Multi-Context TCAM-Based Selective Computing: Design Space Exploration for a Low-Power NN

Ren Arakawa, Naoya Onizawa, *Member, IEEE*, Jean-Philippe Diguet, *Senior Member, IEEE*, and Takahiro Hanyu, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a low-power memory-based computing architecture, called selective computing architecture (SCA). It consists of multipliers and an LUT (Look-Up Table)-based component, that is multi-context ternary content-addressable memory (MC-TCAM). Either of them is selected by input-data conditions in neural-networks (NNs). Compared with quantized NNs, a higher accurate multiplication can be performed with low-power consumption in the proposed architecture. If input data stored in the MC-TCAM appears, the corresponding multiplication results for multiple weights are obtained. The MC-TCAM stores only shorter length of input data, resulting in achieving a low-power computing. The performance of the SCA is determined by three physical parameters concerning the configuration of MC-TCAM. The power dissipation of the target NN can be minimized by exploring these parameters in the design space. The hardware based on the proposed architecture is evaluated using TSMC 65 nm CMOS technology and MTJ model. In the case of speech command recognition, the power consumption at the multiplication of the first convolutional layer in a convolutional NN is reduced by 67% compared to the solution relying only on multipliers.**

***Index Terms*—Neural networks, memory-based computing, look-up table, ternary content-addressable memory, VLSI.**

I. Introduction

NEURAL networks (NNs) are machine learning models that are widely used in image recognition [1], natural language processing [2], speech command recognition [3]–[5], healthcare [6], etc. In NNs, the dynamic power consumption is large because multiplications are performed many times using input values and weights. For simple applications such as image recognition using MNIST [7], the power consumption can be reduced under negligible accuracy-loss using a quantized neural network (QNN) [8]–[10].

It is unclear whether QNNs are useful for other NN applications such as speech command recognition [11]. Another method in reducing power dissipation of NNs is look-up table (LUT)-based computing [12]. In this method, the dynamic power consumption can be reduced by replacing floating-point multiplications with LUT-based computing using ternary content-addressable memories (TCAMs). A TCAM is one of the associative memories and performs high-speed search operations [13]. In such an LUT-based computing approach, all the possible input values and weights are stored in TCAMs, and the corresponding multiplication results can be read from the random access memories (RAMs). When there are pairs of the input value and the weights in the TCAM during the search operation, the multiplication result is directly obtained from the RAM. As the data stored in the TCAM increases, the calculation accuracy increases, and so does the power consumption. For realizing an energy-efficient LUT-based computing, it is important to tune up the trade-off between the power consumption and the computational accuracy in the best balance.

In this paper, we propose selective computing architecture (SCA) based on multi-context TCAM (MC-TCAM) [14] to reduce power consumption while maintaining the computational accuracy. Each bit cell of the MC-TCAM stores multiple bits per cell using MTJ (magnetic tunnel junction) elements [15] with sharing a comparison circuit. Compared with the conventional single-context TCAM [16], MC-TCAM can increase the number of words to be stored while maintaining the power consumption of the search operation. In the SCA, an input value of multiplication is compared with a threshold value to determine whether to use either a multiplier or MC-TCAM. When the input value is below the threshold, the multiplication result is obtained by LUT-based computing using the MC-TCAM. Compared with the conventional LUT-based computing, the value stored in MC-TCAM is small since the input value range is split. The bit width of MC-TCAM is reduced, resulting in lower power consumption. In addition, several parameters of SCA are explored using training data of a NN to minimize the dynamic power consumption.

As a design example, a SCA-based hardware is designed using TSMC 65-nm CMOS and an MTJ model [17]. The target application is a convolutional NN (CNN) model for speech command recognition [11]. When the proposed SCA is applied to the first convolutional layer of this CNN model, the power consumption of the multiplication is reduced by 67% in comparison with a multiplier while maintaining the computational accuracy.

This paper is an extension of the conference paper [18]. The proposed architecture has two main contributions. First, TCAM stores only input values, while the conventional

Manuscript received April 17, 2020; revised July 22, 2020 and September 5, 2020; accepted October 1, 2020. Date of publication October 21, 2020; date of current version December 21, 2020. This work was supported in part by JSPS KAKENHI under Grant JP16H06300 and in part by the VLSI Design and Education Center, The University of Tokyo, with Synopsys Corporation. This article was recommended by Associate Editor T. Serrano-Gotarredona. *(Corresponding author: Ren Arakawa.)*

Ren Arakawa, Naoya Onizawa, and Takahiro Hanyu are with Tohoku University, Sendai 980-8577, Japan (e-mail: ren.arakawa.t3@dc.tohoku.ac.jp).

Jean-Philippe Diguet is with CNRS, Lab-STICC UMR 6285, 56100 Lorient, France.

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2020.3030104

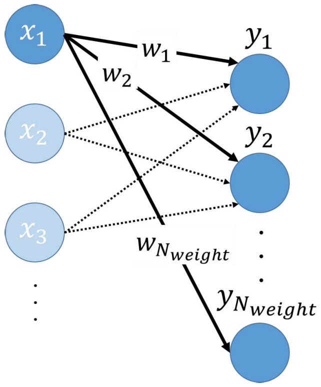


Fig. 1. Multiplication of an input value, x_1, and N_{weight}-times weights in a NN.

method stores both input values and weights [12]. When the input value hits, all multiplication results corresponding to multiple weights are obtained. This approach increases the number of multiplier uses that can be avoided with MC-TCAM. Second, MC-TCAM only applies to small input values, which requires a smaller MC-TCAM and so a reduced power consumption.

The rest of this paper is as follows. Section II reviews Quantized NNs. Section III describes the design concept of the proposed architecture. Section IV reviews MC-TCAM and provides the power consumption model of MC-TCAM. Section V describes the operation of the proposed SCA and the design space exploration. Section VI evaluates performance using an application of speech command recognition. Section VII concludes this paper.

II. Related Work

A. Review of Quantized NN

NNs are used for image recognitions such as MNIST, CIFAR-10 [19], ImageNet [1]. Fig. 1 shows multiplications of an input value and multiple weights in a NN. The power consumption of NNs is large because floating-point multipliers with high power consumption are frequently used. In order to solve this problem, there are many studies on quantized neural networks (QNNs) such as binary neural networks (BNN) [8], [9] and ternary neural networks (TNN) [20]. The input values and weights of the neural network are converted from the floating-point representation to the fixed-point representation in this method. As a result, the bit-width of the multiplier is reduced and so the power consumption. Furthermore, the hardware area is reduced by quantization [21]. As described above, several applications such as MNIST recognition can be executed with low power consumption and a high recognition accuracy using QNNs.

B. Drawbacks of QNN on Computation Accuracy

The effectiveness of quantized neural networks has been reported in several applications such as image recognition. However, it is unclear whether it can be used efficiently for other applications such as speech command recognition applications. Fig. 2 shows an overview of a speech command recognition application [22] and the simulation result of recognition rate when QNNs are applied to the application. The data set contains ten speech commands such as “yes” and “no”, other speeches and background-noise [11]. The input speech signal is converted to a speech spectrogram, which is an input to a CNN [22]. The application runs on MATLAB2019a to check the accuracy depending on the computing precision. Ideally, speeches other than the ten speech commands are classified as unknown, while background-noises are classified as background. The input values and the weights of all convolutional layers are converted from 32-bit floating-point representation to fixed-point representation (32, 24, 16 bit-length). Converting the representation to 32-bit fixed-point hardly modifies the accuracy, while converting the representation to 24 or 16 bit fixed-point significantly reduces the accuracy. Therefore, it is difficult to apply the quantization approach to low-power speech command recognition with a high recognition accuracy.

III. Design Concept of Proposed Architecture

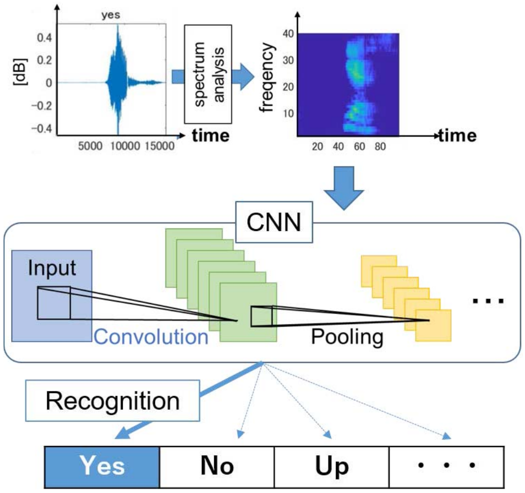
A. Conventional LUT-Based Computing

For applications such as speech command recognition, it makes sense to explore other solutions than QNNs to improve the energy efficiency. Another method to reduce the multiplication cost of neural networks is LUT-based computing using content-addressable memory (CAM) [12]. Fig. 3 shows an LUT-based computing architecture. In this method, the input values and weights of a NN are stored in the TCAM. The RAM stores the corresponding multiplication results, which is connected by match line (ML). When the input value and weights of the multiplication exist in the TCAM during the search operation, the search result is ‘hit‘. When both the input value and the weight hit in the TCAM, the decoder specifies the RAM address where the multiplication result is stored. Then the multiplication result is directly obtained from the specified RAM.

TCAM is one of associative memories that perform fast and parallel search operations [13], [23]. Fig. 4 shows a typical TCAM configuration. Each word stores several TCAM cells, which store ‘0’, ‘1’, or ’X (wildcard)’. A search data is compared with the data of all the words in parallel. Basically, TCAMs have been designed for single-context, therefore each TCAM cell contains a 1-bit information. The state of data retrieved from TCAM is defined as context.

In the LUT-based computing architecture, the input value and weight of a multiplication of a NN are searched from a TCAM. When the input value and weight hit, the multiplication result is read from a RAM. This memory-based computing (MBC) approach is using the lookup process to reduce the power consumption of NNs.

However, the LUT-based computing using TCAMs has a trade-off problem between the power consumption and the calculation accuracy. It is necessary to store a lot of data in TCAM in order to increase the calculation accuracy, while the power consumption increases in proportion to the number of TCAM cells. Thus the designer must do a choice between



(a)

	Floating-point	Fixed-point		
Speech	32-bit	32-bit	24-bit	16-bit
Yes	96.9	96.9	96.1	76.2
No	94.4	94.4	92.1	79.0
Up	98.5	98.5	97.4	79.4
Down	92.1	92.1	93.3	47.4
Left	97.4	97.4	97.4	72.3
Right	93.4	93.4	93.4	86.9
On	96.7	96.7	95.9	64.6
Off	95.8	95.4	91.2	31.3
Stop	97.2	97.2	95.2	47.8
Go	94.8	94.4	91.2	72.1
Unknown	90.5	90.4	83.7	47.7
Background	100.0	100.0	96.8	94.3
Average [%]	95.6	95.6	93.6	66.6

(b)

Fig. 2. Speech command recognition using QNN: (a) overview and (b) accuracy.

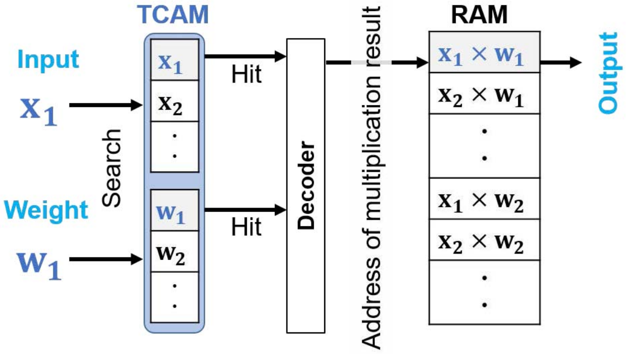


Fig. 3. Conventional LUT-based computing using TCAM [12].

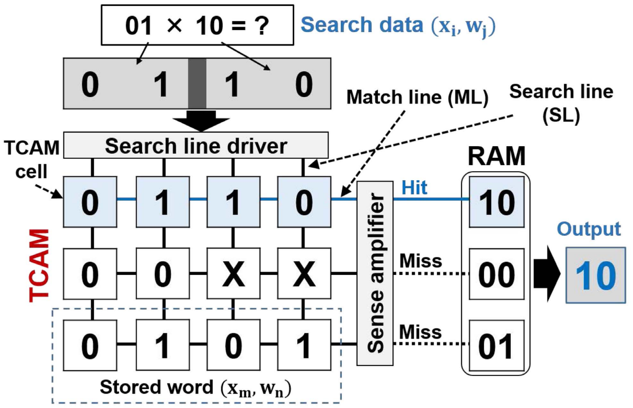


Fig. 4. Replacing multiplications by lookup search using a 4 [bit] × 3 [word] TCAM.

improving the accuracy and reducing the power consumption. The LUT-based computing approach must be improved to come up with this trade-off when accuracy is required.

B. Overview of Proposed LUT-Based Computing Architecture

Fig. 5 shows an overview of the proposed hardware architecture. Multi-context TCAM [14] is used as a TCAM of the proposed LUT-based computing. MC-TCAM stores multiple bit data in one TCAM cell and switches the data according to the context selection signal. Using MC-TCAM, the number of stored data patterns can be increased while maintaining the number of TCAM cells, leading to a reduction in power consumption.

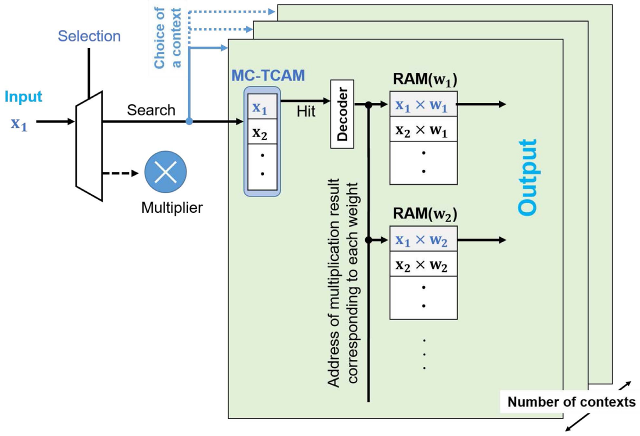


Fig. 5. Proposed LUT-based computing using MC-TCAM.

Table I summarizes the comparison among a multiplier, a conventional LUT-based architecture and the proposed hardware architecture when the number of weights of a NN is N_{weight}. A key feature of trained NN for MBC is that the weights are constant. Therefore, there is no need to search them from TCAM. In the proposed architecture, only input values are stored in TCAM. Only input values are searched from TCAM. Then, when a search result is hit, all multiplication results corresponding to each weight are obtained. Compared with the conventional LUT-based architecture, the proposed architecture searches only input values in TCAMs, resulting in reducing the number of TCAM cells.

In the proposed architecture, MC-TCAM only applies to smaller input values. As a result, the bit-width of MC-TCAM is smaller and so the power consumption.

TABLE I

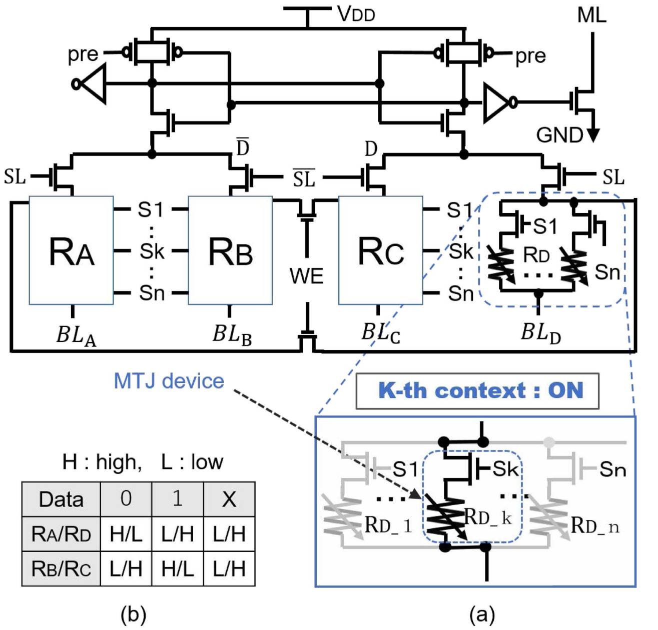
COMPARISON OF DIFFERENT STYLES OF COMPUTING

	Conventional multiplier-based	Conventional LUT-based [12]	Proposed architecture-based
Computing	Multiplier	TCAM and RAM	MC-TCAM, RAM, and multiplier
Stored data in TCAMs	-	Input values and weights	Input values
Number of operations[1]	N_{weight}-times multiplications	N_{weight}-times searches[2]	1-time search[3] or N_{weight}-times multiplications

[1] Number of operations for finding the result of multiplication of one input value and N_{weight} weights

[2] N_{weight}-times search operations using conventional single-context TCAM

[3] 1-time search operation using multi-context TCAM (MC-TCAM)



Data	0	1	X
RA/RD	H/L	L/H	L/H
RB/RC	L/H	H/L	L/H

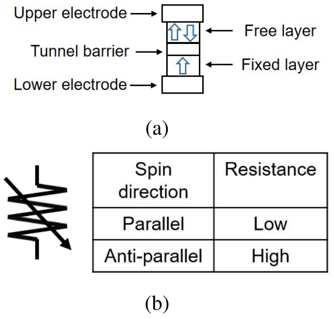
Fig. 6. MC-TCAM cell circuit with n bits: (a) circuit configuration and (b) function.

IV. MULTI-CONTEXT TCAM

A. Overview

Fig. 6 shows the MC-TCAM cell [14]. Each MC-TCAM cell contains several context bits and one context is selected during operation, while the conventional single-context TCAM cells store one bit. The k-th ($1 \leq k \leq n$) context of the MC-TCAM cell stores '0', '1', or 'wildcard (X)' using four resistance: R_{A_k}, R_{B_k}, R_{C_k}, and R_{D_k}. This shared comparison circuit with MTJ devices improves the area utilization and increases the number of stored data. In the search operation at the pre-charge phase, pre is low to pre-charge D and $\overline{D}$. At the evaluation phase, search line (SL and $\overline{SL}$) is active according to input data. When the search result is 'hit', ML is high. In the write operation, WE is high and four bit lines (BLs) generate two current signals.

Fig. 7 shows a two-terminal MTJ device and its symbol [15]. The MTJ device consists of three layers; a free layer, a tunnel barrier, and a fixed layer. The resistance state of the MTJ element is determined by the spin direction of the free layer. The spin direction of the free layer can be changed by passing an current. The free layer is either parallel or anti-parallel to the fixed layer. Since the magnetic spin direction is maintained without power supply, the MTJ element can be used as a non-volatile device. An MTJ model [17] is used to simulate the proposed architecture. The MTJ model has a high resistance of 1964 Ω and a low resistance of 763 Ω.



Spin direction	Resistance
Parallel	Low
Anti-parallel	High

Fig. 7. MTJ device: (a) schematic and (b) symbol.

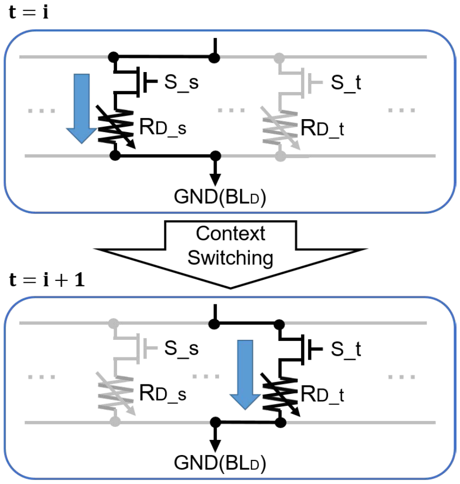


Fig. 8. Example of context-switching behavior in the MC-TCAM.

Fig. 8 shows the context switching of MC-TCAM. When the context is switched in $(i+1)$-th search, switch the selection signal (S_1, S_2, ..., S_s, ..., S_t, ..., S_n) that was selected until the i-th search. When the context is switched, the gates of the access transistors are charged and discharged, which increases power consumption compared to the case without context switching.

B. Modeling of MC-TCAM

The power consumption of the MC-TCAM cell is modeled in order to perform a design space exploration of the selective

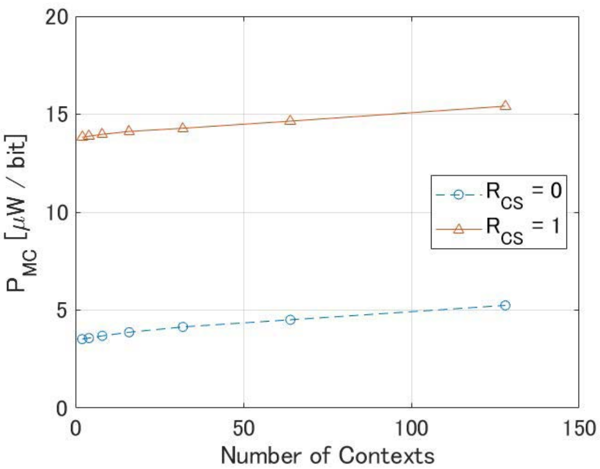


Fig. 9. MC-TCAM power consumption ($=P_{MC}$) dependence on number of contexts.

computing architecture (SCA) parameter. P_{MC}, the power consumption of the MC-TCAM cell in each number of contexts can be expressed as the following equation:

$$P_{MC} = R_{CS} P_{cs\_cell} + (1 - R_{CS}) P_{ck\_cell} \quad (1)$$

where R_{CS} is the probability of the context-switching, P_{cs_cell} is the power consumption of the cell when the context changes every search, and P_{ck_cell} is the power consumption of the cell without the context switching.

Fig. 9 shows P_{cs_cell} and P_{ck_cell} when the number of contexts is changed. A 16-bit × 64-word TCAM is designed and evaluated at 1.0 GHz using a 65-nm TSMC and an MTJ model [17] for the simulation. MC-TCAM increases the power consumption slightly when the number of contexts is increased. The power gap between $R_{CS} = 0$ and $R_{CS} = 1$ is constant regardless of the number of contexts. In Sections V and VI, a design space exploration of parameters in the proposed architecture performs using the power consumption models.

Algorithm 1 Design Space Exploration of SCA Parameters

```
input training data, $P_{cs\_cell}$, $P_{ck\_cell}$, $P_{mul}$, $N_{weight}$, $P_{RAM}$
output explored $CB_{min}$, $WB_{min}$, $N_{w-min}$
$P_{SCA-min} \leftarrow \infty$
for $CB \in N_{CB}$ do
  for $WB \in N_{WB}$ do
    $SB \leftarrow 32 - CB - WB$
    for $N_{word} \leftarrow 1 : 2^{SB}$ do
      for $x \in$ training data do
        $WB_x \leftarrow x[(32 - WB) : 32]$
        $CB_x \leftarrow x[(2 + SB) : (2 + SB + CB)]$
        $SB_x \leftarrow x[1 : (1 + SB)]$
        if $WB_x == 0$ then
          $freq(CB_x, SB_x) \leftarrow freq(CB_x, SB_x) + 1$
        end if
      end for
MC-TCAM
      $\leftarrow N_{word}$ most frequent input at each contexts
      $P_{MC} \leftarrow$ equation (1)
      $P_{SCA} \leftarrow$ equation (2)
      if $P_{SCA} < P_{SCA-min}$ then
        $P_{SCA-min} \leftarrow P_{SCA}$
        $CB_{min} \leftarrow CB$
        $WB_{min} \leftarrow WB$
        $N_{w-min} \leftarrow N_{word}$
      end if
    end for
  end for
end for
```

V. Selective Computing Architecture (SCA)

A. Overview

Fig. 10 illustrates the flow chart and the block diagram of the proposed SCA approach. SCA mainly consists of a multiplier and MC-TCAM. This architecture efficiently finds the results of multiplying input values by N_{weight} weights as shown in Fig. 1. In SCA, an input value is divided into WB (wasted bit), CB (context bit) and SB (search bit). The operation of SCA is as follows.

1) Either the multiplier or the MC-TCAM is selected by comparing the input value with the threshold, th. When the input value is below th, steps 2) and 3) are performed. Otherwise, the multiplier is selected. If the multiplier is selected, steps 2) and 3) are skipped, while the multiplier is used N_{weight} times to obtain the multiplication results.
2) If the input value is smaller than the threshold, the upper $WB (= 32 - \log_2 th)$ bits of the input value are “0”. As the upper WB bits are clearly “0”, they do not have to be searched in the MC-TCAM. One of the 2^{CB} contexts of MC-TCAM is chosen using the CB bits of upper digits among the remaining bits.
3) SB bits of the input value is searched in the MC-TCAM with the selected context. If the search data hits a row of the MC-TCAM, the multiplication results for the N_{weight}-times weights are read from the SRAM at once. Otherwise, the result of multiplication is calculated using the multiplier.

The proposed architecture works with 32-bit fixed-point precision regardless of the parameters, so the recognition accuracy of the application is not compromised.

B. Design Space Exploration of SCA Parameters

SCA has three parameters: WB (wasted bit), CB (context bit), N_{word} (number of searched MC-TCAM words). A design space exploration of these parameters is performed once offline using training data in order to minimize the power consumption

P_{SCA}, the power consumption of SCA-based hardware to calculate the multiplication results of an input value and all weights, is modeled by the following equation:

$$\begin{aligned} P_{SCA} &= (1 - R_{MC}) P_{mul} N_{weight} \\ &\quad + R_{MC} (P_{MC} N_{word} SB + P_{RAM} N_{weight}) \end{aligned} \quad (2)$$

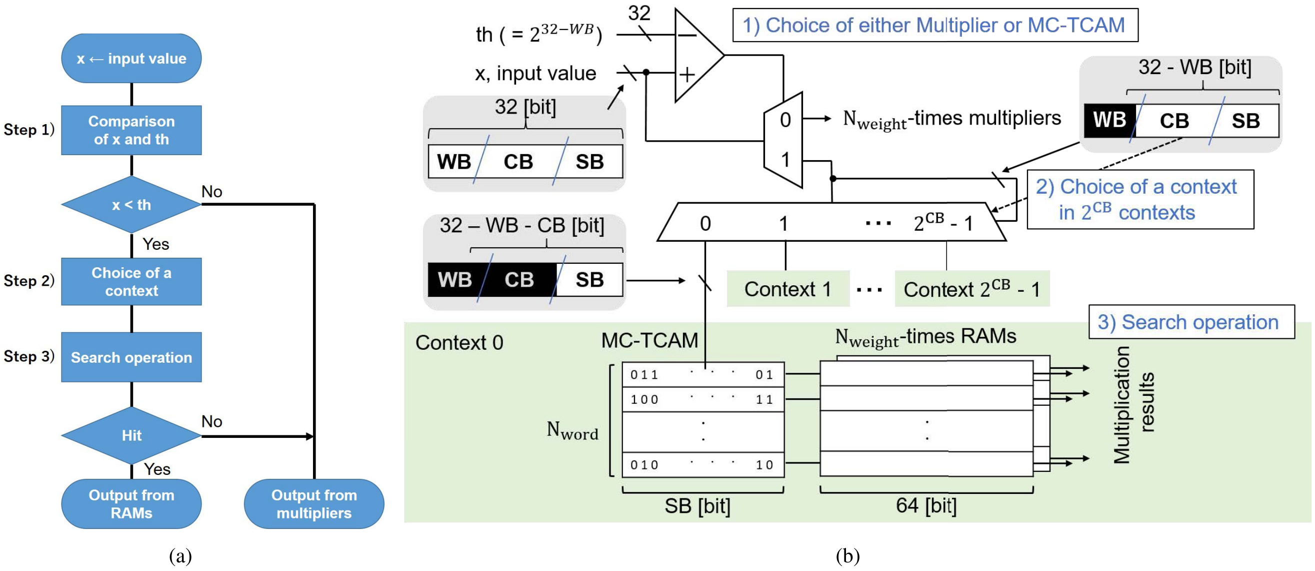
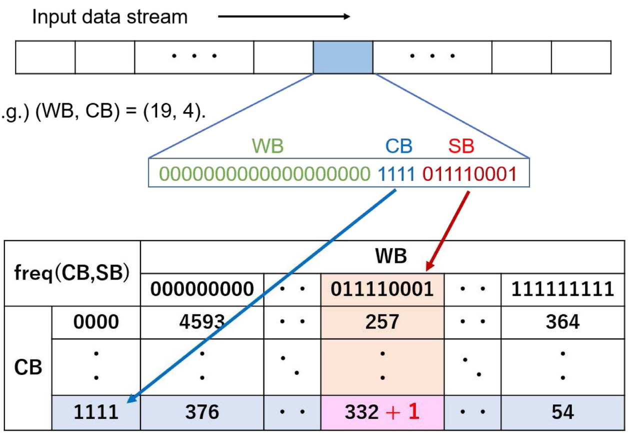


Fig. 10. Proposed SCA: (a) flow chart and (b) block diagram.



freq(CB,SB)		WB				
		000000000	· ·	011110001	· ·	111111111
CB	0000	4593	· ·	257	· ·	364
	:	:	⋱	:	⋱	:
	1111	376	· ·	332 + 1	· ·	54

Fig. 11. Calculating the frequency of input values.

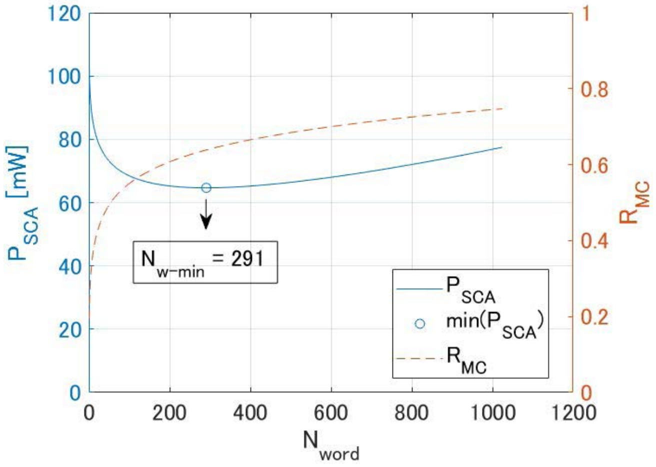


Fig. 12. Effect of number of MC-TCAM words on P_{SCA} and R_{MC} in design space exploration when $(WB, CB) = (21, 1)$ using data set from [11].

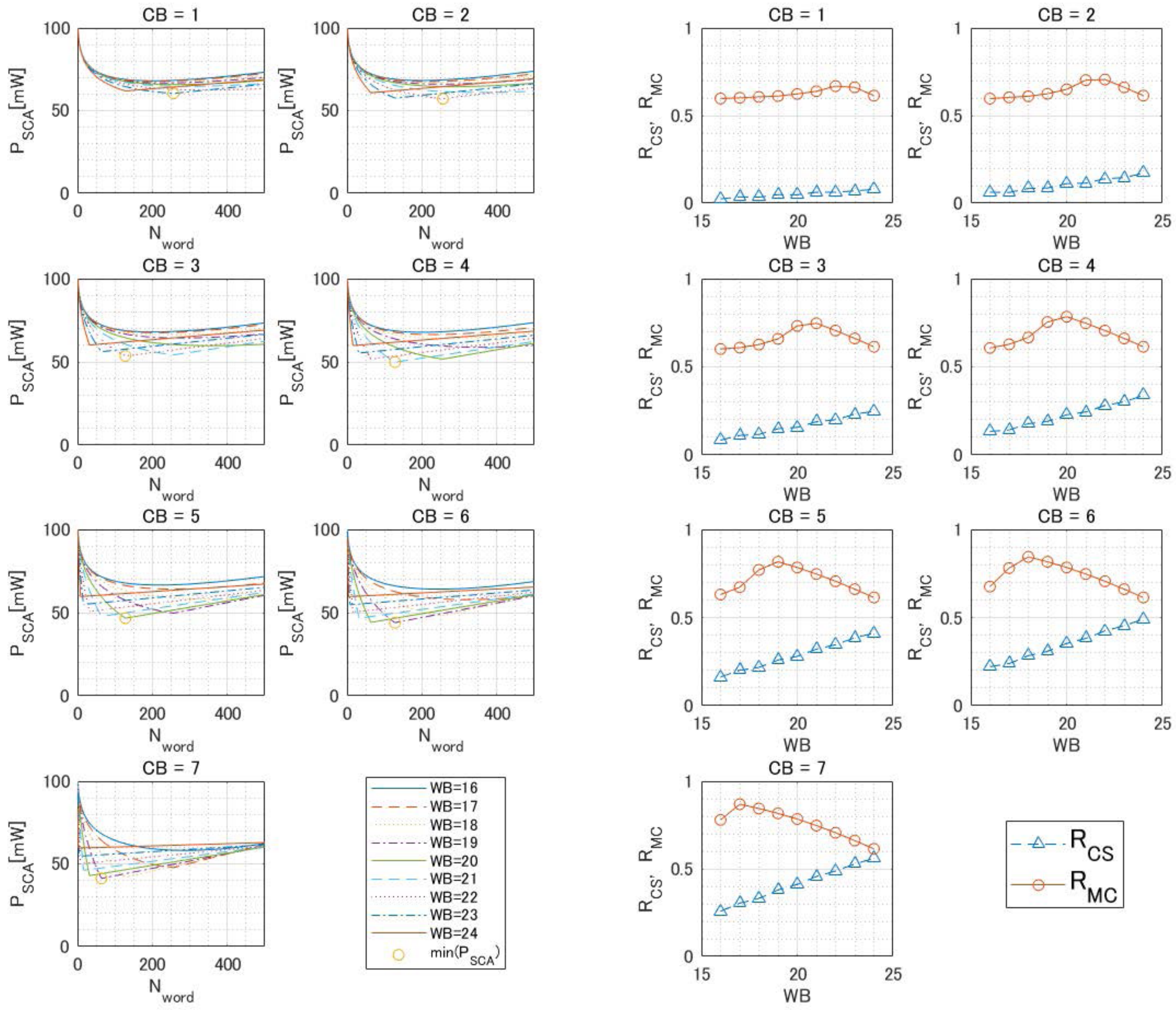
where R_{MC} is the probability that the input value is less than th and hits MC-TCAM, N_{word} is number of the MC-TCAM words. Furthermore, P_{mul}, P_{MC}, P_{RAM} are the power consumption of the multiplier, the MC-TCAM, and the RAM respectively, and N_{weight} is the number of weights for an input value in a NN.

Equation (2) uses P_{MC} preliminary calculated by equation (1). In a parameter exploration of SCA, WB, CB and N_{word} are swept in order to minimize P_{SCA} as shown in Algorithm. 1. Line 2 through 5 of Algorithm 1 sweeps CB, WB, N_{word}. Then, other values are determined as follows: $th = 2^{32-WB}$, $SB = 32 - WB - CB$, number of contexts $= 2^{CB}$. Lines 6 through 13 of Algorithm 1 calculate the frequency of input values in order to determine the data to be stored in MC-TCAM. x, an input value, is extracted one by one from training data stream. Then, x is divided into upper WB bits, intermediate CB bits, and lower SB bits and compared with th. If the input value is less than th (ie, the upper WB bits of x is '0'), a context is selected using CB_x of x. The frequency that each input value is entered to each context of MC-TCAM is counted as shown in Fig. 11. Line 14 of Algorithm 1 stores N_{word} high-frequency input values of each context in MC-TCAM. Line 15 calculate P_{MC}, the power consumption of MC-TCAM cell, according to equation (1). Line 16 calculate P_{SCA} at each CB, WB, N_{word} using R_{MC} and P_{MC} according to equation (2). Lines 17 through 22 explore the parameters that minimize P_{SCA} as shown in Fig. 12. The combination of CB, WB, and N_{word} that minimizes P_{SCA} is determined as the combination of CB_{min}, WB_{min}, and N_{w-min}.

VI. Evaluation and Discussion

A. Experimental Setup

In order to evaluate the proposed method, SCA is applied to the multiplications of the first convolutional layer of a CNN model for the speech command recognition application



(a)

(b)

Fig. 13. Design space exploration: (a) P_{SCA} in each WB, CB, and P_{Nword} and (b) R_{CS} and R_{MC} in each WB, CB and N_{w-min}.

described in [22]. The CNN has five convolutional layers and recognizes 10 speech commands in the dataset [11]. As there are 12 parallel filters in the first convolutional layer of the CNN, N_{weight} is 12. In addition, the speech command dataset is split into training data, validation data and test data. The training dataset contains about 25,000 speech commands, of which about 500 speech commands are used for the design space exploration of SCA. For evaluation, middle frequency components of the speech spectrogram of 256 speech "yes" in the test data set are used.

The proposed hardware is designed using TSMC 65-nm CMOS, an SRAM [24] and an MTJ model [17]. The performance of the hardware except the multiplier is evaluated using HSPICE while the performance of multiplier is evaluated with the gate-level netlist using Synopsys Design Compiler.

B. Design Space Exploration

The design space exploration is performed with WB and CB varying from 16 to 24 and 1 to 7, respectively. Fig. 13 (a) shows P_{SCA} using training data for each WB, CB, and N_{word} configuration. N_{w-min}, which is N_{word} that minimize P_{SCA}, is determined for each WB and CB configuration. Fig. 13 (b) shows R_{CS} and R_{MC} using training data when N_{word} is N_{w-min} in each WB and CB. As CB increases, R_{CS} is large, while WB that maximizes R_{MC} is small.

Fig. 14 shows the power consumption of the proposed hardware using training data when N_{word} is N_{w-min} for each WB and CB configuration. WB that minimizes P_{SCA} is defined as WB_{min} for each CB configuration. When CB is 1, N_{w-min} is 256 and WB_{min} is 23. The explored N_{w-min} and WB_{min} in each CB are used for evaluation.

C. Evaluation

Fig. 15 shows the histogram of the speech spectrograms used for evaluation. If the speech spectrogram is smaller than th, the result of multiplication with the weight is obtained from MC-TCAM.

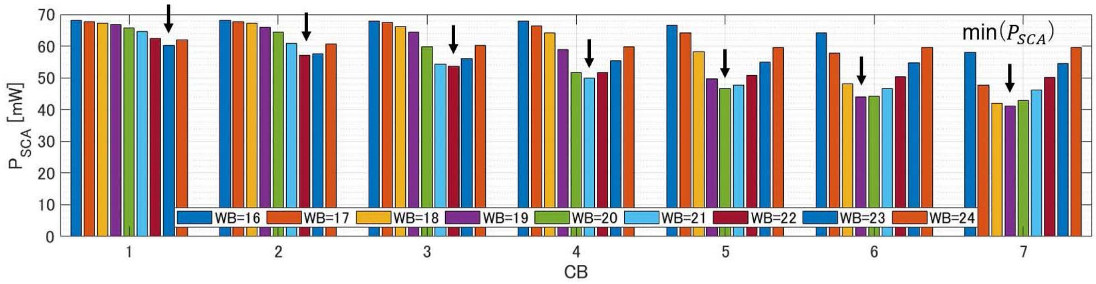


Fig. 14. P_{SCA} in each WB, CB, and N_{w-min} in the design space exploration.

TABLE II

PERFORMANCE COMPARISON FOR THE SPEECH COMMAND RECOGNITION

		WB_{min}	N_{w-min}	R_{MC}	R_{CS}	Power [mW]	Accuracy [%]
32×32-bit multiplier		-	-	-	-	123.6	95.6
Previous	SCA without explorations [18] (CB = 4)	(WB = 19)	(N_{word} = 256)	0.66	0.24	67.9	95.6
Proposed	CB = 1	23	256	0.64	0.08	62.2	95.6
	CB = 2	22	256	0.70	0.15	58.1	95.6
	CB = 3	22	128	0.70	0.24	54.6	95.6
	CB = 4	21	128	0.74	0.28	50.5	95.6
	CB = 5	20	128	0.79	0.32	46.6	95.6
	CB = 6	19	128	0.83	0.36	43.4	95.6
	CB = 7	19	64	0.83	0.43	40.5	95.6

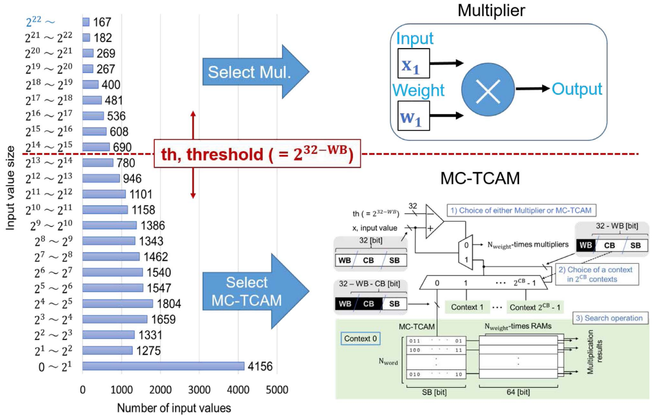


Fig. 15. Histogram of middle frequency components of the speech spectrogram of speech “yes” in the test data set [11].

The proposed SCA is compared with a 32×32-bit multiplier and SCA without parameter explorations [18]. Table II shows the comparison of a 32 × 32-bit multiplier, previous SCA without parameter explorations and proposed SCA after parameter explorations. This table shows R_{MC}, R_{CS}, and power consumption using the test data in each CB after the exploration. As CB increases, WB_{min} and N_{w-min} decrease. As CB increases, R_{MC} and R_{CS} increase and power consumption decreases. We observe the SCA with the best configuration ($CB = 7$, $WB = 19$, $N_{word} = 64$) can achieve a significant gain of 67% compared to the conventional computing using only multipliers, while the computational accuracy is maintained. Table III shows the power breakdown in the best configuration. MC-TCAM consumes 7% of total power consumption.

TABLE III

POWER BREAKDOWN OF SCA BASED HARDWARE IN THE BEST CONFIGURATION

	Multiplier	RAM	MC-TCAM	Total
Power [mW]	21.6	15.9	3.0	40.5

D. Discussion

It is difficult to directly compare the proposed SCA with the conventional LUT-based architecture [12] for several reasons, such as different target applications. In [12], weights of a target NN are quantized and used for LUT-based computing. In this method, the power consumption is smaller than that using only multipliers with a few-percent accuracy loss. The speech recognition application evaluated in the conventional LUT-based computing classifies speech signals of 26 English letters. The quantization method is effective for such simple applications. However, it is difficult to apply conventional LUT-based computing that quantizes weights to the speech command recognition application evaluated in this paper. When the input values and weights are quantized into 16 bits in the speech command recognition, the recognition accuracy is greatly reduced as shown in Fig. 2. As a result, it is difficult to tune up the trade-off between computational accuracy and power consumption with conventional LUT-based computing, while the proposed SCA reduces power consumption while maintaining recognition accuracy.

For the best configuration of the proposed SCA in the evaluation, N_{w-min} is equal to 2^{SB}. In this case, MC-TCAM is not required. SB can be regarded as the input address of SRAM that stores the multiplication result. Power consumption is 37.5 mW when MC-TCAM is not used as shown in Table III. However, this is a special case. MC-TCAM is basically required because the optimal parameters vary depending on the data set.

The area of the SRAM is not considered here since first we reuse a state of the art model of SRAM [24] and secondly because the aim of memory-based methods is primarily power savings while relaxing the area constraint. Therefore, the estimated area of SCA contains only those of multipliers and MC-TCAM here. The areas of a multiplier and explored MC-TCAM are 11,500 μm^2 and 35,600 μm^2, respectively. As shown in Table I, the conventional multiplier-based solution requires N_{weight} multipliers, and the proposed SCA additionally requires MC-TCAM. The areas of conventional multiplier-based and proposed SCA-based computing are estimated to be 138,000 μm^2 and 174,000 μm^2, respectively. The proposed SCA increases the area by 26% compared with the conventional method. Note that when the number of CNN filters increase, the SCA area overhead will decrease.

SCA can actually replace various operations with memory-based computing. SCA can be applied to other operations, such as pooling and activation functions. However, these operations have a limited impact on power consumption compared to filters, so gains would be small. SCA is very effective for operations which are highly numerous and greedy for power such as multiplications in the convolutional layers of a CNN.

As the proposed architecture can be applied to each layer of CNN individually, it is scalable to more layers of CNNs and/or more complex CNNs. The performance merit of SCA is determined by the bias of the dataset and the number of filters in a CNN. For example, if the parameters are $CB = 7$, $WB = 19$, $N_{word} = 64$, $R_{MC} = 0.83$, and $N_{weight} = 24$, SCA can reduce the power consumption by 69% compared to the case where only the multiplier is used.

In usual speech recognition applications, the throughput is moderate since it is equivalent to a phone. For instance, it would be 512 kb/s with 32 bits/sample @16kHz. The proposed LUT operation (MC-TCAM + SRAM) has been confirmed to operate at 1 GHz. Therefore, the throughput is not the bottleneck of the system. In contrast, the leakage current must be considered in this system. The LUT operation is in a sleep state most of the time. MC-TCAM has low leakage current by power gating because it has non-volatile structure using MTJ devices. However, it is necessary to pay attention to the leakage current of the SRAM. The dynamic energy of one SCA operation is 80 pJ, while the static energy of SCA due to the leakage current of SRAM is 1.6 nJ. If the multiplication result is stored in MRAM (magnetoresistive RAM) [25] using MTJ devices instead of the SRAM, the non-volatile property can help to significantly reduce the leakage current of RAM. It is the next step of our work after this study that demonstrates the relevancy of the proposed SCA.

The design space exploration is performed once offline and takes about 1 hour in the data set. However, the novelty of the paper is not the design space exploration method. The design space exploration is used to identify the best configuration and to demonstrate that such a configuration exists. On the other hand, an efficient method is required to identify the best configuration. A machine learning method will be used to perform this training phase. This is another part of our future work.

VII. Conclusion

In this paper, we have proposed the selective computing architecture using MC-TCAM for low-power multiplications in NNs. Either the multiplier or the MC-TCAM is selected by comparing the input value with the threshold. In the proposed architecture, the high-precision multiplication result is obtained with the low-power consumption. SCA can replace many uses of multipliers with MC-TCAM computing. MC-TCAM is used as the LUT, improving the memory capacity, because the MC-TCAM stores multiple data in one memory cell using the CMOS/MTJ device-hybrid circuit technique. By preparing MC-TCAM for only small input values, the bit width of MC-TCAM is small, and the power consumption is reduced. The design space exploration is performed for low-power SCA. As a result, the power consumption of the proposed hardware is reduced up to 67 % compared to the solution relying only on multipliers while maintaining the accuracy in a CNN model using TSMC 65-nm CMOS and the MTJ model.

SCA is a promising low-power solution for NN inference and a future work will be devoted to the exploration of different categories of NN applications. If SCA also performs well on the different layers of deep NN, then very significant gains can be expected.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[2] T. Mikolov, M. Karafiát, L. Burget, and S. Kombrink, "Recurrent neural network based language model," in *Proc. 11th Annu. Conf. Int. Speech Commun. Assoc.*, Sep. 2010, pp. 1045–1048.

[3] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. Audio, Speech Language Process.*, vol. 22, no. 10, pp. 1533–1545, Oct. 2015.

[4] L. Deng *et al.*, "Recent advances in deep learning for speech research at Microsoft," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 8604–8608. [Online]. Available: https://www.microsoft.com/en-us/research/publication/recent-advances-in-deep-learning-for-speech-research-at-microsoft/

[5] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[6] K. Srinivas, B. Rani, and D. Govardhan, "Applications of data mining techniques in healthcare and prediction of heart attacks," *Int. J. Comput. Sci. Eng.*, vol. 2, pp. 250–255, Jan. 2010.

[7] Y. LeCun and C. Cortes. (2010). *MNIST Handwritten Digit Database*. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[8] M. Courbariaux and Y. Bengio, "BinaryNet: Training deep neural networks with weights and activations constrained to +1 or −1," *CoRR*, vol. abs/1602.02830, 2016. [Online]. Available: http://arxiv.org/abs/1602.02830

[9] S. Yu *et al.*, "Binary neural network with 16 Mb RRAM macro chip for classification and online training," in *IEDM Tech. Dig.*, Dec. 2016, pp. 16.2.1–16.2.4.

[10] J. Kim *et al.*, "Area-efficient and variation-tolerant in-memory BNN computing using 6T SRAM array," in *Proc. Symp. VLSI Circuits*, Jun. 2019, pp. C118–C119.

[11] P. Warden, “Speech commands: A dataset for limited-vocabulary speech recognition,” *CoRR*, vol. abs/1804.03209, 2018. [Online]. Available: http://arxiv.org/abs/1804.03209

[12] M. S. Razlighi, M. Imani, F. Koushanfar, and T. Rosing, “LookNN: Neural network with no multiplication,” in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2017, pp. 1775–1780.

[13] K. Pagiamtzis and A. Sheikholeslami, “Content-addressable memory (CAM) circuits and architectures: A tutorial and survey,” *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 712–727, Mar. 2006.

[14] N. Onizawa, R. Arakawa, and T. Hanyu, “Design of an MTJ based nonvolatile multi-context ternary content-addressable memory,” *J. Appl. Logics*, vol. 6, no. 7, pp. 89–105, Jan. 2020.

[15] S. Ikeda *et al.*, “A perpendicular-anisotropy CoFeB-MgO magnetic tunnel junction,” *Nature Mater.*, vol. 9, no. 9, pp. 721–724, Jul. 2010.

[16] B. Song, T. Na, J. P. Kim, S. H. Kang, and S.-O. Jung, “A 10T-4MTJ nonvolatile ternary CAM cell for reliable search operation and a compact area,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 6, pp. 700–704, Jun. 2017.

[17] N. Sakimura *et al.*, “High-speed simulator including accurate MTJ models for spintronics integrated circuit design,” in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2012, pp. 1971–1974.

[18] R. Arakawa, N. Onizawa, J.-P. Diguet, and T. Hanyu, “Multi-context TCAM based selective computing architecture for a low-power NN,” in *Proc. 26th IEEE Int. Conf. Electron., Circuits Syst. (ICECS)*, Nov. 2019, pp. 117–118.

[19] R. Doon, T. K. Rawat, and S. Gautam, “Cifar-10 classification using deep convolutional neural network,” in *Proc. IEEE Punecon*, Nov. 2018, pp. 1–5.

[20] H. Alemdar, V. Leroy, A. Prost-Boucle, and F. Petrot, “Ternary neural networks for resource-efficient AI applications,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2547–2554.

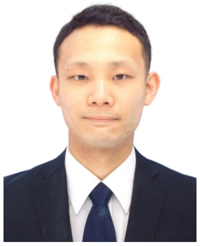
[21] S. Liang, S. Yin, L. Liu, W. Luk, and S. Wei, “FP-BNN: Binarized neural network on FPGA,” *Neurocomputing*, vol. 275, pp. 1072–1086, Jan. 2018.

[22] MATLAB2019a. *Speech Command Recognition Using Deep Learning*. [Online]. Available: https://www.mathworks.com/help/releases/R2019a/deeplearning/examples/deep-learning-speech-recognition.html

[23] W. Choi, K. Lee, and J. Park, “Low cost ternary content addressable memory using adaptive matchline discharging scheme,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–4.

[24] K. Z. Hui and D. A. Tuan, “A 16Kb 10T-SRAM with 4x read-power reduction,” in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 3485–3488.

[25] J.-P. Diguet, N. Onizawa, M. Rizk, J. Sepulveda, A. Baghdadi, and T. Hanyu, “Networked power-gated MRAMs for memory-based computing,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 12, pp. 2696–2708, Dec. 2018.



Ren Arakawa received the B.E. degree in electrical information and physics engineering from Tohoku University, Sendai, Japan, in 2019, where he is currently pursuing the master’s degree with the Research Institute of Electrical Communication. His research interests include the energy-efficient VLSI design using memory-based computing architecture and its applications, such as brain-like computers. He received the Best Young Professionals Paper Award of the 26th IEEE International Conference on Electronics Circuits and Systems (ICECS), Italy, in 2019.

Naoya Onizawa (Member, IEEE) received the B.E., M.E., and D.E. degrees in electrical and communication engineering from Tohoku University, Japan, in 2004, 2006, and 2009, respectively. He is currently an Assistant Professor with the Research Institute of Electrical Communication, Tohoku University, and a JST PRESTO Researcher, Japan. He was a Post-Doctoral Fellow at the University of Waterloo, Canada, in 2011, and at McGill University, Canada, from 2011 to 2013. In 2015, he was a Visiting Associate Professor with the University of Southern Brittany, France. His main interests and activities are in the energy-efficient VLSI design based on asynchronous circuits and probabilistic computation, and their applications, such as brain-like computers. He received the Best Paper Award in 2010 IEEE ISVLSI, the Best Paper Finalist in 2014 IEEE ASYNC, Kenneth C. Smith Early Career Award for Microelectronics Research in 2016 IEEE ISMVL, and the MEXT Young Scientists’ Prize in 2020.

Jean-Philippe Diguet (Senior Member, IEEE) received the Ph.D. degree from Rennes University, France, in 1996. In 1997, he has been a Visitor Researcher at IMEC, Belgium, and then an Associate Professor at UBS University, France, until 2002. In 2003, he co-funded the Dixip company in the domain of wireless embedded systems. Since 2004, he has been a CNRS Researcher at Lab-STICC, where he has been heading the MOCS Team until 2016. He has been a Visitor Researcher at The University of Queensland, Australia, in 2010, and an invited Professor at Tohoku University, Japan, in November 2014 and May 2019, and also at USP University, Brazil, in November 2016. He is currently a CNRS Director of Research at Lab-STICC, Lorient/Brest, France. His current work focuses on various aspects of embedded system design: NoC-based MPSoC architectures, including memory-based computing, embedded/on-chip intelligence, and dedicated hardware accelerators.

Takahiro Hanyu (Senior Member, IEEE) received the B.E., M.E., and D.E. degrees in electronic engineering from Tohoku University, Sendai, Japan, in 1984, 1986, and 1989, respectively. He is currently a Professor and Education/Research Councilor with the Research Institute of Electrical Communication, Tohoku University. His general research interests include nonvolatile logic circuits and their applications to ultra-low-power and/or highly dependable VLSI processors, and post-binary computing and its application to brain-inspired VLSI systems. He received the Sakai Memorial Award from the Information Processing Society of Japan in 2000, the Judge’s Special Award at the 9th LSI Design of the Year from the Semiconductor Industry News of Japan in 2002, the Special Feature Award at the University LSI Design Contest from ASP-DAC in 2007, the APEX Paper Award of Japan Society of Applied Physics in 2009, the Excellent Paper Award of IEICE, Japan, in 2010, Ichimura Academic Award in 2010, the Best Paper Award of IEEE ISVLSI 2010, the Paper Award of SSDM 2012, the Best Paper Finalist of IEEE ASYNC 2014, and the Commendation for Science and Technology by MEXT, Japan, in 2015.