Innovative Computing for Solving Social Issues

● Atsuki Inoue ● Takashi Miyoshi ● Teruo Ishihara ● Yasufumi Honda

Since the development of practical stored-program computers in the late 1940s, performance has risen amazingly by about 10^{12} times over a period of 70 years. However, it is generally recognized that semiconductor transistor scaling is reaching its limits and that Moore's law is coming to an end. Regardless of these technical issues, the explosive increase in the amount of data generated in today's IoT era is expected to continue, and it is highly anticipated that this data will be used to create new value and novel services. Meeting these expectations will therefore require improvements in performance independent of Moore's law. To address these issues, Fujitsu Laboratories proposes domain-specific computing as a new computing paradigm. The aim of domain-specific computing is to break through Moore's law by adopting architecture specific to the type of processing needed in fields such as knowledge processing whose objective is not to obtain rigorous numerical results. For example, in application to deep learning engines, high-speed image search engines, and machines dedicated to combinatorial optimization problems, domain-specific computing has demonstrated that it showed 50–12,000 times higher performance than that of conventional approaches. In this paper, we describe the direction of domain-specific computing as a new computing paradigm and present specific application examples.

1. Introduction

About 70 years have passed since the development in 1949 of the electronic delay storage automatic calculator (EDSAC), a practical stored-program computer, and today, everyone can get a computer in the form of a smartphone. EDSAC was a huge computer using 3,000 vacuum tubes and mercury delay lines as memory and consuming 12 kW of electric power. As basic electronic devices, these vacuum tubes eventually came to be replaced by solid electronic devices, or transistors, invented by W. B. Shockley et al. at around the same time. This invention gave birth to many second-generation commercial computers in the 1950s and 1960s. With the invention of monolithic integrated circuit technology by J. Kilby et al., the cost of computers began to drop rapidly, and the development of high-performance computers began to accelerate through the effective use of even more transistors.

Today, we have Moore's law as an empirical approach to predicting the future of integrated circuits (IC). This law was originally advanced by G. Moore, a co-founder of Intel Corporation, who stated, "The complexity for minimum component (transistor) costs has increased at a rate of roughly a factor of two per year. There is no reason to believe it will not remain nearly constant for at least ten years."1) This was later given theoretical support by the scaling law of R. H. Dennard of IBM. Since that time, the limit to Moore's law was frequently discussed, but it nevertheless drove the development of semiconductor microfabrication technology over a period of about 30 years up until the early 2000s. This is because the miniaturization of semiconductor process dimensions simultaneously improved the performance and power efficiency of transistors, improved the integration density, and brought down costs thereby "killing three birds with one stone."

As a result, plotting computer performance from the EDSAC era up to 2010 would show that it doubled approximately every year and a half thereby maintaining exponential improvement,2) which means an improvement in performance over a 70-year period on the order of 10^{12}, a truly amazing figure. In addition, the power efficiency of computers improved in about the same way doubling about every year and a half, which means an improvement in computer performance for a device of about the same size and price. In other words, this spectacular improvement in computer performance up to the present can be primarily attributed to the progress made in semiconductor microfabrication technology.

Given a generation in which an ideal scaling law was applicable, it was possible to improve integration density while keeping power density fixed. That is to say, the product of power efficiency and performance squared is constant (**Figure 1**). The value of this product (K in the figure) is determined by the semiconductor microfabrication technology that can be achieved in any one generation, so it can be expressed by a straight line for that generation as shown in the figure. What this means is that high performance and high power efficiency cannot be simultaneously achieved if their product falls outside that line. In this sense, such a line can be called Moore's limit line. On the other hand, the value of this product becomes larger as semiconductor microfabrication technology progresses, so it actually became possible to improve both power efficiency and performance over time with each generation.

However, on entering the 2000s, it became difficult to ideally lower the power supply voltage, and as a consequence, increasing the integration density of transistors resulted in a dramatic increase in power consumption. It was therefore difficult to improve performance due to limitations in power consumption. In addition, once process dimensions approach a level on the order of 100 times the size of an atom, it becomes difficult to shorten gate length, a determining factor in transistor performance. From the 45-nm-node era on, it has become practically impossible to shorten transistor gate length. It is therefore thought that Moore's limit lines have become a wall, and that relying solely on progress in semiconductor microfabrication technology will make it difficult to achieve any further improvements in performance.

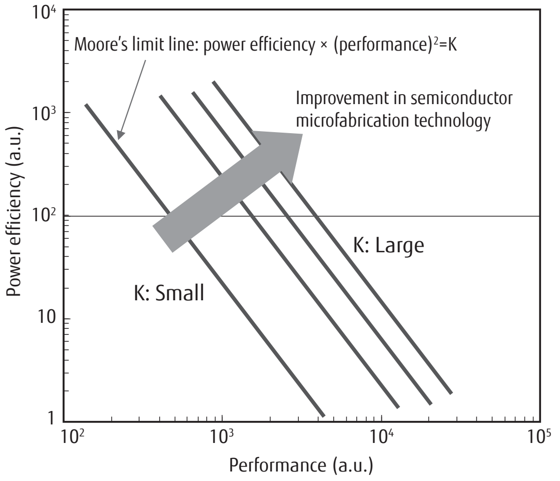


Figure 1
Moore' s limit lines and improvement by microfabrication technology.

Against the above background, many discussions[3] have taken place on the direction of research and development with respect to the end of Moore's law and the possibility of computing beyond it and Moore's limit lines. These include the development of a RISC-V instruction set aimed at domain-specific computing, development by D-Wave Systems Inc. of a quantum annealer using superconducting circuit technology, and development of a tensor processing unit (TPU), an LSI specialized for AI applications from Google.

In this paper, we begin by explaining the concept of domain-specific computing. We then introduce examples of applying this new approach to specific domains focusing on the three fields of AI, media, and combinatorial optimization.

2. Domain-specific computing

The end of Moore's law is imminent, and from a technology perspective, a number of challenges exist involving semiconductor processes, network bandwidth, power consumption, and computing performance. At the same time, the amount of data that needs to be processed continues to increase in an explosive manner. The amount of data generated by IoT is expected to exceed 40 zettabytes by 2020 and to reach 1 yottabyte by 2030.[4] It is clear that this explosive increase in data will come to outpace existing ICT-based data processing power, and that the creation of a new form of information processing to find out valuable information from within a massive amount of data will pose a challenge. For example, the density of information in data generated by IoT devices is thin. However, if that large amount of data were to be consolidated on the cloud and if the essence or meaning of that data were to be extracted using AI, it would be possible to transform that data into knowledge and intelligence, which could be applied to achieving advanced ICT solutions.

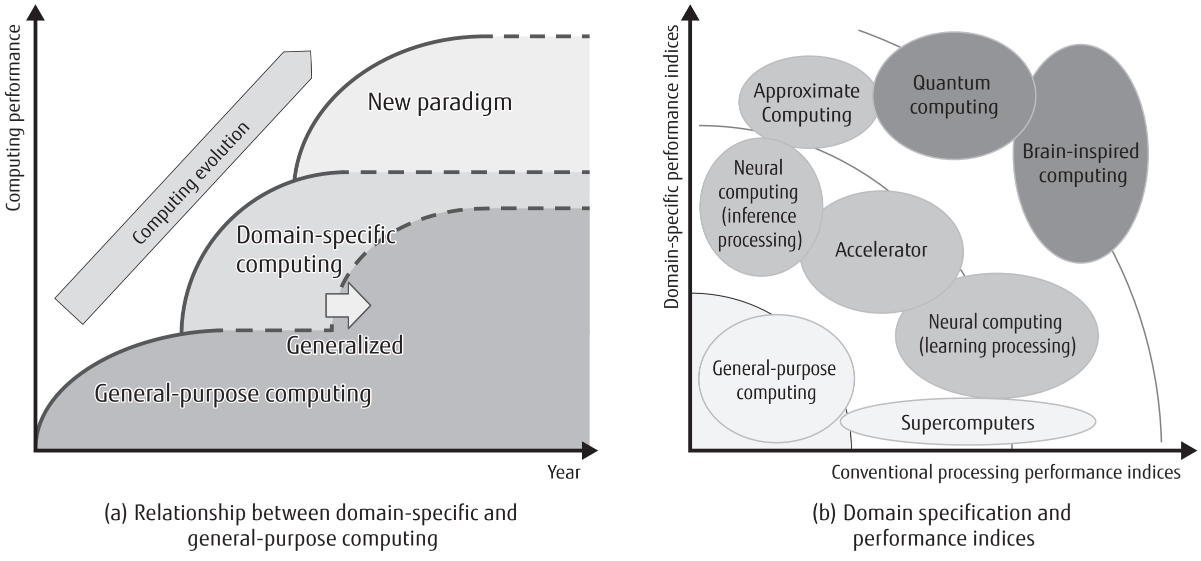
Computing itself must change to support such a new form of information processing. While conventional architecture has excelled in numerical processing, it must evolve into architecture applicable to the efficient creation of knowledge and intelligence. New applications and services making use of this knowledge and intelligence can be created by having techniques for processing large amounts of data evolve in a manner complementary to the evolution of architecture.

The direction of computing architecture evolution is shown in **Figure 2 (a)**. If a new form of architecture arises in a particular field, computing power can jump dramatically suggesting that innovation in architecture can give rise to a paradigm shift. In addition, general-purpose computing itself can gradually incorporate such a paradigm shift and become stronger on the basis of that new architecture. The creation of new applications and services requires ongoing innovation in architecture and the creation of paradigm shifts. Fujitsu Laboratories proposes domain-specific computing as one approach to this end.

Conventionally speaking, computing performance has been evaluated mainly by indices such as integer operations performance, floating-point operations performance, and memory bandwidth for general-purpose applications. These indices are closely related to semiconductor performance, which means that it will be difficult to achieve ongoing improvements in performance once Moore's law comes to an end. On the other hand, by determining computing architecture by narrowing down the field, that is, the domain targeted for processing, and by focusing on the type of processing frequently used in that domain, it should be possible to raise performance by several orders of magnitude beyond Moore's law. This approach is called domain-specific computing.

Architecture-specific characteristics in computing are shown in **Figure 2 (b)**. The horizontal axis represents conventional computing processing performance and the vertical axis shows new performance indices defined for specific domains according to the purpose of computing. Taking, for example, the media domain, a new index may refer to the degree to which required image quality is achieved for the target usage scenario. This could not be measured simply on the basis of conventional CPU performance indices such as operating frequency and numerical operations performance.

To surmount the limits in improving performance by semiconductor downscaling, Fujitsu Laboratories has been promoting improvements in processing power through the use of supercomputers in the horizontal direction and pursuing specialization in specific domains


Computing performance
Computing evolution
New paradigm
Domain-specific computing
Generalized
General-purpose computing
Year
Domain-specific performance indices
Approximate Computing
Quantum computing
Brain-inspired computing
Neural computing (inference processing)
Accelerator
Neural computing (learning processing)
General-purpose computing
Supercomputers
Conventional processing performance indices

(a) Relationship between domain-specific and general-purpose computing

(b) Domain specification and performance indices

Figure 2
Domain-specific computing.

in the vertical direction of Figure 2 (b). Computing is changing from its conventional emphasis on numerical processing to the processing of specific information, knowledge, and intelligence. With this in mind, we are taking on the accelerator and neural computing domains as the first stage of domain-specific computing. Furthermore, as next-generation architecture, we are focusing our attention on quantum computers and brain-inspired computing.

Revising architecture based on the idea of domain-specific computing will take on a direction different from that of the past. Conventional architecture assumes various types of workloads and makes use of a general-purpose CPU designed to demonstrate a level of performance common to those workloads. The goal here is to combine sequential processing and parallel processing to search for highly accurate (or uniformly accurate) solutions.

In contrast, the focus in domain-specific-computing architecture is on core processing essential to that domain. By identifying those characteristics and setting up a large number of lean and simple dedicated cores, highly parallel processing of a huge number of operations with high power efficiency becomes possible. In addition, by pursuing accuracy suitable to the target domain, processing can be made efficient and optimal. On the basis of these policies, the aim is to achieve performance and power efficiency greatly higher than that of conventional computing.

The computing devices used in place of a general-purpose CPU play an important role in achieving domain-specific computing. Examples include general-purpose graphics processing units (GPGPUs) that run highly parallel programs and dedicated hardware such as field-programmable gate arrays (FPGAs) and application specific integrated circuits (ASICs). In addition, the algorithms that act as core processes in the target domain will undergo a transformation from algorithms designed for conventional general-purpose CPUs to algorithms that take hardware structure into account. The point here is that treating algorithms and hardware as closely linked elements and providing optimal processing techniques can achieve greatly higher speeds.

The following section describes the following technologies developed by Fujitsu Laboratories as domain-specific computing:

1) deep learning (DL) dedicated engine,
2) media server, and
3) Digital Annealer.

3. Application examples of domain-specific computing

Excluding data input and output, the basic operations of computing consist of control, computation, and storage (memory), as shown in **Figure 3**. In domain-specific computing, the architectures of each basic operation are tailored in accordance with their characteristics, thereby obtaining high levels of performance not possible by past approaches.

The following presents application examples of domain-specific computing.

3.1 Deep learning dedicated engine: Deep Learning Unit (DLU)

In contrast to successive changes in process flow based on conditional judgments, deep learning learns a little at a time by inputting training data into a fixed process flow and repeating that process. Furthermore, in applications using AI typified by deep learning, it is assumed that a correct answer can be obtained above a certain probability given the characteristics of inference in deep learning. That is to say, obtaining a correct answer without fail is not basic to such AI-based applications.

Although high inference accuracy is associated with high value, training requires a massive amount of processing. For this reason, achieving an inference accuracy that satisfies the requirements of the application with limited resources leads to a higher value. For example, training with a small amount of processing enables various types of training to be performed at the

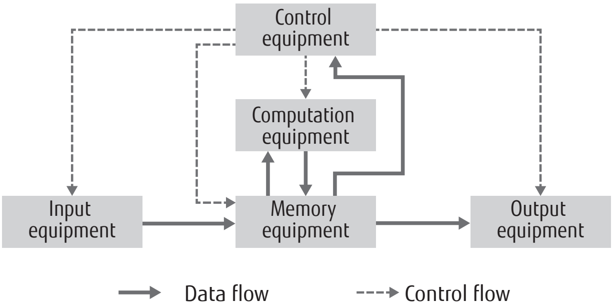


Figure 3
Basic operations of computing.

same time. Also, training with low power consumption broadens the scope of applicable domains.

Deep-learning computing originally performed arithmetic operations in 32-bit floating-point format using a CPU or GPU. However, operations, memory, memory bandwidth, and power consumption have all been increasing as the scale of deep neural networks expands to improve inference accuracy. The need has consequently arisen for architecture that can improve power performance by improving the efficiencies of the above elements.

The DLU is a processor similar to a CPU, but specialized for deep learning processing.[5], [6] It adopts original architecture to achieve high processing performance with low power consumption. Specifically, to maintain a constant level of performance against a massive amount of training data, the control and memory equipment of the DLU features a mechanism for sharing and saving data in register files within a deep learning processing unit (DPU) under software control independently from calculation equipment (**Figure 4**). It also features architecture that can perform calculations in parallel on the basis of large register files (**Figure 5**). In deep learning processing by GPGPU, floating-point 32-bit (FP32) operations have been standard. However, a conversion to low-bit operations to improve processing performance has been gaining momentum as reflected by the announcements of NVIDIA's TensorCore[7] based on 16-bit floating-point operations and Intel's Flexpoint[8] based on 16-bit integer operations. These new devices reflect attempts at performing learning with fewer bits while maintaining learning performance equivalent to FP32 operations.

The DLU incorporates a mechanism called Deep Learning Integer (DL-INT) that achieves necessary accuracy in operations by using statistical information within those operations and enables learning by 8 or 16 bits. This mechanism enables a maximum of four parallel operations for the same amount of memory and memory bandwidth. Furthermore, by reducing power consumption by half as a result of integer operations, DL-INT aims to improve power performance by approximately eight times compared with FP32 operations. Additionally, in multichip learning using more than one

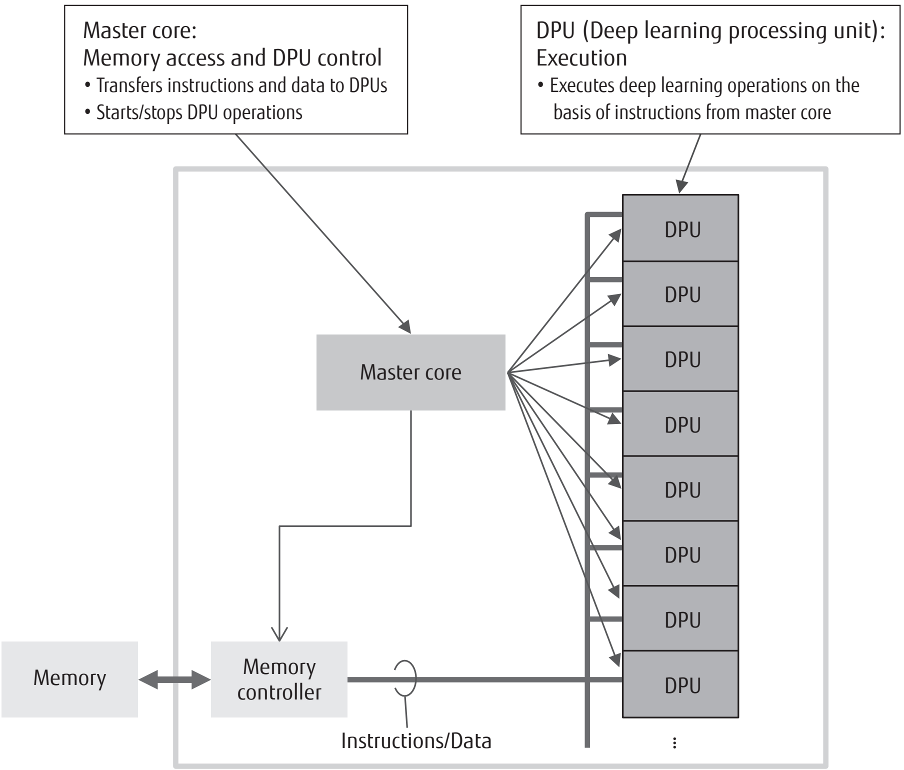

Master core:
Memory access and DPU control
• Transfers instructions and data to DPUs
• Starts/stops DPU operations
DPU (Deep learning processing unit):
Execution
• Executes deep learning operations on the basis of instructions from master core
Master core
DPU
DPU
DPU
DPU
DPU
DPU
DPU
DPU
⋮
Memory
Memory controller
Instructions/Data

Figure 4
Control system and multi-DPU configuration.

semiconductor chip, DL-INT can reduce the amount of data traffic between chips where bottlenecks can occur. This feature contributes to efficient multichip learning in ever expanding deep neural networks.

3.2 Media server

A media server aims to facilitate the reuse of large quantities of data and to make business tasks more efficient through high-speed retrieval of media data such as images and audio clips.

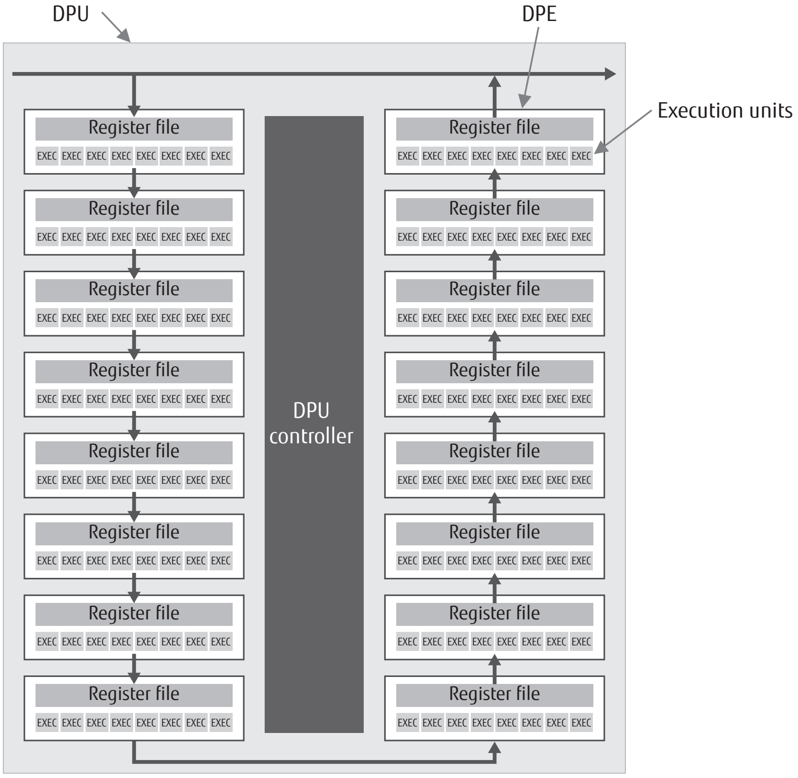
The key features of the architecture adopted here to speed up media retrieval are the parallel processing of algorithms involved in media retrieval processing, their implementation in a hardware engine, and the offloading of operations from CPU processing. Using a FPGA as the base device, this engine raises processing efficiency through balanced pipeline scheduling that transfers stored data to high-speed memory in a timely manner.[9]

Specifically, this architecture allocates hardware resources to 32 parallel feature calculations and 6 parallel matching processes, thereby achieving hardware-based parallel processing of algorithms. It has successfully increased the speed of media retrieval by 50 times.

3.3 Digital Annealer

There are combinatorial optimization problems (such as the traveling salesman problem) that can be solved by conventional computing means given a small number of combinations. However, as the number of combinations increases, the time required to obtain a solution can increase explosively.

Such problems lie outside the scope of conventional general-purpose computing, but implementing a fully connected Ising model in hardware and improving processing speed dramatically have made it possible to solve real-world problems such as the financial portfolio optimization problem (up to 500 stocks). The architecture used here to speed up the processing of combinatorial optimization problems features high-speed processing



DPU: Deep learning processing unit
DPE: Deep learning processing element

Figure 5
DPU architecture.

of a single trial (basic processing) by implementing core operations in dedicated hardware and the 1,024-parallel execution of that processing. Applying the above together with the dynamic offset scheme as a process algorithm achieves an overall improvement in processing speed of approximately 12,000 times.[10)]

4. Conclusion

This paper described the direction and presented application examples of domain-specific computing, a new computing architecture advanced by Fujitsu for breaking through the limits in improving computer performance by semiconductor microfabrication technology.

The demand for improved performance in computing is expected to continue in the years to come. However, we showed in this paper that performance can be raised by several orders of magnitude without relying on improvements in device performance if the fields targeted for processing are narrowed down to those that do not necessarily require optimization, such as knowledge processing, and if attention is focused on frequently used processes.

Going forward, our plan is to develop software and libraries that can extract performance without having to make major changes to applications themselves. We feel that achieving a mechanism that can provide such high-performance processing to customers at low cost is a key element of this endeavor.

References

1) G. Moore: Cramming More Components onto Integrated Circuits. Electronics, Vol. 38, No. 8, p. 114 (1965).

2) J. G. Koomey et al.: Implications of Historical Trends in the Electrical Efficiency of Computing. IEEE Annals of the History of Computing, p. 46 (2011).

3) T. H. Theis et al.: The End of Moore's Law: A New Beginning for Information Technology. Computing in Science & Engineering, Vol. 19, pp. 41–50 (2017).

4) M. Nihei et al.: An Essay on evaluation and quality and value of information. IEICE Technical Report, Vol. 116, No. 290, pp. 9–12 (2016) (in Japanese).

5) A. Ike et al.: Technologies for Practical Application of Deep Learning. Fujitsu Sci. Tech. J., Vol. 53, No. 5, pp. 14–19 (2017).
https://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol53-5/paper03.pdf

6) T. Maruyama: Fujitsu HPC and AI Processors. ISC 2017.
http://www.fujitsu.com/global/Images/fujitsu-hpc-and-ai-processors.pdf

7) P. Micikevicius et al.: Mixed Precision Training. arXiv preprint arXiv:1710.03740, 2017.

8) U. Köster et al.: Flexpoint: An Adaptive Numerical Format for Efficient Training of Deep Neural Networks. In: Advances in Neural Information Processing Systems, pp. 1740–1750 (2017).

9) Y. Watanabe et al.: Domain Specific Computing Using FPGA Accelerator. Fujitsu Sci. Tech. J., Vol. 53, No. 5, pp. 20–25 (2017).
http://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol53-5/paper04.pdf

10) S. Tsukamoto et al.: An Accelerator Architecture for Combinatorial Optimization Problems. Fujitsu Sci. Tech. J., Vol. 53, No. 5, pp. 8–13 (2017).
http://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol53-5/paper02.pdf

Atsuki Inoue
Fujitsu Laboratories Ltd.
Dr. Inoue is currently engaged in research and development of new computer architecture.

Takashi Miyoshi
Fujitsu Laboratories Ltd.
Mr. Miyoshi is currently engaged in research and development of domain-specific computing.

Teruo Ishihara
Fujitsu Laboratories Ltd.
Mr. Ishihara is currently engaged in research of knowledge computing.

Yasufumi Honda
Fujitsu Ltd.
Mr. Honda is currently engaged in Deep Learning Unit (DLU) development.