INVITED PAPER Special Issue on Multimedia, Network, and DRAM LSIs

Future Directions of Media Processors

Shunichi ISHIWATA†a) and Takayasu SAKURAI††, Members

SUMMARY Media processors have emerged so that a single LSI can realize multiple multimedia functions, such as graphics, video, audio and telecommunication with effectively shared hardware and flexible software. First, the difference between media processors and general-purpose microprocessors with multimedia extensions is clarified. Features for processes and data in the multimedia applications are summarized and are followed by the multimedia enhancements that the recent general-purpose microprocessors use. The architecture for media processors reflects the further optimized utilization of these features and realizes better price-performance ratio than the general-purpose microprocessors. Finally, the future directions of media processors are estimated, based on the performance, the power dissipation and the die size of the present microprocessors with multimedia extensions and the present media processors. The demand to improve the price-performance ratio for the whole system and to reduce the power consumption makes the media processor evolve into a system processor, which integrates not only the media processor but also the function of a general-purpose microprocessor, various interfaces and DRAMs.

key words: *media processor, VLIW, multimedia extension, wide memory bandwidth, low power consumption*

1. Introduction

In order to accelerate multimedia functions such as graphics, video and audio, many dedicated single-function LSIs have been developed [1], [2]. The rapid progress implemented in the semiconductor technology makes it possible to unify these accelerators on one chip. However, simply putting them together is not always the best solution. Media processors [3]–[7] have emerged so that processing resources and memories are shared effectively and a variety of multimedia functions are implemented flexibly by software. Media processors will evolve into system processors, which integrate not only performance improved media processors but also the functions of general-purpose microprocessors and a variety of interfaces, through exploiting more parallelism and embedded memories.

2. Definition of a Media Processor

A media processor is a DSP (digital signal processor) which is dedicated to multimedia applications, such as 2D/3D graphics, video, audio and telecommunication, where the video includes:

- MPEG1/MPEG2 decoding
- H.263 encoding and decoding

The audio contains:

- MPEG decoding
- Dolby digital decoding
- Wave table sound synthesis
- FM sound synthesis

The telecommunication includes:

- Modem
- FAX
- Telephony

Single-function DSPs, such as audio DSPs, video DSPs or graphics accelerators, are not new. However, the media processors do perform multiple multimedia functions by just changing software.

On the other hand, compared to general-purpose microprocessors with multimedia functions the media processors are designed so effectively that they deliver better cost-performance. A media processor is different from a general-purpose microprocessor in that the media processor does realize more efficient implementation for multimedia functions at the cost of reduced generality. It is not specially suitable for word processing nor spread sheet applications, but is very efficient in handling multimedia data and algorithms. For this end, the architecture and implementation are different from the general-purpose processor. With this differentiation, media processors provide better power, die size and cost index for multimedia functions than are furnished by the general-purpose processors, which will be described later in detail.

3. Analysis of Multimedia Data and Processing

An analysis and understanding of the multimedia data and processing are required to enable investigating the architecture for media processors. Table 1 shows the critical operations and the data formats required for the multimedia processing [2]. From Table 1 and an analysis of the multimedia data, one can find the following:

- Most of the multimedia functions require only 8 to 16 bits of precision.
- Multimedia data have poor temporal locality,

Manuscript received September 19, 1997.
Manuscript revised January 28, 1998.

† The author is with Microelectronics Engineering Lab., Toshiba Corporation, Kawasaki-shi, 210-8520 Japan.
†† The author is with the Institute of Industrial Science, The University of Tokyo, Tokyo, 106-8558 Japan.
a) E-mail: sisiwata@sdel.eec.toshiba.co.jp

Table 1 Critical operations and data formats in multimedia functions.

Function		Operations	Data format
2D Graphics	Raster operation	generic logic operation	8 bit integer
	Chroma keying	A==B, A?B:C	8 bit integer
3D Graphics	Geometry	$\Sigma a_i * x_i$	32 bit float
	Setup	A-B, A*B, 1/A	32 bit float
	Z buffering	A>B, A+B	16 bit fixed
	Gouraud shading	A+B	16 bit fixed
	Texture filtering	A*(1-C)+B*C, $\Sigma a_i * x_i$	8 bit integer
Video	Quantization	A/B	16 bit integer
	Inverse quantization	A*B, A+B, shift, saturation	16 bit integer
	DCT/IDCT	A*B+C, butterfly (A+B, A-B)	16 bit integer
	Motion compensation	(A+B+1)>>1, A+B, saturation	9 bit integer
	Scaling	A*(1-C)+B*C, $\Sigma a_i * x_i$	8 bit integer
	Color space conversion	$\Sigma a_i * x_i$	8 bit integer
	Motion estimation	$\Sigma \lvert x_i - y_i \rvert$, $\Sigma (x_i - y_i)^2$	8 bit integer
Audio		$\Sigma a_i * x_i$	16 to 24 bit fixed

Table 2 Rough estimation of the performance required by multimedia functions. The above estimation does not include load/store operations.

Function		Required performance (MOPS)	
		Present	Future
3D Graphics	Resolution	640 x 480	1600 x 1200
	Frame rate	30 Hz	60 Hz
	Average superposition	5	8
		1 Mpolygon/sec 50 Mpixel/sec	20 Mpolygon/sec 1,000 Mpixel/sec
	Geometry	70	1,400
	Setup	90 - 200	1,800 – 4,000
	Z buffering	200	4,000
	Gouraud shading	90	1,800
	Texture mapping	200	4,000
	Texture filtering	1,300 – 3,000	26,000 – 60,000
Video		MPEG2 MP@ML	MPEG2 MP@HL
	Resolution	720 x 480	1920 x 1080
	Frame rate	30 Hz	30 Hz
Encode	Motion estimation	8,000 – 60,000	48,000 – 360,000
	DCT / IDCT	500	3,000
	Quantization / Inverse Quantization	80	480
Decode	Inverse quantization	40	240
	IDCT	250	1,500
	Motion compensation	250	1,500
	Scaling	100 – 300	600 – 1800
	Color space conversion	260	1,600

because the multimedia processing is stream processing.

From Table 1 and an analysis of the multimedia processing, one can derive the following:

- The multiply-accumulate operation is very common.
- The saturated arithmetic operation is necessary for the video processing.
- The conditional branch is not used very frequently.
- The pipeline usually works well without being disturbed.
- The inner loop dominates the processing time.
- The operations have inherently high parallelism.
- The processing latency is strictly limited, except for the graphics.

Table 2 shows a rough estimation of the performance required by multimedia functions.

Table 3 Multimedia enhancements in general purpose microprocessors. The blank columns mean the authors do not know.

	Sun VIS	HP MAX2	Intel MMX	DEC MVI	MIPS MDMX
No. of registers	32	31	8	31	32
Register used	MM/FP	Integer	MM/FP	Integer	MM/FP
Packed operations	16 bit × 4 32 bit × 2	16 bit × 4	8 bit × 8 16 bit × 4	8 bit × 8	8 bit × 8 16 bit × 4
Unsaturated add/sub	○	○	○	×	×
Saturated add/sub	×	○	○	×	○
No. of multipliers	4	×	4	×	4 or 8
Data format of multiply-accumulate	8 × 16→16	shift + add	16 × 16→32		8 × 8→24 16 × 16→48
Packed shift	×	○	○	×	○
Packed average	×	○	×	×	×
Packed compare	○	×	○	×	○
Pack/Unpack	○	○	○	○	○
Permute	×	○	×	×	×
Packed $\Sigma \lvert A-B \rvert$	○	×	×	○	×
Packed FP operation	×	×	×	×	○
Block Load/Store	○	×	×	×	×

4. Multimedia Enhancements in General-Purpose Microprocessors

Studying the multimedia enhancements in general-purpose microprocessors is very useful to enable understanding the media processors. Table 3 summarizes the multimedia enhancements in general-purpose microprocessors [8]-[12]. From Table 3, one can find that the important techniques to accelerate the multimedia performance are as follow:

- Packed arithmetic operation. It utilizes the rather low precision of multimedia data and the parallelism of the multimedia processing. It divides the data into multiple sub-words and executes an operation for each sub-word in parallel. It can be implemented with a very small number of extra transistors by just controlling the carry propagation.
- Saturated arithmetic operation. It is useful not only to accelerate multimedia operations, such as filtering, but also to prevent a conditional branch that causes the processing time to become unpredictable.
- Multiply-accumulator with one cycle throughput. It also has higher precision to prevent the overflow of intermediate results.
- Pack/unpack instruction.
- Application-specific instructions. The compare and select instruction is useful for the chroma keying, because they can eliminate a conditional branch. The packed $\sum|A-B|$ instruction is very useful to accelerate the motion estimation function in the video compression. The packed floating-point operations accelerate the geometry calculation in the 3D graphics.
- Block load/store instruction. The poor temporal locality of the multimedia data often makes the data transfer between the register file and the data cache a bottleneck. The block transfer instruction is useful to reduce load/store instructions issued.

5. Architecture of the Media Processors

Multimedia application demands high performance, as Table 2 shows. Fortunately, the algorithm for media processing has inherent high parallelism. In addition, media processors have had no need to be compatible to the accumulation of software in the past and hence have no barrier to adopt a new instruction set architecture. Therefore, most media processors take the VLIW architecture, which means the ALUs actually used are determined statically by a field in an instruction. This leads to effective use of higher parallelism and better optimization with a simple decoder and small control hardware, which realizes smaller die area. On the contrary, the super scalar microprocessors, which control the assignment of ALUs by hardware, suffer from a big overhead, such as an out-of-order execution controller. The VLIW can be considered as another further step from hardware optimization to software optimization in processors, which took place in the transition from a CISC to a RISC.

However, the VLIW architecture has its own issues. It should be noted that some media processors have mechanisms to overcome these VLIW issues. The VLIW means a very long instruction word and programs are prone to occupy more area in the instruction cache. Some media processors have a mechanism to compress instructions to overcome this problem. In addition, not all sub-instructions in an instruction can always be executed in parallel. Most media processors have a mechanism that divides an instruction into two or more groups of sub-instructions and executes them sequentially, so that unnecessary NOPs do not have to be inserted.

The VLIW architecture has another issue in that conditional branches limit the performance. The more sub-instructions are issued per cycle, the more bubbly sub-instructions are produced by a branch misprediction. Philips TriMedia™ and Mitsubishi's D30V make all instructions conditional to utilize some of the bubbles or to eliminate conditional branches. The conditional execution can reduce branch misprediction bubbles without an expensive branch prediction unit.

Another issue of the VLIW architecture is its high barrier to improve the performance by changing the micro-architecture while maintaining binary code compatibility. The inheritance of software property is very important, although the application software that runs on media processors tends to be a collection of short programs. Hence, the survival of VLIW processors depends on whether a compiler can guarantee the source code compatibility without degrading its performance. Progress achieved of the binary code translation and emulation technology will be another solution.

In order to improve the performance of an ALU, the media processors implement the same strategy as the general-purpose microprocessors with the multimedia extensions. That is:

- Packed arithmetic operation
- Saturated arithmetic operation
- Multiply-accumulator with one cycle throughput and higher precision intermediate results
- Pack/unpack instruction
- Application-specific instructions

In addition, the media processors tend to have more application-specific instructions and arithmetic units than the general-purpose microprocessors. For example, the Mpact2™ media processor has the following special unit, as Fig. 1 shows.

- A motion estimation unit for the video compression
- A rendering engine for the 3D graphics

These tasks require more than five billion operations per second, which cannot be achieved without the dedicated hardware. Another example is found in the TriMedia™. It has the following dedicated hardware:

- A variable length decoder for the MPEG decoding
- A scaling unit and a color space conversion unit for the video post processing

Furthermore, some media processors have the same dedicated hardware to accelerate the vector operations as most DSPs do. An example of the vector operation is:

$$\text{for } (i=0;\ i<n;\ i++)$$

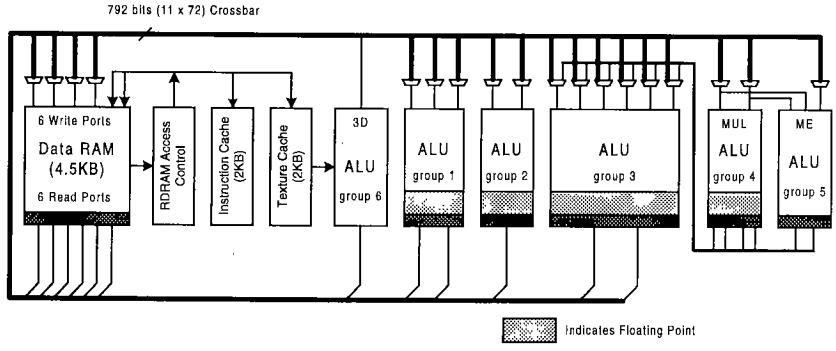


Fig. 1 Architecture of Mpact™-2 media processor.

$Acc\ +=\ \ *a++\ \ *\ \ *x++;$

This type of operation often appears in the innermost loop and dominates the processing time. Hence, improving the performance of the vector operations is critical. To accelerate them, some media processors have the dedicated hardware to control post-increment addressing and overhead-free loops. This strategy is effective, because the additional hardware is small.

The multimedia processing is stream processing, which means data stream as follows:

1. Data are read from the external memory.
2. They are processed. (The process usually consists of a small number of operations.)
3. They are written back to the external memory.

The innermost loop usually consists of a small number of operations. Hence, the bottleneck in the performance does not always involve the performance of an ALU. It could be either the bandwidth between the ALU and the register file, or the bandwidth between the register file and the data cache, or the bandwidth between the data cache and the external memory. In order to improve the overall performance, each individual performance has to be improved.

To improve the bandwidth between ALUs and the register file, the media processors have the following features:

- Very wide data bus. For example, the data bus in the Mpact2™ is 72 bit×11=792 bit wide. The 125 MHz clock frequency leads to the very wide bandwidth of 11 GB/s.
- Multi-ported register file. For example, the register file in the Mpact2™ has the 6 read ports and 6 write ports. The total bandwidth is 12 GB/s.
- Large register file. The large capacity makes it much easier to execute the following three processes in parallel: (a) reading a block of data, (b) executing a vector operation, and (c) writing a block of data. This type of process is often used in the multimedia processing.
- Result forwarding. In order to save the bandwidth of the register file, it is effective to forward the ALU's result to the input of the ALU without writing back to the register file. The Mpact2™ uses this technique.

The most remarkable feature is the size of the data cache. It can be classified into the following three categories:

1. Large data cache or large RAM. This approach is the same as that of the general-purpose microprocessors. However, the data cache often misses, even if it has the large capacity, because of the poor temporal locality of multimedia data. The media processors have data pre-fetch instructions (or load/store instructions in case of data RAM) so that the effective hit ratio for the data cache increases. This means the large data cache helps

Table 4 Features of the media processors for personal computers. The blank columns mean the authors do not know.

	Chromatic Mpact2	Philips TriMedia1	Samsung MSP1
Architecture	VLIW	VLIW	Vector processor
Clock frequency (MHz)	125	100	100
No. of issues per cycle	2	5	1
Bit width of ALU	72	32	288
Packed operations	9 bit×8 16 bit×4 32 bit×2	8 bit×4 16 bit×2 32 bit×1	9 bit×32 16 bit×16 32 bit×8
Peak performance of 16 bit operation	3.0	1.3	3.2
Unsaturated add/sub	○	○	
Saturated add/sub	○	○	
Saturate at the specified bit	×	×	
Packed shift	○	×	
Packed average	○	○	
Packed compare and select	○	×	
Pack/Unpack	○	○	
Permute	○	○	
Packed Σ|A-B|	○	○	
FPU	○	○	
Block Load/Store	○	×	
Zero-overhead loop	○	×	
Address auto increment	○	×	
Conditional execution	×	○	
DRAM	RAMBUS	Sync.	Sync.
DRAM bandwidth (MB/s)	1350	400	800
Instruction RAM (KB)	2	32	2
Data RAM (KB)	(4)	16	5
No. of registers	512	128	
Hardware scaling	×	○	
Hardware color space conversion	×	○	
Hardware VLD	×	○	○

Table 5 Features of the media processors for DVDs. The blank columns mean the authors do not know.

	Fujitsu MMA	Mitsubishi D30V	Matsushita MCP
Architecture	VLIW	VLIW	VLIW
Clock frequency (MHz)	180	250	54
No. of issues per cycle	2	2	4
Bit width of ALU	32	32	
Packed operations	16 bit×2 32 bit×1	16 bit×2 32 bit×1	
Peak performance of 16 bit operation	1.08	1	3.2
Unsaturated add/sub	○	○	
Saturated add/sub	○	×	
Saturate at the specified bit	○	○	
Packed shift			
Packed average		○	
Packed compare			
Pack/Unpack		○	
Permute			
Packed Σ|A-B|			
FPU	×	×	×
Block Load/Store			
Zero-overhead loop		○	
Address auto increment		○	
Conditional execution		○	
DRAM	Sync.	Sync.	Sync.
DRAM bandwidth (MB/s)	720		
Instruction RAM (KB)	8	32	17
Data RAM (KB)	8	32	17
No. of registers	32	64	24
Hardware scaling			
Hardware color space conversion	○		
Hardware VLD	×	○	

the easy scheduling of the data pre-fetch instructions at the cost of its large area.

2. Small data cache or small RAM. This is more cost-effective than the above approach, as long as the scheduling involved in the pre-fetch instructions is not too difficult.
3. Unified data RAM and register file. The Mpact™ takes this approach. It realizes equivalently both the large register file and the wide-band data RAM. The peak bandwidth for a data RAM of Mpact™-2 is 2 GByte/sec (144 bits at 125 MHz), while those for other media processors are only less than or equal to 1 GByte/sec. The wider band data RAM is more effective, even if it is smaller, because the large data cache does not increase the hit ratio for the data cache very well. The total area for the register file and the data cache is the smallest of these three approaches. A queue for the block load/store instructions makes the scheduling easier. As a result, this approach is most effective, from the price-performance point of view.

Another feature is a DRAM interface. Only the Mpact™ uses the Rambus™ DRAMs and has the widest memory bandwidth of 1.3 GB/s.

Table 4 and Table 5 summarize the feature for the media processors [3]-[7]. These tables are separated, because the media processors for the personal computers spend a non-trivial amount of hardware on the PCI bus interface and the 2D/3D graphics, while the media processors for the DVDs do not.

6. Future Media Processors

Let us estimate future media processors based on the relationship between their performance, power dissipation and die size. Figure 2 shows pertinent features for the current microprocessors and the media processors. From Fig. 2, one can find (a) microprocessors with multimedia extensions, (b) media processors for PCs and (c) media processors for digital consumer products form different groups. Hence, they are classified by these groups.

6. 1 Future Microprocessors with Multimedia Extensions

Figure 3 shows the authors' estimation regarding future directions. The high-end microprocessors will increase their performance, as Fig. 3 shows. The performance improvement will enable achieving the future functions in Table 2 by software, with the assist of a graphics accelerator or a media processor. However, the required performance will be far from attainable without the dedicated architecture, which sacrifices generality. On the other hand, low-end microprocessors will give weight to the multimedia

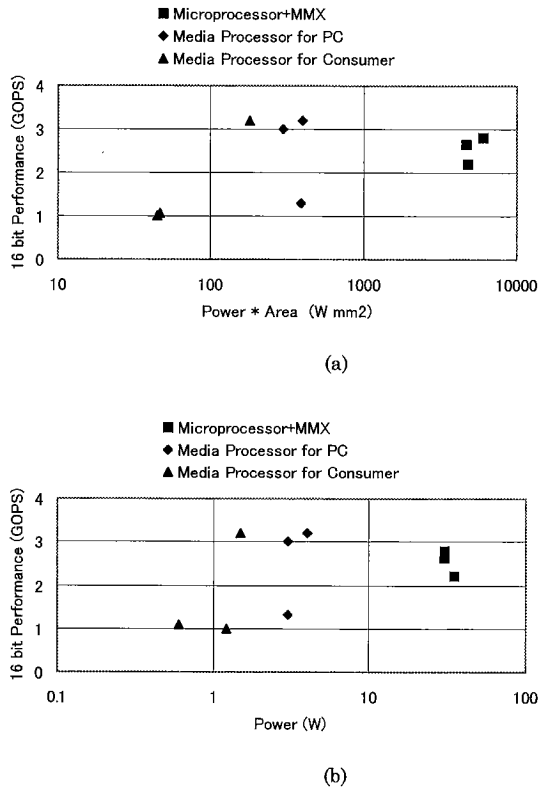


Fig. 2 (a) Relationship between the performance of 16 bit operations and the product of power consumption and die size. (b) Relationship between the performance of 16 bit operations and the power consumption.

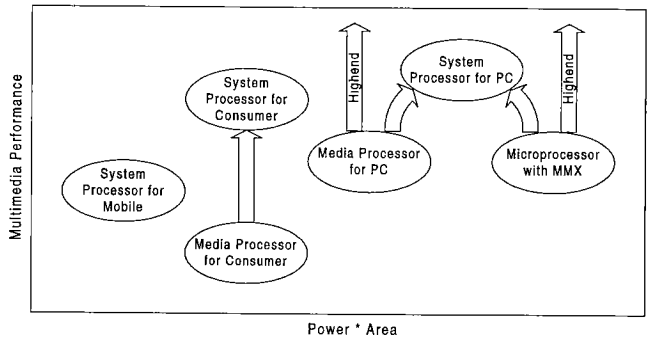


Fig. 3 Future directions of the media processors.

performance and the system cost, because multimedia applications are most important in the low-end PC market. Hence, they will aim at the processors that have the following features:

- High multimedia performance
- Moderate general-purpose performance
- Integration of peripherals

These processors are referred to as "system processors" in this paper.

6.2 Future Media Processors for PCs

The important and performance-critical applications will be:

- MPEG MP@HL decoding
- MPEG MP@ML encoding
- High quality 3D graphics

These tasks require more than 20 billion operations per second. The high quality 3D graphics require a wide memory band of more than 1.6 GB/s. As a result, media processors for high-end PCs will increase their performance, as Fig. 3 shows. To achieve these requirements, the future media processors will use the following technologies:

- The number of ALUs and the number of issues per cycle will be increased. The VLIW architecture will continue to be used to avoid increasing the hardware complexity.
- More transistors will be used for the dedicated units to improve the performance.
- Multiple sets of a register file and ALUs may be used, so that the bandwidth requirement for the data bus and the register file is eased.
- The frame buffer compression technique will be used to increase not only the effective memory capacity but also the effective memory bandwidth.
- For the high-end media processors for PCs, the integrated DRAM technology will be used to achieve a higher memory bandwidth than that which the Rambus™ DRAMs and other high-speed commodity DRAMs can provide.

On the other hand, media processors for low-end PCs will evolve to system processors, as mentioned in the previous section.

6.3 Future Media Processors for Digital Consumer Products

Media processors for digital consumer products, such as digital set top boxes and digital TVs, will increase their performance, as Fig. 3 shows, because the basic killer applications are the same as those for the low-end PCs. In addition, they will integrate more peripherals, so that the system cost will be further reduced than that for the low-end PCs at the cost of reducing flexibility and generality. Hence, they will also evolve to what the authors call system processors.

6.4 Future Media Processors for Mobile Information Terminals

Media processors will be suitable for mobile information terminals as well, because the power dissipation will be reduced as the process technology progresses, as a result, the important and performance-critical applications will be the following:

- Video compression/decompression using e.g. MPEG4
- Audio compression/decompression using e.g. G.723
- Graphics

These tasks require around one billion operations per second of performance. In addition, not only the power dissipation but also the system cost is very important. These requirements are conceptually the same as for the media processors above-mentioned. Hence, they will also become what the authors call system processors.

The low-power implementation is important in future system processors for multiple reasons. For the high-end system processors for PC, where delivering high-performance is the key factor, will suffer from a heat problem, since they will consume several tens of watts. On the other hand, the system processors for consumer applications, where cost is the issue of importance, require low-power design to enable them to be mounted in lower cost plastic packages. The last category for system processors, system processors for mobile use, should reduce the power consumption to lengthen the battery life during use.

Circuit techniques, such as MTCMOS [13] and VTCMOS [14], [15], and technology ideas, such as low-power oriented CMOS device optimization [16], are some of the useful techniques for achieving low-power in system processor implementations.

7. Conclusions

So far, media processors have integrated graphics, video and audio functions. They utilize a variety of techniques, such as packed operations and a multi-ported register file, to improve their price-performance ratio in these multimedia applications. In the future, media processors will have more parallelism and embedded memories, because multimedia applications require higher performance and wider memory bandwidth. On the other hand, the market for multimedia devices will spread further to consumers. The system cost and the power dissipation will be critical in this market. Hence, media processors will evolve into system processors, which integrate not only the media processors, but also the functions of general-purpose microprocessors and various interfaces.

References

[1] M. A. Bayoumi and E. E. Swartzlander, Jr., "VLSI signal processing technology," Kluwer Academic publishers, 1994.
[2] V. Bhaskaran and K. Konstantinides, "Image and video compression standards-alogorithms and architectures," second edition, Kluwer Academic publishers, 1997.
[3] T. Takayanagi, K. Nogami, F. Hatori, N. Hatanaka, M. Takahashi, M. Ichida, S. Kobayashi, T. Higashi, M. Klein, J. Thomson, R. Carpenter, R. Donthi, D. Renfrow,

J. Zheng, L. Tinkey, B. Maness, J. Battle, S. Purcell, and T. Sakurai, "350 MHz time-multiplexed 8-port SRAM and word-size variable multiplier for multimedia DSP," ISSCC Dig. Tech. Papers, pp. 150–151, Feb. 1995.

[4] P. Kalapathy and P. Holden, "Hardware/software interaction on the Mpact media processor," Hot Chips VIII symposium record, pp. 179–192, Aug. 1996.

[5] G. A. Slavenburg, S. Rathnam, and H. Dijkstra, "The Trimedia TM-1 PCI VLIW media processor," Hot Chips VIII symposium record, pp. 171–178, Aug. 1996.

[6] L. T. Nguyen, M. Mohamed, H. Park, Y. Pai, R. Wong, A. Qureshi, P. Song, H. D. Truong, and C. Reader, "Multimedia signal processor (MSP) summary," Hot Chips VIII symposium record, pp. 217–226, Aug. 1996.

[7] E. Holmann, T. Yoshida, A. Yamada, and Y. Shimazu, "VLIW processor for multimedia applications," Hot Chips VIII symposium record, pp. 193–202, Aug. 1996.

[8] L. Gwennap, "Digital, MIPS add multimedia extensions," Microprocessor report, pp. 24–28, Nov. 1996.

[9] M. Tremblay, J. M. O'Connor, V. Narayanan, and L. He, "VIS speeds new media processing," IEEE Micro, vol. 16, no. 4, pp. 10–20, Aug. 1996.

[10] R. B. Lee, "Subword parallelism with MAX-2," IEEE Micro, vol. 16, no. 4, pp. 51–59, Aug. 1996.

[11] A. Peleg and U. Weise, "MMX technology extension to the Intel architecture," IEEE Micro, vol. 16, no. 4, pp. 42–50, Aug. 1996.

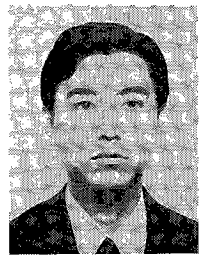
[12] A. K. Jain, R. Preston, P. Bannon, M. Bertone, R. Blake-Campos, G. Bouchard, D. Brasili, D. Carlson, R. Castelino, K. Clark, S. Kobayashi, B. Lilly, S. Mehta, B. Miller, R. Mueller, A. Olesin, and Y. Saito, "1.38 cm^2 550 MHz microprocessor with multimedia extensions," IEEE International Solid-State Circuits Conference, pp. 174–175, Feb. 1997.

[13] S. Mutoh, et al., "1 V high-speed digital circuit technology with 0.5 μm multi-threshold CMOS," Proc. IEEE 1993 ASIC Conf., pp. 186–189, 1993.

[14] T. Kuroda and T. Sakurai, "Low-power circuit design technique for multimedia CMOS VLSI's," IEICE Trans., vol. J80-A, no. 5, pp. 746–752, May 1997.

[15] T. Kuroda, T. Fujita, S. Mita, T. Nagamatsu, S. Yoshioka, K. Suzuki, F. Sano, M. Norishima, M. Murota, M. Kako, M. Kinugawa, M. Kakumu, and T. Sakurai, "A 0.9-V, 150-MHz 10-mW 4 mm^2 2-D discrete cosine transform core processor with variable threshold-voltage (VT) scheme," IEEE JSSC, vol. 31, no. 11, pp. 1770–1779, Nov. 1996.

[16] M. Kakumu, "Process and device technologies of CMOS devices for low-voltage operation," IEICE Trans. Electron., vol. E76-C, no. 5, pp. 672–680, May 1993.



Shunichi Ishiwata received B.E., M.E. and Ph.D degrees in physical electronics engineering from the Tokyo Institute of Technology, Tokyo, Japan in 1987, 1989, 1992, respectively. In 1992, he joined Toshiba Corporation, where he has been working on MPEG2 decoder LSI and media processor development.

Takayasu Sakurai received B.S., M.S. and Ph.D. degrees in EE from the University of Tokyo, Japan, in 1976, 1978, and 1981, respectively. In 1981, he joined Toshiba Corporation, where he designed CMOS DRAM, SRAM and BiCMOS ASIC's. He also worked on interconnect delay and capacitance modeling, and alpha power-law MOS model. From 1988 through 1990, he was a visiting researcher at the Univ. of Calif., Berkeley, doing research in the field of VLSI CAD. From 1990, back in Toshiba, he managed media processor and MPEG2 LSI designs. From 1996, he has been a professor at the Institute of Industrial Science, University of Tokyo, working on low-power and high-performance system LSI designs. Prof. Sakurai served as a program committee member for CICC, DAC, ICCAD, ICVC, ISPLED, ASP-DAC, TAU, CSW, VLSI and FPGA Workshop. He is a technical committee chairperson for the Symp. on VLSI Circuits.