13.3 UNPU: A 50.6TOPS/W Unified Deep Neural Network Accelerator with 1b-to-16b Fully-Variable Weight Bit-Precision

Jinmook Lee, Changhyeon Kim, Sanghoon Kang, Dongjoo Shin, Sangyeob Kim, Hoi-Jun Yoo

KAIST, Daejeon, Korea

Deep neural network (DNN) accelerators [1-3] have been proposed to accelerate deep learning algorithms from face recognition to emotion recognition in mobile or embedded environments [3]. However, most works accelerate only the convolutional layers (CLs) or fully-connected layers (FCLs), and different DNNs, such as those containing recurrent layers (RLs) (useful for emotion recognition) have not been supported in hardware. A combined CNN-RNN accelerator [1], separately optimizing the computation-dominant CLs, and memory-dominant RLs or FCLs, was reported to increase overall performance, however, the number of processing elements (PEs) for CLs and RLs was limited by their area and consequently, performance was suboptimal in scenarios requiring only CLs or only RLs. Although the PEs for RLs can be reconfigured into PEs for CLs or vice versa, only a partial reconfiguration was possible resulting in marginal performance improvement. Moreover, previous works [1-2] supported a limited set of weight bit precisions, such as either 4b or 8b or 16b. However, lower weight bit-precisions can achieve better throughput and higher energy efficiency, and the optimal bit-precision can be varied according to different accuracy/performance requirements. Therefore, a unified DNN accelerator with fully-variable weight bit-precision is required for the energy-optimal operation of DNNs within a mobile environment.

In this paper, we present a unified neural processing unit (UNPU) supporting CLs, RLs, and FCLs with fully-variable weight bit-precision from 1b to 16b. As shown in Fig. 13.3.1, the reuse of input features (IFs) is more efficient than the reuse of weights under low-weight bit-precision and the operations of CLs become identical to those of RLs and FCLs when the IFs of the CLs are vectorized into a 1-dimensional vector so that the hardware can be fully shared in the UNPU by IF reuse. Moreover, the lookup-table-based bit-serial PE (LBPE) is implemented for energy-optimal DNN operations with variable-weight bit-precisions from 1b to 16b through iterations of 1b weight operations. Furthermore, an aligned feature loader (AFL) minimizes the amount of off-chip memory accesses required to fetch IFs by exploiting the data locality among convolution operations.

Figure 13.3.2 shows the overall architecture of the UNPU. It consists of 4 DNN cores, an aggregation core, a 1D SIMD core, and a RISC controller. All of these components are connected to an on-chip network for communication. Each DNN core has 6 LBPEs, 6 AFLs (64×6), a weight memory (48KB), an instruction decoder and a controller. The LBPE receives aligned IF as an input operand through AFLs and calculates 576 (4×12×12) multiplications in parallel in a bit-serial manner. The partial-sums (Psums) calculated by each DNN core are aggregated to an output feature (OF) in the aggregation core. The 1D SIMD core performs the remaining operations, such as non-linear activation or pooling, and the results are stored in off-chip memory through the external gateways.

Figure 13.3.3 elaborates on workload allocation. For RLs and FCLs, its 1D IF is mapped to AFLs with one-to-one (48×1) and sent to a PE. The weights are loaded from 12 channels of OF (48×12b) to calculate multiple channels of Psums with the same IF. For a CL, IFs distributed over multiple input channels are concatenated into a 1D row vector and loaded into the AFLs, as is done with RLs and FCLs. The weights of CLs are converted into 1D column vectors and then the Psums are calculated by multiplying with the 1D IF row vector. 4 LBPEs in a DNN core calculate the product between 48 pairs of IFs and weights, and each LBPE corresponds to 12 IF-weight pairs. The IF is reused for multiple column vectors from other channels. The Psums from each PE are accumulated by 12 adder trees. The weights are reused among the 6 LBPEs for better energy efficiency. For example, in RLs and FCLs, the 6 different IFs are assigned to 6 LBPEs in parallel with the same weights if batch-wise parallelism is possible. For a CL, the 6 consecutive IFs in the same channel are multiplied with the same weights in 6 different LBPEs in parallel. Peak performance for CLs and RLs (or FCLs) is increased by 1.15× and 13.8×, respectively, compared to [1] owing to the higher compute density of the unified DNN core.

Figure 13.3.4 shows the architecture of the LBPE. The key idea of the LBPE is that partial-sums are repeatedly calculated during the weight bit-serial MAC operation. A LBPE consists of 4 PE clusters, adder trees to accumulate the results of each PE cluster, and shift-and-add logic for bit-serial multiplications. Each PE cluster contains 4 look-up-table (LUT) modules and a controller that determines whether the value from LUTs is added or subtracted. In the LUT module, a table with 8 entries is used, supporting 3-way MAC for multi-bit multiplication and 4-way MAC for 1b multiplication. The LUT is updated after IFs load into the AFLs, and IF values are reused for all output channels of the layer currently being processed. The 1b weight Psums are fetched from the LUT prepared in advance and accumulated for MAC operation. The LUT can fetch 12 Psums in parallel so that a total of 48×12 Psums (64×12 for 1b case) can be calculated simultaneously on a LBPE in 1 cycle. With the help of table-based operations, the LBPE improves energy efficiency more than conventional bit-serial PEs [4]. When IFs are reused 1024 times, the energy-consumption of LBPEs, including the LUT update, is reduced by 23.1%, 27.2%, 41.0%, and 53.6%, for the case of 16b, 8b, 4b, and 1b weight operations, respectively, compared with fixed-point MAC units under the same throughput conditions.

Figure 13.3.5 explains the AFL. 6 AFL-LBPE pairs are integrated in a DNN core and each AFL has 64 entries. The data in the AFL can be shifted diagonally across AFL boundaries, as well as shifted inside the AFL itself. In the case of CLs with 3×3 kernels and stride 1, 8 entries of IF from Ch. 1 are loaded on AFL 0 at first. At the next cycle, the 7 top entries of AFL 0, except the top-most entry, are shifted diagonally to AFL 1, while the 8 entries from Ch. 2 are concatenated below the remaining 3 entries of Ch. 1 on AFL 0. And then, 8 entries from Ch. 3 are concatenated below the remaining 3 entries of Ch. 2 on AFL 0, while 6 entries on AFL 1 from Ch. 1 are shifted diagonally to AFL 2, and 7 entries from Ch. 2 are shifted diagonally to concatenate below the remaining 3 entries from Ch. 1 on AFL 1. Iterations of diagonal shifts allocate a 3×3 kernel to each AFL or LBPE so that parallel multiplication is possible to accelerate convolution. Varied stride sizes are supported via the application of multiple shifts. The AFL keeps the PE utilization high, unlike an architecture that moves data only between PEs. In addition, it can skip zeros by an upward-shift within the buffer. When the AFL is applied to AlexNet and VGG-16, external memory access operations for IF load are reduced by 57.2% and 55%, respectively.

Figure 13.3.6 shows measurement results for the fabricated UNPU. The UNPU can operate at 0.63-to-1.1V supply voltage with a maximum 200MHz clock frequency. The power consumption at 0.63V and 1.1V is 3.2mW and 297mW, respectively. The power-efficiency, as measured on CLs (5×5 kernels) with consideration of PE utilization is 3.08, 11.6, and 50.6TOPS/W for the case of 16b, 4b, and 1b weights, respectively. The architecture supports any weight bit-precision from 1b to 16b for optimal DNN operation and shows 1.43× higher power efficiency for CLs at 4b weight compared to [1]. When operating on a 1b weight network, it achieves 8.43× higher efficiency and 7.4× higher peak performance as compared to [6].

The UNPU is fabricated using 65nm CMOS technology and occupies 16mm^2 die area, as shown in the Fig. 13.3.7. The UNPU has been demonstrated successfully on facial expression recognition and dialogue generation tasks with the FER2013 and the Twitter dialogue database for human-computer interaction, respectively.

References:

[1] D. Shin, et al., "DNPU: An 8.1 TOPS/W Reconfigurable CNN-RNN Processor for General-Purpose Deep Neural Networks," *ISSCC*, pp. 240-241, 2017

[2] B. Moons, et al., "Envision: A 0.26-to-10TOPS/W Subword-Parallel Dynamic-Coltage-Accuracy-Frequency-Scalable Convolutional Neural Network processor in 28nm FDSOI," *ISSCC*, pp. 246-247, 2017.

[3] K. Bong, et al., "A 0.62mW Ultra-Low-Power Convolutional-Neural-Network Face-Recognition Processor and a CIS Integrated with Allways-On Haar-Like Face Detector," *ISSCC*, pp.248-249, 2017

[4] P. Judd, et al., "Stripes: Bit-serial Deep Neural Network Computing," *IEEE Computer Architecture Letters*, vol. 16, no. 1, pp. 80-83, Jan.-June 1 2017.

[5] S. Yin, et al., "A 1.06-to-5.09 TOPS/W Reconfigurable Hybrid-Neural-Network Processor for Deep Learning Applications," *IEEE Symp. VLSI Circuits*, 2017.

[6] K. Ando, et al., "BRein memory: A 13-Layer 4.2 K Neuron/0.8 M Synapse Binary/Ternary Reconfigurable In-Memory Deep Neural Network Accelerator in 65 nm CMOS," *IEEE Symp. VLSI Circuits*, 2017.

 978-1-5090-4940-0/18/$31.00

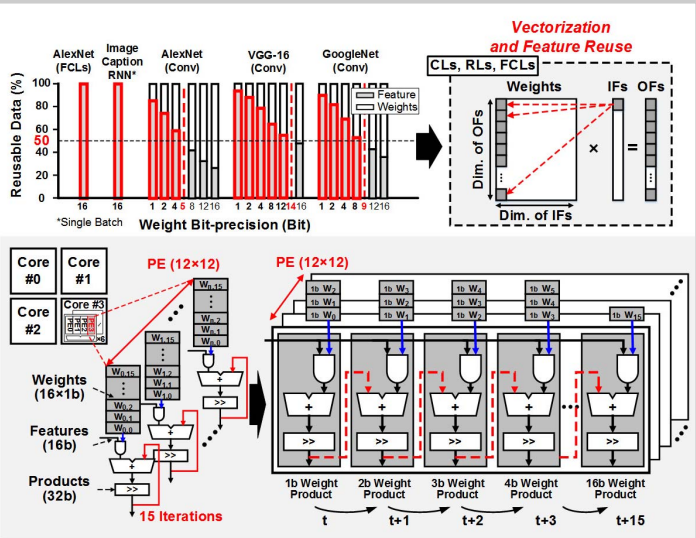


Figure 13.3.1: Fully reconfigurable unified DNN accelerator with bit-serial PEs.

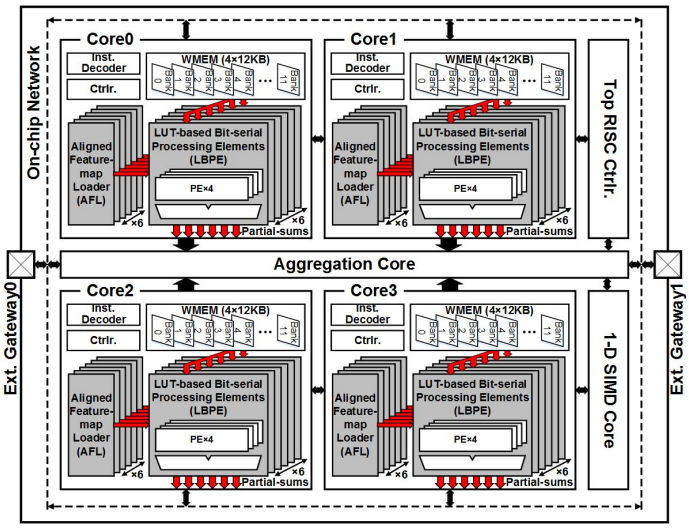


Figure 13.3.2: Overall architecture.

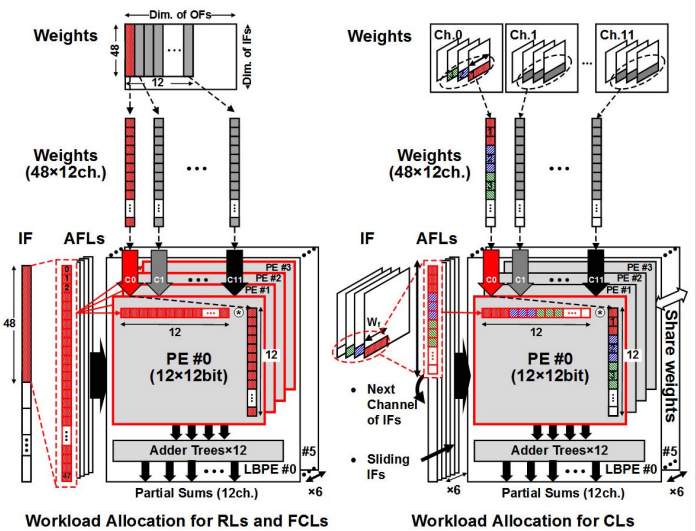


Figure 13.3.3: Workload allocation on the unified DNN core.

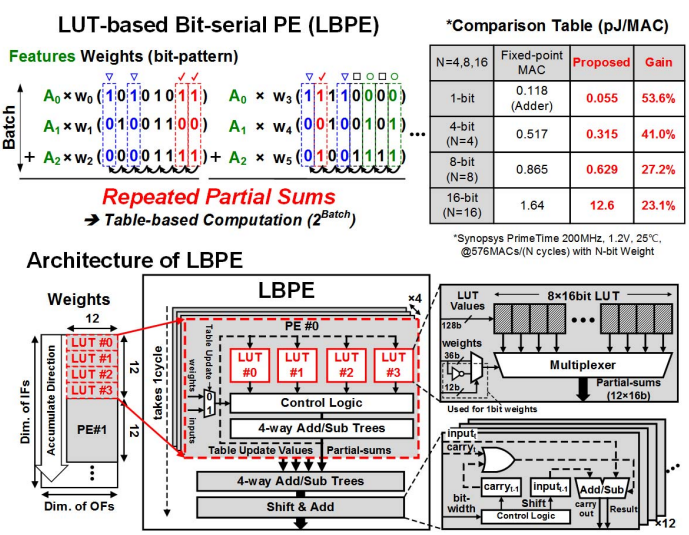


Figure 13.3.4: LUT-based bit-serial processing elements.

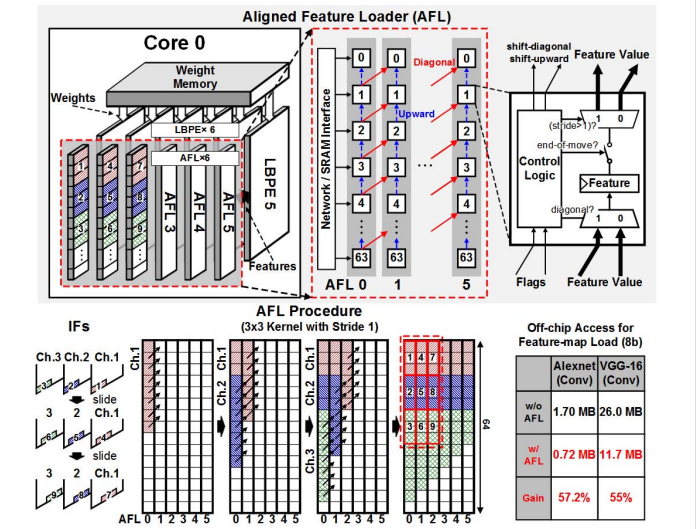
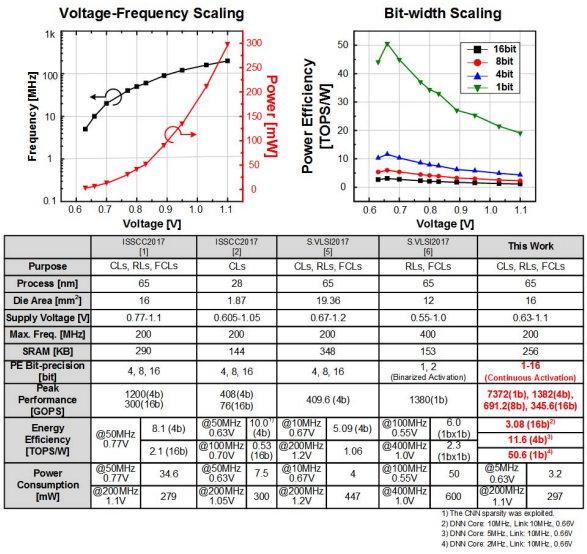


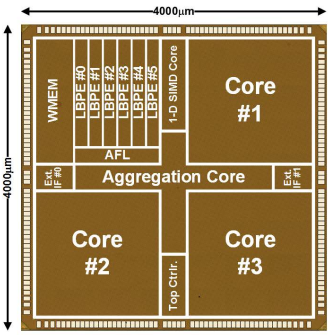
Figure 13.3.5: Aligned feature loader for reduction of off-chip memory accesses.



	ISSCC2017 [1]		ISSCC2017 [2]		S.VLSI2017 [5]	S.VLSI2017 [6]	This Work
Purpose	CLs, RLs, FCLs		CLs		CLs, RLs, FCLs	RLs, FCLs	CLs, RLs, FCLs
Process [nm]	65		28		65	65	65
Die Area [mm^2]	16		1.87		19.36	12	16
Supply Voltage [V]	0.77-1.1		0.605-1.05		0.67-1.2	0.55-1.0	0.63-1.1
Max. Freq. [MHz]	200		200		200	400	200
SRAM [KB]	290		144		348	153	256
PE Bit-precision [bit]	4, 8, 16		4, 8, 16		4, 8, 16	1, 2 (Binarized Activation)	1-16 (Continuous Activation)
Peak Performance [GOPS]	1200(4b) 300(16b)		408(4b) 76(16b)		409.6 (4b)	1380(1b)	7372(1b), 1382(4b), 691.2(8b), 345.6(16b)
Energy Efficiency [TOPS/W]	@50MHz 0.77V	8.1 (4b)	@50MHz 0.63V	10.0[1] (4b)	@10MHz 0.67V	5.09 (4b)	@100MHz 0.55V	6.0 (1bx1b)	3.08 (16b)[3]
		2.1 (16b)	@100MHz 0.70V	0.53 (16b)	@200MHz 1.2V	1.06	@400MHz 1.0V	2.3 (1bx1b)	11.6 (4b)[3] 50.6 (1b)[4]
Power Consumption [mW]	@50MHz 0.77V	34.6	@50MHz 0.63V	7.5	@10MHz 0.67V	4	@100MHz 0.55V	50	@5MHz 0.63V 3.2
	@200MHz 1.1V	279	@200MHz 1.05V	300	@200MHz 1.2V	447	@400MHz 1.0V	600	@200MHz 1.1V 297

1) The CNN sparsity was exploited.
2) DNN Core: 10MHz, Link:10MHz, 0.66V
3) DNN Core: 5MHz, Link: 10MHz, 0.66V
4) DNN Core: 2MHz, Link: 10MHz, 0.66V

Figure 13.3.6: Measurement results and performance comparison table.



	Specifications	
Technology	65nm 1P8M Logic CMOS	
Die Area	4000×4000μm²	
SRAM	256KB	
Supply @ (Nominal)	1.1V	
Frequency (Nominal)	200MHz	
Weight Bit-precision	1 - 16	
Power Consumption [mW]	297 @200MHz, 1.1V	
	3.2 @5MHz, 0.63V	
Power Efficiency [TOPS/W]	CLs	50.6 (1b)
		11.6 (4b)
		5.57 (8b)
	*FCLs, RLs	46.7 (1b)
		10.3 (4b)
		5.32 (8b)

* Evaluated on 512x512 Weights

Figure 13.3.7: Chip micrograph and performance summary.

978-1-5090-4940-0/18/$31.00