A 1-16b Precision Reconfigurable Digital In-Memory Computing Macro Featuring Column-MAC Architecture and Bit-Serial Computation

Hyunjoon Kim, Qian Chen, 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: {kimh0003, e170029}@e.ntu.edu.sg, {tgyoo, thkim, bjkim}@ntu.edu.sg

***Abstract* - This work proposes a digital in-memory computing macro with 1-16b reconfigurable weight and input bit-precisions for energy-efficient DNN processing. The proposed digital macro comprises 128×128 bitcells, and each bitcell consists of three building blocks for in-memory computing, an XNOR-based bitwise multiplier, a full-adder, and an SRAM cell. The two-dimensional bitcell array is then divided into parallel neurons, each with 128× column-shape multiply-and-accumulate (column-MAC) units arranged in a row. Each column-MAC with N-bit variable weight precision is built with 'N+7' bitcells in a column (i.e., 8-to-23 bitcells at 1-to-16bit). The N-bit weights are stored at SRAM cells for in-memory computing with the minimal memory access for fetching weights. The remaining 7 bitcells are needed to extend MSBs for accumulating partial-sums through 128 column-MACs. A bit-serial input is broadcasted to all bitcells in the same column, and parallel bitwise multiply operations are performed. Bitwise multiplied results from each column-MAC are then accumulated using N+7 full-adders which are vertically connected to work as a ripple carry adder. Meanwhile, the input precision is determined by the number of bit-serial input cycles from LSB to MSB. Hence, the post-accumulation is required for multi-bit input precision. A 65nm test-chip is fabricated, and the measured energy-efficiency is 117.3 to 2.06TOPS/W at 1-16bit.**

Keywords - multiply and accumulate, deep neural network, in-memory computing, reconfigurable accelerator, dot-product

I. Introduction

Recently, a variety of mixed-signal and digital accelerators have been developed for energy-efficient deep neural network (DNN) processing with innovations in circuit techniques and memory/processing architectures. While mixed-signal circuits demonstrate excellent energy-efficiency with a small hardware footprint, the utilization is limited due to the low bit-precision, and the performance is susceptible to noise and PVT variation. Digital implementations, on the other hand, are typically less energy/area efficient compared to the mixed-signal counterpart, but they are scalable and tolerant to noise and variation. In this work, we develop a customized digital accelerator with energy and area efficiency comparable to that of analog accelerators while taking advantage of conventional digital accelerators.

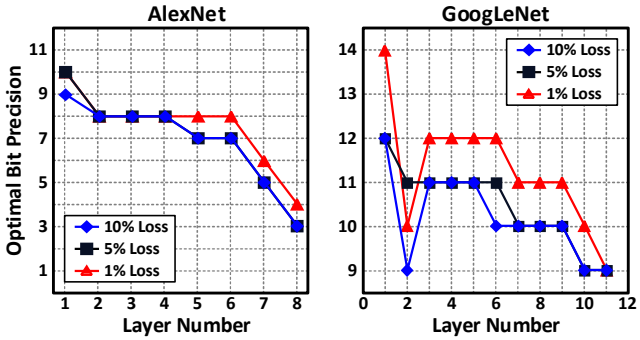

Fig. 1. Optimal bit-precision per each layer for energy-efficient DNN processing under different levels of accuracy loss requirements [5].

Recent advances in reconfigurable architectures [1-2] have enabled fine-grained bit-precision controls in arithmetic logic operations. Based on layer-by-layer bit-precision controls, the energy-efficiency of DNN accelerators can be optimized for a range of accuracy requirements, as shown in Fig. 1. Previously reported digital accelerators [3-7] demonstrated reconfigurable bit-precisions, but they provide a limited set of bit-precisions while having a significant energy/area overhead. In this work, we propose a bit-level fine-grained precision reconfigurable in-memory digital computing macro for achieving highly flexible bit-precisions and the optimal energy and area efficiency. The customized unit cells (i.e., bitcells) are grouped as a column and work as a multiply-and-accumulate (MAC) unit.

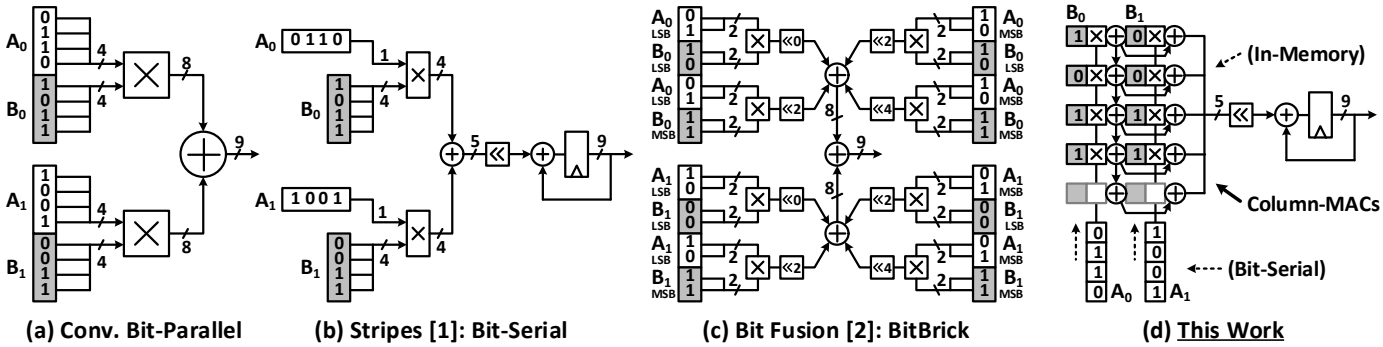

Fig. 2. Comparison of arithmetic logic unit (ALU) architectures with bit-parallel/serial computations and reconfigurable bit-precisions. Two sets of 4-bit input pairs are used for comparison.

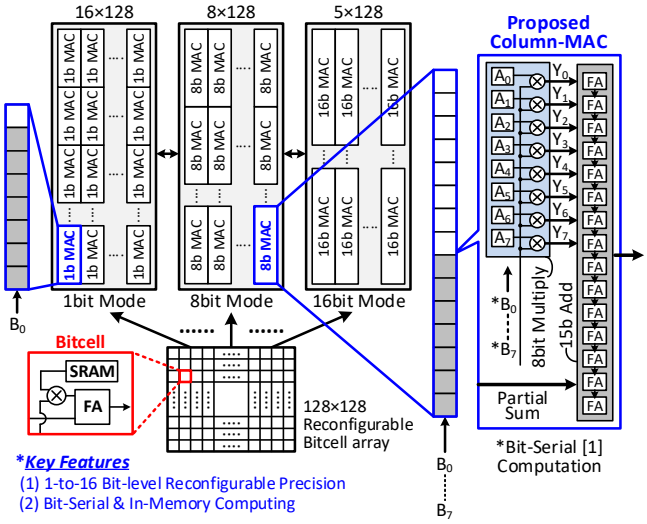

Fig. 3. Proposed 1-16bit reconfigurable in-memory computing macro using column-MAC architecture and bit-serial computation.

II. Proposed Reconfigurable Column-MAC Architecture

A conventional arithmetic logic unit computes multiply and add operations based on a bit-parallel architecture as shown in Fig. 2(a). Hence, the size of multiplier grows quadratically as bit precision increases. On the other hand, bit-serial computing shown in Fig. 2(b) significantly reduces the multiplier footprint by serializing one of the two multiplier inputs, and hence, the size of a multiplier is now proportional to the bit precision. Fig. 2(c) is a recent arithmetic architecture with fine-grained bit-precision control. In this work, we first take advantage of the bit-serial computation, since it offers the highest flexibility in the arithmetic bit-precision while not increasing the footprint too much over a wide bit-precision range (1-16bit). Besides, we have achieved the fine bit-precision step (1bit) by using the smallest hardware. While [2] achieved 1 or 2bit precision step using six adders with >10 logic gates, this work achieves a 1bit resolution step with only a full adder and an XNOR logic gate.

Fig. 3 shows the overall architecture of the proposed work. The proposed digital in-memory computing macro consists of

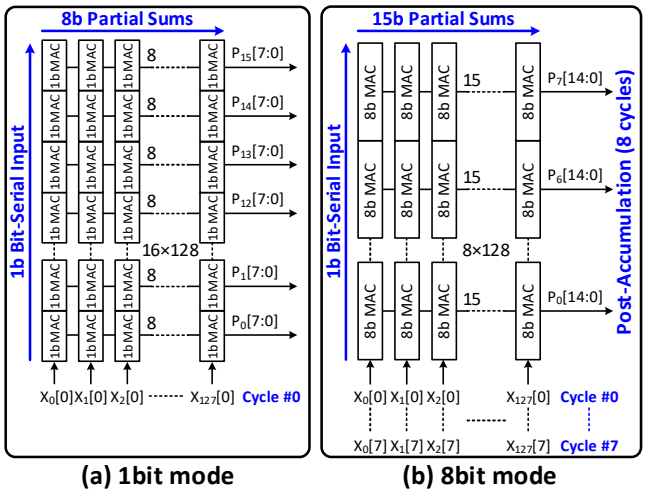

Fig. 4. Reconfigured column-MAC examples: (a) 1bit (b) 8bit mode.

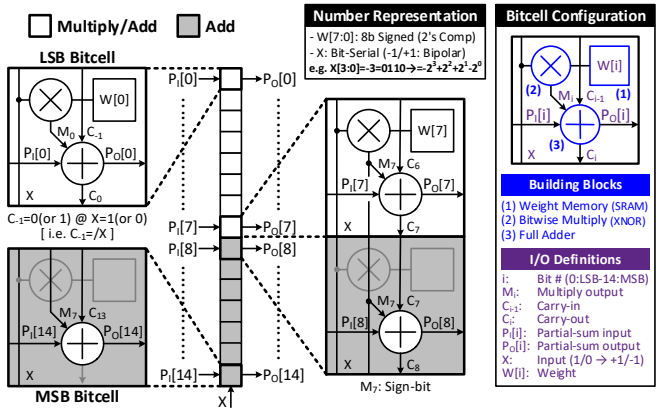

Fig. 5. An example of 8bit column-MAC with detailed reconfigurable bitcell configurations.

128×128 reconfigurable bitcells. A compact bitcell comprising of a 6T SRAM cell, a bitwise multiplier (XNOR), and a full-adder can be reconfigured to a building block of the proposed column-shape MAC (i.e., column-MAC) with a programmable weight precision from 1 to 16bit. For example, a column-MAC with 8bit weight precision is built with 15× bitcells, where 8× XNORs in the top eight bitcells work as an 8b×1b multiplier for bit-serial computation and 15× full-adders from all bitcells in the column-MAC work as a 15bit ripple carry adder. In this way, the proposed 128×128 bitcell array can be reconfigured to an array of N-bit MACs (N+7 bitcells per MAC), resulting in a range of column-MAC array size from the largest (16×128) at 1bit to the smallest (5×128) at 16bit mode. Note that hardware has been only increased by 2.875× for 16bit (vs. 1bit) mode.

Fig. 4 describes 1bit and 8bit examples of the reconfigured column-MAC array. Since N is 1 and 8 for each example, the number of bitcells per MAC is 8 and 15, respectively. Hence, the column-MAC array size is 16×128 and 8×128 for 1bit and 8bit. Now, bit-serial inputs are transmitted to all MACs in the same column. After N cycles of parallel multiply and partial-sum accumulate operations, the output partial-sums are shifted to left by N (e.g., N=0 to 7 at 8bit) and summed up at the post accumulator (Fig. 9).

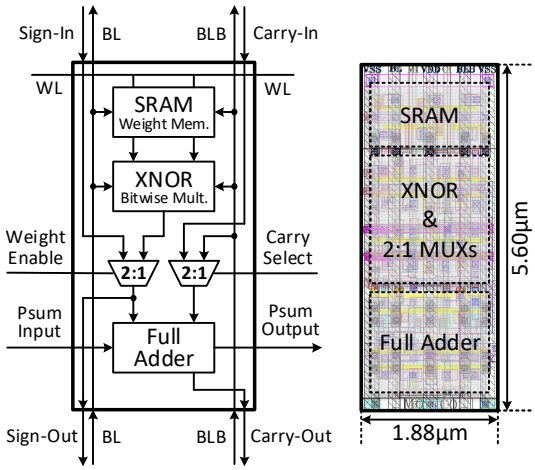

Fig. 6. Bitcell schematic and layout.

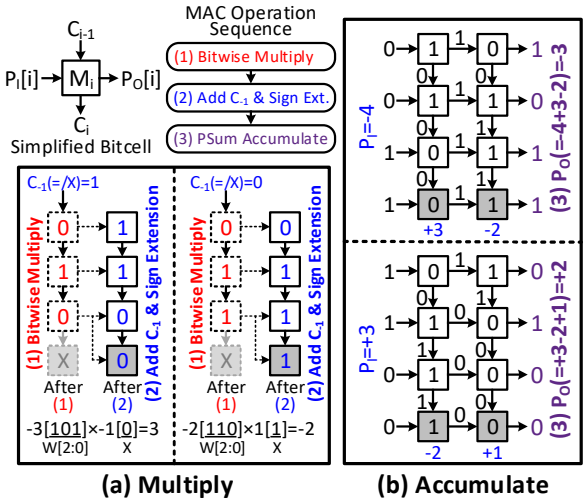

Fig. 7. Operating sequence of 1×2 column-MAC with 3b weight.

Fig. 5 describes a detailed configuration of an 8bit column-MAC comprising of 15× bitcells. A bitcell configuration with essential building blocks and I/O descriptions are summarized in Fig. 5 (right). Note that carry-in/out moves from the upper to the lower cells to form a "column-shape" MAC, while partial-sum input/output moves from left to right for accumulation in a "horizontal" neuron.

In terms of number representation, the weight is stored as a two's complement signed number while the bit-serial input is encoded to a binary-weighted signed number, with the basis of -1 and +1. For instance, the encoded 4bit serial input '0110' represents '-3' since '$-3=-2^3+2^2+2^1-2^0$', where 0 represents -1 and 1 represents +1. For each bit-serial input cycle, the encoded binary input is multiplied by the individual bit of the 8bit signed weight stored in the top eight bitcells (i.e., bitwise multiplication). Then, the bitwise multiply results are delivered to full-adders in the same bitcells. Two additional steps are required for completing the 8bit bit-serial multiplication. First, the sign bit (i.e., M_7) is extended for converting the multiplier outputs from 8 to 15bit signed number. Second, the LSB carry-in (C_{-1}) is connected to an inverted bit-serial input for two's complement inversion when the input is 0 (i.e., -1). After that,

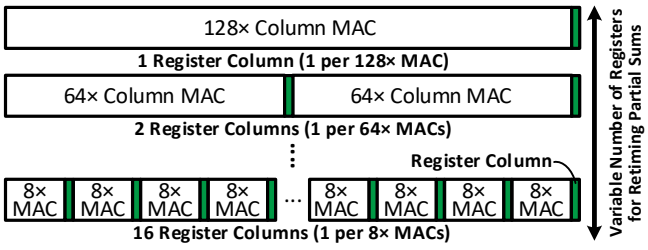

Fig. 8. Retiming partial sums using a variable number of registers.

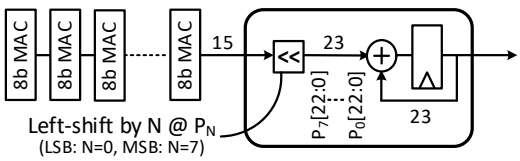

Fig. 9. Post-accumulation for bit-serial computation with 8b MACs.

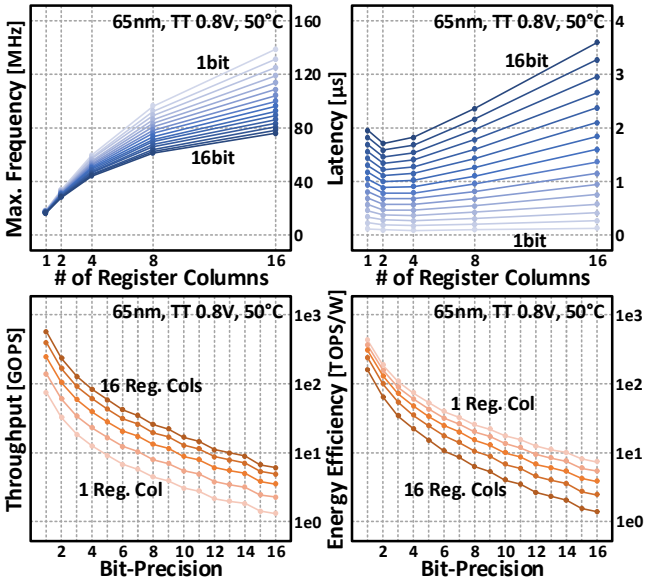

Fig. 10. Performance vs. bit-precision (for both weight and input).

the output 15bit signed number is added to a 15bit partial sum input coming from the left, and the resulting partial sum output is passed to the column MAC on the right.

Fig. 6 shows the bitcell schematic and layout. Two 2-to-1 multiplexers with selection signals are added for programming all the bitcells in the same row. An XNOR gate of each bitcell is enabled for the bitwise multiplication when 'weight-enable' is high, while a carry-out from the upper bitcell is selected as a carry-in for the current bitcell when 'carry-select' is high. An example of bit-serial multiplication between a 3bit weight (e.g., 3b'101 or -3) and a 1bit input (e.g., 1b'0 or -1) is illustrated in Fig. 7(a). After bitwise multiplications, the MSB sign bit '0' is extended to the 4th bit, and the LSB carry-in is 1. Hence, the multiplier output becomes '0011'. Similarly, a multiplication between -2 and +1 results in the output '1110'. The partial-sum accumulation examples based on two 3bit column MACs are shown in Fig. 7(b). The accumulated partial-sums are retimed using a variable number (1, 2, 4, 8, and 16) of register columns as shown in Fig. 8. The more frequent retiming operations for partial-sums result in the higher operating frequency. Hence, the throughput will increase, while the latency and energy efficiency decrease. Fig. 9 shows a post-accumulator which is used for combining the individual 15bit partial-sums from each eight bit-serial input cycles at 8bit weight/input precision.

Reconfigurability	1-16b Weight / 1-16b Input			
Bitcell & Register	128×128 Bitcell Array with 16 Register Columns			
Bit-Precision**	1b/1b	1b/16b	16b/1b	16b/16b
Operation Cycles	1	16	1	16
Column MACs	16×128	16×128	5×128	5×128
Max Frequency	138MHz	138MHz	75.8MHz	75.8MHz
Latency	0.12µs	1.92µs	0.22µs	3.59µs
Throughput	567GOPS	35.4GOPS	97GOPS	6.1GOPS
Energy Efficiency	156TOPS/W	9.7TOPS/W	22TOPS/W	1.4TOPS/W

*Simulated (65nm, TT, 0.8V, 50°C) **Bit-precision setting (Weight/Input)

Table I. Performance summary with different precision settings.

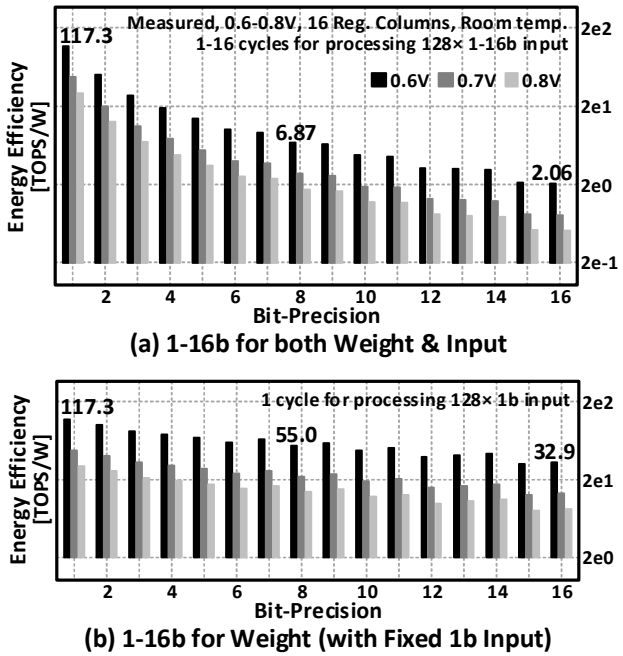

Fig. 11. Measured energy efficiency: (a) 1-16b for both weight and input (b) 1-16b weight with fixed 1bit input.

III. Experimental Results

The performance of the proposed macro including latency, throughput, and energy efficiency versus the reconfigured bit-precision is shown in Fig. 10, and summarized in Table I. The maximum operating frequency of the column-MAC array with 16× register columns is 75.8MHz at 16bit and 138MHz at 1bit mode. Latency is the function of the operating frequency with the number of register columns and bit-serial operation cycles, as shown in its plot and summary (Fig. 10 and Table I). The throughput/energy-efficiency is 567GOPS and 156TOPS/W at 1bit and 6.1GOPS and 1.4TOPS/W at 16bit mode. Meanwhile, a binary weight (or input) mode with 16bit input (or weight) results in the better throughput/energy-efficiency compared to the case when both weights and inputs are 16bit. Fig. 11(a) and (b) shows the measured energy efficiency with bit-precision for both weight/input and weight-only (with fixed 1bit input). The measured energy efficiency at 0.6V supply ranges from 117.3 to 2.06TOPS/W at 1-16 input/weight bit-precision. Meanwhile, the efficiency increases to 32.9TOPS/W at 16bit weight, while the input is fixed to 1bit. Performance comparison with prior reconfigurable accelerators is summarized in Table II, and the fabricated 65nm test-chip die micrograph is shown in Fig. 12.

	[3]Envision ISSCC'17	[5]UNPU ISSCC'18	[7] VLSI'18	This Work
Technology	28nm	65nm	14nm	**65nm**
Supply Voltage	0.65-1.1V	0.63-1.1V	0.28-0.9V	**0.6-0.8V**
Multiply Precision	1-16/N bit (N=1,2,4)	1-to-16bit	FP16b INT8/16b	**1-to-16bit**
Accumulate Precision	48/N bit	32bit	FP32b INT24/48b	**8-to-23bit**
Reconfigurability	Reconfig. Multiplier	Bit-Serial	Fixed Bits (8,16,24,48)	**Column MAC Bit-Serial**
MAC Array	N×256	12×12	4×4	**16×128(1b) 5×128(16b)**
MAC Area [μm^2]	N/A	N/A	1480	**84.2(1b) 242.1(16b)**
Energy per MAC [pJ/MAC]	N/A	0.055(1b) 1.26(16b)	N/A	**0.017(1b) 0.78(16b)**
Min. Energy Eff. [TOPS/W]	0.26	3.08(16b)	0.55(16b)	**2.06(16b)**
Max. Energy Eff. [TOPS/W]	10	50.6(1b)	11.3(8b)	**117.3(1b)**

Table II. Comparison with prior reconfigurable accelerator works.

IV. Conclusion

This paper presents a 1-16b precision reconfigurable digital in-memory computing macro for energy-efficient computation of DNNs. The weight bit-precision is reconfigured from 1 to 16 with minimal hardware overhead thanks to the novel column-MAC architecture based on the bit-serial computation method. In addition to the reconfigurable weight bit-precision, the input precision is also programmed based on the bit-serial operation. A test-chip is fabricated using 65nm CMOS technology, and the measured energy efficiency is 117.3 to 2.06TOPS/W at 1-16bit weight/input precision under 0.6V supply.

References

[1] P. Judd, J. Albericio, T. Hetherington, T. Aamodt, and A. Moshovos, "Stripes: Bit-serial deep neural network computing," in 49th IEEE/ACM International Symposium on Microarchitecture (MICRO), 2016.

[2] H. Sharma, J. Park, N. Suda, L. Lai, B. Chau, V. Chandra, and H. Esmaeilzadeh, "Bit Fusion: Bit-Level Dynamically Composable Architecture for Accelerating Deep Neural Networks," in 45th ACM/IEEE International Symposium on. Computer Architecture (ISCA), 2018.

[3] B. Moons, R. Uytterhoeven, W. Dehaene, and M. Verhelst, "Envision: A 0.26-to-10 TOPS/W subword-parallel dynamic-voltage-accuracy frequency-scalable convolutional neural network processor 28 nm FDSOI," in IEEE Int. Solid-State Circuits Conf. (ISSCC), Feb. 2017.

[4] D. Shin, J. Lee, J. Lee, and H.-J. Yoo, "DNPU: An 8.1 TOPS/W reconfigurable CNN-RNN processor for general-purpose deep neural networks," in IEEE Int. Solid-State Circuits Conf. (ISSCC), Feb. 2017.

[5] J. Lee, C. Kim, S. Kang, D. Shin, S. Kim, and H. Yoo, "UNPU: A 50.6TOPS/W Unified Deep Neural Network Accelerator with 1b-to-16b Fully-Variable Weight Bit-Precision," in IEEE Int. Solid-State Circuits Conf. (ISSCC), Feb. 2018.

[6] B. Fleischer, S. Shukla, M. Ziegler, et al., "A Scalable Multi-TeraOPS Deep Learning Processor Core for AI Training and Inference," in Proc. IEEE Symp. VLSI Circuits (VLSI-Circuits), Jun. 2018.

[7] M. Anders, H. Kaul, S. Mathew, V. Suresh, S. Satpathy, A. Agarwal, S. Hsu, and R. Krishnamurthy, "2.9TOPS/W Reconfigurable Dense/Sparse Matrix-Multiply Accelerator with Unified INT8/INT16/FP16 Datapath in 14nm Tri-gate CMOS," in Proc. IEEE Symp. VLSI Circuits (VLSI-Circuits), Jun. 2018.

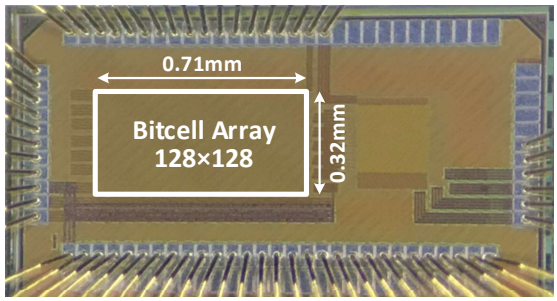

Fig. 12. A 65nm test-chip die micrograph.