IEICE TRANS. ELECTRON., VOL.E104–C, NO.7 JULY 2021

PAPER Special Section on Solid-State Circuit Design — Architecture, Circuit, Device and Design Methodology

Low-Power Implementation Techniques for Convolutional Neural Networks Using Precise and Active Skipping Methods

Akira KITAYAMA[†a)], **Goichi ONO**[†], **Tadashi KISHIMOTO**[†], **Hiroaki ITO**[††], ***Nonmembers***, ***and*** **Naohiro KOHMU**[†], ***Member***

SUMMARY Reducing power consumption is crucial for edge devices using convolutional neural network (CNN). The zero-skipping approach for CNNs is a processing technique widely known for its relatively low power consumption and high speed. This approach stops multiplication and accumulation (MAC) when the multiplication results of the input data and weight are zero. However, this technique requires large logic circuits with around 5% overhead, and the average rate of MAC stopping is approximately 30%. In this paper, we propose a precise zero-skipping method that uses input data and simple logic circuits to stop multipliers and accumulators precisely. We also propose an active data-skipping method to further reduce power consumption by slightly degrading recognition accuracy. In this method, each multiplier and accumulator are stopped by using small values (e.g., 1, 2) as input. We implemented single shot multi-box detector 500 (SSD500) network model on a Xilinx ZU9 and applied our proposed techniques. We verified that operations were stopped at a rate of 49.1%, recognition accuracy was degraded by 0.29%, power consumption was reduced from 9.2 to 4.4 W (−52.3%), and circuit overhead was reduced from 5.1 to 2.7% (−45.9%). The proposed techniques were determined to be effective for lowering the power consumption of CNN-based edge devices such as FPGA.

key words: *convolutional neural network (CNN), SSD500 network, deep neural network (DNN) implementation, low power consumption, embedded AI technique*

1. Introduction

Image recognition technology using deep neural network has dramatically improved recognition accuracy compared with previous rule-based algorithms. Such deep neural network technology is starting to be implemented in edge devices used in autonomous driving, medical equipment, robotics, and infrastructure monitoring systems, etc. Widely used for image classification, object detection, and segmentation a number of convolutional neural network (CNN) models have been proposed [1]–[6]. Although the recognition accuracy of a CNN improves as its layer structure becomes deeper, the amount of computation increases, and 10 giga to 100 giga-operations (GOP) are generally required for each image. The enormous power consumption and processing time of CNNs are dominated by convolutional operations, so they need to be implemented efficiently by edge devices. Network compression methods that do not affect recognition accuracy have been reported, as well as high-speed low-power AI accelerators [7]–[9]. However, the multiplication and accumulation (MAC) occupies 70–90% of the total convolution operation time, so it needs to be calculated more efficiently to further reduce power consumption. In previously developed zero-skipping techniques, the MAC operation is skipped when the input data or weight is zero in the filter unit of the convolution operation. So far, there are some researches that realize speeding up convolution operation by "zero-skipping technique" with FPGA circuits [10], [11], and with a dedicated hardware accelerator [12]–[15]. These techniques make good use of the CNN calculation feature where the output of each layer, i.e., the input data of the next layer, contains many zeros. This feature is effective for increasing computation speed and lowering power consumption. Some studies have driven the effective speed of convolution operations beyond the performance of the accelerators itself [16]–[18].

In the zero-skipping technique, it is necessary to monitor whether the input data or weight is zero for stopping MAC operation. However, the CNN model compressed for implementation in edge device has lower ratio of zero weight than before compression, this is because the compression process preferentially removes the weights that are not calculated (that is, when they are zero). We noticed that stopping each multiplier precisely using each zero input data is more effective for conserving power than stopping the MAC operation using all zero input data, i.e., the zero-skipping technique. We also found that a large-scale logic circuit is needed for monitoring both the input data and weight to stop MAC operation. Thus, we propose the following two techniques: precise zero-skipping, to stop the multiplier and accumulator precisely using small-scale gating-logic circuits that monitor only the input data (not the weight), and active-skipping, to stop calculation if the input data is a small value (e.g., 1 or 2 in 8-bit or 16-bit integers) that does not affect recognition accuracy.

The rest of this paper is organized as follows: Section 2 describes the concepts and architectures of the proposed precise zero-skipping and active data-skipping methods. Section 3 shows the results of implementing CNN on FPGA, and the evaluation results of recognition accuracy, power consumption, and circuit overhead in each method. Finally, Sect. 4 concludes this paper.

Manuscript received July 15, 2020.
Manuscript revised November 8, 2020.
Manuscript publicized December 22, 2020.

[†]The authors are with Center for Technology Innovation – Measurement and Electronics, Hitachi Ltd. Research & Development Group, Kokubunji-shi, 185–8601 Japan.
[††]The author is with Hitachi Automotive Systems Ltd., Hitachinaka-shi, 312–8503 Japan.
a) E-mail: akira.kitayama.er@hitachi.com

DOI: 10.1587/transele.2020CDP0003

2. Proposed Low Power Implementation Techniques

2.1 CNN Implementation Using Zero-Skip Technique

Figure 1 shows a general CNN configuration and a block diagram of one neuron for implementation in hardware. In the convolution layer of CNN, the n-th input data (D_n) and m-th weight (W_m) based on filter size ($F = f \times f$) are multiplied and accumulated. Then, bit-shift processing is applied, a bias coefficient is added, and finally, the ReLU (rectified linear unit), which is one of the activation functions, is processed. The ReLU processing outputs the input data with negative values as zero, increasing the ratio of zero in the input data of the next layer.

Next, we describe the implementation of zero skipping on a hardware (HW) device. One way to implement this technique is to stop operation using clock gating. A circuit structure using zero skipping with clock gating is shown in Fig. 2, where the n-th input data is D_n, the m-th weight is W_m, the bit width of D_n and W_m are K-bit, and the filter size is $F = f \times f$. The input data and weight read from the storage unit are judged by the K-bit NOR circuit as to whether or not they are zero. In this paper, the judgment logic related to clock gating, such as the NOR circuit, is called gating logic (GL) circuit. The GL circuit for the n-th input data is expressed as GL_{dn}, the GL circuit for the m-th weight is GL_{wm}, and their outputs are expressed as Z_{dn} and Z_{wm}, respectively. The conditional expressions for Z_{dn} and Z_{wm} are as follows:

$$Z_{dn} = \begin{cases} true : & D_n = 0 \\ false : & D_n \neq 0 \end{cases} \quad (1a)$$

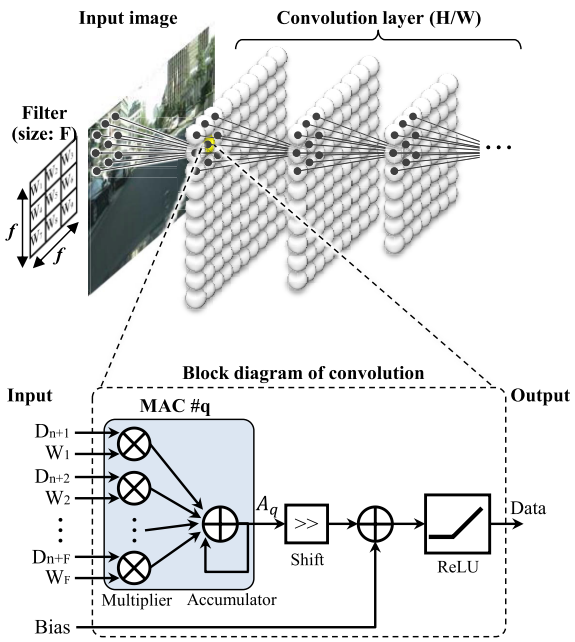


Fig. 1 CNN structure and block diagram of convolution operation for each filter

$$Z_{wm} = \begin{cases} true : & W_m = 0 \\ false : & W_m \neq 0 \end{cases} \quad (1b)$$

When the NOR circuits of Z_{dn} and Z_{wm} are true for all MAC operation units, the clock supply to the MAC operation is stopped. Logic circuits that determine the clock supply for the q-th MAC operation is expressed as GL_{Mq}. The conditional expression for GL_{Mq} to stop MAC is as follows:

$$Z_{Mq} = (Z_{d(n+1)}\ OR\ Z_{w1})\ AND\ (Z_{d(n+2)}\ OR\ Z_{w2})\ AND \cdots \cdots AND\ (Z_{d(n+F)}\ OR\ Z_{wF}) \quad (2)$$

This configuration reduces power consumption since the clock toggles of the flip-flop (FF) circuits in the MAC operation unit, as well as the signal transition toggles for data and weight, are disabled. However, the following two issues arise when using the zero-skipping technique in this configuration: MAC operation is not stopped frequently, and the GL circuit overhead is large.

The GL circuits consist of the K-bit NOR for each input and weight data, the 2-bit OR circuits for each multiplier, and the F-bit AND circuits for each accumulator. The GL circuit increases based on the number of parallel MAC operations that can be implemented in the device. Numerous parallel operations are required to improve processing speed, so the addition of GL circuit results in a very large circuit overhead of around 5% for the MAC circuit. The number of parallel circuits is P and the number of GL circuit input bits (AND, OR, etc.) are as follows:

$$\begin{aligned} B_{GL_{total}} &= P\,(B_{GL_d} + B_{GL_W} + B_{GL_M}) \\ &= P\{K + K + (2F + F)\} \end{aligned}$$

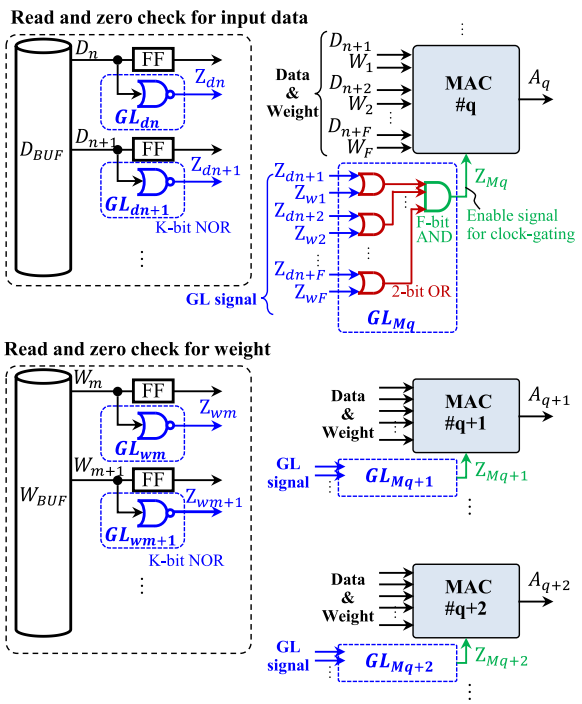


Fig. 2 Overview of clock-gating structure using zero-skipping for MAC operation circuit

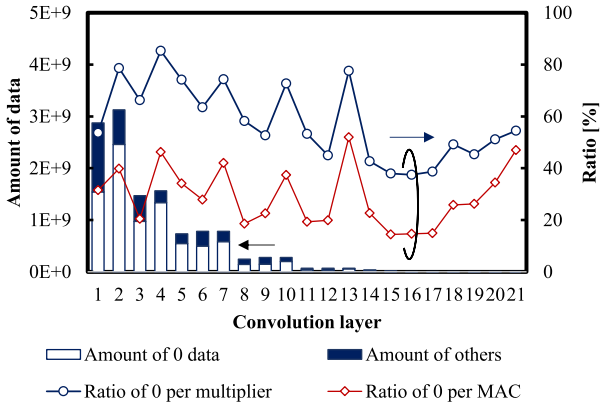


Fig. 3 Distribution of zero data ratio in each layer of SSD500 (data set: 500 cityscape images)

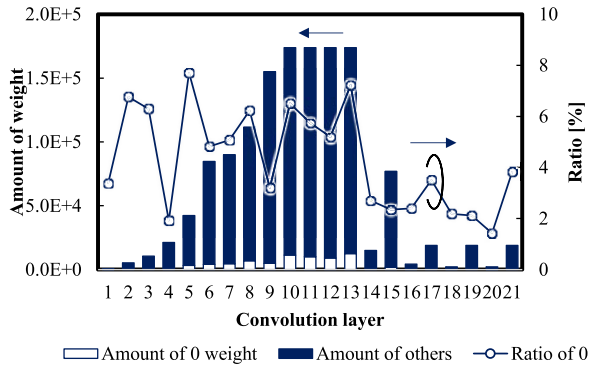


Fig. 4 Distribution of zero weight ratio in each layer of SSD500 (training data set: 3,000 cityscape images)

$$= P(2K + 3F) \tag{3}$$

where B_{GL_d}, B_{GL_W}, and B_{GL_M} are the number of GL_{dn}, GL_{wm}, and GL_{Mq} input bits, respectively, and $B_{GL_{total}}$ is the circuit overhead using zero-skipping technique. The addition of GL circuits negates the reduction in power consumption. Therefore, it is necessary to stop operation units more frequently and implement a clock-gating function with a small circuit overhead in order to further reduce power consumption.

Figure 3 shows the ratio of the input data to zero in the each layers 1–21 in the single shot multi-box detector 500 (SSD500) network model using 500 images of Cityscape dataset. "Ratio of 0 per multiplier" means the ratio of 0 in the input data to each multiplier, and "Ratio of 0 per MAC" means the ratio that all the inputs data ($D_{n+1} \sim D_{n+F}$ in Fig. 1) to MAC are 0 in the each layers. In zero skipping, the MAC operation is stopped when the all multiplication result of the input data and weight is zero. As shown in the figure, the average rate of MAC stopping is about 30% (red line), despite the 50–60% ratio of zero per input data (blue line). Future developments may enable more operations to be stopped.

On the other hand, Fig. 4 shows the ratio of the weight to zero for each layers 1–21 in the SSD500 network model.

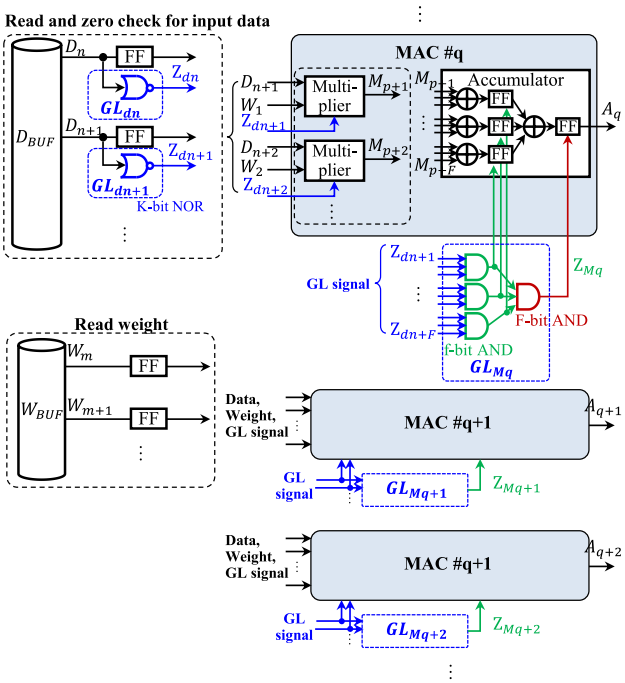


Fig. 5 Proposed precisely zero-skipping clock-gating structure for MAC operation

These weight values were trained with 3,000 images of Cityscape dataset. As shown in the figure, the average ratio of weight to 0 is about only 5% (blue line). The CNN model compressed for implementation in edge device has lower ratio of zero weight than before compression, this is because the compression process preferentially removes the weights that are not calculated (that is, when they are zero).

In the next subsection, we describe the proposed techniques that efficiently conserve the power consumption of the compressed CNN model with above features.

2.2 Proposed Precise Zero-Skipping Method

In conventional zero skipping, many OR circuits are required for Z_{dn} and Z_{wm}, as shown in Eq. (2), which increases the circuit overhead. If the OR circuits are removed and only Z_{dn} is used, the stopping ratio of MAC is reduced by about 5% as shown in Fig. 4. Our proposed approach stops the multipliers using only Z_{dn} (as shown in Fig. 5) instead of stopping the MAC operation entirely. We propose a precise zero-skipping method that increases the stop ratio of multipliers to around 60%. The accumulators implement F (filter size) additions by incorporating multiple additions to the tournament formula. In order to stop the F-input accumulator precisely, we arranged a logic circuit that stops if the input data is all zeros in the first-stage adder unit. If all the first-stage adders are stopped, the second-stage adders can also be stopped, which is the same MAC stop condition for conventional zero-skipping.

As explained above, GL_{wm} was removed by judging the operation stop using only the input data, and the cir-

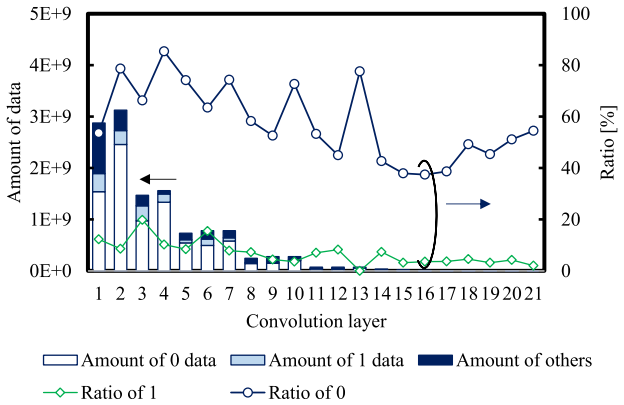


Fig. 6 Distribution of 0 and 1 input data ratio in each layer of SSD500 (data set: 500 cityscape images)

cuit overhead of GL_{Mq} was reduced. The operation unit was stopped more precisely than when using the conventional zero-skipping. All multipliers with input data zero were stopped, and the accumulators were also stopped if the input data was zero in the adder unit. By implementing CNNs using these techniques, power consumption and circuit overhead can be significantly reduced. So the number of GL circuit input bits (AND, OR, etc.) are as follows:

$$\begin{aligned} B_{GL_{total}} &= P\,(B_{GL_d} + B_{GL_M}) \\ &= P\{K + (F + F)\} \\ &= P(K + 2F) \end{aligned} \tag{4}$$

For example, when the number of bits of input data (K) is 8 and the filter size (F) is 9, the number of GL input bits can be reduced by about 40% according to Eqs. (3) and (4).

2.3 Active Data-Skipping Clock-Gating Method

To further reduce power consumption, we propose another method that slightly sacrifices recognition accuracy. As shown in Fig. 6, the ratio of input data of 1 occupies approximately 10% (green line). Therefore, if the multiplier is stopped not only input data of 0 but also at 1, the power consumption can be further reduced. By using only the upper k bits of the K-bit input data to stop computation, the GL_{dn} circuit overhead and power consumption are reduced to k/K. However, the result of the convolution operation will contain errors to some extent. Multipliers and accumulators with input data below threshold D_{th} are stopped, so the output of the MAC operation is zero under this condition. The D_{th} can be expressed as following;

$$D_{th} = 2^{K-k} - 1 \quad (0 \le k \le K) \tag{5}$$

After setting D_{th}, the CNN model is retrained to mitigate the effects of calculation errors. In addition, as shown in Fig. 7, clock gating is disabled by EN_{Layer} signal at layers with a small amount of input data and few zero data values (e.g. layer 11–21), since power consumption is not expected to be reduced. In this paper, we call this technique active data-skipping clock gating method, and in the next section, we discuss the dependence of k (or D_{th}) on accuracy degradation and the effects of reduced power consumption.

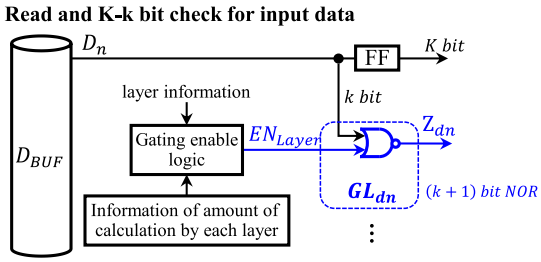


Fig. 7 Proposed active data-skipping for reading and k bit checking for n-th input data

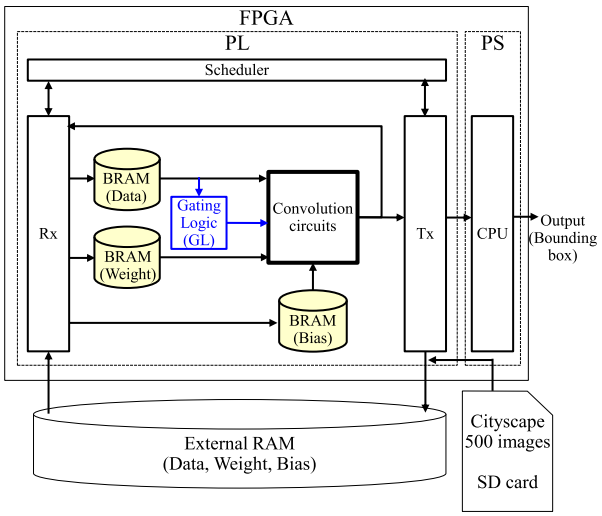


Fig. 8 Architecture of SSD500 implemented on FPGA (Xilinx ZU9)

3. Implementation and Evaluation Result

3.1 Implementation of CNN and Proposed Techniques on FPGA

This section describes the conditions in which the CNN is implemented and the configuration of the MAC operation circuit using the proposed techniques. We selected Xilinx ZU9 FPGA as the edge device for processing CNN calculations in this study. This device is assuming advanced driver assistance systems (ADAS) and autonomous driving (AD) an electrical control unit (ECU), for example. SSD500 was used as the CNN model for object detection with $F = 3 \times 3$ filter, and 88% of the channels were pruned for implementation in the ZU9 circuit resource. In addition, the model size was compressed by quantizing the input data and weight from a floating point to 8-bit integers (K = 8). The calculation was reduced from 182 GOP to 22 GOP. To train the model, 3,000 images from the Cityscapes dataset were used, and 500 images were used to evaluate recognition accuracy.

Figure 8 shows the architecture of SSD500 network model implemented on ZU9. The convolution process that

requires high-speed processing is implemented in the programable logic (PL) section. The transmitter (Tx) and the receiver (Rx) for exchanging input data, calculated data and weight data with external RAM are also implemented here. In addition, the scheduler that manages these circuits is also implemented. The calculation of the position of the bounding box of SSD500 is processed by the CPU of the processing system (PS) section. In order to implement a large-scale neural network with limited circuit resources, all calculations of convolution layer could not be done simultaneously in PL section. For this reason, the parallel number of the MAC circuit was 512, so the weight and bias data were read out layer by layer or channel by channel, and the image data was read out every 612 pixels as shown in Fig. 9.

Table 1 shows each circuit utilization of digital-signal-processor (DSP), look-up-table (LUT) and flip-flop (FF) of implemented SSD500 network model on PL part of ZU9 without proposal gating logic in this paper. Almost of the DSP circuits are used for multiplier in convolution process. Since the DSP include two multipliers, the 2,304 DSPs have 4,608 multipliers, which in 512 MACs. The FF circuit resources use less than 30%, but the LUT uses nearly 70%, of which about 90% is used for convolution circuit. That is, most of the power consumption of FPGA is occupied by convolution circuit. The power consumption of the convolution calculation circuits was 9.2 W (54% of total) as shown in Table 2.

Figure 10 shows the MAC operation and the proposed GL circuit implemented in the convolution operation shown in Fig. 8. An FF was also installed in the GL to match the timing with the FF inside the MAC operation. To stop each operation by clock gating, the clock buffer (CB) was stopped by using the logic output from the GL.

The operating clock of ZU9 for convolution operation was 300 MHz, and the frame rate was 110 fps, so the SSD500 processing speed of ZU9 was 2,420 GOP per second (GOPS). The operation performance per power consumption of only the convolution circuit is 263.0 GOPS/W.

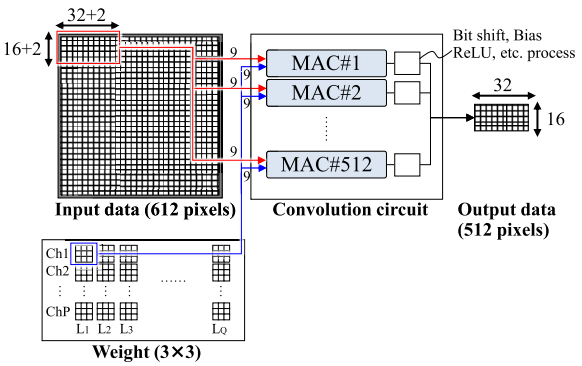


Fig. 9 Data flow processed of convolution circuit in one cycle

Table 1 SSD500 circuit utilization of programable logic (PL) part of Xilinx ZU9 (without proposal gating logic)

Circuit category	DSP	LUT	FF
Control circuit*	0 / 0%	18,354 / 6.7%	62,801 / 11.5%
BRAM	0 / 0%	784 / 0.3%	5,472 / 1.0%
Convolution circuit	2,304 / 91%	165,223 / 60.3%	103.905 / 19.0%
Blank (unused)	216 / 9%	89,719 / 32.7%	375,982 / 68.6%

* Control circuit : Tx, Rx, Scheduler in Fig. 8

Table 2 Breakdown of power consumption of SSD500 circuit implemented on Xilinx ZU9 (without proposal gating logic)

Circuit category		Power consumption	Ratio
PL	Control circuit*	1.8W	10.7%
	BRAM	2.8W	16.5%
	Convolution circuit	9.2W	54.4%
PS		3.1W	18.4%
Total		16.9W	---

* Control circuit : Tx, Rx, Scheduler in Fig. 8

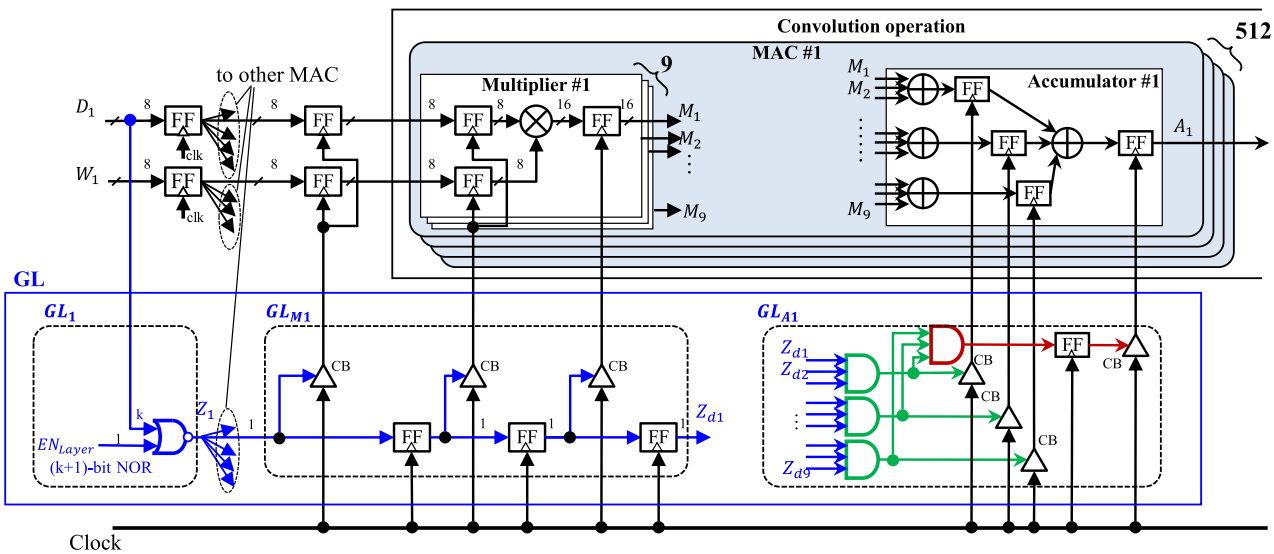
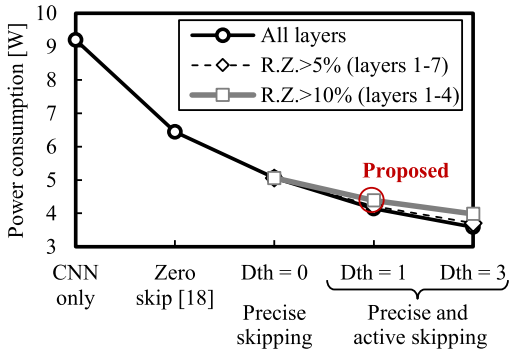
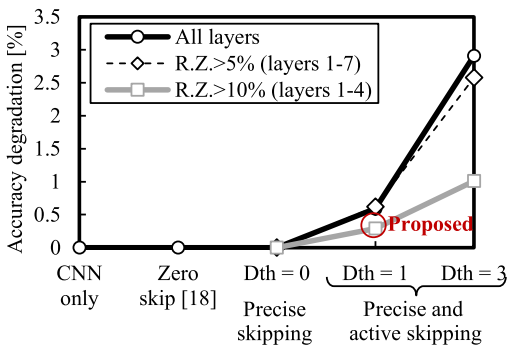


Fig. 10 Implementation of MAC and proposed GL circuit for reduced power consumption and overhead



(a) Measurement results of power consumption of convolution circuits



(b) Simulation results of recognition accuracy degradation

Fig. 11 Relationship between gating threshold and power consumption, and accuracy degradation for gating logic

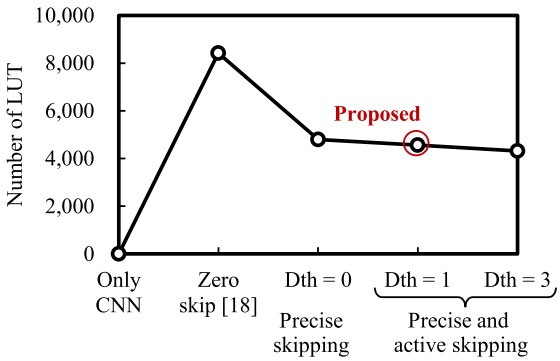


Fig. 12 Relationship between gating threshold and circuit overhead for LUT

3.2 Evaluation Results

Figure 11 shows the relationship between the threshold of input data for stopping computation, power consumption, and recognition accuracy degradation. The power consumption is the result of measuring the current in the power supply of the logic circuit system on the evaluation board of ZU9. We assumed that the logic circuit occupies 80% of the measured current. The ratio of the power consumption of the logic circuit to the convolution circuit was calculated using the FPGA design tool "Vivado". The conventional zero-skipping circuit, which is implemented equivalent to [18] for example, consumed 6.4 W (reduced from 9.2 W which is the power consumption of convolution circuits only). Naturally, there is no recognition accuracy degradation. The number of GL circuits was 8,426 LUTs (with an overhead for convolution circuit of 5.1%) as shown in Fig. 12.

By applying the proposed precise zero-skipping method, the power consumption of the convolution calculation circuits was reduced to 5.1 W, and recognition accuracy was no degradation as shown in Fig. 11. Moreover, the circuit overhead was reduced to 2.9% (−43.1%) as shown in Fig. 12. We verified that the precise zero-skipping method is more effective in reducing power consumption and circuit overhead than the conventional zero-skip method.

Next, the results of applying both precise zero-skipping and active data-skipping are described. At D_{th} equal to 1 and 2 for all layers, the degradation of recognition accuracy was 0.59% and 2.91%, and the power consumption of the convolution calculation circuits was 4.1 W and 3.6 W, respectively. Next, we applied the methods to the layers in which the ratio of zero data (R.Z.) was >5% and >10% of the total zero data of all layers, that is, layers 1–7 and layers 1–4, respectively. When D_{th} was 1 and R.Z. was >10%, the accuracy degraded 0.29% and the power consumption of the convolution calculation circuits was 4.4 W (−52.3%). Furthermore, the circuit overhead was reduced to 2.7% (−45.9%) compared to when conventional zero-skipping circuit was used, as shown in Fig. 12 (c). The operation performance per power consumption of only the convolution circuit with precise and active skipping methods is 550.0 GOPS/W.

4. Conclusion

We proposed two methods, precise zero-skipping and active data-skipping, to reduce power consumption and circuit overhead for CNN-based edge devices. We selected Xilinx ZU9 FPGA assuming ADAS and AD ECU, as example of an edge device processing CNN, and implemented SSD500 on this device. By combining the two proposed methods, we verified reduced power consumption (−52.3%) and low circuit overhead (−45.9%) with only slight accuracy degradation (−0.29%) compared with the conventional zero-skipping method. The results indicate that our proposed methods are useful for processing CNNs by means of edge devices such as FPGA.

Acknowledgments

We thank K. Osada and M. Kudo Hitachi Automotive Systems, Ltd., for providing insights on the requirements and technology trends related to autonomous driving systems.

References

[1] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," Proc. Adv. Neural

Inf. Process. Syst., vol.25, pp.1097–1105, 2012.
[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," CVPR, pp.1–9, 2015.
[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," NIPS, 2015.
[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proc. IEEE Comput. Vis. Pattern Recognit., pp.770–778, June 2016.
[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A.C. Berg, "SSD: Single Shot Multibox Detector," ECCV2016, vol.9905, pp.21–37, 2016.
[7] S. Han, H. Mao, and W.J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," 2016, https://arxiv.org/abs/1510.00149 (online).
[8] J.H. Luo and J. Wu, "An entropy-based pruning method for cnn compression," CoRR, abs/1706.05791, 2017.
[9] D. Murata, T. Motoya, and H. Ito, "Automatic CNN Compression System for Autonomous Driving," 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), pp.838–843, 2019.
[10] C. Farabet, C. Poulet, J.Y. Han, and Y. LeCun, "CNP: An FPGA-based processor for Convolutional Networks," Proc. IEEE Int. Conf. F. Program. Log. Appl., pp.32–37, 2009.
[11] C. Farabet, B. Martini, B. Corda, P. Akselrod, E. Culurciello, and Y. LeCun, "NeuFlow: A Runtime Reconfigurable Dataflow Processor for Vision," Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp.109–116, June 2011.
[12] P.-H. Pham, D. Jelaca, C. Farabet, B. Martini, Y. LeCun, and E. Culurciello, "NeuFlow: Dataflow vision processing system-on-a-chip," Proc. Midwest Symp. Circuits Syst., pp.1044–1047, 2012.
[13] F. Conti and L. Benini, "A Ultra-Low-Energy Convolution Engine for Fast Brain-Inspired Vision in Multicore Clusters," Proc. IEEE Des. Autom. Test Eur. Conf., pp.683–688, 2015.
[14] Z. Du, R. Fasthuber, T. Chen, P. Ienne, L. Li, T. Luo, X. Feng, Y. Chen, and O. Temam, "ShiDianNao: Shifting Vision Processing Closer to the Sensor," Proc. ACM/IEEE Int. Symp. Comput. Archit., vol.43, no.3S, pp.92–104, 2016.
[15] L. Cavigelli, M. Magno, and L. Benini, "Accelerating Real-Time Embedded Scene Labeling with Convolutional Networks," IEEE Design Automation Conf., pp.1–6, June 2015.
[16] Y. Lin, C. Sakr, Y. Kim, and N. Shanbhag, "PredictiveNet: An energy-efficient convolutional neural network via zero prediction," 2017 IEEE International Symposium on Circuits and Systems (ISCAS), pp.1–4, 2017.
[17] D. Kim, S. Kim, and S. Yoo, "FPGA Prototyping of Low-Precision Zero-Skipping Accelerator for Neural Networks," 2018 International Symposium on Rapid System Prototyping (RSP), pp.104–110, 2018.
[18] Y.H. Kim, G.J. An, and M.H. Sunwoo, "CASA: A Convolution Accelerator using Skip Algorithm for Deep Neural Network," 2019 IEEE International Symposium on Circuits and Systems (ISCAS), pp.1–5, 2019.

Akira Kitayama received B.S. and M.S. degrees in engineering from Kyoto Institute of Technology, Kyoto, Japan, in 2005 and 2007. In 2007, he joined the Central Research Laboratory, Hitachi Ltd., Tokyo, Japan, where he has been engaged in research and development of RF circuits, transmission lines that transfer signals with GHz-order frequency, signal processing for automotive mm-wave radars, and embedded AI technique for edge device such as automotive ECU etc. He is a researcher with Hitachi, Ltd., Research & Development Group, Center for Technology Innovation - Electronics, Tokyo, Japan.

Goichi Ono received the B.S. degree in electrical and industrial engineering and the M.S. degree in materials engineering from Hiroshima University, Hiroshima, Japan, in 1996 and 1998, respectively, and received Ph.D. degrees in System informatics from the Kobe University in 2016. In 1998, he joined the Central Research Laboratory, Hitachi Ltd., Tokyo, Japan, where he had been engaged in the research and development of high-speed and low-power CMOS circuit techniques for microprocessors. In the field of wireless systems, from 2003 to 2006, he and his team developed wireless sensor network based on the Ultra-wideband. From 2007, he engaged in the research of high-speed wireline communication technology for 100-Gigabit Ethernet. From 2010 to 2015, he had been working on research of low power SoC in advanced process technology. From 2016, he is working on research of DNN compression techniques.

Tadashi Kishimoto received B.S. and M.S. degrees in engineering from Kyoto University, Kyoto, Japan, in 2016 and 2018. In 2018, he joined the Central Research Laboratory, Hitachi Ltd., Tokyo, Japan, where he has been engaged in research and development of embedded AI technique for edge device such as automotive ECU etc. He is a researcher with Hitachi, Ltd., Research & Development Group, Center for Technology Innovation - Electronics, Tokyo, Japan.

Hiroaki Ito received B.S. and M.S. degrees in electronic information engineering from Yokohama National University, Kanagawa, Japan, in 1990 and 1992. In 1992, he joined Hitachi Ltd., Tokyo, Japan, where he has been engaged in research and development of image compression and decompression technology, and embedded image quality improvement technology. In 2017, he joined Hitachi automotive systems Ltd., Ibaraki, Japan where he has been engaged in development of autonomous driving system.

Naohiro Kohmu was born in Kobe, Japan, in 1988. He received the B.S. and M.S. degrees in electrical engineering from Osaka University, Osaka, Japan in 2012 and 2014, respectively. In April 2014, he worked with the Central Research Laboratory, Hitachi, Ltd., where he was involved in developing high-speed optical/electrical signal transmission and sensing technique. From 2014 to 2018, he developed >25-Gbit/s/ch high-speed wireline techniques. He is currently developing photoacoustic technique for sensors and spectro-photometers. Mr. Kohmu is a member of the IEICE, the JSAP, the JIEP and the IEEE. He was the recipient of the IEICE Young Researcher's Award in 2013 and the JSAP Outstanding Achievement Award in 2014.