A Single-Chip MPEG-2 Codec Based on Customizable Media Embedded Processor

Shunichi Ishiwata, Tomoo Yamakage, Yoshiro Tsuboi, Takayoshi Shimazawa, Tomoko Kitazawa, Shuji Michinaka, Kunihiko Yahagi, Hideki Takeda, Akihiro Oue, Tomoya Kodama, Nobu Matsumoto, Takayuki Kamei, Mitsuo Saito, Takashi Miyamori, Goichi Ootomo, and Masataka Matsui, *Member, IEEE*

***Abstract*—A single-chip MPEG-2 MP@ML codec, integrating 3.8M gates on a 72-mm^2 die, is described. The codec employs a heterogeneous multiprocessor architecture in which six microprocessors with the same instruction set but different customization execute specific tasks such as video and audio concurrently. The microprocessor, developed for digital media processing, provides various extensions such as a very-long-instruction-word coprocessor, digital signal processor instructions, and hardware engines. Making full use of the extensions and optimizing the architecture of each microprocessor based upon the nature of specific tasks, the chip can execute not only MPEG-2 MP@ML video/audio/system encoding and decoding concurrently, but also MPEG-2 MP@HL decoding in real time.**

***Index Terms*—Audio coding, codecs, microprocessors, motion compensation, MPEG-2, multiprocessing, video signal processing.**

I. Introduction

RECENTLY, many single-chip MPEG-2 MP@ML video encoders and coders/decoders (codecs) have been developed [2]–[7]. Most of these devices have the same architectural characteristics: 1) heavy iteration of relatively simple tasks such as motion estimation, discrete cosine transform (DCT), quantization and variable length encoding is implemented using dedicated hardware engines, and 2) a simple embedded reduced-instruction set (RISC) or digital signal processor (DSP) is used to perform the remainder of the tasks. This organization is very common and provides a suitable balance between cost and performance. As semiconductor technology progresses, single-chip MPEG-2 MP@ML encoders have integrated audio and system [8]–[11] while keeping the above-mentioned architectural characteristics. However, the optimum architecture for video encoding/decoding is different from that for audio and system encoding/decoding. This motivates another architectural characteristic: a heterogeneous multiprocessor architecture, where each processor is optimized for a specific application such as video or audio.

Such application-specific processors are common in many applications. However, recent techniques employed in high-end microprocessors, such as out-of-order speculative execution, branch prediction, and circuit technologies to achieve high clock frequency, are unsuitable for these application-specific processors. On the other hand, the addition of application-specific instructions and/or hardware engines to accelerate heavy iteration of relatively simple tasks often results in a large gain in performance at little additional cost, especially in DSP applications. A heterogeneous multiprocessor architecture using simple and small microprocessors achieves a better cost–performance tradeoff than a single high-end general-purpose microprocessor with hardware engines, because a general-purpose microprocessor is too expensive to do the simple job of coordinating hardware engines.

Based on this understanding, we have developed a customizable media embedded processor architecture [1] suitable for digital media processing. We have also developed hardware/software development tools to support the customization. We have applied the architecture and tools to develop a single-chip MPEG-2 MP@ML codec. It has a heterogeneous multiprocessor architecture in which each processor is optimized based on the nature of its intended application. The proposed chip can execute not only MPEG-2 MP@ML video/audio/system encoding and decoding simultaneously, but also MPEG-2 MP@HL decoding. Several MPEG-2 MP@ML video codecs [4], [7], MPEG-2 MP@ML video/audio/system encoders [8]–[11], and MPEG-2 MP@HL decoders [12]–[15] have already been reported. However, none of these previous circuits are capable of performing all these functions with a single integrated circuit. Recently, an integrated circuit that is capable of performing these functions was developed [16] concurrently with our chip, but the detail has not been disclosed as far as the authors know.

The customizable media embedded processor architecture and hardware/software development tools to support customization are described in Sections II and III, respectively. The details of the MPEG-2 codec architecture and its large-scale integrated (LSI) implementation are described in Sections IV and V, respectively.

II. Customizable Media Embedded Processor Architecture

Fig. 1 shows the proposed media embedded processor architecture. It is customizable and consists of a basic part and an extensible part. The former is described in Section II-A and the latter is described in the subsequent sections.

Manuscript received April 19, 2002; revised October 24, 2002.

S. Ishiwata, Y. Tsuboi, T. Shimazawa, T. Kitazawa, S. Michinaka, K. Yahagi, H. Takeda, A. Oue, N. Matsumoto, T. Miyamori, G. Ootomo, and M. Matsui are with the SoC Research and Development Center, Semiconductor Company, Toshiba Corporation, Kawasaki 212-8520, Japan (e-mail: shunichi.ishiwata@toshiba.co.jp).

T. Yamakage and T. Kodama are with the Multimedia Laboratory, Research and Development Center, Toshiba Corporation, Kawasaki 212-8582, Japan.

T. Kamei and M. Saito are with the Broadband System LSI Project, Semiconductor Company, Toshiba Corporation, Kawasaki 212-8520, Japan.

Digital Object Identifier 10.1109/JSSC.2002.808291

0018-9200/03$17.00 © 2003 IEEE

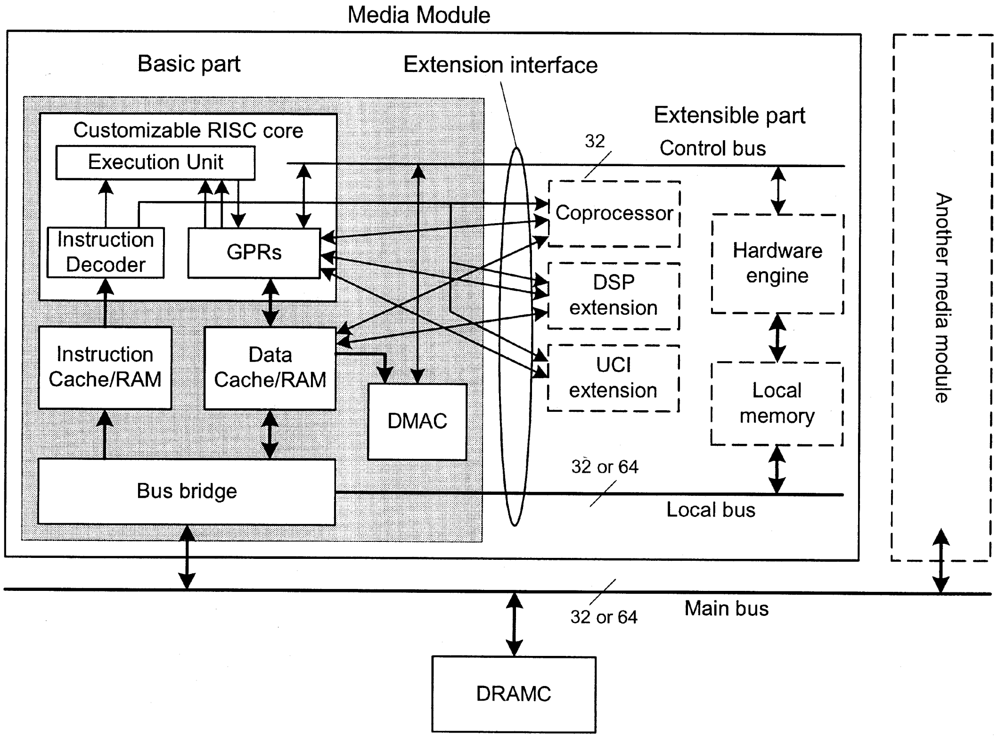


Fig. 1. Proposed media embedded processor architecture.

A. Basic Configuration

The basic configuration of the processor consists of the core instruction set, an instruction cache/RAM, a data cache/RAM, a direct-memory access (DMA) controller, and a bus bridge to a main bus, as shown in Fig. 1. The base processor is a 32-bit five-stage simple RISC core which consists of about 50K gates and can operate at 200 MHz in 0.18-μm CMOS technology. Basic properties as well as customizable extensions of this core are summarized in Table I [17]. The basic configurations include memory size and optional instructions such as multiply and divide. The configurable extensions are depicted in Fig. 1 and described in Sections IV-A–D.

B. VLIW Extension

Currently, the following two very-long-instruction-word (VLIW) extensions are supported:

1) 2-issue VLIW execution unit with a 32-bit instruction length;
2) 3-issue VLIW execution unit with a 64-bit instruction length [1].

In both cases, a core ALU and an extended coprocessor operate in parallel.

In addition, load/store instructions to/from coprocessor registers support an autoincrement with modulo addressing mode, in which the general-purpose registers (GPRs) in the RISC core are used as the address registers.

C. User Custom Instructions

The instruction format of user-custom instructions is limited to two register operands plus a 16-bit sub-opcode or immediate operand. Execution of each of these instruction must be completed within one clock cycle. Branch or load/store instructions cannot be added.

TABLE I
CONFIGURATION AND EXTENSION OF THE MEDIA EMBEDDED PROCESSOR

Basic Core :	
Core function	32-bit 5-stage simple RISC processor 16 GPRs
Instruction format	16-bit / 32-bit variable
Configuration :	
Memory size	Instruction RAM/cache, data RAM/cache
Optional instruction	MAC/mul/div, min/max etc.
Interrupt	Number of ports / levels
Extension :	
VLIW extension	2- / 3- issue, 32-bit /64-bit inst. length, Switch from/to core-mode by call
User custom instruction	Single-cycle combinatorial logic w/ two operands, 16-bit sub-opcode/immediate
DSP extension	Multiple cycle execution w/ two operands, 16-bit sub-opcode/immediate
Hardware engine	Register access mapped onto IO space

D. DSP Extension Instructions

Each DSP extension instruction contains two register operands plus a 16-bit sub-opcode or immediate operand. In these instructions, execution may exceed one cycle. State registers can be implemented. Load/store instructions from/to a data RAM can also be added. For example, a typical DSP

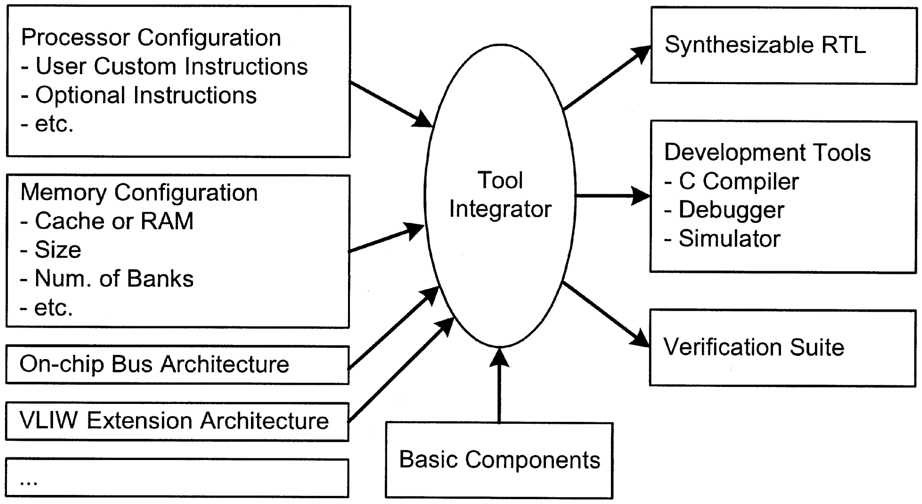


Fig. 2. Tool integrator.

TABLE II
MAIN FUNCTIONS OF THE CHIP

Video encoding format	MPEG-2 MP@ML
	MPEG-4 Simple Profile + B-VOP + interlace, CIF, 30frame/s
Video decoding format	MPEG-2 MP@HL, MP@ML
	MPEG-4 Simple Profile + B-VOP + interlace, CIF, 30frame/s
	MPEG-4 Core Profile, QCIF, 15frame/s
	DV, SD, 4:1:1 format
Audio encoding format	Dolby Digital 2ch,
	MPEG1 Layer II 2ch
Audio decoding format	Dolby Digital 5.1ch
	MPEG1 Layer II 2ch
A/V codec concurrency	2 stream encoding
(MPEG-2 MP@ML)	1 stream enc + 1 stream dec
	4 stream decoding
Motion estimation	Hierarchical multi-field telescopic search
	H : +/-144pixels, V : +/-96pixels
Video input/output format	ITU-R BT.656 (8bit 27MHz)
Audio input/output format	IIS (input : 1port, output : 3ports)
Stream input/output format	Transport stream
	Program stream
	Up to 75Mbps (max 300Mbps)

* Dolby is a trademark of Dolby Laboratories

pipeline might consist of an address calculation using the core ALU followed by load from the data RAM, which is further followed by multiply-accumulation with the extended DSP unit.

E. Hardware Engine and Local Memory

Heavy iteration of a relatively simple task can be implemented as a hardware engine. A C or C++ function is mapped into a hardware engine and its associated code as follows.

1) Arguments and return values which are integers are mapped into specific instructions to read or write registers in a hardware engine through the control bus.
2) Passing large data structure is performed by load/store instructions or using a DMA transfer through the local bus to/from local memory connected to the hardware engine.
3) A function call is performed by starting a hardware engine using a specific instruction to write to a command register in the hardware engine through the control bus.

Both the control bus and the local bus are such that a load/store through the control bus, a DMA transfer over the local bus, and a hardware engine execution can operate in parallel.

F. Heterogeneous Multiprocessor Architecture

Multiple processors can be connected to shared memory using a main bus as shown in Fig. 1. In this context, each processor is referred to as a media module (MM). Each MM contains a basic processor configuration and may also contain a coprocessor, user-custom instructions, DSP extensions, hardware engines, and local memories. The architecture of each MM can be optimized independently based on its target application. Both 32- and 64-bit bus widths are supported.

III. HARDWARE AND SOFTWARE DEVELOPMENT TOOL

We have also developed a tool that generates 1) synthesizable register transfer level (RTL), scripts for synthesis tools, and a verification program, and 2) a software development kit (assembler, compiler, linker, simulator, and debugger). The tool generates these based on a script which describes the architecture extensions (see Fig. 2). This tool can significantly reduce the time required for hardware/software development. Details of this tool are described in [18] and [19].

In practice, extending the architecture also yields the need for modification of the application C source code and addition of a C++ model to describe the function of the extended part. The compiler is extended automatically so that user-custom instructions and DSP extension instructions can be expressed as function calls in the C source code. To support the custom hardware engines, function calls must be replaced by data transfers through the control bus and the local bus. Our tool integrates the C++ model into the core simulator and produces a chip-level simulator. The corresponding RTL is automatically generated from the C++ model by a high-level synthesis tool as long as the extended part is relatively small.

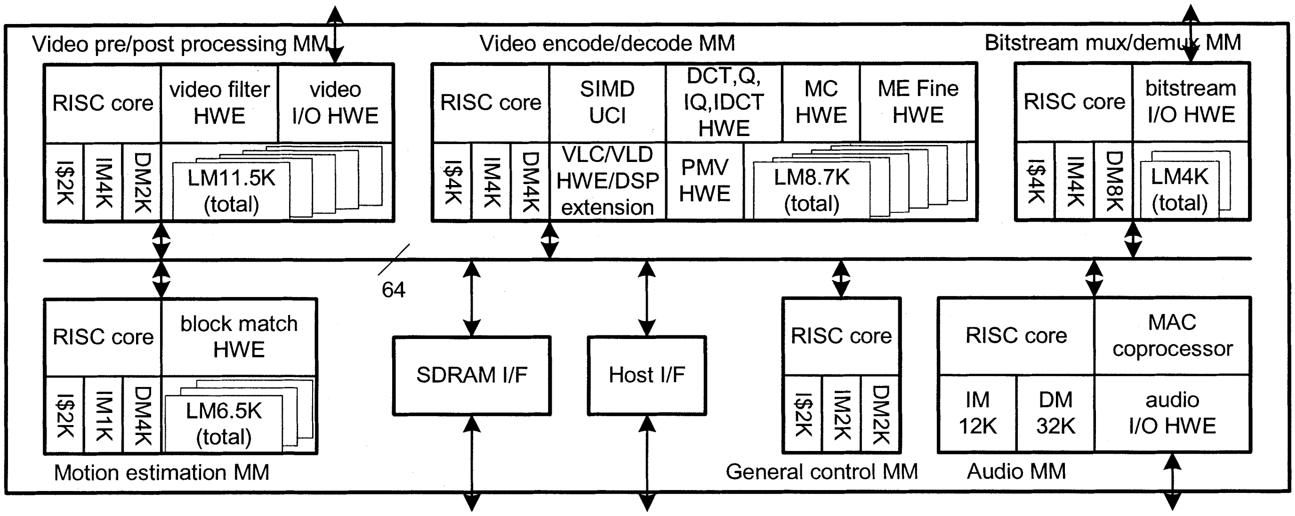


Fig. 3. Block diagram of the MPEG-2 codec LSI.

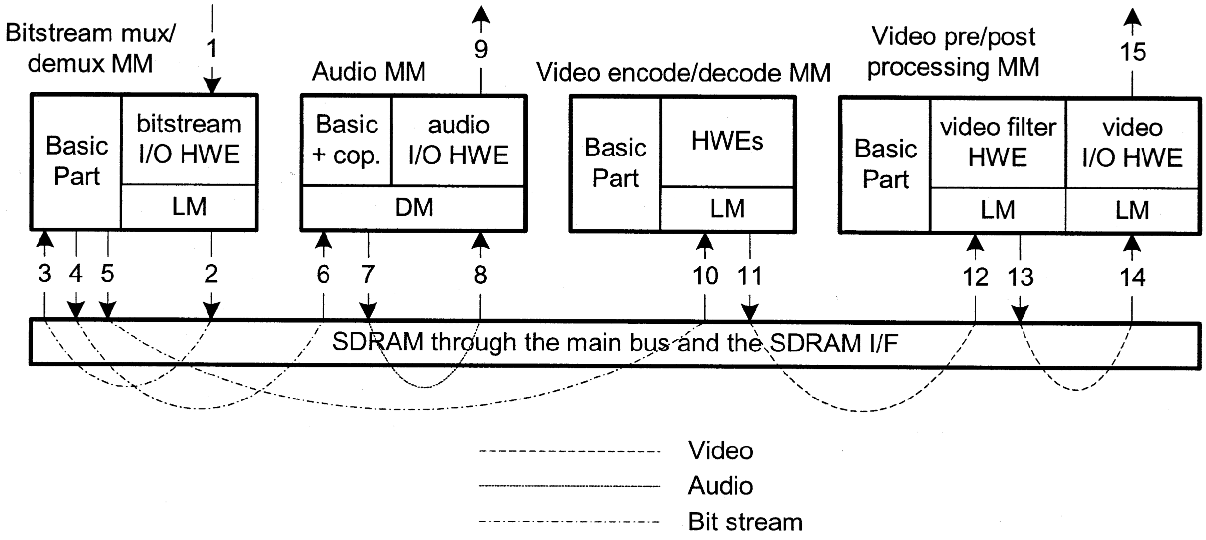


Fig. 4. Outline of data flow in MPEG decoding.

IV. MPEG-2 Codec Architecture

Based on the media embedded processor architecture, a codec integrated circuit was implemented. Table II shows the main functions of the chip.

Fig. 3 shows a block diagram of the chip. It contains six MMs. Each MM is optimized based on the nature of its target application and includes the following extensions:

- bitstream mux/demux MM: bitstream input/output (I/O) hardware engine (HWE);
- audio MM: 2-issue VLIW with a multiply-accumulation coprocessor and an audio I/O HWE;
- video encode/decode MM: five HWEs, a DSP extension for variable length coder (VLC) and variable length decoder (VLD), and user custom instructions for single-instruction multiple-data (SIMD) operations;
- motion estimation (for video compression) MM: block-matching operation HWE;
- video pre/postprocessing MM: video filter HWE and video I/O HWE;
- general control MM: no extension (i.e., basic configuration only).

The fact that the optimized architecture contains such a variety of MMs illustrates the necessity of the proposed extensible media processor. The methodology results in the effective reuse of common components and can be used in other applications such as image recognition [1].

Next, an overview of software implementation is described. As shown in Fig. 4, the MPEG-2 decode task is distributed to each MM as follows.

1) In the bitstream mux/demux MM, a bitstream input to the bitstream I/O HWE is transferred to the SDRAM by an interrupt handler. This bitstream is read from the SDRAM and demultiplexed into an audio elementary stream and a video elementary stream. These streams are written to the SDRAM.

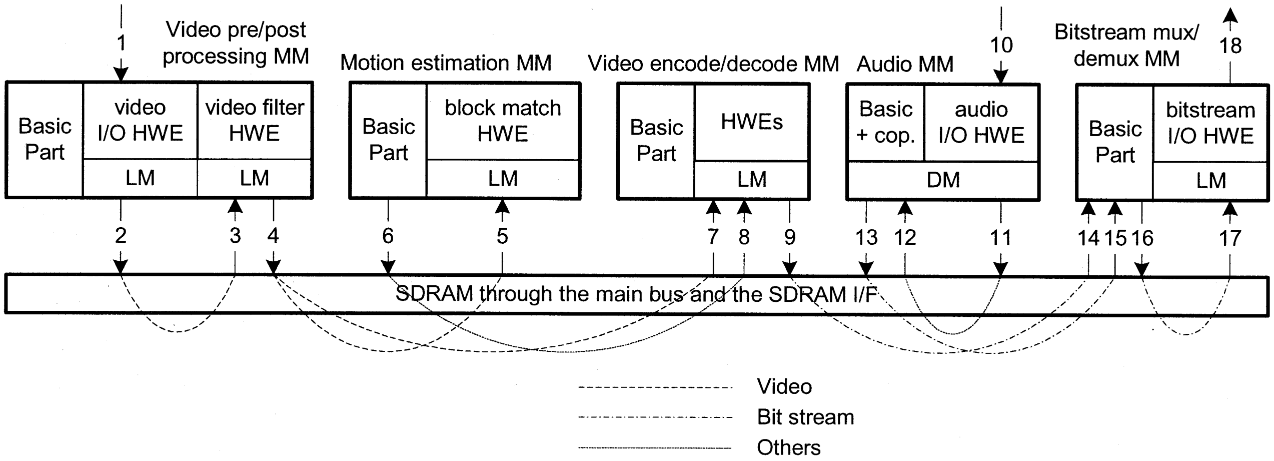


Fig. 5. Outline of data flow in MPEG encoding.

2) In the audio MM, an audio elementary stream is read from the SDRAM and decoded. The result is written to the SDRAM. It is transferred to the audio I/O HWE by an interrupt handler.
3) In the video encode/decode MM, a video elementary stream is read from the SDRAM and decoded. The result is written to the SDRAM.
4) In the video pre/postprocessing MM, the decoded video is read from the SDRAM and postprocessed. The result is written to the SDRAM. It is transferred to the video I/O HWE by an interrupt handler.

The MPEG-2 encode operation is the reverse behavior of decode operation except for motion estimation. An outline of data flow, shown in Fig. 5, is as follows.

1) In the video pre/postprocessing MM, a video source input to the video I/O HWE is transferred to the SDRAM by an interrupt handler. It is read from the SDRAM, preprocessed, and written to the SDRAM.
2) In the motion estimation MM, coarse-grain motion estimation is performed using the preprocessed video and the result is written to the SDRAM.
3) In the video encode/decode MM, the preprocessed video is encoded using the above result and the generated video elementary stream is written to the SDRAM.
4) In the audio MM, an audio source input to the audio I/O HWE is transferred to the SDRAM by an interrupt handler. Then it is encoded and the generated audio elementary stream is written to the SDRAM.
5) In the bitstream mux/demux MM, the video and audio elementary stream is read from the SDRAM and multiplexed into a program or transport stream. The result is written to the SDRAM. Finally, it is transferred to the bitstream I/O HWE by an interrupt handler.

It is noted that data transfer between MMs is executed via the SDRAM. This implementation is essential in that it alleviates the difficulties in scheduling the multitasking operations. It is also indispensable for reducing the peak data transfer rate between the MMs, which justifies the single shared-bus architecture. The bus arbitration algorithm is fair roundrobin so that the worst case latency is easily estimated. Our method of data transfer between MMs causes simultaneous access to shared resources in the SDRAM. To resolve this problem, hardware semaphore registers are used.

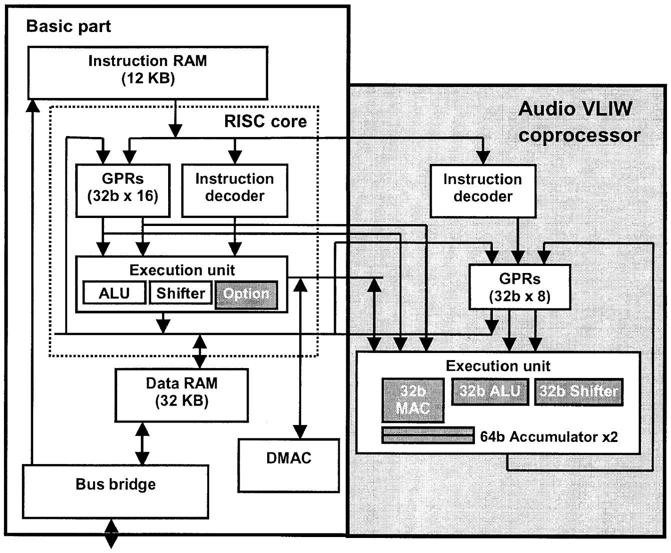


Fig. 6. VLIW extension of audio MM.

On each MM, multiple tasks such as encode, decode, and I/O tasks can run pseudosimultaneously in an interrupt-driven and time-division-multiplex manner. A kernel for each MM to switch tasks is also customized based on the nature of the target application.

Several notable implementation details involving extensions of the media embedded processor are described in Sections IV-A–D.

A. DMA Controller

A DMA controller in each MM performs the following two functions.

1) Transferring a rectangular region between memory regions using a single command. This is often required in video applications and is usually implemented in MPEG-2 encoders. The implementation is similar to the one in [20].
2) Descriptor chain mode to chain some DMA transfers. Descriptor tables are stored in a data RAM by software and

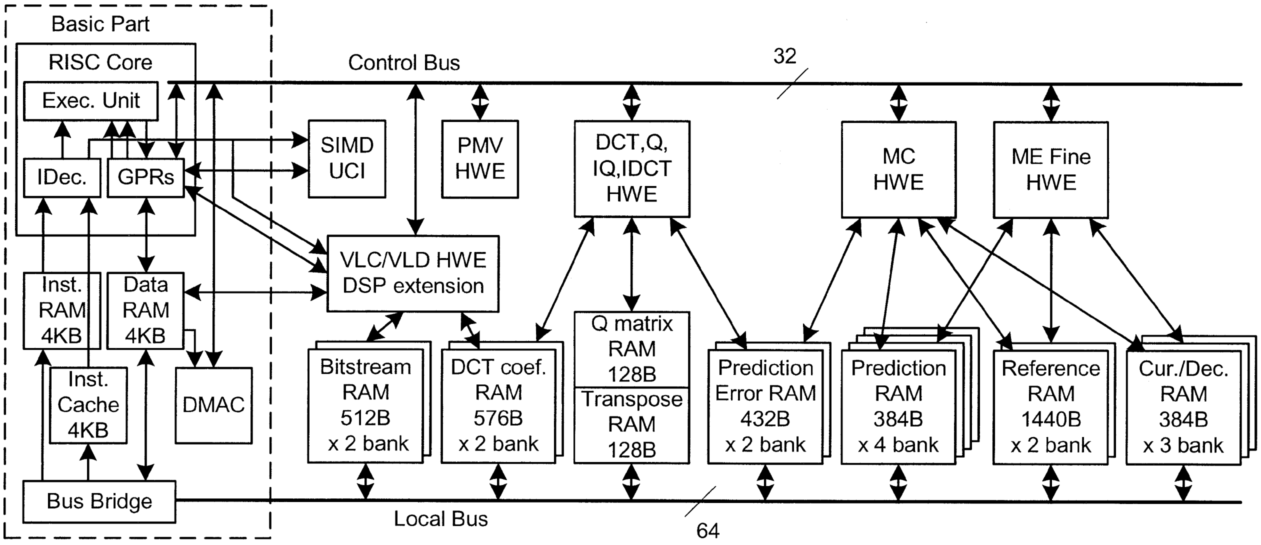


Fig. 7. Block diagram of video encode/decode MM.

are read by a DMA controller. This is indicated by the solid line from the data RAM to the DMAC in Fig. 1. The descriptor chain mode reduces the frequency of interactions between hardware and software and makes software pipelining less complicated. This is especially important in the case of a shared-bus multiprocessor architecture, because it is very difficult to estimate when a DMA transfer finishes.

B. Audio MM

Fig. 6 shows the structure of the VLIW extension in the audio MM. It has the following features:

1) eight 32-bit GPRs;
2) two 64-bit accumulators;
3) 32-bit ALU, shifter, and multiply-accumulator (MAC) supporting the following instructions:
   a) add, subtract, logical, shift, and funnel shift;
   b) multiply, multiply–add, and multiply–subtract;
   c) leading zero detect, absolute difference, MIN/MAX, and clipping.

The processor core has two processing modes: core mode and VLIW mode. The core mode issues a 16- or 32-bit instruction each cycle. In the VLIW mode, a 32-bit fixed length instruction is executed. The mode is changed using a subroutine call and return instructions. Each VLIW instruction consists of a 32-bit core instruction, a 32-bit coprocessor instruction, or a combination of a 16-bit core instruction and coprocessor instruction. For instance, a 16-bit load instruction with address increment and a coprocessor multiply–addition instruction can be executed simultaneously. The GPRs in the core can be used as input data of the 32-bit MAC in the coprocessor. Furthermore, the leading zero detection, absolute difference, MIN/MAX, and clipping instructions are added to the processor core as optional instructions. The customization results in about a threefold improvement in performance over the basic processor core for fast Fourier transform (FFT) and filter operations, which appears in audio decoding and encoding.

C. Video Encode/Decode MM

The block diagram of the video encode/decode MM is shown in Fig. 7. In MPEG-2 video encoding/decoding, most of the time-consuming operations are executed by the hardware engines, while the firmware's jobs are mainly to address calculations for DMA transfers and parameter preparation for the hardware engines. Some of the local memories are double buffered so that DMA transfers and all hardware engines can operate in parallel. This parallel operation is accomplished by a software pipeline technique. The unit of pipelining is the macroblock level.

The VLC and VLD functions utilize the DSP extension with a resource-sharing technique for sharing data between a DSP extension and a hardware engine. The VLD performs with the following two functions:

- as a DSP extension, with instructions to decode a symbol of a fixed/variable length code;
- as a hardware engine, with a command to decode a macroblock.

The use of the single-symbol decode instruction described above leads to an efficient implementation of the macroblock-decode command, since it allows the use of resources such as 1) local memory to store bitstream; 2) arithmetic unit to extract bits; and 3) control logic to refill a temporary bit buffer from the local memory.

An example of the use of these two functions is as follows. The macroblock-decode command stores macroblock header parameters such as the macroblock type and motion vectors in the data RAM. As an exception, the macroblock address increment [22], which is the first variable length code in a macroblock header, is decoded by a DSP instruction because the software should not issue macroblock-decode commands for skipped macroblocks while it controls the rest of the pipeline. In addition, providing both the above DSP extension and the hardware engine makes it possible to support both 1) MPEG-2 MP@HL which requires high performance but little flexibility, and 2) other standards, such as JPEG, DV, and MPEG-4, which demand moderate performance but much flexibility.

Another feature that supports other standards is that hardware engines have not only macroblock commands but also single function commands such as Q, 8 × 8 DCT, and 2 × 4 × 8 IDCT. The single function commands share resources with the macroblock commands and make the hardware slightly more complex. As an example, the quantization method in JPEG [23] is somewhat different from that in MPEG-2 and, therefore, a macroblock command for MPEG-2 is inappropriate. In this case, the single function commands are still useful. Another example is 2 × 4 × 8 IDCT for DV decoding [24]. In DV decoding, the firmware's jobs of unpacking, variable length decoding, and inverse quantization are shared with the motion estimation MM which is not otherwise used in decoding. The same technique is useful for superposition of text and video object layers in MPEG-4 shape decoding [25]. Such a flexible task assignment to a MM is one of the advantages of our architecture compared with conventional dedicated architectures.

Several user-custom instructions for SIMD operations are added to the RISC core. They are parallel add, subtract, shift, set less than, logical operations, special instructions for encoding motion vectors, and byte shuffle. They are especially useful in calculating addresses of reference pictures and encoding motion vectors, which are dominant tasks of firmware in MPEG encoding. In addition, a predictive motion vectors (PMV) hardware engine is added to accelerate decoding motion vectors in MPEG-2 MP@HL decoding. The user-custom instructions are insufficient to achieve this level of performance.

The fine-grain motion estimation (ME Fine) hardware engine has the following functions:

- motion estimation with half pixel precision;
- mode selection (Candidates are {forward or backward or bidirectional} {frame or field} motion compensation (MC), dual-prime MC, no MC, and intra. Some modes are excluded depending on a picture coding type and a progressive frame flag.);
- forming the best predicted picture corresponding to the selected mode and storing it in the prediction RAM;
- calculating macroblock activity which is used for rate control.

With the exception of the dual-prime MC option, all functions can be executed by a single command. If the dual-prime MC option is required, the command is split into two; for the dual-prime MC option, the first command is used to calculate an address, while the second command uses this address to transfer an opposite parity field.

The ME Fine hardware engine is separate from the motion estimation MM since fine-grain motion estimation requires not only block-matching operations but also other functions such as half-pixel interpolation and averaging forward and backward reference picture.

In summary, real-time MPEG-2 encoding/decoding, MPEG-4 encoding/decoding, and DV decoding in Table II are achieved by utilizing the following extensions:

- MPEG-2 encode: DSP extension, DCT, Q, IQ, IDCT HWE, MC HWE, ME Fine HWE, SIMD UCI, VLC/VLD HWE;
- MPEG-2 decode: DSP extension, DCT, Q, IQ, IDCT HWE, MC HWE, PMV HWE, VLC/VLD HWE;
- MPEG-4 encode: DSP extension, DCT, Q, IQ, IDCT HWE, MC HWE, ME Fine HWE, SIMD UCI;
- MPEG-4 decode: DSP extension, DCT, Q, IQ, IDCT HWE, MC HWE;
- DV decode: DSP extension, DCT, Q, IQ, IDCT HWE, MC HWE.

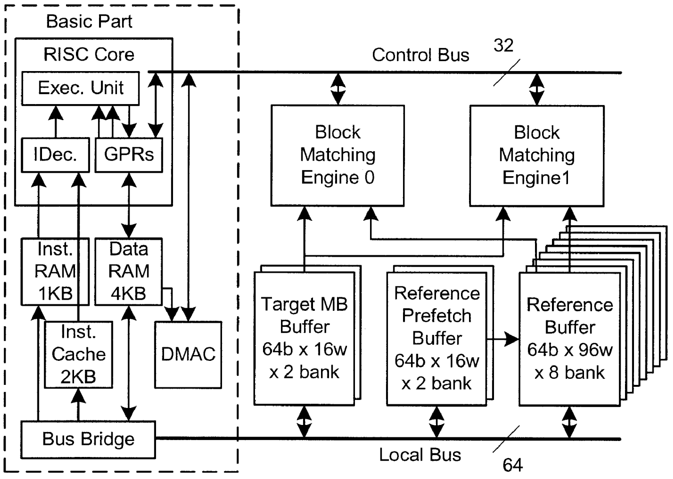


Fig. 8. Block diagram of motion estimation MM.

The firmware's main jobs other than address calculation for DMA transfer and parameter preparation for the hardware engines are:

MPEG-2 encode: rate control;
MPEG-2 decode: nothing;
MPEG-4 encode: VLC, DCT DC/AC coefficient prediction, rate control;
MPEG-4 decode: VLD, DCT DC/AC coefficient prediction, shape decoding;
DV decode: unpack, VLD, IQ.

D. Motion Estimation MM

The block diagram of the motion estimation MM is shown in Fig. 8. The block-matching operation is dominant in MPEG-2 encoding and is implemented as a hardware engine. It consists of a pair of block-matching engines, a reference buffer, a reference prefetch buffer, and a target macroblock buffer. The latter two are double buffered so that the block-matching engines and DMA transfer can operate in parallel. The block-matching engines have the following features.

- Each block-matching engine has ability to perform an 8 × 8-pixel block-matching operation with one cycle throughput, which is attained by utilizing a pipelined SIMD structure. Efficient use of these engines achieves wide search range up to horizontal ±144 pixels and vertical ±96 pixels with a hierarchical multifield telescopic search algorithm [21] modified to reduce the required memory bandwidth to a third of the original value (using telescopic search algorithm causes an increase in the required memory bandwidth because the search location for each macroblock is irregular and it increases the amount of reference reading. The modified algorithm makes

TABLE III
SPECIFICATIONS OF THE CHIP

Technology	0.18um CMOS 6 –level metal
Number of logic gates	3.8M gates (including SRAMs)
Die size	8.5mm x 8.5mm (72.25mm^2)
Package	TBGA 352 pin
Power supply	3.3V (I/O), 1.5V (internal)
Clock frequency	150MHz (system clock)
Power consumption	1.5W

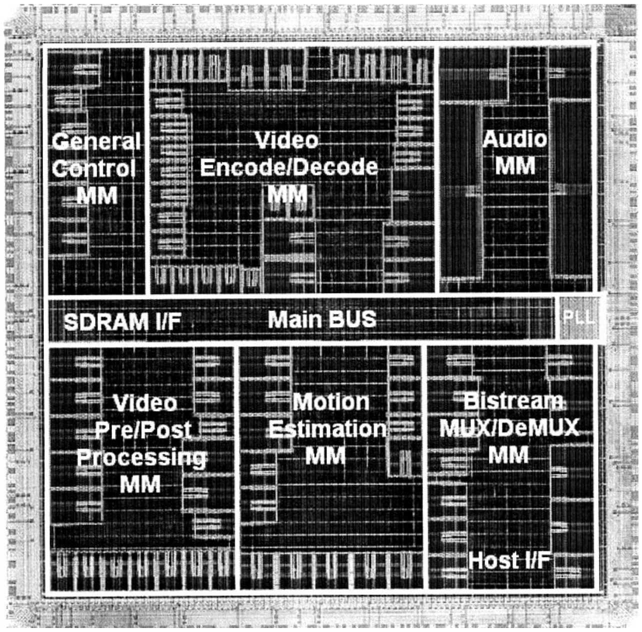


Fig. 9. Microphotograph of the chip.

the search location regular at the cost of more frequent current reading). It is worth noting that the architecture is fairly general and allows use of a wide variety of motion estimation algorithms.

- The shape of the search range can be flexibly set, which makes it possible to widen the search range when the input video is rapidly panning.
- The difference between pixel dc (or average) value of the current picture and that of the reference picture is compensated during the block-matching operation. It provides efficient motion estimation even if the input video contains a fade-in/out pattern.

V. LSI IMPLEMENTATION

Table III shows the chip specification. It is fabricated using 0.18-μm six-level metal CMOS technology, integrating 3.8M logic gates in the 72-mm^2 die. It operates at 150 MHz and consumes 1.5 W to achieve the main functions such as the real-time codec operations listed in Table II. Fig. 9 shows a chip microphotograph and Fig. 10 shows a photograph of the reference system board.

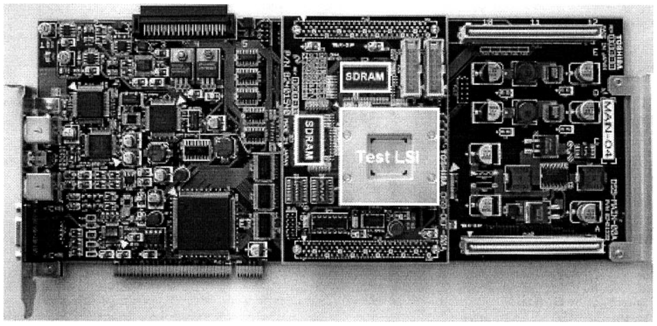


Fig. 10. Photograph of the evaluation board.

VI. CONCLUSION

This paper has presented an customizable embedded processor architecture and hardware/software development tools to support the customization. This architecture enables the quick development of an application-optimized heterogeneous multiprocessor. (By "application," we are referring to media processing applications such as video, audio, bitstream mux/demux, graphics, and image processing.) Based on this customizable architecture, an MPEG-2 MP@ML codec has been successfully developed. Making full use of customization, the chip can execute in real time: 1) MPEG-2 MP@ML video/audio/system simultaneous encoding and decoding; 2) MPEG-2 MP@HL decoding; 3) DV decoding; 4) MPEG-4 core profile plus interlace encoding and decoding; etc. We expect our approach to become more advantageous in the future as the design complexity and the demand for systems-on-chip increase.

ACKNOWLEDGMENT

The authors wish to thank Dr. K. Suzuki, A. Yokosawa, Y. Uetani, S. Koto, and N. Kogure for research on algorithm and firmware implementation, and the media embedded processor team, the software development tool team, and the design verification team for their cooperation and great efforts.

REFERENCES

[1] Y. Kondo, T. Miyamori, T. Kitazawa, S. Inoue, H. Takano, I. Katayama, K. Yahagi, A. Ooue, T. Tamai, K. Kohno, Y. Asao, H. Fujimura, H. Uetani, Y. Inoue, S. Asano, Y. Miyamoto, A. Yamaga, Y. Masubuchi, and T. Furuyama, "A 4-GOPS 3-way VLIW image recognition processor based on a configurable media-processor," in *IEEE Int. Solid-State Circuits Conf. (ISSCC 2001) Dig. Tech. Papers*, Feb. 2001, pp. 148–149.

[2] A. van der Werf, F. Brüls, R. P. Kleihorst, E. Waterlander, M. J. W. Verstraelen, and T. Friedrich, "I.McIC: A single-chip MPEG-2 video encoder for storage," *IEEE J. Solid-State Circuits*, vol. 32, pp. 1817–1823, Nov. 1997.

[3] M. Mizuno, Y. Ooi, N. Hayashi, J. Goto, M. Hozumi, K. Furuta, A. Shibayama, Y. Nakazawa, O. Ohnishi, S. Zhu, Y. Yokoyama, Y. Katayama, H. Takano, N. Miki, Y. Senda, I. Tamitani, and M. Yamashina, "A 1.5-W single-chip MPEG-2 MP@ML video encoder with low-power motion estimation and clocking," *IEEE J. Solid-State Circuits*, vol. 32, pp. 1807–1816, Nov. 1997.

[4] E. Iwata, K. Seno, M. Aikawa, M. Ohki, H. Yoshikawa, Y. Fukuzawa, H. Hanaki, K. Nishibori, Y. Kondo, H. Takamuki, T. Nagai, K. Hasegawa, H. Okuda, I. Kumata, M. Soneda, S. Iwase, and T. Yamazaki, "A 2.2GOPS video DSP with 2-RISC MIMD, 6-PE SIMD architecture for real-time MPEG2 video coding/decoding," in *IEEE Int. Solid-State Circuits Conf. (ISSCC 1997) Dig. Tech. Papers*, vol. 32, Feb. 1997, pp. 258–259.

[5] E. Miyagoshi, T. Araki, T. Sayama, A. Ohtani, T. Minemaru, K. Okamoto, H. Kodama, T. Morishige, A. Watabe, K. Aoki, T. Mitsumori, H. Imanishi, T. Jinbo, Y. Tanaka, M. Taniyama, T. Shingou, T. Fukumoto, H. Morimoto, and K. Aono, "A 100 mm^2 0.95 W single-chip MPEG2 MP@ML video encoder with a 128-GOPS motion estimator and a multi-tasking RISC-type controller," in *IEEE Int. Solid-State Circuits Conf. (ISSCC 1998) Dig. Tech. Papers*, Feb. 1998, pp. 30–31.

[6] E. Ogura, M. Takashima, D. Hiranaka, T. Ishikawa, Y. Yanagita, S. Suzuki, T. Fukuda, and T. Ishii, "A 1.2-W single-chip MPEG2 MP@ML video encoder LSI including wide search range (H:±288, V:±96) motion estimation and 81-MOPS controller," *IEEE J. Solid-State Circuits*, vol. 33, pp. 1765–1771, Nov. 1998.

[7] Y. Tsuboi, H. Arai, M. Takahashi, and M. Oku, "A low-power single-chip MPEG-2 CODEC LSI," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, May 1999, pp. 73–76.

[8] G. Kizhepat, K. Choy, R. Hinchley, P. Lowe, and R. Yip, "A single-chip MPEG-2 video audio and system encoder," in *IEEE Int. Solid-State Circuits Conf. (ISSCC 1999) Dig. Tech. Papers*, Feb. 1999, pp. 266–267.

[9] S. Kumaki, T. Matsumura, K. Ishihara, H. Segawa, K. Kawamoto, H. Ohira, T. Shimada, H. Sato, T. Hattori, T. Wada, H. Honma, T. Watanabe, H. Sato, K.-I. Asano, and T. Yoshida, "A single-chip MPEG2 422@ML video, audio, and system encoder with a 162-MHz media-processor and dual motion estimation cores," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, May 1999, pp. 95–98.

[10] W. H. A. Bruls, E. W. Salomons, A. van der Werf, R. K. Gunnewiek, and L. Camiciotti, "A low-cost audio/video single-chip MPEG2 encoder for consumer video storage applications," in *IEEE Int. Conf. Consumer Electronics (ICCE) Dig. Tech. Papers*, June 2000, pp. 314–315.

[11] S. Kumaki, H. Takata, Y. Ajioka, T. Ooishi, K. Ishihara, A. Hanami, T. Tsuji, Y. Kanehira, T. Watanabe, C. Morishima, T. Yoshizawa, H. Sato, S.-I. Hattori, A. Koshio, K. Tsukamoto, and T. Matsumura, "A 99 mm^2, 0.7-W single-chip MPEG2 422P@ML video, audio, system encoder with a 64-Mbit embedded DRAM for portable 422P@HL encoder system," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, May 2001, pp. 425–428.

[12] H. Kim, W.-S. Yang, M.-C. Shin, S.-J. Min, S.-O. Bae, and I.-C. Park, "Multithread VLIW processor architecture for HDTV decoding," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, May 2000, pp. 559–562.

[13] T. Koyama, E. Iwata, H. Yoshikawa, H. Hanaki, K. Hasegawa, M. Aoki, M. Yasue, T. Schronbenhauser, M. Aikawa, I. Kumata, and H. Koyanagi, "A 250 MHz single-chip multiprocessor for A/V signal processing," in *IEEE Int. Solid-State Circuits Conf. (ISSCC 2001) Dig. Tech. Papers*, Feb. 2001, pp. 146–147.

[14] Y. Watanabe, Y. Otobe, K. Yoshitomi, H. Takahashi, and K. Kohiyama, "A single-chip MPEG2 MP@HL decoder for DTV recording/playback system," in *IEEE Int. Conf. Consumer Electronics (ICCE) Dig. Tech. Papers*, June 2001, pp. 90–91.

[15] H. Yamauchi, S. Okada, K. Taketa, Y. Matsuda, T. Mori, S. Okada, T. Watanabe, Y. Harada, M. Matsudaira, and Y. Matsushita, "A 0.8-W HDTV video processor with simultaneous decoding of two MPEG2 MP@HL streams and capable of 30 frame/s reverse playback," in *IEEE Int. Solid-State Circuits Conf. (ISSCC 2002) Dig. Tech. Papers*, Feb. 2002, pp. 372–373.

[16] LSI Logic. [Online]. Available: http://www.lsilogic.com

[17] T. Miyamori, "A configurable and extensible media processor," presented at the Embedded Processor Forum 2002, Apr. 2002.

[18] A. Mizuno, K. Kohno, R. Ohyama, T. Tokuyoshi, H. Uetani, H. Eichel, T. Miyamori, N. Matsumoto, and M. Matsui, "Design methodology and system for a configurable media embedded processor extensible to VLIW architecture," in *Proc. Int. Conf. Computer Design (ICCD 2002)*, Sept. 2002, pp. 2–7.

[19] SoC Design Methodology With MeP. [Online]. Available: http://www.mepcore.com/english/index_e.html

[20] K. Okamoto, T. Jinbo, T. Araki, Y. Iizuka, H. Nakajima, M. Takahata, H. Inoue, S. Kurohmaru, T. Yonezawa, and K. Aono, "A DSP for DCT-based and wavelet-based video codecs for consumer applications," *IEEE J. Solid-State Circuits*, vol. 32, pp. 460–467, Mar. 1997.

[21] Toshiba Science and Technology Highlights 1998. Single-chip MPEG2 encoder LSI. [Online]. Available: http://www.toshiba.co.jp/tech/review/1998/high98/research/r3/index.htm

[22] *Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Video*, ISO/IEC 13818-2, ITU-T recommendation H.262, May 1996.

[23] *Information Technology—Digital Compression and Coding of Continuous-Tone Still Images—Requirements and Guidelines*, ISO/IEC 10918-1, ITU-T recommendation T.81, 1992.

[24] *Helical-Scan Digital Video Cassette Recording System Using 6.35-mm Magnetic Tape for Consumer Use (525-60, 625-50, 1125-60, and 1250-50 Systems)*, IEC 61834, 1994.

[25] *Information Technology—Generic Coding of Audio-Visual Objects—Part 2: Visual*, ISO/IEC 14 496-2, 1999.

Shunichi Ishiwata received the B.E., M.E., and Ph.D. degrees in physical electronics engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 1987, 1989, and 1992, respectively.

In 1992, he joined Toshiba Corporation, Kawasaki, Japan, where he has been working on the development of MPEG2 decoder LSIs, media processors, and MPEG2 codec LSIs. He is currently a Senior Specialist in the SoC Research and Development Center and is working on MPEG video encoder algorithm, architecture, and software development.

Tomoo Yamakage received the B.E. degree in information engineering and the M.E. degree in electronic engineering from Tohoku University, Sendai, Japan, in 1988 and 1990, respectively.

In 1990, he joined Toshiba Corporation, Kawasaki, Japan, where he has been working on MPEG-2 standardization and the development of MPEG-2 LSIs. He is currently a Research Scientist in the Corporate Research and Development Center and is working on the development of high-definition DVD and video watermarking technology.

Yoshiro Tsuboi received the Ph.D. degree in theoretical physics from the University of Tsukuba, Ibaraki, Japan, in 1989.

He joined the ULSI Research Center, Toshiba Corporation, Kawasaki, Japan, in 1991, where he was engaged in development of high-speed silicon bipolar devices. In 1995, he changed his field of activity to research and development of system LSI design, and is currently dedicated to system LSI firmware design and development in the SoC Research and Development Center.

Takayoshi Shimazawa was born in Saitama, Japan, on April 27, 1965. He received the B.S. and M.S. degrees in electric engineering from Keio University, Yokohama, Japan, in 1989 and 1991, respectively.

In 1991, he joined the Semiconductor Device Engineering Laboratory, Toshiba Corporation, Kawasaki, Japan, where he was engaged in the research and development of advanced system LSIs, including MPEG2 video decoders and MPEG2 video and audio codecs. Currently, he is involved in the design of system LSIs based on a customizable media embedded processor in the SoC Research and Development Center.

Tomoko Kitazawa received the B.S. degree in physics from the University of Tsukuba, Ibaraki, Japan, in 1983.

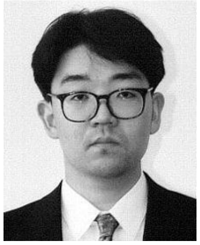
She joined Toshiba Corporation, Kawasaki, Japan, in 1983, where she was engaged in research and development of amorphous-silicon TFT-LCDs. In recent years, she has been working on the development of media processors and MPEG2 codec LSIs. She is currently a Specialist with the SoC Research and Development Center.

Shuji Michinaka received the B.S. and M.S. degrees from Hokkaido University, Sapporo, Japan, in 1990 and 1992, respectively.

In 1992, he joined Toshiba Corporation, Kawasaki, Japan, where he has been working on the development of MPEG2 decoder LSIs.

Kunihiko Yahagi received the B.S. degree in physics from the University of Tokyo, Tokyo, Japan, in 1993.

In 1993, he joined Toshiba Corporation, Kawasaki, Japan, where he has been working on the development of MPEG2 decoder LSIs, media processors, and MPEG2 codec LSIs. He is currently working on the development of memory subsystem for a MPEG codec LSI with the SoC Research and Development Center.



Hideki Takeda received the B.S. degree in electrical engineering and the M.S. degree in electronic engineering from the University of Tokyo, Tokyo, Japan, in 1992 and 1994, respectively.

In 1994, he joined Toshiba Corporation, Kawasaki, Japan, where he has been engaged in the development of MPEG2 decoder LSIs and media processors. He has also been working on the design methodology for those LSIs. He is currently involved in the development of MPEG2 codec LSIs.

Akihiro Oue received the B.E. and M.E. degrees in electrical and electronic engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 1996 and 1998, respectively.

In 1998, he joined Toshiba Corporation, Kawasaki, Japan, where he has been working on the development of MPEG2 codec LSIs. He is currently a Software Engineer with the SoC Research and Development Center.

Tomoya Kodama received the B.E. and M.E. degrees in electronic engineering from Tohoku University, Sendai, Japan, in 1991 and 1993, respectively.

In 1993, he joined Toshiba Corporation, Kawasaki, Japan, where he has been working on the development of MPEG2 software and LSIs. He is currently a Research Scientist with the Corporate Research and Development Center and is working on MPEG video CODEC algorithm and software development.

Nobu Matsumoto received the B.E. and M.E. degrees in electrical engineering from Waseda University, Tokyo, Japan, in 1981 and 1983, respectively.

In 1983, he joined Toshiba Corporation, Kawasaki, Japan, where he has been engaged in the research and development of EDA systems and software development environment.

Takayuki Kamei was born in Tokyo, Japan, in 1972. He received the B.E. degree in electrical engineering and the M.S. degree in computer science from Keio University, Yokohama, Japan, in 1995 and 1997, respectively.

He is currently a Processor Development Engineer with the Broadband System-LSI Project, Toshiba Corporation Semiconductor Company, Kawasaki, Japan.

Mitsuo Saito received the M.S. degree in electronic engineering from the University of Tokyo, Tokyo, Japan, in 1974.

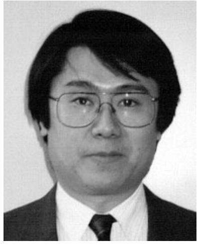
He joined the Research and Development Center, Toshiba Corporation, Kawasaki, Japan, in 1974. His focus has been the incorporation of Japanese (including Kanji) characters into computer systems, and desktop publishing. From 1984 to 1985, he studied at the Massachusetts Institute of Technology Media Laboratory and performed research on three-dimensional (3-D) graphics and human interface. In 1986, after returning to Japan, he started a 3-D graphics chip development project. The successor of this chip was adopted for use in the Sony Playstation. He also started a RISC processor development project which later became a joint project with Silicon Graphics, Inc. Beginning in 1994, he was involved in a server development project and designed a super-fault-resiliant system. Since 1997, he has been a Director of the System ULSI Engineering Laboratory, Toshiba, where he has lead the project team developing the processor for Sony's Playstation2, a media processor project and a communication chip set project. In April 2000, his organization became the System LSI Research and Development Center, by combining the LSI design and the device technology development team. In July 2001, he became Chief Fellow, and now leads the Broadband System LSI Project.

Takashi Miyamori received the B.S. and M.S. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1985 and 1987, respectively.

Since joining Toshiba Corporation, Kawasaki, Japan, in 1987, he has performed research and development of microprocessors. From 1996 to 1998, he was a Visiting Researcher at Stanford University, Stanford, CA, where he researched reconfigurable computer architectures. He is currently a Senior Specialist in the SoC Research and Development Center, Toshiba.

Goichi Ootomo received the B.S. and M.S. degrees in Electronic Engineering from Ibaraki University, Ibaraki, Japan, in 1987 and 1989, respectively.

In 1989, he joined Toshiba Corporation, Kawasaki, Japan, where he was involved in the research and development of the digital signal processors. Since 1992, he has been involved in the development of the MPEG2 video decoder/encoder LSIs.



Masataka Matsui (S'83–M'85) received the B.S. and M.S. degrees in electronic engineering from the University of Tokyo, Tokyo, Japan, in 1983 and 1985, respectively.

In 1985, he joined the Semiconductor Device Engineering Laboratory, Toshiba Corporation, Kawasaki, Japan, where he was engaged in the research and development of static memories including 1-Mb CMOS SRAM and 1-Mb BiCMOS SRAM, BiCMOS ASICs, video compression/decompression LSIs, media processors, and embedded microprocessors. From 1993 through 1995, he was a Visiting Scholar at Stanford University, Stanford, CA, doing research on low-power LSI design. He is currently managing digital media system-on-chip (SoC) development at the SoC Research and Development Center, Toshiba. His current research interests include configurable embedded processor design, low-power and high-speed circuit design, and video compression/decompression SoC design. He is also a Visiting Lecturer at the University of Tokyo.

Mr. Matsui serves as a Program Committee Member for the Symposium on VLSI Circuits and the IEEE Custom Integrated Circuits Conference.