“Empowering the Edge: Advancements in AI Hardware and In-Memory Computing Architectures for TinyML”

Nitin Chawla –Fellow and Director, ST Microelectronics

August 1, 2023
Thank you, tinyML Strategic Partners, for committing to take tinyML to the next Level, together
Executive Strategic Partners
Advancing AI research to make efficient AI ubiquitous

Power efficiency
- Model design, compression, quantization, algorithms, efficient hardware, software tools

Personalization
- Continuous learning, contextual, always-on, privacy-preserved, distributed learning

Efficient learning
- Robust learning through minimal data, unsupervised learning, on-device learning

Perception
- Object detection, speech recognition, contextual fusion

Reasoning
- Scene understanding, language understanding, behavior prediction

Action
- Reinforcement learning for decision making

A platform to scale AI across the industry

Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.
Accelerate Your Edge Compute

SYNTIANT
Making Edge AI A Reality

www.syntiant.com
Platinum Strategic Partners
DEPLOY VISION AI AT THE EDGE AT SCALE
Gold Strategic Partners
Witness potential made possible at analog.com.

Where what if becomes what is.
The Leading Development Platform for Edge ML

edgeimpulse.com
Driving decarbonization and digitalization. Together.

Infineon serving all target markets as Leader in Power Systems and IoT

www.infineon.com
NEUROMORPHIC INTELLIGENCE FOR THE SENSOR-EDGE

www.innatera.com
Renesas is enabling the next generation of AI-powered solutions that will revolutionize every industry sector.
STMicroelectronics provides extensive solutions to make tiny Machine Learning easy
ENGINEERING EXCEPTIONAL EXPERIENCES

We engineer exceptional experiences for consumers in the home, at work, in the car, or on the go.

www.synaptics.com
Silver Strategic Partners
Join Growing tinyML Communities:

**Meetup**

tinyML - Enabling ultra-low Power ML at the Edge

16k members in
49 Groups in 41 Countries

**LinkedIn**

The tinyML Community
https://www.linkedin.com/groups/13694488/

4k members
&
12.7k followers
Subscribe to tinyML YouTube Channel for updates and notifications (including this video)
www.youtube.com/tinyML
tinyML Asia
Technical Forum

November 16, 2023
Seoul, South Korea

Call for Presentations and Posters – Deadline August 7
https://www.tinyml.org/event/asia-2023/
2023 Edge AI Technology Report

The guide to understanding the state of the art in hardware & software in Edge AI.

https://www.wevolver.com/article/2023-edge-ai-technology-report
Nitin Chawla

Nitin Chawla is an ST Fellow and Director in the Technology R&D (Strategy and Innovation) organization at STMicroelectronics, where he leads the research initiatives in the area of Low Power Neural Networks and In-Memory computing architectures for Edge and Tiny ML applications. Nitin has a major in Electronic Circuits and Systems. He is an alumnus of Stanford University and holds a TRIZ diploma from the Massachusetts Institute Of Technology, Cambridge. He has served in different R&D and product organizations over the last 25 years. Before joining STMicroelectronics, he was the Chief Scientist of the HLS Product Division at Mentor Graphics Corporation based in Oregon, USA. Nitin has over 40+ US patents and more than 30 conference and journal publications.
Case Study: Digital SRAM In-Memory Computing Multi-Tiled Neural Processing Unit for Ultra Low Power Inference Applications

Nitin Chawla, Giuseppe Desoli and the ST “Orlando” Team:
• Introduction
• In Memory NPU architecture
• SRAM DIMC tile
• Silicon results
• Mapping strategies
• Inference examples
• Conclusions
• Introduction
• In Memory NPU architecture
• SRAM DIMC tile
• Silicon results
• Mapping strategies
• Inference examples
• Conclusions
AI Applications from Cloud to Tiny Machine Learning

- **Back-end Training (1x)**
  - Big training data
  - Big models
  - Fast iteration

- **Online/Cloud AI Processing (100x)**
  - Cloud AI ASIC
  - Analysis & recognition
  - Power demands

- **Intelligent Edge Devices (100,000x)**
  - SoC with NPU accelerators
  - Optimized algorithms and CNN-light

- **Intelligent Tiny Devices (1,000,000x)**
  - MCU with HW accelerators
  - Very tiny models & computation

**CNN:** Convolutional Neural Networks  
**NPU:** Neural Processing Unit  
**MCU:** Micro Controller Unit
### Deep Learning architecture: a Large space

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arm Ethos U-65</td>
<td>Dataflow</td>
</tr>
<tr>
<td>GreenWaves GAP 89</td>
<td>Dataflow</td>
</tr>
<tr>
<td>Intel Movidius Myriad X</td>
<td>Hybrid RISC-DSP-GPU</td>
</tr>
<tr>
<td>Mythic</td>
<td>Analog Flash IMC</td>
</tr>
<tr>
<td>Arm Ethos N78</td>
<td>Dataflow</td>
</tr>
<tr>
<td>Kneron KL720</td>
<td>Dataflow</td>
</tr>
<tr>
<td>Nvidia Jetson Orin</td>
<td>GPU</td>
</tr>
<tr>
<td>Renesas DRP-AI</td>
<td>Dataflow / reconfigurable</td>
</tr>
<tr>
<td>NXP Neutron eIQ</td>
<td>Dataflow</td>
</tr>
</tbody>
</table>

Deep Learning Architecture: a Large Space

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arm Ethos U-65</td>
<td>Dataflow</td>
</tr>
<tr>
<td>GreenWaves GAP 89</td>
<td>Dataflow</td>
</tr>
<tr>
<td>Intel Movidius Myriad X</td>
<td>Hybrid RISC-DSP-GPU</td>
</tr>
<tr>
<td>Mythic</td>
<td>Analog Flash IMC</td>
</tr>
<tr>
<td>Arm Ethos N78</td>
<td>Dataflow</td>
</tr>
<tr>
<td>Kneron KL720</td>
<td>Dataflow</td>
</tr>
<tr>
<td>Nvidia Jetson Orin</td>
<td>GPU</td>
</tr>
<tr>
<td>Renesas DRP-AI</td>
<td>Dataflow / reconfigurable</td>
</tr>
<tr>
<td>NXP Neutron eIQ</td>
<td>Dataflow</td>
</tr>
</tbody>
</table>

Our focus

Key factors for Deep Learning Hardware

- Compute Density and Energy Efficiency are key Figure of Merits (FOM).

**Diagram:***

- Processor Type
- Bit Precision Configuration
- Configurable Memory
- External Memory
- Memory Access BW/Efficiency

**Abbreviations:***

- CNN: Convolutional Neural Networks
- FCN: Fully Convolutional Networks
- RNN: Recurrent Neural Networks
- GAN: Generative Adversarial Networks
- BERT: Bidirectional Encoder Representation from Transformers
Examples of Embedded AI architectures

A 2.9 TOPS/W deep convolutional neural network SoC in FD-SOI 28nm for intelligent embedded systems,” ISSCC 2017 https://doi.org/10.1109/ISSCC.2017.7870349

MIT Eyeriss http://eyeriss.mit.edu/

NVIDIA NVDLA, http://nvdla.org

Google Coral edge TPU US20190050717A1
NPU roadmap: towards In Memory compute

In-memory compute taxonomy

**Analog IMC**
- Approximate BL accumulation
- Bit cell Vt variation limits row parallelism
- Readout throughput limited by ADC
- Approximate compute with complex BIST/Functional test screening

**Digital IMC**
- Deterministic and dataflow compatible
- Pushed rule/Logic Bitcell
- Duality of memory and computational modes
- Wide support of DVFS and Adaptive Body Bias
- Deterministic compute for DFT & Safety needs
NPU power consumption vs compute density

Key FOM: TOPS/W & TOPS/mm²

- These metrics vary based on:
  - **NPU architectural style**
    - data-flow, bit precision, sparsity support, weights compression
  - **Operating modes**
    - input/weights/output stationary
  - **Process technology**
    - 40nm to 12nm for typical edge@AI Socs

- **1 and 5 TOPS/W** for current MCUs process technology options

- **50-200 TOPS/W** expected to be needed in the next 5 years for AI@edge

Relative Power consumption for a typical NPU:
(1) System Level, (2) NPU + Mem, (3) NPU Core
• Introduction
• In Memory NPU architecture
• SRAM DIMC tile
• Silicon results
• Mapping strategies
• Inference examples
• Conclusions
System Components

- Cortex M Host
- 8 IMNPU Subsystems
- Peripherals to load Inputs
- External memory controllers
- Shared system memories for weight storage
- System Interconnect
IMNPU accelerator subsystem
Architectures support for chaining and tiling

2 tiles with PACK MUX

Two parallel IMC tiles generating packed and unpacked partial sum output data

All tiles chained together

One chain of 8 IMC tiles using filtered [filt] feature data without partial sum in data to generate partial sum output data

2 chains of 2 tiles with OUT MUX

Two parallel chains of 2 tiles using packed and filtered [filt] psum to generate interleaved activation output data

2 chains of 4 tiles with OUT MUX

Two parallel chains of 4 tiles using filtered [filt] feature data to generate interleaved activation output data
• Introduction
• In Memory NPU architecture
• **SRAM DIMC tile**
• Silicon results
• Mapping strategies
• Inference examples
• Conclusions
SRAM DIMC array segment arrangement

- 1R1W 8T bitcell based core array
- Read port decoupled from Write and provides good performance scaling, benefitting from body bias (BB) strategy of FD-SOI
- Better Vmin vs conventional 6T based SRAM
• Address decode scheme enabling 4 segments parallel access (e.g. 32 rows)
• Computation supports full tensor/sub tensor modes
• Unused tensor space can be gated
• Introduction
• In Memory NPU architecture
• SRAM DIMC tile
• **Silicon results**
• Mapping strategies
• Inference examples
• Conclusions
TOPS/W (4bW-4bF): Tile: 176 TOPS/W, IMNPU: 76 TOPS/W
TOPS/W (1bW-1bF): Tile: 770 TOPS/W, IMNPU: 310 TOPS/W

Better energy efficiency for sparse networks

Tile level energy performance gains diminish with data movement costs

1: \((X)S_{(Y)T}\) : \(X\) is the % sparsity in the Kernel data, \(Y\) is the % Inter Kernel transition density
FBB (Forward Body Bias) impact

- Compute density improves with FBB at fixed VDD
- TOPS/mm2 improves 1.8X with FBB across 0V to 1.5V
- TOPS/W degrades by only 14% across 1.5V FBB range
Technology 18nm FDSOI
Multi-Cluster IMNPU along with system interconnect: 4.2 mm²
Voltage range: 0.525-1.0V, FBB 0-1.5V
IMC Capacity 2 Mb
Computation: Deterministic
Precision Mode: 1-4 bits
229 TOPS (Peak Performance) 1 bit Weight - 1bit Feature
57 TOPS (Peak Performance) 4bit Weight - 4bit Feature
310 TOPS/W (1 bit)
77 TOPS/W (4 bit)
54 TOPS/mm² (1 bit)
13.6 TOPS/mm² (4 bit)
CNN, LSTM, RNN
• Introduction
• In Memory NPU architecture
• SRAM DIMC tile
• Silicon results
• **Mapping strategies**
• Inference examples
• Conclusions
Mapping strategies and optimization for IMC

- Quantization 1-8 bits (fixed point today, possibly block scaling next)
- Weight compression/on-the-fly decompression
- Feature-maps compression/decompression
- Structured sparsity
- Layer fusion
- Layer slicing and partitioning: increase parallelism, bandwidth/memory footprint reduction
  - Kernelwise
  - Depthwise
  - Striping
  - Striding
- Kernel and feature broadcasting, layout optimization, and reloads reduction

IMC can deliver massive amounts of OPS/cycle if data movement is reduced → major bottleneck
Mixed quantization precision example: VGG16 tiny

<table>
<thead>
<tr>
<th>shape</th>
<th>Kernels</th>
<th>A/W bits</th>
<th>A/W bits</th>
<th>Feat comp. (*)</th>
<th>Weights comp.</th>
<th>Total comp. perf</th>
</tr>
</thead>
<tbody>
<tr>
<td>3x112x112</td>
<td>32x3x3</td>
<td>8/8</td>
<td>8/8</td>
<td>1x</td>
<td>1x</td>
<td>~7x 2.1x</td>
</tr>
<tr>
<td>32x112x112</td>
<td>64x3x3</td>
<td>8/8</td>
<td>4/4</td>
<td>2x</td>
<td>2x</td>
<td></td>
</tr>
<tr>
<td>64x56x56</td>
<td>112x3x3</td>
<td>8/8</td>
<td>4/4</td>
<td>2x</td>
<td>2x</td>
<td></td>
</tr>
<tr>
<td>112x28x28</td>
<td>224x3x3</td>
<td>8/8</td>
<td>2/2</td>
<td>4x</td>
<td>4x</td>
<td></td>
</tr>
<tr>
<td>224x7x7</td>
<td>224x3x3</td>
<td>8/8</td>
<td>1/1</td>
<td>8x</td>
<td>8x</td>
<td></td>
</tr>
</tbody>
</table>

(*) actual feature map compression and throughput reduction depend on mapping.
Layer partitioning: chaining & striping

Weights broadcasting DMAs

separate DMA streams

Same kernels
Layer partitioning: chaining & kernelwise

- separate broadcasting DMA streams
- Weights shared DMAs
- separate DMA streams
- different kernels

Layer partitioning:
- Chaining & kernelwise
Layer partitioning: chaining & kernelwise

- separate broadcasting DMA streams
- IMC Tile 1
- IMC Tile 2
- IMC Tile 3
- IMC Tile 4
- IMC Tile 5
- IMC Tile 6

Weights shared DMAs

- shared DMA stream

Output layout flexibility is very important to avoid additional costs

- different kernels
- dma

Layer partitioning:
- chaining & kernelwise

Weights shared DMAs

- shared DMA stream

Output layout flexibility is very important to avoid additional costs
• Introduction
• In Memory NPU architecture
• SRAM DIMC tile
• Silicon results
• Mapping strategies
• Inference examples
• Conclusions
### VGG16 style network mapping

<table>
<thead>
<tr>
<th>layer</th>
<th>CxHxW</th>
<th>no of kernels</th>
<th>no of params</th>
<th>activation (bytes)</th>
<th>weights (bytes)</th>
<th>kernel (bits)</th>
<th>MMACs</th>
<th>stripes</th>
<th>chains parallel</th>
<th>IMC utilization</th>
<th>kernel rounds</th>
<th>cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1_1</td>
<td>3x112x112</td>
<td>32</td>
<td>896</td>
<td>18816</td>
<td>432</td>
<td>108</td>
<td>10.84</td>
<td>8,1,1</td>
<td></td>
<td>11%</td>
<td>1</td>
<td>5346</td>
</tr>
<tr>
<td>C1_2</td>
<td>32x112x112</td>
<td>64</td>
<td>18496</td>
<td>200704</td>
<td>9216</td>
<td>1152</td>
<td>231.21</td>
<td>4,2,1</td>
<td></td>
<td>56%</td>
<td>2</td>
<td>114048</td>
</tr>
<tr>
<td>C2_1</td>
<td>64x56x56</td>
<td>64</td>
<td>36928</td>
<td>100352</td>
<td>18432</td>
<td>2304</td>
<td>115.61</td>
<td>2,3,1</td>
<td></td>
<td>56%</td>
<td>2</td>
<td>77568</td>
</tr>
<tr>
<td>C2_2</td>
<td>64x56x56</td>
<td>112</td>
<td>64624</td>
<td>100352</td>
<td>32256</td>
<td>2304</td>
<td>202.31</td>
<td>1,3,2</td>
<td></td>
<td>56%</td>
<td>2</td>
<td>135744</td>
</tr>
<tr>
<td>C3_1</td>
<td>112x28x28</td>
<td>112</td>
<td>113008</td>
<td>43904</td>
<td>56448</td>
<td>4032</td>
<td>88.51</td>
<td>1,4,2</td>
<td></td>
<td>98%</td>
<td>2</td>
<td>50274</td>
</tr>
<tr>
<td>C3_2</td>
<td>112x28x28</td>
<td>112</td>
<td>113008</td>
<td>43904</td>
<td>56448</td>
<td>4032</td>
<td>88.51</td>
<td>1,4,2</td>
<td></td>
<td>98%</td>
<td>2</td>
<td>50274</td>
</tr>
<tr>
<td>C3_3</td>
<td>112x28x28</td>
<td>224</td>
<td>226016</td>
<td>43904</td>
<td>112896</td>
<td>4032</td>
<td>177.02</td>
<td>1,4,2</td>
<td></td>
<td>98%</td>
<td>4</td>
<td>100548</td>
</tr>
<tr>
<td>C4_1</td>
<td>224x14x14</td>
<td>224</td>
<td>451808</td>
<td>21952</td>
<td>225792</td>
<td>8064</td>
<td>88.51</td>
<td>1,8,1</td>
<td></td>
<td>98%</td>
<td>7</td>
<td>71442</td>
</tr>
<tr>
<td>C4_2</td>
<td>224x14x14</td>
<td>224</td>
<td>451808</td>
<td>21952</td>
<td>225792</td>
<td>8064</td>
<td>88.51</td>
<td>1,8,1</td>
<td></td>
<td>98%</td>
<td>7</td>
<td>71442</td>
</tr>
<tr>
<td>C4_3</td>
<td>224x14x14</td>
<td>224</td>
<td>451808</td>
<td>21952</td>
<td>225792</td>
<td>8064</td>
<td>88.51</td>
<td>1,8,1</td>
<td></td>
<td>98%</td>
<td>7</td>
<td>71442</td>
</tr>
<tr>
<td>C5_1</td>
<td>224x7x7</td>
<td>224</td>
<td>451808</td>
<td>5488</td>
<td>225792</td>
<td>8064</td>
<td>22.13</td>
<td>1,8,1</td>
<td></td>
<td>98%</td>
<td>7</td>
<td>39029</td>
</tr>
<tr>
<td>C5_2</td>
<td>224x7x7</td>
<td>224</td>
<td>451808</td>
<td>5488</td>
<td>225792</td>
<td>8064</td>
<td>22.13</td>
<td>1,8,1</td>
<td></td>
<td>98%</td>
<td>7</td>
<td>39029</td>
</tr>
<tr>
<td>C5_3</td>
<td>224x7x7</td>
<td>224</td>
<td>451808</td>
<td>5488</td>
<td>225792</td>
<td>8064</td>
<td>22.13</td>
<td>1,8,1</td>
<td></td>
<td>98%</td>
<td>7</td>
<td>39029</td>
</tr>
</tbody>
</table>

(1) Estimated assuming additional striping and kernels broadcasting, kernel load cycles included
(2) kernels randomly chosen with 50% sparsity

---

### Configuration

<table>
<thead>
<tr>
<th></th>
<th>1 cluster</th>
<th>8 clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACS/inf</td>
<td>1.25E+09</td>
<td></td>
</tr>
<tr>
<td>cycles/inf</td>
<td>865214</td>
<td>125618</td>
</tr>
<tr>
<td>Inf/sec</td>
<td>693</td>
<td>4776</td>
</tr>
<tr>
<td>TOPS/W²</td>
<td>46.8</td>
<td></td>
</tr>
</tbody>
</table>

Measured at 0.525V and 600MHz with 1.5v FBB
**YOLO2 tiny style mapping example**

<table>
<thead>
<tr>
<th>layer</th>
<th>CxHxW</th>
<th>no of kernels</th>
<th>no of params</th>
<th>activation (bytes)</th>
<th>weights (bytes)</th>
<th>kernel (bits)</th>
<th>MMACs</th>
<th>stripes chains parallel</th>
<th>IMC utilization</th>
<th>kernel rounds</th>
<th>cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>3x416x416</td>
<td>16</td>
<td>448</td>
<td>259584</td>
<td>216</td>
<td>108</td>
<td>74.76</td>
<td>8,1,1</td>
<td>5%</td>
<td>1</td>
<td>36531</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4590</td>
</tr>
<tr>
<td>C2</td>
<td>16x208x208</td>
<td>32</td>
<td>4640</td>
<td>346112</td>
<td>2304</td>
<td>576</td>
<td>199.36</td>
<td>8,1,1</td>
<td>56%</td>
<td>1</td>
<td>97632</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12456</td>
</tr>
<tr>
<td>C3</td>
<td>32x104x104</td>
<td>64</td>
<td>18496</td>
<td>173056</td>
<td>9216</td>
<td>1152</td>
<td>199.36</td>
<td>2,2,2</td>
<td>56%</td>
<td>1</td>
<td>98496</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>13320</td>
</tr>
<tr>
<td>C4</td>
<td>64x52x52</td>
<td>112</td>
<td>64624</td>
<td>86528</td>
<td>32256</td>
<td>2304</td>
<td>174.44</td>
<td>1,3,2</td>
<td>56%</td>
<td>2</td>
<td>117600</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16212</td>
</tr>
<tr>
<td>C5</td>
<td>112x26x26</td>
<td>224</td>
<td>226016</td>
<td>37856</td>
<td>112896</td>
<td>4032</td>
<td>152.64</td>
<td>1,4,2</td>
<td>98%</td>
<td>4</td>
<td>88641</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12844</td>
</tr>
<tr>
<td>C6_1</td>
<td>224x13x13</td>
<td>112</td>
<td>225904</td>
<td>18928</td>
<td>112896</td>
<td>8064</td>
<td>38.16</td>
<td>1,8,1</td>
<td>98%</td>
<td>4</td>
<td>32744</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5857</td>
</tr>
<tr>
<td>C6_2</td>
<td>224x13x13</td>
<td>112</td>
<td>225904</td>
<td>18928</td>
<td>112896</td>
<td>8064</td>
<td>38.16</td>
<td>1,8,1</td>
<td>98%</td>
<td>4</td>
<td>32744</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5857</td>
</tr>
<tr>
<td>C7</td>
<td>224x7x7</td>
<td>224</td>
<td>451808</td>
<td>5488</td>
<td>112896</td>
<td>4032</td>
<td>22.13</td>
<td>1,4,2</td>
<td>98%</td>
<td>4</td>
<td>19514</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4879</td>
</tr>
<tr>
<td>C8</td>
<td>224x7x7</td>
<td>512</td>
<td>1E+06</td>
<td>5488</td>
<td>258048</td>
<td>4032</td>
<td>50.58</td>
<td>1,4,2</td>
<td>98%</td>
<td>8</td>
<td>44604</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5576</td>
</tr>
<tr>
<td>C9</td>
<td>512x7x7</td>
<td>30</td>
<td>15390</td>
<td>12544</td>
<td>69120</td>
<td>2048</td>
<td>0.75</td>
<td>1,2,4</td>
<td>100%</td>
<td>1</td>
<td>9008</td>
</tr>
</tbody>
</table>

**Configuration**

<table>
<thead>
<tr>
<th></th>
<th>1 cluster</th>
<th>8 clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACS/inf</td>
<td>9.5E+08</td>
<td></td>
</tr>
<tr>
<td>cycles/inf</td>
<td>577514</td>
<td>90598</td>
</tr>
<tr>
<td>Inf/sec</td>
<td>1039</td>
<td>6623</td>
</tr>
<tr>
<td>TOPS/W²</td>
<td>50.86</td>
<td></td>
</tr>
</tbody>
</table>

Measured at 0.525V and 600MHz with 1.5v FBB

(1) Estimated assuming additional striping and kernels broadcasting, kernel load cycles included
(2) Kernels randomly chosen with 50% sparsity
### Battery-operated device for video surveillance

<table>
<thead>
<tr>
<th>Configuration</th>
<th>MACS/Inference</th>
<th>In/s</th>
<th>IMNPU Power</th>
<th>Total Power</th>
<th>Battery endurance² (1/100 duty cycle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 cluster @ 10 MHz, 0.0V FBB Always-ON, VGG like</td>
<td>1.25 GOPS</td>
<td>10</td>
<td>267 uW</td>
<td>567 uW</td>
<td>363 days</td>
</tr>
<tr>
<td>8 clusters @ 400MHz, 0.3V FBB Post Wakeup, 10x complexity</td>
<td>12.5 GOPS</td>
<td>30</td>
<td>8.0 mW</td>
<td>12.0 mW</td>
<td></td>
</tr>
</tbody>
</table>

(1) Estimated power includes a portion of shared memory, IOs, clock, and external sensor interface, weights stored in ePCM on chip

(2) 6000 mA/h battery capacity assumed (e.g., 2 AA 1.5v batteries)

---

In Memory NPU sub-system example, fixed Vdd, multiple Body Bias island
• Introduction
• In Memory NPU architecture
• SRAM DIMC tile
• Silicon results
• Mapping strategies
• Inference examples
• Summary
Conclusions

- Embedded NPUs are enabling efficient NN inference on the edge
- In Memory Computing is a key enabler to achieving higher compute density and energy efficiency: our results in 18nm FD-SOI show up to 50x improvements compared to pure digital logic
- DIMC-based NPU maintains deterministic computation → general-purpose
- Dedicated compilation and optimization tools are key to efficiently mapping the NN computations on these architectures
Our technology starts with You

Find out more at www.st.com
Copyright Notice

This multimedia file is copyright © 2023 by tinyML Foundation. All rights reserved. It may not be duplicated or distributed in any form without prior written approval.

tinyML® is a registered trademark of the tinyML Foundation.

www.tinyml.org
Copyright Notice

This presentation in this publication was presented as a tinyML® Talks webcast. The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyml.org