse) @ 0.8V - 0.5V	3.42 - 6.83 (Dense) 7.34 - 13.32 (Sparse) @ 0.825V - 0.575V
Area Efficiency (GOPS/mm^2)	6.86	21.55 (8b, chip) 307.64 (8b, CAs)	43.20 (8b)	347.42 (8b)	1185.68 (8b)

1.9 – 2.8x

3.4x

Dense: **0%** of weights is zero

Sparse: **75%** of weights are zero

Outline

- Motivation
- Overall Architecture
- Key Features
- Implementation Results
- Conclusion

Conclusion

- **An energy efficient, BW efficient, high performance, and flexible DLA for smartphones, which features with**
  - Multiple data type support: INT8 ~ FP16, and ASYMM-Q8
  - Optimized data reuse from register banks to all level of memories
  - Weight compression and FC layer zero-skipping
  - Multi-core architecture

Thank You