A 90nm 103.14 TOPS/W Binary-Weight Spiking Neural Network CMOS ASIC for Real-Time Object Classification

Po-Yao Chuang[1], Pai-Yu Tan[1], Cheng-Wen Wu[1,2], and Juin-Ming Lu[1,3]

[1]*Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan*
[2]*Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan*
[3]*Industrial Technology Research Institute, Hsinchu, Taiwan*

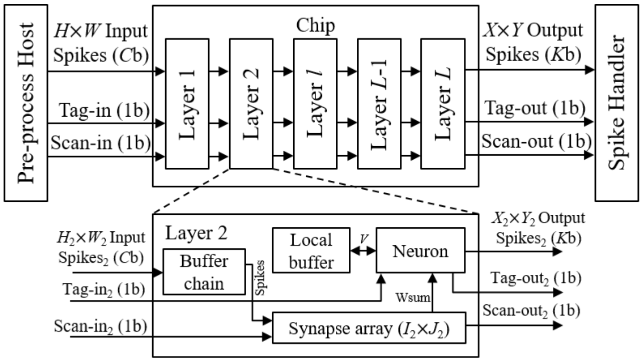
***Abstract*—This paper introduces a low-power 90nm CMOS binary weight spiking neural network (BW-SNN) ASIC for real-time image classification. The chip maximizes data reuse through systolic arrays that house the entire 5-layer BW-SNN, requiring a minimum off-chip bandwidth for data access. The chip achieves 97.57% accuracy for real-time bottled-drink recognition, consuming only 0.62uJ per inference. For comparison purpose, it achieves 98.73% accuracy for MNIST hand-written character recognition, consuming only 0.59uJ per inference. The bottled-drink recognition is demonstrated at 300 fps that is well enough for many other real-time applications. The peak efficiency point is 103.14TOPS/W at a voltage of 0.6V, which outperforms other designs so far as we know. By normalizing to the 28nm technology node, the proposed ASIC is about 5× more efficient and 7× lower hardware cost as compared with the state-of-the-art designs.**

***Index Terms*—AI, convolution neural network, image classification, machine learning, spiking neural network, systolic array**

I. Introduction

The fast growing adoption of *deep neural network* (DNN) for applications like image recognition/classification, natural language processing, gaming, etc., is calling for ever more power-efficient acceleration hardware in recent years. Power-efficient DNN inference chips clearly are required by many emerging applications, such as robotics, end-point devices of IOT systems, mobile communications, etc. Existing high-performance AI chips are mostly for servers, thus are costly in terms of power consumption and hardware complexity. Recently, there are works on scalable *Convolution Neural Network* (CNN) processors, for handling a group of different applications, e.g., [1] and [2], and others involve in-memory processing approaches, e.g., [3]. However, for application-specific inference engine, they still consume too much power. To reduce hardware cost, it has been known that CNN nodes can be scaled down to accommodate only binary or ternary weights, sacrificing the accuracy to a certain degree as a tradeoff. In contrast to CNN, *spiking neural network* (SNN) algorithms uses *membrane potential* (V) to retain the neuron information while minimizing its inputs. The IBM TrueNorth has shown some benefits in hardware design, especially its low power consumption [4].

This paper aims to design a power-efficient application-specific SNN hardware for CNN structures. We use binary-weight SNN (BW-SNN) that has only 1-bit inputs and 1-bit weights, so as to minimize off-chip memory access and on-chip storage, as well as the number of convolution operations, while being able to keep the neuron information by V.



Shape Parameters of a Layer

L	# of conv. layers	H/W	input spikes height/width
C	# of input spike channels	X/Y	output spikes height/width
K	# of filter channels	I/J	filter height/width

Fig. 1: The overall architecture referred from [5] of the bottled-drink classification BW-SNN ASIC.

We use a 2-D systolic array as the basic component of our hardware architecture, which can efficiently process 5-layer convolutional BW-SNN with minimal off-chip memory access, and achieve real-time image classification with better accuracy and much lower power consumption than state-of-the-art designs.

II. Chip Architecture

The architecture of the ASIC is given in Fig. 1 [5], which is a full BW-SNN that consists of L convolutional layers. Each layer is implemented as a *layer module*. The inputs to the chip and a layer module include the spikes from the pre-processing host and previous layer module, respectively. The scan chain in the figure is for entering the configuration data. The tags are for controlling the counting sequences of the *buffer chain* in the layer modules, for the purpose of correct timing and reduced power—eliminating unnecessary operations. The outputs of the chip and layer module are the spikes that go to the *spike handler* of the host and the next

978-1-7281-1085-1/20/$31.00 ©2020 IEEE

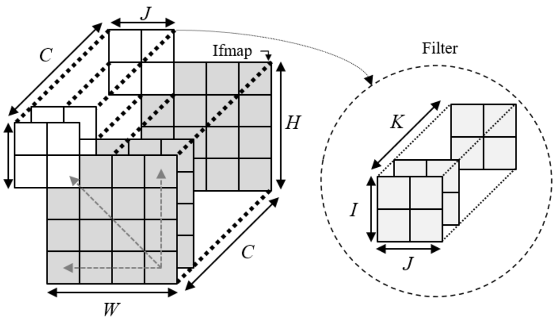


Fig. 2: Six dimensional convolution.

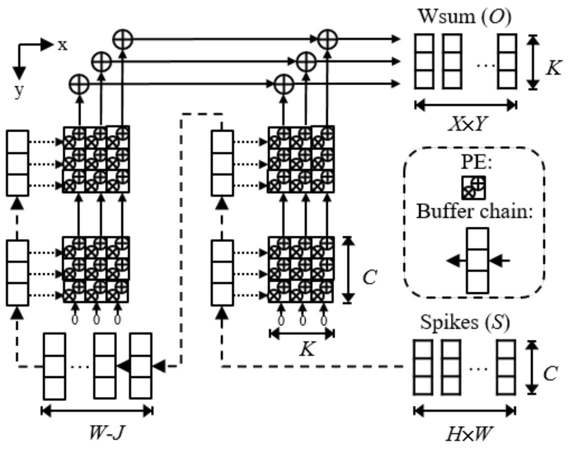


Fig. 3: The design of the synapse array.

level layer module. The shape parameters of a layer module, are listed in Fig. 1, which need to be adjusted to satisfy the model requirement according to its target application. A layer module mainly consists of a $synapse\ array$ and a $neuron\ module$. Basically, the synapse array implements a binary-weight convolution of 1-bit inputs, and the neuron module generates output spikes by comparing the membrane potential (V) and the $firing\ threshold$ (V_{th}). Details will be discussed next.

A. Systolic Synapse Array

To accelerate the computation of a 6-D convolution as shown in Fig. 2, a specific systolic array for low-power design had been introduced in [5]. It is able to minimize the data access from off-chip memories. In Fig 3, we show that the systolic array provides a scheduled data sequence, which properly handles the systolic interaction between two different data, $input\ feature\ map$ (ifmap) and $filters$. The systolic array mainly consists of a buffer chain and a 2-D $processor\ element$ (PE) array. The array takes the spikes (S) as the input data stream, and generates the convoluted output weight sums (Wsums) as the output data stream. The buffer chain provides data with correct timing to all the sub-arrays. Part of the chain is to maintain the correct filter window, and the other part is for storing and properly shifting the ifmap. At each clock cycle, the buffer chain simultaneously shifts the ifmap and generates the output feature that is required by the convolution. All the data are carefully scheduled. Once an input feature has been shifted into the buffer chain, all the computations involved will be finished before it travels out of the buffer chain, i.e., the data has been fully utilized to greatly reduce external memory access.

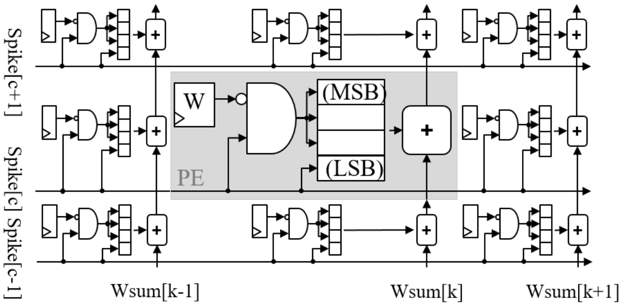


Fig. 4: The 3×3 version of the crossbar design, where the center cell is highlighted.

To handle the correct systolic interactions, the entire computation is mapped to I×J sub-arrays. Each sub-array is a crossbar array, i.e., a K×C array of cells (PEs), where there are C spike inputs (from the previous stage) that are broadcast to all columns, and K weight-sum (Wsum) inputs from the sub-array below (see Fig. 3). A 3×3 version of the crossbar is shown in Fig. 4 as an example. As shown in the figure, each Wsum accumulates its value along the way upward. As shown in the highlighted cell, each PE has a binary weight (W) stored in a 1-bit weight register, i.e., a $flip$-$flop$ (FF), which takes a binary value 0, 1. However, in the SNN model the weight should be -1, +1, so we need to convert the weight in the PE. We design a simple mechanism as shown in the center cell that converts the single-bit binary weight 0, 1 to multiple-bit binary weight -1, +1, which will be added to the partial sum (Psum), i.e., the temporary Wsum, from the cell below, and sent to the cell above. Depending on the number of bits that we represent the Wsum, the weight conversion performs sign extension at the output of the AND gate as shown in the figure. Note that the final converted weight will be -1, 0, 1. All the FFs are connected as a scan chain, so the input weights can be entered easily from the scan chain.

B. Neuron Module

The data inputs of the neuron module are from the outputs of the synapse array, i.e., the Wsums. We need also the control input, i.e., the Tag, as shown in Fig. 5. The Tag is associated with the Wsum input, to signal whether the current input should be processed, i.e., whether the read and write operations from/to the memory (local buffer) should be enabled. When Tag = 1, the input will be processed, otherwise the data will be discarded. This is to avoid processing garbage data that are outside the specified window. In the figure, we can see that the neuron module is partitioned into three pipeline stages, separated by the pipeline registers. Now assume Tag = 1. In the first stage, we read the original (old)

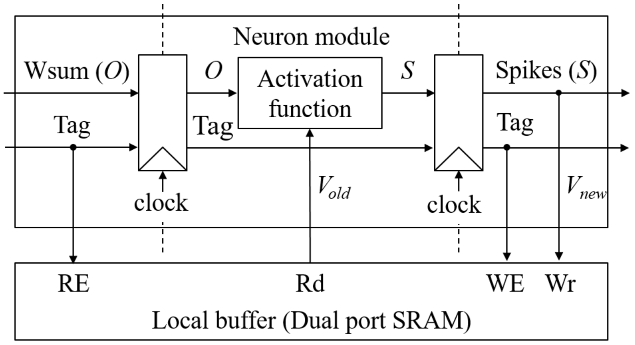


Fig. 5: The 3-stage pipeline design and memory controls in the neuron module.

membrane potential (V_{old}) of the BW-SNN from the local buffer, which is a dual-port SRAM. Then, in the second stage, we perform the $activation\ function$ [6], which is simplified in our case. In the last stage, the updated (new) membrane potentials (V_{new}) are write back to the local buffer. In the BW-SNN, the weights are associated with the synapses, and the neurons store information in terms of the membrane potentials (V). In Fig. 5, the output spike (S) can be derived from the Wsum (O) and the original (old) membrane potential (V_{old}) of the neuron. Note that each neuron has its own bias of the membrane potential (V_{bias}), as well as a neural firing threshold (V_{th}) that determines whether the firing condition is met. Both V_{bias} and V_{th} are predetermined during the training phase, and configured into the neuron modules during the system initialization phase. The simplified activation function can be expressed as follows:

$$\textbf{Integrate: } V(t) = V(t-1) + O(t) + V_{bias}, \tag{1}$$

$$\textbf{Fire: } S(t) = \begin{cases} 1, & \text{if } V(t) \geq V_{th} \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

$$\textbf{Reset: } \text{If } V(t) \geq V_{th}, \text{ then } V(t) = V(t) - V_{th}, \tag{3}$$

The S, O, V, V_{bias}, and V_{th} denote the values of spike, weight sum, membrane potential, bias, and firing threshold, respectively, where $V(t-1) = V_{old}$ and $V_{new} = V(t)$. In the expressions (1) - (3), the activation function of the neurons consists of three main parts, i.e., Integrate, Fire, and Reset, to emulate the behavior of neuron firing [6]. In iteration (time) t, the neuron integrates the Wsum inputs into its membrane potential, as shown in (1). If the updated membrane potential exceeds V_{th}, the neuron generates a spike (see (2)), and resets its membrane potential by subtracting V_{th} (see (3)). Otherwise, no spike is generated and the membrane potential stays the same.

III. Design Optimization and Trade-offs

According to Fig. 1, the hardware complexity is determined by the shape parameters, which also affect some characteristics of the design, e.g., throughput, storage size, bandwidth, etc. In general, when designing an ASIC, we try to minimize the hardware cost and energy consumption, given an acceptable inference accuracy as specified by the application. Therefore, before we design the bottled-drink classification ASIC, we have developed a behavior-level simulation model for the SNN, which is necessary for us to explore the huge search space for an appropriate SNN model. We have explored hundreds of possible SNN models in the search space, represented by different values of the shape parameters, before we picked the most efficient hardware set-up for our chip. An efficient shape parameter configuration not only leads to the minimum hardware cost, but also achieves acceptable accuracy for the bottled-drink classification application. We have also analyzed the shape parameters to identify the most critical parameters for hardware cost and accuracy. There are normally design trade-offs between high-performance and low-power designs, which will be discussed next.

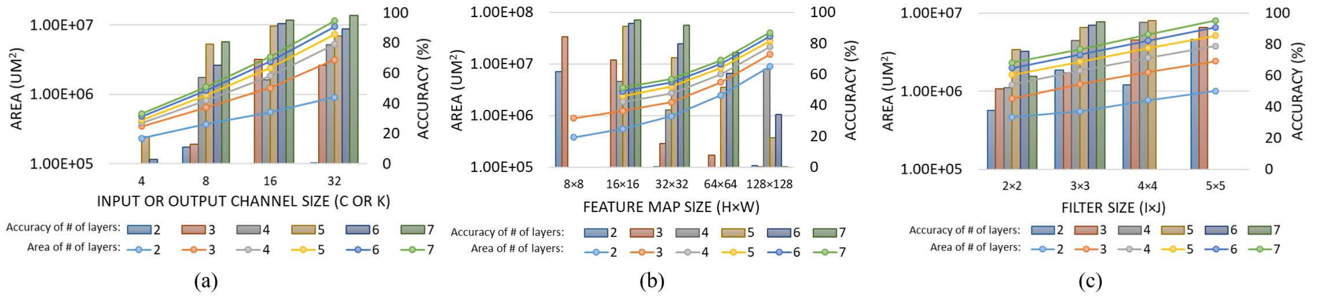
To observe the correlation among the shape parameters, power consumption, and accuracy, we show in Fig. 6 the experimental synthesis results, which are based on a commercial 90nm CMOS technology. In Fig. 6(a), we compare the area and accuracy for different input channel size C and output channel size K, in (b) for different ifmap size (H×W), and in (c) for different filter size (I×J). The impact of the parameters is discussed separately as follows.

(1) Parameters C and K determine the number of PEs in the crossbar sub-array, where C determines the fanout of a broadcast input spike, which in turn affects the load and speed, and K determines the number of accumulation steps of Psum, which in turn affects the latency. The C and K values also determine the required off-chip memory access traffic, which should be as low as possible to save the computation energy, because off-chip data access normally dominates the system power consumption. However, as shown in Fig. 6(a), if C and K are small, the inference accuracy will be low. Therefore, there is a trade-off between power consumption and accuracy.

(2) Parameters H and W determine the height and width of the ifmap, respectively. They affect the length of the buffer chain and the latency of data transmission from the host computer. If H and W are large, the data transmission latency will be long, which may also induce higher energy consumption. However, as shown in Fig. 6(b), if H and W are too small, the accuracy will suffer. Therefore, H and W should be as low as possible, subject to acceptable accuracy constraint.

(3) Parameters I and J determine the height and width of the filter window. In general, expanding the window size can improve the accuracy, but the drawback is that higher I and J values normally will result in a shallower $neural\ network$ (NN) in order to maintain the accuracy. However, as shown in Fig. 6(c), if I and J are large, the step of the systolic array will be complex, which in turn increase the storages for a longer length of buffer chain and induce higher power consumption. Therefore, there is a trade-off among the area, power consumption and accuracy.

In Fig. 7, we show two options for system implementation, where Option 1 is that the chip implements a single layer, and the system is implemented by connecting multiple chips, as shown in Fig. 7(a). In (b), we show Option 2, where

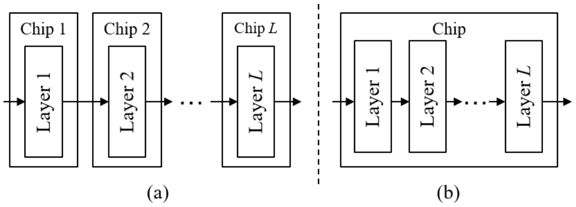


(a)

(b)

(c)

Fig. 6: Comparison of the area overhead (left) and accuracy (right) for different (a) input or output channel size (C or K) when $H = W = 16$ and $I = J = 3$, (b) ifmap size (W×H) when $C = K = 16$ and $I = J = 3$, and (c) filter size (I×J) when $C = K = 16$ and $H = W = 16$, based on a commercial 90nm CMOS technology.



(a)

(b)

Fig. 7: Implementation options: (a) the chip implements a single layer, and the system is implemented by connecting multiple chips, or (b) the chip implements all the layers, i.e., the entire system.

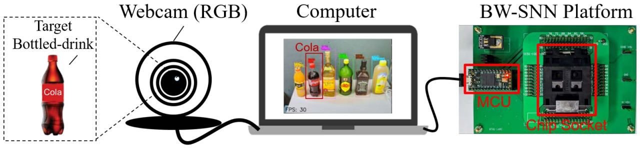


Fig. 8: System demonstration of the bottled-drink classification application.

TABLE I: Layer Shape Parameters for the BW-SNN.

Layer	C	(H, W)	(I, J)	K	(X, Y)
Conv 1	3	16	3	16	14
Conv 2	16	14	3	16	12
Conv 3	16	12	3	16	10
Conv 4	16	10	3	16	8
Conv 5	16	8	3	6	6

the chip implements all the layers, i.e., the entire system. The benefit for Option 1 is that the die size is small, and yield can be higher, so the tape-out cost is low. It also has higher flexibility so far as system configuration is concerned, when used in different applications. However, the constraint is that the system will be implemented by an NN that contains identical layers, so the system cost can be higher. In Option 2, different layers have their own shape parameters, so the redundancy hardware can be minimized, and the entire NN can be fabricated on a small chip. Furthermore, there is no off-chip communication between the layers, so the performance can also be higher.

IV. Chip Implementation

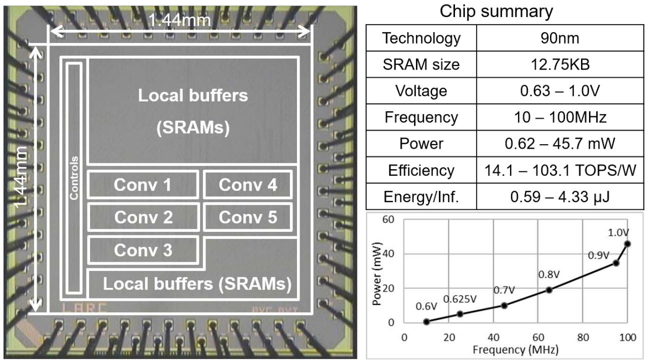
We have implemented a chip for classification of 6 different bottled-drinks, i.e., whisky, tequila, cola, lemon juice, orange juice, and pineapple juice. The chip is part of the camera module for a bartender robot, which can distinguish more than 10 objects in a frame and classify the objects among the 6 classes. In Fig. 8, we show the demonstration environment and the BW-SNN platform that we have developed. The system input is the video that is generated by the webcam. Bottle images in each frame are segmented by YOLOv3 [7] on the computer and sent to the MCU. To maximize streaming efficiency of the bottle images, we scale down the images to 16×16 RGB pixels, pre-process the RGB pixels to a 3-bit (R, B, and G) spike vector, and slow down the image throughput of the ASIC which actually can process up to 300 fps. The pre-process function and the spike accumulation are implemented in the MCU by software that communicates with the ASIC by spike signals. The inference result is sent back to the computer and shown on the screen.

To reduce the circuit complexity and power consumption, we have develped a system simulation platform, based on which we obtained shape parameters that minimize costs while maintaining acceptable accuracy. The parameters are summarized in Tab. I for all the layers, which achieve an accuracy of about 97% for the bottled-drink application on our system simulation platform. The 3-bit input of the ASIC corresponds to the R, G, and B spikes of the current input pixel. The 6-bit output of ASIC represents the 6 output neurons that in turn stand for the 6 bottled-drinks, respectively.

To be able to test the chip, we have inserted design-for-test (DFT) circuits, including a scan-chain and some memory $built\text{-}in\ self\text{-}test$ (BIST) circuits. There are 12,760 FFs that are all connected as a single scan-chain, providing over 98% SAF coverage under just 65 test patterns. The memory BIST modules use the March C+ test algorithm for all the SRAMs. Each SRAM has its own BIST module, but all memories share the same test control signals.

V. Measurement Result

The chip is fabricated in a 90nm CMOS technology. The measurement result of our 5-layer BW-SNN ASIC is shown



Chip summary

Technology	90nm
SRAM size	12.75KB
Voltage	0.63 – 1.0V
Frequency	10 – 100MHz
Power	0.62 – 45.7 mW
Efficiency	14.1 – 103.1 TOPS/W
Energy/Inf.	0.59 – 4.33 μJ

Fig. 9: Chip micrograph and measured power consumption versus operating frequency.

in Fig. 9. In the figure, the measured power across a range of operating clock frequency is plotted from a near-threshold voltage (0.6V) to a normal operation voltage (1V). The highest operating frequency at 0.6V and 1V are 10MHz and 100MHz, where it consumes 625uW and 45.71mW of power, respectively.

The number of iterations that can be executed by the BW-SNN chip within a time unit is dependent on the chip operating frequency. Apparently, a higher operating frequency results in more iterations that can be performed within a time interval. In Fig. 10 we plot the number of iterations per inference under different operating frequencies and their corresponding operating voltages. If we can perform more iterations for an inference, the BW-SNN can achieve a higher accuracy before saturation. In Fig. 11, we show the plot of accuracy versus energy/inference for the bottled-drink classification and MNIST hand-written character recognition problems, with the corresponding operating frequencies. In the figure, we can see that the accuracy grows significantly if we increase energy/inference when the value is low, but it saturates when the value is about 0.6. Beyond that point, most of the energy can be spent on a very small accuracy gain. Therefore, the chip should be operated at a high-efficiency point, e.g., Vdd = 0.6V, f = 10MHz, and 37 iterations per inference. Under this configuration, the ASIC performs 0.62uJ/inf, with 97.57% accuracy for real-time bottled-drink classification.

Figure 12 shows the recognition accuracy degradation when the chip operates below 0.6V, near the MOSFET threshold voltage. In this case, 0.58V is clearly the limit based on our measurement results, below which the system will malfunction.

Table II shows the cost and performance summary and the comparison with some state-of-the-art designs. Our design is based on an efficient systolic-array algorithm that maximizes data reuse and requires no off-chip memory-access. The novel systolic-array approach maps the 6-D data graph of the computation algorithm onto a 2-D array, in addition to the enhanced BW-SNN architecture that minimizes computation overhead. As a result, the proposed BW-SNN ASIC outperforms [8]–[10] in efficiency (TOPS/W), even under an older 90nm process. For the MNIST hand-written character recognition problem, the chip achieves 0.59uJ/inf with 98.01% accuracy. Although [8]–[10] achieve similar accuracy and lower energy as compared with our design, only the FC layers are included, which require less computation than the CONV layers, which greatly limit the applicability to real-world problems. If we only compare the designs supporting CONV layers, such as BinarEye [11], our design achieves a higher accuracy (98.01% versus 97.4%) and a higher inference throughput (1.05k inf/s versus 0.5k inf/s). Moreover, if our

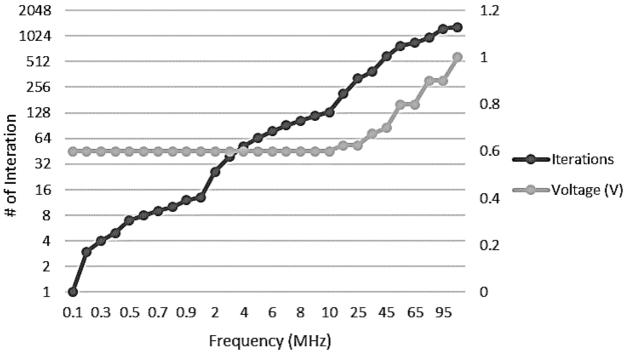


Fig. 10: The number of iterations per inference under different operating frequencies and corresponding operating voltages.

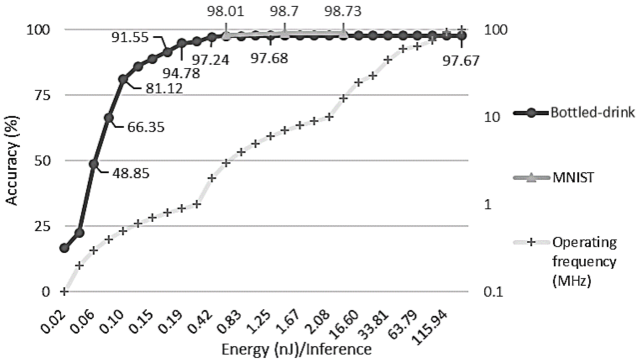


Fig. 11: Accuracy versus Energy/Inf. for the bottled-drink classification and MNIST hand-written character recognition problems, with corresponding operating frequencies.

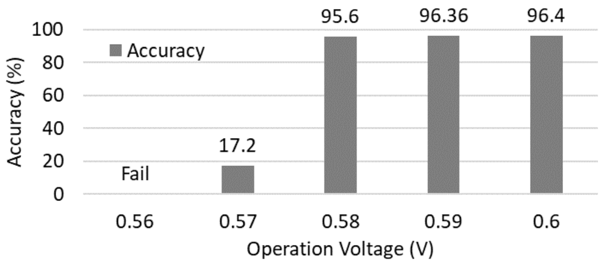


Fig. 12: Accuracy degradation when the chip operates below 0.6V.

TABLE II: Performance summary and comparison with related works for MNIST benchmark. Scaled values are normalized to the 28nm node [12], i.e., power and energy are scaled with $(28/current\_node)$.

Items	This work			Moon CICC'18 [11]			Lee ESSCIRC'18 [8]		Park ISSCC'19 [9]	Chen JSSC'18 [10]	
Technology (nm)	90			28			16		65	10	
Area (mm^2)	2.07			1.4			6.25		10.08	1.72	
Algorithm	5L-CONV, BWSNN			9L-CONV, BNN			4L-FC, DNN		3L-FC, SNN	3L-FC, SNN	
Frequency (MHz)	10	100	100	1.5-48			100	1360	20	105	506
Voltage (V)	0.6	1	1	0.66-0.9			0.44	1	0.8	0.525	0.9
MNIST Acc. (%)	98.01	98.01	98.73	96.7	97.4	98.85	98.51	98.51	97.83	97.9	98.15
Throughput (Iterations/s)	39k	390k	390k	1.7k	0.5k	0.15k	2.2k	29k	100k	-	-
Iterations/inf.	37	37	212	1	1	1	1	1	-	-	-
Inference Energy (uJ/inf.)	0.59	4.33	24.82	0.92	3.47	14.4	0.15	0.93	0.24	1.70	12.41
Efficiency (TOPS/W)	103.14	14.1	14.1	204	250	300	1.81	0.75	3.34	0.26	0.12
Scaled Area (mm^2)	0.2004			1.4			19.14		1.87	13.48	
Scaled Energy (uJ/inf.)	0.18	1.35	7.72	0.92	3.47	14.4	0.81	4.96	0.02	37.32	272.42
Scaled Efficiency (TOPS/W)	331.5	45.31	45.31	204	250	300	0.59	0.24	18.00	0.03	0.02

chip is normalized to the same 28nm technology and compared with BinarEye, it is about 5× more efficient in energy consumption. As to the area cost, as compared with the 9-CONV DNN design in BinarEye, our design reduces the area by 7×, with only 0.12% accuracy loss.

VI. Conclusion

We have designed and implemented a low-power and low-cost BW-SNN ASIC fabricated in a commercial 90nm CMOS technology for real-time bottled-drink image classification. Our design maximizes data reuse and requires no off-chip memory-access, thanks to the proposed novel systolic array that maps the 6-D data graph of the computation algorithm onto a 2-D array, in addition to the enhanced BW-SNN architecture that minimizes computation overhead. The fabricated ASIC is demonstrated in real-time for bottled-drink recognition that operates at 103.14 TOPS/W under peak efficiency. It provides a 98.73% accuracy with 0.62uJ/Inf for the MNIST hand-written character recognition problem, and 97.57% accuracy with 0.59uJ/Inf for bottled-drink classification. Its efficiency outperforms many state-of-the-art designs. If normalized to the 28nm technology, our ASIC is about 5× more efficient and 7× lower cost while achieving higher accuracy, as compared with the state-of-the-art designs.

Acknowledgement

This work was supported in part by the Ministry of Science and Technology under Grant 108-2218-E-007-026 and Taiwan Semiconductor Research Institute (TSRI) on chip fabrication.

References

[1] B. Moons, R. Uytterhoeven, W. Dehaene, and M. Verhelst, "14.5 envision: A 0.26-to-10TOPS/W subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28nm FDSOI," in *Proc. IEEE Int. Solid-State Circuits Conference (ISSCC)*, pp. 246–247, 2017.

[2] S. K. Esser, P. Merolla, J. V. Arthur, A. S. Cassidy, R. Appuswamy, A. Andreopoulos, D. J. Berg, J. L. McKinstry, T. Melano, D. Barch, C. di Nolfo, P. Datta, A. Amir, B. Taba, M. Flickner, and D. S. Modha, "Convolutional networks for fast, energy-efficient neuromorphic computing," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 113(41), pp. 11441–11446, 2016.

[3] K. Ando, K. Ueyoshi, K. Orimo, H. Yonekawa, S. Sato, H. Nakahara, M. Ikebe, T. Asai, S. Takamaeda-Yamazaki, T. Kuroda, and M. Motomura, "BRein memory: A 13-layer 4.2K neuron/0.8M synapse binary/ternary reconfigurable in-memory deep neural network accelerator in 65nm CMOS," in *Proc. IEEE Symp. VLSI Circuits*, pp. C24–C25, 2017.

[4] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, G.-J. Nam, *et al.*, "Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 10, pp. 1537–1557, 2015.

[5] P.-Y. Tan, P.-Y. Chuang, Y.-T. Lin, C.-W. Wu, and J.-M. Lu, "A power-efficient binary-weight spiking neural network architecture for real-time object classification," *arXiv:2003.06310*, 2020.

[6] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers in neuroscience*, vol. 11, p. 682, 2017.

[7] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[8] S. K. Lee, P. N. Whatmough, N. Mulholland, P. Hansen, D. Brooks, and G.-Y. Wei, "A wide dynamic range sparse FC-DNN processor with multi-cycle banked SRAM read and adaptive clocking in 16nm FinFET," in *Proc. IEEE 44th European Solid State Circuits Conference (ESSCIRC)*, pp. 158–161, 2018.

[9] J. Park, J. Lee, and D. Jeon, "A 65nm 236.5 nJ/classification neuromorphic processor with 7.5% energy overhead on-chip learning using direct spike-only feedback," in *Proc. IEEE Int. Solid-State Circuits Conference (ISSCC)*, pp. 140–142, 2019.

[10] G. K. Chen, R. Kumar, H. E. Sumbul, P. C. Knag, and R. K. Krishnamurthy, "A 4096-neuron 1M-Synapse 3.8-pJ/SOP spiking neural network with on-chip STDP learning and sparse weights in 10-nm FinFET CMOS," *IEEE Journal of Solid-State Circuits*, vol. 54, pp. 992–1002, Apr. 2019.

[11] B. Moons, D. Bankman, L. Yang, B. Murmann, and M. Verhelst, "BinarEye: An always-on energy-accuracy-scalable binary CNN processor with all memory on chip in 28nm CMOS," in *Proc. IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1–4, 2018.

[12] J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective*. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.