

A Single-Chip Video/Audio CODEC for Low Bit Rate Application

Seongmo Park^{a)}, Seongmin Kim, Igkyun Kim, Kyungjin Byun, Jin Jong Cha, and Hanjin Cho

In this paper, we present a design of video and audio single chip encoder/decoder for portable multimedia application. The single-chip called as video audio signal processor (VASP) consists of a video signal processing block and an audio signal processing block. This chip has mixed hardware/software architecture to combine performance and flexibility. We designed the chip by partitioning between video and audio block. The video signal processing block was designed to implement hardwired solution of pixel input/output, full pixel motion estimation, half pixel motion estimation, discrete cosine transform, quantization, run length coding, host interface, and 16 bits RISC type internal controller. The audio signal processing block is implemented with software solution using a 16 bits fixed point DSP. This chip contains 142,300 gates, 22 kbits FIFO, 107 kbits SRAM, and 556 kbits ROM, and the chip size is 9.02 mm×9.06 mm which is fabricated using 0.5 micron 3-layer metal CMOS technology.

I. INTRODUCTION

Recently the market for portable multimedia applications, such as MPEG video camera, wireless videophone, and portable wireless multimedia terminal, has been on the rise. Video/audio processors are essential to make this multimedia system high performance, low cost and low power consumption. These processors can be widely used to implement multimedia application such as joint photographic expert group (JPEG), H.261, H.263 [1], MPEG, G.723, H.324, and other data compress standards. A general multimedia video/audio system is shown in Fig. 1, which is composed of camera, decoder, VASP ASIC, graphic overlay, display, memory, microphone, speaker, audio subsystem, CPU, IO interface, display, and public switched telecommunication networks (PSTN). The VASP is the main hardware core in this system.

In designing the video/audio single chip, the key question is how to partition between hardware and software in order to maximize performance and minimize cost. Most implementation use dedicated video processors for complex and parallel functions, like video compression and programmable Digital Signal Processors (DSPs) for serial data processing, however as powerful RISC processors are available, the sole software solutions are becoming feasible.

Presently, several vendors have developed highly integrated dedicated video processors. For example, C-Cube Microsystems and LSI Logic offer single chips for JPEG and MPEG algorithms. The Vision Processor (VP) from Integrated Information Technology (IIT) is a high-speed programmable signal processor that implements both DCT and motion compensation.

In our design, the video signal processing part has been implemented by hardwired ASIC and audio signal processing part by software for RISC-type DSP. Since low power con-

Manuscript received May 14, 1999; revised January 25, 2000.

^{a)} Electronic mail: smpark@etri.re.kr

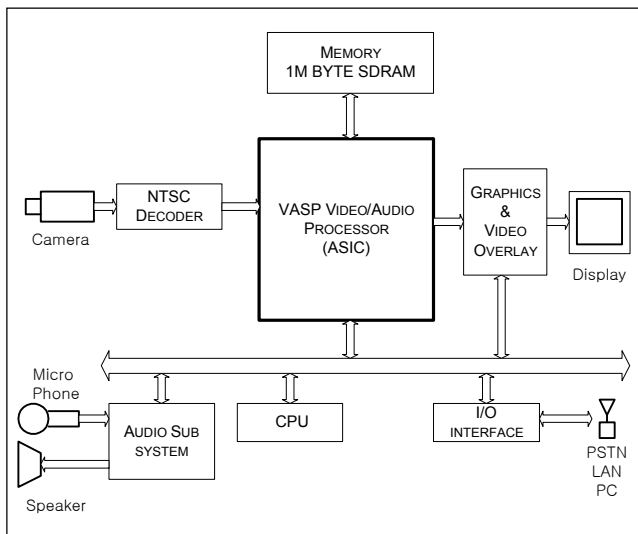


Fig. 1. System block diagram for VASP.

sumption is essential in mobile terminals such as video camera, video phone and IMT-2000 cellular phone and video processing algorithms are not much different from one application to others, video signal processing part is implemented with a hardwired ASIC. On the other hand, as audio signal processing part is varying with application, a flexible software solution with DSP is appropriate. The designed VASP ASIC, which integrated video signal processor and audio signal processor, are targeted for the H.263 and the G.723 standards. The sub-QCIF, QCIF (176×144 , 15 frames/sec), CIF (352×288 , 7 frames/sec) of resolutions is supported and the output bit rates is from 16 to 24 kbps.

In this paper, we present the design and the implementation of the VASP ASIC. In Section II, dedicated and programmable architectures are designed for video/audio signal encoder/decoder using efficient pipeline architecture, optimizing circuitry for each module, optimized buffer size on data processing, and a flexible software solution with DSP. The design verification and methodology are discussed in Section III.

II. DESIGN OF THE VIDEO/AUDIO SIGNAL PROCESSOR

The VASP chip consists of video signal processing block and audio signal processing block. The video signal processing block is designed to implement hardwired solution. The audio signal processing block is implemented with software solution using 16 bits fixed point DSP. This chip has mixed hardware/software architecture to combine performance and flexibility.

The block diagram for the entire design is given in Fig. 2. As shown in Fig. 2, there are basically thirteen major operational blocks in this implementation, namely; pixel input/output, full

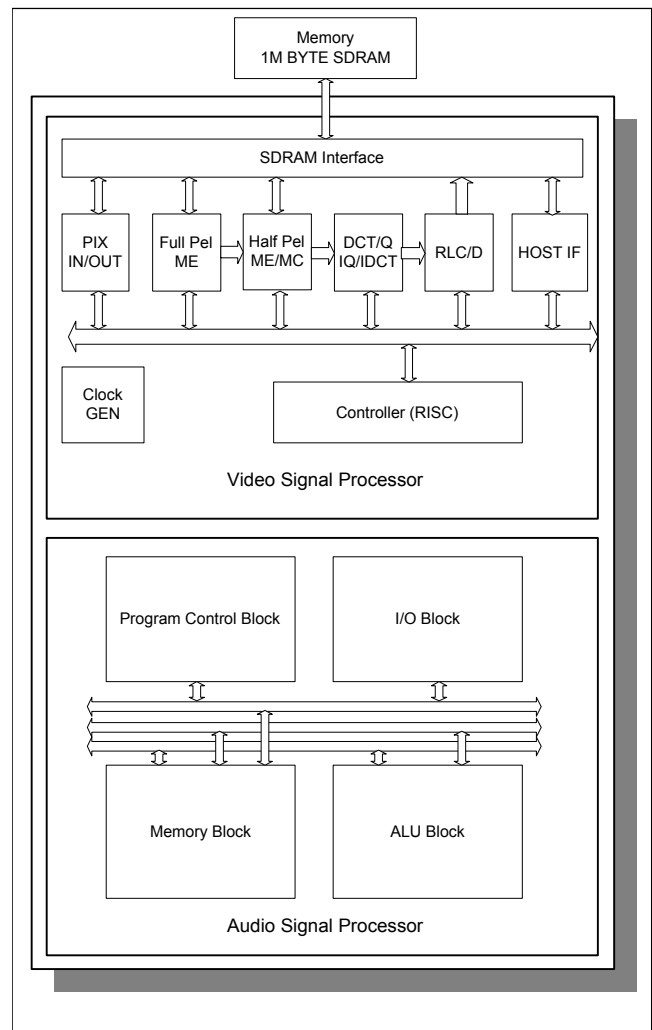


Fig. 2. Block diagram of VASP.

pixel motion estimation, half pixel motion estimation, discrete cosine transform, quantization, run length coding, host interface, internal 16bits RISC type controller, program control block, I/O block, memory block, clock generator block, and ALU block.

Using 54 MHz clock input, clock generation block generates multiple clocks such as 27MHz and 16MHz. The pixel input/output block performs functions such as 4:2:0 chroma-format conversion, temporal noise reduction, and temporal image store. The Full-pel ME block executes full-pel motion estimation (ME), half-pel ME and motion compensation (MC) and Half-pel ME/MC block, respectively. These blocks improve the efficiency of the compression. The Discrete Cosine Transform (DCT), Quantization (Q), Inverse Discrete Cosine Transform (IDCT), and Inverse Quantization (IQ) block transform image data from time domain to frequency domain. Typically, there are few non-zero frequency coefficients in our application and these can be efficiently coded with these blocks. The host interface (Host IF) performs handshaking between host processor

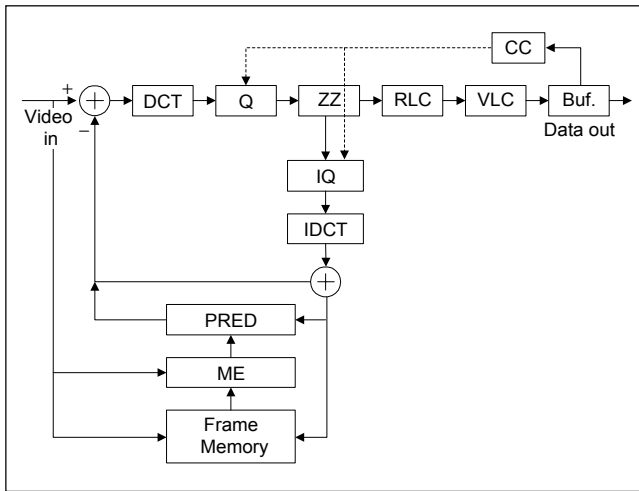


Fig. 3. Encoding flow of video data compression.

and internal registers. The RISC controller performs initializing for each block, scheduling, and error diagnosis. To implement speech coder for audio signal processor, we use 16 bits fixed point DSP, which is developed in ETRI for specific application such as speech compression [15]. The DSP consists of four major function blocks. They are program control block, a memory block, an ALU block and an I/O block

1. Video Signal Processor

Figure 3 shows encoding block diagram for video compression. The VASP architecture has a duplex mode operation. In general, encode mode is more complex than decoding mode because image data reconstruction at this mode is needed. The encoding flow of video data begins by coding an intra-frame block using the DCT transform coding and quantization (intraframe coding), and then sends it to the entropy coding block (RLC, VLC, Buffer) controlled by coding control (CC). The same frame is then decompressed using the inverse quantizer and IDCT, and then stored in the frame memory for inter-frame coding. During the interframe coding, the prediction based on the DPCM algorithm is used to compare every macro block of the actual frame with the available macro block of the previous frame.

The decoder, shown in Fig. 4, consists of the receiver buffer, the Huffman decoder, inverse quantizer, IDCT block, and the motion-compensation predictor, which includes frame memory.

• DCT/IDCT

The Discrete Cosine Transform (DCT) is an orthogonal transform to a set of basis vectors that are sampled cosine functions. A normalized Nth-order DCT matrix C is defined by

$$C_{k,m} = (2/N)^{1/2} \cos[(2k-1)(m-1)\pi/2N] \quad (1)$$

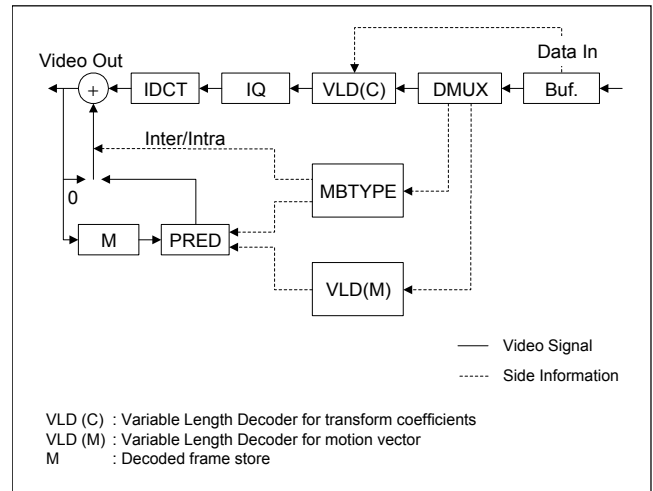


Fig. 4. Decoding flow of video data compression.

For $k = 1, 2, \dots, N$, $m = 2, 3, \dots, N$, and $C_{k,m} = N^{-1/2}$ for $m = 1$. The two-dimensional (2-D) DCT of size $N \times N$ is defined as

$$Z = C^t X C, \quad (2)$$

where C^t is the transpose of C and X is the data matrix.

The DCT has been widely recognized as the most effective technique among various transform coding methods for image and video signal compressions. However, it is computationally intensive and is very costly to implement using discrete component. The source image is divided into 8×8 blocks, and one sample from each block is transformed into the frequency domain using the forward discrete cosine transform equation in the DCT block, as shown in Fig. 3. The components of transformed 64-point discrete signal are called spatial frequencies or DCT coefficients. The DCT is implemented on dequantized coefficients to convert the image from frequency domain into spatial domain using DCT equation as shown in equation (1).

A direct implementation of DCT is computationally intensive. If the two dimension (2-D) DCT of size $16 \times 16 (N \times N)$, about half a billion multiplication and accumulations per second are required. In order to reduce the number of required in is accomplished by using various forms of butterfly structure with fewer number of multiplier [4]. Due to the relatively large area required by a multiplier in an IC realization, many proposed approaches of implementing the DCT have emphasized reducing the required number of multipliers. Nevertheless, many multipliers are still needed to achieve the required throughput rate. Bufferfly approach often results in an irregular architecture and complicated routing that may require large silicon area and long design time. Also, since rounding or truncation in finite-precision arithmetic accompanies multiple stages of multiplication, fixed internal precision can cause the resulting accuracy to be seriously degraded. In order to achieve a regular and efficient

IC realization, the concurrent architecture incorporates distributed arithmetic [5] and the memory-oriented structure to achieve high speed, high accuracy, and efficient hardware realization of the 2-D DCT [6].

In order to solve hardware complexity and long delay time, we implemented DCT block such as row-column separation, modified decomposition, and distributed arithmetic algorithm. The DCT block contains 8 processing elements working in parallel and 64×16 transpose memory. The architecture is highly regular and modular, thus efficient for VLSI implementation. The block was designed for real-time processing of 33 MHz sampled video data. It performs 9-bit of pixel and 12-bit coefficient. The block contains approximately 12,000 gates and 64×16 bit memory.

• Q/IQ

The quantization is basically lossy process, but it is essential for the enhancement of data compression efficiency with perceptual tolerance. The functionality of Q/IQ block conforms to ITU-T Rec. H.263 and its test model, TMN5. To summarize their specifications, the first coefficient of Intra blocks (Intra DC) value is uniformly quantized with a step size of 8 while all other coefficients are done with the step size $2 \times QP$ (Quantization Parameter) by 31 quantizers. Therein, the QP values that derive from transmission rate control may take integer values from 1 to 31. Within a macroblock the same quantizer is used for all coefficients except Intra DC. The overall quantization operation is performed as the equations described below. In case of quantization,

$$\begin{aligned} \text{For Intra DC:} \quad & \text{LEVEL} = \text{COF}/8 \\ \text{For Intra non-DC:} \quad & \text{LEVEL} = \text{Sign}(\text{COF}) \times \{|\text{COF}|/(2 \times \text{QP})\} \\ \text{For Inter:} & \text{LEVEL} = \text{Sign}(\text{COF}) \times (|\text{COF}| - \text{QP}/2)/(2 \times \text{QP}) \end{aligned}$$

where LEVEL indicates the quantized version of the transform coefficients, COF is a transform coefficient to be quantized, and '/' means the integer division with truncation towards zero. For Intra DC the LEVEL values are clipped to the range 1 to 254, otherwise are clipped to the range -127 to 127. Also, in case of the inverse operation,

$$\begin{aligned} \text{If LEVEL} = 0 \quad & \begin{aligned} \text{For Intra DC:} \quad & \text{COF}' = \text{LEVEL} \times 8 \\ \text{Otherwise:} \quad & \text{COF}' = 0 \end{aligned} \\ \text{If LEVEL} \neq 0 \text{ and QP is odd.} \quad & \text{COF}' = \text{Sign}(\text{LEVEL}) \times (2 \times \text{QP} \times \text{LEVEL} + \text{QP}) \\ \text{If LEVEL} \neq 0 \text{ and QP is even.} \quad & \text{COF}' = \text{Sign}(\text{LEVEL}) \times (2 \times \text{QP} \times \text{LEVEL} + \text{QP} - 1) \end{aligned}$$

where COF' is the reconstructed transform coefficient after inverse quantization. Then, the reconstruction levels of all coefficients other than the Intra DC are clipped to the range -2,048 to 2,047. The implemented Q/IQ module hard-wired logic is specific to H.263 only. In general the quantization unit for standards such as JPEG and MPEG contains two or more recommendable quantizer matrixes in ROM or some memory space for user-defined one. Because of hardware complexity and speed, the division operation for quantization is performed by multiplication following by barrel shifting. The code length of look-up table is determined as 13 bits through full input range simulation C model error-free quantization results and minimizing the ROM size. To optimize the code length the amount of barrel shifting is differently applied according to QP range. The underflow and overflow signals for clipping are resulted from barrel shift unit. The input signals to Q unit are transform coefficients, QP, block type, block start, and function mode signal. The Q unit enabled in only encoding mode for reducing the power consumption. The inverse quantization is simple as compared with quantization. It only performs the multiplication and the conversion to odd number. The IQ unit has a register temporarily storing the QP value used at quantization process in encoding mode. The inverse quantization process ended with clipping. The Q/IQ block contains about 3,000 gates.

• Zigzag Block

The zigzag block consists of the zigzag scan, quantization buffer, and inverse zigzag scan. The zigzag block consists of an input register, four types of address generators, RAM (64×8), and control part. Input register latches data from quantization block. The address generators block is four types of address counter. These are raster scan of run length coding, zigzag scan of run length coding, raster for inverse quantization, and inverse zigzag of run length decoding. We designed multiple access architecture using the proper control of memory timing. This architecture uses one memory for the implementation of raster scan, zigzag scan, quantization buffer, and inverse zigzag scan [7]. There are two ideas for implementing this block as follows; one idea is those memories read and writes cycles are two times larger than data speed. This means the data process of memory performs read and write operation for one cycle. The other is that the control timing of read memory starts after 32 cycles of write. Zigzag address and raster address can operate during read/write cycles without overlapping.

• RLC/D

The run length coding algorithm is widely used to implement multimedia applications such as JPEG, H.261, H.263, and MPEG standardization. The zigzag scan is the ordering of Dis-

crete Cosine Transform (DCT) coefficients in an order from low to high frequency. This ordering increases the number of long runs for more efficient run coding. Run length coding is the characterization of a signal in terms of its non-zero values and the number of zero between adjacent non-zero values. After transform coding and quantization, there are typically few non-zero frequency coefficients and these can be efficiently coded with this technique. The run length encoding/decoding block is one of entropy reduction. Run length encoding block is aimed to generate the run for block layer streams (8×8), and also to identify intra DC, inter DC, AC, and last AC. The format of the output is compatible with the stream of the variable length encoding in H.263 recommendation. The operation of this block is based on 8×8 pixel data with the serial input data. The first operation step is initial condition that the count value is reset and the state is initialized. The second operation step generates run counting and level through the finite state machine until count value is 62. The third operation step if count value is the 63rd, decides to the last AC or run value of temporary register. Last step, after decision for the last AC or run value outputs it as the format of 16 bits after delayed two cycles. The proposed technique is based on the specified scheme of the finite state machine for the run length coder and the multiple access architecture with one memory for the implementation of the zigzag scan, quantization buffer, and inverse zigzag scan. The run length coder block is designed by register transfer level (RTL) of VHDL. Also the simulation tool is used Compass Qsim. The run length coder block contains 4,000 logic gates and total 1,536 bits of Static RAM. The case simulations confirm the function of the designed block [8].

• ME/PRED

Typical video compression processors today include VLSI motion estimators that implement the full-search block-matching (FSBM) algorithm. In the block-matching motion estimation, the motion vector is the displacement of a macroblock with the minimum distortion from the reference macroblock. The full-search block-matching algorithm determines the motion vector by identifying a macroblock with the minimum distortion from a pool of all possible candidate blocks in the search area. The FSBM algorithm thus offers the optimal solution; however, existing implementation of this algorithm are computationally expensive and power hungry because they typically compute the distortion value of all possible candidate macroblock. To reduce the computational complexity and the power consumption of motion estimation, several fast block-matching algorithms, such as two-dimensional (2-D) logarithmic search [9], three-step search [10], and conjugate direction search [11], has been proposed. Although these approaches reduce power consumption, they result in sub-optimal solutions because the search

spaces are necessarily reduced. References [9]–[11] have suggested ways of lowering power consumption by implementing the binary level-matching criterion. Lowering the supply voltage in the motion estimator to save power motion estimation, which results from search window management and demand clocking. A low-power chipset for a portable multimedia was introduced for one-way full-motion video in [12]. Bierling [13] gives an overview of video coding VLSI's, focusing on the power consumption reduction. However, none of the above addresses the issue of lowering power consumption of the block-matching algorithm at the architectural level without sacrificing the optimality of the solution. We introduce an architectural enhancement to reduce the power consumption of the full-search block-matching motion estimation. Our approach to reduce the power consumption in FSBM motion estimation is based on eliminating unnecessary computation using conservative approximation.

Proposed motion estimation is based on linear semi-systolic array architecture using full-search block matching algorithm. Memory architecture for proposed motion estimator is suitable for VLSI because of its minimized physical size and maximized efficiency of sixteen processing elements.

• RISC/DMA

The RISC block performs the setting of initial value for internal registers, the scheduling of control scheme for data flows, and error diagnostics. It has a Harvard architecture that separated instruction bus with data bus, and 4-step pipeline operation. The Direct Memory Access (DMA) block performs a data transfer between internal processes units and external frame memory.

2. Audio Signal Processor

To implement speech coder for audio signal processor, we use 16 bits fixed point DSP, which is developed in ETRI for specific application such as speech compression [15]. The DSP consists of four major function blocks. They are program control block, memory block, ALU block and I/O block as shown in Fig. 5.

The DSP has RISC type instruction set, distributed decoding, alternative program fetch, repeat control and dual bank memory system. The major features of DSP are as follows.

- 28 nsec Instruction cycle (40 MIPS)
- MAC operation and 32 bits data load in one cycle
- Two 36 bits accumulators
- 16 K \times 24 bits program ROM internally
- Internal 3 K \times 16 bits data RAM and 10 K \times 16 bits data ROM
- One serial port and one 16 bits parallel port

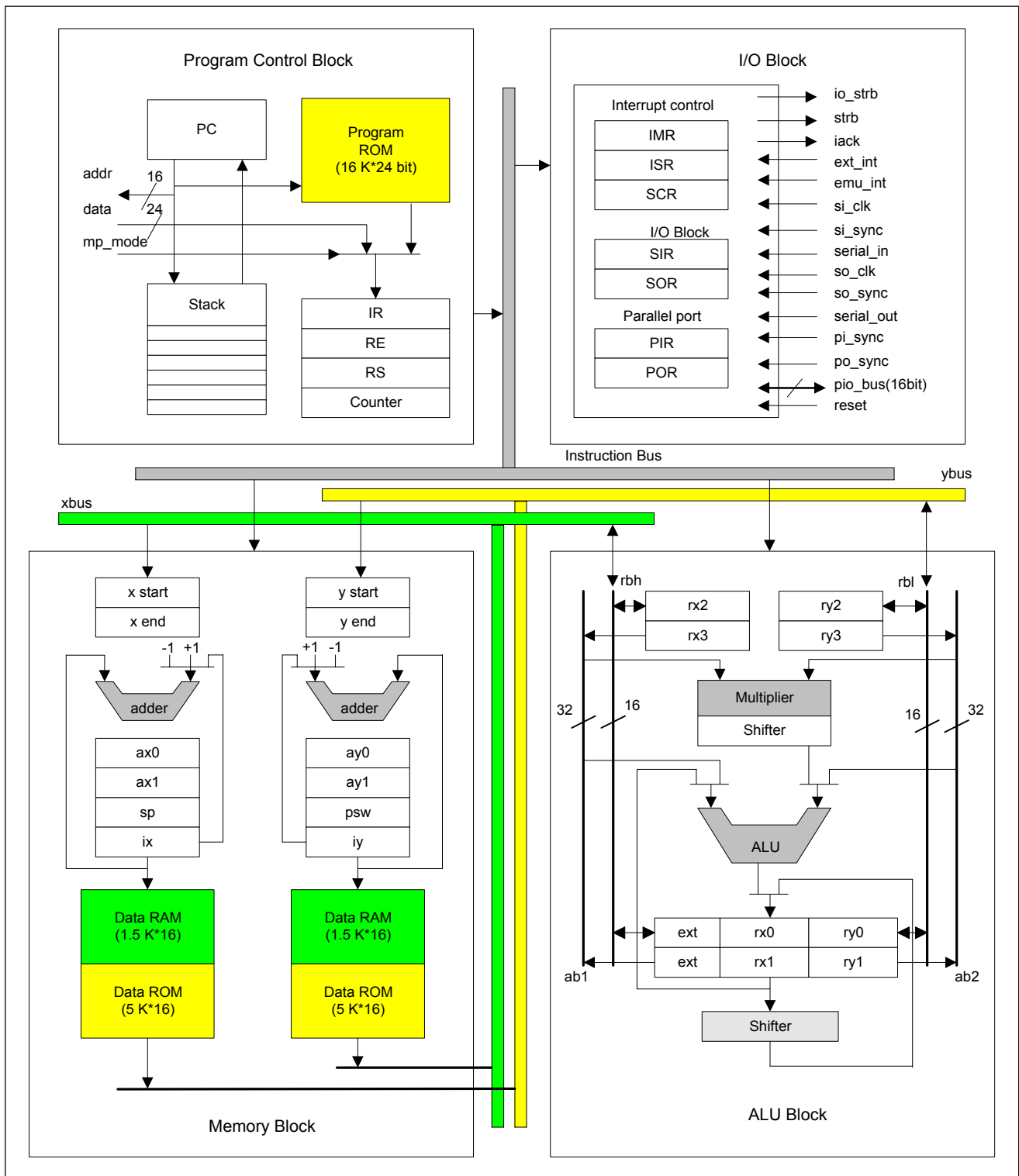


Fig. 5. Block diagram of the DSP for VASP.

We use the G723.1 fixed-point C source code that is standardized by ITU. Therefore we developed assembly code that is bit-exactly same as standard C code. Because we can not have an opportunity to develop compiler for our DSP, we im-

plement G723.1 by fully handwriting assembly code. The table as follows shows a G723.1 implementation result.

To implement speech coder, DSP must have low power consumption, fast operating capability more than 30 MIPS and small

Table 1. Result of implementation of G.723.1.

Program ROM	11 K words
Data RAM	2.9 K words
Data ROM	9.5 K words
Complexity	27 MIPS

chip size. To meet the requirements mentioned above, We adopted RISC type instruction set, distributed decoding, alternative program fetch, repeat control and dual bank memory system in our DSP architecture.

In general purpose DSPs, CISC type instruction set and diverse addressing modes are supported. However, RISC type instructions set and limited addressing modes simplify the control logic, which reduces die size and power consumption. Moreover, by adopting 24bits instruction format, the immediate and direct addressing modes can be coded in one instruction word. This feature enables execution of all instructions in one clock cycle.

The DSP contains internal program ROM whose size is 16K \times 24bits. ROM access time imposes bottleneck for pipeline scheme. The enough precharge time in the given pipeline time constraints is obtained by dividing the program ROM into 2 separate banks. The one bank includes odd address instruction codes, the other contains even address instruction codes. When PC address is odd, the odd part ROM operates, and the even part ROM is precharged. This alternating scheme can reduce ROM access time to half.

Repeat instruction in Vocoder DSP executes loop operation without branch overhead of the pipelining. Repeat block consists of RE (repeat end register), RS (repeat start register) and RC (repeat counter register). Each register has three stack registers. Repeat stack stores current RS, RE and RC contents when the repeat is nested. Therefore, the vocoders DSP can supports four depth nested repeat.

• G.723.1 Speech Coder

G.723.1 speech coder is designed to compress the speech or other audio signal component of multimedia services at a very low bit rate and standardized in 1995. The block diagram of the coder is shown in Fig. 6 [16].

It has two different bit rates (6.3kbit/s and 5.3kbit/s). This coder has a good quality for speech and audio signal at both rates. The coder works on 240-sample frame at 8 KHz (30ms) with an additional look ahead of 7.5ms, so there is an algorithmic delay of 37.5ms.

The coder is based on the principles of linear prediction analysis-by-synthesis coding and attempts to minimize a per-

ceptually weighted error signal. Each frame samples are first high pass filtered to remove the DC component and then divided into four sub-frames of 60 samples each. For every sub-frame, a 10th order linear prediction coefficient is computed using the unprocessed input signal. The LPC filter for the last sub-frame is quantized using a Predictive Split Vector Quantizer (PSVQ).

The unquantized LPC coefficients are used to construct the short-term perceptual weighting filter, which is used to filter the entire frame and to obtain the perceptually weighted speech signal.

For every two sub-frames (120 samples), the open loop pitch period, is computed using the weighted speech signal. This pitch estimation is performed on blocks of 120 samples. The pitch period is searched in the range from 18 to 142 samples.

From this point the speech is processed on a 60 samples per sub-frame basis. Using the estimated pitch period computed previously, a harmonic noise-shaping filter is constructed. The combination of the LPC synthesis filter, the formant perceptual weighting filter, and the harmonic noise-shaping filter is used to create an impulse response. The impulse response is then used for further computations.

Using the pitch period estimation, L_{ol} , and the impulse response, a closed loop pitch predictor is computed. A fifth order pitch predictor is used. The pitch period is computed as a small differential value around the open loop pitch estimate. The contribution of the pitch predictor is then subtracted from the initial target vector. Both the pitch period and the differential value are transmitted to the decoder.

Finally the non-periodic component of the excitation is approximated. For the high bit rate, multi-pulse maximum likelihood quantization (MP-MLQ) excitation is used, and for the low bit rate, an algebraic codebook excitation (ACELP) is used.

III. DESIGN AND VERIFICATION METHODOLOGY

The top-down ASIC design begins with an ASIC requirement specification, followed by the behavioral verification of system/ASIC algorithms. However, analyzing the behavior of ASIC may not be sufficient to detect all the errors in the circuit. In such cases, register transfer level [RTL] of gate-level description in more appropriate for pinpointing the errors related to the critical behavior of the ASIC. Considering all the above possibilities, mixed level simulation with VHDL simulator is the best candidate for the overall system simulation [14].

Design verification and methodology was developed by us as shown in Fig. 7. We developed C models for major functional blocks of the H.263 video codec and performed high level simulation. Also we modeled external environment using VHDL, which are ARM system, ARM Bridge, and synchronous dynamic

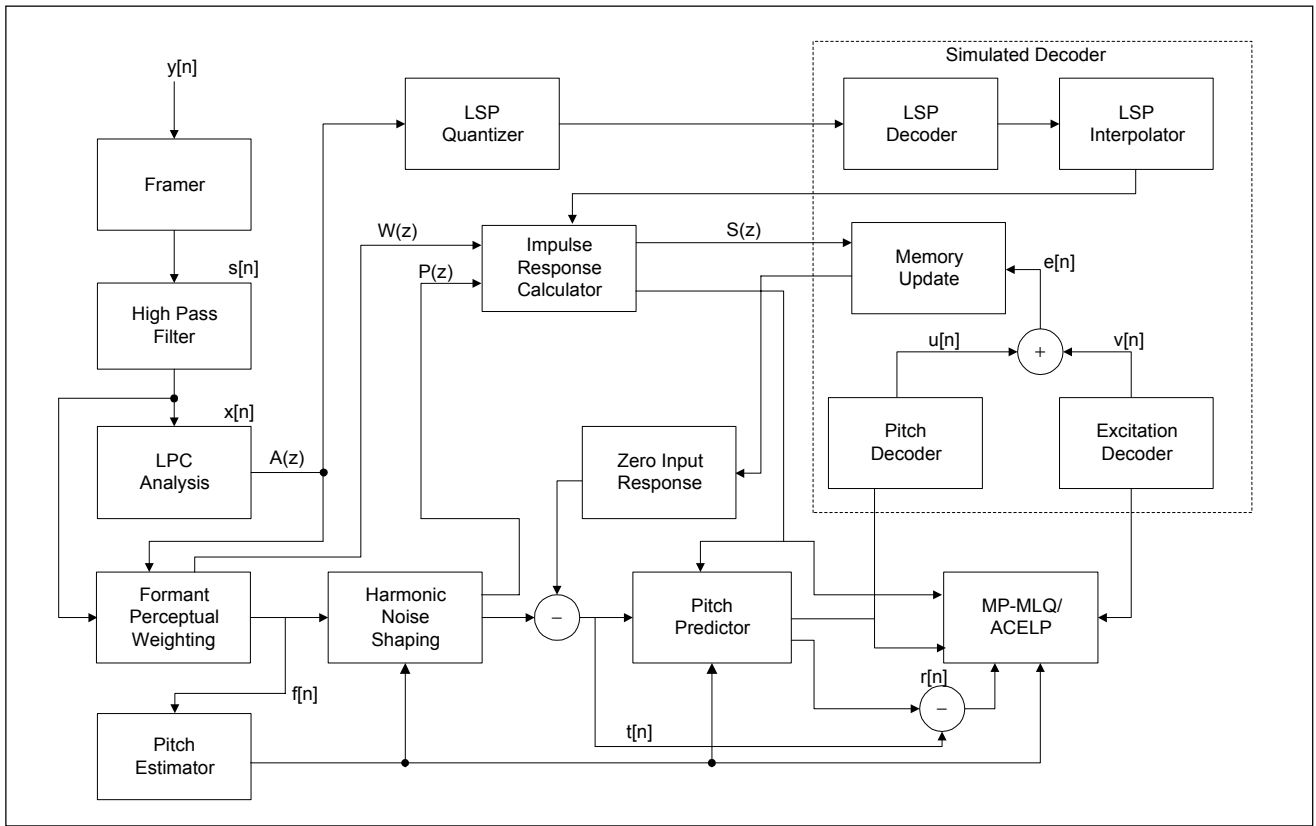


Fig. 6. Blockdiagram of the G.723.1 coder.

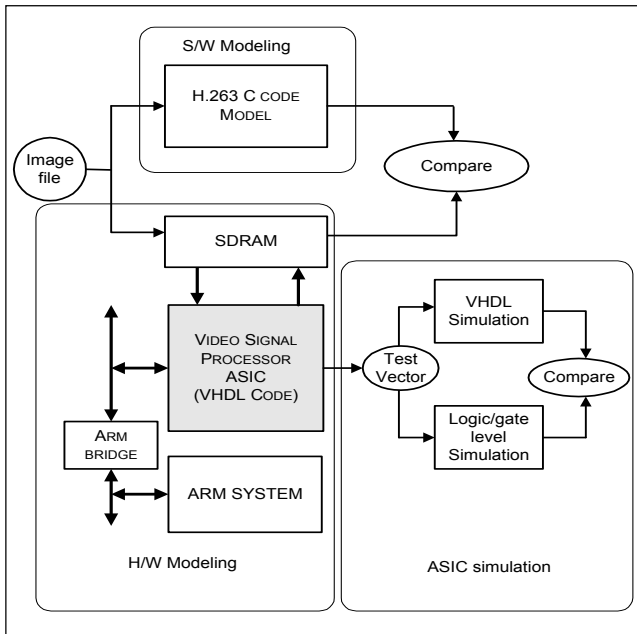


Fig. 7. Verification environment for VASP.

access memory (SDRAM). Simulation and testing for the result of the software and the hardware are carried using co-simulation environment. The test vectors for high-level simulation are used

for verification from RTL-level VHDL simulation through gate-level simulation.

Hardware accelerator is used for a long time simulation of video codec. Using Compass simulator, the simulation time took about 7 days on ULTRA1 workstation for one frame image of video codec, which reduced to two hours for one frame image of video codec using IKOS NSIM64 hardware accelerator.

IV. CONCLUSION

In this paper, we present a design of video and audio single chip processor for low bit rate application as H.263 and G.723 standards. This chip has a mixed hardware and software architecture to combine performance and flexibility.

We perform top-down design methodology followed by the behavioral verification of system/ASIC algorithms. C models are developed for major functional blocks of the H.263 video codec and high level simulation are performed. The chip is verified by system level simulation and gate level simulation with top-down methodology.

The video signal processing block was designed to implement hardwired solution of pixel input/output, full pixel motion estimation, half pixel motion estimation, discrete cosine transform, quantization, run length coding, host interface, and 16 bit

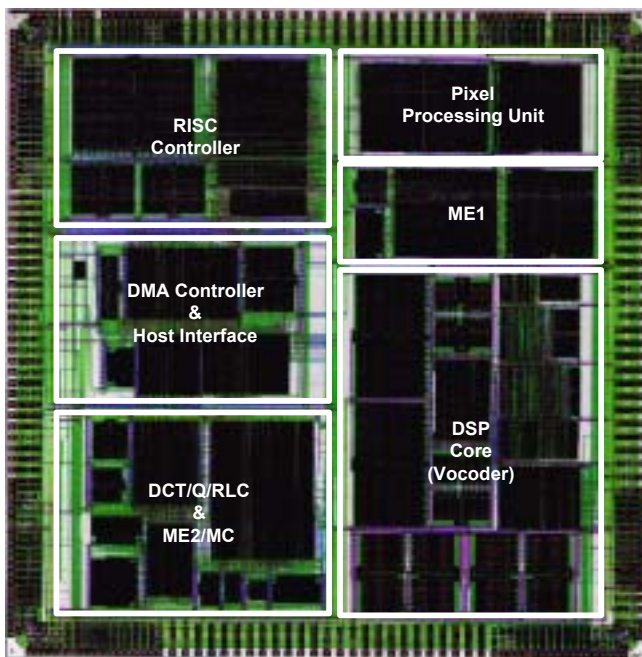


Fig. 8. VASP layout photograph.

RISC type internal controller. The audio signal processing block was designed to implement software solution of 16 bits fixed point DSP. This chip contains 142,300 gates, 22 k FIFO, 107k SRAM, and 556k ROM, and the chip size is 9.02 mm × 9.06 mm, which is fabricated using 0.5 micron 3 metal CMOS technology. It was packaged in 208 metrics quad flat package (MQFP). Figure 8 shows a microphotograph of the VASP chip.

REFERENCES

- [1] ITU-T Recommendation Draft H.263, *Video Coding for Low Bit Rate Communication*, 1995.
- [2] Majid Rabbani and Paul W. Jones, *Digital Image Compression Techniques*, SPIE Optical Engineering Press, 1991.
- [3] Reza Hashemian, "Design and Hardware Implementation of a Memory Efficient Huffman Decoding," *IEEE Consumer Elec.*, Vol. 40, NO.3, August 1994, pp. 345–352.
- [4] W. H. Chen, C. H. Smith, and S. C. Fralick, "A Fast Computational Algorithms for the Discrete Cosine Transform," *IEEE Trans. Commun.*, Vol. COM-25, September 1977, pp. 1004–1009.
- [5] A. Peled and B. Liu, "A New Hardware Realization of Digital Filters," *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-22, December 1974, pp. 456–462.
- [6] M. T. Sun, L. Wu, and M. L. Liou, "A Concurrent Architecture for VLSI Implementation of Discrete Cosine Transform," *IEEE Trans. Circuits Syst.*, Vol. CAS-34, August 1987, pp. 992–994.
- [7] S. M. Park *et al.*, "An Area Efficient Implementation of Zig-Zag and Quantization Buffer for H.263 Application," *ITC-CSCC '98 Proceeding*, Vol. I, July 1998, pp. 907–909.
- [8] S. M. Park *et al.*, "An Efficient Implementation of Run Length

- Coder for H.263 Application," *5th International Conference on VLSI and CAD Proceeding*, October 1997, pp. 430–432.
- [9] J. R. Jain and A. K. Jain, "Displacement Measurement and its Application Interframe Image Coding," *IEEE Trans. Commun.*, Vol. COM-29, December 1981, pp. 1799–1808.
- [10] S. G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 11, July 1989, pp. 674–693.
- [11] I. Daubechies, "Orthonormal Bases of Compactly Supported Wavelets," *Commun. Pure Appl. Math.*, Vol. 41, November 1988, pp. 909–996.
- [12] F. Dufaux and F. Moscheni, "Motion Estimation Techniques for Digital TV: A Review and a New Contribution," *Proc. IEEE*, Vol. 83, June 1995, pp. 856–876.
- [13] M. Bierling, "Displacement Estimation by Hierarchical Block Matching," *Proc. SPIE, Visual Commun. Image Processing*, Vol. 1001, Cambridge, MA, Nov. 1988, pp. 942–951.
- [14] Kwangil Yeon *et al.*, "Design of chip for CDMA Mobile Station," *ETRI Journal*, Vol. 19, No. 3, October 1997, pp. 228–241.
- [15] ITU-T Recommendation G723.1, *Dual rate speech coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s*, 1996.
- [16] H. Y. Yoo, J. J. Kim, K. J. Byun, K. C. Han, D. K. Kim, J. S. Kim, H. B. Lee, and M. J. Bae, "Efficient DSP Design for Vocoder Application," *CICC '95*, pp. 189–192.



Seongmo Park received the B.S and M.S. degrees in electronics engineering from Kyungpook National University, Taegu, Korea, in 1985 and 1987, respectively. From 1987 to 1992, he was with the LG Semiconductor Company, Gumi, Korea, where he worked on ASIC design and Mask ROM design. In 1992, he joined the

Electronics and Telecommunications Research Institute (ETRI), Taejon, Korea for the development of ASIC design. He currently engaged in research on MPEG-4 ASIC design, image compression algorithm and VLSI architecture design. His main research interests are video coding, image compression, and low power VLSI architecture design.



Seongmin Kim was born in Gumi, Korea, on April 11, 1959. He received the B.S. degree in electronic engineering from Kyungpook National University in 1982. In 1983 he joined the Semiconductor Division, KIET, which was consolidated to ETRI in 1985. During 1983 through 1995 he was engaged in the semiconductor quality and reliability research. From 1997 he has developed the motion estimation and compensation and quantization macrocell for the video compression/decompression processor VLSI's in VLSI Architecture Team.



Igkyun Kim received the B.S. and M.S. degrees from Kyungpook Nat'l University, Taegu, Korea, in 1978 and 1980, respectively. Since joining the Electronics and Telecommunications Research Institute (ETRI), Teajon, Korea, in 1984, he has been engaged in the research and development in the field of CMOS LSI's. His recent research involves the development of processor system

for video signal processing including high-speed signal processor architecture.



Kyungjin Byun was born in Seoul, Korea, on February 11, 1962. He received B.S. degree in electronics engineering from Kookmin University, Korea, in 1987, and M.S. degree in electronics engineering from Information and Communications University (ICU), Taejeon, Korea, in 2000. Since 1987, he has been with Electronics and Telecommunications Research Institute (ETRI),

Taejeon, Korea, as a senior member of technical staff, and involved in the development of digital communication system. His current research interests are in speech coding, digital signal processing and its applications.



Jin Jong Cha received the B.S. degree in electronic engineering from Hanyang University, Seoul, Korea, in 1980. He received the M.S. degree in electrical and electronics engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1982. He joined the Electronic and Telecommunications Research Institute (ETRI) in 1982. He is a principal member

of engineering staff and project manager of VLSI Architecture Team. His current interests include VLSI architecture and circuit design on video signal processing, voice and audio signal processing, and base-band signal processor in wireless and mobile applications.



Hanjin Cho was born in Seoul, Korea on July 8, 1960. He received the B.S. degree in electronic engineering from Hanyang University in 1982. He received M.S. and Ph.D. degrees in electrical engineering from New Jersey Institute of Technology in 1987, and from University of Florida in 1992, respectively. He joined ETRI in 1992, where he is currently in charge of the Integrated

Circuit Design Research Department. His research interests include VLSI circuit design, system-on-a-chip design, design methodology development.