VECTOR UNIT ARCHITECTURE FOR EMOTION SYNTHESIS

TWO VECTOR UNITS EMBEDDED IN THE EMOTION ENGINE CHIP SUPPORT HIGH-QUALITY 3D GRAPHICS, EMOTION SYNTHESIS, AND 300-MHZ, 5.5-GFLOPS OPERATION FOR THE RECENTLY INTRODUCED PLAYSTATION2 GAME ENTERTAINMENT SYSTEM.

Atsushi Kunimatsu
Nobuhiro Ide
Toshinori Sato
Yukio Endo
Hiroaki Murakami
Takayuki Kamei
Masashi Hirano
Fujio Ishihara
Haruyuki Tago
Toshiba Corporation

Masaaki Oka
Akio Ohba
Teiji Yutaka
Toyoshi Okada
Masakazu Suzuoki
Sony Computer Entertainment

Processors designed for computer entertainment must perform 3D graphics calculations, especially geometry and perspective transformations. In the PlayStation2, we introduced the idea of synthesizing emotion called Emotion Synthesis and devised a new processor architecture to support its graphics demands.[1] The architecture is embodied in the PlayStation2's Emotion Engine CPU,[2] which uses vector units (VUs)[3] as the key units for floating-point calculations.

Emotion synthesis means the real-time synthesis of a computer graphics animation scene that projects a great deal of atmosphere. For example, when a female character walks into a video game scene, her motion must be determined by solving physical equations in response to interactive events instead of replaying prerecorded data. Moreover, differential equations with a large number of variables must be used to describe, for example, the waving motions of her hair in a breeze. For authenticity in emotion synthesis, the CPU must execute these calculations in real time.

The Emotion Engine has achieved a peak performance of 5.5 Gflops at an operation frequency of 300 MHz. Its vector units operate in two modes: *VLI*, for use as a stand-alone processor and *coprocessor* for use as a MIPS COP2. This arrangement allows a vector unit to simultaneously execute huge amounts of 3D graphics calculations and flexible calculations in collaboration with the CPU core.[2] Both calculation types are indispensable in producing emotion synthesis. By employing software pipelining techniques, a vector unit can execute 3D perspective transformations with seven-cycle throughput at 300 MHz (85 Mvectors/sec). The vector unit in the 0.25-micron CMOS Emotion Engine chip contains 5.8 million transistors in 8.76 mm × 7.87 mm.

Architecture design strategy

The 3D graphics calculations in emotion synthesis require perspective transformation and lighting calculations, which are achieved through well-established algorithms. We designed vector unit VU1 for this purpose, enabling larger instruction and data memory comparison with vector unit VU0 and equipping it with better stand-alone performance. There is no direct control path from the CPU core.

An example of flexible calculation is a motion calculation of a body or a liquid; their algorithms are rich in variety. To collaborate with the CPU core, we closely coupled VU0 with the CPU core via a 128-bit coprocessor bus. An example of the roles that VU0 and VU1 play in a fighting game is a scene that consists of main characters fighting each other and background objects such as buildings or a cheering audience. Since the main characters need to move and change their shapes in real time for a player's

0272-1732/00/$10.00 © 2000 IEEE

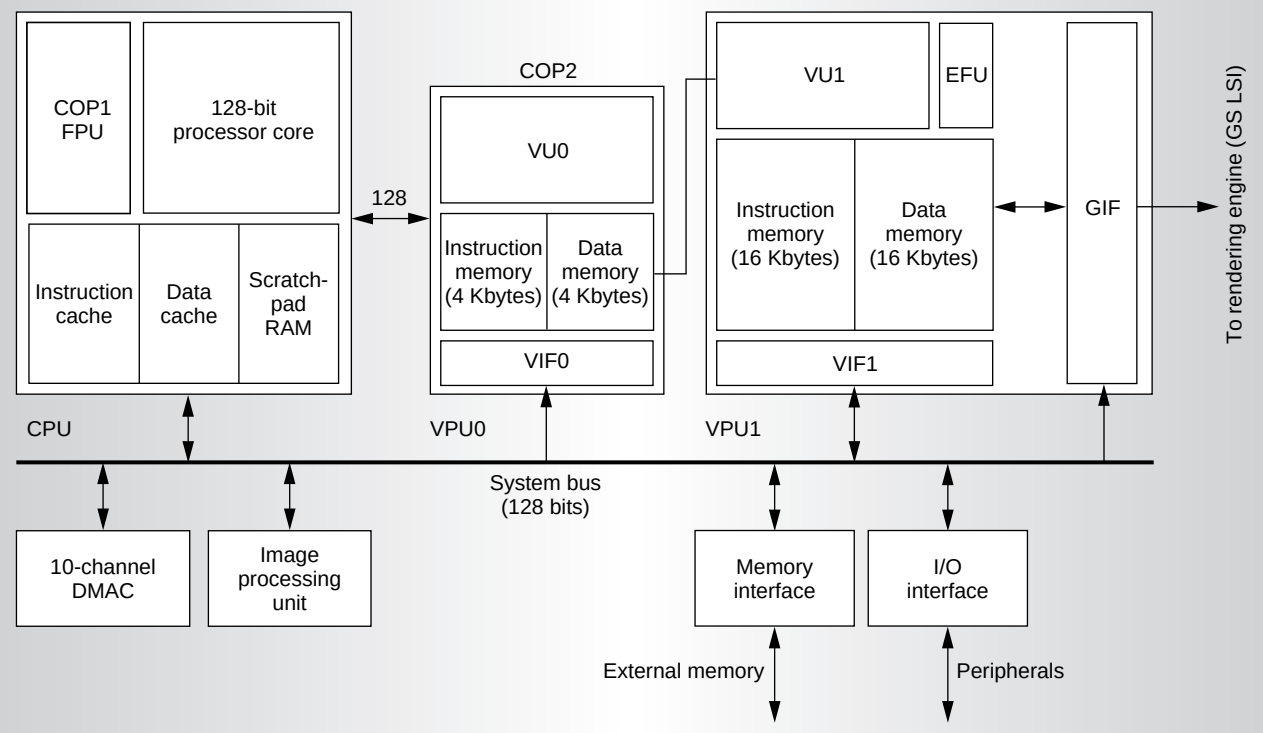

Figure 1. Emotion Engine block diagram.

input, the flexible calculations in VU0 process these needs. On the other hand, VU1 processes the background objects, which consist of many polygons.

The vector units feature four parallel fMACs (floating-point multiply accumulation units),[4] a high-speed fDIV (floating-point division unit), a broadcast mechanism, and a VLIW architecture. To compute 4×4 matrix operations efficiently, we use the four parallel fMACs with the broadcast mechanism. We also employ VLIW instruction formats and the high-speed fDIV for efficient perspective transformations and vector normalizations.

VU architecture

Figure 1 shows the Emotion Engine block diagram. The chip contains three processors: the CPU core, VU0, and VU1. The CPU core[2,5] contains 128-bit registers, 128-bit ALUs for multimedia processing, and a coprocessor interface for VU0 and VU1.

A vector processing unit (VPU) block includes vector units VPU0 and VPU1, instruction and data memories, and some interface units. The interfaces are VU interface 0 (VIF0), VU interface 1 (VIF1), and graphics synthesizer interface (GIF). These interfaces are responsible for DMA access from and to the vector unit memories together with data expansion and compression. By using these interfaces, the vector unit, and its memory double-buffering techniques simultaneously, we achieve nonstop, continuous processes.

As mentioned earlier, while the VLIW mode is available in both VU0 and VU1, the coprocessor mode is available only in VU0. For the interfaces, VU0 also includes the coprocessor interface and VIF0, while VU1 includes VIF1 and the graphics synthesizer interface. VU0 includes a 4-Kbyte instruction RAM and a 4-Kbyte data RAM. VU1 includes a 16-Kbyte instruction RAM and a 16-Kbyte data RAM. VU1 also has an elementary function unit (EFU). By using the fMAC's 0.6 Gflops and the fDIV's 0.04 Gflops for performance calculation, the VU0 reaches a peak performance of 2.44 Gflops, and VU1 achieves 3.08 Gflops for a total performance of 5.52 Gflops.

VU1 is a stand-alone processor mainly responsible for conventional 3D graphics calculations. Therefore VU1 must process larger amounts of data and calculations than the

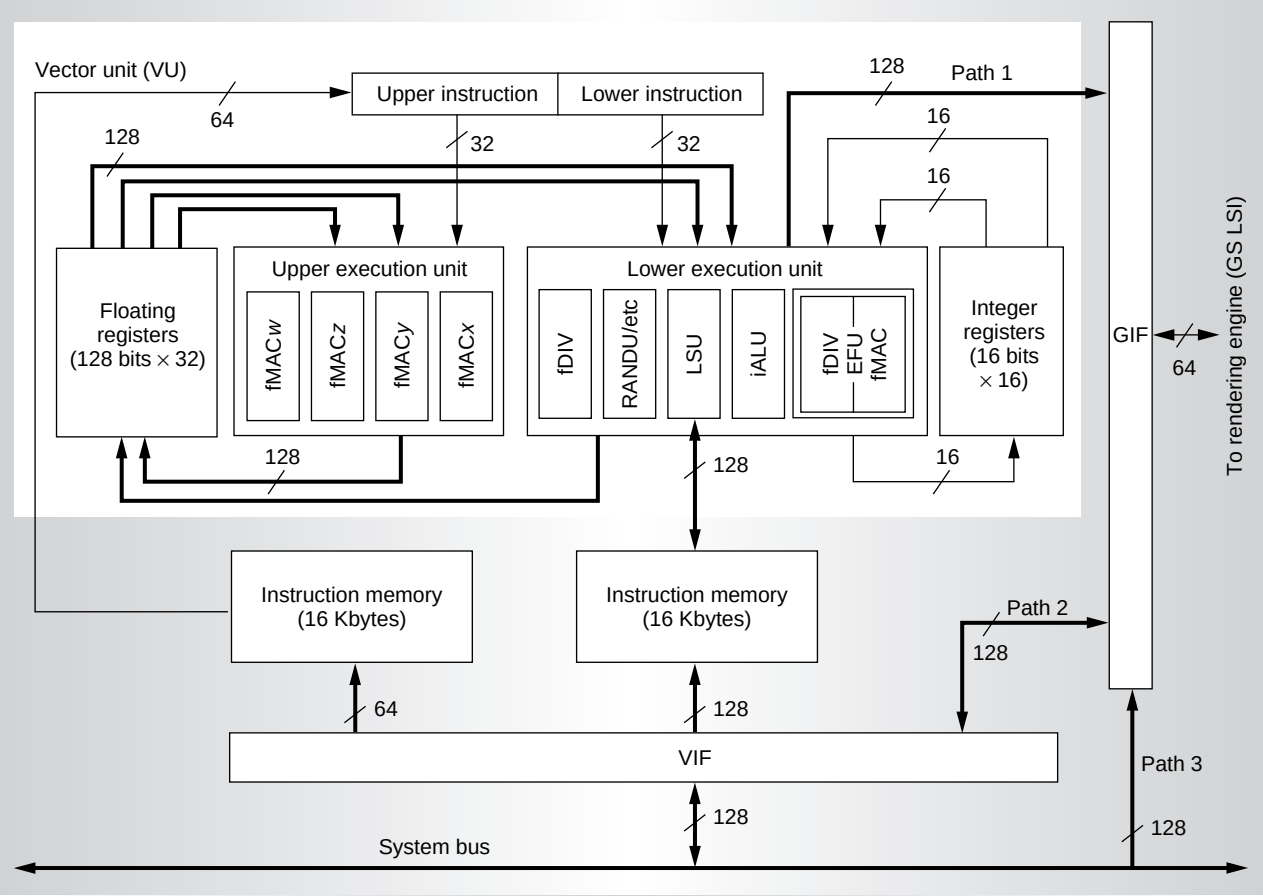

Figure 2. VPU1 block diagram.

Table 1. VU0 and VU 1 features.

Feature	VU0	VU1
Main task	Flexible calculation with CPU control	Well-defined 3D calculations
VLIW mode	Available	Available
Coprocessor mode	Available	Not available
VPU	Instruction memory (4 Kbytes)	Instruction memory (16 Kbytes)
	Data memory (4 Kbytes)	Data memory (16 Kbytes)
	VIF (system bus interface)	VIF (system bus interface)
		GIF (graphics interface)
		EFU (vector unit option)
Total performance	fMAC × 4 (2.40 Gflops)	fMAC × 4 (2.40 Gflops)
(5.5 Gflops)	fDIV (0.04 Gflops)	fDIV (0.04 Gflops)
		EFU (0.64 Gflops)

VU0. VU1 has four times more memory than VU0 and the additional elementary function unit. This unit includes an fMAC and an fDIV to calculate elementary functions such as exp, sin, and so on.

However, this job assignment for the vector units is just one of the examples of emotion synthesis. Individual user programmers can alter it, depending on interpretation of the chip. For example, they can execute conventional 3D graphics calculations on both of the vector units.

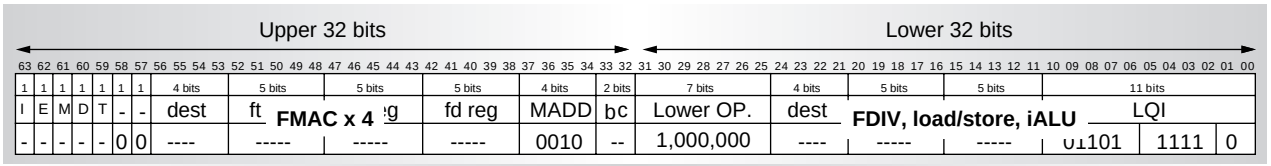

Figure 3. VLIW-mode, 64-bit VLIW instruction format.

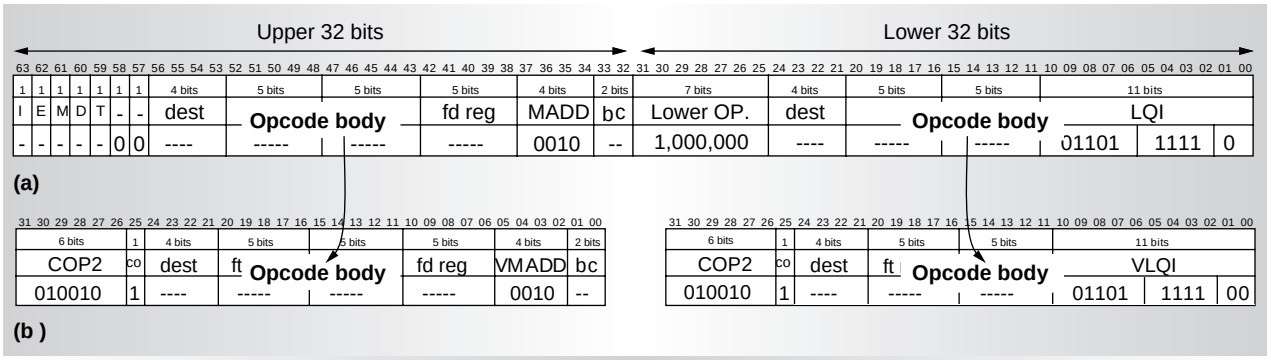

Figure 4. Instruction format to support two modes: 64-bit VLIW (a) and 32-bit COP2 (b).

Figure 2 shows the block diagram of the VPU1.[5] The major internal blocks of the vector unit are

1. an upper execution unit with four parallel fMACs,
2. a lower execution unit with fDIV, load/store, iALU, and branch,
3. 128-bit × 32 floating registers, and
4. 16-bit × 16 integer registers.

Table 1 lists the VU0 and VU1 features.

In the VLIW mode, each 64-bit VLIW instruction format is split into upper instruction and lower instruction parts. In the coprocessor mode, each 32-bit COP2 instruction consists of a COP2 header and an "opcode body" of the upper or lower instruction parts. Only an upper or a lower instruction can be selected in a COP2 instruction.

Figure 3 provides operation examples of the four parallel fMACs. The upper half of this figure shows a normal four-parallel SIMD multiply-accumulation operation. The lower half shows a four-parallel SIMD multiply-accumulation operation with broadcasting.

Figure 4 shows the VLIW mode pipeline stages and the coprocessor mode pipeline stages. The fMAC executes sequential single-precision floating-point multiply-accumulate operations with a throughput of one cycle. The fDIV executes a single-precision floating-point divide/square-root operation with a throughput/latency of seven cycles. Floating-point numbers calculated by the VU are compatible with the IEEE-754 format, while its rounding technique is not.

The coprocessor mode has two kinds of instructions. One corresponds to an upper instruction, a lower instruction, and a COP2 instruction. The other is a "call VLIW mode" instruction. Prior to using this instruction, the program stores VLIW mode instructions and large data to the VU0 memory first using DMA via VIF0. Then the program executes a call VLIW mode instruction as a subroutine call instruction. For example, in the case of calculating a physical dynamics simulation, an inner loop may be coded as a VLIW mode program, while the CPU core controls complicated flows.

Instruction sets

The vector units have various instruction sets. The VLIW mode has 164 instructions: 95 upper instructions (which include 68 instructions with broadcasts) and 69 lower instructions. The coprocessor mode has 130 instructions, including most of the upper instructions and the lower instructions, and all of the COP2 instructions.

Table 2 (next page) summarizes the upper instruction varieties.

Table 2. Upper instruction. (GPR: general-purpose register; ACC: accumulation register; w/w.o.: with/without; 1/2/3/4: selectable one to 4 parallel)

		Variation	
Operation	**Broadcast**	**Parallel degree**	**Output register**
Addition	w/w.o.	1/2/3/4	GPR/ACC
Subtraction	w/w.o.	1/2/3/4	GPR/ACC
Multiply	w/w.o.	1/2/3/4	GPR/ACC
Multiply-add	w/w.o.	1/2/3/4	GPR/ACC
Multiply-subtract	w/w.o.	1/2/3/4	GPR/ACC
Maximum	w/w.o.	1/2/3/4	GPR/ACC
Minimum	w/w.o.	1/2/3/4	GPR/ACC
Outer product (pre)	w/w.o.	—	ACC
Outer product (post)	w/w.o.	—	GPR
Absolute	—	1/2/3/4	GPR
Convert F to I	—	1/2/3/4	GPR
Convert I to F	—	1/2/3/4	GPR
Clipping check	—	—	Status flag

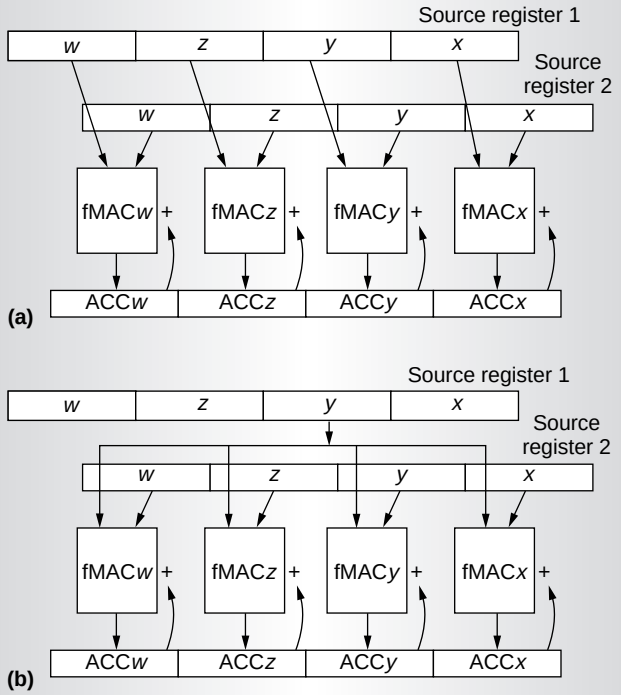

Figure 5. Geometry transformation pipeline: four parallel FMADDs without broadcast (a) and with broadcast (b).

The operations in the lower instruction are

- Division/square root/reciprocal square root;
- Integer addition/integer subtract/integer AND/integer OR;
- Integer addition with immediate operand;
- Move floating register to floating register;
- Move from the integer register to the floating register;
- Move from the floating register unit to the integer register unit;
- Rotate the 32-bit floating register;
- Load 128 bits with post increment/pre-decrement;
- Store 128 bits with post increment/pre-decrement;
- Integer load/store;
- Random unit instructions;
- Wait instruction for division operation;
- Flag operation instructions;
- Branch instructions; and
- EFU instructions.

Examples

By employing a broadcast mechanism and the four parallel fMACs with a throughput of one cycle, the vector unit can execute a 4×4-matrix geometry transformation with only four instructions (see Figure 5a,b). By using two operation issues in the VLIW mode, the fDIV with seven-cycle latency, a 128-bit load/store unit, and software pipelining techniques, the vector unit can execute a perspective transformation with a throughput of seven cycles.

The flow of software pipelining for perspective transformation follows:

1. Read four single-precision floating numbers (x, y, z, w) by a 128-bit load instruction with address post increment.
2. Execute geometry transformation.
3. Calculate $1/w$ using the results of the geometry transformation, which move to a temporary register to keep the current x, y, z.
4. Multiply x, y, z in the temporary register by $1/w$.
5. Write final results to memory by a 128-bit store instruction with address post increment.

With this flow, vector unit performance reaches as high as 85 Mvectors/sec at 300 MHz. This seven-cycle loop of perspective

transformation includes load/store operations of vertex data and loop controls.

Performance

Figure 6 compares our chip's performance with that of the Intel Pentium III SSE. The right bar indicates the performance of the Emotion Engine at 300 MHz, and the left bar shows that of the Pentium III SSE at 600 MHz. Table 3 lists the conditions and assumptions for this comparison.

For the vector unit performance values, we developed an actual program and counted the number of execution clocks. Figure 6 reflects the combined performance of VU0 and VU1. For Pentium III SSE, we assume the maximum software pipelining efficiency where the latencies of the instructions limit the performance. We employ the latency values from Intel reference manuals.[6] In the Pentium III SSE, we do not use approximate functions in order to keep the same condition on computational precisions. With respect to physical dynamics simulations frequently processed in emotion synthesis, we believe approximate functions are not usable.

This performance chart reveals that the Emotion Engine at 300 MHz performs at twice that of the 600-MHz Pentium III SSE. In practical applications, the performance gap may be wider. This is because by using DMA transfer and the VIF/GIF, the vector units can operate without stopping as long as the 128-bit system bus at 150 MHz and the two-channel Direct RDRAM interface are not jammed.

Implementation

We implemented the Emotion Engine in 0.25-micron CMOS process technology with a 0.18-micron gate length. Figure 7 shows a micrograph of this chip. The 15.02-mm × 15.04-mm die contains 13.5 million transistors; it operates at 300 MHz and typically consumes 18 watts of power. The literature describes the detail of design methodology.[7-10] The vector unit portion of the die, shown in the micrograph in Figure 8 (next page), contains 5.8 thousand transistors and measures 8.76 mm × 7.87 mm.

Our goal of achieving high-quality 3D graphics performance and emotion synthesis in the vector unit architecture has been

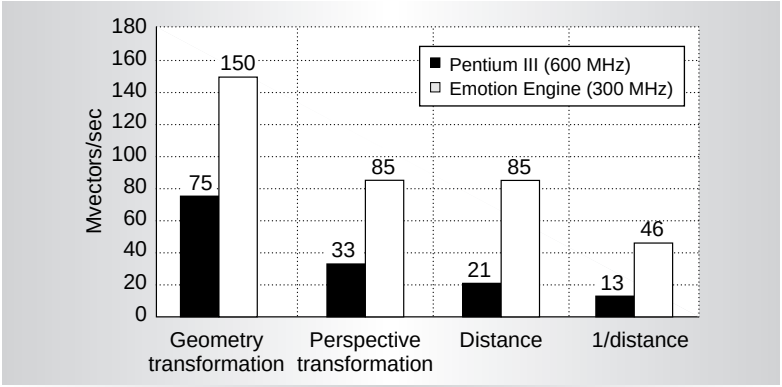

Figure 6. Performance comparison: Emotion Engine (right bars) and Pentium III SSE (left bars).

Table 3. Conditions and assumptions for performance comparison in Figure 6.

Task	Critical operation
Geometry transformation	4 × 4 parallel multiply-adds
Perspective transformation	Division
Distance calculation	Square root
Reciprocal distance calculation	Reciprocal square root

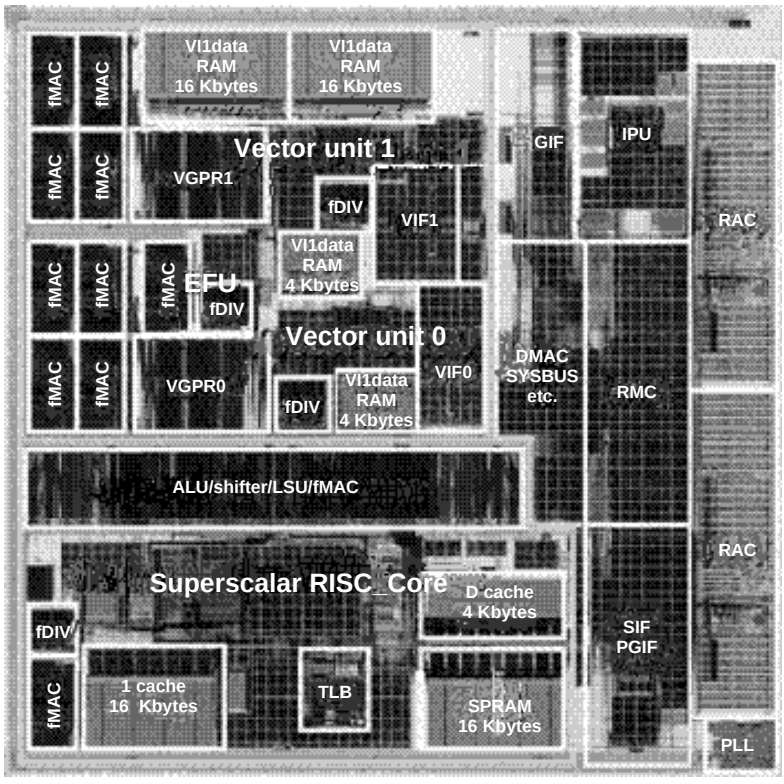

Figure 7. Emotion Engine micrograph.

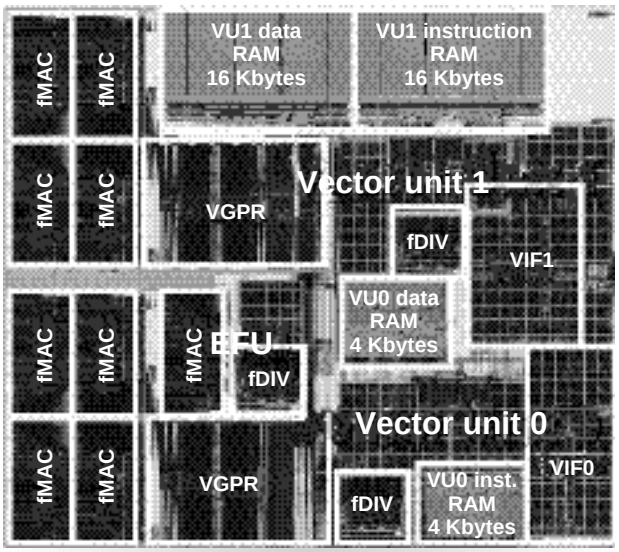

Figure 8. Vector unit micrograph.

successful. The architecture—including the two different vector units equipped with both VLIW and coprocessor modes—can simultaneously process flexible calculations as well as conventional 3D graphics calculations. By using a broadcast mechanism, the fMACs with a throughput of one cycle, and the fDIVs with a latency of seven cycles, the vector units achieve 85-Mvectors/sec perspective transformation performance.

We are now researching processor architecture for the next-generation computer entertainment system. MICRO

References

1. K. Kutaragi et al., "A Microprocessor with 128b CPU, 10 Floating-Point MACs, 4 Floating-Point Dividers, and MPEG2 Decoder," *ISSCC (Int'l Solid-States Circuit Conf.) Digest Tech. Papers,* IEEE Press, Piscataway, New Jersey, Feb. 1999, pp. 256-257.
2. F. Michael Raam et al., "A High-Bandwidth Superscalar Microprocessor for Multimedia Applications," *ISSCC Digest Tech. Papers,* IEEE Press, Feb. 1999, pp. 258-259.
3. A. Kunimatsu et al., "5.5 GFLOPS Vector Units for Emotion Synthesis," *Hot Chips 11 Conf. Record,* Aug. 1999, pp. 71-82.
4. N. Ide et al., "2.44 GFLOPS 300MHz Floating-Point vector Processing Unit for High-Performance 3D Graphics Computing," *Proc. European Solid-State Circuits Conf. (ESSCIRC 99)*, Editions Frontieres, Gif-sur-Yvette, France, ISBN 2-86332-246-X, 1999. pp. 106-109.
5. M. Oka and M. Suzuoki, "Designing and Programming the Emotion Engine," *IEEE Micro,* Jan.-Feb 2000, pp. 20-28.
6. Intel Corporation, "Streaming SIMD Extensions Throughput and Latency," *Intel(R) Architecture Optimization Reference Manual*, 1999, pp. D-1; http://support.intel.com/design/pentiumiimanuals/245127.htm.
7. H. Tago et al., "Importance of CAD Tools and Methodologies in High Speed CPU Design." *Proc. Asia and South Pacific Design Automation Conf. (ASP-DAC 2000),* ACM, New York, Jan. 2000, pp. 631-633.
8. T. Kamei et al., "300MHz Design Methodology of VU for Emotion Synthesis," *Proc. ASP-DAC 2000,* ACM, Jan. 2000, pp. 635-640.
9. N. Kojima et al., "Repeater Insertion Method and Its Application to the 300MHz 128-Bit 2-Way Superscalar Microprocessor," *Proc. ASP-DAC 2000,* ACM, Jan. 2000, pp. 641-646.
10. F. Ishihara et al., "Clock Design of 300MHz 128-Bit 2-Way Superscalar Microprocessor," *Proc. ASP-DAC 2000,* ACM, Jan. 2000, pp. 647-652.

Atsushi Kunimatsu, Nobuhiro Ide, Yukio Endo, Hiroaki Murakami, Takayuki Kamei, Masashi Hirano, Fujio Ishihara, and Haruyuki Tago work for the Toshiba Semiconductor Company in the System ULSI Engineering Laboratory in Kawasaki, Japan. Kunimatsu develops logic LSI chips. He received his BSEE and MS in computer science from Keio University, Kanagawa, and is a member of the IEEE. Ide holds BSEE and MSEE degrees from Waseda University, Tokyo, and is a member of the IEEE. Endo develops logic LSI chips. He received the BS degree in material engineering from Tokyo University. Murakami works in the Microprocessor Design Department. He holds a BS degree in mathematical engineering from the University of Osaka Prefecture in Osaka, Japan. Kamei is a processor development engineer. He received a BEEE and MS in computer Science from Keio University. Hirano is engaged in the research and development of CMOS logic VLSI chips. He graduated from the Ohsawano Technical High School in Toyama, Japan. Ishihara holds BSEE and MSEE degrees from Keio University. Tago

is working on the design of a microprocessor core and a system LSI chip for entertainment applications. Earlier, he was a visiting scholar at the Computer Science Department of the University of Illinois.

Toshinori Sato is an associate professor in the Department of Artificial Intelligence at Kyushu Institute of Technology in Iizuka, Japan. Sato holds BE, ME, and PhD degrees in electronic engineering from Kyoto University. He is a member of the IEEE Computer Society; ACM; Institute of Electronics, Information and Communication Engineers; and Information Processing Society of Japan.

Masaaki Oka, Akio Ohba, Teiji Yutaka, Toyoshi Okada, and Masakazu Suzuoki work for Sony Computer Entertainment in Tokyo. Oka holds a BS degree in science from Kyoto University. Ohba works in the Architecture Laboratory and holds an MS degree in biophysical engineering from Osaka University. Yutaka received his BSEE from Keio University, Tokyo, and develops real-time computer graphics systems such as the PlayStation. Okada received the BEng degree in electrical engineering from Keio University and develops video game consoles such as the PlayStation. Suzoki works in the Software Development Department and holds an MS degree in electronic engineering from Tokyo University.

Direct comments concerning this article to Atsushi Kunimatsu, System ULSI Engineering Laboratory, Toshiba Corporation; atsushi.kunimatsu@toshiba.co.jp.