

Low-Power Implementation of H.324 Audiovisual Codec Dedicated to Mobile Computing

Takao Onoye, Gen Fujita, Hiroyuki Okuhata, Morgan H. Miki, and Isao Shirakawa

Dept. Information Systems Engineering
Osaka University
Suita, Osaka, 565-0871 Japan

Phone: +81(6)879-7807

Fax: +81(6)875-5902

e-mail: {onoe, fujita, okuhata, miki, sirakawa}@ise.eng.osaka-u.ac.jp

Abstract—A VLSI implementation of the H.324 audiovisual codec is described. A number of sophisticated low-power architectures have been devised dedicatedly for the mobile use. A set of specific functional units, each corresponding to a process of H.263 video codec, is employed to lighten different performance bottlenecks. A compact DSP core composed of two MAC units is used for both ACELP and MP-MLQ coding schemes of the G.723.1 speech codec. The proposed audiovisual codec core has been implemented by using $0.35\mu\text{m}$ CMOS 4LM technology, which contains totally 420k transistors with the dissipation of 224.32mW from single 3.3V supply.

I. INTRODUCTION

The H.324[1] international standard specifies the low bitrate audiovisual communication based on PSTN (Public Switched Telephone Network), which is mainly composed of the H.263[2] video coding algorithm and the G.723.1[3] speech coding algorithm. Actually, by means of H.324 the QCIF (176×144) sized pictures at 10 fps (frames/sec) together with the 8kHz 16-bit linear PCM digital voices can be coded at 28.8kbps. Thus various applications of this H.324 standard are to be realized extensively in mobile computing, wireless multimedia communication, etc. In particular, portable multimedia facilities in the wireless environment can be regarded as the enormous potentialities of multimedia communications.

This audiovisual codec may be implemented with the use of those multimedia enhanced DSPs[4, 5, 6, 7, 8, 9], which have been developed specifically for MPEG1/2. In this case, however, the power consumption can hardly be held down any more, since to raise the performance any of these DSPs should operate at such a high frequency that the H.324 codec core would be forced to dissipate 500mW or more. Moreover, in terms of the system-on-chip implementation, these audiovisual codec facilities should be integrated in a single chip together with other functionalities, such as communication controller[10], special-purpose memory[11], and CMOS image sensor[12]. Thus a more radical innovation should be pursued for the reduction of not only the area occupancy but also the power dissipa-

tion, and hence there arises the technical issue of how to develop the H.324 architectures dedicatedly for the mobile use.

The present paper describes a number of VLSI architectures for implementing the H.324 audiovisual codec. The main feature of this codec consists in the reduction of power consumption such that the operation frequency can be lowered to 15MHz. In what follows, the main subject is focussed on how to lower the power consumption for both speech and video codecs. A part of VLSI implementation results are also shown, which are attained with the use of an ASIC design system *COMPASS Design Tools*.

II. COMPACT DSP CORE FOR G.723.1

The G.723.1 supports two different bitrates i.e. 5.3kbps and 6.3kbps. The Algebraic Code Excited Line Prediction (ACELP) is used as the excitation signal for the lower rate (5.3kbps) coder, and the Multipulse Maximum Likelihood Quantization (MP-MLQ) for the higher rate (6.3kbps) coder. The input speech signal is sampled by the 16-bit linear PCM at 8kHz, and a sequence of 240 samples constitutes a unit of *frame*, which corresponds to a period of 30 msec.

Fig. 1 shows an outline of G.723.1 codec process. A digital voice is encoded through means of parameters; gain, pitch, prediction coefficient, and non-periodic component. A gain and a set of non-periodic components are generated by ACELP/MP-MLQ, a pitch by Pitch Estimator and Pitch Predictor, and a set of prediction coefficients by LPC (Linear Predictive Coding) and LSP (Line Spectrum Pair). In the decoding process, the prediction coefficients constitute a synthesis filter by which the pitch and non-periodic components are filtered, and the gain is multiplied by the output signal of the filter so as to restore the original speech signal.

A. Computational Cost Analysis

Since these two different coders are selectively used for generating bitstreams, a DSP should be necessarily employed for the G.723.1 codec. To attain an optimal set of microarchitectures, the computational complexity of G.723.1 codec is analyzed by software simulation. As a result, Table I shows the number of clock cycles necessary

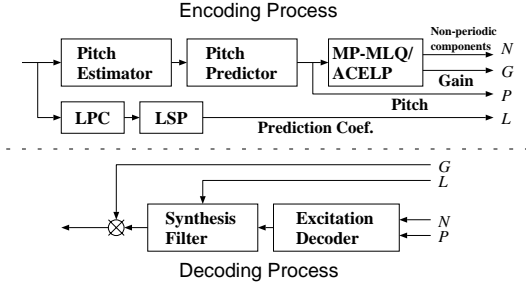


Fig. 1. Outline of G.723.1 codec process.

TABLE I
NUMBER OF CLOCK CYCLES AT EACH PROCESS (CYCLE/FRAME).

process		bitrate	
		5.3 kbps	6.3 kbps
Enc.	IPC	53,280	53,280
	ISP	41,284	41,288
	Pitch Est.	14,600	13,063
	Pitch Pred.	55,006	42,748
	MPMLQ	—	39,225
	ACELP	13,449	—
	Gain	14,337	15,648
Total		1,094,197	1,407,255
Dec.	Upsp. k	10,884	10,933
	Syn filter	57,232	57,467
	Gain	14,412	14,339
	Total	82,518	82,739

for executing each codec process with the use of a 16-bit general purpose DSP.

As can be readily seen from Table I, the encoding process needs 1.0 and 1.4 million cycles per frame at 5.3 and 6.3 kbps, respectively, and the decoding process needs 80 thousand cycles per frame at each of these bitrates. Since one frame is composed of 240 samples with the sampling rate of 8kHz, the whole codec process per frame has to be completed within 30 msec so that a decoding speech signal should not be broken. Consequently, in order to realize a realtime codec with the use of this conventional 16 bit DSP, a clock frequency needs

$$(1,407,255 + 82,518) / (30/1000) = 49.66 \text{ MHz} \quad (1)$$

in case of the bitrate of 6.3kbps.

Seeing that the computational labor of Pitch Predictor and MP-MLQ amounts to almost 70% of the total encoding process, the detailed analysis has been executed mainly for these processes. As a result, it turns out that the multiply accumulation (MAC) operation frequently appears in the deepest loop of these two processes, and moreover that there are two kinds of MACs which can be classified according to whether or not there needs a shift operation after the multiplication. Generally, audio data are treated as fixed point decimal data, and if operands of MAC are of the fixed point type, the result of multiplication should be shifted by 1 bit to left. In contrast with this, if operands are of the integer type, there is no shift operation after the multiplication. Thus in the G.723.1 codec process, there are only two kinds of MAC operation. Paying attention to this feature, henceforth an architecture of DSP is discussed.

B. Microarchitecture

Since MAC operations, decimal or integer, appear frequently both in Pitch Predictor and in MP-MLQ, our primary subject is how to execute these operations in a limited number of clock cycles. A general DSP has an MAC unit which is composed of a multiply block and an accumulate block. In order to improve the performance of such a MAC unit, the following two mechanisms may be adopted:

- A: to insert pipeline registers to improve the clock frequency[13],
- B: to construct a number of MAC units to be run in parallel[14].

Mechanism A improves the processing ability, but increases the chip size because of the added pipeline registers. On the other hand, Mechanism B enlarges the chip size, but not the clock frequency. It should be added that Mechanism A requires a higher clock frequency than Mechanism B, and that Mechanism B increases the chip area much more than Mechanism A. For our purpose of low-power consumption, it is a major priority to reduce the clock frequency, and hence Mechanism B is to be employed here.

Fig. 2 shows a structure of our MAC scheme in the proposed DSP. Two parallel MAC units are used in order to reduce the the total number of instruction cycles. The required clock frequency is not so high for two MAC units to be pipelined, and hence these two units are connected in parallel so that two MAC operations can be executed within 1 clock cycle. The multiplexer can select the 1 bit left shift according to whether or not MAC operations are fixed decimal, and hence this mechanism can achieve two MAC operations in 1 clock cycle.

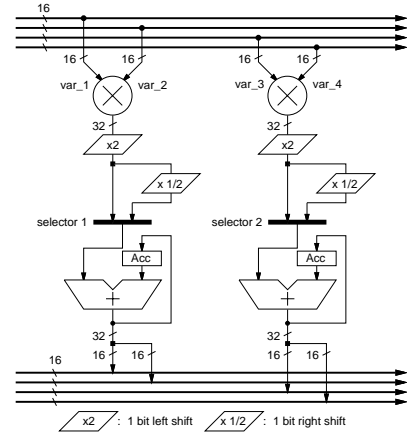


Fig. 2. Block diagram of MAC units.

The proposed DSP has a 16-bit ALU and a barrel shifter unit in addition to these MAC units. The 16-bit ALU can execute add/sub, logic operation, negate, and absolute operation in a clock cycle. Each of 32-bit precision operations is executed in two clock cycles. The barrel shifter can shift the 16-bit data ranging from 1 to 16 bits in both of the left and right directions. Each of the ALU and the shifter has the facility of saturation against overflow, which can reduce the number of comparison and branch instructions

The on-chip memory is composed of two-banked RAM for one-frame work memory, IROM (Instruction ROM) for instructions, and TROM (Table ROM) for filter coefficient table. According to our G.723.1 software simulation, it is decided that the sizes of each RAM

and TROM are 2K and 5K words, respectively. Since the one-frame codec process can be executed without an access to the external memory, not only the total processing speed is improved, but also the bandwidth to the external memory is reduced.

Fig. 3 shows the whole structure of the proposed DSP. In order to operate two MAC units in parallel, a 64-bit internal data bus is employed. This internal bus is divided into two 32-bit buses, say, bus A and bus B. The MAC units are connected to both of these buses, the ALU and the shifter are connected to bus A. The switching multiplexer satisfies the connection requests between RAMs and buses.

A 13-bit address port and a 16-bit data port are provided as Input/Output ports. The I/O port bandwidth is 128kbps for the output of generated PCM, and is either 6.3kbps or 5.3kbps for the input of a bitstream. Accordingly, the total bandwidth requires about 130kbps.

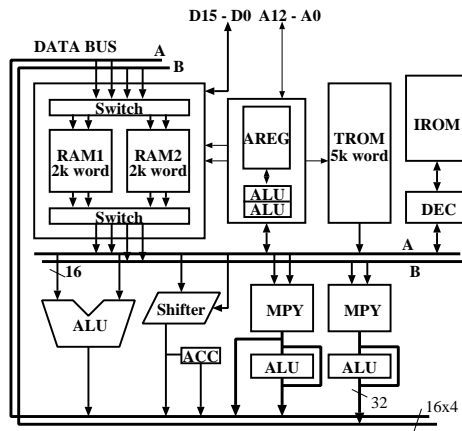


Fig. 3. Block diagram of DSP.

In order to realize the G.723.1 encoding process with the use of the proposed DSP, 57,702 cycles are necessary in Pitch Predictor, 67,251 cycles in MP-MLQ, and 250,295 cycles in the total encoding process. On the other hand, the decoding process needs 26,327 cycles. Consequently, in order to realize a realtime G.723.1 codec by the proposed DSP, the necessary clock frequency is given by

$$(250,995 + 26,327) / (30/1000) = 9.23 \text{ MHz}. \quad (2)$$

III. H.263 VIDEO CODEC CORE

The main encoding/decoding process of H.263 is executed by means of the so-called MC-DCT coding in the same manner as H.261 and MPEG1. The distinctive features of the H.263 consist in simple syntax, half-pel prediction, block-level motion estimation (advanced prediction mode), paired coding of the P-frame and the B-frame (i.e. the PB-frame mode), motion detection for the outside of frame (i.e. the unrestricted vector mode), SAC (syntax-based arithmetic coding mode), and so on.

The picture coding can be achieved with the use of several functional units; Motion Estimator (ME), Discrete Cosine Transformer (DCT), Quantizer (Q), and Variable Length Coder (VLC)/Syntax-based Arithmetic Coder (SAC). The bitstream decoding can be performed with the use of another set of functional units; Variable Length Decoder (VLD)/Syntax-based Arithmetic Decoder (SAD), Inverse

Quantizer (IQ), Inverse Discrete Cosine Transformer (IDCT), and Motion Compensator (MC).

A. Organization

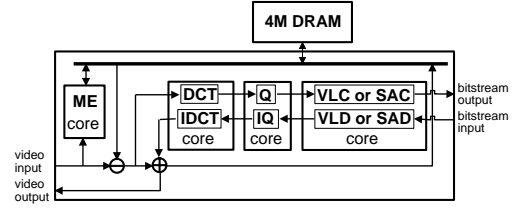


Fig. 4. Organization of H.263 codec core.

The overall organization of our video codec core is summarized in Fig. 4. The main factor to achieve a high throughput at a low operation frequency can be attributed to the mechanism that the I/O and processing conflicts can be mitigated at each stage of the codec. The detailed architecture of each functional unit is outlined in what follows.

B. ME Core

As to the so-called block-matching algorithm for the ME core, a number of authors[15, 16, 17, 18] have attempted to reduce the computational labor of the *full-search*, which is to detect a motion vector exhaustively within a search range by reference to MADs (Mean Absolute Differences). However, there still remain defects in vector quality[15, 17] as well as in VLSI implementation capability[16, 18]. Recently, a *macroblock clustering* algorithm has been proposed[19], which has the following distinctive features;

- 1) high quality vectors,
- 2) low computational costs, and
- 3) VLSI implementation capability.

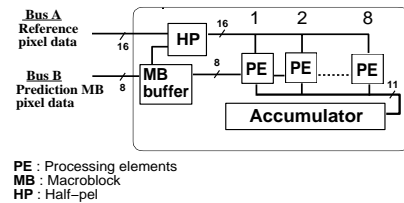


Fig. 5. Organization of ME core.

The organization of the ME core is illustrated in Fig. 5, which consists of a one-dimensional PE (Processing Element) array, an accumulator for calculating macroblock vectors and block vectors, a macroblock buffer for the bi-directional prediction, and a half-pel calculator.

Furthermore, this ME core successfully treats the following H.263 coding options which can not be achieved so far by the conventional ME core[20].

[Advanced Prediction (AP) Mode] In addition to the normal macroblock prediction, the advanced prediction mode detects motions of four blocks in a macroblock. In other words, as outlined in Fig. 6, a macroblock can have either one macroblock vector or four block vectors. To deal with this, an accumulator is devised, in which the

MADs for macroblock and four blocks are calculated simultaneously by accumulating the MADs of 8 pixels output from the PEs.

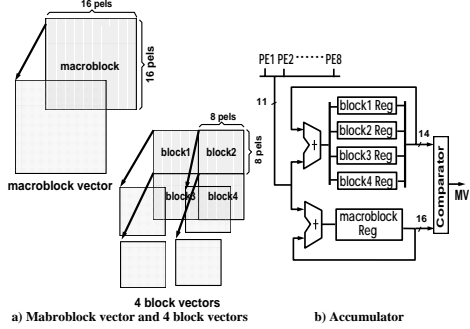


Fig. 6. Accumulator architecture for AP mode.

[PB-frame Mode] The H.263 supports the PB-frame mode so that two frames (P-frame and B-frame) can be coded as one unit, and the ME core seeks the vectors for a pair of macroblocks of these two frames simultaneously. Apart from MPEG, the motion estimation for the B-frame of H.263 requires the concurrent reference of two frames, since only one vector per macroblock is used for the bi-directional prediction as illustrated in Fig. 7. Therefore, the half-pel calculator determines the average of forward and backward reference pixel data, and feeds them to the PE array. The former reference pixel data are read from the external memory, and the latter from the macroblock buffer.

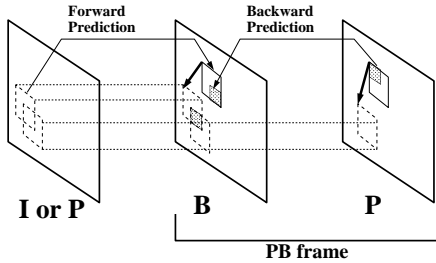


Fig. 7. Bi-directional prediction of H.263.

C. DCT/IDCT Core

The computational labor of H.263 codec is lower than that of MPEG1/2 core. The DCT/IDCT architectures developed for MPEG1/2[21, 22], should not be employed for this H.263 due to the hardware cost, and therefore in what follows a novel specific architecture is devised.

For implementing DCT/IDCT the Chen's algorithm (butterfly computation) is widely used in conjunction with a distributed arithmetic. This algorithm can reduce the number of multiplications in DCT/IDCT by half. Specifically the 8×1 DCT and 8×1 IDCT are calculated by means of the following equations,

$$\begin{bmatrix} X_0 \\ X_2 \\ X_4 \\ X_6 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} A & A & A & A \\ B & C & -C & -B \\ A & -A & -A & A \\ C & -B & B & -C \end{bmatrix} \begin{bmatrix} x_0 + x_7 \\ x_1 + x_6 \\ x_2 + x_5 \\ x_3 + x_4 \end{bmatrix}, \quad (3)$$

$$\begin{bmatrix} X_1 \\ X_3 \\ X_5 \\ X_7 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} D & E & F & G \\ E & -G & -D & -F \\ F & -D & G & E \\ G & -F & E & -D \end{bmatrix} \begin{bmatrix} x_0 - x_7 \\ x_1 - x_6 \\ x_2 - x_5 \\ x_3 - x_4 \end{bmatrix}, \quad (4)$$

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} A & B & A & C \\ A & C & -A & -B \\ A & -C & -A & B \\ A & -B & A & -C \end{bmatrix} \begin{bmatrix} X_0 \\ X_2 \\ X_4 \\ X_6 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} D & E & F & G \\ E & -G & -D & -F \\ F & -D & G & E \\ G & -F & E & -D \end{bmatrix} \begin{bmatrix} X_1 \\ X_3 \\ X_5 \\ X_7 \end{bmatrix}, \quad (5)$$

$$\begin{bmatrix} x_7 \\ x_6 \\ x_5 \\ x_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} A & B & A & C \\ A & C & -A & -B \\ A & -C & -A & B \\ A & -B & A & -C \end{bmatrix} \begin{bmatrix} X_0 \\ X_2 \\ X_4 \\ X_6 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} D & E & F & G \\ E & -G & -D & -F \\ F & -D & G & E \\ G & -F & E & -D \end{bmatrix} \begin{bmatrix} X_1 \\ X_3 \\ X_5 \\ X_7 \end{bmatrix}, \quad (6)$$

$$A = \cos \frac{\pi}{4}, B = \cos \frac{\pi}{8}, C = \sin \frac{\pi}{8}, D = \cos \frac{3\pi}{8}, E = \cos \frac{5\pi}{8}, F = \sin \frac{3\pi}{8}, G = \sin \frac{5\pi}{8}.$$

According to the distributed arithmetic, as illustrated in Fig. 8, each multiply accumulation can be executed by using accumulators and those ROMs which contain tables of products calculated in advance. However, this scheme requires additionally *bit slicer* and *reorder buffer*, and these units as well as ROMs occupy a considerable area in the H.263. That is, to implement H.263 DCT/IDCT core, this overhead must be a serious obstacle.

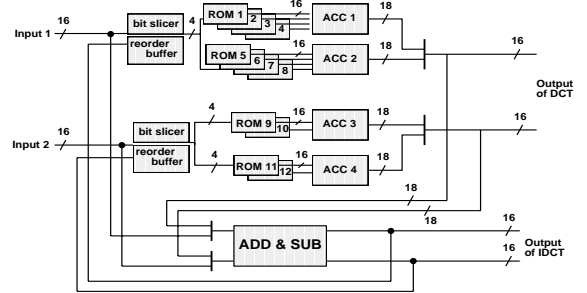


Fig. 8. DCT/IDCT core by distributed arithmetic.

On the contrary, as illustrated in Fig. 9, our DCT/IDCT is devised dedicatedly for H.263, where ROMs **ROM A** ~ **ROM G** and accumulators **ACC 1** ~ **ACC 4** calculate each multiplication of equations (3)–(6) (i.e. $A(x_0 + x_7)$, $A(x_1 + x_6)$, ..., AX_0 , BX_2 , ..., etc.), 4 bits at a time. As a result, the *bit slicer* and the *reorder buffer* can be removed, and the number of ROMs can be reduced.

Table II summarizes the comparison among proposed architecture, the distributed arithmetic architecture, and the 16-bit MAC unit. The proposed architecture reduces the number of gates without degrading the performance.

D. Q/I Core

The operations of Q and IQ are simply divisions and multiplications. To reduce the area occupancy, our Q is implemented by means of a 2-bit sequential non-restoring divider, and IQ by a radix-4 sequential Booth's multiplier without parallel/array facilities, as illustrated in Fig. 10. Both of these Q and IQ can output the division/multiplication result at every 4 cycles.

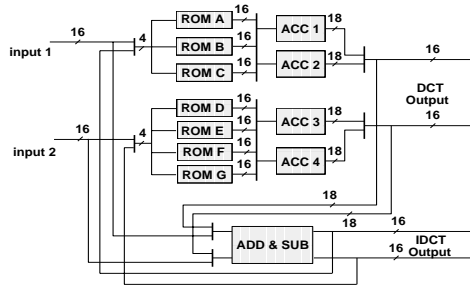


Fig. 9. Block diagram of proposed DCT/IDCT core.

TABLE II
COMPARISON OF DCT/IDCT CORE ARCHITECTURE.

Architecture	Direct	DA	MAC
Gates	3,707	4,118	4,784
Cycles/Block	IDCT	38	40
	IDCT	48	52

E. VLC/VLD and SAC/SAD Core

In addition to VLC, the H.263 standard supports SAC, which is based on arithmetic operations and table index search. In order to achieve high efficiency, either of those two encoding modes is chosen picture by picture.

Fig. 11 shows a block diagram of the coding core. In both of two coding modes, a set of *run*, *level*, and *last* is indicated by an index, and then the index is coded to a bitstream. Thus, it turns out that the *index generator* can be shared by different coding modes. As for the VLC table, the compressing mechanism proposed by [23] is employed to reduce the table size. The *arithmetic unit* calculates arithmetic operations required for SAC and SAD.

F. Pipeline Configuration

Now that we have outlined all functional units, there remains an issue of how to control the encoding/decoding pipeline. Fig. 12 shows the encoding/decoding pipeline for each block. In terms of the throughput, this core can process blocks every 1,280 cycles. As for the processing delay, encoding and decoding of the bitstream for one block are done in 2,096 cycles and 1,280 cycles, respectively.

IV. IMPLEMENTATION RESULTS

A set of these sophisticated architectures for a low-power audiovisual codec is implemented through the use of ASIC design system

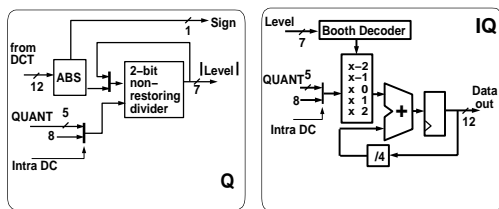


Fig. 10. Block diagram of Q/IQ core.

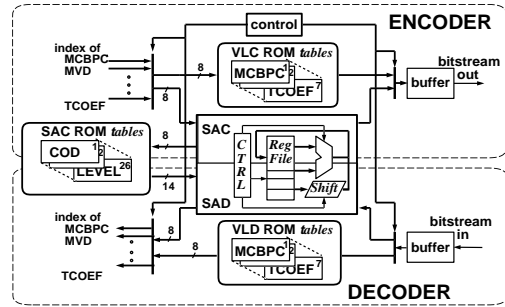


Fig. 11. Block diagram of VLC/VLD and SAC/SAD core.

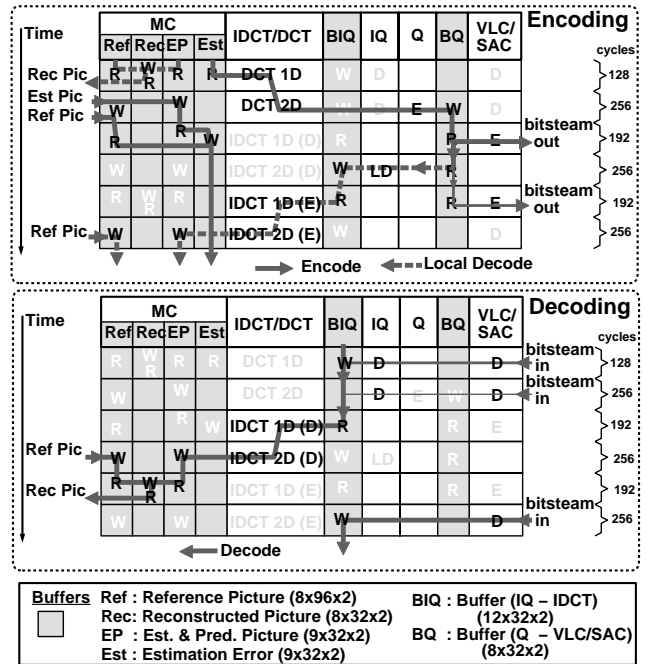


Fig. 12. Encoding/decoding pipeline configuration.

COMPASS Design Tools. Table III indicates the main chip features of the H.324 audiovisual codec core, and Fig. 13 shows the layout patterns obtained by a 0.35 μ m CMOS 4LM technology. It should be added that the operation frequency can be lowered to 15MHz to reduce the total power dissipation to 224.32mW, and hence the core can be of mobile use. Beside this architectural technique, other device-level low-power techniques can attain much more power reduction.

Table IV shows the required memory size and memory bandwidth for audio and video codec. It can be readily seen that the external memory can be constructed of a 4Mbit EDO-DRAM to be connected via 16-bit bus.

V. CONCLUSION

This paper has discussed the low-power implementation of H.324 audiovisual codec, which is dedicated to the mobile computing. A number of low-power microarchitectures have been introduced into the design of low bitrate video and speech codec. Specifically, a set of specific functional units reduces the hardware amount without degrad-

TABLE III
MAIN FEATURES OF THE CODEC CORE.

Technology	0.35 μm CMOS 4LM
Chip size	4.65 mm \times 3.27 mm
Transistors	420,898
Clock frequency	15MHz
Power dissipation	224.32mW (3.3V, 15.0MHz)
Video I/O	QCIF, sub-QCIF 10fps
Speech I/O	16bit linear PCM 8KHz
Video options	AP mode, Bframes, SA C/SAD
Speech coding	ACELP, MPMLQ



Fig. 13. Final layout Patterns (4.65 \times 3.27 mm²).

ing the performance of the video codec. A compact DSP core, which has two MACs, effectively executes two different coding schemes of speech codec.

The main feature of this codec consists in the reduction of not only chip area but also power consumption. The proposed audiovisual codec core has been implemented by using 0.35 μm CMOS 4LM technology, which occupies 15.21 mm² and enables realtime codec with the dissipation of 224.32mW from single 3.3V supply. Since this H.324 codec core should be integrated into wireless multimedia communication system, development is continuing on the low-power implementation of other components, such as communication controller, special purpose memory, and CMOS sensor.

REFERENCES

- [1] ITU-T Rec. H.324: "Terminal for low bitrate multimedia communication," Draft International Standard, Nov. 1995.

TABLE IV
MEMORY SIZE AND MEMORY BANDWIDTH.

	I/O		Size	Bandwidth
			(bits)	(bps)
Enc.	I	Camera in	608,256	3,041,280
	O	MC	6,400	121,702,400
	O	MC	608,256	16,536,960
	I/O	Bitstream	131,072	64,000
Local Dec.	I/O	MC	0	16,536,960
Dec.	I/O	Bitstream	131,072	64,000
	I/O	MC	912,334	16,536,960
	O	Video out	0	3,041,280
Audio	I/O		537,200	537,200
Total			2,934,690	178,061,040

- [2] ITU-T Rec. H.263: "Video coding for low bitrate communication," Draft International Standard, May 1996.
- [3] ITU-T Rec. G.723.1: "Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s," International Standard, Mar. 1996.
- [4] J. Golston: "Single-chip H.324 videoconferencing", *IEEE Micro*, 16, 4, pp. 21–33, Aug. 1996.
- [5] D. Brinthaup, J. Knoblock, J. Othmer, B. Petryna and M. Uyttendaele: "A programmable audio/video processor for H.320, H.324 and MPEG", *IEEE ISSCC Digest of Tech. Papers*, pp. 244–245, Feb. 1996.
- [6] G. A. Slavenburg, S. Rathnam, and H. Dijkstra: "The Trimedia TM-1 PCI VLIW media processor", *Hot Chips VIII*, pp. 171–177, Aug. 1996.
- [7] P. Kalapathy: "Hardware/software interaction on the Mpaact media processor", *Hot Chips VIII*, pp. 179–191, Aug. 1996.
- [8] E. Holmann, T. Yoshida, A. Yamada, and Y. Shimazu: "VLIW processor for multimedia applications", *Hot Chips VIII*, pp. 193–202, Aug. 1996.
- [9] K. Okamoto, et al.: "A DSP for DCT-based and wavelet-based video CODEC's for consumer applications", in *IEEE Journal of Solid-State Circuits*, pp. 460–467, Mar. 1997.
- [10] H. Okuhata, H. Uno, K. Kumatani, I. Shirakawa, and T. Chiba: "A 4Mbps infrared wireless link dedicated to mobile computing", in *Proc. IEEE Int'l Performance, Computing, and Communica. Conf.*, pp. 463–467, Feb. 1997.
- [11] T. Sumi: "Ferroelectric nonvolatile memory technology", *IEICE Trans. Electronics*, vol. E79-C, no. 6, pp. 812–818, June 1996.
- [12] E.R. Fossum: "CMOS image sensors: electronic camera on a chip", in *IEEE Int'l Electron Devices Meeting Tech. Digest*, pp. 1.3.1–1.3.8, Dec. 1995.
- [13] H. Kabuo et al.: "An 80-MOPS-peak high-speed and low-power-consumption 16-b digital signal processor", *IEEE Journal of Solid-State Circuits*, vol. 31, no. 4, pp. 494–503, Apr. 1996.
- [14] T. Shiraishi, et al.: "A 1.8V 36mW DSP for the half-rate speech CODEC", in *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 371–374, May 1996.
- [15] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro: "Motion-compensated interframe coding for video conferencing," in *Proc. National Telecommunication Conference*, pp. G.5.3.1–G.5.3.5, Nov. 1981.
- [16] H. Tominaga, N. Komatsh, T. Miyashita, and T. Hanamura: "A motion detection method on video image by using hierarchical pixels", *IEICE Trans. Inf. & Syst.*, vol. J72-D-II, no.3, pp.395–403, March 1989.
- [17] M. C. Chen and A. N. Willson Jr.: "A high accuracy predictive logarithmic motion estimation algorithm for video coding", in *Proc. IEEE Int'l Symp. Circuits and Systems*, pp.617–620, May 1995.
- [18] Y. Kim, C.S. Rim, and B. Min: "A block matching algorithm with 16:1 subsampling and its hardware design", in *Proc. IEEE Int'l Symp. Circuits and Systems*, pp.613–616, May 1995.
- [19] G. Fujita, T. Onoye, and I. Shirakawa: "A new motion estimation core dedicated to H.263 video coding", in *Proc. IEEE Int'l Symp. Circuits and Systems*, pp. 1161–1164, Jun. 1997.
- [20] H. Lin, A. Anesko, and B. Petryna: "A 14-Gops programmable motion estimator for H.26X video coding", *IEEE J. Solid-State Circuits*, vol.31, no.11, pp.1742–1750, Nov. 1996.
- [21] S. Uramoto, et al.: "A 100MHz 2-D discrete cosine transform core processor", *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp. 492–499, Apr. 1992.
- [22] T. Masaki, Y. Morimoto, T. Onoye, and I. Shirakawa: "VLSI implementation of inverse discrete cosine transformer and motion compensator for MPEG2 HDTV video decoding", *IEEE Trans. Circuits and Systems for Video Technology*, vol. 5, no. 5, pp. 387–395, Oct. 1995.
- [23] E. Tanaka and T. Enomoto: "70mW variable length codec for MPEG2", in *Proc. IEICE General Conference*, C-577, Mar. 1995.