Superconductor Computing for Neural Networks

Koki Ishida, Department of Advanced Information Technology, Kyushu University, Fukuoka, 464-8601, Japan

Ilkwon Byun, Department of Electrical and Computer Engineering, Seoul National University, Gwanak-gu, 08826, South Korea

Ikki Nagaoka, Department of Electronics, Nagoya University, Nagoya, 464-8601, Japan

Kosuke Fukumitsu, Department of Advanced Information Technology, Kyushu University, Fukuoka, 464-8601, Japan

Masamitsu Tanaka, Department of Electronics, Nagoya University, Nagoya, 464-8601, Japan

Satoshi Kawakami, Teruo Tanimoto, and Takatsugu Ono, Department of Advanced Information Technology, Kyushu University, Fukuoka, 464-8601, Japan

Jangwoo Kim, Department of Electrical and Computer Engineering, Seoul National University, Gwanak-gu, 08826, South Korea

Koji Inoue, Department of Advanced Information Technology, Kyushu University, Fukuoka, 464-8601, Japan

The superconductor single-flux-quantum (SFQ) logic family has been recognized as a promising solution for the post-Moore era, thanks to the ultrafast and low-power switching characteristics of superconductor devices. Researchers have made tremendous efforts in various aspects, especially in device and circuit design. However, there has been little progress in designing a convincing SFQ-based architectural unit due to a lack of understanding about its potentials and limitations at the architectural level. This article provides the design principles for SFQ-based architectural units with an extremely high-performance neural processing unit (NPU). To achieve our goal, we developed and validated a simulation framework to identify critical architectural bottlenecks in designing a performance-effective SFQ-based NPU. We propose SuperNPU, *which outperforms a conventional state-of-the-art NPU by 23 times in terms of computing performance and 1.23 times in power efficiency even with the cooling cost of the 4K environment.*

We are about to enter an era where both Moore's law and Dennard scaling do not hold anymore. We are running out of effective options to improve the performance of CMOS-based computer systems while maintaining their power and temperature budgets.[1] Therefore, we believe that now is the right time to exploit emerging device technologies with significant potential and make a serious effort to improve their feasibility.

Among the several candidates, the superconductor single-flux-quantum (SFQ) logic family[2] has emerged as a highly promising solution for the post-Moore era. SFQ technology, which utilizes superconductor devices operating at 4K, has significant potential for both high performance and energy efficiency. Specifically, SFQ logic gates use low-voltage impulse-shaped signals for their operations and achieve the ultrafast ($\sim 10^{-12}$ s) and low-energy ($\sim 10^{-19}$ J)

Digital Object Identifier 10.1109/MM.2021.3070488
Date of publication 5 April 2021; date of current version 25 May 2021.

switching. By focusing on these strong points, many researchers have contributed to SFQ-related research in various aspects, especially in device and circuit design.

However, little research has been conducted on SFQ-based architectures[3] due to a lack of understanding about its architecture-level potentials and limitations. As we show later, SFQ logic, with its unique pulse-driven nature, requires completely different architecture designs from conventional CMOS technology. In consideration of SFQ-specific architectural tradeoffs, the following questions must be clearly addressed. 1) Which architecture is most promising for this technology? 2) How can we implement various microarchitectural units in an SFQ-friendly manner? 3) How can we fully exploit its potential at the architecture level while considering its limitations? 4) How do we simulate and validate SFQ architectural units?

Our article, presented at MICRO'20, provides straightforward answers to the above questions in the form of *SuperNPU*, our design for an SFQ-based neural processing unit (NPU). The main contributions of this work are as follows. 1) Architecting an SFQ-based NPU: to the best of our knowledge, this is the first work to design an NPU, which addresses the architectural tradeoffs of SFQ technology. 2) Simulation framework: it is also the first work to model and validate a simulator for SFQ-based architectures. 3) SFQ-specific architectural optimizations: we identify critical architectural bottlenecks and optimizations, which can cause a performance variance. 4) Significant results: SuperNPU provides extremely high performance and power efficiency; it outperforms a conventional design by 23 times and 1.23 times in terms of performance and power efficiency even with the cooling cost of the 4K environment. Our thorough architectural analysis with a validated modeling tool clearly shows the impact of our SFQ-specific optimization process and the potential of SFQ computing as a post-Moore-era solution.

BACKGROUND ON SFQ LOGIC TECHNOLOGY

SFQ circuits utilize a small-voltage pulse as an information carrier, which can be stored as a single magnetic flux quantum (SFQ) in a superconductor ring [Figure 1(a)]. To store and transfer the SFQ, the superconductor ring contains superconducting devices called Josephson junctions (JJs). Each JJ consists of a thin insulator sandwiched by the superconductors, and it has unique electrical characteristics with which to generate a voltage pulse [Figure 1(b)].

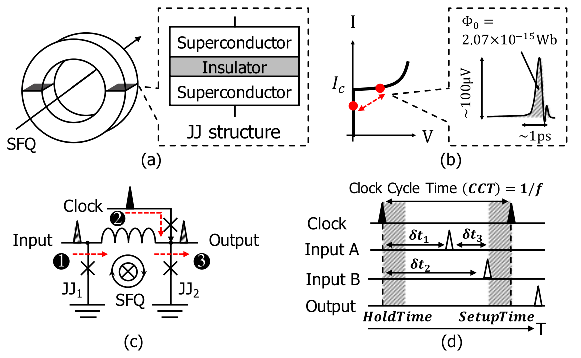

FIGURE 1. Circuit elements and working principle of SFQ logic technology. (a) Superconductor ring with SFQ. (b) Electrical characteristics of JJ. (c) Circuit diagram of an SFQ-based DFF. (d) Determination of SFQ circuits' frequency with an operation example of a two-input AND gate.

Figure 1(c) shows the working principle of SFQ logic gates with an SFQ-based delay flip flop (DFF), which consists of a single superconductor ring and a clock line. First, when the input pulse enters the ring, it is stored in the ring as an SFQ by switching JJ$_1$ (❶). Next, by receiving a clock pulse (❷), JJ$_2$ is activated, and the stored SFQ is transferred to the output as a voltage pulse (❸). In this manner, SFQ gates can represent the logical value "1" (or "0") by the existence (or absence) of the stored SFQ between the two adjacent clock pulses.

The SFQ logic gates are implemented using storage rings and pulse interactions in a similar way. Figure 1(d) shows an operation example of a two-input AND gate. The SFQ circuits' frequency is determined by

$$f = 1/\mathrm{CCT} = 1/(\mathrm{SetupTime} + \mathrm{Max}(\delta t_1, \delta t_2))$$

where SetupTime is the timing constraint, and δt_1 and δt_2 are the timing gaps between the input pulses and the first clock pulse arrival. $\delta t_1, \ \delta t_2 \geq$ HoldTime must be satisfied, where HoldTime is the other timing constraint. δt_1 and δt_2 can be shortened by delaying the clock pulse arrival. Therefore, the timing gap between the two inputs' arrivals (δt_3) is important. If there is a large difference between the arrivals, the clock frequency decreases because both inputs must arrive at the destination gate in the same clock cycle period. Because the timing constraints are fixed for each SFQ gate, minimizing these two kinds of timing gaps is essential for achieving a high clock frequency in an SFQ design.

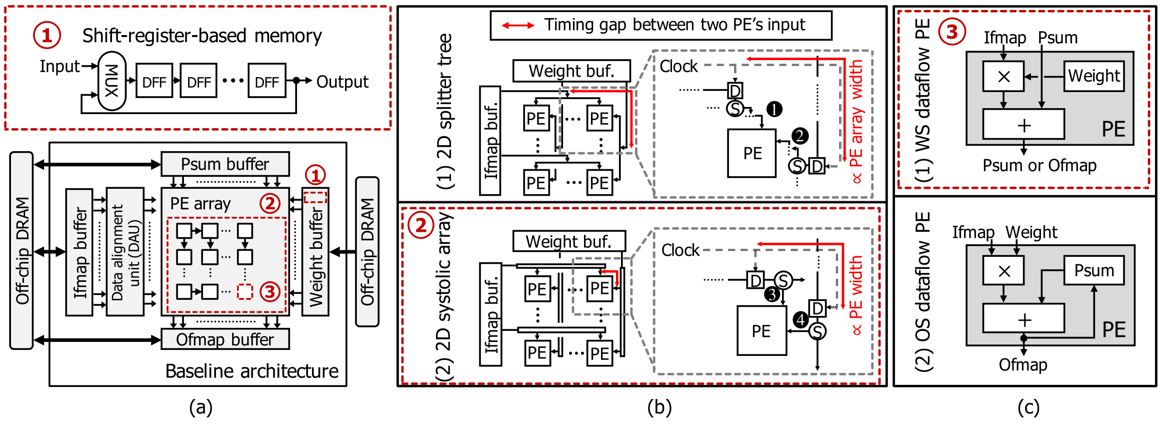

FIGURE 2. Our baseline SFQ-based NPU with each microarchitecture unit's design alternatives. (a) Overview of our baseline architecture. (b) Network designs. (c) PE designs.

SFQ-FAVORABLE ARCHITECTURAL CHARACTERISTICS

The aim of this article is contributing to the architecture community by introducing SFQ technology from the architects' perspective. To achieve this goal, we describe below SFQ-favorable architectural characteristics by fully considering the SFQ technology's unique features, which originate from its pulse-driven nature.

First, SFQ technology favors simple control flows due to its deeply pipelined nature. Architects can naturally apply gate-level pipelining to achieve a high frequency,[4,5] because all SFQ logic gates consist of superconductor rings, i.e., all SFQ gates have a latch functionality and, thus, can be pipelined without additional DFFs. However, the deep pipeline structure makes it difficult to avoid data (or control) hazards and, thus, may suffer from huge pipeline stalls. Therefore, applications with streaming execution are more favorable than those with a complex control flow.

Second, the SFQ technology favors sequential memory access patterns due to its on-chip memory implementation. There are two options for an SFQ-based on-chip memory design: random access memory (RAM) and shift-register-based memory. However, SFQ-based RAM suffers from poor driving capability and scalability, mainly because of the difficulty of driving the word lines and bit lines with the small pulses. On the other hand, a shift-register-based memory does not have such problems, so it is a much more practical option for on-chip memory. This means that applications with sequential memory accesses are much more suitable for SFQ technology.

Third, SFQ technology favors fewer off-chip memory accesses due to a lack of scalable off-chip memory technology. It has been a long-standing challenge to implement a large-scale and high-speed off-chip memory able to operate in a 4K environment. Although there have been a few studies on JJ-based memories and they are currently being developed, these technologies have not been put to practical use yet. For this reason, it is currently more practical to use CMOS memory technology, which is slower than the JJ-based memory but is large and reliable. Thus, computation-oriented applications with minimal off-chip memory accesses are suited to the current SFQ technology.

BASELINE SFQ-BASED NPU DESIGN

After considering the presented characteristics shown in the "SFQ-Favorable Architectural Characteristics" section, we chose an NPU as an example of SFQ-favorable architecture and designed the baseline SFQ-based NPU, as shown in Figure 2(a). Specifically, we implemented the on-chip buffer as a shift-register-based memory (①), the network unit (NW unit) as a 2D systolic network (②), and the processing element (PE) with weight-stationary dataflow (③) in an SFQ-friendly manner.

Network Unit for Systolic Array

To design an SFQ-friendly on-chip network, we compared two representative NW unit designs: fan-out network (e.g., splitter tree) and store-and-forward chain (e.g., systolic array), as shown in Figure 2(b). We selected the systolic array because it is superior to the splitter tree in both clock frequency and area. The splitter tree significantly suffers from the low frequency due to the increasing difference between the

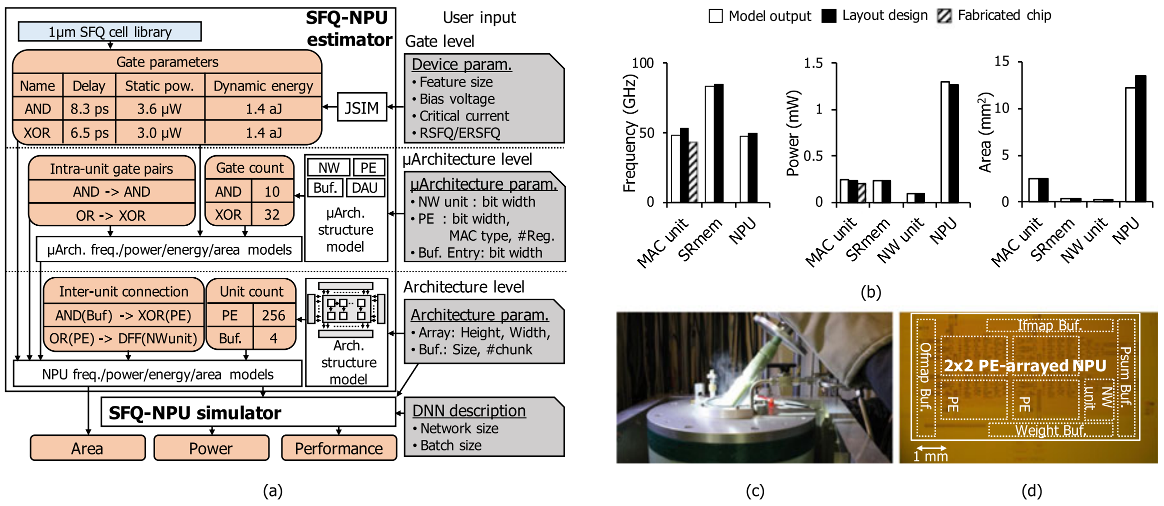

FIGURE 3. Simulation framework and validation. (a) SFQ-NPU overview. (b) Validation results in terms of frequency, power consumption, and area of microarchitectural and architectural units. (c) Validation setup. (d) Chip microphotograph of 2×2 PE arrayed NPU prototype design.

arrival timings of the two PE inputs. As shown in Figure 2(b)(1), the timing gap between input❶ and input❷ increases in proportion to the PE array width due to the large difference in clock arrival at two splitter trees.

On the other hand, as shown in Figure 2(b)(2), the systolic network has a smaller timing difference between input❸ and input❹, which does not scale with the PE array width. Thus, the systolic array can achieve a higher clock frequency than the splitter tree can. In addition, a systolic network has a simpler structure than the splitter tree and, thus, occupies less area. For these reasons, we decided that the systolic array is more SFQ-friendly and chose it to be part of our on-chip network design.

Processing Element With Weight-Stationary Dataflow

For an SFQ-friendly PE design, we compared the designs with two major dataflows in a systolic network, i.e., weight stationary (WS), where the PE stores weight pixels in its local register, and output stationary (OS), where the PE stores ofmap pixels in its local register, as shown in Figure 2(c). We chose the PE with WS dataflow to maximize the frequency, because it does not include any feedback loop. Unlike in CMOS technology, in which the clocking scheme synchronizes all the gates, the SFQ logic employs point-to-point (or gate-to-gate) synchronization. SFQ circuits can achieve a higher clock frequency without a feedback loop, because the data propagation delay can be hidden by making the clock pulse flow in the same direction as the data, resulting in a small timing gap between the arrivals of the clock and data. However, a circuit with a feedback loop cannot apply such a frequency speed-up because the clock and data pulses cannot flow in the same direction. Thus, we concluded that a PE design without a feedback loop, PE with WS, is a more SFQ-friendly choice.

SIMULATION FRAMEWORK

To enable exploration and optimization of the SFQ-based architectures, we have developed the architectural simulation framework for SFQ technology. Figure 3(a) shows an overview of our tool targeting SFQ-based NPU architecture, which consists of two engines: an SFQ-NPU estimator and SFQ-NPU simulator. In what follows, we describe these engines, their implementation, and validation.

SFQ-NPU Estimator

The SFQ-NPU estimator predicts the clock frequency, static power, access energy, and area of the target NPU configuration. To carefully consider the unique features of SFQ logic ranging from the device to architecture, the estimator takes a three-layer abstraction strategy: gate-level, microarchitecture-level, and architecture-level estimations.

First, in regard to the device-level parameters, the gate-level estimation layer derives the timing parameters, power information, and area for all of the SFQ logic gates and wire cells. The gate models are compatible with two SFQ technologies; rapid single-flux-quantum (RSFQ)[2] and energy-efficient RSFQ (ERSFQ).[6] For the RSFQ gates, all gate parameters are extracted by running JSIM[7] simulations with the RSFQ cell library.[8] On the other hand, the ERSFQ gate parameters are estimated from the RSFQ gate parameters and ERSFQ gate features[9] because of the lack of detailed fabrication information.

Next, the microarchitecture-level layer estimates the frequency, static power, access energy, and area of each microarchitectural unit by utilizing the gate-level layer's output and the input configuration parameters. For an accurate frequency estimation, this layer generates the intra-unit gate pair information consisting of all source and destination gate pairs in each unit because the two adjacent gates determine the SFQ circuit frequency. With the intra-unit gate pair information, this layer calculates the frequencies of all gate pairs in the target unit and takes the minimum value as the unit frequency. This layer also calculates the static power, access energy, and area of each unit on the basis of the gate count information and the gate-level layer's output.

Lastly, the architecture-level layer reports the estimation results regarding the area, static power, access energy, and clock frequency of the target NPU configuration. For an accurate prediction, this layer not only integrates the microarchitecture-level estimations based on the unit counts but also considers the inter-unit connection. For instance, it calculates all the inter-unit communication latencies on the basis of the interfacing gates' timing parameters and accounts for them when deriving the frequency of the target NPU. The layer also calculates the area of the wire cells required to connect each unit and includes it in the final area result.

We thoroughly validated our SFQ-NPU estimator's accuracy in terms of frequency, power, and area by comparing it with a fabricated 4-bit MAC unit measured in a 4K environment [Figure 3(c)] and in post-layout characterizations of an 8-bit 8-entry shift-register-based memory (SRmem), 8-bit NW unit, and 2-bit, and 2×2 PE-arrayed NPU. We have also fabricated a prototype chip of the NPU [Figure 3(d)] and plan to measure it in detail. As Figure 3(b) shows, the SFQ-NPU estimator accurately predicts the frequency, power, and area of each microarchitecture unit and target NPU. Even though our validation was conducted with a small NPU prototype, the estimation accuracy is rather convincing, thanks to the systolic network's scalable structure.

SFQ-NPU Simulator

For the given SFQ-based NPU design running DNN applications, the SFQ-NPU simulator reports the effective performance and power consumption based on the obtained frequency and power information from the SFQ-NPU estimator. The SFQ-NPU simulator first analyzes all of the required weight mappings by taking a DNN description file as an input. Next, it runs a cycle-based simulation for each weight mapping to derive the consumed cycles and hardware activation ratio. Finally, it aggregates the mapping results and reports the performance and the power results.

OPTIMIZING SFQ-BASED NPU DESIGN

By using our validated simulation framework, we identified and resolved architectural performance bottlenecks in our baseline SFQ-based NPU design. Then, we devised our SFQ-optimal NPU architecture, SuperNPU, which resolves the bottlenecks with architecture-level solutions.

To make observations and SFQ-specific optimizations, we conducted performance analyses by running six CNN workloads (i.e., Alexnet, FasterRCNN, GoogLeNet, MobileNet, ResNet50, and VGG16). As input information on the fabrication process, we used the currently available AIST 1.0-μm process in order to show the SFQ technology's potential conservatively. In addition, we assumed a memory bandwidth of 300 GB/s, which is the typical value of HBM used in the recent TPUv2.[10] Note that the estimated area of the baseline SFQ-based NPU design might be comparable to the TPU core (<330 mm^2) if JJs are equivalently scaled to 28 nm because CMOS technology is used in the TPU design.

Architectural Bottlenecks and Design Implications

We identified performance bottlenecks and design implications by conducting analyses with the baseline SFQ-based NPU design in the "Baseline SFQ-Based NPU Design" section (called the Baseline from here on). To show the design implications, we started from the Baseline by following the TPU core's architectural specifications (e.g., number of PEs, on-chip memory capacity) by focusing on their similar hardware structures.[11]

Our performance analyses identified two architecture-level performance bottlenecks. First, we found that the data movement overhead among different on-chip buffers (or within a single buffer) can significantly degrade performance. As the Baseline uses shift-register-based on-chip buffers, it should consume a huge amount of cycles corresponding to the

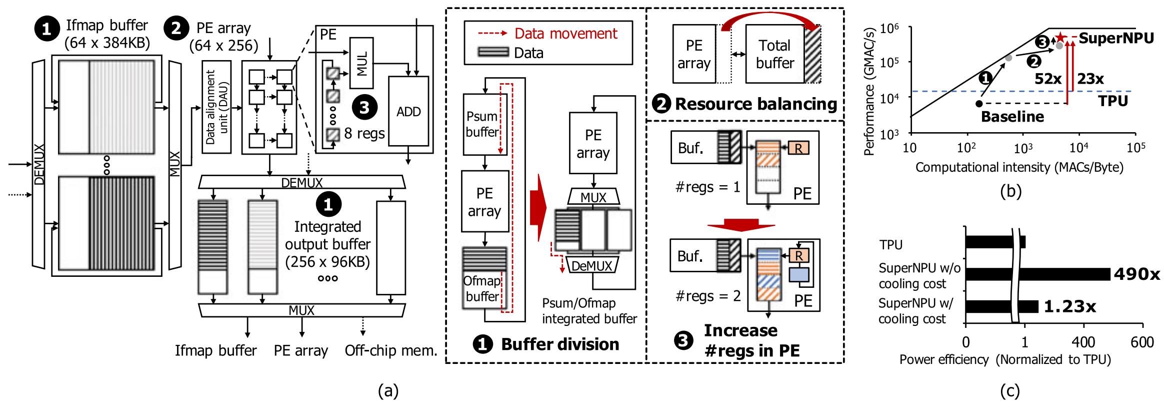

FIGURE 4. Architectural optimization summary. (a) Overview of SuperNPU with three architectural optimizations. (b) Performance evaluation result. (d) Power efficiency evaluation result normalized to the efficiency of the TPU.

buffers' length for moving the data from each buffer's tail to head. Second, we found that the PEs are significantly underutilized because the SFQ computing units are too fast compared with the slow off-chip memory access. If these bottlenecks are not resolved, even our SFQ-friendly baseline design with 52.6-GHz clock frequency cannot outperform a conventional CMOS design [Figure 4(b)]. Therefore, we should minimize the wasteful data movements while maximizing the PE utilization at the architectural level.

SuperNPU: SFQ-Optimal NPU Architecture

In light of the above implications, we devised our SFQ-optimal NPU architecture, SuperNPU. Figure 4(a) shows an overview of SuperNPU, driven by three SFQ-specific architectural optimizations: buffer optimization (❶), resource balancing (❷), and increasing the number of registers in PE (❸). We briefly introduce our optimization techniques and their performance impact here; interested readers should refer to our original paper for more details.[12]

First, we almost completely eliminated the data movement overhead among the on-chip buffers by optimizing the buffer architecture. Specifically, we divided each on-chip buffer into small buffer chunks (256 and 64 divisions for ofmap and ifmap buffers) and connected them with a multiplexer and demultiplexer. The optimized buffer architecture eliminates unnecessary data movements and improves buffer utilization. As a result, the Baseline's performance was significantly improved (by 19 times), as shown in Figure 4 (b)❶.

Next, we efficiently narrowed the gap between the computation and memory speeds by increasing the on-chip buffer capacity while reducing the number of PEs [Figure 4(a)❷]. Our key idea for this design choice is to reduce off-chip memory accesses by sacrificing excessive computation speed without a performance loss. By using resource balancing, we can increase each workload's computational intensity (i.e., the number of MAC operations executed with one weight data mapped on the PE) by increasing the batch size without additional off-chip memory accesses. As a result, we improved performance a further 2.1 times [Figure 4(b)❷].

OUR SUPERNPU OUTPERFORMS THE BASELINE BY 52 TIMES AT A 52.6-GHZ CLOCK FREQUENCY WHILE CLEARLY SHOWING THE RIGHT DIRECTIONS FOR ARCHITECTURE-LEVEL OPTIMIZATIONS.

Finally, to increase the PE utilization, we increased the number of weight registers in each PE. With the larger number of local weight registers, SuperNPU achieves higher PE utilization by filling several PE pipeline stages with a single input data. For example, if each PE holds two different weights from different weight filters, as in Figure 4(a)❸, PE can compute two different MAC operations with one ifmap pixel. As a result, we get an additional performance improvement of 1.3 times [Figure 4(b)❸].

In summary, our SuperNPU outperforms the Baseline by 52 times at a 52.6-GHz clock frequency while clearly showing the right directions for architecture-level optimizations. Besides, in our performance evaluation, SuperNPU outperformed TPU by 23 times, even with its immature device technology (i.e., 1-μm niobium process). Meanwhile, our power–efficiency evaluation [Figure 4(c)] showed that SuperNPU with ERSFQ technology achieves 490 times higher power efficiency without considering cooling power requirements. Even with the enormous cost of cooling to 4K (i.e., 400 times the power consumption of the device), SuperNPU attains 1.23 times higher power efficiency compared with TPU.

IMPACTS AND PROSPECTIVE

The major impacts of our work are as follows.

- Our validated modeling framework for SFQ logic enables architects to conduct fast and accurate SFQ architecture design space explorations. It can be used in other SFQ-promising domains. The model will encourage researchers to study SFQ architectures.
- The SFQ-based hardware design was optimized at the microarchitecture and architecture levels. The optimization process for NPU provided several crucial insights, e.g., on the timing adjustment of SFQ pulses at the circuit level and on the increasing buffer and pipeline utilization. These insights will guide the designs of follow-up SFQ-based architectures.
- Our work shows the true potential of SFQ logic by evaluating an SFQ-based architecture in comparison with its state-of-the-art CMOS counterpart. With our model-driven analysis and optimization, SuperNPU outperforms TPU by 23 times on average while achieving 1.23 times higher power efficiency, even including the enormous cooling cost at 4K. These significant results will motivate industry and academia to work on SFQ technology to prepare for the post-Moore era.

Meanwhile, there are critical future challenges regarding *scalability* that have to be addressed before SFQ computing platforms come into practical use.

- **Scale up**: Although our work shows the potential for high performance even with immature device technology (i.e., 1.0 μm niobium process), advances in device integration technology are essential for constructing sophisticated computer systems. In addition to device shrinkage as with conventional CMOS technology, a 3D-stacked SFQ design would be promising due to the ultra-low-power feature of superconductor logic devices. Besides, manually adjusting the timing of SFQ pulses on picosecond order is a challenging problem. Thus, placement and routing automation is indispensable to design large-scale SFQ circuits.
- **Scale out**: The superconductor transmission line in the circuit enables low-latency lossless signal propagation since it does not need charge/discharge processes. It can be used for both on-chip and off-chip communications, which means that SFQ circuits will be suitable for large-scale multichip architectures. Therefore, SFQ computing platforms have a potential for continuous growth independently of device shrinkage.

ACKNOWLEDGMENTS

This work was supported in part by the JST-Mirai Program under Grant JPMJMI18E1; in part by JSPS KAKENHI under Grant JP19H01105, Grant JP18H05211, and Grant JP18J21274; and in part by the National Research Foundation of Korea (NRF) under Grant NRF-2019R1A5A1027055 and Grant NRF-2021R1A2C3014131. The circuit was designed with the support by VDEC of the University of Tokyo in collaboration with Cadence Design Systems, Inc., and was fabricated in the CRAVITY of AIST.

REFERENCES

1. H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, “Dark silicon and the end of multicore scaling,” in *Proc. 38th Annu. Int. Symp. Comput. Archit.*, 2011, pp. 365–376, doi: 10.1109/77.80745.
2. K. K. Likharev and V. K. Semenov, “RSFQ logic/memory family: A new Josephson-junction technology for sub-terahertz-clock-frequency digital systems,” *IEEE Trans. Appl. Supercond.*, vol. 1, no. 1, pp. 3–28, Mar. 1991, doi: 10.1145/2000064.2000108.
3. G. Tzimpragos *et al.*, “A computational temporal logic for superconducting accelerators,” in *Proc. 24th Int. Conf. Archit. Support Program. Lang. Oper. Syst.*, 2020, pp. 435–448, doi: 10.1145/3373376.3378517.
4. I. Nagaoka, M. Tanaka, K. Inoue, and A. Fujimaki, “A 48GHz 5.6mW gate-level-pipelined multiplier using single-flux quantum logic,” in *Proc. IEEE Int. Solid- State Circuits Conf.*, 2019, pp. 460–462, doi: 10.1109/ISSCC.2019.8662351.
5. K. Ishida *et al.* “32 GHz 6.5 mW gate-level-pipelined 4-bit processor using superconductor single-flux-quantum logic,” in *Proc. IEEE Symp. VLSI Circuits*, 2020, pp. 1–2, doi: 10.1109/VLSICircuits18222.2020.9162826.
6. D. E. Kirichenko, S. Sarwana, and A. F. Kirichenko, “Zero static power dissipation biasing of RSFQ circuits,” *IEEE Trans. Appl. Supercond.*, vol. 21, no. 3, pp. 776–779, Jun. 2011, doi: 10.1109/TASC.2010.2098432.

7. E. Fang and T. V. Duzer, "A Josephson integrated circuit simulator (JSIM) for superconductive electronics application," in *Proc. Extended Abstr. Int. Supercond. Electron. Conf.*, 1989, pp. 407–410. [Online]. Available: https://ci.nii.ac.jp/naid/10006481720/
8. Y. Yamanashi *et al.*, "100 {GHz} demonstrations based on the single-flux-ouantum cell library for the 10 kA/cm^2 Nb multi-layer process," *IEICE Trans. Electron.*, vol. 93, no. 4, pp. 440–444, 2010.
9. O. A. Mukhanov, "Energy-efficient single flux quantum technology," *IEEE Trans. Appl. Supercond.*, vol. 21, no. 3, pp. 760–769, Jun. 2011, doi: 10.1109/TASC.2010.2096792.
10. "Hot Chips 2017: A. Closer Look At Google's TPU v2," [Online]. Available: https://www.tomshardware.com/news/tpu-v2-google-machine-learning,35370.html
11. N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proc. 44th Annu. Int. Symp. Comput. Archit.*, 2017, pp. 1–12, doi: 10.1145/3079856.3080246.
12. K. Ishida *et al.*, "SuperNPU: An extremely fast neural processing unit using superconducting logic devices," in *Proc. 53rd Annu. IEEE/ACM Int. Symp. Microarchit.*, 2020, pp. 58–72, doi: 10.1109/MICRO50266.2020.00018.

KOKI ISHIDA is currently working at Kyushu University. His research interests include computer system architecture using superconductor single flux quantum logic. Ishida received a Ph.D. degree from Kyushu University in 2021. He is one of the co-first authors of this article. Contact him at koki.ishida@cpc.ait.kyushu-u.ac.jp.

ILKWON BYUN is currently working toward a Ph.D. degree with the Department of Electrical and Computer Engineering, Seoul National University, Seoul, South Korea. His research focuses on architecting cryogenic CMOS and superconductor-based computer systems by using computer architecture modeling and simulation techniques. He is a Student Member of IEEE and ACM. He is one of the co-first authors of this article. Contact him at ik.byun@snu.ac.kr.

IKKI NAGAOKA is currently working toward a Ph.D. degree with the Department of Electronics, Nagoya University, Nagoya, Japan. His research interests include designing LSIs using the superconductor single-flux quantum logic. Contact him at nagaoka@super.nuee.nagoya-u.ac.jp.

KOSUKE FUKUMITSU is currently working toward an M.E. degree with the Department of Advanced Information and Technology, Kyushu University, Fukuoka, Japan. His research interests include designing the superconductor single-flux-quantum circuits. Contact him at kosuke.fukumitsu@cpc.ait.kyushu-u.ac.jp.

MASAMITSU TANAKA is currently an Assistant Professor with the Department of Electronics, Graduate School of Engineering, Nagoya University, Nagoya, Japan. His research interests include ultra-fast/energy-efficient computing using the SFQ-based technology and logic design methodologies. He is a senior member of the IEEE. Contact him at masami_t@ieee.org.

SATOSHI KAWAKAMI is currently an Assistant Professor with the Department of Advanced Information and Technology, Kyushu University, Fukuoka, Japan. His research interests include computer architecture with emerging devices. He is a Member of IEEE and ACM. Contact him at satoshi.kawakami@cpc.ait.kyushu-u.ac.jp.

TERUO TANIMOTO is currently an Assistant Professor at the Research Institute for Information Technology, Kyushu University, Fukuoka, Japan. His research interests include edge computing systems, secure computer architecture, and quantum computer system architecture. He is a Member of IEEE and ACM. Contact him at tteruo@kyudai.jp.

TAKATSUGU ONO is currently an Associate Professor with the Department of Advanced Information and Technology, Kyushu University, Fukuoka, Japan. His research interests include the areas of computer architecture with particular emphasis on secure computing, high-performance computing, and memory system (including nonvolatile memory). Ono received a Ph.D. degree from Kyushu University in 2009. He is a Member of IEEE. Contact him at takatsugu.ono@cpc.ait.kyushu-u.ac.jp.

JANGWOO KIM is currently a Professor with the Department of Electrical and Computer Engineering, Seoul National University, Seoul, South Korea. His primary research interests lie in computer architecture, server and datacenter, system modeling, and intelligent systems. Kim received a Ph.D. degree in electrical and computer engineering from Carnegie Mellon University in 2008. He is a Member of IEEE and ACM. Contact him at jangwoo@snu.ac.kr.

KOJI INOUE is a Professor with the Department of Advanced Information and Technology, Kyushu University, Fukuoka, Japan. He is currently the Director of System LSI Research Center. His research interests include computer architecture and low-power system designs. Inoue received a Ph.D. degree from Kyushu University. He is a Member of IEEE and ACM. He is the corresponding author of this article. Contact him at inoue@ait.kyushu-u.ac.jp.