A Mixed-Signal Binarized Convolutional-Neural-Network Accelerator Integrating Dense Weight Storage and Multiplication for Reduced Data Movement

Hossein Valavi[1], Peter J. Ramadge[1], Eric Nestler[2], and Naveen Verma[1]

[1]Princeton University, Princeton, NJ, USA. [2]Analog Devices Inc., Cambridge, MA, USA. (hvalavi@princeton.edu)

Abstract

We present a 65nm CMOS mixed-signal accelerator for first and hidden layers of binarized CNNs. Hidden layers support up to 512, 3×3×512 binary-input filters, and first layers support up to 64, 3×3×3 analog-input filters. Weight storage and multiplication with input activations is achieved within compact hardware, only 1.8× larger than a 6T SRAM bit cell, and output activations are computed via capacitive charge sharing, requiring distribution of only a switch-control signal. Reduced data movement gives energy-efficiency of 658 (binary) / 0.95 TOPS/W and throughput of 9438 (binary) / 10.64 GOPS for hidden / first layers.

System Overview

Energy and performance of neural-network (NN) accelerators is bottlenecked by communication costs of bringing together many input activations (IAs) and neuron weights, and distributing many output activations (OAs). Algorithmically, lowering bit precision has been effective in reducing these costs [1, 2], and binarized NNs (BNNs), taking weights/activations to 1b [3], are showing success in increasing applications. This work exploits BNNs with charge-domain computation to achieve weight storage and multiplication in a bit cell only marginally larger than a 6T SRAM cell. This eliminates weight movement from memory, and minimizes /eliminates IAs/OAs movement.

Fig. 1 (top) shows the structure of a deep convolutional NN (CNN), consisting of convolutional first/hidden layers and fully-connected output layers. This work focuses on first/hidden layers, motivated by: (1) the trend of increasing depth (more hidden layers), making data movement in hidden layers dominate; and (2) mixed-signal implementation enabling direct sampling of analog sensors in first layer, potentially eliminating need for ADCs in sensor-inference applications, which are increasing in prominence. The chip architecture (bottom) implements one first/hidden layer, and enables layers to be cascaded in a pipeline. An Input-Activation (IA) SRAM and Input-Activation (IA) Buffer enable pipeline buffering, and a Neuron Array supports up to 512/64, 3×3×512/3×3×3 hidden/first-layer filters, followed by a Binarized Batch Normalization (Bin Batch Norm) block. The filter number (max. 512/64) and depth (max. 512/3) can be configurably reduced at proportional energy, and the height/width (min. 3×3) can be configurably increased by convolving outputs [4]. The pipeline computes output activations row by row (as inputs would be provided by active-matrix sensor). The IA SRAM stores 4 rows, processing 3 and buffering incoming, and the IA Buffer stores up to 3×3×512 IAs for a current filtering operation. The IA Buffer enables striding over IAs via 3-b shift registers and circular interface to the IA SRAM. Below, operation of hidden and first layers is described.

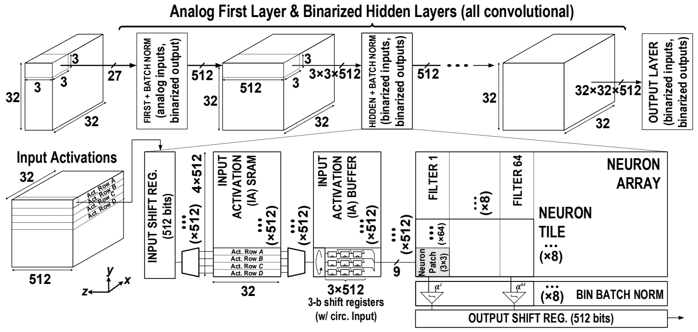

Fig.1: Structure of a deep neural network, with chip implementing convolutional first/hidden layers.

Hidden-Layer Circuit Design

The Neuron Array is organized into an 8×8 array of Neuron Tiles, each with clock-gating control. Fig. 2 shows a Neuron Tile, itself consisting of 3×3 Neuron Patches (filter width,height x, y), arrayed 64 vertically for a single neuron (filter depth z), and 64 horizontally for different neurons n. Input activations $IA_{x,y,z}$ are broadcast over the neurons and multiplied by locally-stored weights $w^n_{x,y,z}$. In BNNs IAs/weights are 1b, requiring just XNOR operation. The XNOR output $o^n_{x,y,z}$, is sampled as charge on a local cap, and accumulation for the output pre-activation PA^n is computed by shorting together all caps in neuron n.

Key to eliminating/minimizing weights/IA movement is the dense Multiplying Bit Cell (M-BC) shown in Fig. 2, and key to eliminating pre-activation movement is accumulation by charge shorting. The M-BC performs storage and XNOR in minimal area by introducing two PMOSs, in a standard 6T SRAM cell. Weights are stored as in an SRAM (via $WL, BL/BLb$). For neuron operation (see waveforms), all local capacitors are first pre-discharged, via PRE and $T-SHORT$, and charged only XNOR-conditionally via the PMOSs, by broadcast of differential $IA_{x,y,z}/IAb_{x,y,z}$ signals. Then, accumulation is performed by switches (controlled by $T-SHORT$), best situated outside the M-BC, since they share a common node in the neuron.

Layout density of the M-BC is critical for: (1) storing a large number of weights on chip, i.e., comparable to SRAM yet eliminating explicit accessing incurred in an SRAM; and (2) enabling up to 512 parallel neurons, so all input activations are broadcast just once, and over minimal distance. Fig. 3 shows the M-BC layout, having ~1.8x area of a standard 6T cell. This is achieved by: (1) employing PMOSs for the XNOR, to give more regular layout by balancing the N/PMOSs; and (2) employing charge-domain accumulation via MOM caps above, occupying no additional area. For custom layout, cell areas are estimated using logic, rather than SRAM, design rules; but, device sizing is equivalent to 6T cell, and M-BC employs regular gate-poly layout, for compatibility with SRAM rules. Monte Carlo (MC) simulations (see transient waveforms), show that, in addition to read/write, M-BC stability is easily maintained for XNOR, since only small isolated caps (1.2fF) are charged via the PMOSs. Note that while possible to use a 6T cell for current-domain XNOR and accumulation on BL/BLb [5], this can restrict neuron scalability

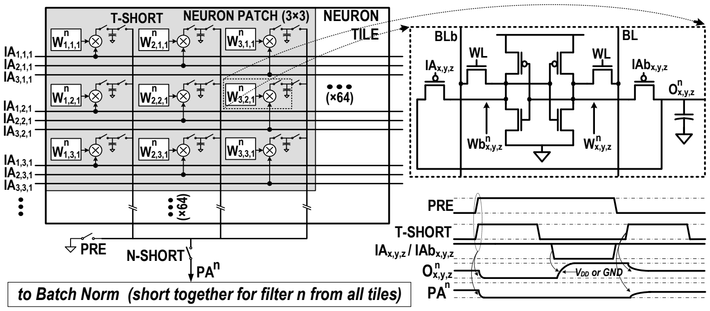

Fig. 2: Details of Neuron Tile, consisting of 3×3 M-BC Neuron Patches.

due to limited dynamic range of BL/BLb for I-V conversion, and can depend on transistor-current nonlinearity and temperature/process variations. In contrast, charge-domain accumulation benefits dynamic range by normalizing Q-V conversion by the total shorted capacitance, and caps are less susceptible to temperature/process variations/nonlinearity.

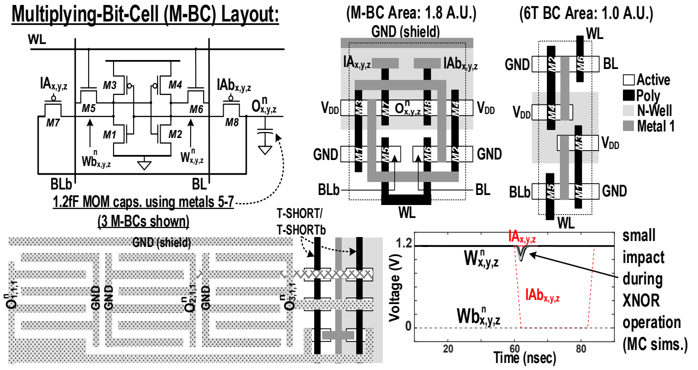

Fig. 3: Layout of M-BC and MC transient-stability analysis for XNOR.

Following charge accumulation, the pre-activation PA^n is fed to the Bin Batch Norm in Fig. 4, which, with binarization, only requires comparison against a reference value α^n, from training. Analog α^n is generated using the cyclic DAC shown, where pulse propagation in a shift register controls VDD/GND charging on a transfer cap CDAC1, as well as shorting to an equal-sized accumulation cap CDAC2. A cyclic DAC avoids binary-weighted caps, enabling small area for the 512 DACs needed. Simulating a range of image-recognition applications shows sampling noise is negligible and 6-b resolution is adequate. Since analog α^n takes values from GND-VDD, an N/PMOS-input comparator is selected via the MSB of α^n, to ensure comparator overdrive. The resulting output activation OA^n then feeds an output shift register.

First-Layer Circuit Design

For the first layer, which takes analog inputs, the precise input interface needed depends on the choice of sensor. For exploration, the 3×3×3 input activations are presumed to be available at once for sampling (depth is 3 for R/G/B imager). Fig. 5 shows the first layer realization. To improve analog-sampling fidelity, all capacitors of a neuron within one tile are shorted via $T-SHORT$, yielding 690fF input samplers. After discharging (via PRE), input charge is stored on a positive or negative sampler, depending on the weight value in a shift-register element. Charge from 8 samplers vertically across the Neuron Tiles is shorted together via $N-SHORT$. This yields differential partial pre-activation signals, with 4 such signals $PPA(1-4)^n_{+/-}$ needed to support filtering over 32 inputs, i.e., more than the required 3×3×3. $PPA(1-4)^n_{+/-}$ are then provided to the Signed-Analog-Accumulation (SAA) block. As shown, the SAA performs sampling and subtraction of positive and negative charge, while also offsetting the voltage to mid-VDD. Then, comparison with an analog reference α^n is performed for binarizing batch normalization.

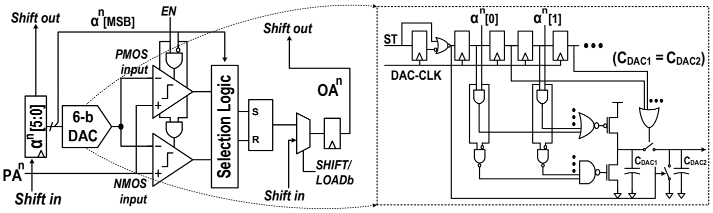

Fig. 4: Details of Binarizing Batch-Normalization (Bin Batch-Norm) block.

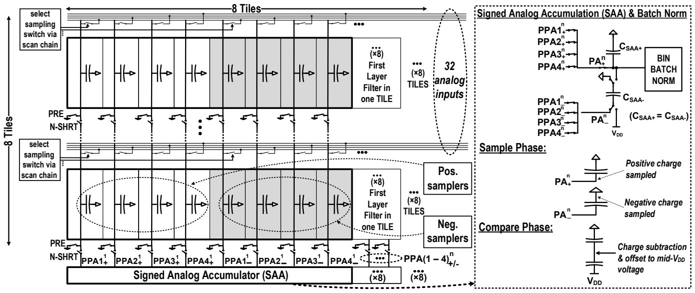
8 Tiles
8 Tiles
select sampling switch via scan chain
PRE
N-SHRT
(×8) First Layer Filter in one TILE
(×8) TILES
32 analog inputs
Pos. samplers
Neg. samplers
PPA1+ PPA2+ PPA3+ PPA4+ PPA1- PPA2- PPA3- PPA4-
PPA(1 – 4)+/-
Signed Analog Accumulator (SAA)
(×8)
Signed Analog Accumulation (SAA) & Batch Norm
CSAA+
CSAA-
BIN BATCH NORM
(CSAA+ = CSAA-)
Sample Phase:
Positive charge sampled
Negative charge sampled
Compare Phase:
Charge subtraction & offset to mid-VDD voltage
VDD

Fig.5: First layer, based on pos./neg. samplers and signed accumulation.

Measurement Results

Fig. 6(a) shows the prototype IC in 65nm CMOS. Fig. 6(b)/(c) show operation of hidden/first-layer neurons by plotting α^n at which the output activation switches vs. the PA^n value. Initially, some nonlinearity can be observed due to variable comparator offset over the analog α^n range (especially at α^n=6'd31, where N/PMOS-input comparator selection changes); but, since the charge-domain PA^n operation is much more linear, self-calibration is easily performed, by setting IAs and weights to sweep PA^n and adjusting α^n for linearity. This yields the linearity shown, measured with random IAs and weights subsequently applied, and with tight error bars showing sigma over all 512/64 hidden/first-layer filters. With M-BC and Bin Batch Norm at 940mV, and IA Drivers and control signals at 680mV, the energy per 3×3×512/3×3×3 hidden/first-layer filter is 14pJ/68pJ, both scaling proportionally with the number and size of neurons. Summary Table I shows the design integrates more weights on chip and achieves higher energy-efficiency and throughput than prior art. Table II shows performance equal to SW implementation is achieved for standard datasets and NNs.

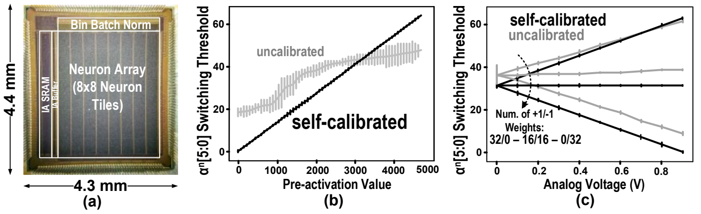

Fig. 6: Prototype (a) die photo; filter operation, for (b) hidden layer and (c) first layer (error bars over 512/64 filters).

TABLE I: MEASUREMENT SUMMARY & COMPARISON TABLE

	Chen, ISSCC"16	Moons, ISSCC"17	Bang, ISSCC"17	Ando, VLSI"17	This Work
Technology	65nm	28nm	40nm	65nm	**65nm**
Chip Area	16mm^2	1.87mm^2	7.1mm^2	12mm^2	**17.6mm^2**
Operating VDD	0.8 – 1.2V	1.0V	0.63 – 0.9V	0.55 – 1.0V	**0.68 / 0.94 / 1.2V**
CLK Freq.	200MHz	200MHz	1.9-19.3MHz	100-400MHz	**100MHz**
Bit Precision	16b	4 – 16b	6-32b	1b	**1b**
On-chip Mem.	108KB	128kB	270kB	100kB	**295kB**
Throughput	120 GOPS	400 GOPS	108 GOPS	1264 GOPS	**9438 GOPS**
TOPS/W	0.0096	10	0.384	6	**658**

TABLE II: ACCURACY COMPARISON FOR DATASETS & NNs.

	MNIST	CIFAR-10	SVHN
Test Accuracy (chip / SW)	**98.60%** / 98.92%	**83.27%** / 83.50%	**94.35%** / 95.10%
Validation Acc. (chip / SW)	**98.58%** / 98.75%	**84.09%** / 84.37%	**94.03%** / 94.63%
Baseline Neural Network (NN)	L1-2 : 64 CONV3 – BN L3-4: 128 CONV3 – BN L5: 10 FC	L1-2 : 64 CONV3 – BN L3-4: 128 CONV3 – BN L5-6: 256 CONV3 – BN L7-8: 1024 FC – BN L9: 10 FC	L1-2 : 64 CONV3 – BN L3-4: 128 CONV3 – BN L5-6: 256 CONV3 – BN L7-8: 1024 FC – BN L9: 10 FC

* L_{m-n}: indicates mth to nth layer. ** BN: indicates batch normalization layer.
*** X CON3: indicates X 3×3 binarized filters (depth defined by previous layer). **** Y FC: indicates Y fully-connected layers.

References

[1] B. Moons, et al., *ISSCC Dig. Tech. Papers*, pp. 246–247, 2017.
[2] S. Bang, et al., *ISSCC Dig. Tech. Papers*, pp. 250–251, 2017.
[3] L. Hubara, et al., *Proc. NIPS*, pp. 4107–4115, 2016.
[4] K. Simonyan, et al., *arXiv:1409.1556, Sep. 2014*.
[5] J. Zhang, et al., *Symp. VLSI Circuits*, pp. 252–253, 2016.