SPARC64 VIIIfx: CPU for the K computer

● Toshio Yoshida　　● Mikio Hondo　　● Ryuji Kan　　● Go Sugizaki

SPARC64 VIIIfx, which was developed as a processor for the K computer, uses Fujitsu Semiconductor Ltd.'s 45-nm CMOS process for semiconductors and is composed of eight cores, a 6 MB shared level 2 cache, and memory controllers. Peak performance of 128 GFLOPS at an operating frequency of 2 GHz is achieved with power consumption as low as 58 W. The performance per unit of power is more than six times that of the SPARC processor, our previous model. To achieve this performance per unit of power, we extended the SPARC-V9 architecture to develop high performance computing-arithmetic computational extensions (HPC-ACE), the optimum instruction set for scientific computations. In addition, we successfully reduced the leakage power by water cooling and dynamic power by clock gating to achieve a lower power consumption. Furthermore, high-reliability technology for mainframes and UNIX servers is used to ensure stable operation of a system connecting more than 80 000 processors. This paper outlines the technologies used to achieve the high performance, low power consumption and high reliability of SPARC64 VIIIfx.

1. Introduction

Fujitsu developed SPARC64 VIIIfx[1] (**Figure 1**) as a processor for the supercomputer ("K computer").[note] The K computer has more than 80 000 processors installed to give it a computational performance in excess of 10 PFLOPS. These processors need to have high performance, low power consumption, and high reliability. This paper outlines the technologies used to achieve these goals.

2. Goals in development of SPARC64 VIIIfx

The goals in the development of SPARC64 VIIIfx include:

---

note) "K computer" is the English name that RIKEN has been using for the supercomputer of this project since July 2010. "K" comes from the Japanese word "Kei," which means ten peta or 10 to the 16th power.

1) High performance

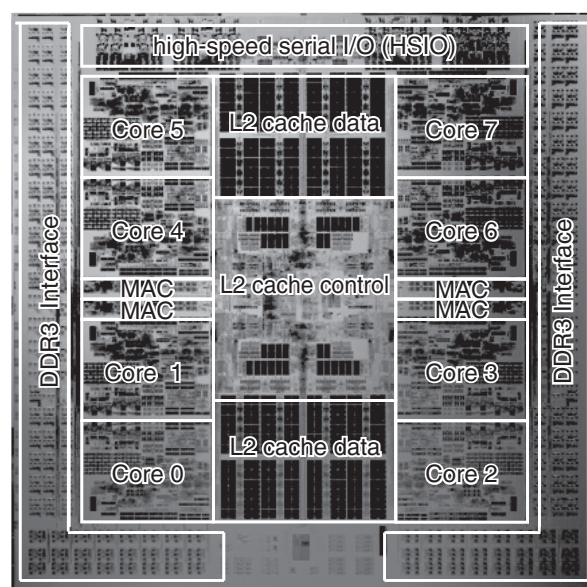SPARC64 VIIIfx is a multicore processor

Figure 1
SPARC64 VIIIfx chip.

integrating eight cores, a shared level 2 cache, memory access controllers (MACs), and a high-speed serial I/O (HSIO).

For each core to exhibit high-execution performance in a real application, we extended the SPARC-V9 architecture[2)–4)] and developed High Performance Computing-Arithmetic Computational Extensions (HPC-ACE),[5)] an instruction set capable of efficiently executing scientific computations.

To achieve a higher speed of parallel processing by having eight cores on the chip, the architecture must also have a function to share the level 2 cache across all cores and synchronize cores by means of hardware. Combining this with Fujitsu's automatic parallel compiler allows the user to handle multiple cores when programming as if it were one high-speed CPU without having to be aware of the multiple cores. At Fujitsu, this is called Virtual Single Processor by Integrated Multicore Parallel Architecture (VISIMPACT).

2) Low power consumption

Due to the limited amount of power available to the entire system, the power consumption of the processor needed to be reduced to 58 W or less. To that end, Fujitsu used low-leakage transistors and water cooling to lower the junction temperature to 30°C so as to reduce the leakage power. In addition, the processor must make full use of clock gating to reduce its dynamic power.

3) High reliability

The processor requires high-reliability technology used for mainframes and UNIX servers[6),7)] to ensure it operates stably.

3. Microarchitecture of SPARC64 VIIIfx

The pipeline of SPARC64 VIIIfx is shown in **Figure 2** and the specifications are shown in **Table 1**.

A core is composed of an instruction control unit, execution unit and level 1 cache. The instruction control unit is responsible for instruction fetch, instruction decode, out-of-order instruction control, and instruction commit control.

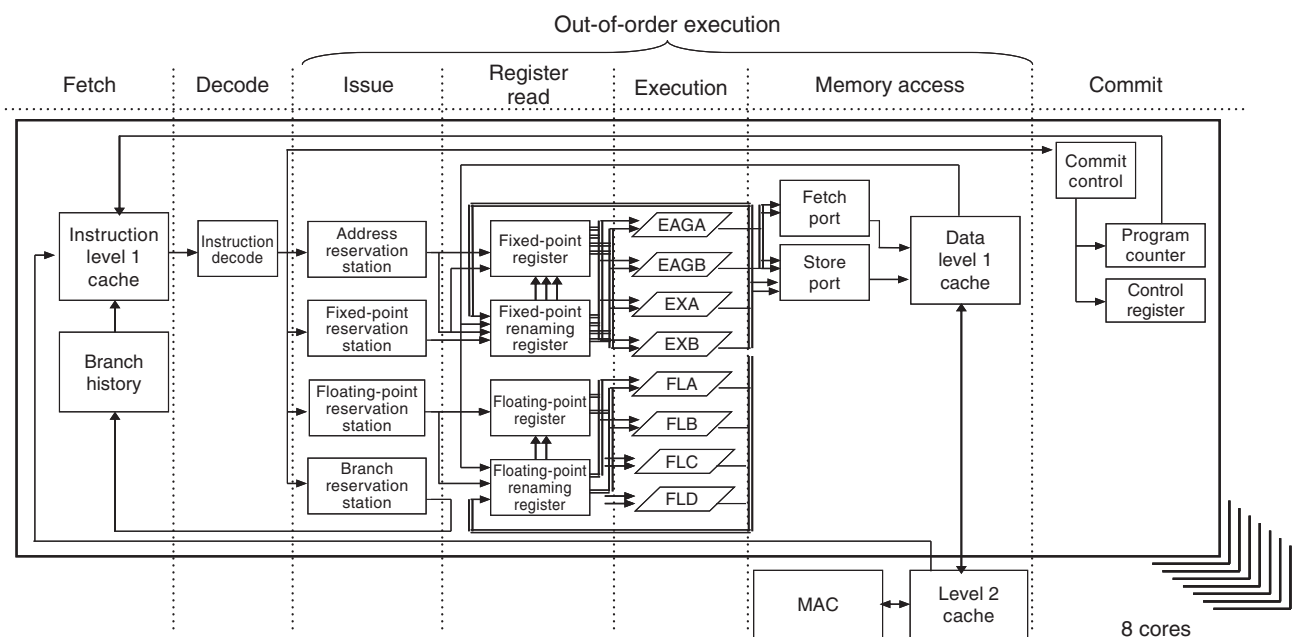The execution unit is equipped with

Figure 2
SPARC64 VIIIfx pipeline.

Table 1
SPARC64 VIIIfx specifications.

Item	Specification
No. of cores	8
Level 2 cache	6 MB
Operating frequency	2 GHz
Process technology	FSL 45-nm CMOS
Die size	22.7 mm × 22.6 mm
No. of transistors	Approx. 760 million
Peak performance	128 GFLOPS
Memory bandwidth	64 GB/s (theoretical peak value)
Power consumption	58 W (process condition: TYP)

FSL: Fujitsu Semiconductor Ltd.

two fixed-point functional units (EXA/B), two functional units for load/store address computation (EAGA/B) and four floating-point multiply-and-add (FMA) units (FLA/B/C/D). The FMA units have a single instruction multiple data (SIMD) architecture and execute two parallel operations with one instruction. One FMA unit is capable of conducting floating-point multiplication and addition for each cycle and each core can execute eight double-precision floating-point operations per cycle. Hence the chip is capable of making 64 double-precision floating-point operations per cycle. The operating frequency is 2 GHz and the peak performance is 128 GFLOPS. There are 192 fixed-point registers and 256 floating-point registers.

The level 1 cache processes load/store instructions. Each core has a 32 Kbyte two-way instruction cache and data cache. The data cache has a dual-port structure capable of two simultaneous load accesses and executes two 16-byte SIMD loads or one 16-byte SIMD store.

The level 2 cache is shared by the eight cores and cache coherence is ensured for each core. An inter-core hardware barrier is provided to allow high-speed synchronization between the cores, as will be described later.

SPARC64 VIIIfx incorporates memory controllers to reduce its latency and improve the throughput of its memory access. The memory bandwidth is 64 GB/s as a theoretical peak value.

In addition, a K computer exclusive InterConnect Controller chip and HSIO are used for connections to ensure a high inter-chip communication throughput.

4. Instruction extensions HPC-ACE

HPC-ACE is an extended instruction set intended for scientific computation for the SPARC-V9 architecture. It was developed based on the analysis of many HPC applications jointly with Fujitsu software development department.

1) Expansion of number of registers

The number of floating-point registers of SPARC-V9 is 32, which is not sufficient for HPC applications. Increasing the number of registers, however, is not possible with the 32-bit SPARC architecture because there is an insufficient instruction length. As a solution to this, a new prefix instruction called the set extended arithmetic register (SXAR) has been defined for HPC-ACE. An SXAR instruction extends register addressing for up to two following instructions. The register address length is extended by 3 bits, which allows for 256 floating-point registers, eight times that of SPARC-V9 (**Figure 3**).

The compiler uses this high-capacity set of registers for optimization including software pipelining and maximizes the instruction-level parallelism of an application. In terms of the Himeno Benchmark, a representative HPC benchmark, the performance has improved by 1.65 times.

2) SIMD operations and load/store instructions

SIMD is a technology to allow parallel execution of more than one data process with one instruction. HPC-ACE uses SIMD technology to execute two FMA operations with one instruction. SIMD operations for more quickly multiplying complex numbers are also supported. In addition, SIMD execution is possible with load and store instructions. SIMD processing of load instructions is executed without penalty in an 8-byte alignment for double precision and 4-byte
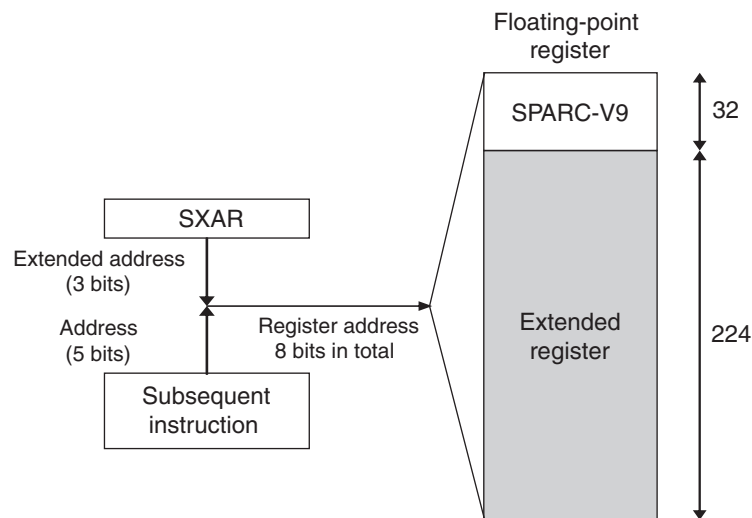
Figure 3
Register address extension by SXAR instruction.

alignment for single precision.

3) Sector cache mechanism

For HPC-ACE, a cache mechanism (sector cache) that can be software-controlled has been developed. The conventional caches cannot be controlled by software. Even if the user is aware of the high frequency with which data is reused, the hardware evicts the data from the cache when registering other data in the cache, which might hinder improvements in performance. To address this issue, the sector cache mechanism splits the cache into two sectors and allows software to be used to register frequently reused data in a sector separately from other data. Implementing control to have the user keep frequently reused data in the cache contributes to a better performance.

4) Acceleration instructions for trigonometric functions sine and cosine

Instructions to accelerate the trigonometric functions sine and cosine have been added. They have traditionally been processed by combining many instructions, but providing dedicated instructions has reduced the number of instructions, leading to an increase in speed of more than five times.

5) Conditional execution

To efficiently process loops containing *if* statements, it is necessary to eliminate conditional branch instructions. For that purpose, conditional execution instructions have been added for HPC-ACE. Specifically, a new compare instruction is used to write the result of comparison in a floating-point register and a conditional execution instruction is used based on the result of such comparison. As the conditional execution instructions, data transfer between floating-point registers and store from a floating-point register to memory have been defined. Combining these instructions to eliminate conditional branching instructions allows the compiler to optimize loops containing *if* statements by software pipelining or other means.

6) Division and square root approximation

Instructions for finding reciprocal approximations have been added. This has allowed division and square root pipeline processing, resulting in a throughput improvement of about four times as a combined effect with the greater number of registers.

Functions 1) to 6) all make it possible to have a better performance without increasing the frequency and significantly contribute to

FUJITSU Sci. Tech. J., Vol. 48, No. 3 (July 2012)

277

the higher performance per unit of power of SPARC64 VIIIfx.

5. VISIMPACT

This section describes the hardware mechanisms used for VISIMPACT.

1) Shared level 2 cache

SPARC64 VIIIfx is provided with a 6 MB level 2 cache shared by all of the eight cores. Making it easier to share data between cores allows efficient parallel processing of one process with multiple cores.

2) Hardware barrier

SPARC64 VIIIfx is equipped with a hardware barrier for high-speed synchronization between cores. When one process is executed by multiple cores in parallel, *wait* (synchronization) may be implemented between cores. While ordinary processors use software for synchronization, SPARC64 VIIIfx uses dedicated hardware to increase the computational speed by more than ten times. The significant reduction in synchronization overhead allows small loops to be processed in parallel by using multiple cores for higher speeds.

6. Low power consumption

SPARC64 VIIIfx uses long-gate transistors and water cooling as a cooling method to lower the junction temperature to 30°C, thereby decreasing the leakage power to 10% of the power of the entire chip.

In addition, complete clock gating is provided for each latch so that it is more effective in power reduction, which has successfully decreased the dynamic power consumed in operation.

As a result, the average power consumption of the SPARC64 VIIIfx is as low as 58 W and it has a high computing performance of 128 GFLOPS. This is more than six times that of the SPARC processor, our previous model, in terms of performance per unit of power.

7. High reliability

SPARC64 VIIIfx is provided with high-reliability technology that Fujitsu has nurtured through the development of mainframes and UNIX servers.

A processor is composed of very fine-structured transistors, and signals may be affected by cosmic rays or other factors. To ensure continued processing without any malfunction in spite of such intermittent and transient faults, SPARC64 VIIIfx has an instruction retry mechanism in which hardware automatically re-executes any instruction affected by faults. In addition, 1-bit errors of all RAM and floating-point and fixed-point registers in the processor are corrected by hardware. Sections relating to program execution are protected by error detection codes so as to ensure data integrity.

By making use of these technologies, Fujitsu has achieved stable operation of a system connecting more than 80 000 processors.

8. Conclusion

The development of SPARC64 VIIIfx was a really challenging project for us developers. For the development, members from the software development department, laboratories and other departments in addition to the processor development team gathered together to combine all of Fujitsu's strengths. We believe that developing new technologies and inheriting the processor technologies that Fujitsu has nurtured over many years have led to the successful development of a supercomputer.

We anticipate that the K computer, which uses this processor, will help solve problems in various fields in the future.

References
1) T. Maruyama et al.: SPARC64 VIIIfx: A New-Generation Octocore Processor for Petascale Computing. *IEEE Micro*, Vol. 30, Issue 2, pp. 30–40 (2010).
2) SPARC International: The SPARC Architecture Manual (Version 9).
http://www.sparc.org/standards/SPARCV9.pdf

3) Fujitsu: SPARC Joint Programming Specification (JPS1) Commonality. (in Japanese).
http://jp.fujitsu.com/solutions/hpc/brochures/
4) Fujitsu: SPARC JPS1 Implementation Specification SPARC64 V. (in Japanese).
http://jp.fujitsu.com/solutions/hpc/brochures/
5) Fujitsu: SPARC64 VIIIfx Extensions. (in Japanese).

http://jp.fujitsu.com/solutions/hpc/brochures/
6) A. Inoue: SPARC64 V Processor for UNIX Servers. (in Japanese), *FUJITSU*, Vol. 53, No. 6, pp. 450–455 (2002).
7) T. Maruyama et al.: Past, Present, and Future of SPARC64 Processors. *Fujitsu Sci. Tech. J.*, Vol. 47, No. 2, pp. 130–135 (2011).

Toshio Yoshida
Fujitsu Ltd.
Mr. Yoshida is currently engaged in development of processor cores.

Ryuji Kan
Fujitsu Ltd.
Mr. Kan is currently engaged in development of processor cores.

Mikio Hondo
Fujitsu Ltd.
Mr. Hondo is currently engaged in performance evaluation of processors and systems.

Go Sugizaki
Fujitsu Ltd.
Mr. Sugizaki is currently engaged in development of processors.