A 16×128 Stochastic-Binary Processing Element Array for Accelerating Stochastic Dot-Product Computation Using 1-16 Bit-Stream Length

Qian Chen, Yuqi Su, Hyunjoon Kim, Taegeun Yoo, Tony Tae-Hyoung Kim, and Bongjin Kim
School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
50 Nanyang Avenue, Singapore, 639798
Email: { e170029, yuqi003, kimh0003}@e.ntu.edu.sg, {tgyoo, thkim, bjkim}@ntu.edu.sg

***Abstract*- This work presents 16×128 stochastic-binary processing elements for energy/area efficient processing of artificial neural networks. A processing element (PE) with all-digital components consists of an XNOR gate as a bipolar stochastic multiplier and an 8bit binary adder with 8× registers for accumulating partial-sums. The PE array comprises 16× dot-product units, each with 128 PEs cascaded in a single row. The latency and energy of the proposed dot-product unit is minimized by reducing the number of bit-streams required for minimizing the accuracy degradation induced by the approximate stochastic computing. A 128-input dot-product operation requires the bit-stream length (N) of 1-to-16, which is two orders of magnitude smaller than the baseline stochastic computation using MUX-based adders. The simulated dot-product error is 6.9-to-1.5% for N=1-to-16, while the error from the baseline stochastic method is 5.9-to-1.7% with N=128-to-2048. A mean MNIST classification accuracy is 96.11% (which is 1.19% lower than 8b binary) using a three-layer MLP at N=16. The measured energy from a 65nm test-chip is 10.04pJ per dot-product, and the energy efficiency is 25.5TOPS/W at N=16.**

Keywords- stochastic computation, artificial neural networks, dot-product, multi-layer perceptron, image classification

I. Introduction

Stochastic computing [1-3] is an approximate computation method based on probabilities of random bit-streams (i.e., how many 1's in a bit-stream). Since bit-wise operations for stochastic computing are performed using compact digital logic gates, it is suitable for mobile applications which require small footprint. Despite the potential, stochastic computing has not been employed in practical applications due to major concerns in the latency and energy consumption. In practice, the typical bit-stream length for achieving the acceptable level of computation error ranges from several hundreds to thousands. For instance, a bit-stream length of 1024 is used to achieve 1.58% error for computing a 64-input dot-product using MUX, a representative stochastic adder [1]. Recent deep neural network (DNN) accelerators [4-6] have proposed novel circuits to improve the performance of stochastic computing. However, despite their efforts, most of the prior works still require a long stream of bits and consume significant energy while occupying the large footprint. In this work, we focus on the reduction of latency and energy of dot-product computation (i.e., an essential computation occupying >95% of state-of-the-art DNN processing), while keeping the high computation accuracy. The key contributions include the proposed circuit using XNOR-based stochastic multipliers and binary adders and spatial parallelism used for accelerating the stochastic dot-product computations.

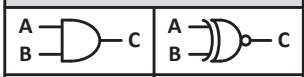
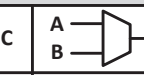
Stochastic Multiply		Stochastic Add	
			
Unipolar	Bipolar	OR-based	MUX-based
$P_C = P_A \cdot P_B$ where $0 \le P_i \le 1$	$S_C = S_A \cdot S_B$ where $S_i = 2 \cdot P_i - 1$	$P_C = P_A + P_B - P_A \cdot P_B$ (Error)	$P_C = (P_A + P_B)/2$ (Scaled)

Fig. 1. Typical multiply/add logic gates for stochastic computing.

II. Proposed Stochastic-Binary Dot-Product Macro

Fig. 1 shows a list of typical multiply and add logic gates used for stochastic computing. Either AND or XNOR gate can be used for stochastic multiplication. A unipolar multiplication based on AND gate is performed based on probabilities of bit-

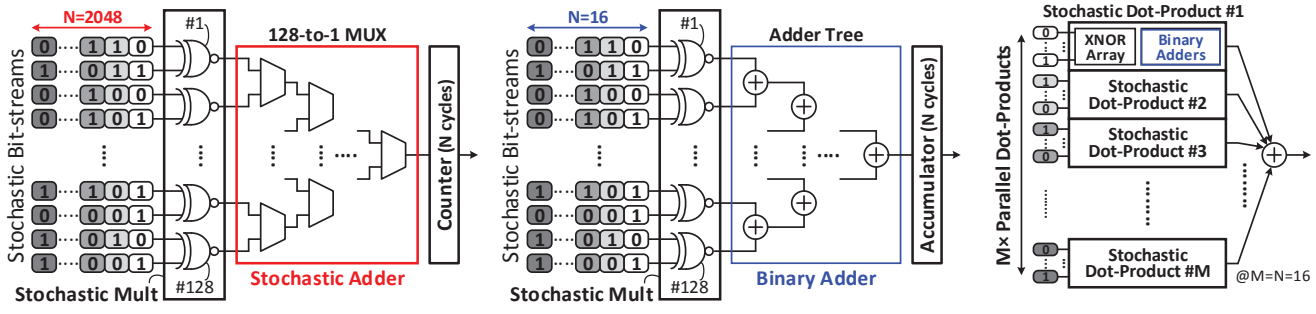


Fig. 2. A 128-input dot-product circuit built with (a) baseline stochastic multipliers/adders (left) and the proposed hybrid stochastic multipliers and binary adders (right) (b) parallelism using M× parallel stochastic dot-product circuits (M=16 in this work).

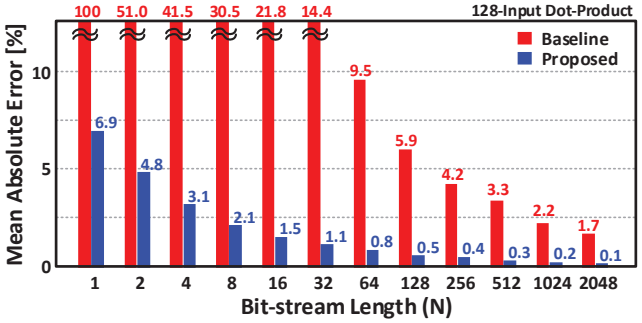
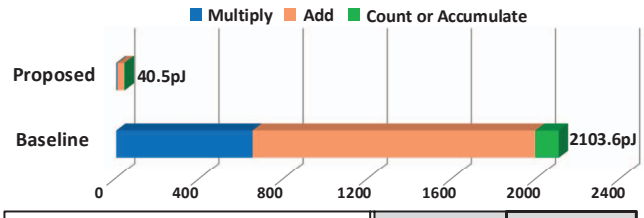


Fig. 3. Simulated mean absolute error vs. bit-stream length.

streams between 0 (when all 0's) and 1 (when all 1's). On the other hand, an XNOR gate is used for a bipolar multiplication of bit-streams with probabilities between -1 (when all 0's) and +1 (when all 1's). Note that both inputs for the multipliers have to be random and uncorrelated. Now, the stochastic addition is performed by using either OR gate or MUX. Both candidates suffer from significant errors due to the probabilistic operation itself (OR) and missing information while selecting an output from two possible input choices (MUX). Hence, the number of stochastic bits (i.e., bit-stream length) has to be large enough to minimize the error by accumulating the individual computation results. Typically, the required bit-stream length is in the range of several hundreds to thousands to achieve an acceptable level of error [1]. Due to such inefficiency, stochastic computing has not been considered as a method for efficient DNN processing. In this work, we resolve the long latency issue by replacing the stochastic adder with a conventional binary adder. The XNOR gates are still used as stochastic multipliers to keep the overall energy consumption and hardware complexity low.

Fig. 2(a) shows two different 128-input dot-product circuits implemented with MUX-based stochastic adders (baseline) and binary adders (this work). Both dot-product circuits have 128× frontend XNOR gates as multipliers for 128× input pairs. Once the multiply operations are done, a 128-to-1 MUX of baseline selects only one from 128× XNOR outputs. Hence, it takes 128 cycles for an arbitrary XNOR gate output to be selected in the



	Baseline	Proposed
Stochastic Multiply: XNOR (baseline/proposed)	647.5pJ	5.1pJ
Stochastic Add (baseline) / Binary Add (proposed)	1344.7pJ	33.2pJ
10b CNT (baseline) / 10b ACC (proposed)	111.4pJ	2.2pJ
Mean Absolute Error @ Bit-stream Length (N)	1.7% @ N=2048	1.5% @ N=16
Total Energy / 128-Input Dot-Product	2103.6pJ	40.5pJ

*Simulated (65nm, TT, 0.8V, 50°C, 100MHz) **Energy: XNOR(2.47fJ), MUX(5.17fJ), FA(8.4fJ)

Fig. 4. Simulated energy breakdown and comparison per dot-product when the operating clock frequency is 100MHz and mean absolute error is 1.7% (at N=2048, baseline) or 1.5% (at N=16, proposed).

average. In other words, most of the XNOR outputs (127 out of 128 or 99.2%) are not used, while still consuming significant energy. On the other hand, the proposed stochastic-binary dot-product circuit utilize all 128× XNOR outputs by using binary adders. Therefore, the required bit-stream length to achieve the similar computation error is reduced by 128× (i.e., as much as the number of inputs) since a cycle of the proposed dot-product is equivalent to 128 cycles in baseline operation. Accordingly, the computation energy is also saved as much as the reduced length of bit-streams. Fig. 2(b) describes parallel dot-products which improves the performance of distributed dot-products by combining their results over space. By using both a hybrid dot-product circuit with N× bit-stream length and M× parallel dot-products, a total of 'M/N' dot-products are computed per cycle.

Fig. 3 shows the simulated mean absolute errors from the 128-input dot-product circuits with different bit-stream lengths for both the baseline and the proposed design. As expected, the overall dot-product computation errors have been significantly reduced for the proposed dot-product circuit. Accordingly, the required bit-stream length for achieving a similar level of error is reduced as much as 128×. For instance, the error from the proposed dot-product circuit is 6.9-to-1.5% when N=1-to-16, while the error is 5.9-to-1.7% for the baseline when N=128-to-2048.

Fig. 4 summarized the simulated energy breakdown and the comparison between baseline and proposed dot-products when the average error is similar (i.e., 1.7% at N=2048, baseline and 1.5% at N=16, proposed). The most significant energy saving

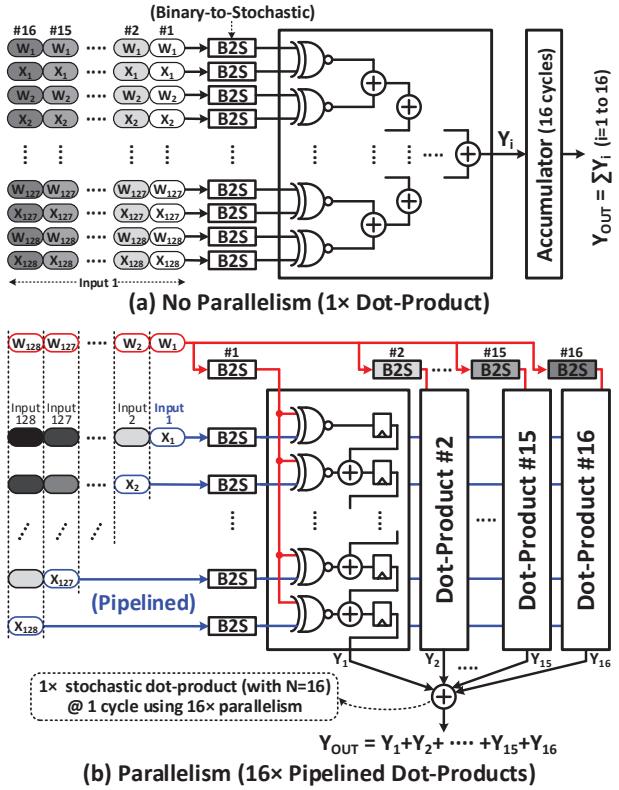


Fig. 5. Two dot-product implementations with/without parallelism.

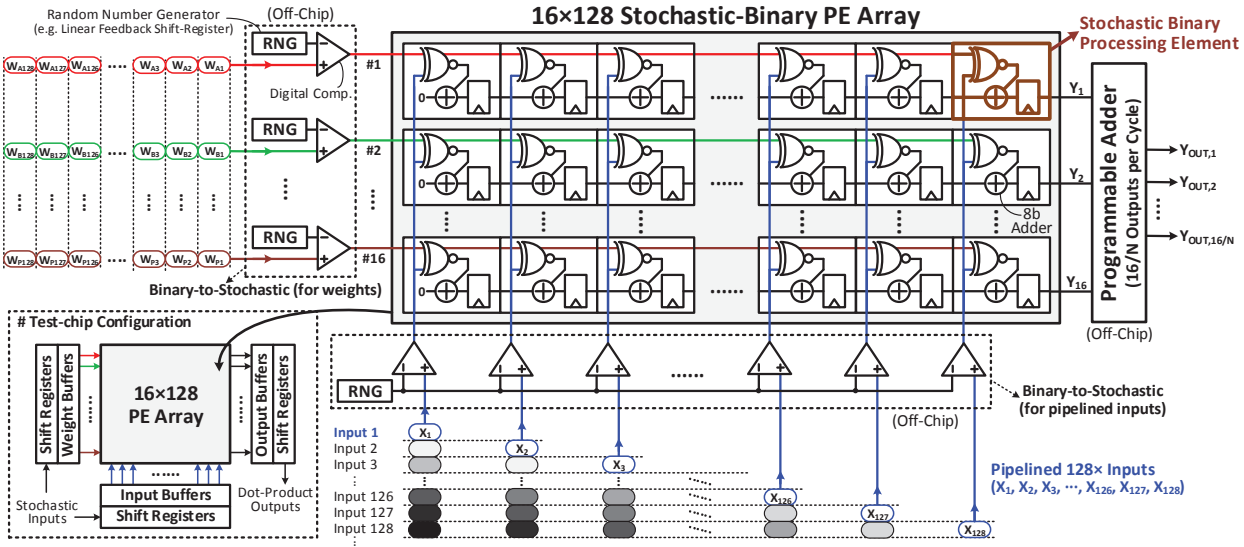


Fig. 6. Proposed 16×128 stochastic-binary PE array for 16× parallel pipelined stochastic dot-product computations.

is from the stochastic multiplication (126.9×), while the binary adders and post-accumulators have the energy savings as much as 40.5× and 50.6×, respectively. The total simulated energy per a 128-input dot-product operation is 40.5pJ at N=16, 0.8V, 50°C, TT, and 100MHz, which is 51.9× smaller than baseline.

To further improve the overall throughput of the proposed dot-product circuits while minimizing the essential overhead in converting the binary inputs to stochastic bit-streams, pipelined dot-product circuits are implemented as shown in Fig. 5. For 16× parallel pipelined dot-product operations, both inputs and weights are rearranged as shown in Fig. 5. Fig. 6 illustrates the overall architecture of the proposed 16×128 stochastic-binary PE array for 16× parallel stochastic dot-product computations. 128× PEs in each row work as a dot-product unit for 128 pairs of inputs and weights. For each dot-product, 128× input stochastic bits are generated from the same number of binary-to-stochastic converters on the bottom of the array while the weights are broadcasted from a single converter on the left to all SPEs in the same row. The inputs are also transmitted column-wise for 16× parallel dot-products. The 16× individual dot-product outputs are combined using the post-adders based on the number of bit-streams (N), and a total of '16/N' combined dot-product outputs are generated off-chip. Note that the number of required random number generators (RNGs) is only 17 (16 for weights and 1 for inputs) thanks to the massive reuse of stochastic weight and input bit-streams based on the proposed parallelism and pipeline architecture.

III. Experimental Results

Fig. 7(a) illustrates a multi-layer perceptron (MLP) network with two hidden layers and an output classification layer. Three different configurations with 1-to-3 layers of stochastic-binary dot-products have been tested as shown in Fig. 7(b). Fig. 8(a) shows the simulated MNIST accuracy, when sweeping N from 1 to 16, and compared with an 8bit binary implementation. The mean accuracy also has been measured by running 16 repeated runs as shown in Fig. 8(b). Fig. 8(c) summarizes the resulting mean accuracies with 1, 2, and 3 layers of the stochastic-binary dot-products using 1K test-images (#7001-8000) from MNIST dataset. The results are 96.66%, 96.81%, and 96.11% for 1, 2, and 3 layers with N=1, 4, and 16, respectively. A 65nm test-chip is fabricated and measured over a range of supply voltage from 0.5V to 0.8V, with different bit-stream lengths as shown in Fig. 9. The measured energy per OP and the energy-efficiency at 0.5V and N=16 are 39.2fJ and 25.5TOPS/W. The energy per dot-product is scalable over supply voltage and the required classification accuracy. For applications with relaxed accuracy requirements, energy efficiency can be improved by reducing the number of bit-streams per computation. Fig. 10 compares the proposed work with state-of-the-art dot-product circuits. Note that the proposed work has achieved the energy and area efficiency, that are comparable to that of an analog in-memory computing work [8] performing analog computations

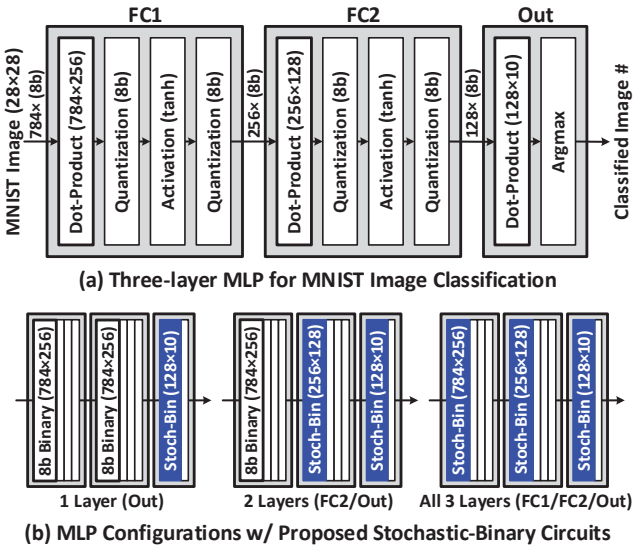
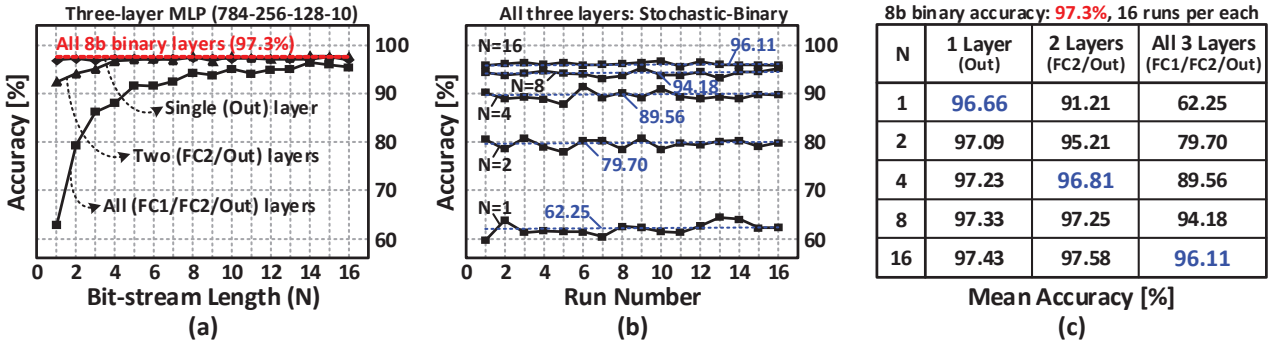


Fig. 7. A three-layer multi-layer perceptron (MLP) network for testing accuracy using the proposed stochastic-binary dot-product circuits.



N	1 Layer (Out)	2 Layers (FC2/Out)	All 3 Layers (FC1/FC2/Out)
1	96.66	91.21	62.25
2	97.09	95.21	79.70
4	97.23	96.81	89.56
8	97.33	97.25	94.18
16	97.43	97.58	96.11

Fig. 8. Classification accuracy with the proposed dot-product circuits for MNIST dataset with 1K images (from #7001-#8000): (a) Accuracy vs. stochastic bit-stream length (N=1 to 16) (b) Accuracy vs. transient run numbers with fixed N=1, 2, 4, 8, and 16 (c) Summary table.

with lower bit-precisions. While achieving a similar level of energy and area efficiency, the proposed stochastic-binary dot-product solution is from critical issues such as PVT variations and device/supply noise (i.e., the fundamental limits of analog computing). The fabricated 65nm test-chip die micrograph and a summary table are shown in Fig. 11.

IV. Conclusion

This work presents 16× parallel 128-input stochastic-binary dot-product circuits based on stochastic multipliers and binary adders. Unlike previous stochastic computing works, this work focuses on the reduction of dot-product computation errors by avoiding the use of popular stochastic adders such as MUX and OR logic gate. By using the proposed dot-product circuit, the mean absolute error of 1.5% has been achieved when the bit-stream length (N) is 16. The proposed dot-product circuit is used to classify MNIST images using a three-layer MLP and achieved 96.11% accuracy (i.e., which is only 1.19% lower vs. 8bit binary) at N=16. A 65nm test-chip is fabricated, and the measured energy per dot-product is 0.63-to-10.04pJ, and the energy efficiency is 407.6-to-25.5TOPS/W at N=1-to-16.

References

[1] A. Ren, J. Li, Z. Li, C. Ding, X. Qian, Q. Qiu, B. Yuan, and Y. Wang, "SC-DCNN: Highly-Scalable Deep Convolutional Neural Network using Stochastic Computing," ASPLOS, Apr. 2017.

[2] A. Alaghi, W. Qian, and J. Hayes, "The Promise and Challenge of Stochastic Computing," IEEE TCAD, vol. 37, no. 8, Aug. 2018.

[3] B. Moons, and M. Verhelst, "Energy-Efficiency and Accuracy of Stochastic Computing Circuits in Emerging Technologies," IEEE JETCAS, vol. 4, no. 4, Dec. 2014.

[4] A. Ardakani, F. Leduc-Primeau, N. Onizawa, T. Hanyu, and W. Gross, "VLSI Implementation of Deep Neural Network Using Integral Stochastic Computing," IEEE TVLSI, vol. 25, no. 10, Oct. 2017.

[5] K. Kim, J. Kim, J. Yu, J. Seo, J. Lee, and K. Choi, "Dynamic Energy-Accuracy Trade-off Using Stochastic Computing in Deep Neural Networks," DAC, Jun. 2016.

[6] V. Lee, A. Alaghi, J. Hayes, V. Sathe, and L. Ceze, "Energy-Efficient Hybrid Stochastic-Binary Neural Networks for Near-Sensor Computing," DATE, Mar. 2017.

[7] D. Bankman, and B. Murmann, "An 8-bit, 16 input, 3.2 pJ/op Switched-Capacitor Dot Product Circuit in 28-nm FDSOI CMOS," IEEE ASSCC, Nov. 2016.

[8] A. Biswas, and A. Chandrakasan, "CONV-SRAM: An Energy-Efficient SRAM With In-Memory Dot-Product Computation for Low-Power Convolutional Neural Networks," IEEE JSSC, vol. 54, no. 1, 2019.

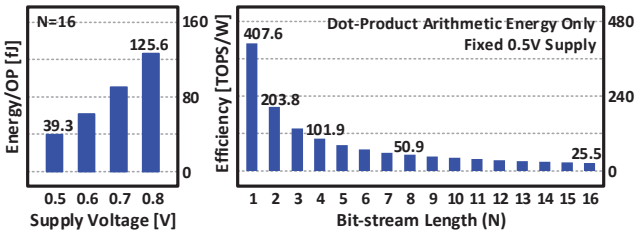
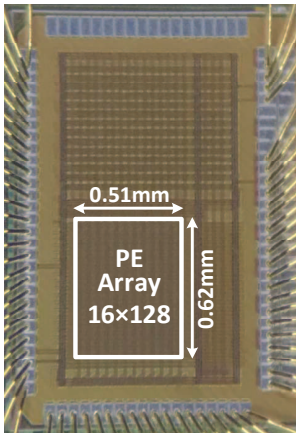


Fig. 9. Measured energy and energy efficiency.

	[7] ASSCC'16	[8] JSSC'19	Proposed
Computing Type	Analog (Deterministic)	Analog (Deterministic)	Digital (Stochastic)
Technology	28nm	65nm	65nm
Precision Control	Fixed (8b)	Fixed (6b/1b)	Reconfigurable
MAC Circuit Type	Analog	Analog In-Mem.	Digital
ADC/DAC Overhead	Embedded	Required	Not Required
Parallelism	No	16×	16×
Energy Efficiency	9.61TOPS/W	51.3TOPS/W	25.5TOPS/W @[D]N=16
Energy FoM[A]	208fJ	39.0fJ	39.3fJ @[D]N=16
Area FoM[B]	720.4μm^2	61.5μm^2	154.4μm^2
Accuracy[C]	N/A	98% (CNN/4-layers)	96.1% @[D]N=16 (MLP/3-layers)

[A] Energy FoM=Energy/(# of inputs)×(# of dot-products)
[B] Area FoM=Area/(# of inputs)×(# of dot-products) [C] MNIST dataset [D] Bit-stream length

Fig. 10. Comparison with energy/area-efficient dot-product circuits.



Process	65nm
Core Supply	0.5-0.8V
Processing Element	XNOR (Stochastic Mult) 8bit Adder/Register
Test-Chip Description	Stochastic-Binary PE Array
PE Array Size	16×128
Core Area	0.51×0.62mm^2
# Bit-Streams	N=1-to-16
Dot-Product Error	6.9-1.5% @N=1-16 (Average Absolute Error)
MNIST Mean Accuracy*	62.25-96.11% @N=1-16 vs. Ref: 97.3% (8b binary)
Dot-Product Energy	0.63-10.04pJ @N=1-16 @ 128-input dot-product
Dot-Product Area	0.0198mm^2 @ 128-input dot-product
Efficiency	408-26TOPS/W @N=1-16

*All three-layers (FC1/FC2/Out) are stochastic-binary

Fig. 11. Die micrograph and summary table.