Analog In-Memory Subthreshold Deep Neural Network Accelerator

L. Fick, D. Blaauw and D. Sylvester
University of Michigan
{lfreyman, blaauw, dmcs}@umich.edu

S. Skrzyniarz, M. Parikh and D. Fick
Isocline Engineering
{skylarjs, malav.parikh, dave.fick}@isosemi.com

***Abstract*—Low duty-cycle mobile systems can benefit from ultra-low power deep neural network (DNN) accelerators. Analog in-memory computational units are used to store synaptic weights in on-chip non-volatile arrays and perform current-based calculations. In-memory computation entirely eliminates off-chip weight accesses, parallelizes operation, and amortizes readout power costs by reusing currents. The proposed system achieves 900nW measured power, with an estimated energy efficiency of 0.012pJ/MAC in a 130nm SONOS process.**

***Keywords*—low power; subthreshold; neuromorphic; in-memory; non-volatile**

I. Introduction

Commercial hardware neural network algorithms rely on data connectivity to perform cloud-based computation, or high power digital processors for hardware acceleration [1,2]. Some applications, shown in Fig. 1, don't require the high speeds and throughput (187 GMAC/s) achieved in these implementations, and would benefit from low peak-power in order to extend battery life. Low-power mobile applications could operate in a wake-up routine using a slow, always-ON subthreshold DNN, and a higher power, faster DNN that wakes up infrequently.

Traditional hardware neural networks store weights in off-chip non-volatile memories, calculating a given neuron dot-product by cycling through its synaptic weights, multiplying them by input values and accumulating in a digital register. These approaches suffer from high power consumption due to the large number of off-chip memory accesses required, on the order of hundreds of thousands for state-of-the-art architectures [3]. Researchers are working to lower power consumption through reducing the total number of off-chip memory accesses.

One method of reducing off-chip memory accesses is to use convolutional neural network (CNN) architectures, which heavily reuse weights [4]. Data sparsity in both inputs and synaptic weights also provides a reduction in memory accesses, allowing the system to selectively choose whether to fetch memory based on known values. Previous research has shown that data sparsity between 30-90% produces significant energy savings [4,5].

The proposed circuit expands upon these ideas by storing synaptic weights in an on-chip non-volatile memory. This allows for the complete elimination of off-chip synaptic weight accesses, reducing the total number of off-chip memory reads by $1/(N+1)$ times for an N neuron system. State of the art neural networks often employ thousands of neurons, making this memory reduction significant [3].

Using non-volatile memory cells as computational units through analog current summation allows for the amortization of synaptic weight read current with calculation current, the parallelization of all multiply-accumulate functions for a neuron, and an inherent use of data sparsity. Rather than cycling through synaptic weights one by one, the proposed system calculates a full dot-product for all inputs and weights of a given neuron simultaneously, increasing energy efficiency. Operating the memory cells in subthreshold allows for further reduction of current consumption, lowering the total peak power consumed by each neuron.

The proposed system achieves 900nW power consumption in a 130nm SONOS process, storing 14,592 analog synaptic weights in on-chip memory.

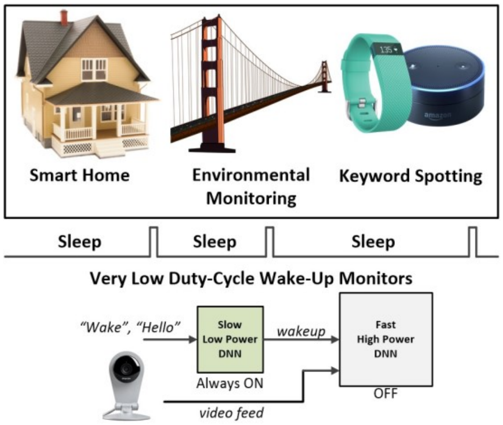

Fig. 1 Low-duty wake up routine applications can benefit from slow, ultra-low power DNNs

II. Subthreshold Computation

The following sections discuss the neuron dot-product calculation and how it is accomplished using in-memory subthreshold current computation.

A. In-Memory Current Computation

Synaptic weights are stored as threshold voltages in an array of 2T SONOS cells. Fig. 2 shows the basic computational unit proposed in this work. The unit consists of a SONOS cell which stores weights as threshold voltages, and an access transistor

978-1-5090-5191-5/17/$31.00@2017 IEEE

which gates the current and takes a neural network input as its gate voltage. This 2T pair will function as a current source in an array, producing a current proportional to the multiplication of its input and synaptic weight. When tiled in a standard NOR flash topology, the currents produced by each computational unit will sum together along a shared bit line. The resulting current is equal to the full dot-product of the selected neuron.

The goal of this work is to create a subthreshold compute cell whose current is equal to the linear multiplication of its input and threshold voltages. The equation shown in Fig. 2 is the subthreshold MOSFET current equation. In order to perform a linear multiplication, we cancel the exponential components of this equation. The threshold voltage is programmed via an offline routine to precisely set the voltage and current through each cell. Using an exponential synaptic weight mapping allows us to cancel the exponent related to V_{th}. Gate-to-source voltage represents the output value of a previous neuron, and can be mapped logarithmically to cancel its exponent as well. The equations developed in Fig. 2 (right) show how these terms are cancelled, producing a linear multiplication between V_{th} and V_{gs}. The following sections discuss how these mappings are implemented on-chip.

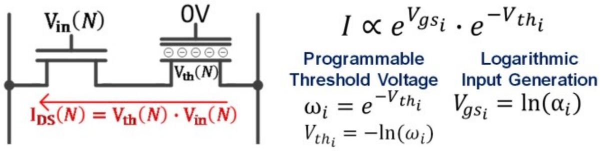
$V_{in}(N)$
0V
$V_{th}(N)$
$I_{DS}(N) = V_{th}(N) \cdot V_{in}(N)$
$I \propto e^{V_{gs_i}} \cdot e^{-V_{th_i}}$
Programmable Threshold Voltage
Logarithmic Input Generation
$\omega_i = e^{-V_{th_i}}$
$V_{th_i} = -\ln(\omega_i)$
$V_{gs_i} = \ln(\alpha_i)$

Fig. 2 In-memory analog computation unit

B. *Off-Chip Weight Programming*

Synaptic weight programming is achieved via an offline write routine consisting of a series of program and read pulses. By using small enough program pulses (10s of microseconds), we are able to precisely set the on-chip threshold voltages to produce any arbitrary mapping. Setting these voltages to produce a logarithmic transfer function cancels the exponential component of the subthreshold current equation.

Analog weight storage requires periodic re-writes in order to maintain precision. Estimated threshold voltage leakage would require top-off programming each day, based on simulations.

C. *Logarithmic Voltage Conversion*

In order to cancel the exponential effects of gate-to-source voltage, we can apply a logarithmic conversion from the output of one DNN layer to the input of another. Input voltages are determined by the output current of a previous layer, and applied to the gate of an access transistor in a subsequent layer, shown in Fig. 3.

Amplifiers are used to precisely hold the drain-to-source voltages across all 2T SONOS cell pairs. This forces a low-distortion operating condition, and also allows us to measure the amount of current consumed by an array of synaptic weights. Both amplifiers source and sink the current required to hold the drain and source voltages at a constant value. This current is equal to the neuron dot-product.

By forcing current through a subthreshold diode connected MOSFET we can invert the current-to-voltage relationship and generate a logarithmic voltage output: $V_{out}=\ln(I_{neuron}\times e^{Vth})$. The threshold voltage of the MOSFET is constant, producing a logarithmic voltage dependent only upon the neuron dot-product current. This voltage is fed to the access gate of a subsequent layer current cell, and generates a linear relationship between the current in layer 1 versus the current in layer 2. The graphs in Fig. 3 show the relationship between current, voltage, and current in the subthreshold accelerator.

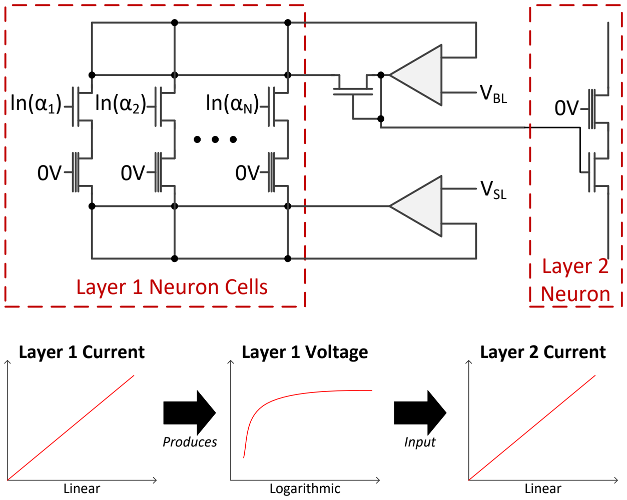

Fig. 3 (Top) Logarithmic voltage generation and layer-to-layer connections (Bottom) Linear/Logarithmic current-voltage-current transformation

D. *Multiplexing Between Neurons*

When connected as a tiled array, neurons share gate voltages, but drain and source bit lines are separated, allowing for different current flows through each neuron. Using a force-sense feedback structure, the amplifiers are multiplexed onto a selected neuron in order to calculate the dot-product. When selected, the amplifiers drive the drains and sources of the neuron to precise voltages, and measure the current. All other neurons are un-selected, their drain and source voltages are floating, and they draw no current.

A tradeoff between peak power consumption, area, and throughput can be made by changing the number of multiplexed neurons. In a fully parallel architecture, each neuron has its own amplifier to source/sink its dot-product current, with no multiplexers, and all neuron, input and weight dot-products are calculated simultaneously.

E. *System Architecture*

The fabricated system architecture is shown in Fig. 4 and includes a 228×64 SONOS cell array, two charge pumps for program and erase functionality, drivers, multiplexers and a diode connected MOSFET for logarithmic voltage readout. For proof-of-concept purposes the amplifiers are sourced from off-chip. The system includes one layer of a neural network, but has analog voltage pads and multiplexers to provide voltages to the access transistor inputs of the 2T current cells. To test the

This work was supported by the SRC STARnet SONIC center.

functionality of a multi-layer system, currents are produced within the synaptic array, logarithmic voltages are read off-chip and then fed back into the chip via the analog pads to the access transistors. This forces a linear current to be produced again within the synaptic array.

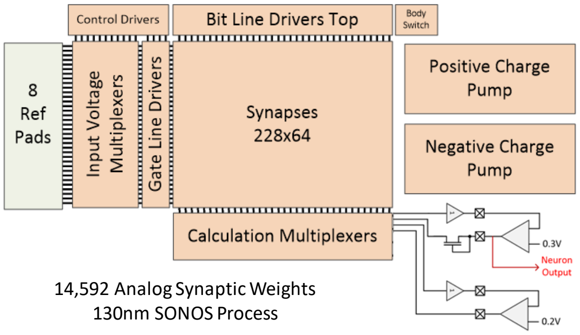

Fig. 4 Full system architecture

III. Measured Results

The proposed circuit was fabricated in a 130nm SONOS process. With off-chip amplifiers, the in-memory analog computational units were characterized to determine the efficacy of subthreshold computation in non-volatile cells.

A. Logarithmic Voltage Generation

Measured results in Fig. 5 show the logarithmic output voltage generation achieved by the on-chip diode connected MOSFET. Between 0.1uA and 1uA total neuron current consumption the voltage at the output of the diode is logarithmic. With a logarithmic best fit the output voltage has an r-square value of 0.9989.

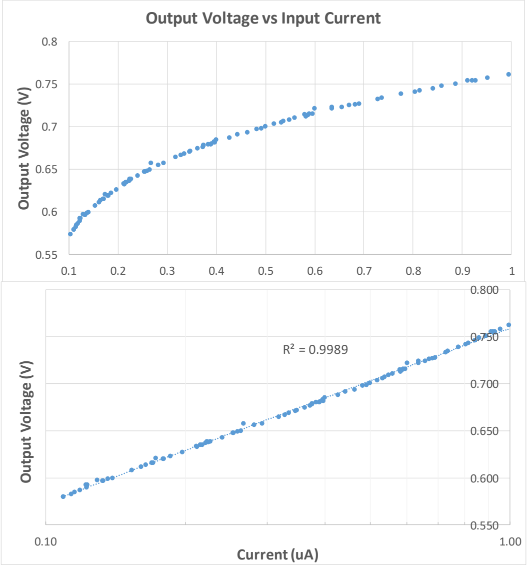

Fig. 5 Logarithmic output voltage measurement

Each point in the graph represents a read cycle during the program/read routine. In precisely setting the threshold voltages of the SONOS cells, we are able to fully characterize the cell current range, programming capability and logarithmic output voltage functionality.

B. Current Generation and Dot-Product Calculation

When applied to the gate of an access transistor, and using different threshold voltages, we can demonstrate the efficacy of the analog multiplication, shown in Fig. 6. Current produced by the multiplication of logarithmic input voltages, at different synaptic weights, is linear with r-squared values between 0.9714 and 0.9991. In the opposite dimension, current produced by logarithmically transformed synaptic weights, at different input voltages, is also linear but with reduced linearity between 0.9017 and 0.9932. Measured neuron current ranges between 0 and 1µA for the full neuron dot-product, designed to be within the subthreshold region of operation for the diode connected MOSFET.

Future work on this project could increase linearity in relation to threshold voltage by further analyzing programming characteristics, including fast-shift induced threshold voltages changes, leakage, and weak-programming caused by write disturbs. Nonlinearities within the multiplication function can also be calibrated out through the neural network training routine. Given a set of nonlinear neurons with measured transfer curves, a training routine can learn to compensate, adjusting synaptic weights appropriately to achieve high accuracy inference.

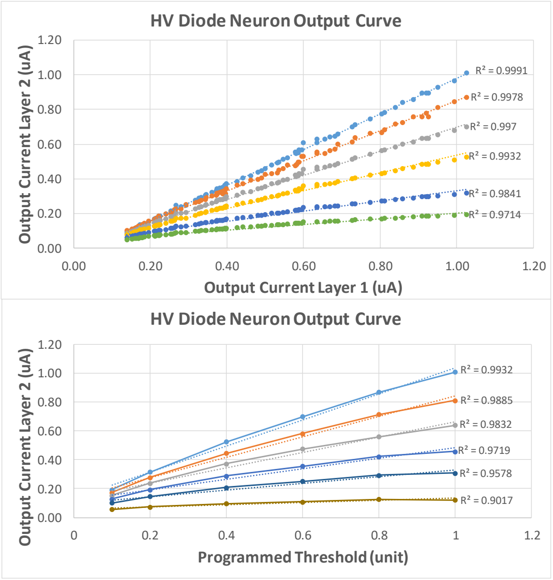

Fig. 6 Current multiplication (Left) layer 2 current vs layer 1 current (Right) layer 2 current vs threshold voltage

After each synaptic weight is programmed, a series of input voltages are applied to the neuron to produce a full dot-product summation. Because current sums in parallel along the shared bit line, we are able to compute the full dot-product of a single neuron in one cycle. To test the accuracy of the current summation of all multiplied 2T cell currents, 626 combinations of both synaptic weights and input voltages were generated. Input voltages varied among 8 different analog values, and synaptic weights were defined on a range that produced currents less than 1uA in total, from Fig. 5 (top).

Fig. 7 shows the measured results of the sub-threshold dot-product calculations. Each point is a full dot-product calculation for a selected neuron, and is plotted against the expected value. The proposed system generates an output current with an r-squared error value of 0.9994.

C. Power Consumption and Energy Estimations

Neural networks are trained in order to maximize inference accuracy, and to minimize total cost, defined as synaptic weight values. Because of this cost function, synaptic weights typically exist in a mean zero Gaussian distribution. When mapped to hardware, with threshold voltages representing weights, we define a zero weight as a fully off cell, producing no current. A zero-value input is a voltage that produces zero current in the 2T SONOS cells, regardless of V_{th}.

The proposed circuit inherently exploits data sparsity in both synaptic weights and input values. If either value is zero, the current produced by the 2T cell will also be zero, consuming no power. Similar to the works presented in [4,5], the proposed system benefits from increased data sparsity, but does not require any additional circuitry to exploit it.

Average current is estimated to be 50% of the total neuron current range, equal to 500nA. At 1.8V supply voltage, the system is measured at 900nW power consumption. A subthreshold amplifier presented in [6] operates at 75nW power consumption, with an estimated settling time of 1µs, based on its bandwidth neuron RC load. A full system would consist of a synaptic weight array where one bit line is driven by a subthreshold amplifier, and the other line is shorted to a supply. The total system has an estimated power consumption of 975nW, at 1µs, with an energy efficiency of 0.012 pJ/MAC. Throughput of the system is estimated at 0.07 GMAC/s, but could be increased to 1.22 GMAC/s with a fully parallel implementation (one amplifier per neuron).

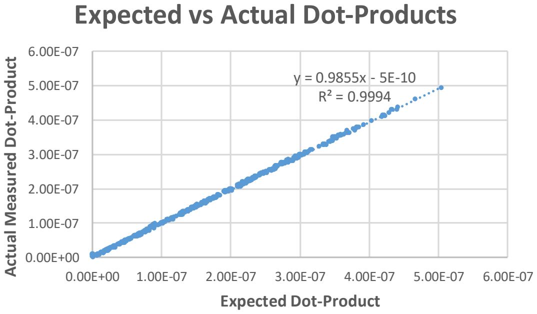

Fig. 7 Full neuron dot-product current vs expected value

D. Comparison to Prior Works

Compared to prior hardware implementations, shown in Table I, the proposed system achieves a power consumption of 900nW, a 79,000× reduction over the lowest power implementation presented in [5]. To achieve this reduction in power, we tradeoff performance, measured as GMAC/s, which is estimated at 0.07, when compared to the low power speed of 31, this is 442× slower. Performance can be increased up to 1.22 GMAC/s with a fully parallel implementation, drawing an estimated power consumption of 62.4µW.

The system was fabricated in a 130nm SONOS process, and the die shot is shown in Fig. 8. The chip occupies a die area of 1.93mm×1.25mm, for a total area of 2.41mm^2.

TABLE I. COMPARISON OF PRIOR WORK

	Power and Energy Comparison					
	Units	***CM1K***	***Tegra X1***	***Eyeriss***	***VLSI '16***	***This Work***
Off-Chip Memory?		Yes	Yes	Yes	Yes	No
Non-Volatile?		No	No	No	No	Yes
Technology	nm	130	20	65	40	130
Power	W	0.3	5.1	0.278	0.071	900e-9
Energy Efficiency	pJ/MAC	11.06	30.47	12.02	2.27	0.012[a]
Speed	GMAC/s	27.1	187	23	31	0.07

[a] Cycle time estimated at 1µs from simulation

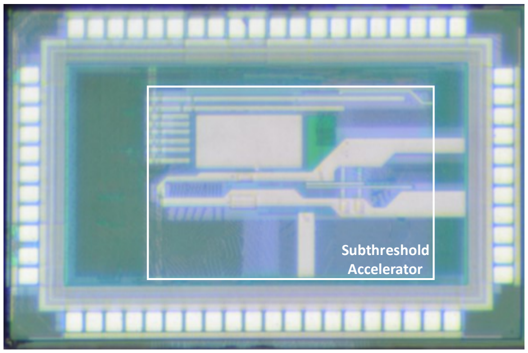

Fig. 8 Die shot of fabricated subthreshold DNN accelerator

IV. CONCLUSION

This works present a subthreshold accelerator for ultra-low power DNNs. The analog accelerator consumes 900nW, with an estimated 975nW total power consumption for a full system, operating at 1µs cycle time, with 0.012 pJ/MAC energy efficiency and 0.07 GMAC/s performance.

[1] S. Swamy and K.V. Ramakrishnan, An Efficient Speech Recognition System, No. 4, Vol. 3, CSEIJ, 2013, pp.21—27.

[2] NVIDIA, White paper: GPU-Based Deep Learning Inference, 2015.

[3] A. Krizhevsky, I. Sutskever and G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, NIPS, 2012.

[4] Y.H. Chen, T. Krishna, J. Emer and V. Sze, Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks, ISSCC, 2016.

[5] B. Moons and M. Verhelst, A 0.3-2.6 TOPS/W Precision-Scalable Processor for Real-Time Large-Scale ConvNets, VLSI, 2016.

[6] L. Magnelli, F.A. Amoroso, F. Crupi, G. Cappuccino, and G. Iannaccone, Design of a 75-nW, 0.5V subthreshold complementary metal-oxide-semiconductor operational amplifier, IJCTA 2012.