ASIA SOUTH PACIFIC DESIGN AUTOMATION CONFERENCE

ASP-DAC 2024

29th Asia and South Pacific Design Automation Conference

A Precision-Scalable RISC-V DNN Processor with On-Device Learning Capability at the Extreme Edge

Longwei Huang, Chao Fang, Qiong Li,

Jun Lin and Zhongfeng Wang

School of Electronic Science and Engineering, Nanjing University, China

lwhuang@smail.nju.edu.cn

Outline

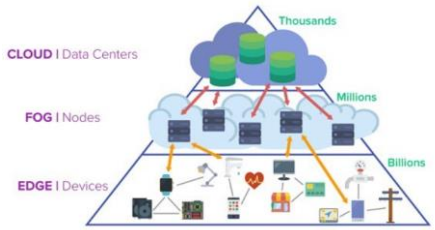
- 1. Motivation and Related Works
- 2. Our Proposed Processor
- 3. Experiments
- 4. Conclusion

Outline

- **1. Motivation and Related Works**
- 2. Our Proposed Processor
- 3. Experiments
- 4. Conclusion

1.1.1 Motivation (TinyML)

- **TinyML: Deploy DNNs at the Extreme-Edge**
  - AI-enhanced IoT applications
  - In-vehicle, wearable smart devices, etc.
- **Challenges:**
  - Constrained computing resources
  - Small memory space
  - Power-sensitive
- **Solutions:**
  - Deploy **quantized DNNs** on **DNN processor**



Courtesy of Rusci M. «Example on MobilenetV1_224_1.0.»C

Precision Level	Top1 Accuracy	Weight Memory Footprint / MB
Full	70.9%	16.3
INT8	70.1%	4.1
INT4	66.5%	2.4
Mixed	68.0%	2.1

-2.9%

7.8x

1.1.2 Motivation (DNN Processor)

- **Challenges:**
  - 1. **Different precision levels** of quantized DNNs
  - 2. Need for **model accuracy** and **data privacy**
  - 3. Significant **variations in resource requirements** for extreme edge applications
- **Solutions:**
  - 1. **Precision-scalable** computation
  - 2. **Floating-point** computation for on-device learning
  - 3. **Configurable** hardware architectures

1.1.3 Motivation (FPGA Platform)

- **FPGA**
  - Energy-efficient
  - Cost-effectiveness
  - Decent programmability

Platforms	Energy Efficiency	Cost	Programmability	Quantized DNN Support Level
MCU	Low	Low	High	Low
GPU	Medium	Medium	High	Medium
ASIC	High	High	Low	High
FPGA	**Medium**	**Medium**	**Medium**	**High**

1.2 Related FPGA-based DNN Processors

Works	INT Precisions	ODL Perf.
Angel Eye	INT16	×
ThroughputOpt	INT8	×
Mix and Match	INT4	×
XpulpNN	INT16-2	Low
Ours	**INT16-2**	**High**

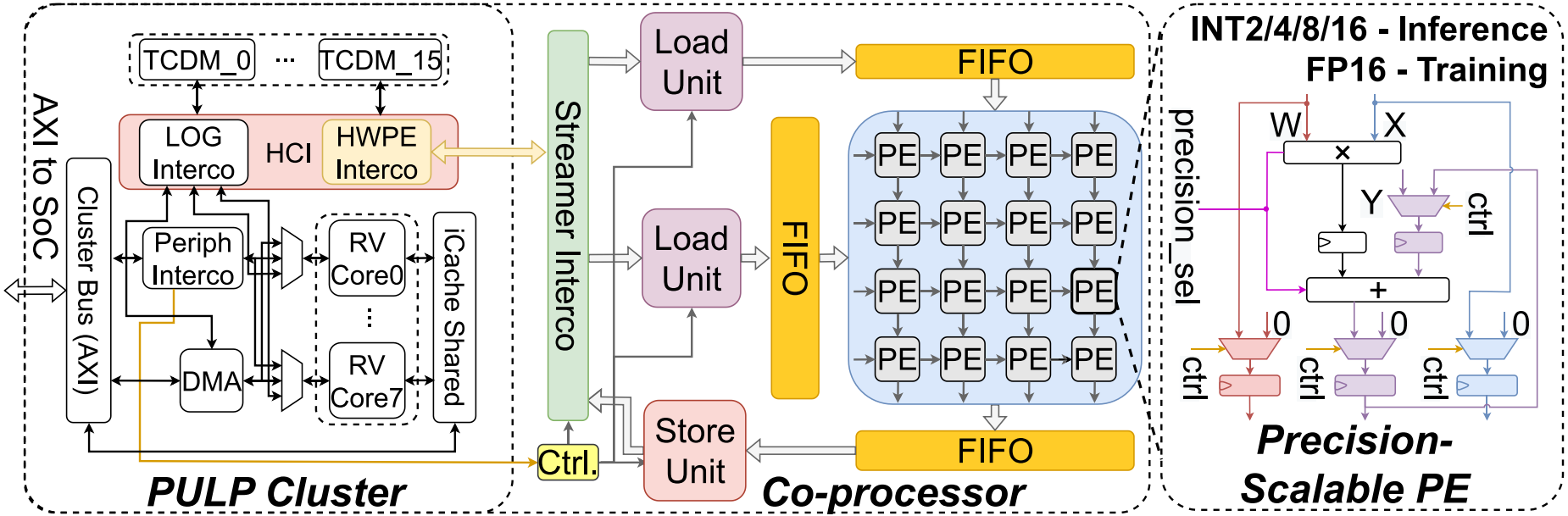
- **Our Processor**
  - Precision-scalable, ranging from INT16 to INT2
  - FP16 computation capability for on-device learning (ODL)

Outline

- 1. Motivation and Related Works
- **2. Our Proposed Processor**
  - 2.1 Features of Our DNN Processor
  - 2.2 Customized RISC-V Instruction-Driven Mapping
  - 2.3 Precision-Scalable Processing Element (PE)
  - 2.4 Balancing LUT and DSP Mapping
- 3. Experiments
- 4. Conclusion

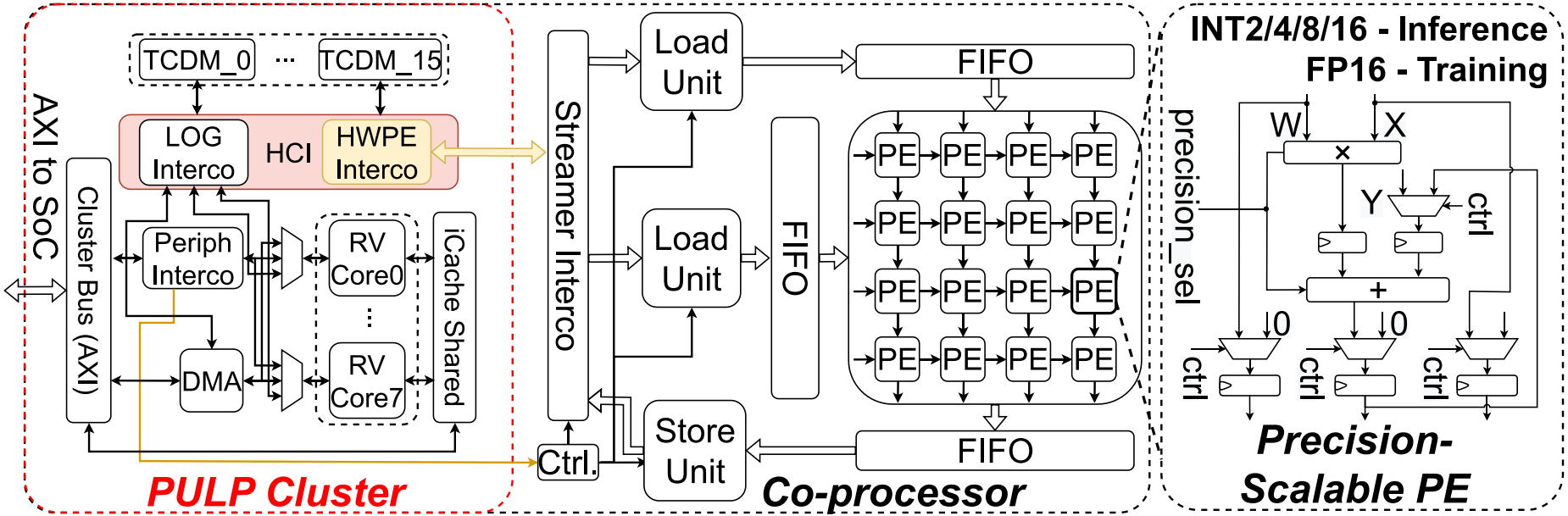
2. Our Proposed Processor

- Precision-scalable, high-throughput, energy-efficient and resource-efficient DNN processor



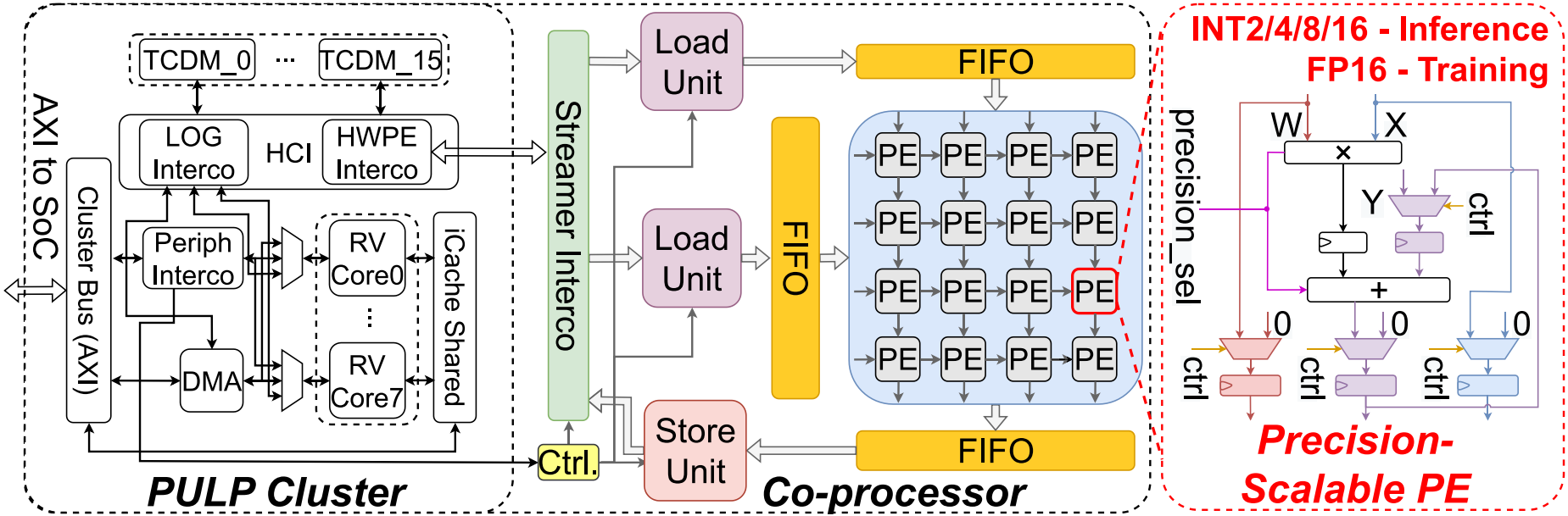
2. Our Proposed Processor

- Based on open-source PULP platform



2. Our Proposed Processor

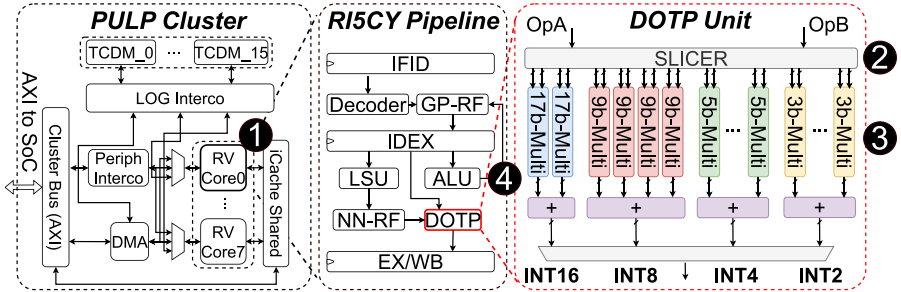
- Supports INT16, 8, 4, 2 and FP16 computation for on-device learning



2.1 Features of Our DNN Processor

XpulpNN

- 1. **Integrated** in RV pipeline
- 2. **SIMD** computing
- 3. Redundant precision-scalable DOTP unit **w/o hardware reuse**
- 4. Limited on-device learning capability **from ALU/FPU**



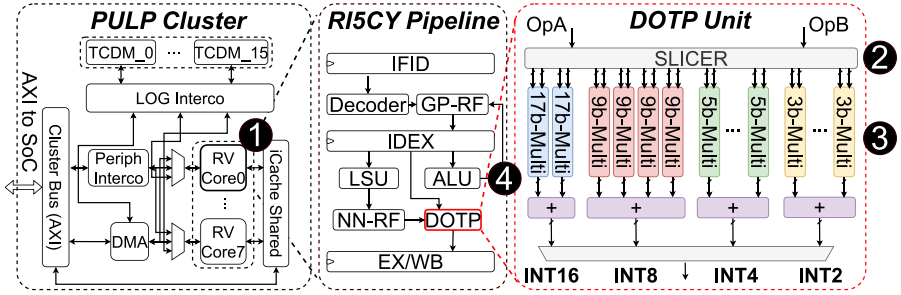
2.1 Features of Our DNN Processor

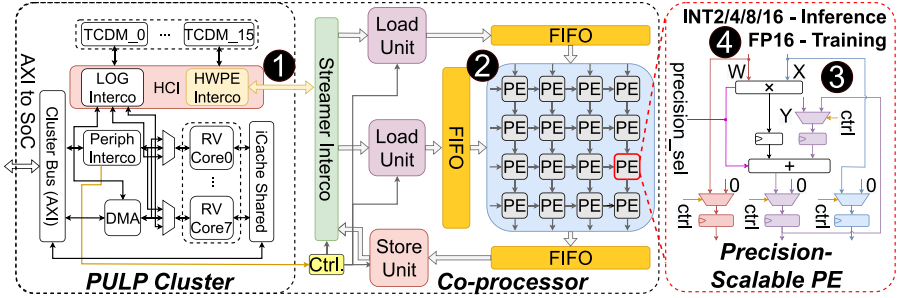
XpulpNN

- 1. **Integrated** in RV pipeline
- 2. **SIMD** computing
- 3. Redundant precision-scalable DOTP unit **w/o hardware reuse**
- 4. Limited on-device learning capability **from ALU/FPU**

Our Processor

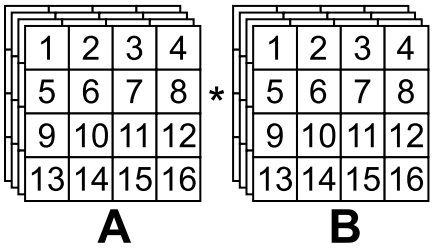
- 1. **Tightly-coupled** co-processor
- 2. High-throughput **systolic** computing
- 3. Efficient precision-scalable PE **w/ hardware reuse**
- 4. Boosted on-device learning capability **from co-processor**





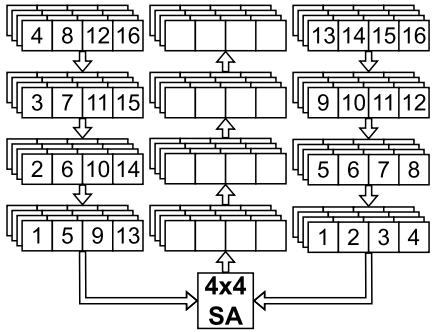
2.2 Customized RV Instr.-Driven Mapping

- **XpulpNN**
  - SIMD computing
  - **Redundant** instr. and cycles
  - **Low** theoretical throughput



four INT8 4x4 matrix multiplication

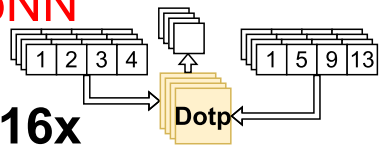
Ours



4 cycles	hwpe.setup	Setup
+1 cycle	hwpe.xaddr	
+1 cycle	hwpe.waddr	
+1 cycle	hwpe.len	
+19 cycles	hwpe.load	Compute
+7 cycles	hwpe.store	

6 instr. 33 cycles
15.5 OPs/cycle

XpulpNN

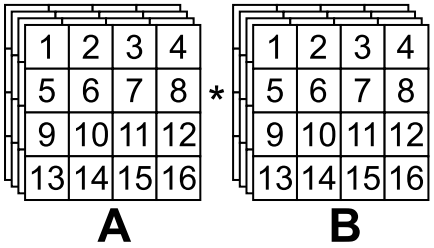


8 cycles	5x pv.nndotup.h	Setup
+1 cycle	lp.setup	
+1 cycle	pv.nndotup.h	Compute
+1 cycle	16x pv.nnsdotusp.b	
+16 cycles	16x p.sw	
+1 cycle	pv.nndotup.h	
+1 cycle	16x pv.nnsdotusp.b	
+16 cycles	16x p.sw	
+1 cycle	pv.nndotup.h	
+1 cycle	16x pv.nnsdotusp.b	
+16 cycles	16x p.sw	
+1 cycle	pv.nndotup.h	
+1 cycle	16x pv.nnsdotusp.b	
+16 cycles	16x p.sw	

138 instr. 81 cycles
6.3 OPs/cycle

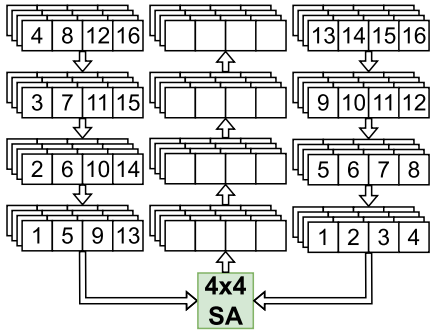
2.2 Customized RV Instr.-Driven Mapping

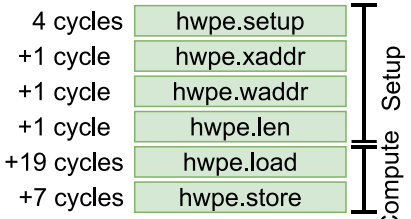
- Compared to XpulpNN, **our processor**
  - Realizes more **efficient** setup and compute process
  - Consumes only **4%** instr. and **41%** cycles for the same matmul
  - Achieves **2.5x** theoretical throughput improvement



four INT8 4x4 matrix multiplication

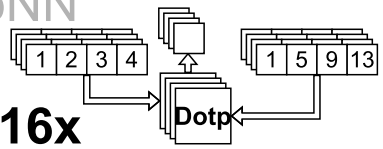
Ours

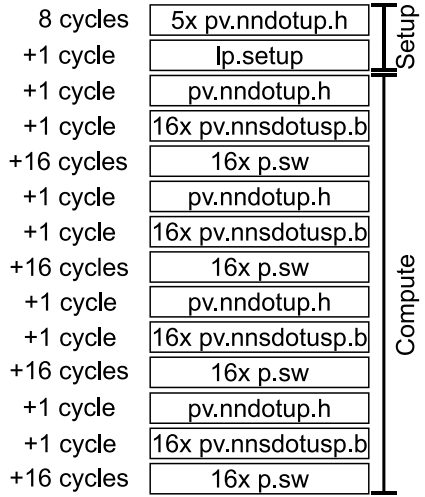




6 instr. 33 cycles
15.5 OPs/cycle

XpulpNN



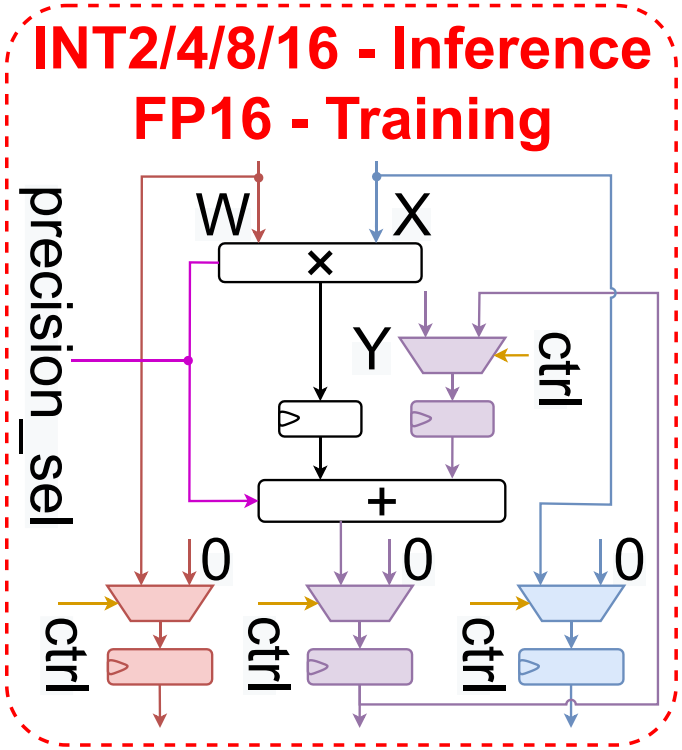


138 instr. 81 cycles
6.3 OPs/cycle

2.3 Precision-Scalable Processing Element

- Supporting **one** FP16 MAC, **one** INT16 MAC, **four** INT8 MACs, **eight** INT4 MACs and **sixteen** INT2 MACs

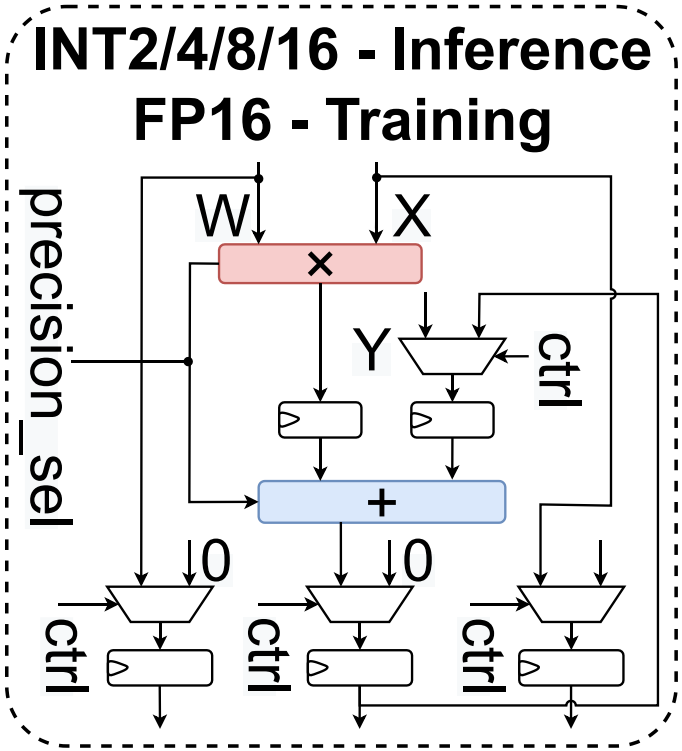
- One precision-scalable multiplier
- One precision-scalable adder
- Resource-efficient, achieved by reusing the resources of precision-scalable multiplier



2.3 Precision-Scalable Processing Element

- Supporting one FP16 MAC, one INT16 MAC, four INT8 MACs, eight INT4 MACs and sixteen INT2 MACs

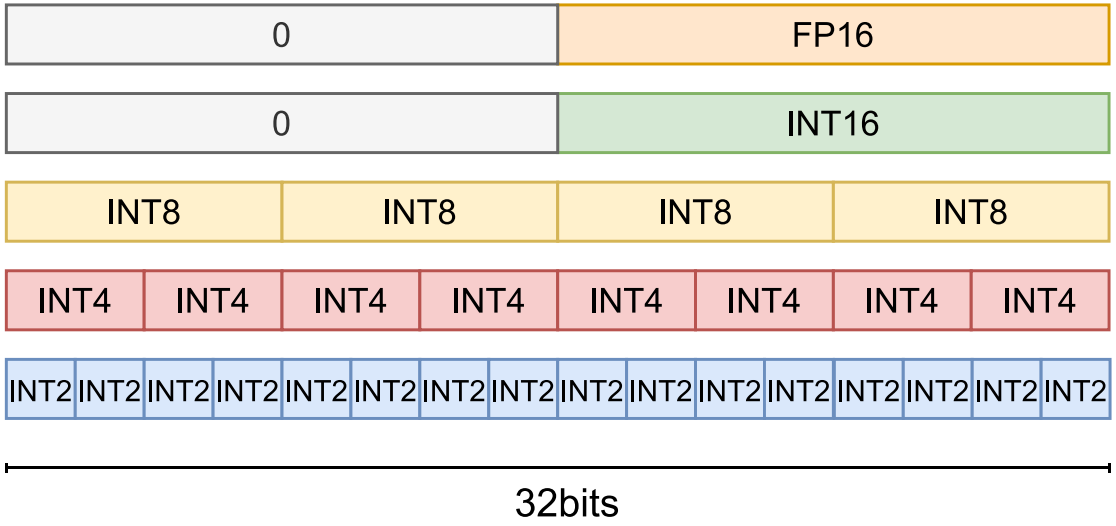
- One precision-scalable **multiplier**
- One precision-scalable **adder**
- **Resource-efficient**, achieved by reusing the resources of precision-scalable multiplier



2.3.1 Data Packing Method

To **accommodate** PEs

- 4 INT8 ➔ 32bits
- 8 INT4 ➔ 32bits
- 16 INT2 ➔ 32bits
- 1 INT16 + 16bits 0 ➔ 32bits
- 1 FP16 + 16bits 0 ➔ 32bits

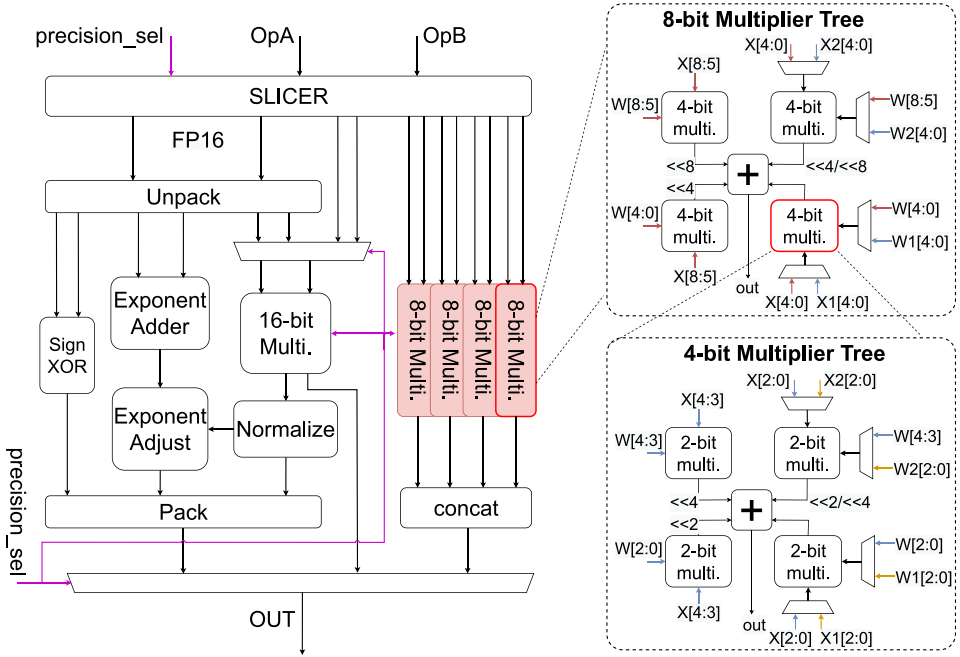


2.3.2 Precision-Scalable Multiplier

- One FP16/INT16 multiplier
- Four 8-bit multiplier tree

- Highly reused, hardware resources of eight 4-bit multipliers, sixteen 2-bit multipliers and one 16-bit multiplier are saved

- LUT and DSP resource overhead are reduced by 25.8% and 7.9%, respectively

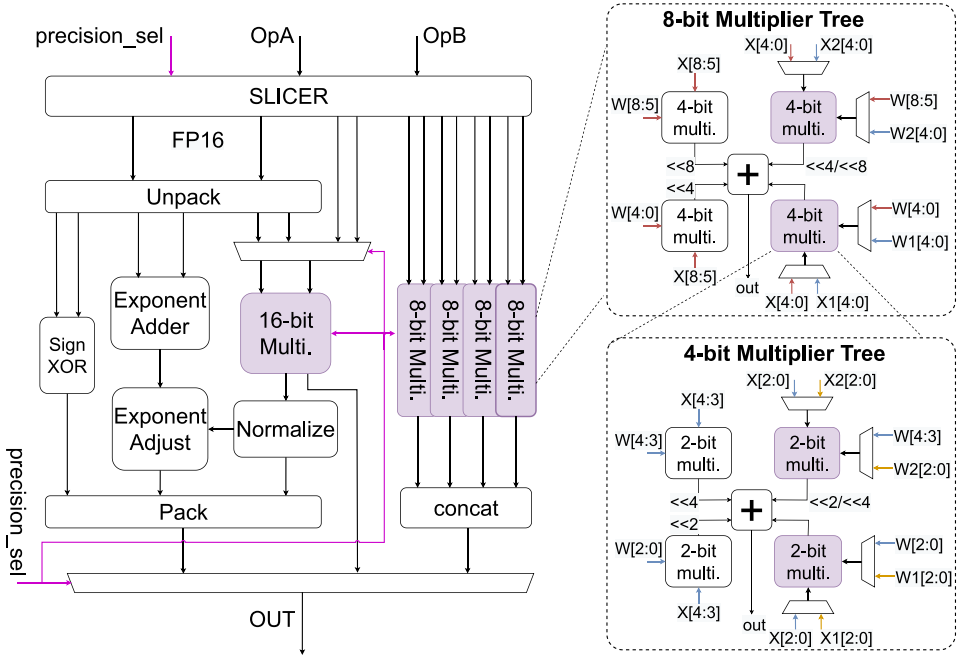


2.3.2 Precision-Scalable Multiplier

- One FP16/INT16 multiplier
- Four 8-bit multiplier tree

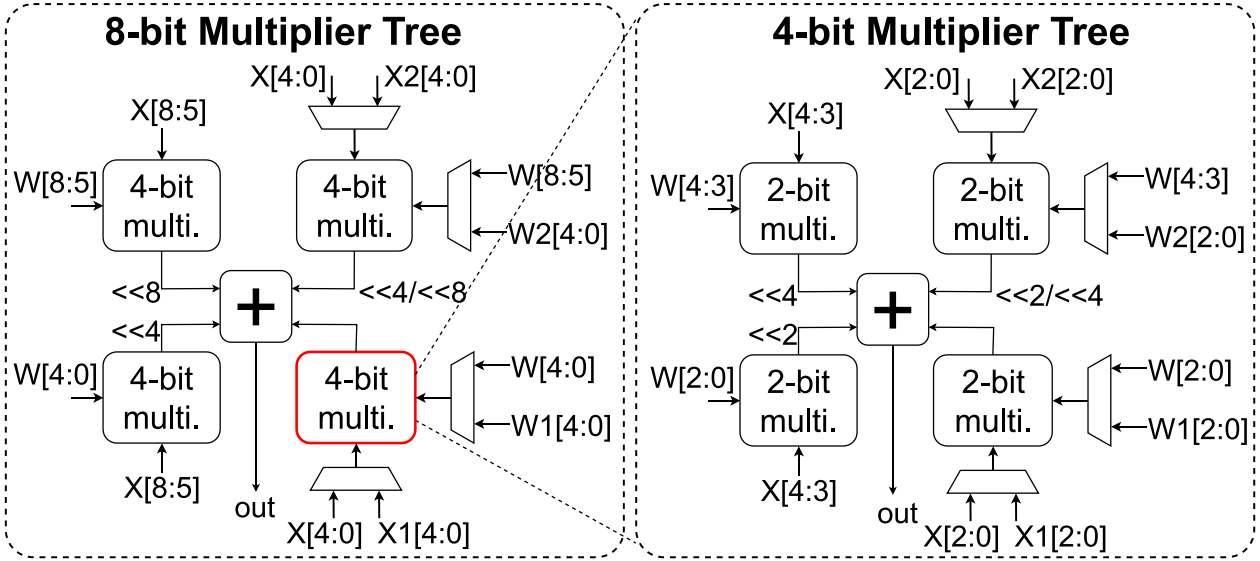
- **Highly reused**, hardware resources of eight 4-bit multipliers, sixteen 2-bit multipliers and one 16-bit multiplier are saved

- LUT and DSP resource overhead are reduced by **25.8%** and **7.9%**, respectively



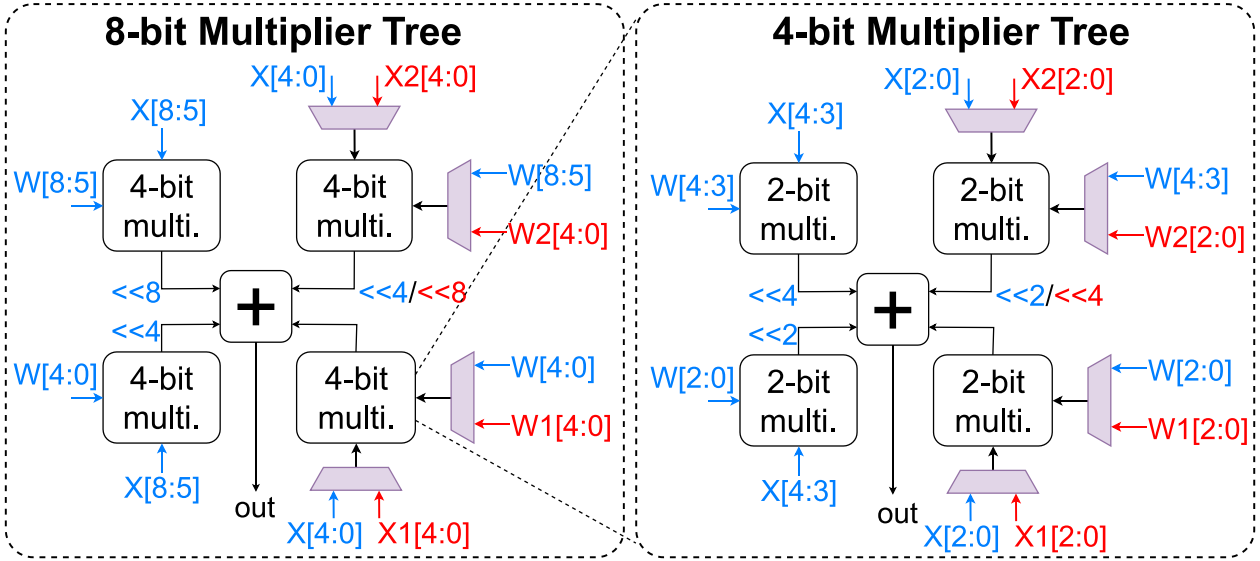
2.3.3 8-bit Multiplier Tree

- Efficient computation of **one INT8, two INT4 or four INT2** multiplication
- Each 8-bit tree consists of **four** 4-bit trees
- Each 4-bit tree consists of **four** 2-bit multipliers



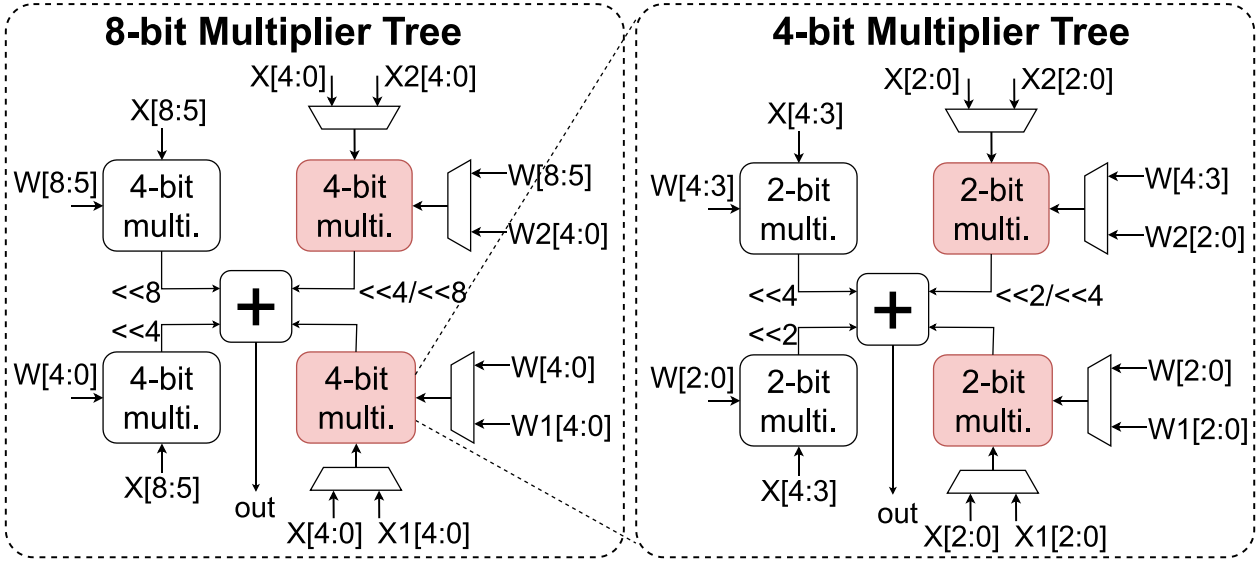
2.3.3 8-bit Multiplier Tree

- The result is obtained by **accumulating the partial products** of the high-bit data or **splicing** the low-bit data



2.3.3 8-bit Multiplier Tree

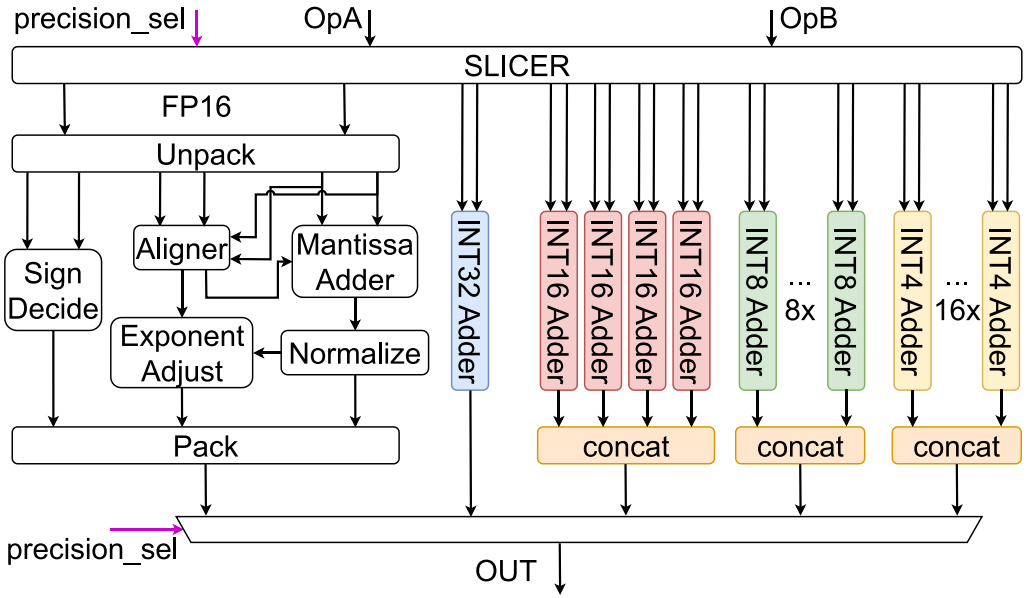
- Only **half** of the 4-bit and 2-bit multipliers are reused to ensure the **output bit-width remains the same** at different precisions



2.3.4 Precision-Scalable Adder

- One FP16 floating-point adder
- Four 16-bit adders
- Eight 8-bit adders
- Sixteen 4-bit adders

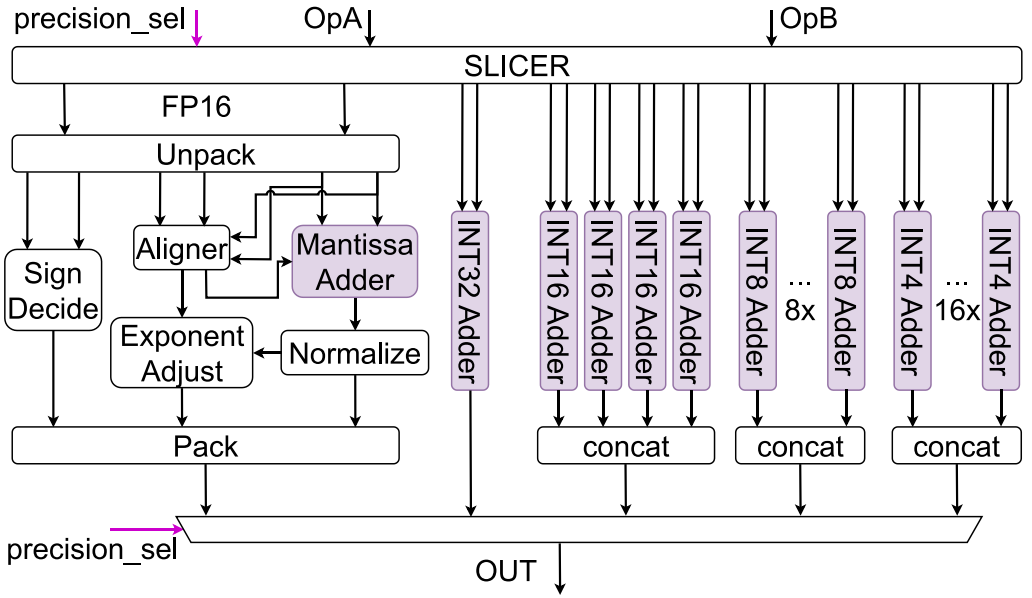
- Not reused due to the additional overhead caused by the increased MUXs is close to the resources it aims to reduce



2.3.4 Precision-Scalable Adder

- One FP16 floating-point adder
- Four 16-bit adders
- Eight 8-bit adders
- Sixteen 4-bit adders

- **Not reused** due to the additional overhead caused by the increased MUXs **is close to** the resources it aims to reduce



2.4 Balancing LUT and DSP Mapping

- Automatic resource mapping often leads to **inefficiencies**

- 16-bit mantissa multiplier, INT adders ➔ **DSPs**
- 8-bit multiplier trees, FP16 adder ➔ **LUTs**

- LUT and DSP resource overheads are reduced by up to **25.1%** and **63.5%**, respectively

Outline

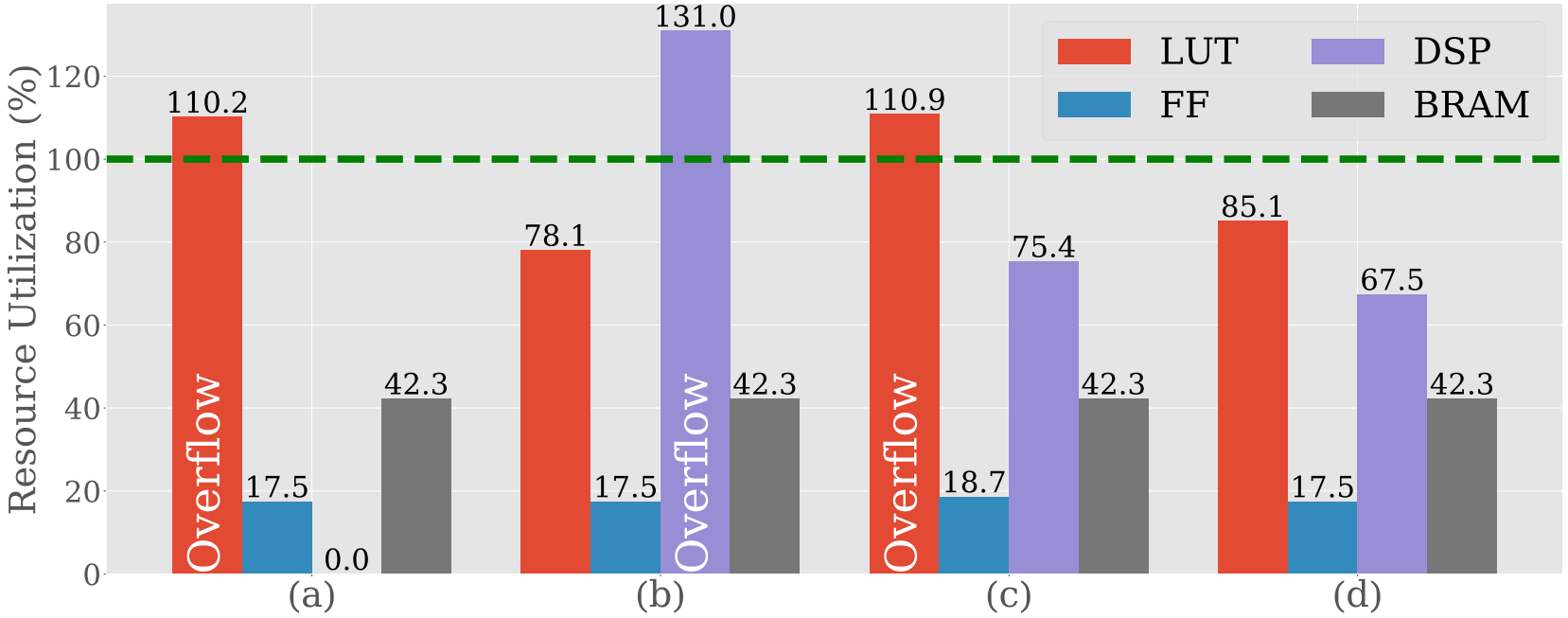
- 1. Motivation and Related Works
- 2. Our Proposed Processor
- **3. Experiments**
  - 3.1 Experimental Setup
  - 3.2 FPGA Resource Utilization Analysis
  - 3.3 Throughput and Energy Efficiency Comparison
  - 3.4 Comparison to CPU, GPU, and FPGA-based Prior Arts
- 4. Conclusion

3.1 Experimental Setup

- **Implementation 1:** ZCU102, 200MHz, 12x12 PEs
- **Implementation 2:** PYNQ-Z2, 100MHz, 4x4 PEs
- **DNN models:** MobileNetv2, VGG-16, ResNet-18, ResNet-50, ViT/B-16
- **Baselines:** ARM-Cortex A7, i5-10505, Jetson Nano, XpulpNN, etc.
- **Power consumption:** Vivado Power Analysis tool

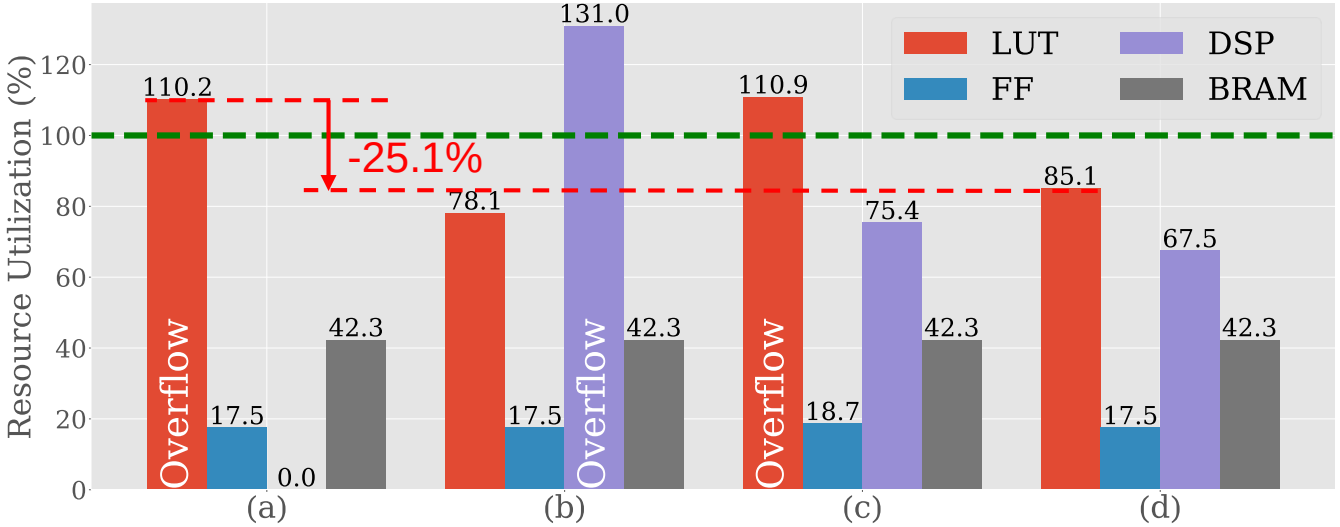
3.2.1 Utilization of Different Mapping Methods

- **Balanced** mapping to LUTs and DSPs + **reused** multipliers in (d) ➔ Significantly **reduce** of resource overheads



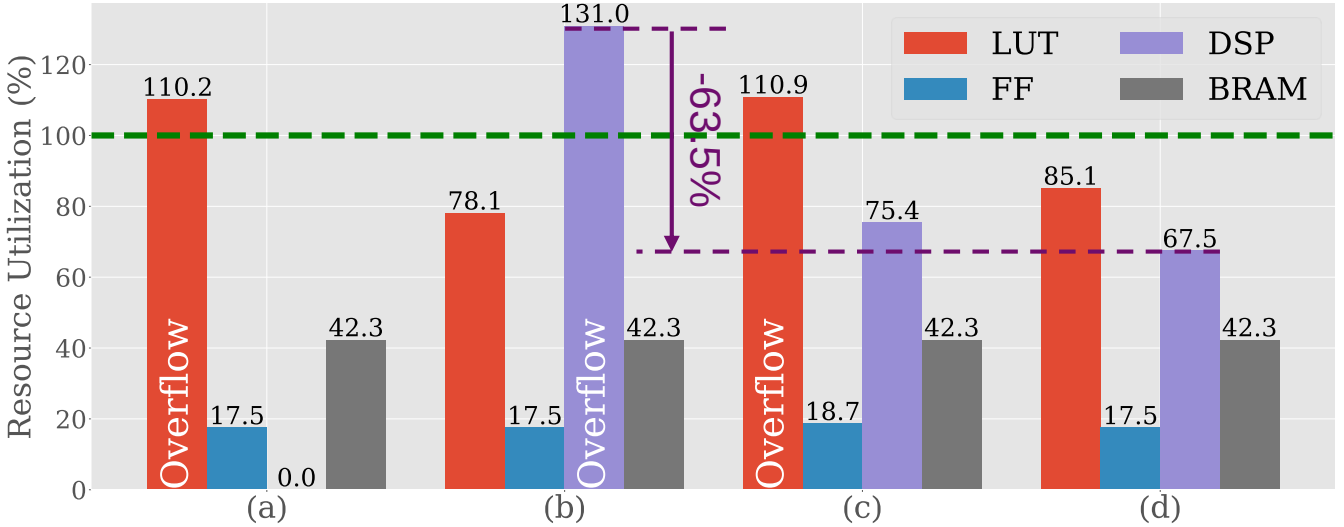
3.2.1 Utilization of Different Mapping Methods

- (d): Balanced mapping to LUTs and DSPs + reused multipliers
- **-25.1% LUTs compared to mapping to LUTs only in (a)**
- -63.5% DSPs compared to mapping to DSPs whenever possible in (b)
- -25.8% LUTs and -7.9% DSPs compared to w/o reused multipliers in (c)



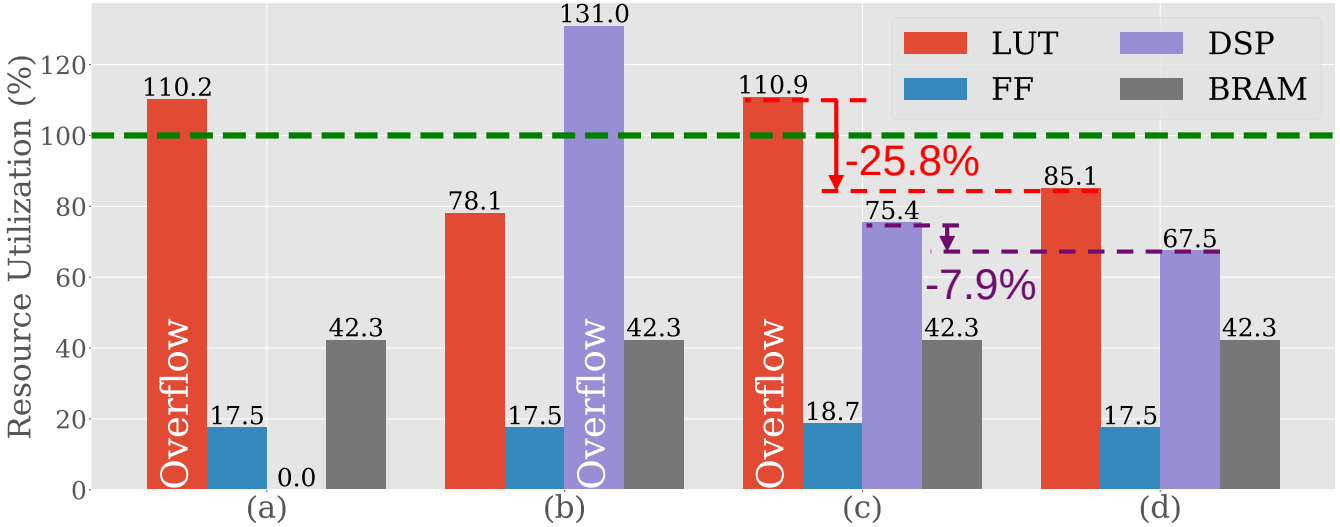
3.2.1 Utilization of Different Mapping Methods

- (d): Balanced mapping to LUTs and DSPs + reused multipliers
- -25.1% LUTs compared to mapping to LUTs only in (a)
- **-63.5% DSPs compared to mapping to DSPs whenever possible in (b)**
- -25.8% LUTs and -7.9% DSPs compared to w/o reused multipliers in (c)



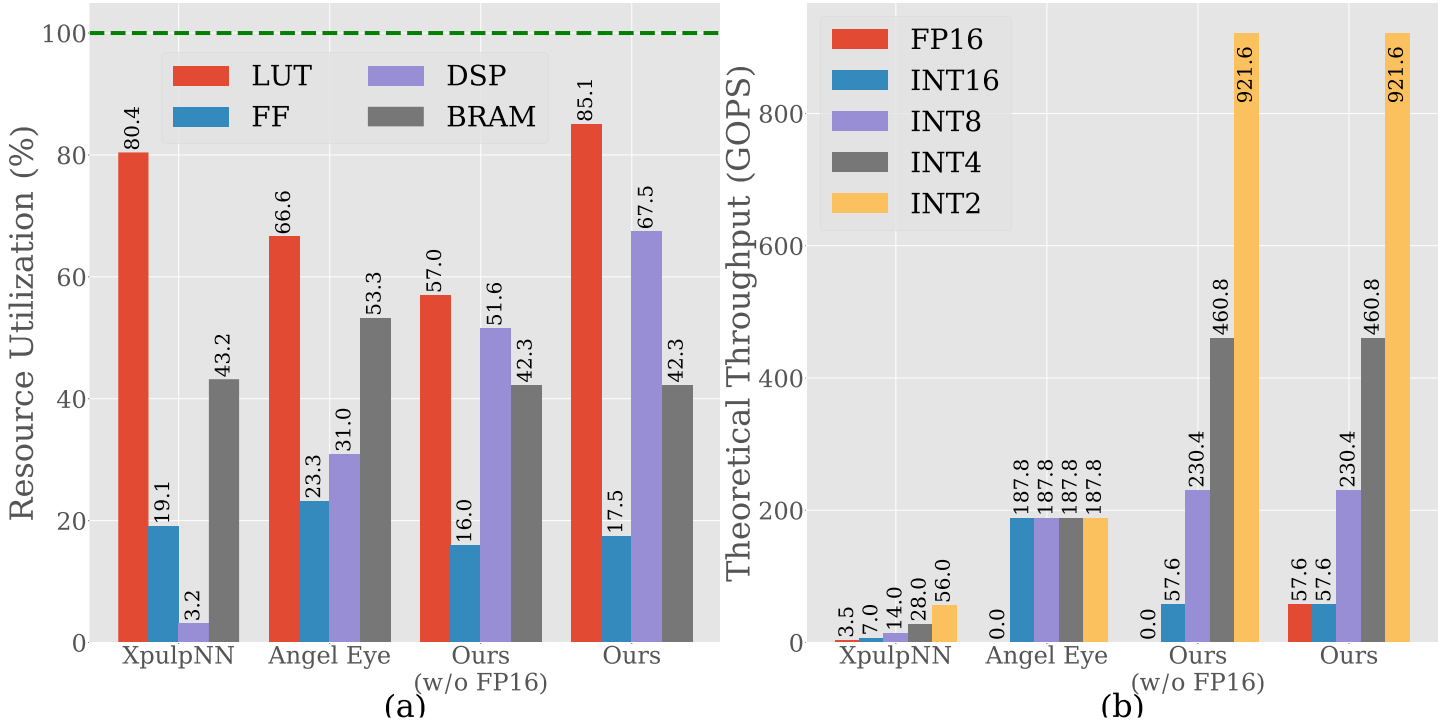
3.2.1 Utilization of Different Mapping Methods

- (d): Balanced mapping to LUTs and DSPs + reused multipliers
- -25.1% LUTs compared to mapping to LUTs only in (a)
- -63.5% DSPs compared to mapping to DSPs whenever possible in (b)
- **-25.8% LUTs and -7.9% DSPs compared to w/o reused multipliers in (c)**



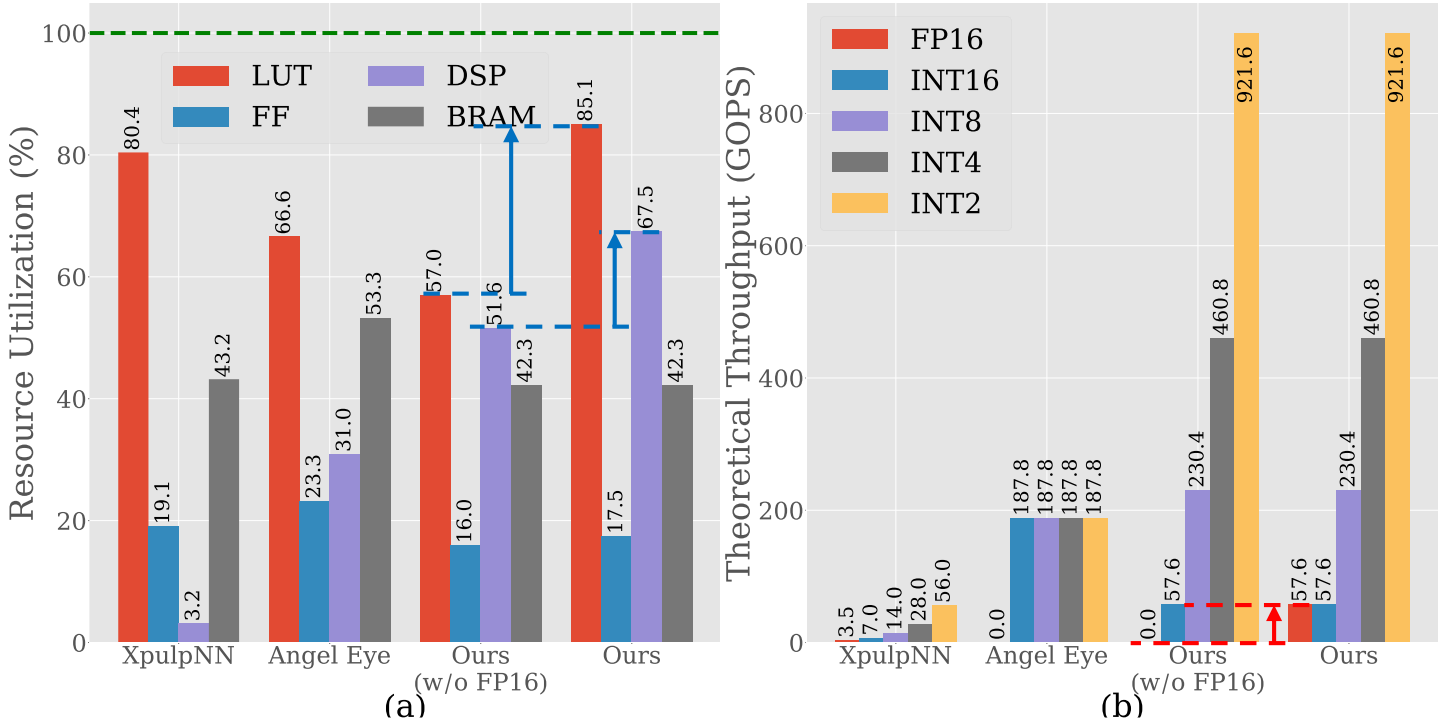
3.2.2 Overhead for ODL Support

- **Increase** of LUTs and DSPs ➔ considerable INT throughput **improvement** and FP16 **capability**



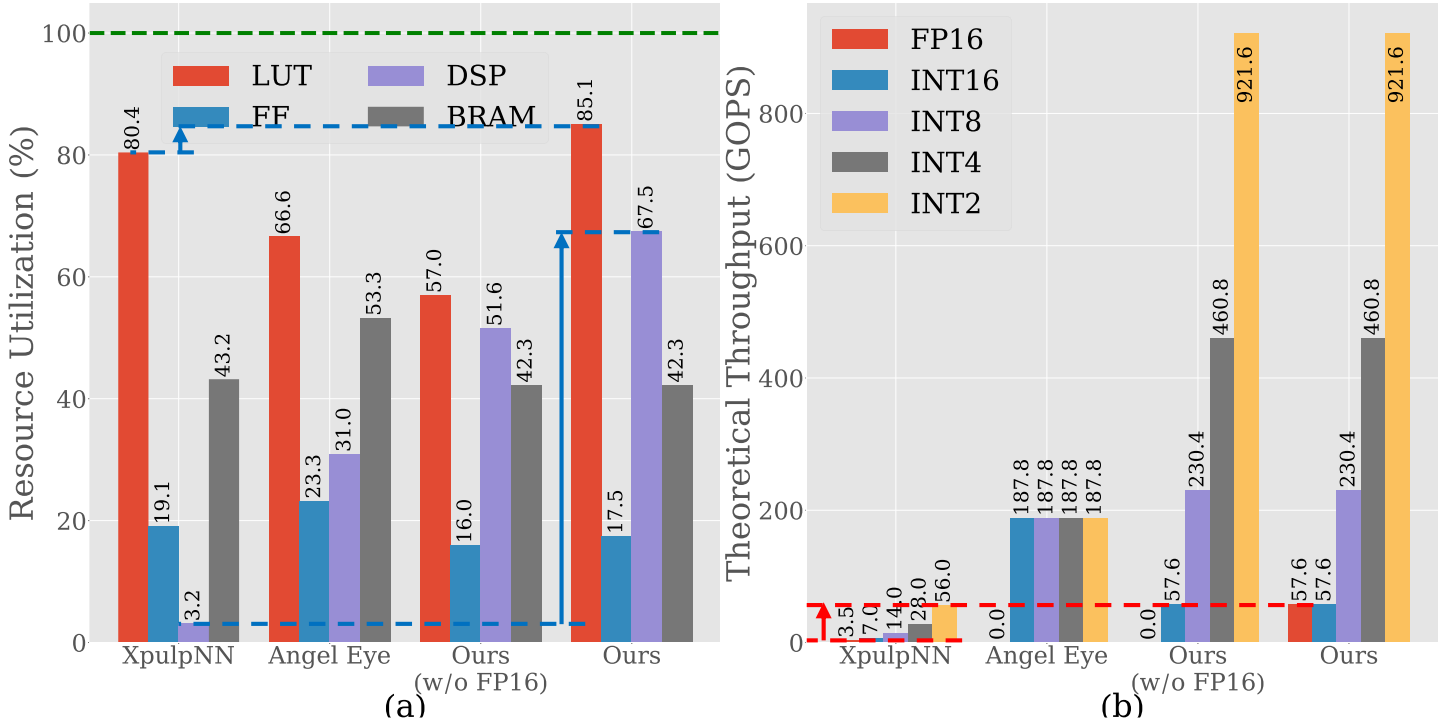
3.2.2 Overhead for ODL Support

- **+28.1%** LUTs and **+15.9%** DSPs compared to **w/o FP16** ➔ **+57.6** FP16 GOPs



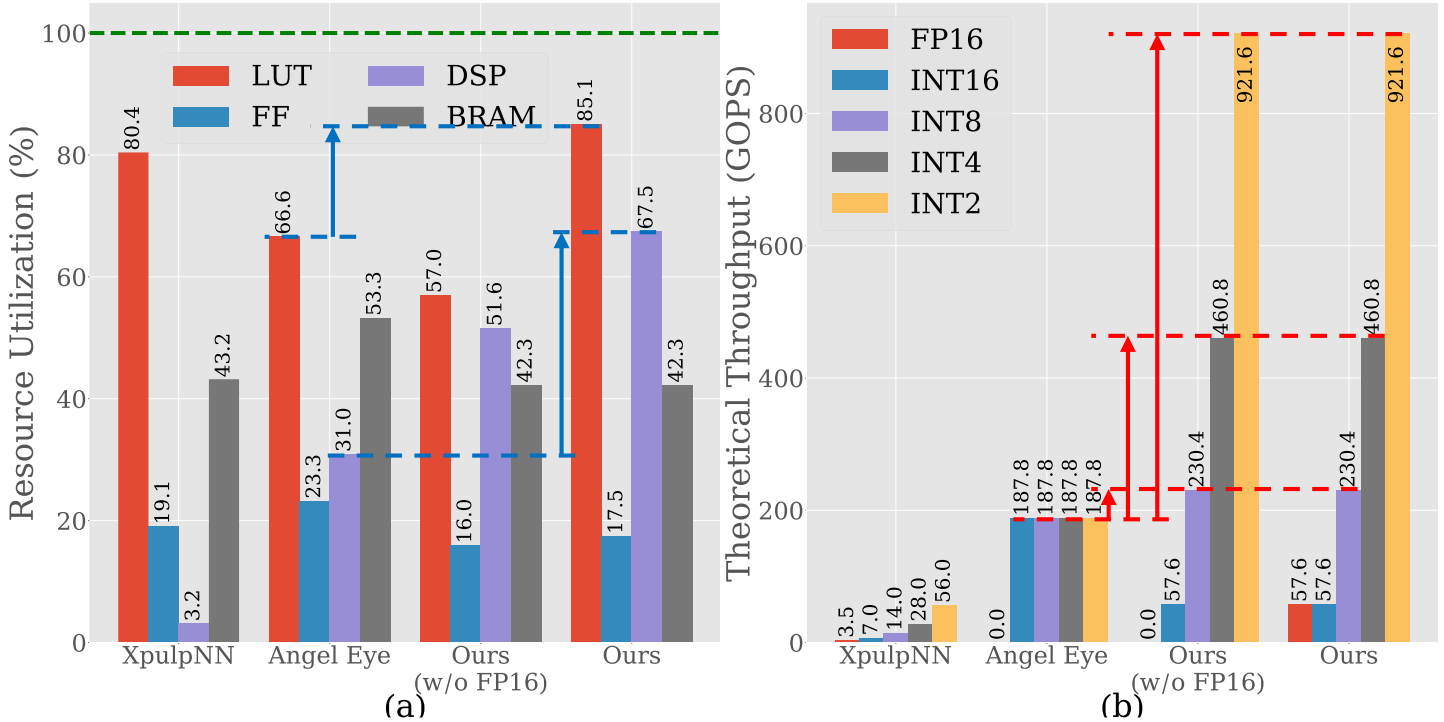
3.2.2 Overhead for ODL Support

- **+4.7%** LUTs and **+64.3%** DSPs compared to **XpulpNN ➔ 16.5x** FP16 and **8.2x** INT16 theoretical throughput improvement



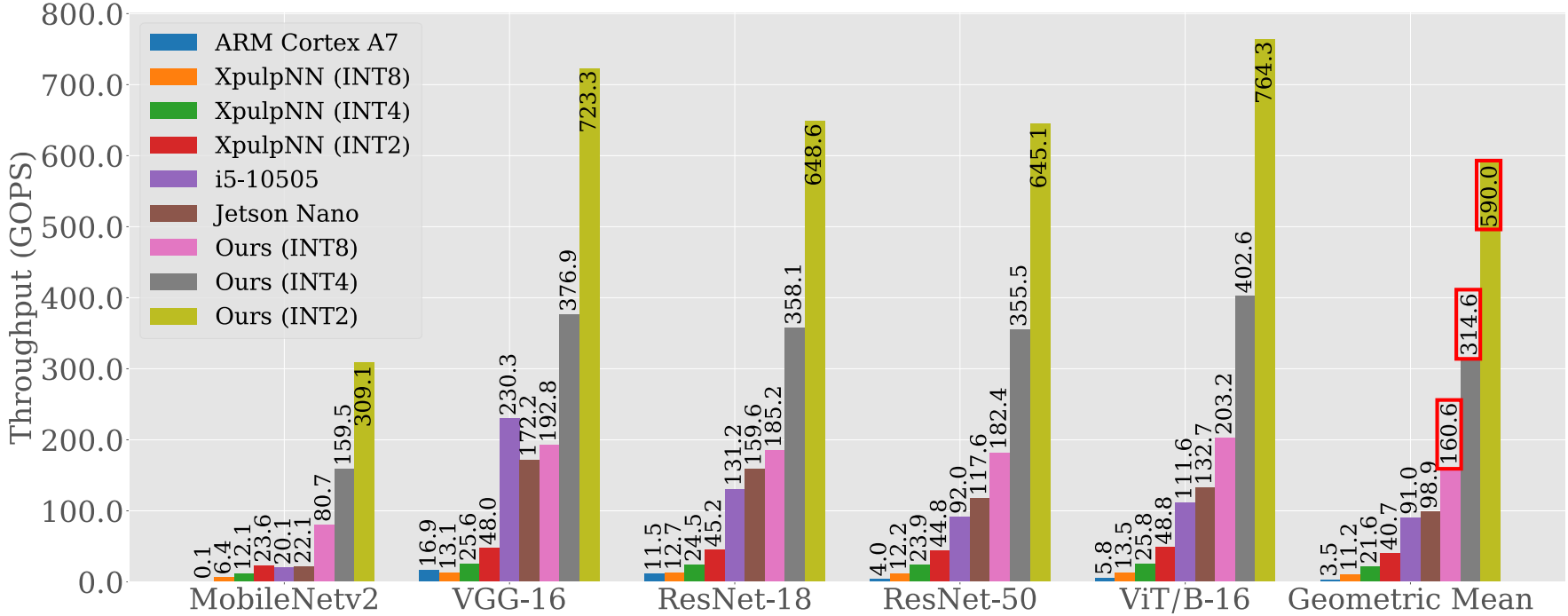
3.2.2 Overhead for ODL Support

- **+18.5%** LUTs and **+36.5%** DSPs compared to **Angel Eye** ➔ **+57.6** FP16 GOPs, **1.2x** INT8, **2.5x** INT4 and **4.9x** INT2 theoretical throughput improvement



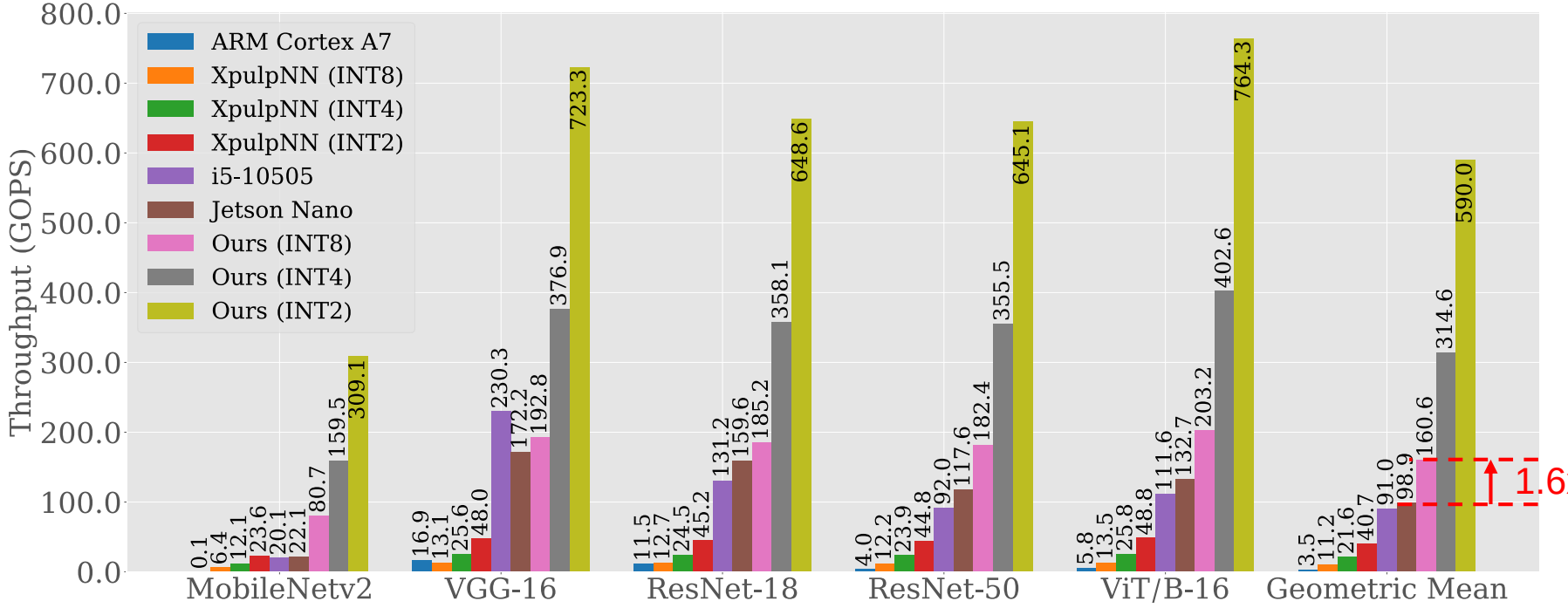
3.3.1 Throughput Comparison

- Considerable **throughput** on average



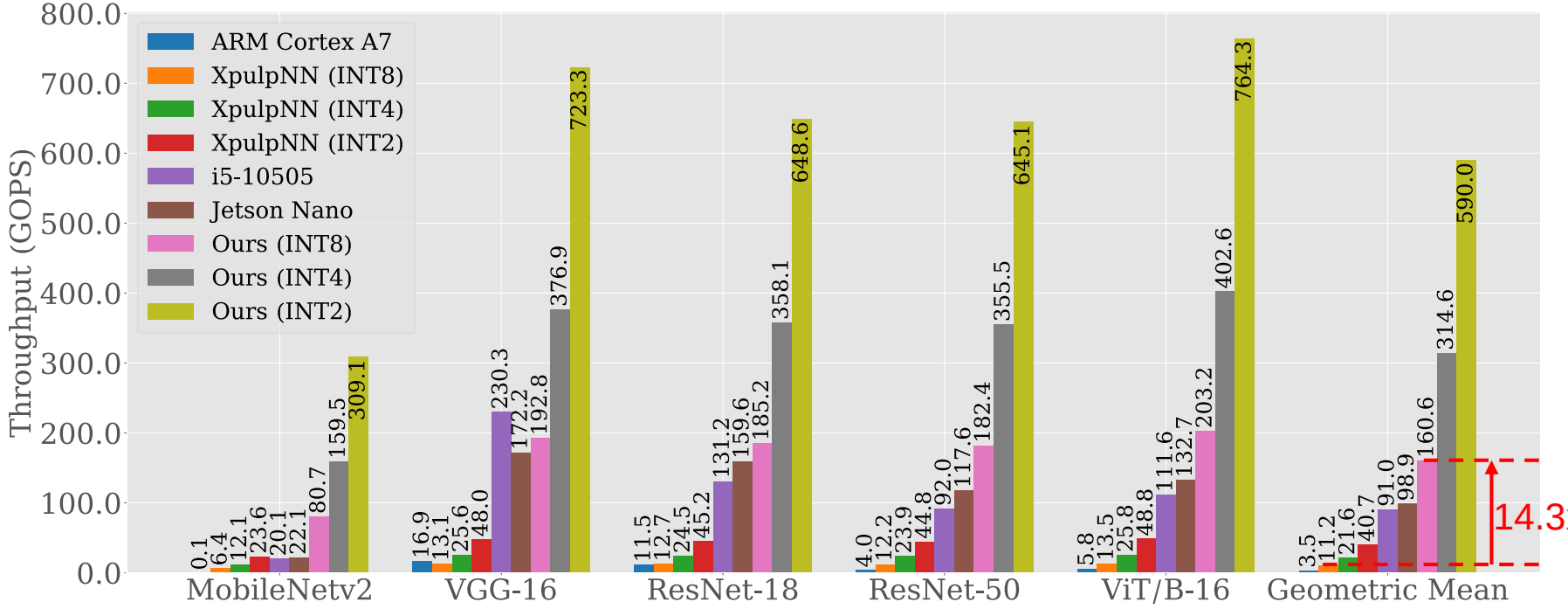
3.3.1 Throughput Comparison

- **1.6x** average INT8 throughput improvement over **Jetson Nano**



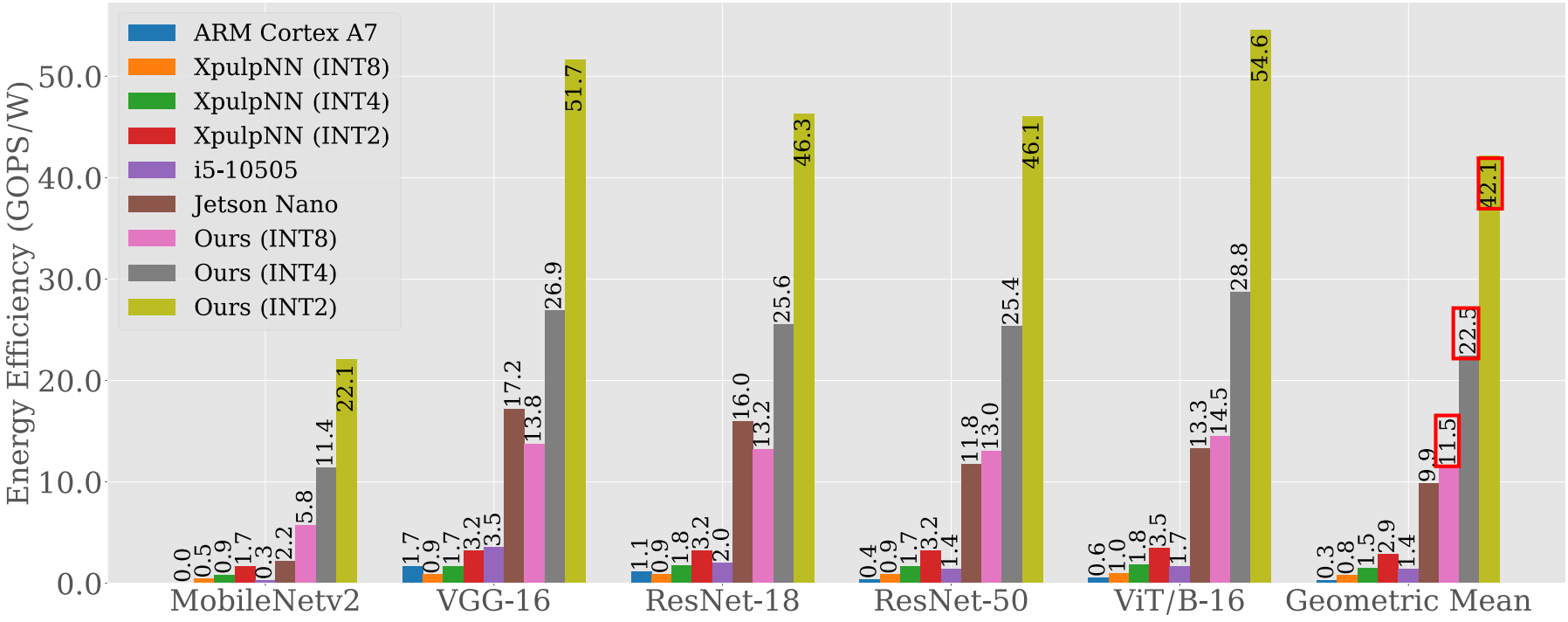
3.3.1 Throughput Comparison

- **14.3x** average INT8 throughput improvement over **XpulpNN**



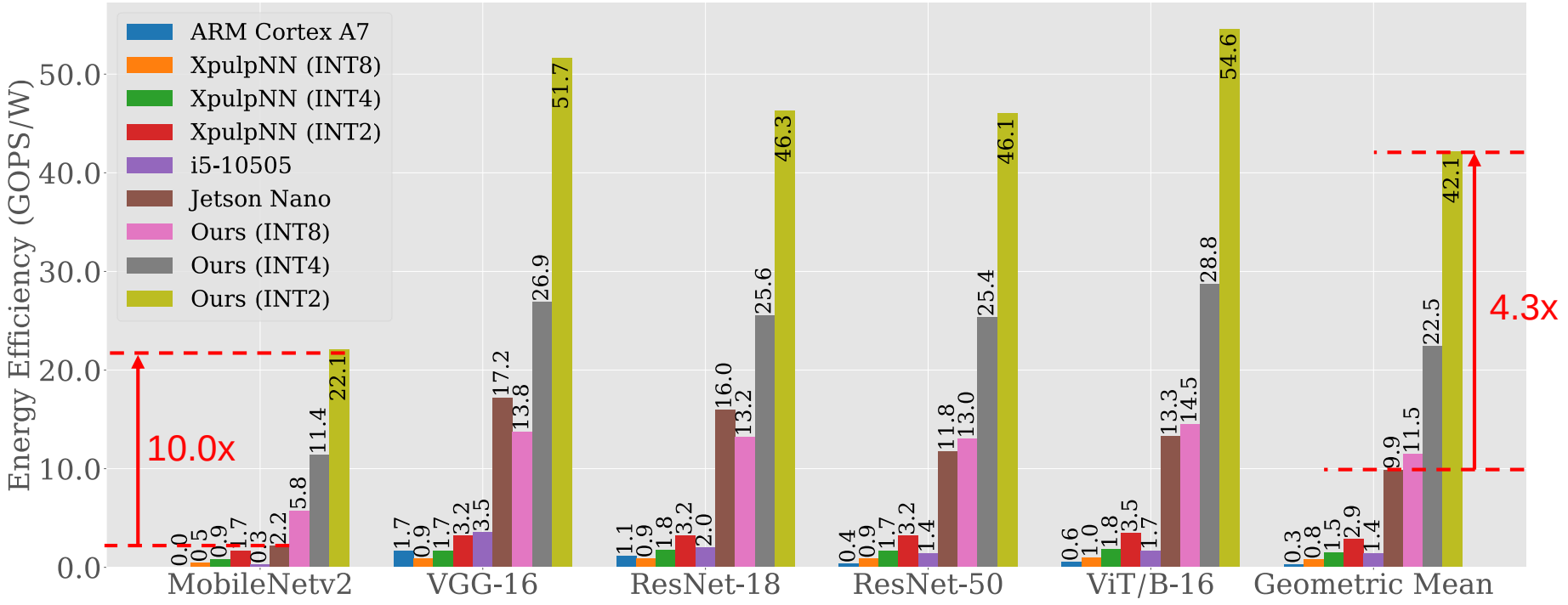
3.3.2 Energy Efficiency Comparison

- Considerable **energy efficiency** on average



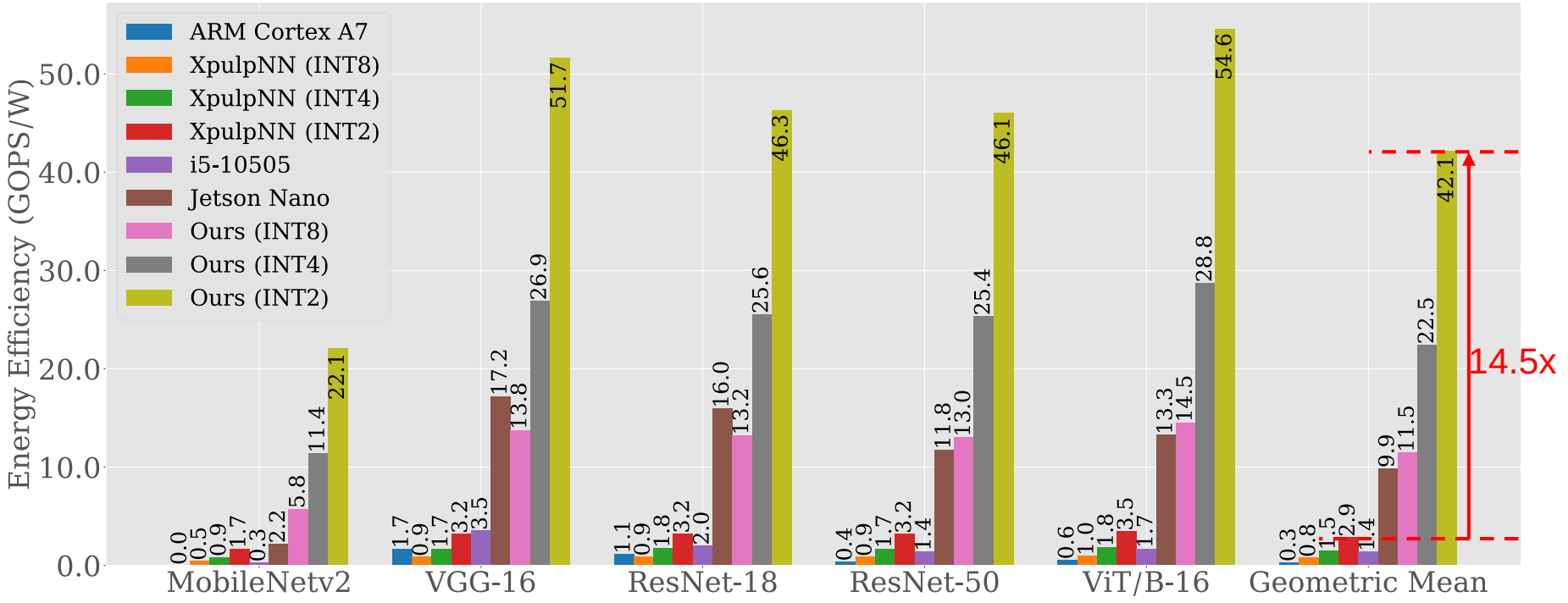
3.3.2 Energy Efficiency Comparison

- **1.1~4.3x** average improvements over **Jetson Nano** at various precisions (**2.6~10.0x** when running a lower computational density DNN, i.e., MobileNetv2)



3.3.2 Energy Efficiency Comparison

- **14.6x** and **14.5x** average improvements over **XpulpNN** at INT4 and INT2, respectively



3.4.1 Comparison to CPUs and GPU

- Noteworthy energy efficiency enhancements of **2.4x** (**i5-10505**) and **8.5x** (**Arm-Cortex A7**) at INT16
- **1.6x** and **1.1x** throughput and energy efficiency improvements, respectively, over energy-efficient GPU for the extreme-edge (**Jetson Nano**)

Work	Freq. (MHz)	Model	Perf. (GOPS)		En. Eff. (GOPS/W)	
			INT16	INT8	INT16	INT8
Arm-Cortex A7	1430	ResNet-50	4.0	N/A	0.4	N/A
i5-10505	3200	ResNet-50	92.0	N/A	1.4	N/A
Jetson Nano	640	ResNet-50	N/A	117.6	N/A	11.8
Ours (ZCU102)	**200**	**ResNet-50**	**47.0**	**182.4**	**3.4**	**13.0**

1.6x

8.5x

3.4.2 Comparison to Prior Arts (ZCU102)

- **1.6~15.0x** (perf.) and **1.7~14.4x** (en. eff.) with precision-scalability and ODL capability

Work	Platform	kLUT	DSP	Freq. (MHz)	Model	Performance (Perf.) (GOPS)				Energy Efficiency (En. Eff.) (GOPS/W)				ODL FP support
						INT16	INT8	INT4	INT2	INT16	INT8	INT4	INT2	
Going Deeper	XC7Z045	218.6	N/A	150	VGG-16	137.0	×	×	×	N/A	×	×	×	×
Angel Eye	XC7Z045	182.6	780	150	VGG-16	187.8	×	×	×	14.2	×	×	×	×
ThroughputOpt	Stratix V	153	246	120	VGG-16	×	117.8	×	×	×	N/A	×	×	×
Mix and Match	XC7Z045	145.1	900	100	ResNet-18	×	×	359.2	×	×	×	N/A	×	×
FILM-QNN	ZCU102	174.5	2.1k	150	ResNet-50	×	N/A	387.8	×	×	N/A	28.9	×	×
BARVINN	Alveo U250	201.1	512	250	ResNet-50	N/A	N/A	N/A	380.4	N/A	N/A	N/A	17.7	×
XpulpNN	ZCU102	220.4	80	200	ResNet-50	6.0	12.2	23.9	44.8	0.4	0.9	1.7	3.2	√
Ours	**ZCU102**	**233.3**	**1.7k**	**200**	**ResNet-50**	**47.0**	**182.4**	**355.5**	**645.1**	**3.4**	**13.0**	**25.4**	**46.1**	√

15.0x

14.4x

3.4.2 Comparison to Prior Arts (PYNQ-Z2)

- Comparable throughput
- **1.8~3.6x** energy efficiency gains compared to **XpulpNN**

Work	Platform	kLUT	DSP	Freq. (MHz)	Model	Performance (Perf.) (GOPS)				Energy Efficiency (En. Eff.) (GOPS/W)				ODL FP support
						INT16	INT8	INT4	INT2	INT16	INT8	INT4	INT2	
XpulpNN	ZCU102	220.4	80	200	ResNet-50	6.0	12.2	23.9	44.8	0.4	0.9	1.7	3.2	√
Ours	**PYNQ-Z2**	**32.9**	**190**	**100**	**ResNet-50**	**2.8**	**11.8**	**24.3**	**46.5**	**0.7**	**3.0**	**6.1**	**11.6**	√

Outline

- 1. Motivation and Related Works
- 2. Our Proposed Processor
- 3. Experiments
- **4. Conclusion**

4. Conclusion

- A **FPGA-based** high-throughput, energy-efficient and precision-scalable RISC-V DNN processor with on-device learning capability at the extreme edge, supporting **INT16, 8, 4, 2 and FP16** precisions
- Up to **14.6x** inference throughput and energy efficiency improvement **both** over SoTA solutions **across various DNNs**
- Boosted **on-device learning** capability with up to **16.5x** ODL speedup over SoTA work, **XpulpNN**
- **Precision-scalable** PEs with **highly-reused** multiplier
- **Reuse methods** to fully leverage hardware resources, LUT and DSP resource overheads are reduced by up to **25.1%** and **63.5%**, respectively

Thanks for Listening!

If You Have any Question, Please Contact Us at

lwhuang@smail.nju.edu.cn
fantasysee@smail.nju.edu.cn