

Binarized ImageNet Inference in 29 μ s

Tong Geng ^{*^}, Ang Li^{*}, Tianqi Wang[^], Shuaiwen Leon Song^{*}, Martin Herbordt[^]

^{*}Pacific Northwest National Laboratory (PNNL), [^]Boston University (BU)

SC'18, Dallas, TX, USA



Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by Battelle Since 1965

Introduction

BNN vs CNN:

- ✓ **Easier Computation:** floating-point-multiply-add (FMA) operations $\rightarrow$ single-bit XNOR/POPCOUNT.
- ✓ **Less Storage:** floating-point parameters and activations $\rightarrow$ single-bit except the first and last layer.
- ✓ **Potential Low Latency:** ultra-low latency is always the target of BNN.

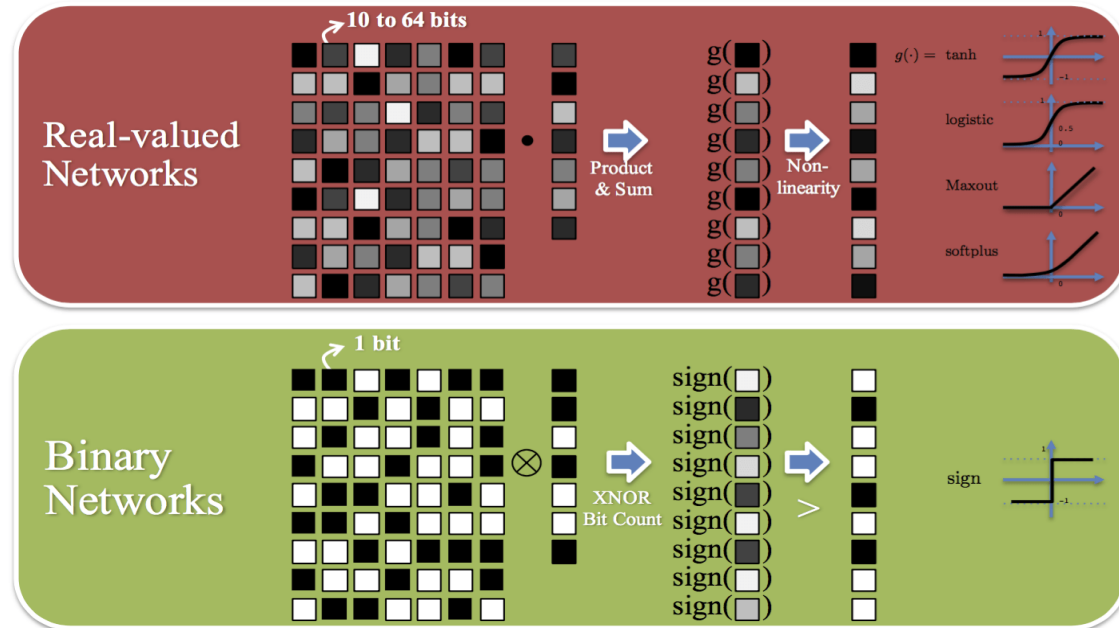


Figure 1. the difference between CNN and BNN [1]

Utilization using different platforms to accelerate the inference of AlexNet with batch size of 10:

- × **GPU:** ~ 7%. **CPU:** ~ 10%.
- ✓ **FPGA:** >60%. The configurability enables millions of one-bit ALUs to be implemented on a single device.

Challenges to make truly low-latency inference using FPGA

1. The critical Normalization Layer (NL) uses full-precision floating point (i.e., 2 FP MUL/DIV + 3 FP ADD/SUB).
2. Optimal designs for all layers need to be simultaneously configured on FPGA with no reconfiguration.
3. Existing works process layers sequentially. Hence, the their latencies are accumulated with no overlapping.

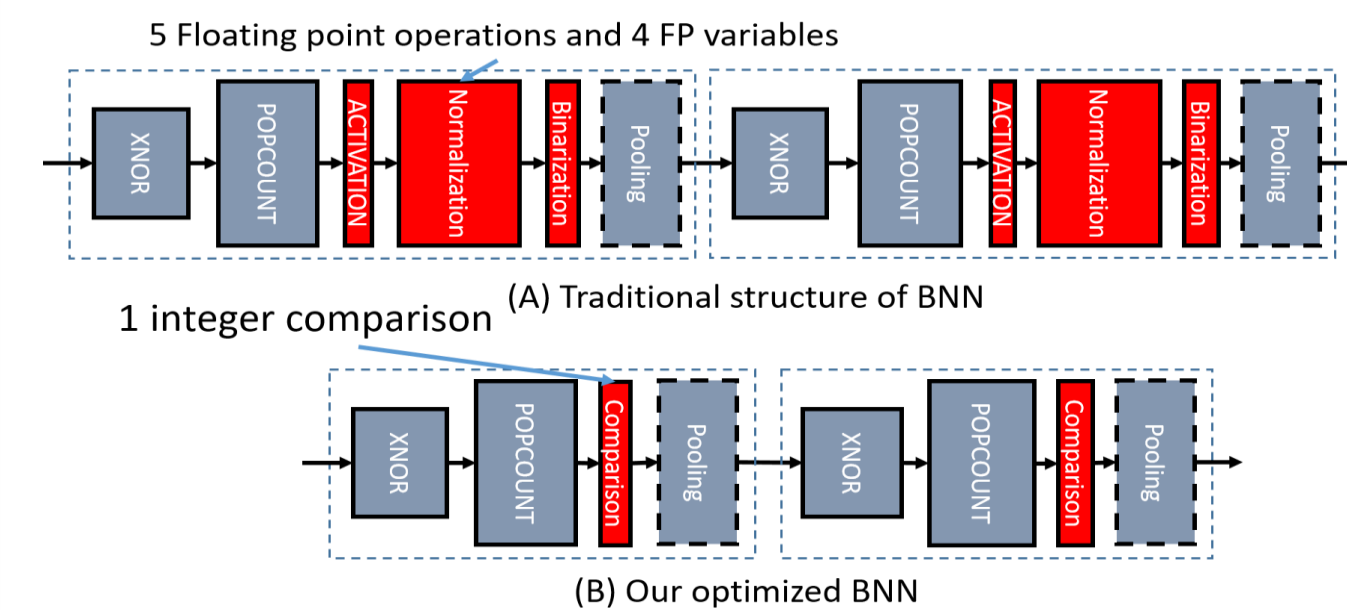
Intra-layer Fusion

Addressing the 1st challenge: Intra-layer fusion

- **3-to-1 fusion:** Activation (ACT), NL and Binarization Layer (BL) are fused and replaced by a comparison layer.
- **Simplified Computation:** Two comparison from ACT and BL and 5 floating-point operations from NL are eliminated and replaced by an integer-based comparison.

$$y_{ij} = \begin{cases} 0 & \text{if } \max(x_{ij}, 0) < E_{*j} - \frac{\beta_j * \sqrt{\text{Var}[x_{*j}] + \epsilon}}{\gamma_j} \\ 1 & \text{others} \end{cases}$$

- **Less Storage:** 4 floating-point variables (means, variance & 2 scaling factors) in NL become 1 integer threshold.
- **Lower Latency:** for single CONV and FC layer, latency is reduced: results are shown in the Table below.

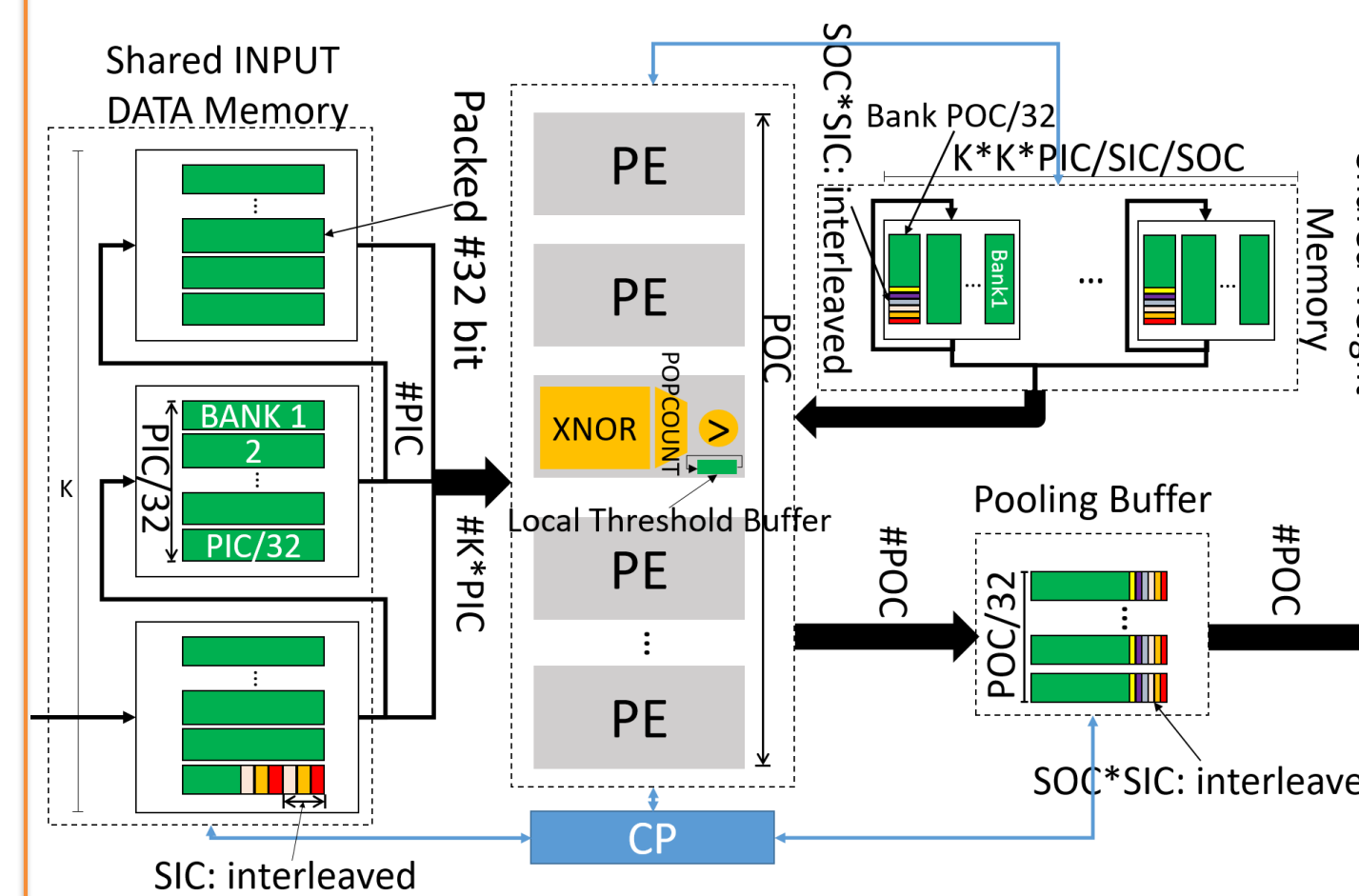


Template	Maxeler MPC-X2000 [2]		Our Work: VCU118		
Macro Layer	Time (ms)	Perf (GOP/s)	Time (ms)	Perf (GOP/s)	speedup (times)
CONV2	5.84x10 ⁻²	15348	1.63x10 ⁻²	54989	3.58
CONV3	2.03x10 ⁻²	14712	1.62x10 ⁻²	18435	1.25
CONV4	2.71x10 ⁻²	16560	1.62x10 ⁻²	27702	1.67
CONV5	1.81x10 ⁻²	16546	1.63x10 ⁻²	18373	1.11
FC1	4.31x10 ⁻³	17503	1.57x10 ⁻²	48050	2.75

Parameterized Architecture

Addressing the 2nd challenge: the decent architecture design

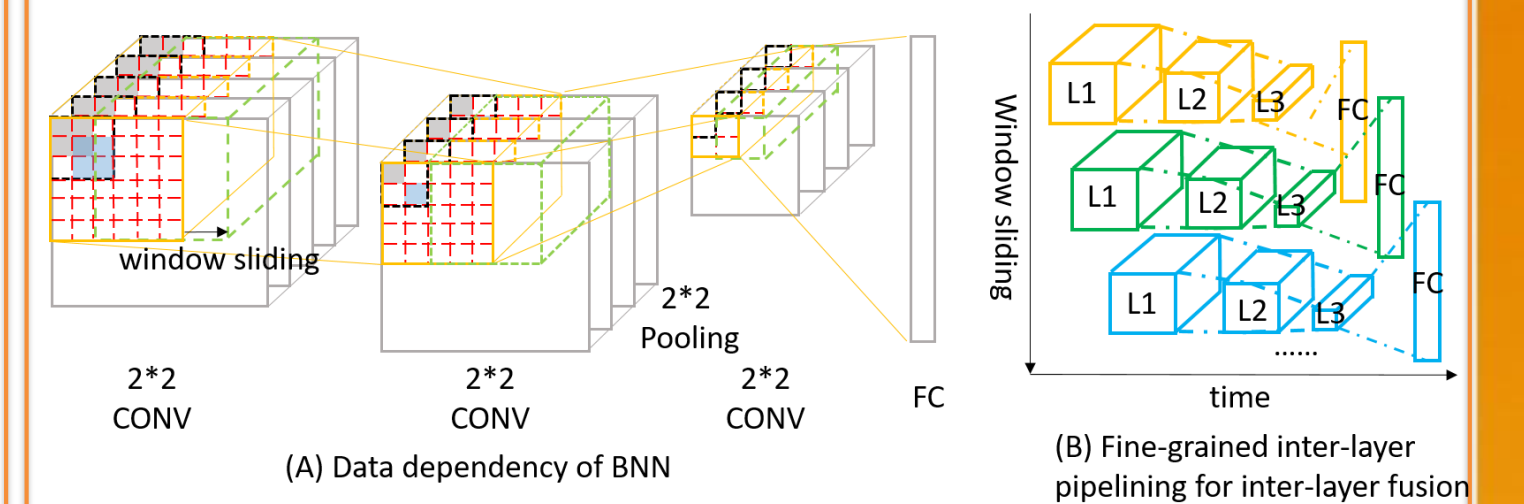
- A parameterized architecture which is flexible enough to support load balancing is proposed.
- All layers are simultaneously configured on single FPGA using model parallelism.
- **~100% utilization:** Tuning the parameters of the proposed architecture, each CONV/FC layer can be easily designed so that producing and consuming rates match.
- **Minimal Resources:** Keeping load balancing, parameters which are tuned are selected with the criteria of using minimal resources.
- **Efficient storage method:** Data and parameters for the channels processed in parallel are packed in the shared memory; while the ones for the channels processed sequentially are stored in an interleaved manner.
- All PEs work in lockstep manner and share the same Control Processor.



Inter-layer Fusion

Addressing the 3rd challenge: inter-layer fusion:

- Fine-grained pipelining to fuse CONV & 1st FC layers.
- An image is processed based on the data dependency.
- Layers are processed in parallel $\rightarrow$ overlapped latencies.



Experimental Results

	CPU	GPU	FPGA	
Platform	Xeon E5-2640 [2]	Tesla K40 [2]	Stratix V [2]	Our Work: VCU118
Latency (ms)	10789	1257	1.16 (1x)	0.029 (40x)
Performance (Image/s)	0.093	0.8	862 (1x)	34480 (40x)
Energy Efficiency (Image/J)	0.00098	0.0034	32.9 (1x)	1078 (32.7x)
Frequency (MHz)	2500	810	150	200
Utilization	-	-	-	99.7%
LUT	635k/1182k (53.7%)	FF	BRAM	DSP
		1527k/2364k (64.6%)	1459/2160 (67.5%)	5808/6840 (84.9%)

[1] "Accelerating Neural Networks with Binary Arithmetic," <https://ai.intel.com/accelerating-neural-networks-binary-arithmetic>.

[2] Shuang Liang, Shouyi Yin, Leibo Liu, Wayne Luk, and Shaojun Wei. 2018. FP-BNN: Binarized neural network on FPGA. Neurocomputing 275 (2018)