N³H-Core: Neuron-designed Neural Network Accelerator via FPGA-based Heterogeneous Computing Cores

Yu Gong*, Zhihan Xu*, Zhezhi He, Weifeng Zhang, Xiaobing Tu, Xiaoyao Liang, Li Jiang

Shanghai Qi Zhi Institute, Shanghai Jiao Tong University, Alibaba Group

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

* Means equal contribution

Background and Motivation

FPGA can adapt to various algorithms with outstanding performance.

Problem: Rich on-chip LUT resource are not well utilized.

The DNN quantization with varying bit-width is a compression scheme.

Problem: A high-performance architecture supporting mixed-precision operation is absent.

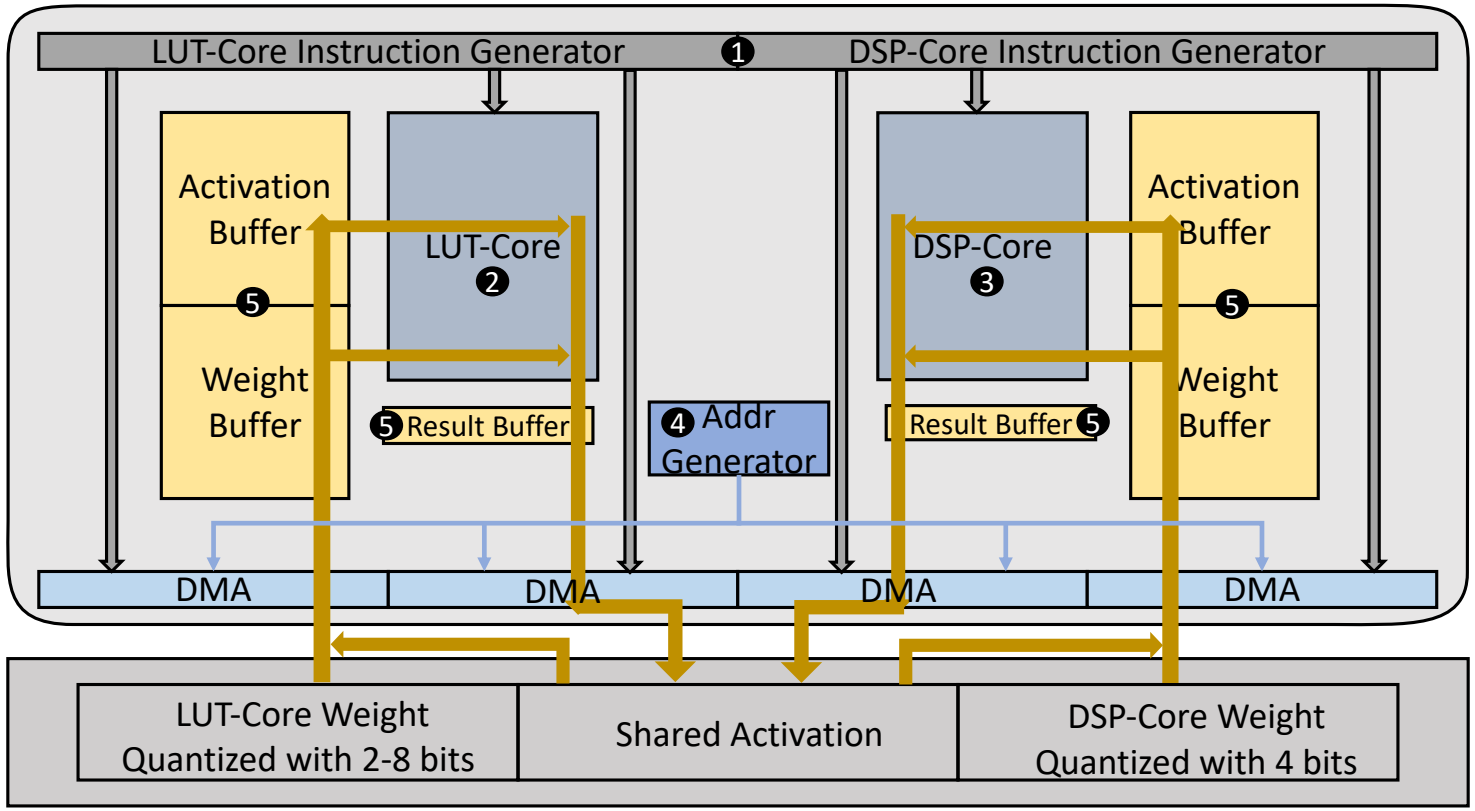
Design space exploration is conducted to limit the design space

Problem: Lacking the systematic design methodology for the single-chip heterogeneous system.

We develop a single-chip heterogeneous system for mixed-precision operation, and design the end-to-end optimization framework as a systematic design methodology for it.

Architecture of N^{3}H-Core

Overall architecture, including heterogenous computation cores: LUT-Core and DSP-Core, composed of LUT and DSP resource on FPGA respectively.



Heterogenous Computation Cores

LUT-Core, adopt BISMO (Yaman et al., 2018) as backbone

Based on bit-serial computation, supporting varying-bit operand

DSP-Core

Based on bit-parallel computation, supporting fixed-bit operand

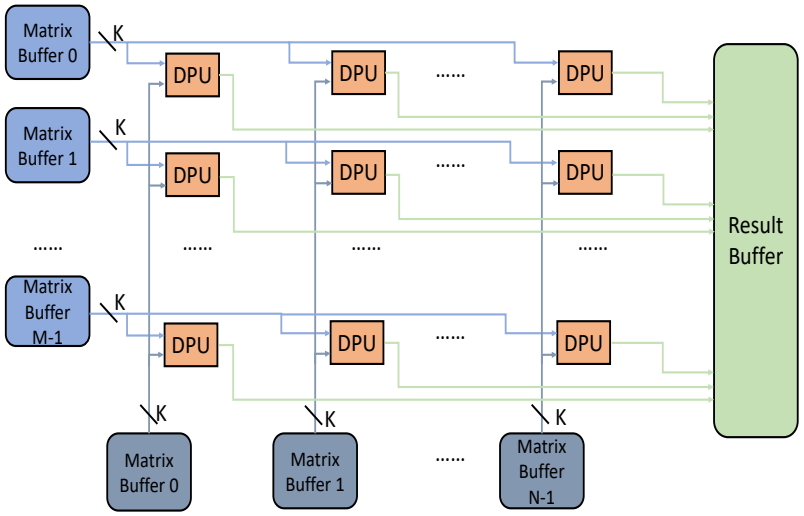
$$\boldsymbol{L} = \begin{bmatrix} 2 & 0 \\ 1 & 3 \end{bmatrix} = 2^1\boldsymbol{L}^{[1]} + 2^0\boldsymbol{L}^{[0]} = 2^1\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + 2^0\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$$

$$\boldsymbol{R} = \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix} = 2^1\boldsymbol{R}^{[1]} + 2^0\boldsymbol{R}^{[0]} = 2^1\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} + 2^0\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

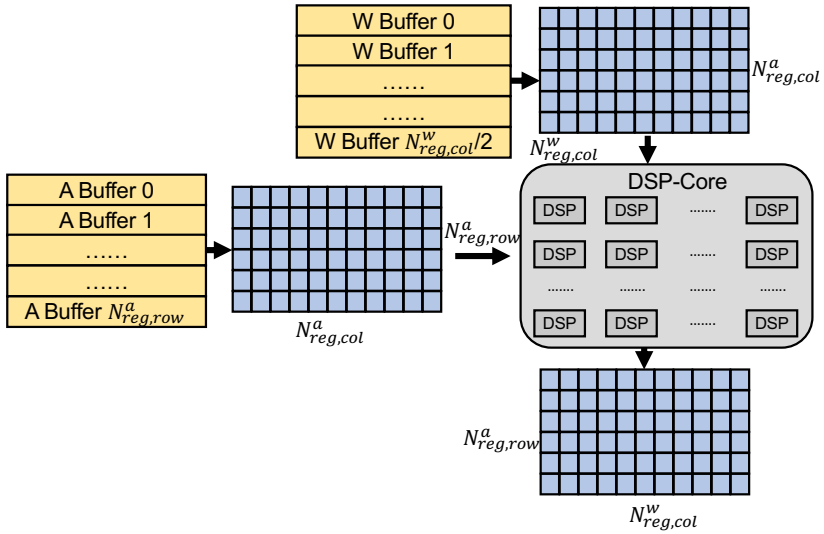
$$\boldsymbol{P} = \boldsymbol{L}\cdot\boldsymbol{R} = \left(2^1\boldsymbol{L}^{[1]} + 2^0\boldsymbol{L}^{[0]}\right)\cdot\left(2^1\boldsymbol{R}^{[1]} + 2^0\boldsymbol{R}^{[0]}\right)$$

$$= 2^2\boldsymbol{L}^{[1]}\cdot\boldsymbol{R}^{[1]} + 2^1\boldsymbol{L}^{[1]}\cdot\boldsymbol{R}^{[0]} + 2^1\boldsymbol{L}^{[0]}\cdot\boldsymbol{R}^{[1]} + 2^0\boldsymbol{L}^{[0]}\cdot\boldsymbol{R}^{[0]}$$

Bit-serial computation



LUT-Core



DSP-Core

Notation	*Description*
M	Numbers of rows in LUT-Core (DPU array)
N	Numbers of columns in LUT-Core (DPU array)
K	Input bit width of DPU
$D^{\mathrm{a}}_{\mathrm{L,buf}}$	Depth of **activation buffers** in **LUT-Core**
$D^{\mathrm{w}}_{\mathrm{L,buf}}$	Depth of **weight buffers** in **LUT-Core**
$N^{\mathrm{a}}_{\mathrm{reg,row}}$	Numbers of **rows** in **activation register** array
$N^{\mathrm{a}}_{\mathrm{reg,col}}$	Numbers of **columns** in **activation register** array
$N^{\mathrm{w}}_{\mathrm{reg,col}}$	Numbers of **columns** in **weight register** array
$D^{\mathrm{a}}_{\mathrm{D,buf}}$	Depth of **activation buffers** in **DSP-Core**
$D^{\mathrm{w}}_{\mathrm{D,buf}}$	Depth of **weight buffers** in **DSP-Core**
B^{a}	**Bit-widths** of quantized activation
$B^{\mathrm{w\text{-}L}}$	**Bit-widths** of quantized **weight** on **LUT-Core**
$B^{\mathrm{w\text{-}D}}$	**Bit-widths** of quantized **weight** on **DSP-Core**

Design knobs for N³H-Core

Cost and Latency Model

Build the cost and latency model to bridge the gap between hardware and software.

Cost model:

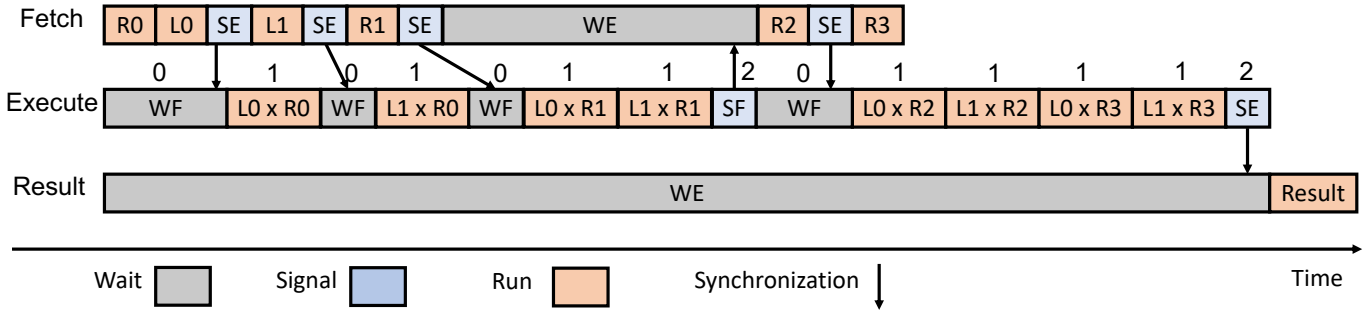
DSP: $DSP_{D\text{-}Core}=DSP_{available}$

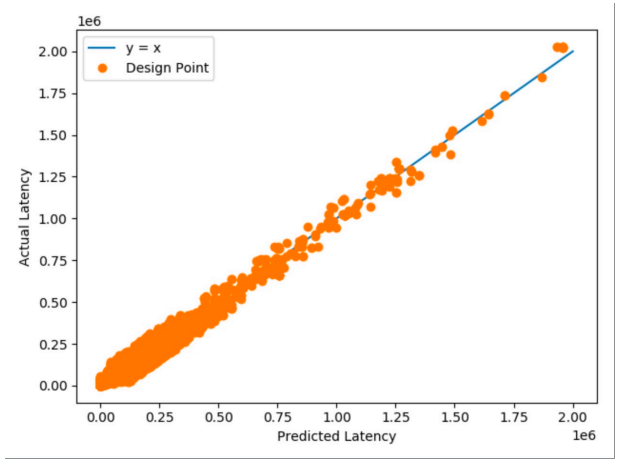
LUT: inherit from BISMO

BRAM: based on buffer size and physical structure of BRAM

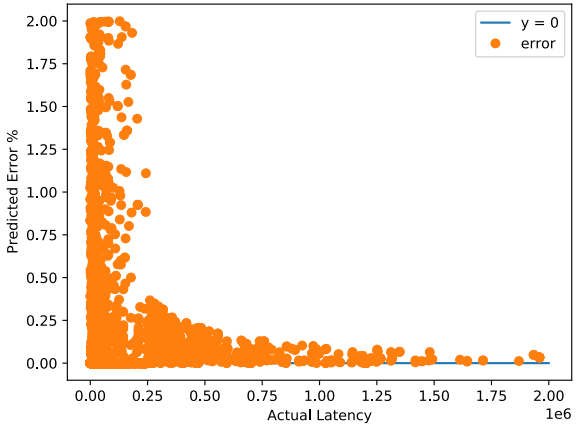
Latency model:

Build the model via instruction pipeline





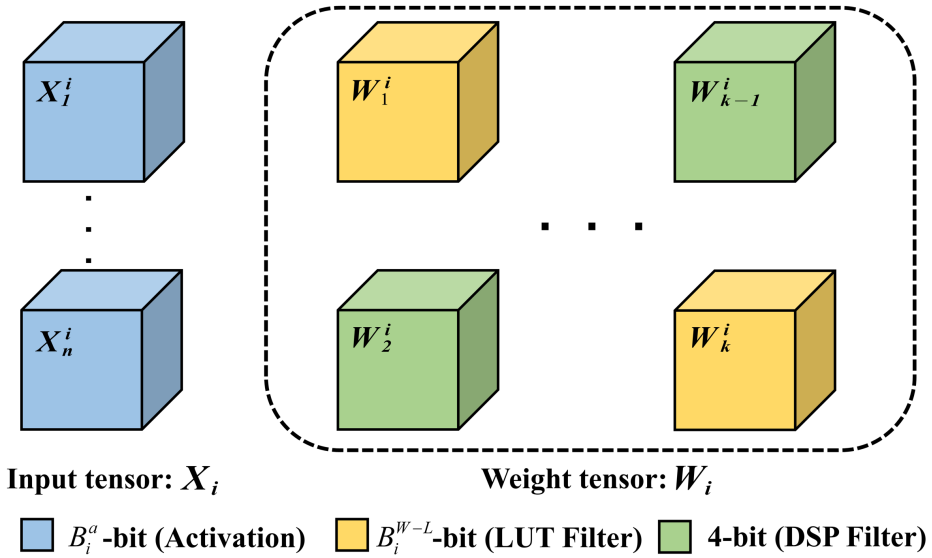
Predicted latency vs actual latency



Prediction error with design size

Hybrid Quantization

Uniform- and Mixed-precision: Filters in each layer computed by DSP-core and LUT-core are quantized with uniform- and mixed-precision respectively.



Neuron-based workload split ratio (*i-th* layer):

$$\text{ratio}_i = \frac{\text{Filter}_{\text{LUT}}^i}{\text{Filter}_{\text{all}}^i}$$

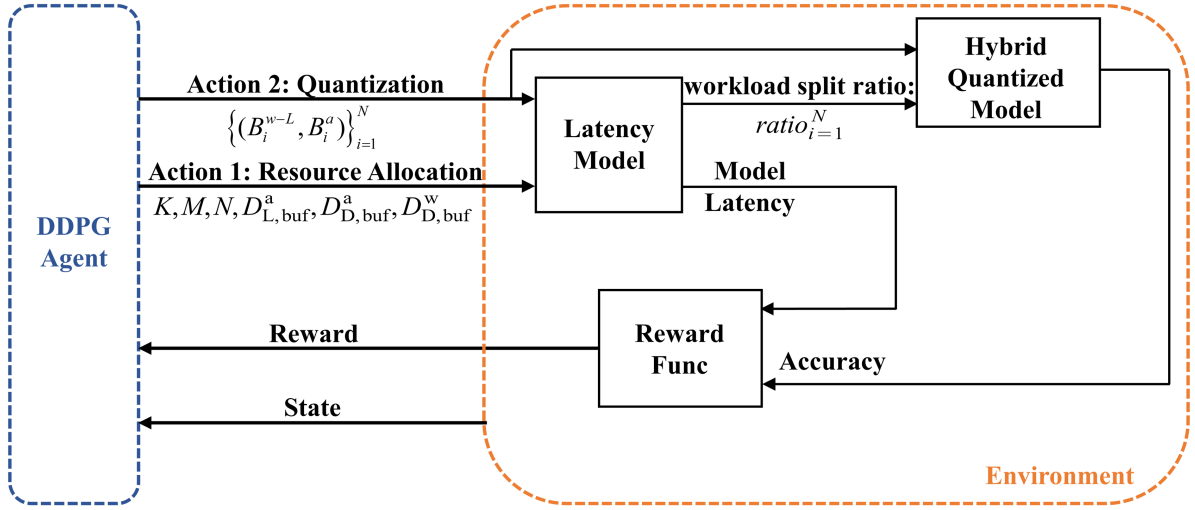
Design Space Exploration

Hardware- and Software-related factors

Design Factors	Range on D_A	Range on D_B
K	$64 \cdot v$ $(0 < v < 5, v \in \mathbb{N})$	$64 \cdot v$ $(0 < v < 5, v \in \mathbb{N})$
M	1-50	1-252
N	1-50	1-252
$D^{a}_{L,buf}$	$1024 \cdot v$ $(0 < v \leqslant 50, v \in \mathbb{N})$	$1024 \cdot v$ $(0 < v \leqslant 252, v \in \mathbb{N})$
$D^{a}_{D,buf}$	$1024 \cdot v$ $(0 < v \leqslant 25, v \in \mathbb{N})$	$1024 \cdot v$ $(0 < v \leqslant 126, v \in \mathbb{N})$
$D^{w}_{D,buf}$	$1024 \cdot v$ $(0 < v < 5, v \in \mathbb{N})$	$1024 \cdot v$ $(0 < v \leqslant 16, v \in \mathbb{N})$
$\{B_i^{w\text{-}L}\}_{i=1}^{N}$	2-8	2-8
$\{B_i^{a}\}_{i=1}^{N}$	2-4	2-4

RL search range of design factors (variables). {D_A, D_B} denotes the device {XC7Z020, XC7Z045}.

RL-based optimization framework



$$\mathcal{R} = \begin{cases} \frac{L_t - L_m}{L_t} - 1 & L_m > L_t \\ (acc_q - acc_b) \cdot \lambda & L_m \leqslant L_t \end{cases}$$

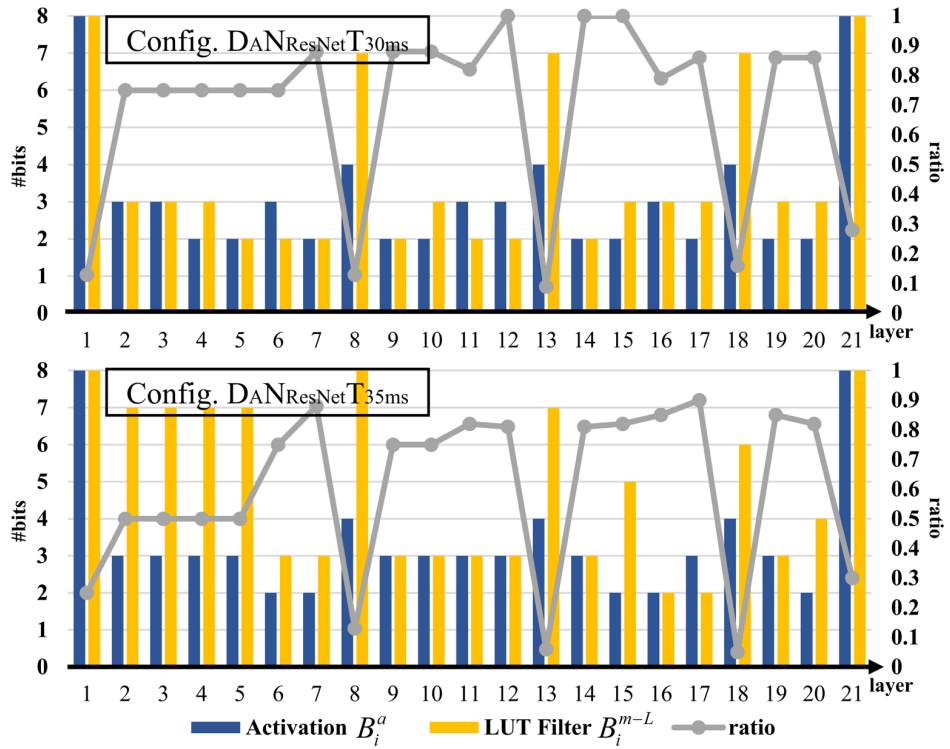
Results

Architecture design configuration automatically generated by the framework. {D, N, T} denotes {Device, Neural Network, Target latency} respectively. {D_A , D_B} denotes the device {XC7Z020, XC7Z045}, and {N_{ResNet}, $N_{MobileNet}$} denotes the target DNN of {ResNet18, MobileNet-v2}, and T$t$ms denotes the target latency set in the framework is t-ms ($t \in \{5, 6, 7, 25, 30, 35\}$).

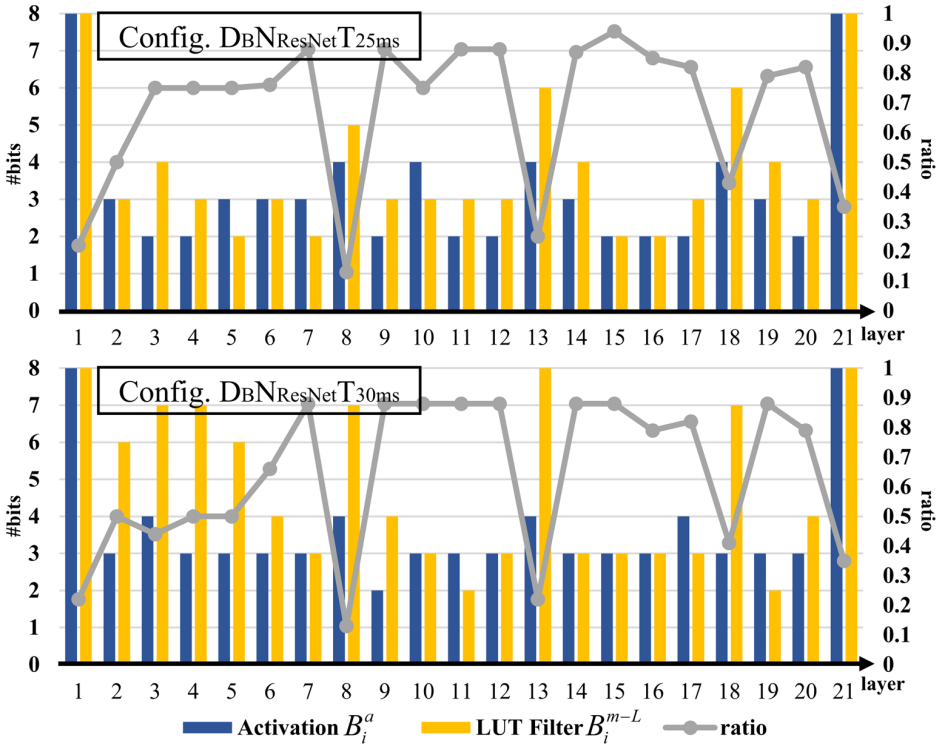
Configuration	K	M	N	$D^{a}_{L,buf}$	$D^{a}_{D,buf}$	$D^{w}_{D,buf}$
$D_A N_{ResNet} T_{30ms}$	128	7	17	1024	1024·2	1024
$D_A N_{ResNet} T_{35ms}$	128	8	16	1024	1024·2	1024
$D_A N_{MobileNet} T_{5ms}$	64	18	12	1024	1024·11	1024
$D_A N_{MobileNet} T_{7ms}$	64	26	8	1024	1024·9	1024
$D_B N_{ResNet} T_{25ms}$	512	11	17	1024	1024·6	1024·2
$D_B N_{ResNet} T_{30ms}$	512	14	14	1024	1024·15	1024
$D_B N_{MobileNet} T_{5ms}$	64	35	22	1024	1024·19	1024·11
$D_B N_{MobileNet} T_{6ms}$	64	44	18	1024	1024·20	1024·8

Results

Software Design Config. (ResNet18)



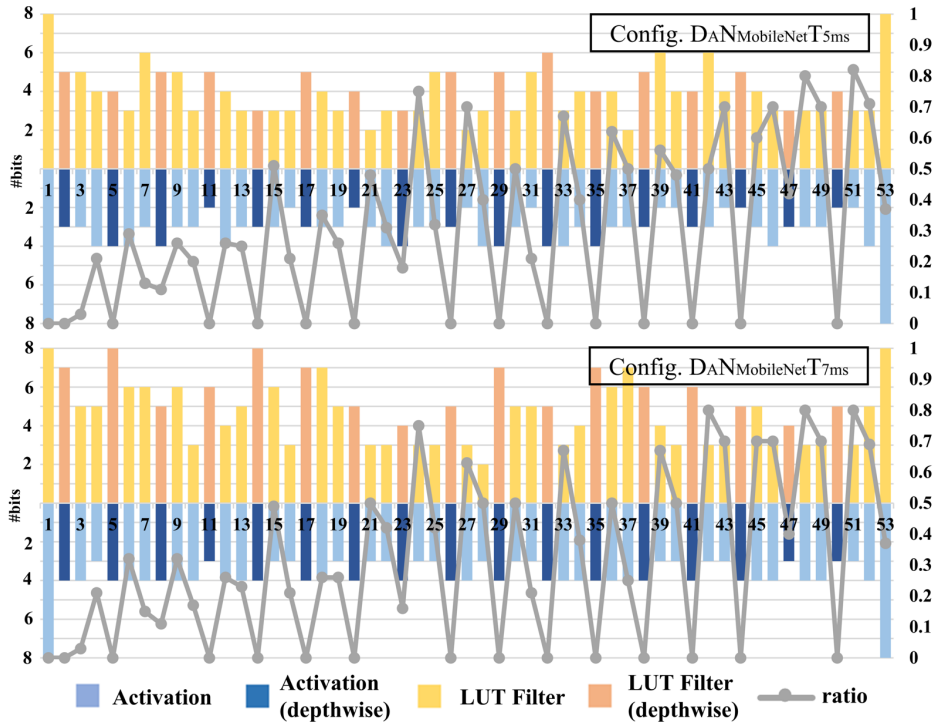
Layer-wise bit-width setting and workload split ratio in Config. $D_A N_{ResNet} T_{30ms}$ and $D_A N_{ResNet} T_{35ms}$.



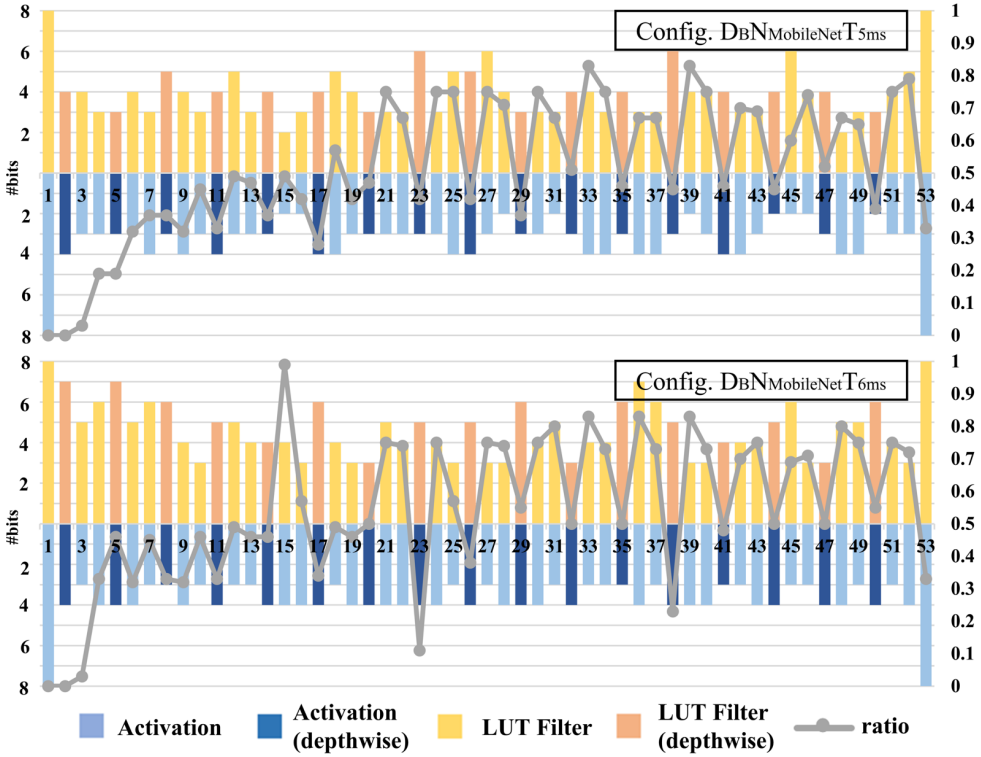
Layer-wise bit-width setting and workload split ratio in Config. $D_B N_{ResNet} T_{25ms}$ and $D_B N_{ResNet} T_{30ms}$.

Results

Software Design Config. (MobileNet-v2)



Layer-wise bit-width setting and workload split ratio in Config. $D_AN_{MobileNet}T_{5ms}$ and $D_AN_{MobileNet}T_{7ms}$.



Layer-wise bit-width setting and workload split ratio in Config. $D_BN_{MobileNet}T_{5ms}$ and $D_BN_{MobileNet}T_{6ms}$.

Results

Evaluation Accuracy on ResNet18 and MobileNet-V2:

Higher accuracy than baseline.

IDE simulated latency is **accurate enough to predict** the FPGA measured latency.

N^3H-Core Configuration	Bit width (W./A.)	Accuracy (%) (Top1/Top5)	Model latency (ms)	Measured latency (ms)
ResNet-18				
Pretrained Baseline	32/32	69.76/89.08	N/A	N/A
Manual Config. D_AN_{ResNet}	4/4	69.65/89.11	40.96	42.26
Auto. Config. $D_AN_{ResNet}T_{30ms}$	Flexible	67.28/87.85	29.14	**31.43**
Auto. Config. $D_AN_{ResNet}T_{35ms}$	Flexible	**70.45/89.54**	34.95	35.79
Manual Config. D_BN_{ResNet}	4/4	69.65/89.11	30.26	32.77
Auto. Config. $D_BN_{ResNet}T_{25ms}$	Flexible	66.81/87.19	24.83	**26.90**
Auto. Config. $D_BN_{ResNet}T_{30ms}$	Flexible	**70.39/89.33**	29.52	32.47
MobileNet-V2				
Pretrained Baseline	32/32	71.88/90.29	N/A	N/A
Mannual Config. $D_AN_{MobileNet}$	4/4	65.18/75.77	8.85	9.15
Auto. Config. $D_AN_{MobileNet}T_{5ms}$	Flexible	62.76/73.32	4.93	**5.66**
Auto. Config. $D_AN_{MobileNet}T_{7ms}$	Flexible	**66.25/76.09**	6.95	7.51
Auto. Config. $D_BN_{MobileNet}T_{5ms}$	Flexible	63.41/73.56	4.86	**5.33**
Auto. Config. $D_BN_{MobileNet}T_{6ms}$	Flexible	**66.04/75.88**	5.93	6.62

Results

Comparison with [3] (Mix and Match, Sung-En et al., 2021):

Outperforms [3] in **latency**, **throughput** and **resource utilization** (GOPS/DSP) with a comparable accuracy and GOPS/kLUT.

Comparison with [35] (Di et al., 2019):

outperforms [35] **2.48 ~ 2.89× FPS** when it operates at 100MHz.

Implementation	MobileNet-V2 [35]		ResNet-18 [3]		MobileNet-V2 [3]		ResNet-18 Ours[1]		MobileNet-V2 Ours[2]	
Device	ZU2EG	ZU9EG	XC7Z020	XC7Z045	XC7Z020	XC7Z045	XC7Z020	XC7Z045	XC7Z020	XC7Z045
Bit-width (W/A)	8/8		4/4		4/4		Flexible	Flexible	Flexible	Flexible
Top-1 Accuracy (%)	68.1		70.27		65.64		70.45	70.39	66.25	66.04
Frequency (MHz)	430	333	100		100		100		100	
LUT	31198	161944	28288	145049	28288	145049	39623	152868	45765	192624
DSP	212	2070	220	900	220	900	220	900	220	900
BRAM	145	771	56	225.5	56	225.5	137	541	137	461
Latency (ms)	-	-	47.10	40.36	8.29	7.28	35.79	32.47	7.51	6.62
Throughput (GOPS)	-	-	77.0	359.2	71.8	326.9	101.3	446.8	80.1	363.5
Frame Rate (FPS)	205.3	809.8	21.3	99.1	120.7	549.3	27.9	123.2	133.2	604.2
GOPS/DSP	-	-	0.350	0.391	0.326	0.363	0.460	0.496	0.364	0.404
GOPS/kLUT	-	-	2.725	2.475	2.538	2.252	2.557	2.923	1.750	1.887

[1] Config. $D_A N_{ResNet} T_{35ms}$ & Config. $D_B N_{ResNet} T_{30ms}$
[2] Config. $D_A N_{MobileNet} T_{7ms}$ & Config. $D_B N_{MobileNet} T_{6ms}$

Thank you.