2020 IEEE International Electron Devices Meeting (IEDM) | 978-1-7281-8888-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/IEDM13553.2020.9371973

Unassisted True Analog Neural Network Training Chip

Y. Kohda[1], Y. Li[2], K. Hosokawa[1], S. Kim[2], R. Khaddam-Aljameh[3], Z. Ren[2], P. Solomon[2], T. Gokmen[2], S. Rajalingam[2], C. Baks[2], W. Haensch[2], E. Leobandung[2]

[1]IBM Research, Tokyo, Japan, [2]IBM Research, Yorktown Heights, NY, USA, [3]IBM Research, Zurich, Switzerland, email: eleoban@us.ibm.com

***Abstract*—** Analog In-Memory Computing using Resistive Processing Unit (RPU) has been proposed for Neural Network (NN) training. However, hardware demonstration has been limited to using some digital emulation to assist the analog chip function. **Using capacitor as analog weight, we report the first analog Neural Network training chip, where ALL Multiple and Accumulate (MAC) function are performed in analog cross-point arrays, and all weights are updated in parallel.** The chip measure full MNIST training accuracy of 92.7% with run time faster than digital system in *real* time.

I. Introduction

Analog In-Memory Computing with RPU uses analog cross-point array to compute matrix-vector products in one step, irrespective of the size of the matrix as shown in Fig. 1 [1]. Each weight represents varying conductance and the sum of the current through the weights does the single step matrix multiplication. RPU also stores weights locally in the array and updates all weights in parallel pass, thus minimizing data movement, resulting in significant speed and power benefit.

Prior works on Analog NN training used digital emulation of MAC combined with analog arrays, in which case the error generated by non-ideal analog hardware was effectively compensated by digital part. None has reported full hardware training where all MAC functions are completed on analog chip, and all weight are updated in parallel. Such non-emulation-assisted demonstration requires novel cross-point array and peripheral design with tightly distributed symmetric weights. We achieved this using capacitor plus read-out FET as analog weight because of the excellent symmetry. Fig. 2 shows previously reported capacitor base unit cell and the excellent symmetry data [2]. The weight is represented by the capacitor voltage which varies linearly and symmetrically with capacitor charge. The capacitor drives a read-out FET (T3) whose conductance is then proportional to capacitor charge.

II. Analog ASIC Design

The Analog ASIC is designed and fabricated in CMOS 90nm technology using MIMCAP as capacitor weight. Fig. 3 shows the block diagram of the ASIC chip. Fig. 4 shows the x-section of the fabricated chip with C4 as I/O pin. The ASIC chip has three independent Analog cores (AC). Each core consists of cross-point array with capacitor weights. Specifically, AC1 has 529x100 weights, AC2 has 100x10 weights, and AC3 has 100x100 weights. Two banks of Pulse Generator (PG) are attached to the sides of the cross-point array to input signal into the array. For output, each array has integrator bank(s) multiplexed to ADC banks with 100 or 10 units depending on its output size. Each Analog Core can be connected independently to form flexible NN configuration. During training, the ASIC chip is controlled using external FPGA board as shown in Fig. 5. Training NN typically uses backpropagation algorithm consisting of forward pass, backward pass, and weight update pass [3]. In forward or backward pass, SRAM obtain the input from FPGA through the I/O interface. Next, the input bits are converted into voltage pulses by the PGs for all the input rows or columns and simultaneously sent through the cross-point array. Output currents for all columns or rows are then collected by the integrator bank which converts them to output voltages. Next, ADCs convert the output voltages to digital bits and send them back to FPGA. **All analog MAC step in one array is done in parallel for all inputs, which is first key element of RPU speed up.** The duration of the input pulses can be adjusted by global register. Non-MAC function, i.e. Non-linear Function (NLF) and Error Vector, are calculated by FPGA.

The weight update operation is performed using stochastic update rule [1]. Weight update is calculated by multiplying the neuron inputs (encoded in rows) and the Error Vector (encoded in columns). This multiplication is achieved using AND operation of stochastic pulses from rows and columns, meaning each weight is only updated when there is coincident pulse for both its row and column. The stochastic pulses are generated from FPGA input data and sent to all rows and columns simultaneously through PGs and the update lines. All weights on same row get same row pulses and all weights on same column get same column pulses. But since each weight has unique row and column number, each weight update is different. **All analog weights in one Analog Core are updated simultaneously and in parallel, which is second key element of RPU speed up.** For example, the AC1 with 52.9k weights are updated in just 2 passes for positive and negative update vector. One update pulse represents ~ 10^{-3} of full capacitor charge, thus each analog capacitor can represent ~ 1000 states.

Fig. 6 shows the block diagram of the integrator. The integrator integrates charges from the cross-point array output. The charge is stored on a capacitor. One terminal end is pinned at $V_{DD}/2$ and the other terminal end generates a voltage defined by the integrated charge. The voltage is then transformed into VNeg and VPos by a voltage shifter and differential circuit, which is then read by an ADC. The integrator charge can be discharged through a transistor triggered by a reset signal.

The block diagram for asynchronous self-clock ADC is shown in Fig. 7. A 200 MS/s 9bit Successive-approximation-register (SAR)-ADC design is employed [4]. By choosing an asynchronous self-clocked scheme for the ADC, as opposed to the conventional synchronous one, two things are achieved. First, the conversion speed is maximized. Second, the amount of control signals needed to coordinate the ADC operation is minimized to a sole sample pulse signal CK_SMP. While this signal is high, the differential input voltage is sampled via a pair of bootstrapped switches onto the ADC's internal capacitor array. Following the falling edge of the CK_SMP signal, the internal state-machine is activated, and the ADC starts to perform the successive approximation algorithm for digitizing the sampled voltage. The differential design of the ADC helps reject supply noise stemming from other digital circuits as well as the many other ADCs that are operating in parallel. A single conversion procedure takes 6.7ns and consumes 15.6pJ, out of which 8pJ are attributed to the internal reference voltage buffer that also consumes 1.17mW in standby. Fig. 8 shows linear behavior of the ADC within input range of 0.4V to 1.3V.

The detail schematic of the unit cell is shown in Fig. 9. The weight capacitor is connected to the gate of a pair FETs (T3 and T4) to provide differential readout, which is controlled by a NOR gate. During forward path, the Xr line is connected to PG for input pulse and the Yr line is connected to an integrator. T5 and T6 are off and T7 is on, and the differential current of T3 and T4 is collected by the integrator through Yr line. During backward path, the row and column are switched, and Yr line is connected to input line while Xr line is connected to the integrator. T5 and T7 are off and T6 is on, and the readout current is collected through Xr line. Fig. 10 shows the simulated readout current as a linear function of capacitor voltage. Due to differential readout, the output current can be negative; therefore, negative weights are naturally implemented. The weight capacitor is charged/discharged by two current source FETs (T1, T2) as controlled by an AND gate. As described before, during weight update, two series of stochastic pulses (Yw and Xw) are sent to the input of AND gate through the update row and column lines. The coincidence of the two stochastic pulses will turn on the AND gate. As a result, global analog voltage VP(VN) will be applied to the gate of current source T1(T2) to provide negative(positive) update. The current sources are operating in saturation so that the update current is independent of capacitor voltage. Positive and negative weight update are done separately. When not doing positive update, T2 gate voltage is biased at or below GND to turn-off T2. Similarly, when not doing negative update, T1 gate voltage is biased at or above V_{DD} to turn-off T1.

Due to off-state leakage through the current source (T1/T2), the weight capacitor will decay systematically. Impact to training is small if the weight decay is < 1E-6 per update cycle because all weights are constantly updated during training [2]. Fig. 11 shows the measured weight decay with estimated ~ 5E-7 decay per training update cycle. This decay will vary with FET I_{off} variation. As described previously, update symmetry is a critical parameter for RPU. This is measured by performing 6000 positive and negative updates on a row and measure the weight at each update, as shown in Fig. 12. Variation is seen in NFET/PFET current source symmetry which can be improved with optimized layout of the unit cells.

III. Neural Network Training Results

A two layers Deep Neural Network (DNN) is used to test the ASIC chip with cropped MNIST data input (22 by 24 pixels) as shown in Fig. 13. Two out of the three Analog cores, AC1 with 529x99 weights and AC2 with 99x10 weights, (the last row in two ACs are used as bias) in the ASIC chip are used. To optimize the NN, the learning rate can be tuned by adjusting the gate voltage (VN/VP) of the current sources or by adjusting multiplier in the Error Vector calculation. Fig. 14 shows the measured relative training error rate as a function of Error Vector multiplier for various VN/VP.

Update Symmetry can be adjusted by adjusting the magnitude of VN and VP. Fig. 15 shows the relative training error rate as function of VP at fixed VN. If NFET and PFET current source match perfectly, best symmetry and lowest error should be at VN=V_{DD}-VP. Due to different threshold voltage and on-current, this is shifted by 20-30 mV. Simulation in Fig. 16 shows reducing current source variation can improve accuracy. As described, capacitor charge continuously leaks through the NFET or PFET current source during off-state. This leakage can be minimized by adjusting the current source off-state gate voltage. Relative training error as function of off-state gate voltage is shown in Fig. 17. Best accuracy, i.e. lowest leakage, is with off-state gate voltage of 0.1V below GND or above V_{DD} for NFET/PFET. With parameter optimization, Fig. 18 shows full MNIST training accuracy of 92.7%.

The ASIC clock is running at 100 MHz and current training time is approximately 1.02s for one epoch (60K images) limited by interface with FPGA. The ASIC analog MAC function itself takes 0.28s for one epoch using two analog cores. In contrast, typical digital system MNIST training time is > 1 min [5]. While it has been projected before, **this is the first demonstration that analog Neural Network training can be faster than digital system in *real* time.** The unit cell area of ASIC is 114 um^2 fabricated in CMOS 90nm. By scaling to 7nm technology and using simpler unit cell, we project to achieve >200x reduction in FET area. The capacitor area can also be reduced by >200x by reducing the current source leakage and using high density capacitor such as DRAM stack capacitor. We project the total unit area can be reduced to < 0.23 um^2. Weight Sync can also further reduce area for CNN training [6].

IV. Conclusion

We demonstrated a fully analog non-assisted NN training chip for the first time. During training, all MAC operations are done in analog cross-point arrays, and all analog weights are updated simultaneously and in parallel, resulting in faster run time than typical digital system in *real* time. This Analog ASIC chip is built using CMOS 90nm technology using MIMCAP as symmetric analog weight. We reported optimum learning rate, leakage, and symmetry parameters. The chip demonstrates 92.7% full MNIST training accuracy.

The authors acknowledge T.C. Chen for his support.

References

[1] T. Gokmen et al., “Acceleration of Deep Neural Network Training with Resistive Cross-Point Devices: Design Considerations,” Frontiers in Neuroscience, vol. 10, pp. 333-345, Jul. 2016.

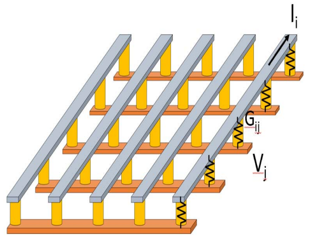
[2] Y. Li et al., "Capacitor-based Cross-point Array for Analog Neural Network with Record Symmetry and Linearity,” IEEE Symposium on VLSI Technology, pp. 25-26, Jun. 2018.

[3] D.E. Rumelhart et al. “Learning representations by back-propagating errors,” *Nature* 323, pp. 533–536, 1986.

[4] L. Kull et al., “A 24-to-72GS/s 8b time-interleaved SAR ADC with 2.0-to-3.3pJ/conversion and >30dB SNDR at Nyquist in 14nm CMOS FinFET,” ISSCC, pp. 358–360, Feb. 2018.

[5] Y. Wu et al., "Experimental Characterizations and Analysis of Deep Learning Frameworks," IEEE International Conference on Big Data, pp. 372-377, 2018.

[6] E. Leobandung et al., “Synchronized Analog Capacitor Arrays for Parallel Convolutional Neural Network Training, “MWSCAS Proceedings, pp. 387-390, Aug. 2020.



$$\begin{pmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{pmatrix} = \begin{pmatrix} G_{11} & G_{12} & \cdots & G_{1m} \\ G_{21} & G_{22} & \cdots & G_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ G_{n1} & G_{n2} & \cdots & G_{nm} \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ V_m \end{pmatrix}$$

Fig. 1. Cross-point Resistive Processing Unit (RPU) for single step Vector-Matrix Multiplication. V_n is the input to the array and sum of current I_n is the output based on Kirchhoff's Law.

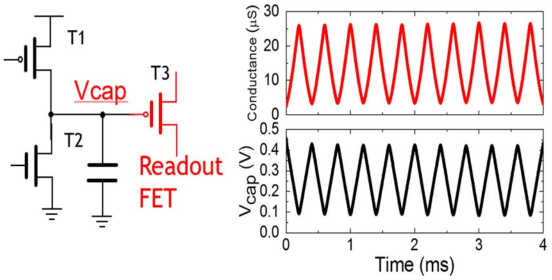


Fig. 2. (a) Schematic of a capacitor-based weight. T1 and T2 are current source of capacitor and T3 is read-out FET driven by capacitor (b) Experimental results for single-cell update with 8000 pulses and corresponding capacitor voltage change [2].

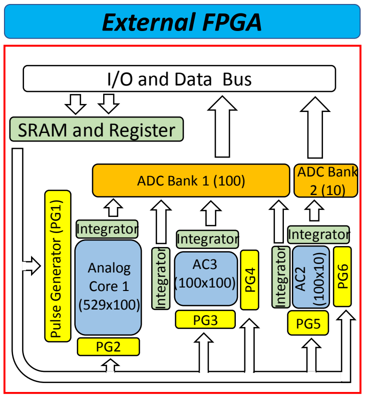


Fig. 3. ASIC chip block diagram with three cross-point Analog Cores. Each core can be used and controlled independently using external FPGA.

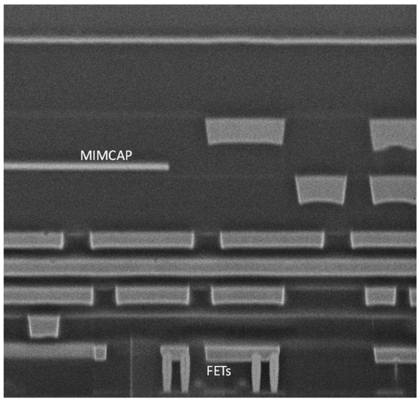


Fig. 4. X-section of the chip fabricated in CMOS 90nm technology with MIMCAP. Seven level metals are used with C4 I/O pin out.

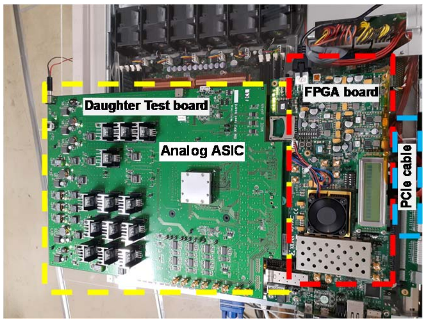


Fig. 5. Test setup where the Analog ASIC chip is connected and controlled by the FPGA board. The FPGA is connected to computer using PCIe interface for programming.

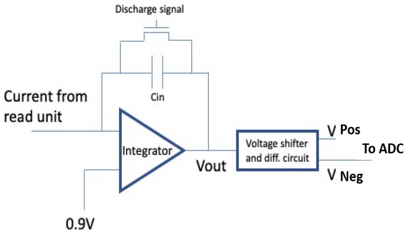


Fig. 6. Schematic of the Integrator. Current from cross-point array output is read and transformed into differential voltage input to ADC. A discharge signal is then employed to reset the integrator for next cycle.

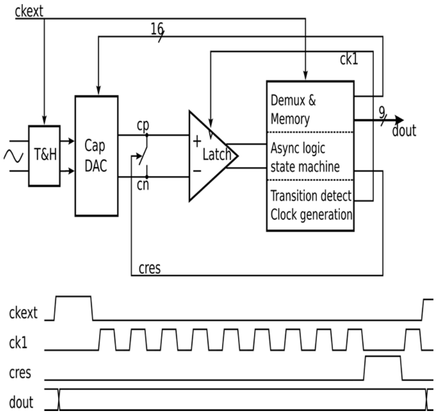


Fig. 7. Schematic Block Diagram of the self-clock asynchronous ADC.

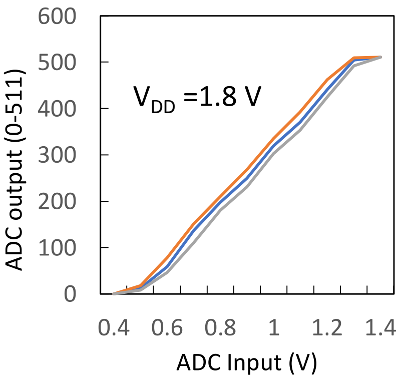


Fig. 8. Measured ADC output as function of ADC input voltage. Linear behavior is observed on the full range of ADC input voltage

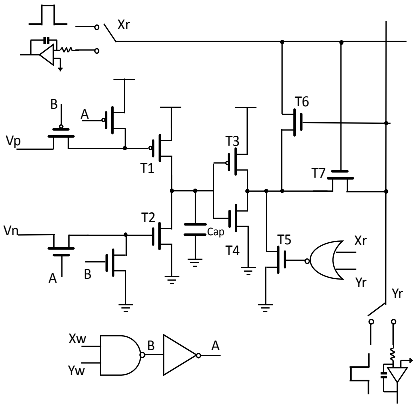


Fig. 9. Schematic of the unit cell. T3 and T4 represent readout FETs. Xr and Yr line are the cross-point array row and column lines. Xw and Yw signal come from the row and column update lines through Pulse Generator. VN/VP control the magnitude of the update current.

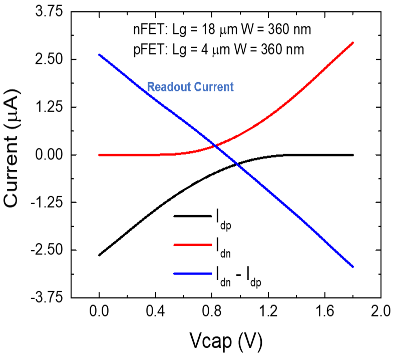


Fig. 10. The simulated readout current as a function of capacitor voltage. Both positive and negative current is possible to represent negative and positive weight.

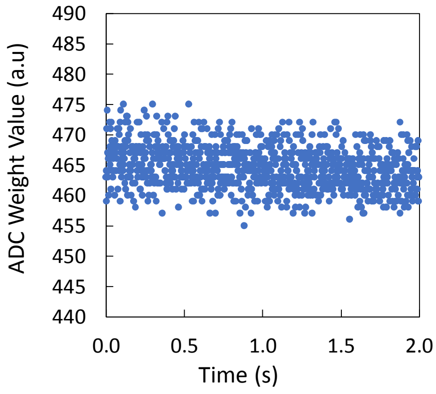


Fig. 11. Typical measured weight decay in the unit cell as measured by ADC output. Weight decay depends primarily on variation of current source leakage current.

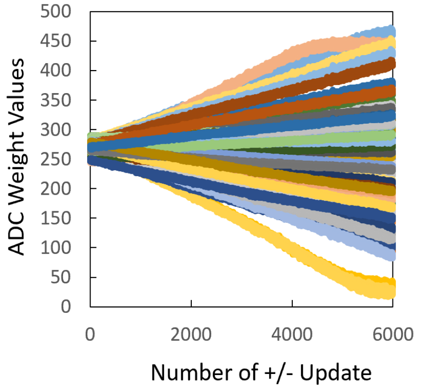


Fig 12 Statistical measurement of update symmetry. Positive and negative updates are performed on a row of 100 weights with zero net update. Ideal symmetry means weight should be flat.

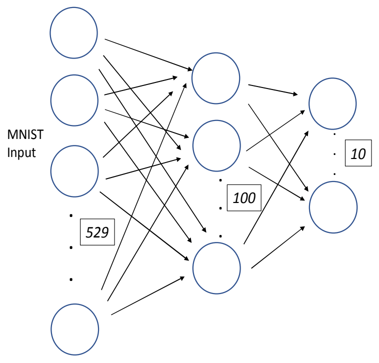


Fig. 13. Two layers Neural Network used in the training exercise. The first input layer has 528 MNIST input plus one input for bias, follows by hidden layer and output layer.

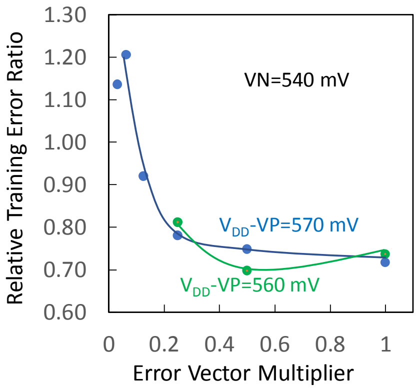


Fig. 14. Measured MNIST relative error rate as a function of learning rate adjustment. This test is used to determine optimum learning rate for the Neural Network.

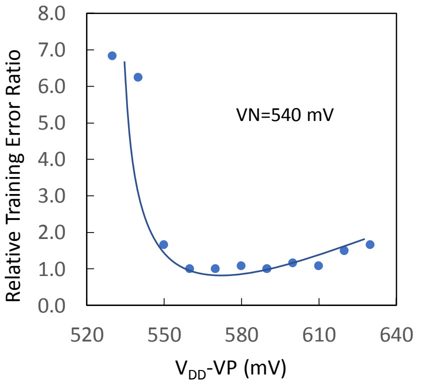


Fig. 15. Measured relative MNIST error rate as a function of V_{DD}-VP with VN=540 mV. Best symmetry is around V_{DD}-VP=560 mV. Since FET current changes much faster at lower gate voltage, worse asymmetry and hence accuracy, is seen at lower V_{DD}-VP.

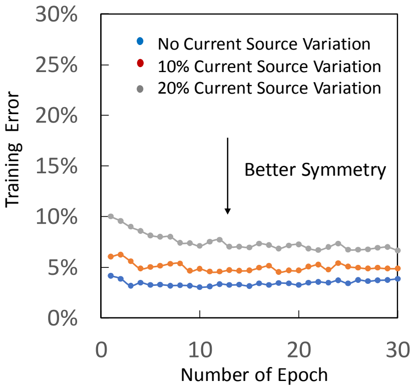


Fig. 16. Simulation of the Neural Network showing improving accuracy as the current source variation is reduced to improve symmetry. Weight decay of 1E-6 per update cycle is used.

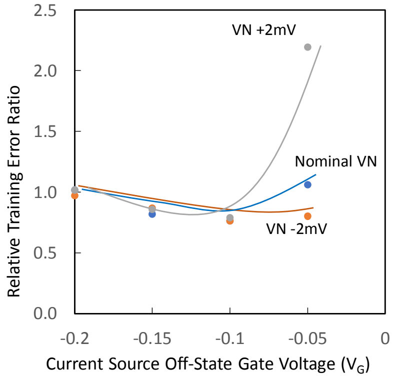


Fig. 17. Measured MNIST relative training error vs. current-source off-state gate voltage input. Lowest leakage is around V_G=-0.1V. When leakage is high, training error is worse and more sensitive to small variation such as on-state VN variation.

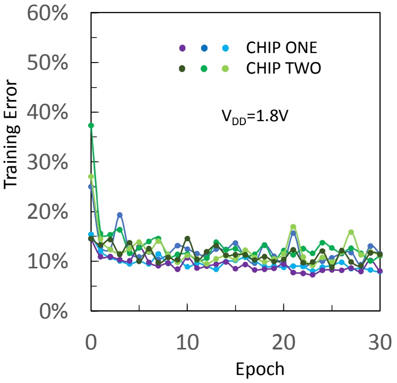


Fig. 18. Measured training accuracy for full MNIST data set for two different chips with test parameters optimization. Training accuracy of 92.7% has been achieved by optimizing the test parameters.