# tinyML® Foundation

*Enabling Ultra-low Power Machine Learning at the Edge*

## tinyML Summit April 22 - 24, 2024

TINY ML

www.tinyML.org

# An Energy Efficient, TOSA compatible, NPU for Transformer Networks

Ethos-U85

Rakesh Gangarajaiah
2024-04-24



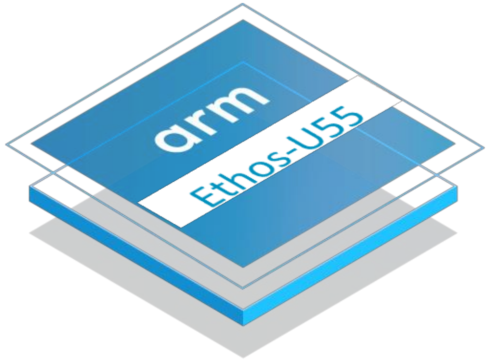
AI-generated image

# Energy efficiency at the Edge

- Challenges
  - Wide variety of workloads
  - Multiple machine learning frameworks
  - Limited power budget
  - Cost, Power, Performance and Area tradeoffs

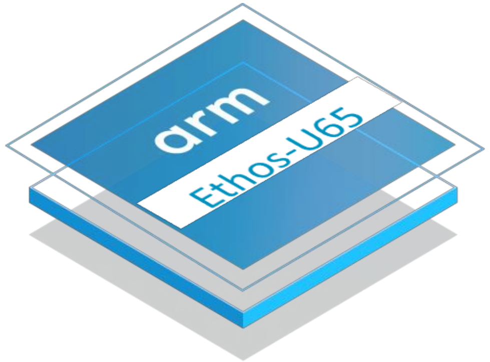
# Energy efficiency at the Edge

- Challenges
  - Wide variety of workloads
  - Multiple machine learning frameworks
  - Limited power budget
  - Cost, Power, Performance and Area tradeoffs
- NPUs from Arm: Ethos-U55 and Ethos-U65




**Cortex-M compatible**

- Orders of magnitude increase in NN perf



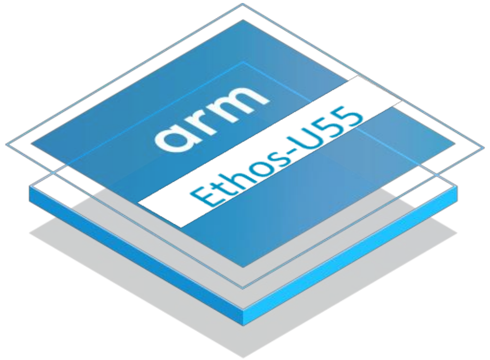

**Cortex-M and Cortex-A compatible**

- Designed to run in Cortex-A based systems and is tolerant to high DRAM latencies

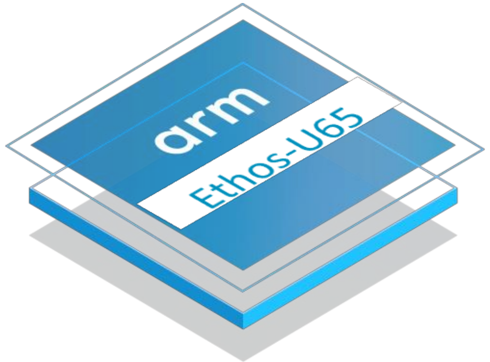
# Energy efficiency at the Edge

- Challenges
  - Wide variety of workloads
  - Multiple machine learning frameworks
  - Limited power budget
  - Cost, Power, Performance and Area tradeoffs
- NPUs from Arm: Ethos-U55 and Ethos-U65



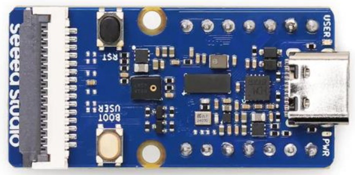

**Cortex-M compatible**

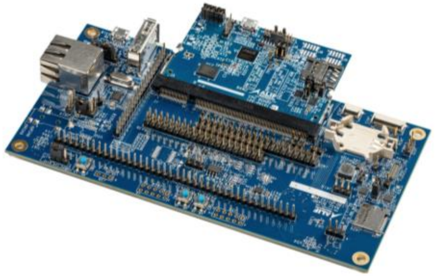
- Orders of magnitude increase in NN perf




**Cortex-M and Cortex-A compatible**

- Designed to run in Cortex-A based systems and is tolerant to high DRAM latencies



Himax ALIF SEMICONDUCTOR NXP infineon nuvoTon synaptics

arm

# Workloads and ML Frameworks

- Different ML workloads need different optimizations for high efficiency
  - CNNs are usually compute bound whereas Fully Connected (FC) layers are weight memory BW bound
- Different ML frameworks such as TensorFlow and PyTorch
  - Challenging to support all different operators in all popular frameworks
  - Common operators across each framework
- Hardware efficiency has a large dependency on SW driving/controlling the ML processor

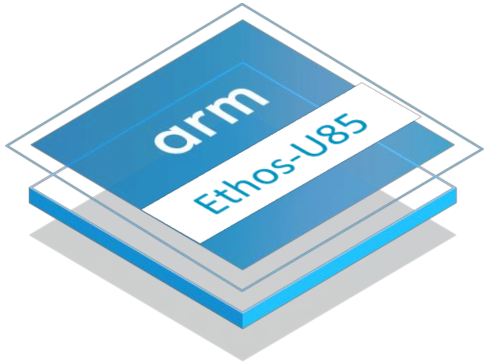
# Workloads and ML Frameworks

- Different ML workloads need different optimizations for high efficiency
  - CNNs are usually compute bound whereas Fully Connected (FC) layers are weight memory BW bound
- Different ML frameworks such as TensorFlow and PyTorch
  - Challenging to support all different operators in all popular frameworks
  - Common operators across each framework
- Hardware efficiency has a large dependency on SW driving/controlling the ML processor
- TOSA : Tensor Operator Set Architecture*
  - A minimal set of tensor operators to which many ML frameworks can be reduced
    - Around 70 operators
  - Contains functional and numerical descriptions of each operator : Quantized int and Floating point
    - Numerical precision description ensure consistent results across a wide range of hardware + frameworks
    - Includes reference code and conformance tests
  - Bridging the gap between hardware and software

*https://www.mlplatform.org/tosa/

# Arm Ethos-U85 NPU

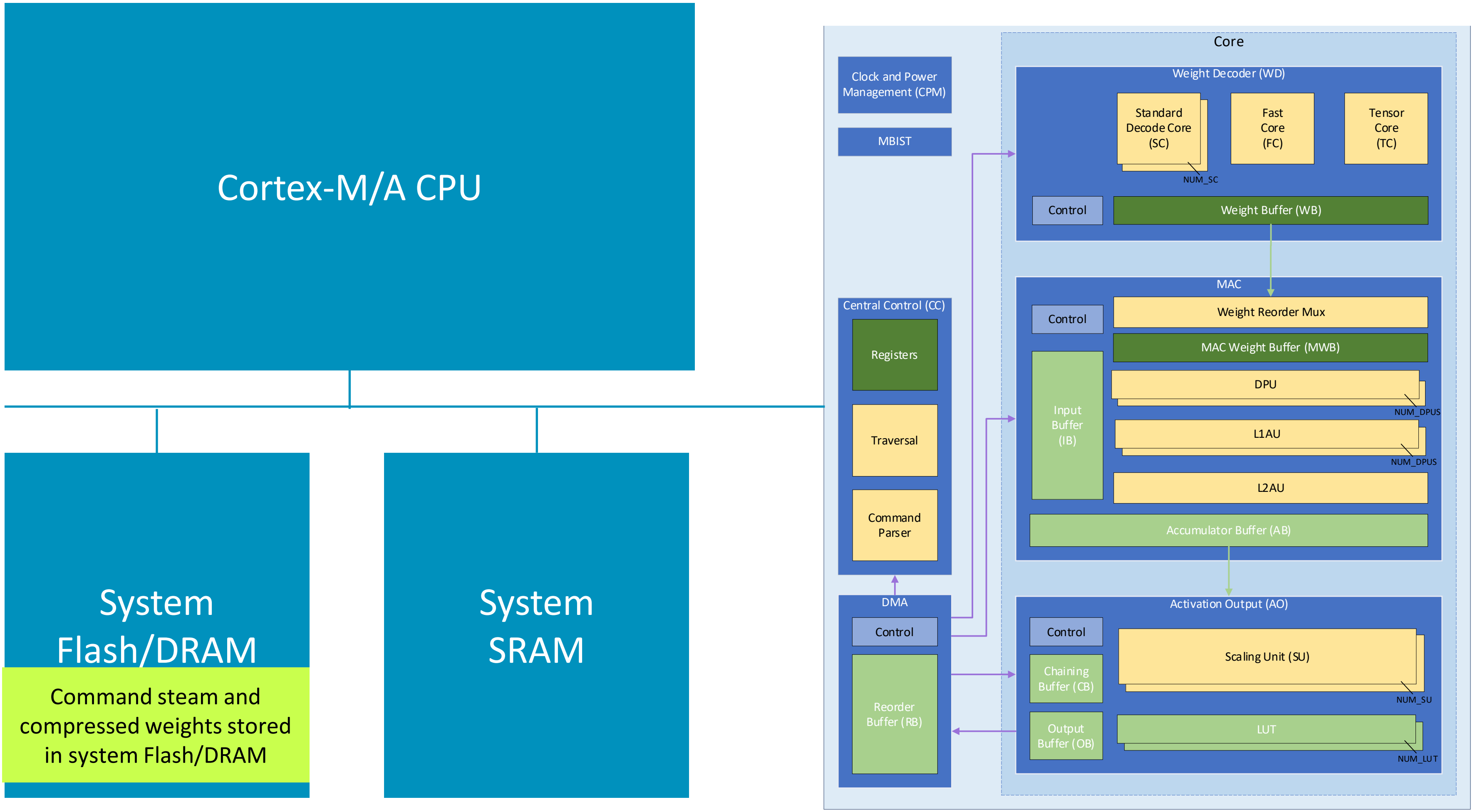
- Configurable : 128MACs/Cycle to 2048 MACs/Cycle (Up to 4TOPs @ 1GHz clock)
  - Enables a wide variety of implementations targeting different workloads from Audio to vision ML
- Improved energy efficiency over the Ethos-U55 and Ethos-U65
- Lowered overall external SRAM and DRAM memory bandwidth usage
  - Improves system level energy efficiency
- TOSA compatible : Base inference profile (Int8 and Int16 Feature maps, Int8 Weights)
  - Transformer networks fully run on NPU without fallback to host CPU
  - Enables operator decomposition – Future proofing
- Can be paired with both Cortex-A and Cortex-M type CPUs
  - Flexibility and future proofing
  - Compatibility : Compiler frontend is same as previous generation




Cortex-M and Cortex-A compatible

# Typical Ethos-U85 data flow - same as for Ethos-U55/U65

0 An offline compiled command stream with corresponding compressed weights are put into system Flash/DRAM/SRAM.

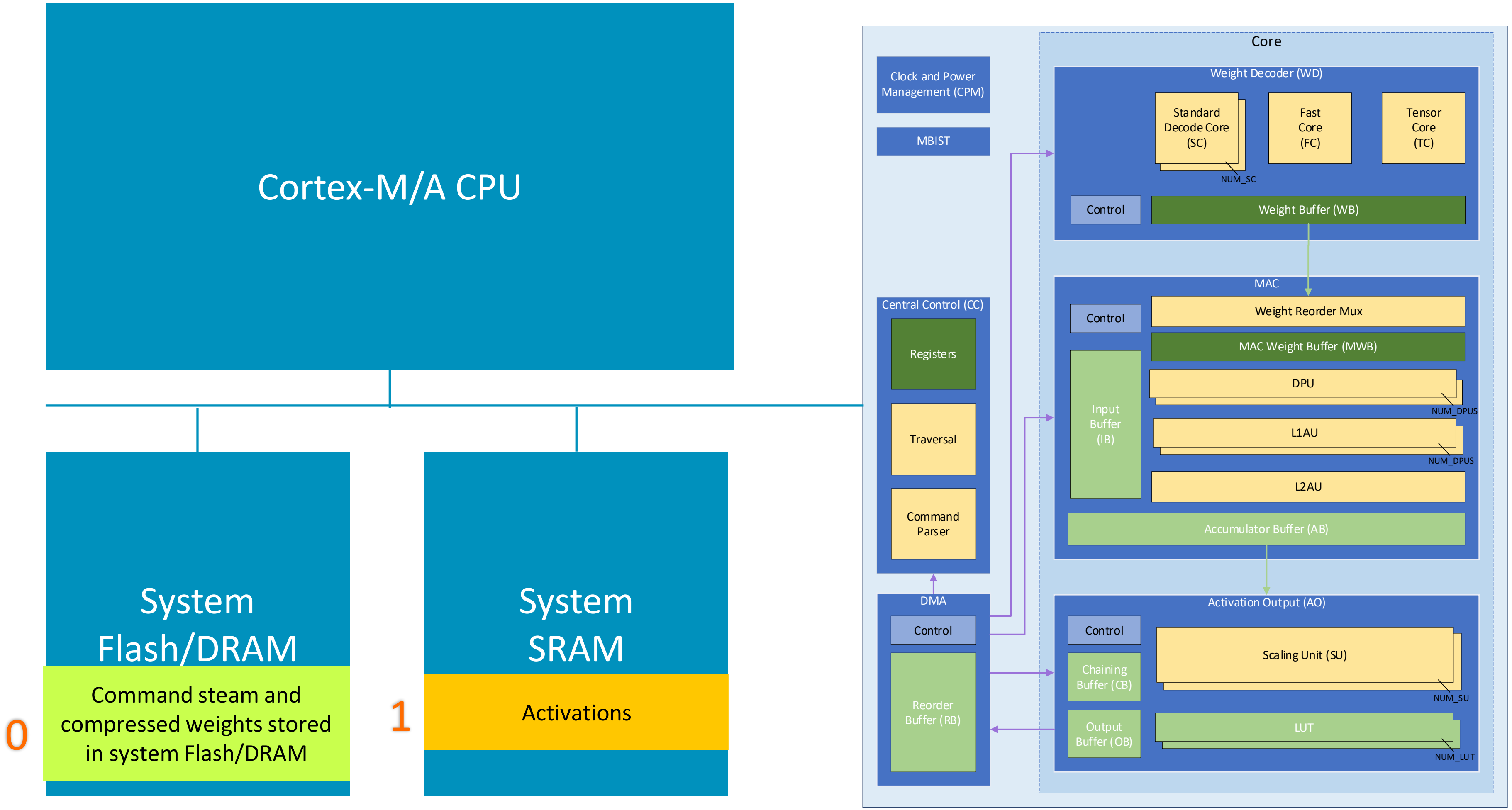


# Typical Ethos-U85 data flow - same as for Ethos-U55/U65

0 An offline compiled command stream with corresponding compressed weights are put into system Flash/DRAM/SRAM.

1 Input activations are put into system SRAM or DRAM.



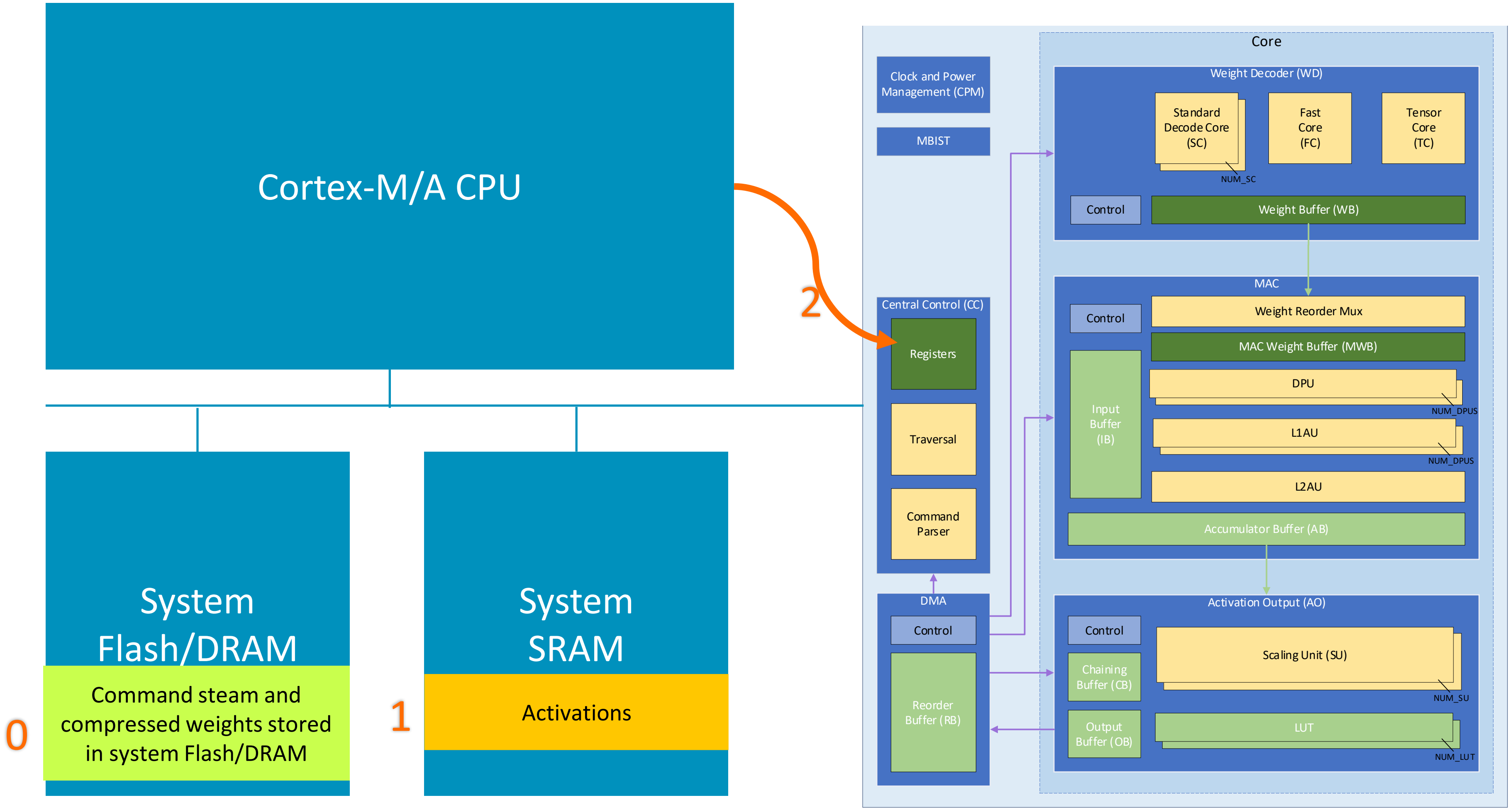

 

# Typical Ethos-U85 data flow - same as for Ethos-U55/U65

0 An offline compiled command stream with corresponding compressed weights are put into system Flash/DRAM/SRAM.

1 Input activations are put into system SRAM or DRAM.

2 The host starts Ethos-U85 by defining all memory regions to be used. In particular the location of the command stream and input activations.



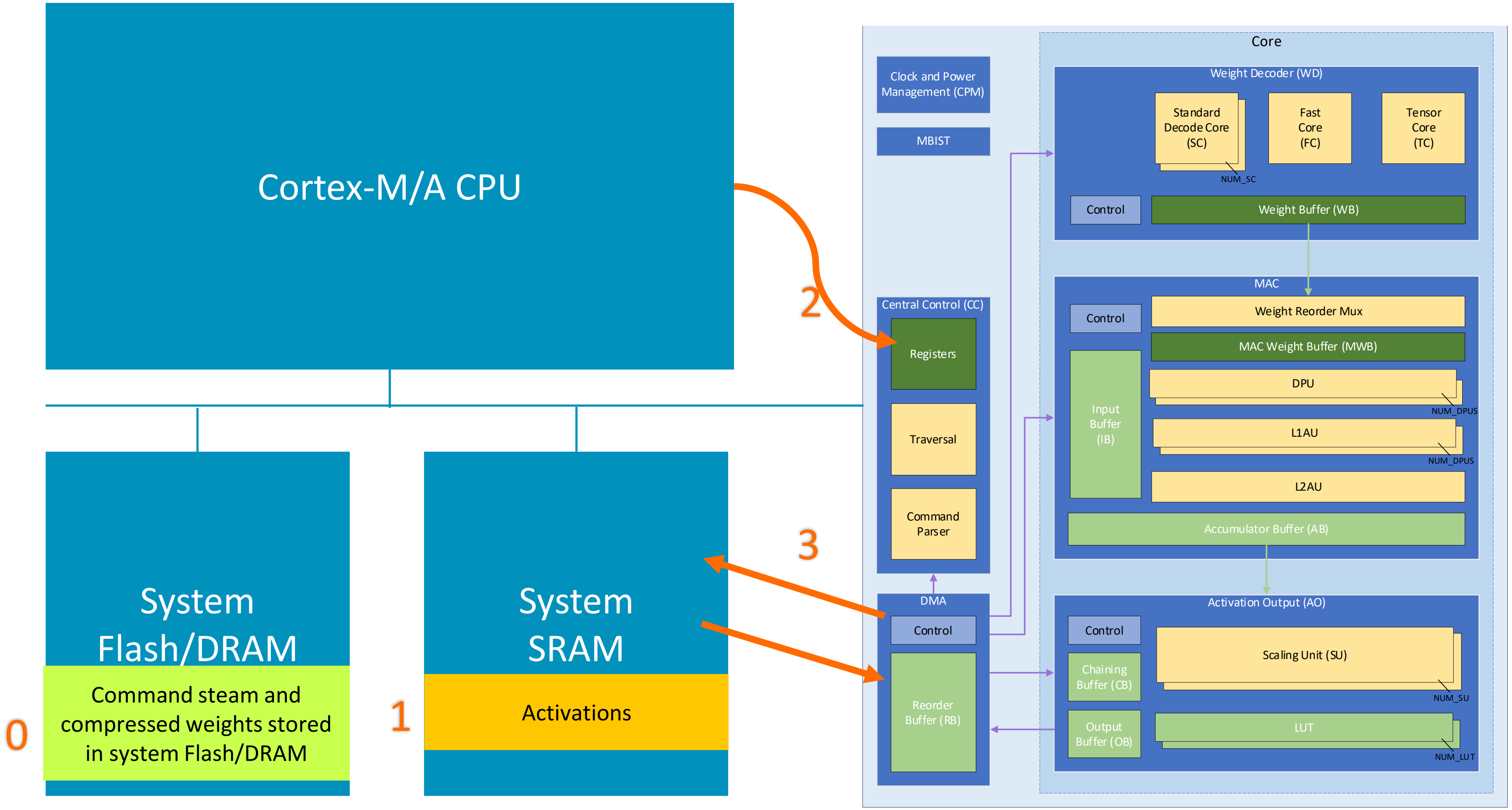
# Typical Ethos-U85 data flow - same as for Ethos-U55/U65

0 An offline compiled command stream with corresponding compressed weights are put into system Flash/DRAM/SRAM.

1 Input activations are put into system SRAM or DRAM.

2 The host starts Ethos-U85 by defining all memory regions to be used. In particular the location of the command stream and input activations.

3 Ethos-U85 autonomously runs all commands, using SRAM as a scratch buffer. Final results are written to a defined SRAM or DRAM buffer.



# Typical Ethos-U85 data flow - same as for Ethos-U55/U65

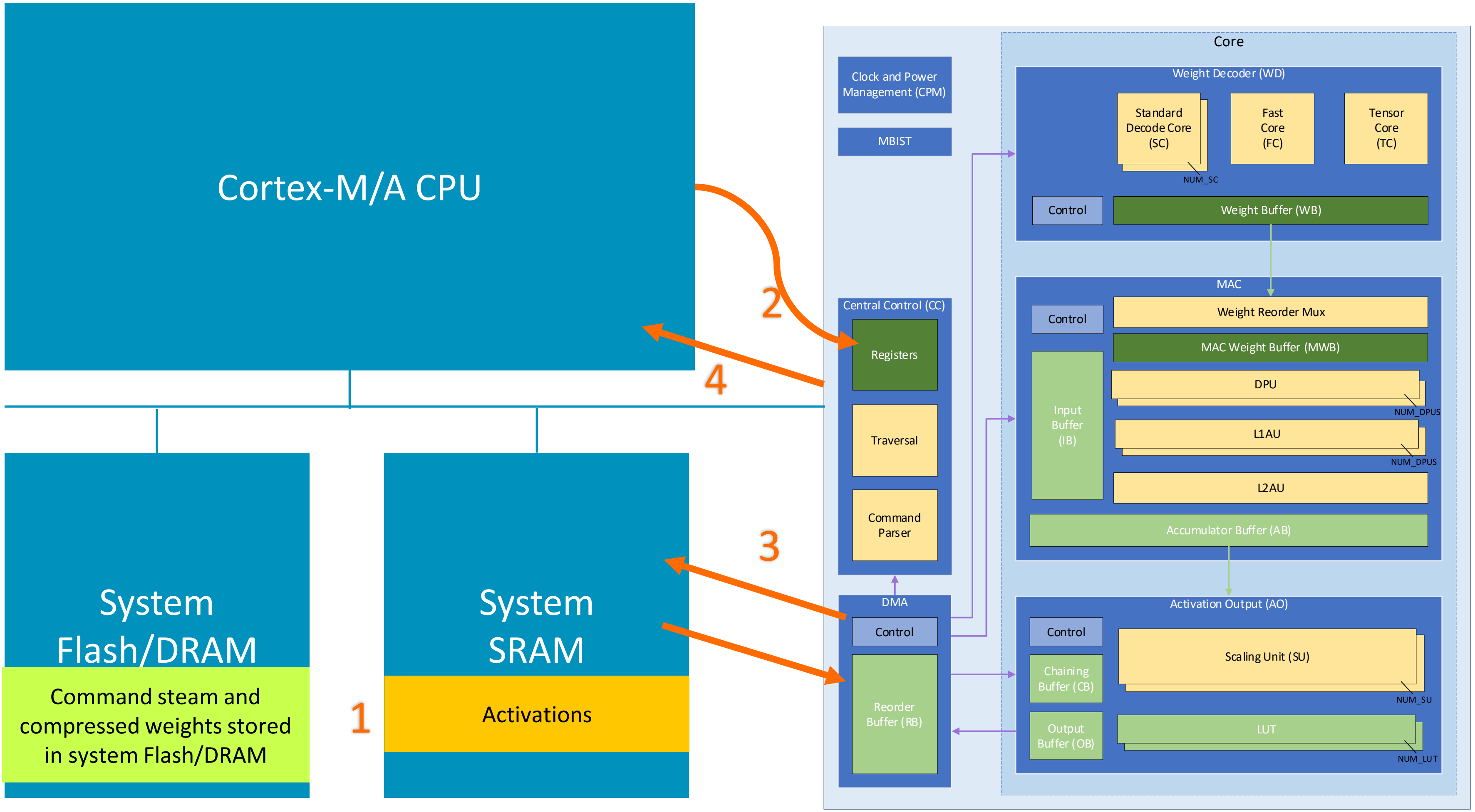
0 An offline compiled command stream with corresponding compressed weights are put into system Flash/DRAM/SRAM.

1 Input activations are put into system SRAM or DRAM.

2 The host starts Ethos-U85 by defining all memory regions to be used. In particular the location of the command stream and input activations.

3 Ethos-U85 autonomously runs all commands, using SRAM as a scratch buffer. Final results are written to a defined SRAM or DRAM buffer.

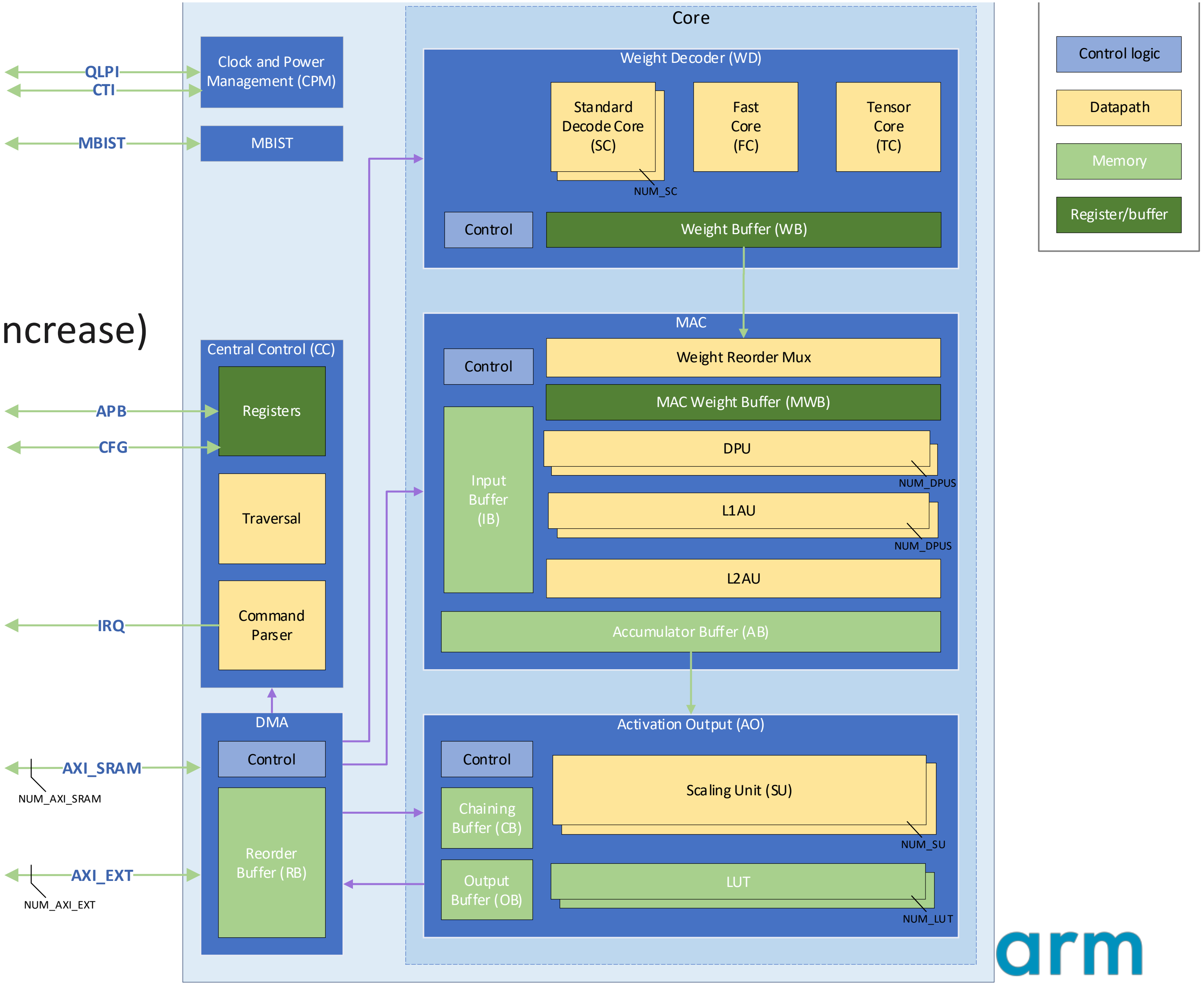
4 Interrupt on completion of writing the final result.



# Ethos-U85 Top level

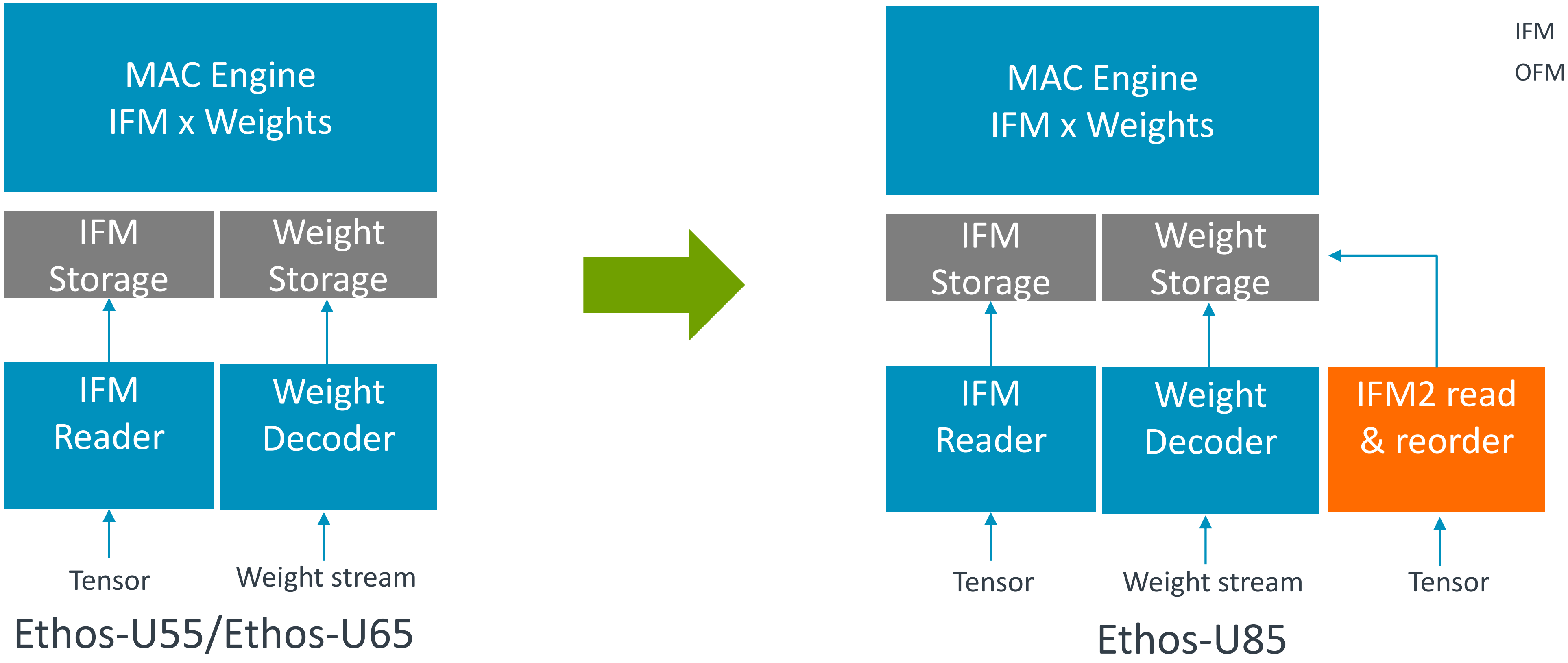
- 128/256/512/1024/2048
  8x8 MAC/cc configs
- Updated weight decode core
  - Tensor for MATMUL
  - Fast WD core for FC layers (up to 4x increase)
- Flexibility
  - New operators & Operator Chaining
  - Transformer networks support
  - HW support for 2/4 sparsity
    - Doubles MAC throughput
- Multiple AXI interfaces
  - SRAM & DRAM type of memories
  - Memory striping + Higher BW
- Higher energy efficiency
  - ~ 20 % better over U55/65



# Ethos-U85 Matrix Multiply Support: Updated WD core

- Ethos-U85 adds hardware to read a second IFM input into the MAC unit
  - This supports transposed matrix multiply (1x1 convolution) of two 2D tensors (1xWxC)
  - For constant weights (weight decoder) weights are compressed and reordered by the NN optimizer (Vela)




IFM : Input Feature Map

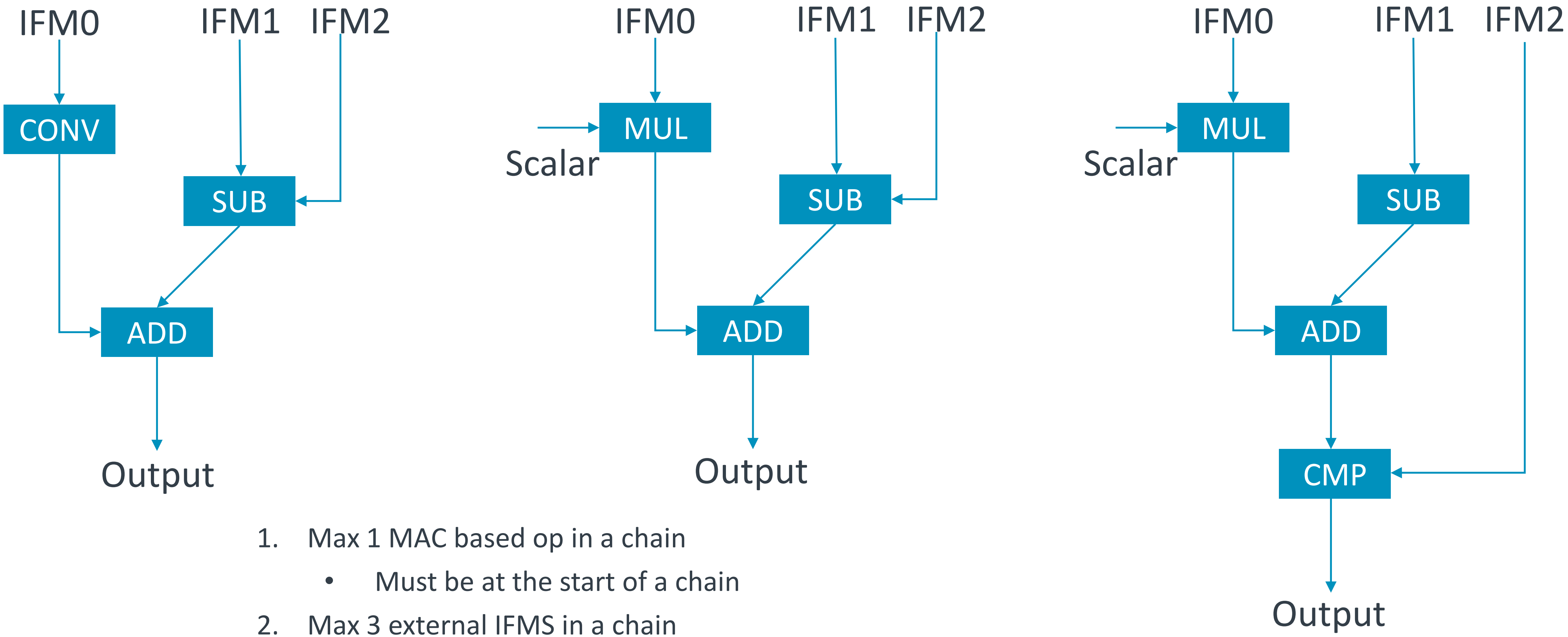
OFM : Output Feature Map

# Fast Weight Decoder Core

- Designed for high decode throughput
  - Needs good BW from AXI interfaces, ideally 32 bytes/cycle
  - Can be achieved with 2 striped AXI SRAM ports of 128-bit
  - RAW mode and Look-Up-Table (LUT) mode
    - RAW Mode = 32 Weights/cycle
    - LUT Mode = 64 Weights/cycle; Max 16 unique values
  - Targets fully connected layers that can fit in SRAM
    - for example : 128x128 Fully connected layer
      - 16384 weights needed
      - 512 CC with 32 w/cc, or 2048 CC with 8 w/cc
- Simulation results from RTL
  - With 1 Port and FWD ~1.9x improved in decode throughput over using Standard weight decoder
  - With 2 Ports ~3.5x improvement
  - With LUT mode and 2 ports ~6.2x improvement

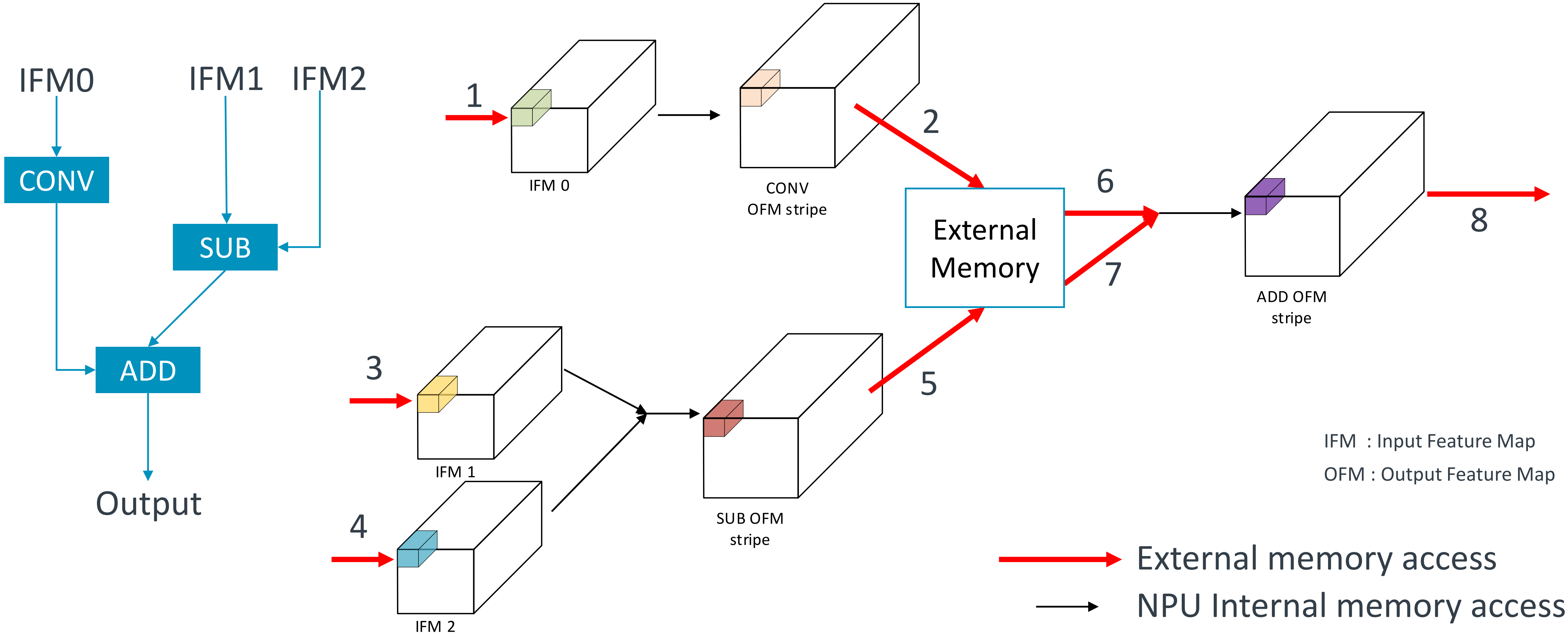
# Chaining of operators : Reduces external memory BW

## Examples of chained operators



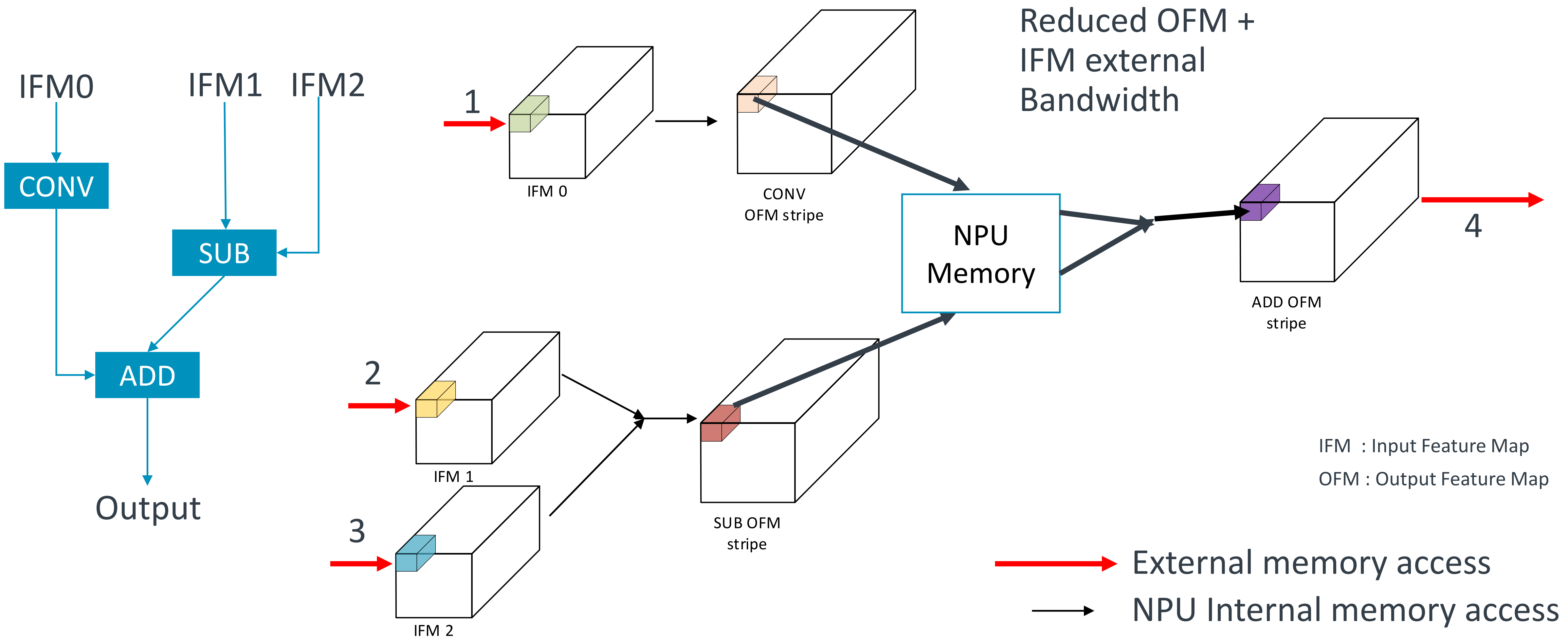

1. Max 1 MAC based op in a chain
   - Must be at the start of a chain
2. Max 3 external IFMS in a chain
3. Max 4 operators in a chain

# Chaining of operators : Un-chained flow




Numbers 1-8 indicate an example of the order of memory accesses needed

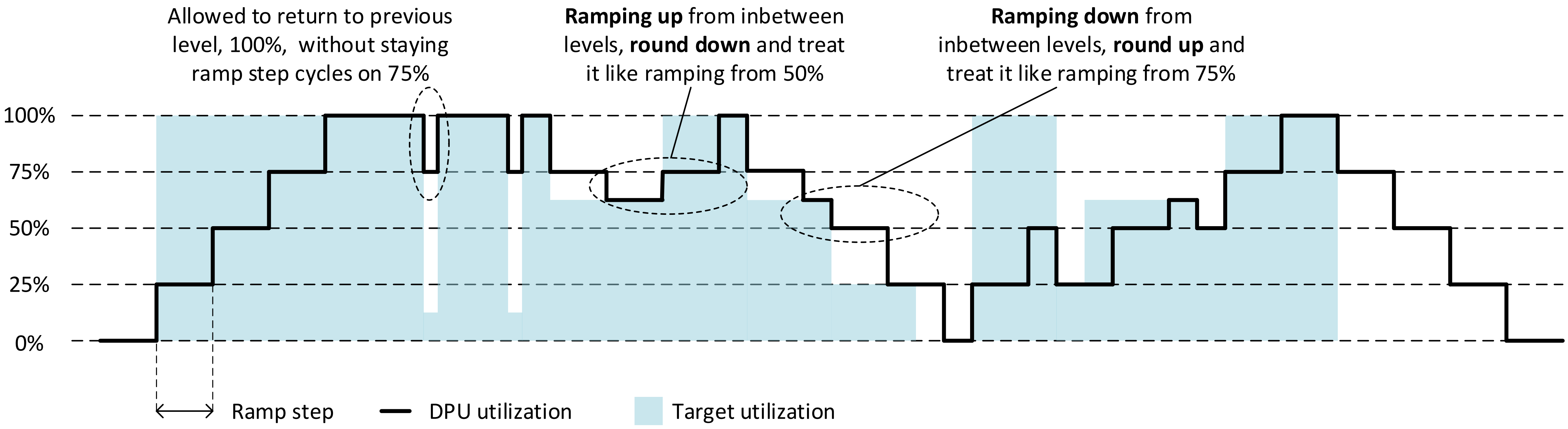
# Chaining of operators : Chained flow




Numbers 1-4 indicate an example of the order of memory accesses needed

# Power Ramping for higher MAC configuration

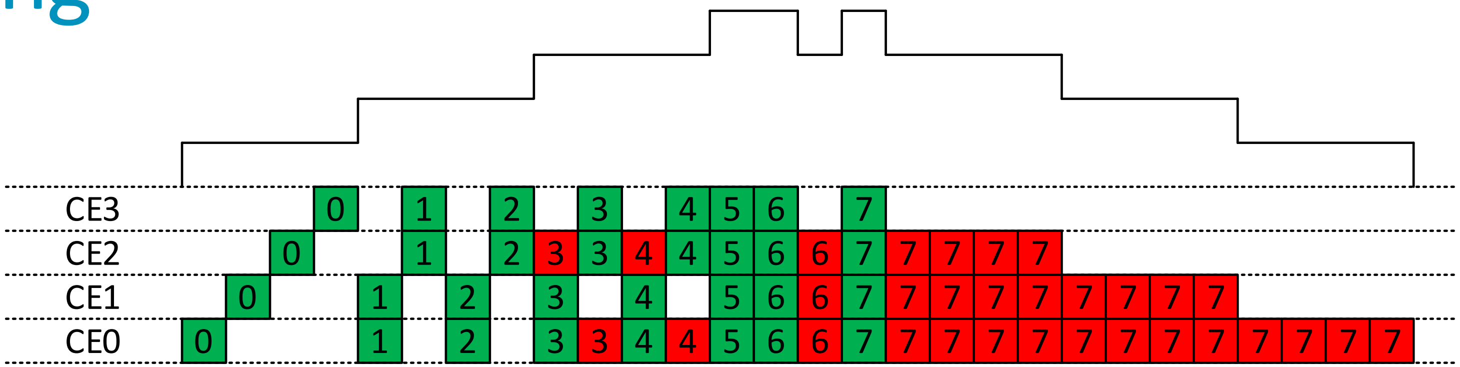
- High clock frequency and higher number of MAC compute engines
  - May result in large di/dt in certain scenarios : Requires extra effort in power planning
  - Avialable in 512, 1024 and 2048 MAC configs
- Ramping a quarter of MACs in each step instead of going to full utilization



# Ethos-U85 Power ramping

- Configurable number of clock cycles spent per step
- Logic inside the MAC engine
- Previously used data is reused by shuffling it during rampdown




Ramping disabled

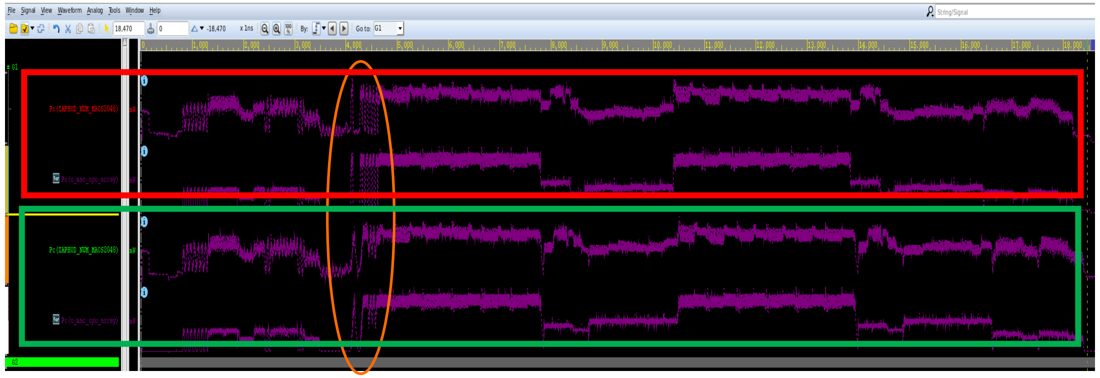

Ramping enabled

arm

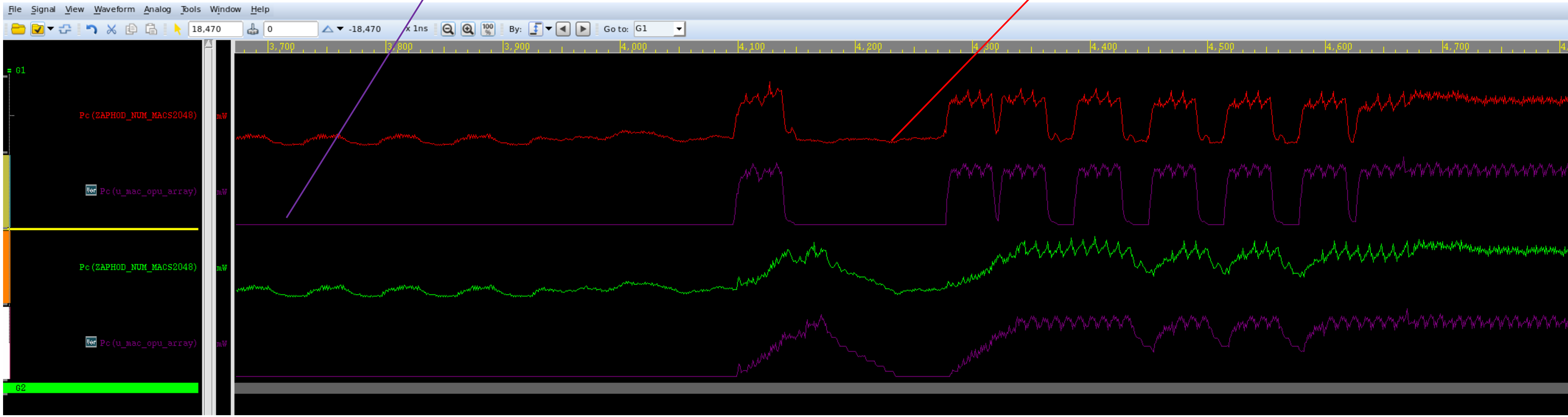
# Ethos-U85 Power ramping

- Lowers di/dt
- Minimal impact to overall inference/s



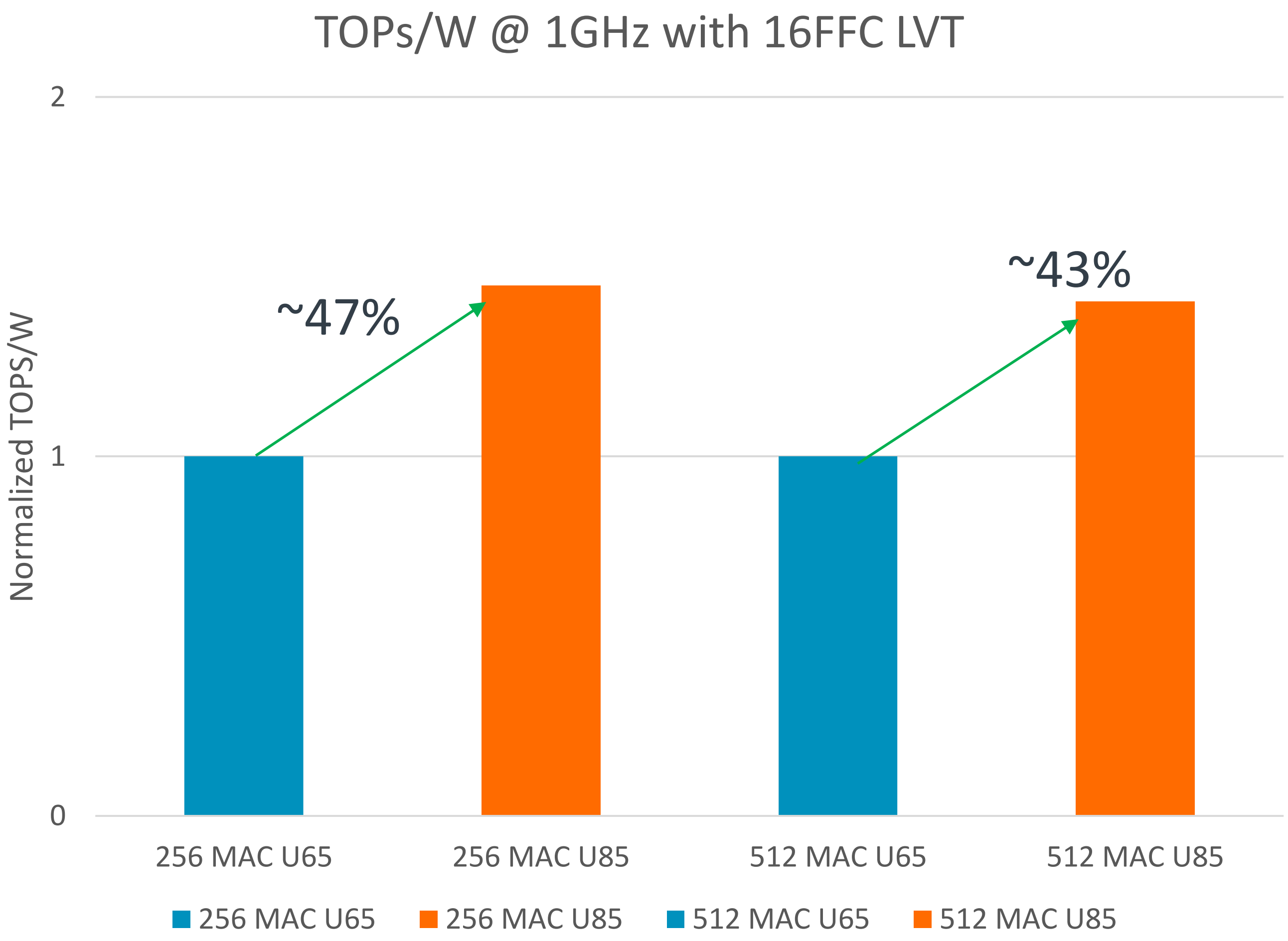
MAC array power

Top level power



# Ethos-U85 MAC Energy Efficiency

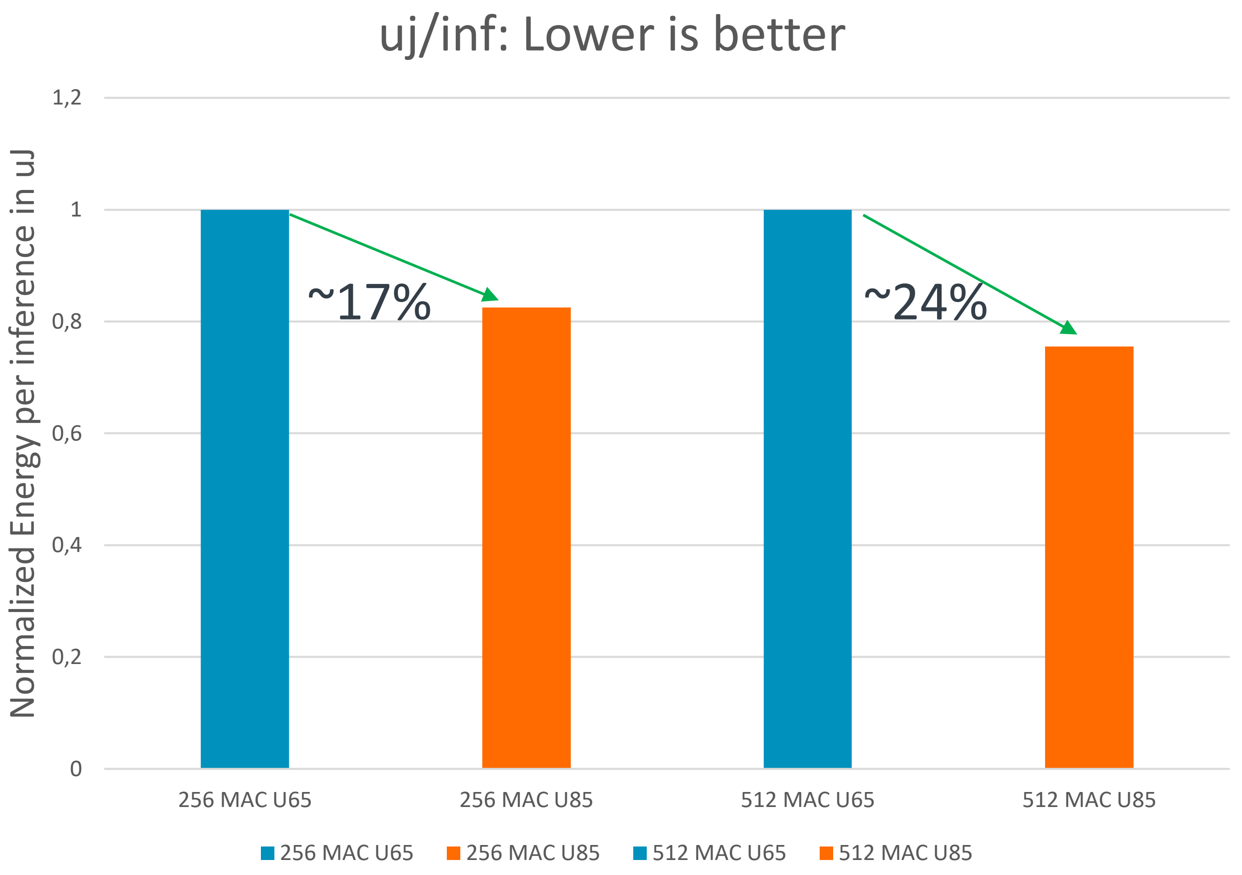
- Measured using a workload that used “normal” distributed weights and Feature map data
- Designed to reach around 99% MAC utilization
- Ethos-U85 compared to Ethos-U65 on similar workload shows up to 45% improvement in MAC energy efficiency




*All values given here are very early estimates and are subject to change.*

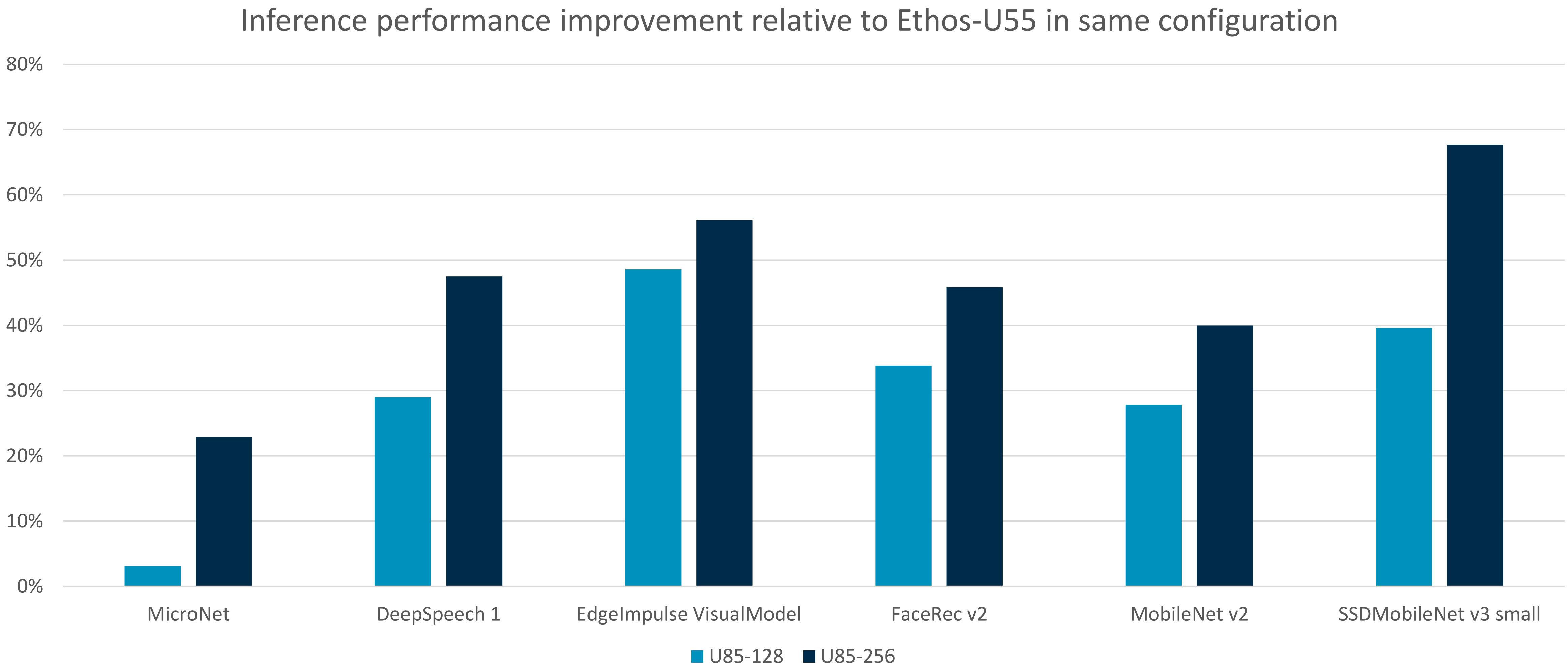
# Power Simulations in 16FF for Mobilenet-V2

- Comparison Runs with Same settings as U65 measurements
  - SRAM @ 16GB/s
  - DRAM @ 3.776 GB/s
  - Dedicated SRAM of 384kB
- U85-256 is 17% better than U65-256
- U85-512 is 24% better than U65-512




*All values given here are very early estimates and are subject to change.*

# Performance Report – CNN/RNN

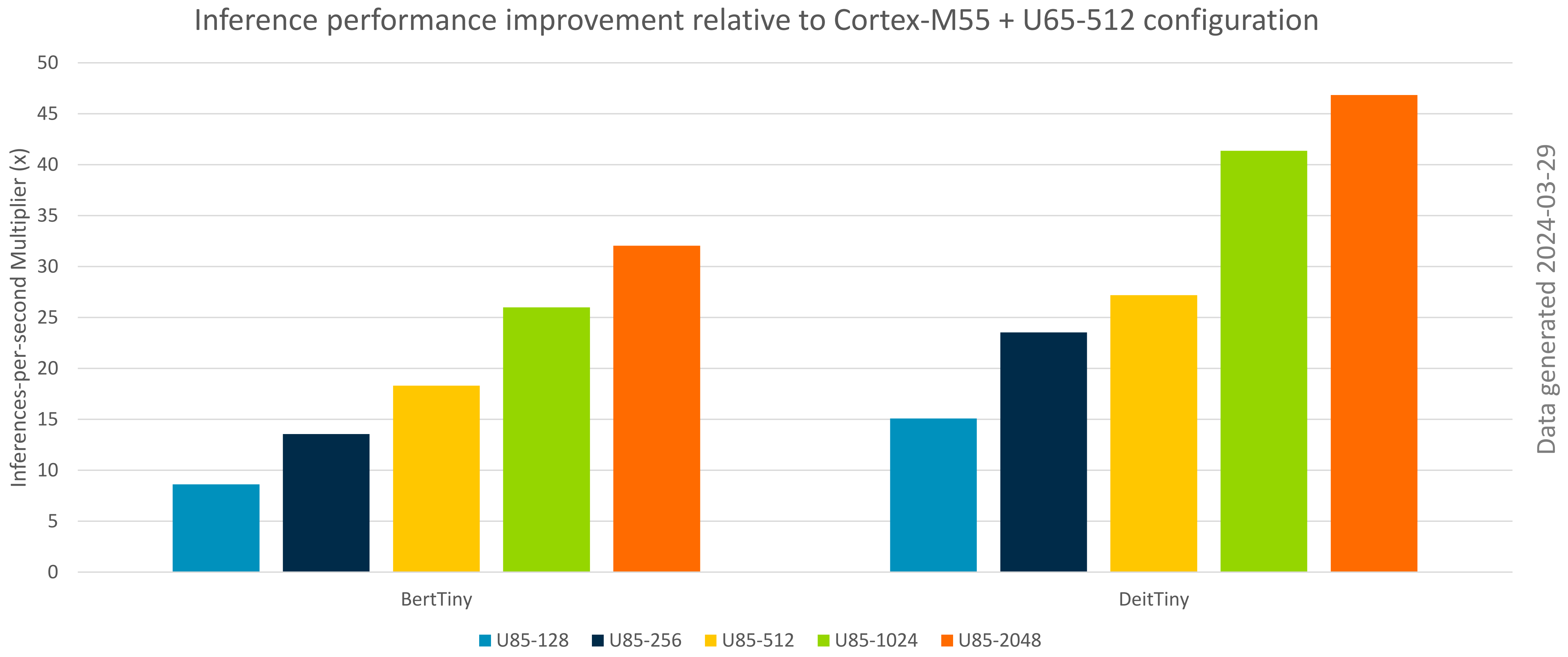



Data generated 2024-03-15




All numbers measured on FPGA with pre-EAC RTL and early software support

# Performance Report – Transformers




All numbers measured on FPGA with pre-EAC RTL and early software support

# Summary

- Ethos-U85 : The next generation of NPU from Arm
  - Configurable from 0.5 TOPs to 4 TOPs (@1GHz)
  - Easy attach to Cortex-A and Cortex-M systems
- TOSA compatibility for efficient ML processing at edge
  - New operators support end-to-end execution of transformer type networks
- Hardware features to improve energy efficiency
  - Lower SRAM/DRAM memory bandwidth
  - Optimized weight decoders for higher throughput in FC layers
- Power ramping support to ease implementation for higher TOPS configs
- Performance improvements over previous Ethos-U generation of NPUs

arm

Thank You
Danke
Gracias
Grazie
谢谢
ありがとう
Asante
Merci
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה
ధన్యవాదములు

# Copyright Notice



**www.tinyml.org**