14.7 A 288µW Programmable Deep-Learning Processor with 270KB On-Chip Weight Storage Using Non-Uniform Memory Hierarchy for Mobile Intelligence

Suyoung Bang[1], Jingcheng Wang[1], Ziyun Li[1], Cao Gao[1], Yejoong Kim[1,2], Qing Dong[1], Yen-Po Chen[1], Laura Fick[1], Xun Sun[1], Ron Dreslinski[1], Trevor Mudge[1], Hun Seok Kim[1], David Blaauw[1], Dennis Sylvester[1]

[1]University of Michigan, Ann Arbor, MI
[2]CubeWorks, Ann Arbor, MI

Deep learning has proven to be a powerful tool for a wide range of applications, such as speech recognition and object detection, among others. Recently there has been increased interest in deep learning for mobile IoT [1] to enable intelligence at the edge and shield the cloud from a deluge of data by only forwarding meaningful events. This hierarchical intelligence thereby enhances radio bandwidth and power efficiency by trading-off computation and communication at edge devices. Since many mobile applications are “always-on” (e.g., voice commands), low power is a critical design constraint. However, prior works have focused on high performance reconfigurable processors [2-3] optimized for large-scale deep neural networks (DNNs) that consume >50mW. Off-chip weight storage in DRAM is also common in the prior works [2-3], which implies significant additional power consumption due to intensive off-chip data movement.

We introduce a low-power, programmable deep learning accelerator (DLA) with all weights stored on-chip for mobile intelligence. Low power (<300µW) is achieved through 4 techniques: 1) Four processing elements (PEs) are located amidst the weight storage memory of 270kB, minimizing data movement overhead; 2) A non-uniform memory hierarchy provides a trade-off between small, low-power memory banks for frequently used data (e.g., input neurons) and larger, high density banks with higher power for the large amount of infrequently accessed data (e.g., synaptic weights). This exploits the key observation that deep learning algorithms can be deterministically scheduled at compilation time, predetermining optimal memory assignments and avoiding the need for traditional caches with significant power/area overhead; 3) A 0.6V 8T custom memory is specifically designed for DNNs with a sequential access mode, bank-by-bank drowsy mode control, power-gating for peripheral circuits, and voltage clamping for data retention; 4) Highly flexible and compact memory storage is realized via independent control of reconfigurable fixed-point bit precision ranging from 6–32b for neurons and weights. These techniques were implemented into a complete deep learning processor in 40nm CMOS, including the DLA, an ARM Cortex-M0 processor, and MBus [4] interface to enable integration into a complete sensor system (Fig. 14.7.1). The DLA consumes 288µW and achieves 374GOPS/W efficiency. We demonstrate full system operation for two mobile-oriented applications, keyword spotting and face detection.

In the DLA, a non-uniform memory access (NUMA) architecture is carefully designed to strike a balance between memory area and access energy. Fig. 14.7.1 shows that smaller SRAM banks have lower access energy with relatively worse density, while the opposite is true for larger banks. The number of NUMA hierarchical levels and the memory size of each hierarchy were determined via extensive simulations that analyzed NUMA configurations for various DNN topologies. In the proposed architecture, NUMA memory has 67.5kB in total with four banks in each level of hierarchy. Unit bank sizes are 0.375, 1.5, 3, and 12kB (Fig. 14.7.1). The DLA operations are optimized for implementing the fully-connected layer (FCL) in deep neural networks. The FCL performs matrix-vector multiplication, offset addition, and a non-linear activation function. The proposed NUMA architecture leverages the energy efficiency of the NUMA architecture by partitioning a large FCL to form multiple smaller ‘tiles’ to optimally fit in the proposed NUMA architecture (Fig. 14.7.2). A matrix-vector tile uses the same input vector for all rows. Therefore, we strategically map the input vector to the nearest local memory so that the DLA can reuse it as many times as possible once loaded. An example of NUMA-based FCL computation is illustrated in Fig. 14.7.2, where a weight (W) matrix is first partitioned into two tiles, input neurons for each tile are then loaded to nearby memory, and finally the output neurons of each tile are accumulated using local memory. Infrequently accessed W tiles are loaded from dense (but higher access energy) upper hierarchy memory. At compilation time, the optimal memory access pattern is statically scheduled, and corresponding W matrix tiling is determined to maximize energy efficiency by exploiting NUMA. Simulations show that combining NUMA with the tiling strategy for 4 PEs leads to >40% energy saving with 2% area overhead compared to UMA (unit bank = 16kB) for the same tasks and total memory capacity (Fig. 14.7.1). The tiling approach exploiting NUMA locality is also used to perform energy efficient FFT operations by the DLA. By incorporating FFT operations, the DLA can support convolutional layers by transforming convolution to a matrix-vector multiplication in the frequency domain.

Figure 14.7.3 shows the overall DLA architecture. The DLA has four PEs surrounded by their memory with a 4-level NUMA architecture (Fig. 14.7.4). Each PE has an instruction buffer, status register, data buffers, controller, ALU, memory address mapping unit, and memory arbitration unit. Data buffers perform data unpacking (packing) from (to) 96b to enable configurable data precisions; 6/8/12/16b for weights and neurons, and 16/24/32b for accumulation, which are shifted and truncated when stored as next layer neuron inputs. The PE is programmed by two ping-pong CISC instruction registers, which are 192b long including start address, size, precision, and operation-specific flags. The reconfigurable PE CISC operations are: 1) FCL processing, 2) FFT, 3) data-block move, and 4) LUT-based non-linear activation function. The memory address mapping unit and memory arbitration unit in each PE governs prioritized memory access arbitration, enabling a PE to access another PE’s memory space. PEs can be programmed via offline scheduling optimization to avoid memory access collisions and contamination. The DLA operation sequence is controlled by the Cortex-M0, which loads data and instructions into PE memory. As a PE instruction can take many cycles to complete, the Cortex-M0 supports clock-gating and it wakes upon PE completion. An external host processor can program the Cortex-M0 and DLA using a serial bus interface.

PE NUMA memory uses custom SRAM banks with a HVT 8T bitcell for low leakage. Each bank consists of sub-arrays to share an address decoder and read-out circuits for access power reduction (Fig. 14.7.4). PE memory uses gating circuits to prevent unnecessary signal switching in hierarchical memory accesses. That is, lower level memory access signals do not propagate to higher levels (Fig. 14.7.4). The optimal tiling and deterministic scheduling allow further optimization of the memory address decoder using a sequential access mode. Given that only a few banks are actively accessed in a specific PE while the others stay idle during the majority of processing time (due to the static tiling schedule), we employ a dynamic drowsy mode for SRAM leakage reduction. Each PE dynamically controls power gating and clamping headers of SRAM peripheral circuits and arrays, bank-by-bank based on the schedule (Fig. 14.7.4). During drowsy mode, peripherals are power-gated, but array voltage is clamped with an NMOS header and a programmable on-chip V_{REF} to ensure data retention.

Measurement results of the 40nm CMOS test chip confirm effectiveness of the proposed NUMA and drowsy mode operation (Figs. 14.7.5, 14.7.6). Measured data access power consumption in L1 is 60% less than in L4. Memory drowsy-mode operation reduces leakage by 54%, which is mainly attributed to peripheral circuits as the bitcell is inherently low leakage. The test chip achieves peak efficiency of 374GOPS/W while consuming 288µW at 0.65V and 3.9MHz. Keyword spotting (10 keywords) and face detection (binary decision) DNNs are successfully ported onto the proposed DLA with layer dimensions and precision mapping specified in Fig. 14.7.6. Both DNN classifications fit into the 270kB on-chip memory and exhibit <7ms latency, allowing for real-time operation. Fig. 14.7.6 compares against state-of-the-art prior work and Fig. 14.7.7 shows the die photo.

Acknowledgements:
The authors thank TSMC University Shuttle Program for chip fabrication.

References:

[1] N. D. Lane, et al., “Can Deep Learning Revolutionize Mobile Sensing?” *ACM International Workshop on Mobil Computing Systems and Applications* (HotMobile), pp. 117-122, 2015.
[2] Y.-H. Chen, et al., “Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks,” *ISSCC*, pp. 262-263, 2016.
[3] G. Moons, et al., “A 0.3-2.6 TOPS/W Precision-Scalable Processor for Real-Time Large-Scale ConvNets,” *IEEE Symp. VLSI Circuits*, 2016.
[4] G. Kim, et al., “A Millimeter-Scale Wireless Imaging System with Continuous Motion Detection And Energy Harvesting,” *IEEE Symp. VLSI Circuits*, 2014.

978-1-5090-3758-2/17/$31.00 ©2017 IEEE

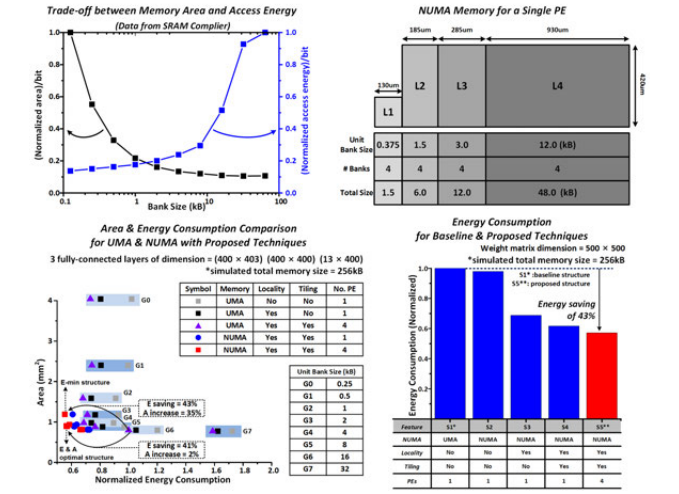

Figure 14.7.1: SRAM area and access energy trade-off (top left). Proposed NUMA memory for a PE (top right). Area and energy comparison with UMA & NUMA and proposed techniques (bottom).

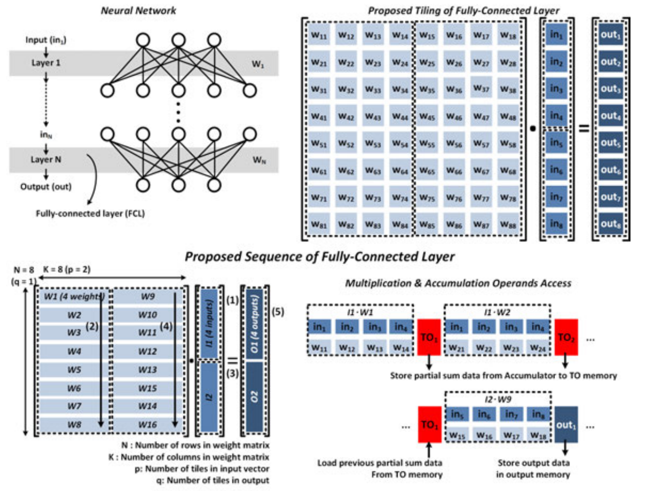
Neural Network
Input (in₁)
Layer 1
W₁
in_N
Layer N
W_N
Output (out)
Fully-connected layer (FCL)
Proposed Tiling of Fully-Connected Layer
W₁₁ … W₈₈
in₁ … in₈
out₁ … out₈
Proposed Sequence of Fully-Connected Layer
N = 8
K = 8 (p = 2)
(q = 1)
W1 (4 weights)
W2
W3
W4
W5
W6
W7
W8
W9
W10
W11
W12
W13
W15
W14
W16
(1)
(2)
(3)
(4)
(5)
I1 (4 inputs)
I2
O1 (4 outputs)
O2
N : Number of rows in weight matrix
K : Number of columns in weight matrix
p: Number of tiles in input vector
q: Number of tiles in output
Multiplication & Accumulation Operands Access
I1 · W1
I1 · W2
I2 · W9
TO₁
TO₂
out₁
Store partial sum data from Accumulator to TO memory
Load previous partial sum data
From TO memory
Store output data
in output memory

Figure 14.7.2: A neural network with fully-connected layers (top left). Proposed tiling of fully-connected layer (top right). Proposed operation sequence of fully-connected layer (bottom).

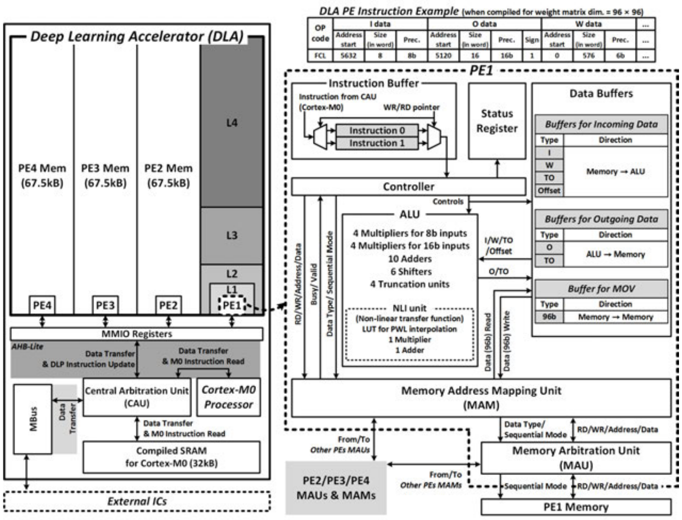

Figure 14.7.3: Top-level diagram of proposed deep learning accelerator (DLA) (left). DLA PE instruction example (top). DLA PE block diagram (right).

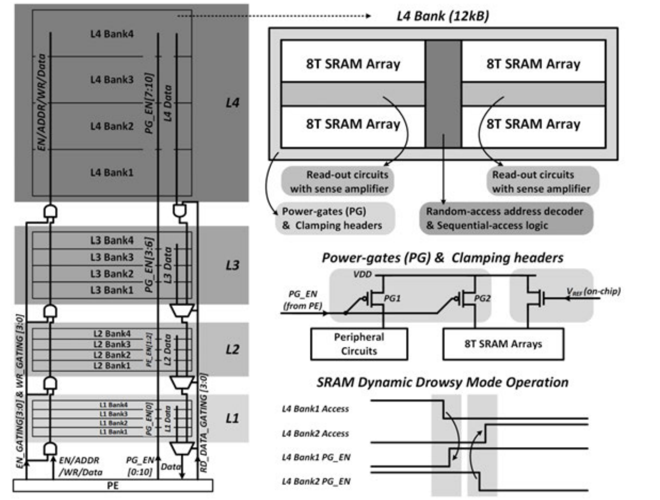

Figure 14.7.4: PE NUMA memory floorplan with signal gating circuits (left). L4 bank floorplan (top right). Power-gates and clamping headers, and dynamic drowsy mode operation (bottom right).

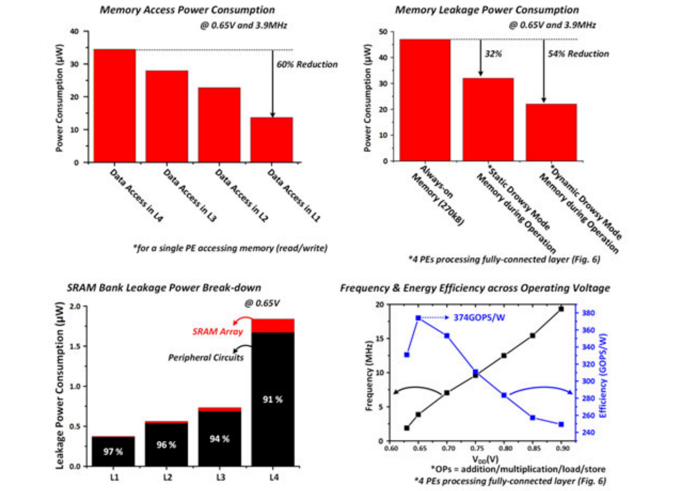

Figure 14.7.5: Memory access power consumption (top left). Memory leakage power comparison (top right). SRAM bank leakage break-down (bottom left). Performance and efficiency across voltage (bottom right).

Performance Summary

App.	Layer*	#PE in use	Weight matrix	Precision (bit)				Power (μW)	Latency (msec)	GOPS/W
				W	I	O	TO			
General	1st	4	384 × 408	6	12	12	32	288	2.97	374

App.	Layer*	#PE in use	Weight matrix	Precision (bit)				Power (μW)	Latency (msec)	GOPS/W
				W	I	O	TO			
Keyword spotting	1st	4	384 × 408	6	12	12	32	360	2.98	316
	2nd	4	384 × 384	6	12	12	32	312	2.80	365
	3rd	1	24 × 384	6	12	12	32	189	0.72	140
Total								321	6.50	318
Face detection	1st	1	16 × 1032	8	8	16	32	208	1.70	113
	2nd	1	2 × 16	8	16	16	32	195	0.02	112
Total								208	1.72	113

@ 0.65V and 3.9MHz
*Fully-connected layer

Comparison Table

	This work	[2]	[3]
Technology	40nm	65nm	40nm
Chip area	7.1mm^2	16mm^2	2.4mm^2
Design target	Fully-connected layer & FFT	Convolutional layer	Convolutional layer
Operating voltage	0.63 – 0.9 V	0.82 – 1.17 V	0.55 – 1.1 V
Operating frequency	1.9 – 19.3 MHz	100 – 250 MHz	12 – 204 MHz
Efficiency	374 GOPS/W*,†	9.6 GOPS/W‡,#	300 – 2600 GOPS/W‡
Operating power	0.288 mW†	278 mW‡	76 mW‡
On-chip core SRAM size	270 kB	181.5 kB	144 kB
Off-chip memory requirement	No	Yes	Yes
Variable precision	6/8/12/16/24/32-bit	No	1 – 16-bit

*OPs = addition/multiplication/load/store. †Operating at 0.65V and 3.9MHz.
‡DRAM access power not included. #Calculated based on reported GOPS and power.

Figure 14.7.6: Performance summary for neural networks with a variety of layer specification (top). Comparison table (bottom).

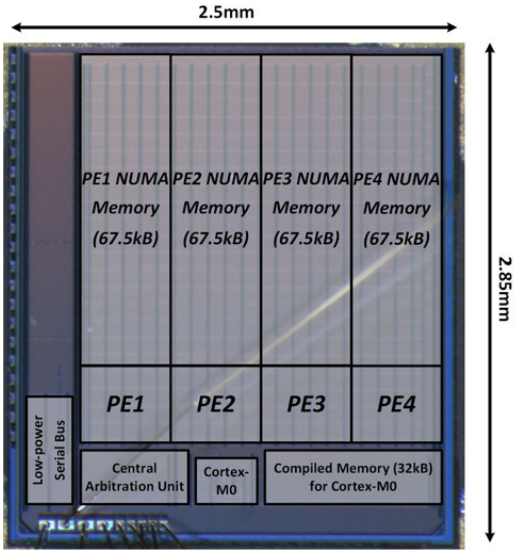

Figure 14.7.7: Die photo.