BRIEF PAPER

Exploiting Sparse Activation for Low-Power Design of Synchronous Neuromorphic Systems

Jaeyong CHUNG[†a)], ***Member and*** **Woochul KANG**[††b)], ***Nonmember***

SUMMARY Massive amounts of computation involved in real-time evaluation of deep neural networks pose a serious challenge in battery-powered systems, and neuromorphic systems specialized in neural networks have been developed. This paper first shows the portion of active neurons at a time dwindles as going toward the output layer in recent large-scale deep convolutional neural networks. Spike-based, asynchronous neuromorphic systems take advantage of the sparse activation and reduce dynamic power consumption, while synchronous systems may waste much dynamic power even for the sparse activation due to clocks. We thus propose a clock gating-based dynamic power reduction method that exploits the sparse activation for synchronous neuromorphic systems. We apply the proposed method to a building block of a recently proposed synchronous neuromorphic computing system and demonstrate up to 79% dynamic power saving at a negligible overhead.

key words: *neuromorphic systems, VLSI design, computer-aided design*

1. Introduction

Recent developments in neural network (a.k.a., deep learning) [1] are tackling the problems successfully that only humans have been solved well. Deep learning is now becoming an essential tool for various cognitive applications such as image understanding, search, voice recognition, natural language processing, and self-driving cars. Deep learning for real-world applications uses huge neural networks, which often contain millions of parameters (e.g., 160M [2]). The amount of computation to evaluate such a neural network in real-time is also tremendous, consuming significant power. This is a critical problem in battery-powered systems. CPUs cannot deliver the performance necessary for real-time evaluation, and mobile GPUs and FPGA-based accelerators [3] consume more than 10W, which is not adequate for mobile devices such as phones and wearables. Also, for always-on devices, we may have more stringent power requirements (e.g., <1W). Thus, there have been research efforts that seek for a radically different computing platform from the Von Neumann architecture-based systems.

Neuromorphic engineering, started in 1980s, is a concept that mimics neuro-biological architectures in the nervous system using analog circuits, but the "neuromorphic" term is being used broadly for circuits and systems that implement models of neural systems. Neuromorphic computing systems may come to the rescue to address the power and performance issues of the traditional computing platforms. However, recent large-scale neuromorphic systems such as BrainScaleS [4] and Neurogrid [5] are developed for a different goal that accelerates the brain simulation and scales it up to the human-level. They are mainly used for neuroscience research. Unlike these two systems, TrueNorth [6], a digital neuromorphic system from IBM, aims at real-world applications as well as the brain simulation [7]. TrueNorth employs a spiking neuron model and uses mixed synchronous-asynchronous design [8], which allows it to consume little power when neurons are not activated. INSight [9] is a recently proposed neuromorphic computing system that is designed specifically for deep learning-based applications. This system adopts a fully synchronous design so its implementation is fully supported by modern design automation tools. However, synchronous neuromorphic systems such as INSight may suffer from large amounts of dynamic power consumption due to clocks.

In this paper, we propose a low-power design method for digital synchronous neuromorphic systems. This method is originally developed for neuromorphic systems, but it can be applied to any synchronous processors that can evaluate neural networks. The contributions of this paper are summarized as follows.

- We show why asynchronous circuits have emerged again in neuromorphic computing by analyzing large-scale deep convolutional neural networks.
- The proposed design method allows even synchronous systems to reduce power consumption substantially depending on the activations of neurons, as in spike-based, asynchronous systems.
- We demystify the conventional belief that synchronous systems waste large amounts of power due to clocks.

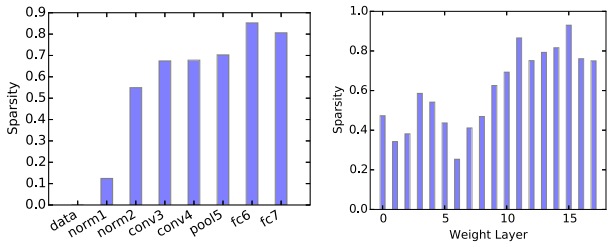
2. A Low-Power Design Technique for Synchronous Neuromorphic Systems

Asynchronous circuits consume dynamic power only when they work, whereas synchronous circuits are often considered to waste much power even if they do not perform useful work, due to clocks. However, it may be not the case considering clock gating. Clock gating is a low-power circuit design technique that deals with the power issue of

Manuscript received April 13, 2017.
Manuscript revised June 8, 2017.

[†]The author is with the Department of Electronic Engineering, Incheon National University, Incheon, Korea.
[††]The author is with the Department of Embedded System Engineering, Incheon National University, Incheon, Korea.
a) E-mail: jychung@inu.ac.kr
b) E-mail: wckang@inu.ac.kr (Corresponding author)

DOI: 10.1587/transele.E100.C.1073



(a) AlexNet (b) VGG-19

Fig. 1 (a) The sparsity for the output activation of each layer in AlexNet is measured for 256 images. As going toward the output layer, the sparsity increases, and near the output layer, it is as high as 0.8. (b) The same trend is confirmed in a current state-of-the-art deep neural network.

clocks, and it is well supported by modern design automation tools. These tools automatically extract the 'enable' signal for each flop (if exists) and for a group of flops that share the same 'enable' signal, replace the original clock with a new clock gated by the 'enable' signal. This shields a large amount of clock loads in clock trees and eliminates unnecessary switching at clock pins. This transformation is usually very effective and is performed at a fine-grained level (e.g., even for ≤ 10 flops). In digital signal processing circuits, the 'enable' signals often become 'data valid' signals.

2.1 Sparse Activation in Neural Networks

Nonetheless, asynchronous circuits are still more power-efficient than clock-gated circuits in neuromorphic computing systems owing to sparse activation. The Rectified Linear Unit (ReLU) $f(x) = max(0, x)$ is the most common choice for the activation function in recent deep neural networks. Since it produces exact zero when $x < 0$, the activations of a layer often have many zeros (i.e., it is sparse). Also, the activations tend to get sparser as going up toward the output layer. Figure 1 (a) shows the average sparsity of the activations of each layer in AlexNet for the first 256 input images in ILSVRC 2012 validation data set. It is clear that the sparsity increases toward the output layer, and it is very high near the output layer. Figure 1 (b) shows the same trend of VGG-19 [2], which is a most recent CNN that has produced the state-of-the-art results for ILSVRC and has 19 (weight) layers and 160M parameters.

The sparseness in the activations is naturally exploited in spike-based, asynchronous circuits. "No activation" is usually encoded into zero spikes, so it does not consume dynamic power at all. However, this can also be exploited even in synchronous circuits by extending clock gating.

2.2 Data-Driven Clock Gating

Figure 2 depicts the concept of the proposed method. The clock for a processing element can be gated by detecting "no activation". In this approach, it is important to keep the nonzero activation detector small for the overhead not to overtake benefits. We show how this approach is realized in

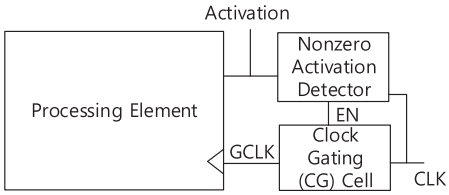


Fig. 2 The clock for the processing element is gated depending on the activations of neurons.

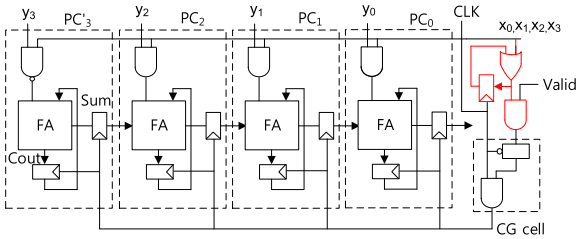


Fig. 3 The clock pins of the registers in the primitive cells are not switching for the leading zeros of the input activation by adding a simple circuit, which reduces the dynamic power consumption significantly.

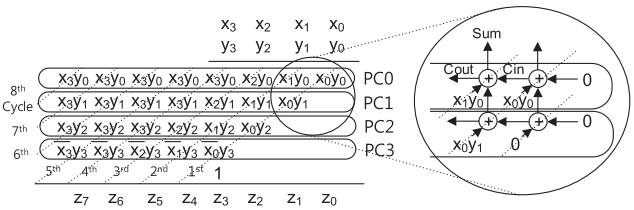


Fig. 4 In the semi-systolic multiplier, each primitive cell is dedicated to operations associated with a partial product.

INSight neuromorphic computing system [9]. INSight is a digital, synchronous neuromorphic system where synapses are a main processing element. In [9], the synapses take up more than 80% of the chip area, and their power consumption dominates the power consumption of the chip. In other words, the synapse design is replicated a number of times and reducing its power leads to a significant reduction of the chip power. Figure 3 shows the synapse design based on the semi-systolic multiplier. Let x and y be the activation of a neuron and the weight of a synapse represented in n-bit binary, respectively. The i-th bit of x (y) is denoted by x_i (y_i). Each bit (digit) of x is coming in to the input of the synapse serially least significant bit (LSB) first. The semi-systolic multiplier is based on a primitive cell, which consists of two registers, a two-input AND gate, and a full adder. A n-bit semi-systolic multiplier is simply a composition of the n primitive cells. We denote the primitive cells by $PC_0, \ldots, PC_{n-1}$. The synapse weight is provided in parallel to the synapse and PC_i takes y_i. This multiplier is called "semi-systolic" because a global wire for the input is needed, which is not like systolic systems where neighbor processing elements are interconnected regularly via local wires. The long wire may not be adequate for high frequency operations, but it enables more compact design, which fits to the neuromorphic system.

Figure 4 depicts how a 4-bit semi-systolic multiplier

generates partial products and adds them up. Each primitive cell is dedicated to a partial product and the digits on a dotted, diagonal line are generated at the same cycle. The multiplier produces each bit of the product at a time, adding the partial products up gradually. The figure in the circle shows how the partial products are added. The full adder of PC_i takes the 1-bit product of the current cycle as one input. The other two inputs come from the previous cycle via the flops; one is from the same primitive cell PC_i and the other is from PC_{i+1}. For signed multiplication, the multiplicand x needs to be signed-extended for last 4 cycles and PC_{n-1} needs a modification since the sign bit of y has the weight of -2^{N-1}. Thus, we initialize the Cout register to 1 and replace the AND gate by a NAND gate. This modified PC is denoted by PC'. This subtracts the last partial product. The registers in the semi-systolic multiplier maintain the initial values until the first '1' comes in from the input. For the leading '0's before the first '1', the Cout register's value in PC'_{n-1} remains to '1' because the full adder keeps generating a carry. Thus, for the leading '0's, it is not necessary to load the new values to the registers, and clock gating is possible. Figure 3 shows the logic added for this data-driven clock gating in red. The overhead is marginal; only two extra combinational gates and one extra register can detect the clock-gating condition.

Table 1 Design statistics

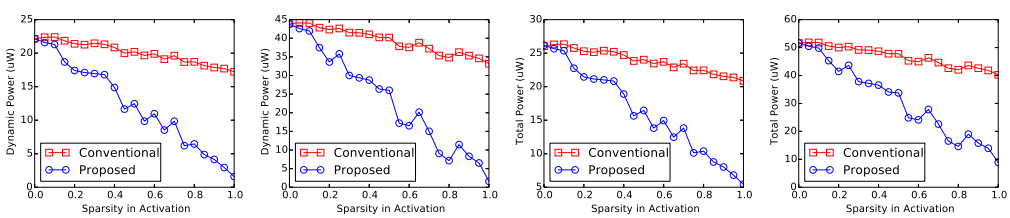
Metric	Word size			
	8-bit		16-bit	
	Baseline	Proposed	Baseline	Proposed
Critical path delay	0.73ns	0.73ns	0.78ns	0.78ns
Cell area (um^2)	210.18	218.06	405.36	413.24
Comb. cell count	39	39	72	72
Seq. cell count	17	18	33	34
Gated reg.	15	16	31	32

3. Experimental Results

We implement the baseline design of the n-bit semi-systolic multiplier and the proposed design in Verilog. Note that both of the designs are clock-gated. They are synthesized using Synopsys Design Compiler and mapped into a 32nm technology using a high V_t library characterized at 0.95V, 125°C and SS corner. Clock gating is automatically performed during the synthesis. The results of the synthesis are summarized in Table 1. The baseline design has $2n-1$ registers for the primitive cells and extra 2 registers. We eliminate the sum register for PC_0. One extra register is used for the sign extension of the serial input, and the other is the latch in the clock gating cell. The $2n-1$ registers are clock-gated correctly. The register for the sign extension is not clock-gated because it has a different enable condition from that of the others. The combinational gate count is high be-

Table 2 The proposed design saves 65% dynamic power compared to the baseline when the sparsity is 0.8.

Spar sity	Power (μW)	Baseline (8-bit)				Proposed (8-bit)				Red
		clk	seq	comb	total[a]	clk	seq	comb	total[b]	b/a
0.2	Internal	1.1	16.3	1.5	19.1	1.1	12.8	1.5	15.5	0.81
	Switching	1.5	0.2	0.5	2.3	1.2	0.2	0.5	1.9	0.83
	Leakage	0.1	2.3	1.5	3.9	0.1	2.4	1.5	4.0	1.04
	Dynamic	2.7	16.5	2.0	21.4	2.3	13.1	2.1	17.4	**0.81**
	Total	2.8	18.8	3.5	25.3	2.4	15.5	3.5	21.5	0.85
0.6	Internal	1.3	15.5	1.0	17.8	0.9	7.9	1.0	9.8	0.55
	Switching	1.5	0.1	0.4	2.1	0.7	0.1	0.3	1.2	0.56
	Leakage	0.1	2.2	1.5	3.8	0.1	2.4	1.5	4.0	1.04
	Dynamic	2.9	15.7	1.4	19.9	1.6	8.0	1.4	11.0	**0.55**
	Total	3.0	17.9	2.8	23.7	1.8	10.4	2.8	14.9	0.63
0.8	Internal	1.3	15.0	0.6	16.8	0.8	4.5	0.6	5.8	0.35
	Switching	1.5	0.1	0.3	1.9	0.4	0.1	0.2	0.6	0.33
	Leakage	0.1	2.2	1.4	3.7	0.1	2.4	1.4	3.9	1.05
	Dynamic	2.9	15.0	0.8	18.7	1.2	4.5	0.8	6.5	**0.35**
	Total	3.0	17.2	2.3	22.4	1.3	6.9	2.2	10.4	0.46



(a) 8-bit (Dynamic) (b) 16-bit (Dynamic) (c) 8-bit (Total) (d) 16-bit (Total)

Fig. 5 The proposed method allows synchronous neuromorphic systems to save the dynamic power when neurons are not active, closing the gap between synchronous neuromorphic systems and asynchronous ones.

cause flip-flops with synchronous reset are mapped into an AND gate and a simple D flip-flop for better timing. Thus, it is possible to reduce area more if we design at the gate level. The proposed design just adds one more register to the baseline design (the combinational logic is mapped slightly differently) and the difference in the cell area is marginal.

We randomly generate 20 sets of 1000 vectors and the sparsity of the vector sets varies from 0.0 to 1.0 with 0.05 steps. For each synthesized netlist, we perform gate-level simulation using these vector sets using Synopsys VCS and generate SAIF (Switching Activity Interchange Format) files. The switching activities in each SAIF file are back annotated to the netlist and dynamic power analysis is performed in Design Compiler, yielding a power report for each vector set. Table 2 shows the results when the sparsity is 0.2, 0.6, and 0.8. As shown in Fig. 1, the sparsity of the second layer and the 7th layer of AlexNet are around 0.2 and 0.8, respectively, so we can expect the corresponding results for each layer. Most of the power is consumed in the sequential logic due to short-circuit currents. In the 8-bit baseline design (when the input sparsity is 0.2), the dynamic power takes up 85% of the total power, and the static power occupies 15%. When the sparsity is 0.2, the 8-bit proposed design saves 19% dynamic power at the cost of 4% static power increase, yielding 15% total power saving. When the sparsity increases to 0.8, more significant improvements are made. The 8-bit proposed design saves 65% of the dynamic power of the baseline design and saves 54% of the total power. When the word size increases to 16-bit, the improvements are even more significant because clock gating hides more clock pins. The proposed design consumes 79% less dynamic power than the baseline design.

Figure 5 compares the dynamic and total power of the proposed design with those of the baseline design when the sparsity varies. Even in the baseline design, switching activities decrease as the sparsity increases, which in turn reduces power consumption. In the proposed design, the dynamic power consumption becomes very small as the sparsity approaches to 1.0, as in spike-based, asynchronous circuits.

4. Conclusion

We have proposed a low-power design method for synchronous neuromorphic computing systems exploiting sparse activation in deep neural networks. We have shown that it is possible for even synchronous circuits to save dynamic power depending on amount of the activation as in spiked-based, asynchronous circuits. Considering the simplicity of synchronous circuits and commercial tool support for them, synchronous neuromorphic computing systems can become very attractive with the proposed design method. The proposed method has been demonstrated in a digital synapse design based on the semi-systolic multiplier, but it can be easily extended to other type of digital circuits, which may be our future work.

Acknowledgments

This work was supported by Incheon National University Grant in 2015. The EDA tools used in this work were supported by IDEC.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol.521, no.7553, pp.436–444, 2015.
[2] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556, 2014.
[3] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing fpga-based accelerator design for deep convolutional neural networks," Proc. 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, pp.161–170, ACM, 2015.
[4] J. Schemmel, D. Briiderle, A. Griibl, M. Hock, K. Meier, and S. Millner, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," Proc. 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp.1947–1950, 2010.
[5] B.V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A.R. Chandrasekaran, J.-M. Bussat, R. Alvarez-Icaza, J.V. Arthur, P.A. Merolla, and K. Boahen, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," Proceedings of the IEEE, vol.102, no.5, pp.699–716, 2014.
[6] P.A. Merolla, J.V. Arthur, R. Alvarez-Icaza, A.S. Cassidy, J. Sawada, F. Akopyan, B.L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S.K. Esser, R. Appuswamy, B. Taba, A. Amir, M.D. Flickner, W.P. Risk, R. Manohar, and D.S. Modha, "A million spiking-neuron integrated circuit with a scalable communication network and interface," Science, vol.345, no.6197, pp.668–673, 2014.
[7] A.S. Cassidy, R. Alvarez-Icaza, F. Akopyan, J. Sawada, J.V. Arthur, P.A. Merolla, P. Datta, M.G. Tallada, B. Taba, A. Andreopoulos, A. Amir, S.K. Esser, J. Kusnitz, R. Appuswamy, C. Haymes, B. Brezzo, R. Moussalli, R. Bellofatto, C. Baks, M. Mastro, K. Schleupen, C.E. Cox, K. Inoue, S. Millman, N. Imam, E. Mcquinn, Y.Y. Nakamura, I. Vo, C. Guok, D. Nguyen, S. Lekuch, S. Asaad, D. Friedman, B.L. Jackson, M.D. Flickner, W.P. Risk, R. Manohar, and D.S. Modha, "Real-time scalable cortical computing at 46 giga-synaptic ops/watt with ~100x speed up in time-to-solution and ~100,000x reduction in energy-to-solution," In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp.27–38, 2014.
[8] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, G.-J. Nam, B. Taba, M. Beakes, B. Brezzo, J.B. Kuang, R. Manohar, W.P. Risk, B. Jackson, and D.S. Modha, "Truenorth: Design and tool flow of a 65mw 1 million neuron programmable neurosynaptic chip," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol.34, no.10, pp.1537–1557, 2015.
[9] J. Chung, T. Shin, and Y. Kang, Insight: A neuromorphic computing system for evaluation of large neural networks, arXiv preprint arXiv:1508.01008, 2015.