**TNPU:** Supporting Trusted Execution with Tree-less Integrity Protection for Neural Processing Unit

**Sunho Lee**, Jungwoo Kim, Seonjin Na, Jongse Park, and Jaehyuk Huh
Vulnerabilities of integrated NPU

- NPU is widely used in the form of System-on-a-Chip.
**Trusted Execution Environment (CPU)**

- Access control
- Counter-based memory protection
Trusted Execution Environment (NPU)

- CPU: On-chip hardware and related software
- TNPU: + NPU-related hardware/software

1) Access control, 2) Memory Protection for NPU
Validate Access from NPU

- Access control
  - CPU MMU: Traditional validation table
  - NPU IOMMU

![Diagram](image-url)
Validate Access from NPU

- **Access control:** Extended validation table (EEPCM)
  - CPU MMU: Traditional validation entries
  - NPU IOMMU: Additional validation entries
Naive Memory Protection to NPU

- Memory protection
  - Counter-based encryption & integrity protection
  - **Counter Freshness Validation**

### Advantage of counter caching
1. Bandwidth saving
2. Traffic of the security engine
Naive Memory Protection to NPU

- Average **19.2%** performance degradation
- Reason: Counter-cache miss rate (**7.9%**)

A novel memory protection technique for NPU is necessary!

![Graph showing normalized execution time and counter cache miss rates for Small NPU, Large NPU, and Average](image-url)
NPU Execution Model

- Execution: \(*mvin \rightarrow \text{preload} \rightarrow \text{compute} \rightarrow **mvout*
  - The **software** controls NPU data movement by commands

*\text{*mvin: move-in, **mvout: move-out, ***SPM: Scratchpad Memory*}
Tensor-based Computing

- Tensor-granular computation
  - **Per-tensor version number** is sufficient: Tensor-unit memory access

Residual Block (Resnet 50)
Tree-less Integrity Protection

- Counter → **Version number** controlled by **software**
  - Security granularity: Cacheline → **Tensor**
  - Storage requirement: Only **0.14KB** on average

Problem: NPU executes layer operation at once? (i.e. Many large tensors are not fitted into SPM)
Challenge: Intra-layer Computing

- Tensor → One or multiple tiles for intra-layer computing

Tile-granular version number is necessary in intra-layer!
Tile-granular Version Number

- Tensor → One or multiple tiles for intra-layer computing

 Tensor A \[\times\] Tensor B = Tensor C

Version Number

$\begin{array}{cc}
0 & 0 \\
0 & 0 \\
\end{array}$

$\begin{array}{c}
2 \\
\end{array}$
Tensor/Tile Version Number

- Tensor/Tile version number
  - Granularity: Cacheline → Tensor/Tile (Intra-layer)
  - Storage requirement: Only 1.3KB on average

- expand, merge: Granularity translation operation

<table>
<thead>
<tr>
<th>Tensor</th>
<th>Tile</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tensor</th>
<th>Tile</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>
**Evaluation Environment**

- Cycle-level simulation modified from *SCALE-Sim*
- Two edge-level system-on-a-chip configurations
  - Samsung Exynos 990 (Small NPU), ARM Ethos N77 (Large NPU)
- Workloads: 14 models in MLPerf, DeepBench

<table>
<thead>
<tr>
<th></th>
<th>Small NPU (Samsung Exynos 990)</th>
<th>Large NPU (ARM Ethos N77)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PE</td>
<td>32 x 32</td>
<td>45 x 45</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>11 GB/s (4 channels)</td>
<td>22 GB/s (4 channels)</td>
</tr>
<tr>
<td>Frequency</td>
<td>2.75 GHz (both processor/memory)</td>
<td>1 GHz (both processor/memory)</td>
</tr>
<tr>
<td>SPM</td>
<td>480KB in total</td>
<td>1MB in total</td>
</tr>
<tr>
<td>Precision</td>
<td>Float16</td>
<td>Float16</td>
</tr>
</tbody>
</table>

* A systematic methodology for characterizing scalability of DNN accelerators using SCALE-Sim (ISPASS 2020)
Evaluation Result (Single NPU)

- Performance improvement: **8.75%**
  - Data traffic reduction: **7.67%**
- Remaining performance degradation: **8.80%** (Comp. Unsecure)
  - Stored-hash-value (Message-authentication-code; MAC)

![Bar chart showing performance results for Small and Large NPUs with categories: goo, mob, yt, alex, rcnn, df, med, tx, res, agz, sent, ds2, tf, ncf, avg.](chart.png)

<table>
<thead>
<tr>
<th>Category</th>
<th>Small NPU</th>
<th>Large NPU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bandwidth</td>
<td>bandwidth</td>
</tr>
<tr>
<td></td>
<td>counter-cache</td>
<td>counter-cache</td>
</tr>
<tr>
<td></td>
<td>33.30%</td>
<td>22.20%</td>
</tr>
<tr>
<td></td>
<td>25.64%</td>
<td>21.13%</td>
</tr>
<tr>
<td></td>
<td>1.090</td>
<td>1.086</td>
</tr>
</tbody>
</table>

Legend:
- Unsecure
- Baseline
- TNPU
Evaluation Result (Multiple NPUs)

- Scalability: Slope (TNPU) < Slope (Baseline)
- Performance improvement: 8.75% → 11%
Summary

▪ Result
  ▪ Trusted Execution environment for NPU
  ▪ Performance improvement: 8.75% (single), 11% (3-NPU)

▪ Challenge
  ▪ Counter tree overhead

▪ Idea
  ▪ Counter → Tensor/tile-granular version number

▪ Further Work
  ▪ Stored-hash-value (MAC) optimization
Thank you