**Session 14 Overview: Deep-Learning Processors**

**DIGITAL ARCHITECTURE AND SYSTEMS SUBCOMMITTEE**

**Session Chair:** Takashi Hashimoto, STMicroelectronics, Cornaredo, Italy

**Session Co-Chair:** Mahesh Mehendale, Texas Instruments, Bangalore, India

**Subcommittee Chair:** Byeong-Gyu Nam, Chungnam National University, Daejeon, Korea

Processors targeting embedded perception and cognition have evolved tremendously in the past decade. Thanks to CMOS process scaling, the cost in terms of area and energy per operation have decreased drastically. This makes it feasible to equip next-generation processing with human-like intelligence for emerging applications, such as detection, recognition, and classification.

This session covers highly integrated embedded next-generation processing systems for improved accuracy of speech recognition, face recognition for next-generation UI/UX of wearable devices. In addition, programmable or scalable deep-learning accelerators for Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), and fully programmable deep learning SoCs are presented. The session concludes with a data processor for Next-Generation Sequencing (NGS).

### 1:30 PM

**14.1 A 2.9TOPS/W Deep Convolutional Neural Network SoC in FD-SOI 28nm for Intelligent Embedded Systems**

G. Desoli, STMicroelectronics, Cornaredo, Italy

In Paper 14.1, STMicroelectronics presents a deep convolutional neural network SoC in 28nm FD-SOI with energy efficiency of 2.9TOPS/W and peak performance of more than 676GOPS, operating at 200MHz with supply voltage of 0.575V.

### 2:00 PM

**14.2 DNPU: An 8.1TOPS/W Reconfigurable CNN-RNN Processor for General-Purpose Deep Neural Networks**

D. Shin, KAIST, Daejeon, Korea

In Paper 14.2, KAIST presents a reconfigurable CNN-RNN processor SoC in 65nm CMOS with energy efficiency of 8.1TOPS/W, operating at 50MHz with supply voltage of 0.77V.

### 2:30 PM

**14.3 A 28nm SoC with a 1.2GHz 568nJ/Prediction Sparse Deep-Neural-Network Engine with >0.1 Timing Error Rate Tolerance for IoT Applications**

P. N. Whatmough, Harvard University, Cambridge, MA

In Paper 14.3, Harvard University presents a fully connected (FC)-DNN accelerator SoC in 28nm CMOS, which achieves 98.5% accuracy for MNIST inference with 0.36µJ/prediction at 667MHz and 0.57µJ/pred at 1.2GHz.
3:15 PM
14.4 A Scalable Speech Recognizer with Deep-Neural-Network Acoustic Models and Voice-Activated Power Gating

M. Price, Massachusetts Institute of Technology, Cambridge, MA

In Paper 14.4, MIT presents an IC designed in a 65nm LP process for DNN-based automatic speech recognition (ASR) and voice-activity detection (VAD). Real-time ASR capability scales from 11 words (172µW) to 145k words (7.78mW) and the noise-robust VAD has power consumption of 22.3µW.

3:45 PM
14.5 ENVISION: A 0.26-to-10TOPS/W Subword-Parallel Dynamic-Voltage-Accuracy-Frequency-Scalable Convolutional Neural Network Processor in 28nm FDSOI

B. Moons, KU Leuven, Leuven, Belgium

In Paper 14.5, KU Leuven presents an energy-scalable CNN processor in 28nm FDSOI achieving efficiencies up to 10TOPS/W by modulating computational accuracy, voltage and frequency, while maintaining recognition rate and throughput.

4:15 PM
14.6 A 0.62mW Ultra-Low-Power Convolutional-Neural-Network Face-Recognition Processor and a CIS Integrated with Always-On Haar-Like Face Detector

K. Bong, KAIST, Daejeon, Korea

In Paper 14.6, KAIST presents an ultra-low-power CNN-based face recognition (FR) processor and a CMOS image sensor integrated with an always-on Haar-like face detector in 65nm CMOS. The analog-digital hybrid Haar-like face detector improves the energy efficiency of face detection by 39% and the FR system dissipates 0.62mW at 1fps.

4:45 PM
14.7 A 288µW Programmable Deep-Learning Processor with 270KB On-Chip Weight Storage Using Non-Uniform Memory Hierarchy for Mobile Intelligence

S. Bang, University of Michigan, Ann Arbor, MI

In Paper 14.7, the University of Michigan presents a programmable fully connected (FC)-DNN accelerator in 40nm CMOS. It achieves 374GOPS/W at 0.65V (288µW) and 3.9MHz, with configurable data precision, strategic data assignment in NUMA memory (270KB) and dynamic drowsy memory operation.

5:00 PM
14.8 A 135mW Fully Integrated Data Processor for Next-Generation Sequencing

Y-C. Wu, National Taiwan University, Taipei, Taiwan

In Paper 14.8, National Taiwan University presents a data processor for Next-Generation Sequencing (NGS) in 40nm CMOS, which realizes DNA mapping, including suffix-array (SA) sorting and backward searching. With 135mW at 200MHz, it achieves significantly higher energy efficiency over CPU/GPU-based implementations.
A booming number of computer vision, speech recognition, and signal processing applications, are increasingly benefiting from the use of deep convolutional neural networks (DCNN) stemming from the seminal work of Y. LeCun et al. [1] and others that led to winning the 2012 ImageNet Large Scale Visual Recognition Challenge with AlexNet [2]. A DCNN significantly outperforming classical approaches for the first time. In order to deploy these technologies in mobile and wearable devices, hardware acceleration plays a critical role for real-time operation with very limited power consumption and with embedded memory overcoming the limitations of fully programmable solutions.

We present a state-of-the-art performance and energy efficient HW-accelerated DCNN processor with the following features: (1) An energy-efficient set of DCNN HW convolutional accelerators supporting kernel compression, (2) an on-chip reconfigurable data-transfer fabric to improve data reuse and on-chip and off-chip memory traffic, (3) a power-efficient array of DSPs to support complete real-world computer vision applications, (4) an ARM-based host subsystem with peripherals, (5) a range of high-speed IO interfaces for imaging and other types of sensors, (6) a chip-to-chip multilink to pair multiple devices together.

The test chip is a complete system demonstrator (Fig. 14.1.2) integrating an ARM Cortex microcontroller with 128KB of memory, various peripherals, eight DSP clusters (2 DPSs, 4-way 16KB instruction caches, 64KB local RAMs and a 64KB shared RAM), a reconfigurable dataflow accelerator fabric connecting high-speed camera interfaces with sensor processing pipelines, counters, color converters, feature detectors, video encoders, an 8-channel digital microphone interface, streaming DMAs and 8 convolutional accelerators. The chip includes 4 MB SRAM banks, a dedicated bus port and fine-grained power gating, to sustain the maximum throughput for convolutional stages fitting DCNN topologies, such as AlexNet without pruning, or larger ones if fewer bits are used for activations and/or weights without the need to access external memory to save power. It is possible to connect multiple chips together via four chip-to-chip high-speed serial links (similar to [5]) of up to 8Gbps to support larger networks without sacrificing throughput and/or using the chip as a co-processor.

State-of-the-art DCNNs require deep topologies with many layers, millions of parameters, and varying kernel sizes (Fig. 14.1.1), resulting in escalating bandwidth, power and area costs, presenting challenging constraints for embedded devices and applications. Efficiency can be improved with a hierarchical memory system and efficient reuse of local data. DCNN convolutional layers account for more than 90% of the total operations and accelerating these calls for the efficient balancing of the computational vs. memory resources to achieve maximum throughput without hitting associated ceilings. In our SoC, a design-time configurable accelerator framework (CAF) (Fig. 14.1.3) is used, based on unidirectional links transporting data streams via a configurable fully connected switch to/from different kinds of sources/sinks, namely, DMA units, IO interfaces (e.g., cameras), and various types of accelerators, including the Convolution Accelerator (CA) (Fig. 14.1.4). This fabric permits the definition of an arbitrary number of concurrent, virtual processing chains at runtime. A full-featured backpressure mechanism handles the dataflow, control and stream multicasting, enabling the reuse of a data stream at multiple block instances. Linked lists control the fully autonomous processing of an entire convolution layer. Multiple accelerators grouped or chained together handle various sizes of feature maps and multiple kernels in parallel. Grouping the CAs to achieve larger computational entities enables selection of an optimal tradeoff between the available data bandwidth, power consumption, and the available processing resources. Kernel sets are partitioned in batches (Fig. 14.1.3) processed sequentially and intermediate results can be stored in on-chip memory. Various kernel sizes (up to 12x12), batch sizes (up to 16), and parallel kernels (up to 4) can be handled by a single CA instance, but any size kernel can be accommodated with the accumulator input. The CA includes a line buffer to fetch up to 12 feature map data words in parallel with a single memory access. A register-based kernel buffer provides 36 read ports, while 36 16b fixed-point multiply-accumulate (MAC) units perform up to 36 MAC operations per clock cycle. An adder tree accumulates MAC results for each kernel column (Fig. 14.1.4). The overlapping, column-based calculation of the MAC operations allows an optimal reuse of the feature map data for multiple MACs, thereby reducing the power consumption associated with redundant memory accesses. The configurable batch size and variable number of parallel kernels enable optimal tradeoffs for the available input and output bandwidth, sharing across different units, and the available computing logic resources. At present, a CA’s configuration per DCNN layer is defined manually, while in future, this will be automatically generated off-line with the aid of a holistic tool starting from a DCNN description format, such as Caffe or TensorFlow. The CA supports on-the-fly kernel decompression and rounding when the kernel is quantized nonlinearly with 8 or fewer bits per weight, with top-1 error rate increasing up to 0.3% for 8 bits.

Each 32b DSP provides specific instructions (Min, Max, Sqr, Mac, Butterfly, Average), a 2-4 SIMD ALU and a dual MAC (m = a×b + c×d) with 16b operands and 40b for the accumulator to accelerate other operations, including those typically found in CNNs [2]. The dual-MAC operation loop loading operands from memory executes in a single cycle, while an independent 2D DMA channel allows the overlap of data transfers. The DSPs are tasked with max or average pooling, nonlinear activation, cross-channel response normalization and classification, representing a small fraction of the total DCNN computation but more amenable to future algorithmic evolutions. The DSPs can operate in parallel with CAs and data transfers, synchronizing by way of interrupts and mailboxes for concurrent execution. DSPs are activated incrementally as required by throughput targets, leaving ample margins to support additional tasks associated with complex applications, such as object localization and classification, multisensory (e.g. audio and video) DCNN-based data fusion and recognition, scene classification, etc.

The prototype chip is manufactured with STMicroelectronics 28nm FD-SOI technology, incorporating a mono-supply SRAM-based single-well bitcell with low-power features and adaptive circuitry to support a wide voltage range from 1.1-0.575V. Globally asynchronous and locally synchronous clocking reduces the clock network dynamic power and skew sensitivity due to on-chip variation at lower voltages and eases dynamic frequency scaling. Fine-grained power gating and multiple sleep modes for memories decrease the overall dynamic and leakage power consumption. The die size is 6.2 × 5.5 mm$^2$ and multiple sleep modes for memories decrease the overall dynamic and leakage power consumption. The die size is 6.2 × 5.5 mm$^2$ and the chip reaches 1.175GHz at 1.1V with a theoretical peak CA performance of 676GOPS (Fig. 14.1.7). It is operational at 200MHz with a 0.575V supply at 25°C and achieves an average power consumption of 41mW on AlexNet with 8 chained CAs, representing a peak efficiency of 2.9TOPS/W. Fig. 14.1.5 shows improvements over recent NN processors [3-6] of 2.06×, 3.14×, 8.37× and 9.7×. Fig. 14.1.6 illustrates AlexNet per-layer and total throughput and utilization efficiency proving the device effective for advanced real-world power-constrained embedded applications, such as intelligent IoT devices and sensors.

References:


Figure 14.1.1: DCNN complexity increase over time; AlexNet topology.

Figure 14.1.2: SoC top-level block diagram.

Figure 14.1.3: Accelerator subsystem top level; configuration of virtual processing chains; accelerator chaining.

Figure 14.1.4: Convolutional Accelerator (CA) architecture block diagram.

Figure 14.1.5: Comparisons with prior work.

<table>
<thead>
<tr>
<th>Layer</th>
<th>MOPS</th>
<th>Latency (ms)</th>
<th>Utilization</th>
<th>GOPU/W</th>
<th>Power (mW)</th>
<th>GOPU/W</th>
<th>Power (mW)</th>
<th>GOPU/W</th>
<th>Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>210.8</td>
<td>2.5</td>
<td>80%</td>
<td>1228</td>
<td>968</td>
<td>86</td>
<td>1471</td>
<td>1183</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>447.8</td>
<td>6.5</td>
<td>86%</td>
<td>1475</td>
<td>1262</td>
<td>54</td>
<td>1767</td>
<td>1512</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>299</td>
<td>3.6</td>
<td>73%</td>
<td>1987</td>
<td>1445</td>
<td>58</td>
<td>2380</td>
<td>1731</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>214.2</td>
<td>2.7</td>
<td>73%</td>
<td>1987</td>
<td>1445</td>
<td>58</td>
<td>2380</td>
<td>1731</td>
</tr>
<tr>
<td></td>
<td></td>
<td>149.6</td>
<td>1.8</td>
<td>72%</td>
<td>1987</td>
<td>1434</td>
<td>58</td>
<td>2380</td>
<td>1717</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>1331.6</td>
<td>17.1</td>
<td>77%</td>
<td>1677</td>
<td>1287</td>
<td>61</td>
<td>2009</td>
<td>1542</td>
</tr>
</tbody>
</table>

Figure 14.1.6: AlexNet performance at 0.575V, 200MHz with 8 CAs for 16 and 8b MAC operands.
Figure 14.1.7: Chip micrograph and specifications.

<table>
<thead>
<tr>
<th>Technology</th>
<th>FD-SOI 28nm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chip area</td>
<td>1023x1.2 mm</td>
</tr>
<tr>
<td>Package</td>
<td>FBGA 15x15x1.05</td>
</tr>
<tr>
<td>Clock freq</td>
<td>2.0GHz - 1.17GHz</td>
</tr>
<tr>
<td>Supply voltages</td>
<td>0.90V - 1.1V digital, 1.8V I/O</td>
</tr>
<tr>
<td>Power</td>
<td>4.1mW</td>
</tr>
<tr>
<td>On-chip RAM</td>
<td>4MB, 6x16/2 KB, 128 KB</td>
</tr>
<tr>
<td>No of DPIs</td>
<td>16</td>
</tr>
<tr>
<td>Peak DSP performance (*)</td>
<td>75 GOPS (8-bit 16-bit MAC)</td>
</tr>
<tr>
<td>No of Cpsx</td>
<td>8</td>
</tr>
<tr>
<td>Peak Cpu performance (*)</td>
<td>675 GOPS</td>
</tr>
</tbody>
</table>

(*) 1 MAC defined as 2 DSPs (AXD + AXL)
14.2 DNPU: An 8.1TOPS/W Reconfigurable CNN-RNN Processor for General-Purpose Deep Neural Networks

Dongjoo Shin, Jinmook Lee, Jinsu Lee, Hoi-Jun Yoo
KAIST, Daejeon, Korea

Recently, deep learning with convolutional neural networks (CNNs) and recurrent neural networks (RNNs) has become universal in all-around applications. CNNs are used to support vision recognition and processing, and RNNs are able to recognize time varying entities and to support generative models. Also, combining both CNNs and RNNs can recognize time varying visual entities, such as action and gesture, and to support image captioning [1]. However, the computational requirements in CNNs are quite different from those of RNNs. Fig. 14.2.1 shows a computation and weight-size analysis of convolution layers (CLs), fully-connected layers (FCLs) and RNN-LSTM layers (RLs). While CLs require a massive amount of computation with a relatively small number of filter weights, FCLs and RLs require a relatively small amount of computation with a huge number of filter weights. Therefore, when FCLs and RLs are accelerated with SoCs specialized for CLs, they suffer from high memory transaction costs, low PE utilization, and a mismatch of the computational patterns. Conversely, when CLs are accelerated with FCL- and RL-dedicated SoCs, they cannot exploit reusability and achieve required throughput. So far, works have considered acceleration of CLs, such as [2-4], or FCLs and RLs like [5]. However, there has been no work on a combined CNN-RNN processor. In addition, a highly reconfigurable CNN-RNN processor with high energy-efficiency is desirable to support general-purpose deep neural networks (DNNs).

In this paper, we present an 8.1TOPS/W reconfigurable CNN-RNN processor with the following 3 key features: 1) A reconfigurable heterogeneous architecture with a CL processor (CP) and a FC-RL processor (FRP) to support general-purpose DNNs, 2) a LUT-based reconfigurable multiplier optimized for the dynamic fixed point with on-line (on-chip) adaptation via overflow monitoring to exploit maximum efficiency from kernel reuse in the CP, 3) a quantization table (Q-table)-based matrix multiplication to reduce off-chip memory access and remove duplicated multiplications in the FRP.

The overall architecture of the proposed deep neural processing unit (DNPU) is shown in Fig. 14.2.2. It consists of the CP, FRP, and a top RISC controller. The CP is composed of 4 convolution clusters and 1 aggregation core. Each convolution cluster performs convolution operations with 4 convolution cores, and transfers the accumulation results to the accumulation core. One convolution core contains 4 PE groups with 48 PEs. The FRP performs matrix multiplication with the 128-entry Q-table, and 8 16b fixed-point multipliers are used to update the Q-table. The CP and FRP are able to process 4 different CLs and 8 RLs, respectively, in parallel.

The proposed input-layer division method (mixed division) is shown in Fig. 14.2.3. Due to limitations in on-chip memory size, the input layer image must be divided into several parts. There are three possible division methods: image division (ID), channel division (CD) and mixed division (MD). In the case of ID, the same weight parameters must be loaded multiple times for each divided image. For the CD, final output elements cannot be calculated with a single divided image, therefore multiple off-chip accesses are required. These two methods combine in the MD, which uses both ID and CD. In the MD, image division # and channel division # are selected to minimize the off-chip accesses. For various channel divisions, the proposed CP supports various channel sizes (#16 to 1024) and image sizes (32×16 to 256×128) with 5 different accumulation hierarchies (cluster, core, memory, bank and in-bank division). Supported configurations can be processed without any degradation of PE utilization.

The detailed architecture of the Q-table-based FRP is shown in Fig. 14.2.5. The FRP has a reconfigurable 127-entry Q-table. It can function as one 7b Q-table to eight 4b Q-tables. Each entry of the Q-table contains the pre-computed multiplication result between a 16b fixed-point input and a 16b fixed point weight. After the Q-table is constructed once, only quantized indexes are required to compute the product. With the Q-tables, off-chip accesses can be reduced by 75%, and 99% of the 16b fixed-point multiplications can be avoided.

Measurement results are shown in Fig. 14.2.6. The DNPU can operate from 0.765-1.1V supply with 50-200MHz clock frequency. The power consumption at 0.765V and 1.1V are 34.6mW and 279mW, respectively. For a particular frame rate, energy-efficiency and bit-width (accuracy) can be traded off with another one. For the CP, word length can be changed from 4b to 16b, and quantization bit-width can be configured from 4b to 7b. With the proposed layer-by-layer on-line adaptation via overflow monitoring, the fraction length for each layer is adjusted rapidly without any off-chip learning for dynamic fixed-point. The softmax score of the answer object increases with the fraction length adaptation. If the scene is changed to another object, the adaptation flow is invoked for a new fraction length. As shown in the graph, 32b floating point precision is achieved with only 4b word length.

The DNPU shown in Fig. 14.2.7 is fabricated using 65nm CMOS technology and occupies 16mm² die area. The proposed DNPU is the first CNN-RNN SoC with the highest energy efficiency (8.1TOPS/W). The table shows a performance comparison with the 3 previous deep learning SoCs. This work is the only one that supports both CNNs and RNNs. Compared to [2] and [3], this work shows 20x and 4.5x higher energy efficiency, respectively. Also, DNPU shows 8.5x higher energy efficiency compared to [5].

References:
Figure 14.2.1: Heterogeneous characteristics of deep neural networks.

Figure 14.2.2: Overall architecture of the proposed CNN-RNN processor (DNPU).

Figure 14.2.3: Mixed division methods and reconfigurability.

Figure 14.2.4: Layer-by-layer dynamic fixed-point with on-line adaptation and optimized LUT-based multiplier.

Figure 14.2.5: Quantization table-based matrix multiplication.

Figure 14.2.6: Measurement results.
Figure 14.2.7: Chip photograph and performance summary.

<table>
<thead>
<tr>
<th>Technology</th>
<th>65nm FDSM CMOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Die Area</td>
<td>400um x 400um</td>
</tr>
<tr>
<td>SRAM Size</td>
<td>381K x 160K</td>
</tr>
<tr>
<td>Supply Voltage</td>
<td>2.7V - 1.3V</td>
</tr>
<tr>
<td>Operating Frequency</td>
<td>200MHz - 300MHz</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>2W - 3W</td>
</tr>
<tr>
<td>Energy Efficiency</td>
<td>2.1 TOPS/W - 3.5 TOPS/W</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>CLP</th>
<th>FC/LP</th>
<th>ETO</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>MANO</td>
<td>5%</td>
<td>5%</td>
<td>0%</td>
<td>10%</td>
</tr>
<tr>
<td>CLP Power (µW)</td>
<td>39.5</td>
<td>47.0</td>
<td>47.0</td>
<td>47.0</td>
</tr>
<tr>
<td>FC/LP Power (µW)</td>
<td>38.5</td>
<td>46.0</td>
<td>46.0</td>
<td>46.0</td>
</tr>
<tr>
<td>ETO Power (µW)</td>
<td>38.0</td>
<td>46.0</td>
<td>46.0</td>
<td>46.0</td>
</tr>
<tr>
<td>Total Power (µW)</td>
<td>125</td>
<td>99</td>
<td>99</td>
<td>99</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>CLP</th>
<th>FC/LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLP Time (µs)</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>FC/LP Time (µs)</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>ETO Time (µs)</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Total Time (µs)</td>
<td>45</td>
<td>45</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>CLP</th>
<th>FC/LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLP Energy (µJ)</td>
<td>39</td>
<td>47</td>
</tr>
<tr>
<td>FC/LP Energy (µJ)</td>
<td>38</td>
<td>46</td>
</tr>
<tr>
<td>ETO Energy (µJ)</td>
<td>38</td>
<td>46</td>
</tr>
<tr>
<td>Total Energy (µJ)</td>
<td>125</td>
<td>99</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>CLP</th>
<th>FC/LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLP Throughput (TOPS)</td>
<td>2.1</td>
<td>2.1</td>
</tr>
<tr>
<td>FC/LP Throughput (TOPS)</td>
<td>2.1</td>
<td>2.1</td>
</tr>
<tr>
<td>ETO Throughput (TOPS)</td>
<td>2.1</td>
<td>2.1</td>
</tr>
<tr>
<td>Total Throughput (TOPS)</td>
<td>2.1</td>
<td>2.1</td>
</tr>
</tbody>
</table>
This paper presents a 28nm SoC with a programmable FC-DNN accelerator design that demonstrates: (1) HW support to exploit data sparsity by eliding unnecessary computations (4× energy reduction); (2) improved algorithmic error tolerance using sign-magnitude number format for weights and datapath computation; (3) improved circuit-level timing violation tolerance in datapath logic via time-borrowing; (4) combined circuit and algorithmic resilience with Razor timing violation detection to reduce energy via $V_{dd_{min}}$ scaling or increase throughput via $F_{clk}$ scaling; and (5) high classification accuracy (98.36% for MNIST test set) while tolerating aggregate timing violation rates $>10^{-1}$. The accelerator achieves a minimum energy of 0.36µJ/pred at 667MHz, maximum throughput at 1.2GHz and 0.57µJ/pred, or a 10%-margined operating point at 1GHz and 0.58µJ/pred.

The SoC (Fig. 14.3.1) is based around an ARM Cortex-M0 cluster. The DNN engine connects through an asynchronous bridge, allowing independent $F_{clk}$ and $V_{dd}$ scaling to balance throughput and energy efficiency. A 4-way banked on-chip memory (W-MEM) stores the weights for the DNN model (up to 1MB) and provides low-latency access to the DNN engine.

The DNN engine (Fig. 14.3.2) is a 5-stage SIMD-style programmable sparse matrix-vector ($MxV$) machine for processing arbitrary DNNs. A sequencer dynamically schedules operations for different DNN configurations—up to eight FC layers with 1-to-1024 nodes per layer. The host CPU loads the input vector into the IPBUF scratchpad and the 8-way MAC datapath processes eight concurrent neuron computations at a time, fed from either IPBUF (input layers) or XBUF (hidden layers). Once the in-flight neurons have accumulated all the weight-activation products, the activation stage adds a bias term and applies a rectified linear unit ($ReLU$) activation function. The resulting neuron activations are written back to XBUF, which is double buffered to allow simultaneous reads from the previous layer and writes to the current layer. The MAC unit uses optimized 16b fixed-point precision throughout, with support for programmable rounding modes, two’s compliment (TC) or sign-magnitude (SM) numbers, and 8b or 16b weight precision.

We exploit the abundant dynamic sparsity in the input data and activations. Prior work clock-gates functional units to save power for zero operands, but still consume cycles for pipeline bubbles [3,5]. Instead, we eliminate bubbles by dynamically eliding all zero operands at XBUF writeback. Moreover, we also skip even small non-zero values, without degrading prediction accuracy, leading to further savings. After $ReLU$, the activation stage compares output activations against per-layer programmable thresholds to produce a SKIP control signal that predicates writeback of data to XBUF. A small 512B SRAM (NBUF) keeps the list of active node indexes in the previous layer, from which W-MEM addresses are generated. For MNIST, the average number of loads, ops, and cycles are reduced by over 75%, significantly improving energy and throughput (Fig. 14.3.5).

To enhance error-tolerant operation, the DNN accelerator augments two timing-critical stages, W-MEM load and MAC unit, with Razor flip-flops (RZFFs) on timing end-points. The MUX cell in the dual-mode RZFF (Fig. 14.3.2) supports operation as a datapath FF or latch with time borrowing. A global pulse clock (90-to-300ps pulse width) defines both the timing detection window and latch transparency time, while satisfying hold delays set at design time. All other paths include 30% margin.

Figure 14.3.3 plots measured power and timing violation rate (measured at word granularity) results running MNIST inference on a $784 \times 256 \times 256 \times 256 \times 10$ DNN for TC and SM number formats with 16b weights. The design targeted a softoff $F_{max}$ of 667MHz (1.5ns period) under worst-case conditions (SS, 0.81V, 125°C). At 667MHz, $V_{dd}$ can scale from nominal (0.9V) down to 0.77V before on-chip counters record the first timing violation, translating voltage margin to 30% power reduction (Fig. 14.3.3 top). Alternatively, at 0.9V, $F_{clk}$ can scale beyond 1GHz before the first timing violations occur.

Further improvements are possible by leveraging inherent resilience of the DNN. Fig. 14.3.4 plots measured classification accuracy vs. timing violation rates for W-MEM loads, datapath MACs, and the combination. For the memory, SM numbering exploits the zero-mean Gaussian distribution of the weights matrix to reduce switching activity in the MSBs and thus bit-flips. Adding a bit-masking (BM) technique to mask individual bit errors in the weight word allows the accelerator to tolerate SRAM read timing violation rates $>10^{-1}$ at 98.36% accuracy. Error tolerance in the datapath is harder to achieve because bit-flips persist in the accumulator. Although SM offers some benefit, circuit-level time borrowing is much more effective, tolerating timing violation rates commensurate to levels seen for the memory. Generous time borrowing from the accumulator is possible through the feedback path of the adder in the MAC unit (Fig. 14.3.2). Together, timing violation tolerance improves by several orders of magnitude at 98.36% accuracy, over the whole 10k vector MNIST test set, which supports further $V_{dd}$ reduction to 0.715V (no margin).

Figure 14.3.5 summarizes energy and throughput improvements offered by different optimizations and techniques. Overall, energy reduces by $\sim$9x down to 0.36µJ/pred via 8b weights, aggressive $V_{dd}$ scaling (no margin), and exploiting sparsity (skip). Fig. 14.3.6 shows the Pareto front of prediction accuracy vs. energy emerging from running different network topologies on the test chip with comparisons to all other measured HW that reports MNIST energy and accuracy results. Fig. 14.3.7 provides chip details and die microphotograph.

Acknowledgements:
This work is supported in part by DARPA under Contract No. HR0011-13-C-0022. We are grateful to ARM for providing physical IP.

References:
Figure 14.3.1: System block diagram of 28nm SoC with DNN engine.

Figure 14.3.2: Simplified microarchitecture of the five-stage DNN accelerator, split sign-magnitude (SM) accumulator design and RZFF.

Figure 14.3.3: Power and timing error rate results for voltage scaling at sign-off FMAX of 667MHz (top), and frequency scaling at 0.9V (bottom).

Figure 14.3.4: MNIST accuracy vs. timing error rates in the W-MEM SRAM, MAC datapath, and combined.

Figure 14.3.5: Energy/pred and throughput across different configurations at 98.36% accuracy for the MNIST test set.

Figure 14.3.6: MNIST accuracy vs. energy/pred for multiple topologies on the test chip and other reported hardware measurements.
Figure 14.3.7: Chip summary and annotated microphotograph.
14.4 A Scalable Speech Recognizer with Deep-Neural-Network Acoustic Models and Voice-Activated Power Gating

Michael Price1, James Glass2, Anantha P. Chandrakasan1

1Massachusetts Institute of Technology, Cambridge, MA
2Analog Devices, Cambridge, MA

The applications of speech interfaces, commonly used for search and personal assistants, are diversifying to include wearables, appliances, and robots. Hardware-accelerated automatic speech recognition (ASR) is needed for scenarios that are constrained by power, system complexity, or latency. Furthermore, a wakeup mechanism, such as voice activity detection (VAD), is needed to power gate the ASR and downstream system. This paper describes IC designs for ASR and VAD that improve on the accuracy, programmability, and scalability of previous work.

Figure 14.4.1 shows the components of our embedded speech recognizer. To support low-duty-cycle operation, the external memory should be non-volatile; today’s consumer devices use MLC flash, with higher energy-per-bit than DRAM (typically 100pJ/b). This memory can easily consume more power than the ASR core itself. As shown by [1], modern ASR modeling and search techniques can be modified to reduce memory bandwidth without compromising accuracy. These techniques include deep neural networks (DNNs), weighted finite-state transducers (WFSTs), and Viterbi search over hidden Markov models (HMMs).

Most recent work on neural network hardware (e.g. [2]) targets convolutional networks (CNNs) for computer vision applications. IC implementations of keyword detection with DNNs have achieved power consumption as low as 3.3 mW [3], but DNNs have not yet been incorporated into a standalone hardware speech recognizer. We tailor DNNs for low-power ASR by using limited network widths and quantized, sparse weight matrices.

Our feed-forward DNN accelerator uses a SIMD architecture shown in Fig. 14.4.2. A compressed model is streamed in, and the decoded parameters are broadcast to 32 parallel execution units (EUs) that each process one feature vector. This reduces memory bandwidth by up to 32x at some expense in latency. Each EU has enough local memory to handle NN layers with up to 1k hidden nodes. As shown in Fig. 14.4.3, the EUs are organized in eight “chunks”, which can be reconfigured to handle networks with up to 4k hidden nodes while disabling some of the EUs. Sparse weight matrices are supported by storing a run-length encoding of the nonzero coefficient locations at the beginning of each row; for an acoustic model with 31% nonzero weights, this allows a 54% reduction in memory bandwidth. Quantization tables stored in SRAM allow the weights to be stored with 4-12 bits each. The EU supports sigmoid or rectified linear activation functions with a long temporal context and classifies them with a stripped-down version of the DNN evaluator described above; this proved to be the most robust algorithm in challenging noise conditions. The VAD supports downsampling (so ASR and VAD can use different sample rates) with a 65-tap FIR antialiasing filter, and buffers input samples for later retrieval by the ASR module (which is powered down until speech is detected).

Our ASR/VAD test chip is shown in Fig. 14.4.7. This chip performs all stages of ASR transcription from audio samples to text. ASR is functional with logic supply voltages from 0.60V (10.2MHz) to 1.20V (86.8MHz), and VAD is functional from 0.50V (1.68MHz) to 0.90V (47.8MHz). The core is partitioned into five voltage areas; SRAM can be operated at 0.15-0.20V above the logic supply (up to a limit of 0.50V). On the WSJ eval92-5k task that was demonstrated by [5], we obtained 4.1× fewer word errors (3.12% vs. 13.0%), 3.3× lower core power (1.78W vs. 6.0mW), and 12.7× lower memory bandwidth (4.84MB/s vs. 61.6MB/s). Our framework is designed to interoperate with the open-source Kaldi tools [6], allowing software recognizers trained in Kaldi to quickly be ported to the hardware platform. We hope that these contributions will facilitate the deployment of high-quality speech interfaces in low-power devices.

A summary of ASR and VAD performance results is shown in Fig. 14.4.6. A variety of ASR tasks, with vocabularies ranging from 11 words to 145k words, can be run in real-time on this chip. Core power scales by 45x from the easiest to the hardest task, and memory bandwidth scales by 136x. On the WSJ eval92-5k task that was demonstrated by [5], we obtained 4.1× fewer word errors (3.12% vs. 13.0%), 3.3× lower core power (1.78W vs. 6.0mW), and 12.7× lower memory bandwidth (4.84MB/s vs. 61.6MB/s). Our framework is designed to interoperate with the open-source Kaldi tools [6], allowing software recognizers trained in Kaldi to quickly be ported to the hardware platform. We hope that these contributions will facilitate the deployment of high-quality speech interfaces in low-power devices.

Acknowledgements:
This work was funded by Quanta Computer via the Qmulus Project. The authors would like to thank the TSMC University Shuttle Program for providing chip fabrication.

References:
Figure 14.1.1: Power gated speech recognizer concept (gray region is power gated by VAD decision).

Figure 14.2.2: Block diagram of SIMD neural network evaluator.

Figure 14.3.3: NN evaluator execution units are grouped into chunks that can be reconfigured for best utilization across different network sizes.

Figure 14.4.4: VAD block diagram and system power model.

Figure 14.5.5: Explicit clock gating hierarchy: VAD clock domain (top), ASR clock domain (bottom), and measured impact on real-time ASR power versus clock frequency.

Figure 14.6.6: Summary of ASR/VAD test chip results, with highlighted results illustrated.
Figure 14.4.7: Die photo and table of key specifications.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>65 nm LP</td>
</tr>
<tr>
<td>Core size</td>
<td>3.1 × 3.1 mm</td>
</tr>
<tr>
<td>Die size</td>
<td>3.63 × 3.63 mm</td>
</tr>
<tr>
<td>Package</td>
<td>88-pin QFN</td>
</tr>
<tr>
<td>Logic gates</td>
<td>2016K (KAND2 equiv.)</td>
</tr>
<tr>
<td>SRAM</td>
<td>5.83 MB</td>
</tr>
<tr>
<td>Supply voltage</td>
<td>0.60–1.20 V</td>
</tr>
<tr>
<td>Power consumption</td>
<td>1.8–7.8 mW (typ.)</td>
</tr>
<tr>
<td>Clock frequency</td>
<td>3–65 MHz</td>
</tr>
<tr>
<td>Neural network efficiency</td>
<td>16–56 pJ/neuron</td>
</tr>
<tr>
<td>Viterbi search efficiency</td>
<td>2.5–6.3 nJ/hypothesis</td>
</tr>
</tbody>
</table>
14.5 ENVISION: A 0.26-to-1TOPS/W Subword-Parallel Dynamic-Voltage-Accuracy-Frequency-Scalable Convolutional Neural Network Processor in 28nm FDSOI

Bert Moons, Roel Uytterhoeven, Wim Dehaene, Marian Verhelst
KU Leuven, Leuven, Belgium

ConvNets, or Convolutional Neural Networks (CNN), are state-of-the-art classification algorithms, achieving near-human performance in visual recognition [1]. New trends such as augmented reality demand always-on visual processing in wearable devices. Yet, advanced ConvNets achieving high recognition rates are too expensive in terms of energy as they require substantial data movement and billions of convolution computations. Today, state-of-the-art mobile GPUs and ConvNet accelerator ASICs [2][3] only demonstrate energy-efficiencies of 10's to several 100's GOPS/W, which is one order of magnitude below requirements for always-on applications. This paper introduces the concept of hierarchical recognition processing, combined with the Envision platform: an energy-scalable ConvNet processor achieving efficiencies up to 1TOPS/W, while maintaining recognition rate and throughput. Envision hereby enables always-on visual recognition in wearable devices.

Figure 14.5.1 demonstrates the concept of hierarchical recognition. Here, a hierarchy of increasingly complex individually trained ConvNets, with different topologies, different network sizes and increasing computational precision requirements, is used in the context of person identification. This enables constant scanning for faces at very low average energy cost, yet rapidly scales up to more complex networks detecting a specific face such as a device’s owner, all the way up to full VGG-16-based 5760-face recognition. The opportunities afforded by such a hierarchical approach span far beyond face recognition alone, but can only be exploited by digital systems demonstrating wide-range energy scalability across computational precision. State-of-the-art ASICs in references [3] and [4] only show 1.5× and 8.2× energy-efficiency scalability, respectively. Envision improves upon this by introducing subword-parallel Dynamic-Voltage-Accuracy-Frequency Scaling (DVAFS), a circuit-level technique enabling 40× energy-precision scalability at constant throughput. Figure 14.5.2 illustrates the basic principle of DVAFS and compares it to Dynamic-Accuracy Scaling (DAS) and Dynamic-Voltage-Accuracy Scaling (DVAS) [4]. In DAS, switching activity and hence energy consumption is reduced for low precision computations by rounding and masking a configurable number of LSB’s at the inputs of multiply-accumulate (MAC) units. DVAS exploits shorter critical paths in DAS’s reduced-precision modes by combining it with voltage scaling for increased energy scalability. This paper proposes subword-parallel DVAFS, which further improves upon DVAS, by reusing inactive arithmetic cells at reduced precision. These can be reconfigured to compute 2×1-16b or 4×1-4b (N=1-16b/kW, k is level of subword-parallelism), rather than 1×1-16b words per cycle, when operating at less than 8b precision. At constant data throughput, this permits lowering the processor’s frequency and voltage significantly below DVAS values. As a result, DVAFS is a dynamic precision technique which simultaneously lowers all run-time adaptable parameters influencing power consumption: activity α, frequency f and voltage V. Moreover, in contrast to DAS and DVAS, which only save energy in precision-scaled arithmetic blocks, DVAFS allows lowering f and V of the full system, including control units and memory, hereby shrinking non-compute energy overheads drastically at low precision.

Energy-efficiency is further improved by modulating the body bias (BB) in an FDSOI technology. This permits tuning of the dynamic vs. leakage power balance while considering the computational precision. At high precision, reducing Vt allows a scaling down of the supply voltage to reduce dynamic consumption while maintaining speed, at a limited leakage energy cost and an overall efficiency increase. At low precision, and reduced switching activity, Vt and the supply voltage are increased to lower the leakage overhead at constant speed. This increases dynamic energy, but reduces the overall energy consumption.

Figure 14.5.3 shows the top-level architecture of Envision. This chip is a multi-power and multi-body-bias domain, sparsity-guarded ConvNet processor exploiting DVAFS. It is fully C-programmable, allowing deployment of a wide range of ConvNet topologies, and has a 16b SIMD RISC instruction set extended with custom instructions, similar to [4]. The processor is equipped with 20× (for convolutions) and 10×-SIMD arrays (for ReLU, max-pooling) and a scalar unit. An on-chip memory (DM) consists of 64×2kB single-port SRAM macros, subdivided into 4 blocks of 16 parallel banks, storing a maximum of 66536×N words. 3 blocks can be read or written in parallel: 2 blocks by the processor, another by the Huffman DMA, used for compressing IO bandwidth up to 5.8×. The system is divided into three power- and body-bias domains to enable granular dynamic voltage scaling.

Figure 14.5.4 shows how the 6-stage pipelined processor executes convolutions in its 16×16 2D-SIMD MAC array. Each MAC is a single cycle N-subword-parallel multiplier, followed by a N=48b/V reconfigurable accumulation adder and register. As such, the 16×16 array can generate N×256 intermediate outputs per cycle while consuming only N×16 filter weights and N×16 features in a first convolution cycle. In subsequent cycles, a 256b FIFO further reduces memory bandwidth by reusing and shifting features along the x-axis, requiring only a single new feature fetch per cycle. As all intermediate output values are stored in accumulation registers, there is no data-transfer between MACs and no frequent write-back to SRAM. Sparsity is exploited by guarding both memory fetches and MAC operations [4], using flags stored in a GMR memory. This leads to an additional 1.6× system-wide gain in energy consumption compared to DVAFS alone for typical ConvNets (30-60% zeroes).

Envision was implemented in a 28nm FDSOI technology on 1.87mm2 and runs at 200MHz at 1V and room temperature. Fig. 14.5.5 shows measurement results highlighting its wide-range precision-energy scalability, with nominal and optimal body-biasing. All modes run the same 5× ConvNet-layer, with a typical MAC efficiency of 73%, or 0.73×kN×256×2 effective operations per second. When scaling down from 16× to 4× sparse computations at 76GOPS, power goes from 290mW down to 7.6mW, as supply voltage and body bias are modulated between 0.65-1.1V and ±0.2-1.2V. Measurements for the convolutional layers in hierarchical face recognition are listed in Fig. 14.5.1, demonstrating 6.2μJ/f at average 6.5mW instead of 23100μJ/f at 77mW. This illustrates the feasibility of always-on recognition through hierarchical processing on energy-scalable Envision.

Figure 14.5.6 shows a comparison with recent ConvNet ASICs. Envision scales efficiency on the AlexNet convolutional layers between 0.8-3.8TOPS/W, compared to 0.16TOPS/W [3] and 0.56-1.4TOPS/W [4]. Efficiency is 2TOPS/W average for VGG-16 and up to 10 TOPS/W peak. This further illustrates Envision’s ability to minimize energy-consumption for any ConvNet, demonstrating an energy-scalability of up to 40× at nominal throughput in function of precision and sparsity, hereby enabling always-on hierarchical recognition.

Figure 14.5.7 shows a die photo of Envision, illustrating the physical placement of its 3 power domains in a 1.29×1.45mm2 active area.

Acknowledgements:
This work is partially funded by FWO and Intel Corporation. We thank Synopsys for providing their ASIP Designer tool suite and STMicroelectronics for silicon donation. Special thanks to CEA-LETI and CMP for back-end support.

References:
### Figure 14.5.1: Hierarchical face recognition.

<table>
<thead>
<tr>
<th>Complexity</th>
<th>6M MACs</th>
<th>12M MACs</th>
<th>30M MACs</th>
<th>50M MACs</th>
<th>15.4G MACs</th>
</tr>
</thead>
<tbody>
<tr>
<td>sparsity</td>
<td>544%</td>
<td>845%</td>
<td>10-33%</td>
<td>8-47%</td>
<td>5-82%</td>
</tr>
<tr>
<td>accuracy</td>
<td>2:6b (N=4)</td>
<td>3:8b (N=4)</td>
<td>4:6b (N=2)</td>
<td>4:6b (N=2)</td>
<td>4:6b (N=2)</td>
</tr>
<tr>
<td>conv. mode size</td>
<td>32x32 on-chip</td>
<td>42x42 on-chip</td>
<td>12x12</td>
<td>74x74</td>
<td>15x15</td>
</tr>
<tr>
<td>recognition rate</td>
<td>94.3%</td>
<td>95.9%</td>
<td>95.4%</td>
<td>94%</td>
<td>90% - 95%</td>
</tr>
<tr>
<td>chip efficiency</td>
<td>4.2 TOPS/W</td>
<td>4 TOPS/W</td>
<td>1.8 TOPS/W</td>
<td>1.8 TOPS/W</td>
<td>1.3 TOPS/W</td>
</tr>
<tr>
<td>freq</td>
<td>25 MHz</td>
<td>25 MHz</td>
<td>50 MHz</td>
<td>14.4 MHz</td>
<td>200 MHz</td>
</tr>
<tr>
<td>avg. power</td>
<td>6.4 mW</td>
<td>6.6 mW</td>
<td>15 mW</td>
<td>14.4 mW</td>
<td>17 mW</td>
</tr>
</tbody>
</table>

**Figure 14.5.2: DVAFS and body bias tuning.**

- **Features -** $F_{B}^{16b}$, $F_{B}^{16b}$, $F_{B}^{32b}$
- **Weights -** $W_{BM}$
- **N-subword parallel MAC (N=4 ex.)**
  - N x 16b | N = 4 | N x 4b | N = 4 | N x 8b | N = 4 | N x 16b | N = 4
  - Reconfig. Booth-Wallace Multiplier
  - Prog. Status Reg.
  - Accumulator

**Figure 14.5.3: Top-level architecture of Envision.**

- **DNA data**
- **Programs**
- **Subword-parallel, 2D-SIMD MAC array**
- **Input Precision & Guard Control**

**Figure 14.5.4: Parallel, rounded and guarded data flow.**

- **3x3 CONV FIFO flow**
- **3x3 CONV FIFO flow**
- **N-subword parallel MAC (N=4 ex.)**
  - N x 16b | N = 4 | N x 4b | N = 4 | N x 8b | N = 4 | N x 16b | N = 4
  - Reconfig. Booth-Wallace Multiplier
  - Prog. Status Reg.
  - Accumulator

**Figure 14.5.5: Measured efficiency up to 10 TOPS/W.**

- **Nominal body biasing**
  - $BB = +0.6V$ symmetrical
  - $BB = 0V$ to $+1.2V$ symmetrical
- **Optimized body biasing**
  - $BB = +0.2V$ symmetrical
  - $BB = +0.6V$ symmetrical

### Figure 14.5.6: Embedded ConvNet comparison.

- **Core Voltage $V_{CC}$ [V]**
  - 0.53 TOPS/W
  - 0.33 TOPS/W
- **Core Voltage $V_{CC}$ [V]**
  - 0.53 TOPS/W
  - 0.33 TOPS/W

**Table:**

<table>
<thead>
<tr>
<th>Technology</th>
<th>GLSVLSI ’15</th>
<th>ISSCC ’16</th>
<th>VLSI ’16</th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal Frequency (MHz)</td>
<td>500</td>
<td>200</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>Supply @ $I_{ss}$ (V)</td>
<td>Fixed 196</td>
<td>Fixed 67</td>
<td>Fixed 102</td>
<td>Dynamic N x 102</td>
</tr>
<tr>
<td>Peak performance (GOPS)</td>
<td>1.31</td>
<td>12.25</td>
<td>2.4</td>
<td>1.87</td>
</tr>
<tr>
<td>Active Area [mm²]</td>
<td>0.96m²</td>
<td>1.85m²</td>
<td>1.85m²</td>
<td>Dynamic N x 256</td>
</tr>
<tr>
<td>Gate Count (NANO-2)</td>
<td>43</td>
<td>184.5</td>
<td>144</td>
<td>144</td>
</tr>
<tr>
<td>On-Chip SRAM [kB]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$N_x$, $N_y$, $N_z$</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>$F_{B}$, $F_{W}$</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>Precision [bits]</td>
<td>Fixed 12</td>
<td>Fixed 16</td>
<td>Dynamic 1-16</td>
<td>Dynamic N x 1-16</td>
</tr>
<tr>
<td>AlexNet Conv-layers (mM)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>510 @ 1x</td>
</tr>
<tr>
<td>VGG Conv-layers (mM)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>145 GOPS</td>
</tr>
<tr>
<td>Dynamic power range @ GOPS/min (mW)</td>
<td>0.44</td>
<td>0.58</td>
<td>55-56</td>
<td>7.5-100</td>
</tr>
<tr>
<td>Min. efficiency (TOPS/W)</td>
<td>0.15</td>
<td>0.27</td>
<td>2.6</td>
<td>0.26</td>
</tr>
<tr>
<td>Max. efficiency (TOPS/W)</td>
<td>0.35</td>
<td>0.8</td>
<td>2.6</td>
<td>10</td>
</tr>
</tbody>
</table>
Figure 14.5.7: Die micrograph of Envision.
14.6 A 0.62mW Ultra-Low-Power Convolutional-Neural-Network Face-Recognition Processor and a CIS Integrated with Always-On Haar-Like Face Detector

Kyeongryeo Bong, Sungpill Choi, Changhyeon Kim, Sanghoon Kang, Youchang Kim, Hoi-Jun Yoo
KAIST, Daejeon, Korea

Recently, face recognition (FR) based on always-on CIS has been investigated for the next-generation UI/UX of wearable devices. A FR system, shown in Fig. 14.6.1, was developed as a life-cycle analyzer or a personal black box, constantly recording the people we meet, along with time and place information. In addition, FR with always-on capability can be used for user authentication for secure access to his or her smartphone and other personal systems. Since wearable devices have a limited battery capacity for a small form factor, extremely low power consumption is required, while maintaining high recognition accuracy. Previously, a 23mW FR accelerator [1] was proposed, but its accuracy was low due to its hand-crafted feature-based algorithm. Deep learning using a convolutional neural network (CNN) is essential to achieve high accuracy and to enhance device intelligence. However, previous CNN processors (CNNP) [2-3] consume too much power, resulting in <10 hours operation time with a 190mAh coin battery.

We introduce an ultra-low-power CNN FR processor and a CIS integrated with an always-on Haar-like face detector [4] for smart wearable devices. For ultra-low power consumption, it adopts 3 key features: 1) an analog-digital hybrid Haar-like face detector (HHFD) integrated on the CIS for low-power face detection (FD), 2) an ultra-low-power CNNP with wide I/O local distributed memory (DM-CNNP), and 3) a separable filter approximation for convolutional layers (SF-CONV) and a transpose-read SRAM (T-SRAM) for low-power CNN processing.

Figure 14.6.2 depicts the overall architecture of the proposed FR system. The system consists of two chips: a face image sensor (FIS) and the CNNP. Firstly, the FIS performs always-on imaging and Haar-like FD. Once a face is detected, FIS transmits only the face image to the CNNP, and then the CNNP completes FR. The system consists of two chips: a face image sensor (FIS) and the CNNP. Firstly, the FIS performs always-on imaging and Haar-like FD. Once a face is detected, FIS transmits only the face image to the CNNP, and then the CNNP completes FR.

Figure 14.6.3 shows the hybrid integration of the AHFC and DHFU for low-power FD. FD is composed of several cascaded classifying stages, and in each stage, Haar-like filters compare the intensity summation of a black pixel region with a reference. The AHFC performs block summation of intensity voltage levels directly from the CIS by utilizing a 20μA analog memory. The analog memory cell consists of a sampling capacitor C_AHFC, an input capacitor C, a unity gain buffer, and several switches. As shown in the timing diagram, after the reset phase, the analog memory cells in black or white regions transfer charge proportional to (V_blk-V_ref) to C_blank or C_tree, respectively, and then the AHFC outputs ‘pass’ or ‘fail’ as the result of the comparison. The proposed HHFD, the AHFC is used only for the first stage to minimize the static current consumption. Consequently, the DHFU avoids more than 60% of the initial processing, integral image generation, by receiving the results from the AHFC to save power consumption, and it performs the block summation efficiently with 3 add/sub operations.

The architecture of DM-CNNP is shown in Fig. 14.6.4. Each PE fetches 32 words/cycle from local T-SRAM to support 4 convolution units, where each unit has a 64 MAC array. Therefore, the CNNP with 4×4 PEs fetches 512 words/cycle from the wide I/O local distributed memory and executes 1,024 MAC operations/cycle simultaneously. Such wide memory bandwidth and massively parallel MAC operations enable high throughput operation with low clock frequency, 5MHz, at near threshold voltage (NTV), 0.46mV. When a convolution operation is performed, the MAC input registers shift the words by one column at each cycle to accumulate the partial sums in the accumulation registers. The PEs connected to the same row can transfer data to other PEs to reduce the overhead in processing cycles due to inter-PE data communication. Also, the MAC units can be clock-gated with mask bits to reduce the unnecessary power consumption.

Figure 14.6.5 shows the schematic diagram of the SF-CONV and the T-SRAM. The separable filter approximation can replace the convolution of a d×d filter with two convolution stages of a d×1 vertical filter and a 1×d horizontal filter with <1% error [5]. The number of MAC operations and processing cycles of SF-CONV are reduced by 74.7% and 77.1%, respectively, in a given test case. Since SF-CONV cannot read the column feature data connected to the same bitline, vertical filtering must fetch the column data through multiple SRAM accesses. In this work, T-SRAM, having two read modes: row-access read and column-access read, is utilized to read the column data at once. In a T-SRAM cell, a decoupling MOS for read is added to a conventional 6T cell, and its source and drain nodes are connected to the word line, H_RDWL and the bitline, H_RDBL respectively, during the row-access data read. For the column-access data read, the two lines interchange their roles: H_RDWL works as V_RDBL, and H_RDBL works as V_RDWL. Both row and column paths have separate word line drivers and sense amplifiers, and for the 16b words/cycle access through column path, each bit of a 16b word is placed in different banks. With the help of the column-access data read, the activity factor for the input image readout in SF-CONV can be reduced by 76.2% in the test case.

The measurement results are shown in Fig. 14.6.6. With the help of the AHFC, the overall energy consumption of the HHFD can be reduced by 39%, compared to that of only the DHFU. The CNNP can operate at 0.46-to-1.0V supply with 5-to-100MHz clock frequency. The peak power consumption with maximum PE utilization at 0.46V and 1.0V is 5.3mW and 211mW, respectively. The minimum energy point (MEP) is 0.46V, and the energy efficiency at the MEP is 1.06J/cycle, which is 2× lower than the energy efficiency at 1.0V. The DM-CNNP and SF-CONV process the CNN operations for FR in 26.3ms at the MEP, achieving 97% accuracy in LFW dataset [6]. The accuracy degradation from SF-CONV is less than 1%. The FR result using FIS and CNNP is accurate as shown in Fig. 14.6.6, and the proposed FR system dissipates 0.62mW on average at 1fps frame rate.

The 3.30×3.36mm² FIS and 4×4mm² CNNP are fabricated in 65nm CMOS technology, as shown in Fig. 14.6.7. The ultra-low-power CNN FR system is successfully realized for always-on wearable devices.

References:
Ultra-low-Power Always-on Wearable Device

- **Power:** ~1W
- **Framerate:** ~1fps
- **Accuracy:** >95%

**Figure 14.6.1:** Ultra-low-power CNN face recognition in wearable device.

**Figure 14.6.2:** Overall architecture.

**Figure 14.6.3:** Analog-digital Hybrid Haar-like face detector.

**Figure 14.6.4:** Ultra-low-power CNN processor with local distributed memory.

**Figure 14.6.5:** Convolution with separable filter approximation and T-SRAM.

**Figure 14.6.6:** Measurement results.
Figure 14.6.7: Chip photograph and performance summary.

<table>
<thead>
<tr>
<th>Face Image Sensor</th>
<th>CNN Processor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Technology</strong></td>
<td>65nm FPD Logic CMOS</td>
</tr>
<tr>
<td><strong>Die Area</strong></td>
<td>3300x3300μm$^2$</td>
</tr>
<tr>
<td><strong>Pixel Array</strong></td>
<td>320x240</td>
</tr>
<tr>
<td><strong>Pixel Size</strong></td>
<td>7x7μm$^2$</td>
</tr>
<tr>
<td><strong>Supply Voltage</strong></td>
<td>2.5V, 5.0V</td>
</tr>
<tr>
<td><strong>Clock Frequency</strong></td>
<td>50GHz</td>
</tr>
<tr>
<td><strong>Power Consumption</strong></td>
<td>23.8uW@560G</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>100MHz-1GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Supply Voltage</strong></td>
<td>0.4V-1.4V</td>
</tr>
<tr>
<td><strong>Peak Power Consumption</strong></td>
<td>3.3mW (800MHz)</td>
</tr>
<tr>
<td><strong>Energy Efficiency</strong></td>
<td>0.16mJ/μW (340MHz)</td>
</tr>
</tbody>
</table>
14.7 A 288µW Programmable Deep-Learning Processor with 270KB On-Chip Weight Storage Using Non-Uniform Memory Hierarchy for Mobile Intelligence

Suyoung Bang1, Jingcheng Wang1, Ziyun Li1, Cao Gao1, Yeojoong Kim1,2, Qing Dong1, Yen-Po Chen1, Laura Fick1, Xun Sun1, Ron Dreusli1, Trevor Mudge1, Hun Seok Kim1, David Blaauw2, Denis Sylvester1

1University of Michigan, Ann Arbor, MI
2CubeWorks, 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 0.8V 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 readout 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 Vth 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 288µJ 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:
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).

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).

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.