A Low Power Design of H.264 Codec Based on Hardware and Software Co-design

Seongmo Park | Sukho Lee | KyoungSeon Shin | Jae-Jin Lee
Moo-Kyoung Chung | Jun-Young Lee | Nak-Woong Eum
ETRI

Abstracts

In this paper, we present a low-power design of H.264 codec based on dedicated hardware and software solution on EMP(ETRI Multi-Core Platform). The dedicated hardware scheme has reducing computation using motion estimation skip and reducing memory access for motion estimation. The design reduces data transfer load to 66% compared to conventional method. The gate count of H.264 encoder and the performance is about 455k and 43Mhz@30fps with D1(720x480) for H.264 encoder. The software solution is with ASIP(Application Specific Instruction Processor) that it is SIMD(Single Instruction Multiple Data), Dual Issue VLIW(Very Long Instruction Word) core, specified register file for SIMD, internal memory and data memory access for memory controller, 6 step pipeline, and 32 bits bus width. Performance and gate count is 400MHz@30fps with CIF(Common Intermediated Format) and about 100k per core for H.264 decoder.

I. Introduction

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 JPEG (Joint Photographic Expert Group), H.261, H.263, MPEG, G.723, H.324, H.264[1] and other data compress standards. The new H.264 video coding standard has gained more and more attention recently, mainly due to its higher coding efficiency over previous standards [2]. These applications requires low power consumption and to reduce memory bandwidth access. MPCore(Multi Processor Core) and embedded platforms are becoming popular for mobile video applications because of their good balance between power efficiency, performance, and flexibility [3]. In designing a single-chip implementation, the key question is how to balance the partition between hardware and software in order to maximize performance and minimize cost.

Most implementations use dedicated video processors for complex and parallel functions, like video compression and programmable DSPs for serial data processing. However, as powerful reduced instruction set computer (RISC) processors are becoming available, sole software solutions

This work was partly supported by the IT R&D program of MIC/IITA. [2007-S-026-02, Multi-Format Multimedia SoC based on MPCore Platform]

have become feasible. H.264/AVC provides higher coding efficiency through added features and functionalities. It also analyzes computational complexity of the software-based H.264/AVC baseline profile decoder [4].

However, such features and analyses are software-based solutions, and it is hard to implement them in real time. The H.264 profile video decoder with extremely low power dissipation meets the growing demands for low-cost implementation of such terminals. These applications require low power consumption and fast memory bandwidth access. To make a chip with low cost, high flexibility, and low power, a programmable processor is suitable for adaptive processing and various algorithms. However, power consumption of high-performance processors is high. On the other hand, dedicated hardware has lower power and higher performance than software implementation [5], [6], and [9].

In this paper, we introduced reducing power consumption at an algorithm level, architectural level using dedicated hardware solution and software solution for H.264 Codec. In algorithm level, we try to design parameters of encoder such as searcher Search range, Hadamard on/off, Number of reference frames, format of stream out, and Rate Distortion on/off. We selected to reduce complexity and hardware cost with little degradation and performance. The dedicated hardware of architecture level presents a design motion estimation to reduce the power consumption and the external memory bandwidth access with mixed mode motion estimation algorithm and its implementation. The software of architecture level is that ASIP(Application Specific Instruction Set Processor) is a powerful solution when the contradicting requirements such as performance and flexibility have to be jointly satisfied with a single task block[7-8].

This paper presents low power design for the H.264 video codec with high level design and architecture level design. Section II describes algorithm of the H.264 encoder using parameters. In Sections III and IV, low-power schemes employed in the H.264 codec, design are discussed. Verification FPGA implementation are presented in Sections V and VI, respectively.

II. Design of Parameters for H.264 Encoder

We try to design parameters of encoder such as searcher Search range, Hadamard on/off, Number of reference frames, format of stream out, and Rate Distortion on/off. We selected to reduce complexity and hardware cost with little degradation and performance.

Search range is main parameter in motion estimation, we simulate with JM software. Table 1 is result of simulation with PSNR, Simulation time, motion estimation Time, and memory cost. Best condition is +/- 16 of search range.

Table 1. Compare to Search Range

Search Range	PSNR Memory	Simulation Time(sec)	Motion Estimation Time (sec)
+/-16	35.74 48x48	40,624	15,712
+/-32	35.73 90x90	94,110	66,612
+/-64	35.74 44x144	321,157	290,247
+/-128	35.74 272x272	2010,209	1984,931

* Reference frame : 1 , foreman CIF 10 frame, PC 3.4 GHz

Table 2. Foreman (CIF, 30Hz, 256kbps)

RD	Ref.	PSNR	Mode					
		Y	copy	16x16	16x8	8x16	8x8	4x4
off	1	34.10	19238	11428	3570	3249	1529	95
	2	34.17	18868	11789	3891	3597	880	89
on	1	34.56	16995	11223	4737	4690	1107	57
	2	34.75	16031	12856	4287	4310	1287	65

Table 2 describes simulation on RD-off/on, references

frame, the mode 0 (copy), the mode 1 (16x16), the mode 2 (16x8), the mode 3 (8x16), the mode 4 (8x8), and mode 5 intra 4x4 with Foreman test sequences. Table 2 shows the 0.06 dB degradation of PSNR between reference frame one and reference frame two for foreman sequences. The degradation of PSNR is 0.46 dB for RD off and on. The intra 4x4 of mode has below 0.3% (92/31680 MB) with foreman sequence. We decide the parameters that search range is +/- 16.5, Hadamard is on, reference frame is one frame. block mode is 16x16, 16x8, 8x16, and 8x8. rate distortion is off.

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.

This method described here is on the motion estimation using motion estimation skip and hierarchical search (2:1 sampling) for image compression. We proposed a design method of motion estimation with motion estimation skip.

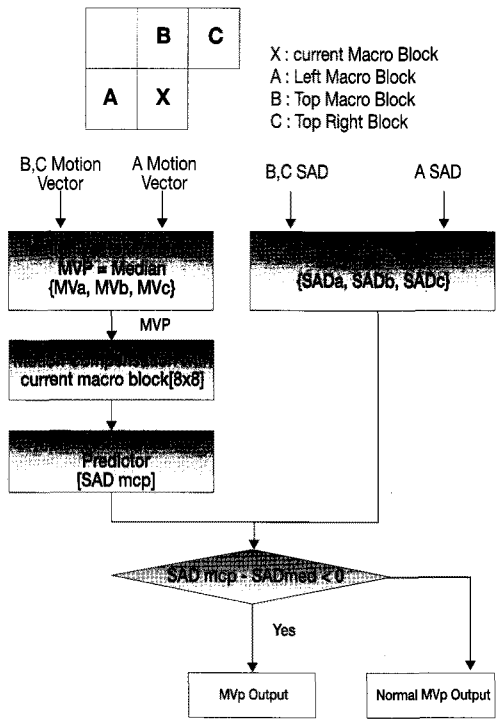

Fig. 1 shows the mixed mode motion estimation operated as follows.

The proposed hardware consists of reference data down sampling block, current data down sampling block, demultiplex, two reference memories, current memory, processing elements array, motion estimation skip, motion Fig. 1. Motion estimation Skip flow chart estimation skip processing block and comparator.

First

1. Three Motion Vectors (B,C, and A) inputs motion vectors for motion vector prediction.

To calculation for motion vectors gets median value among inputs vectors.

To run motion compensation with motion predictor value and current macro block.

To get SAD (Sum of Absolute Difference) for motion compensation predictor value.

Second

1. Three SAD values take a previous macro block, top macro block, top right macro block.
2. To get a maximum SAD from inputs SADs

Third

1. The first step and the second step.
2. Compare to SADmcp and SADmax
3. If SADmcp is less then SADmax, skip flag is on.

If skip flag is on, motion estimation is not normal operation. If skip flag is off, motion estimation is normal operation.

III. Design of Architecture for H.264 Encoder

Fig. 2 shows the proposed H.264 encoder architecture. It has full hardware architecture and consists of video input module (VIM), integer motion estimation (IME), fine motion

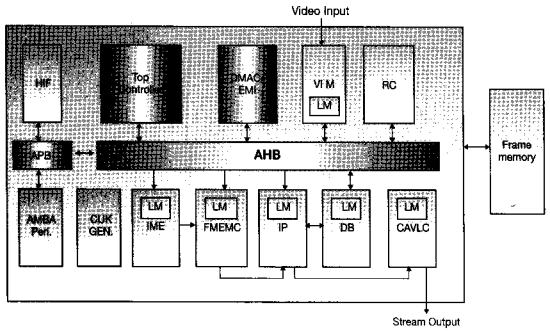

Fig. 2. H.264 encoder hardware architecture.

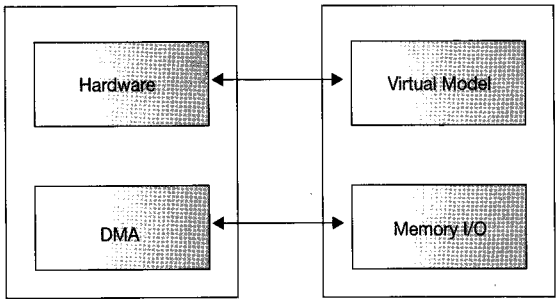

Fig. 3. Cycle estimation model

estimation (FME)/motion compensation(MC), intra prediction(IP) that include discrete cosine transform(DCT), inverse DCT(IDCT), quantization(Q) and inverse quantization(IQ) for reconstruction. In addition, context-adaptive variable length coding(CAVLC), de-blocking filter(DB), 6-channel direct memory access controller and SDRAM controller(DRAMC) are in this engine. Its scheduler and pipeline controller are also fully hardwired FSM (Finite State Machine) controller. Each VIM, IME, FME/MC,IP and DB has private local memory. That reads and stores current frame data and reference (previous) frame data from frame memory in the unit of a macroblock.

The macroblock, corresponding to a 16x16-pixel region of a frame, is the basic block unit for motion compensated prediction in a number of important visual coding standard including MPEG-1, MPEG-2, MPEG-4 Visual, H.261, H.263 and H.264. An MPEG-4 Visual or H.264 CODEC process each video frame in units of a macroblock[2].

Fig. 3 is cycle estimation of H.264 encoder. We describe a method of using virtual model for hardware, and memory I/O for direct memory access(DMA). The method consists of three steps. In the first step, we computer the number of cycles required to execute a hardware part. In the second step, the cycle count estimate is derived by direct memory access from external memory to local memory. It contains cycle estimation for modules, which are real time operation for H.264 decoder. The computational cost and the I/O bandwidth greatly increase, which is a serious problem in the multimedia system-on-a-chip (SoC) design. To overcome these problems for implementation as hardware.

Direct Memory Access and External Memory Interface block performs a data transfer between internal memory and external frame memory. Data transfer of the chip has one cycle operation between DMA and external frame memory. It supported several mode operations with AMBA AHB specification, high performance bus.

DMA controller has special features. It can interface with all of internal modules with only one Channel, is made up programmable DMA and supports both burst block mode and packet mode for data transfer. On the other hand, it has architecture of dual addressed DMA without buffered memory. In dual addressed DMA transfer, and explicit address is required for both and destination. Each DMA transfer then requires two bus transactions; the first to write the data to the destination. An explicit address is generated for each transaction, and the data must be buffered internally by the DMA controller between read and write. But the DMA controller of this chip need not buffered memory. So it can transfer data between internal local memories and external memory at two times speed better than conventional DMA controller. It can reduce required clock rate and data bandwidth. Read and Write cycle is based on 32bits bus.

Because most of processing cycles is due to memory

access between external frame memory, we present the reducing memory access scheme at IME module to reduce memory bus bandwidth. When integer motion estimation reads and stores current and reference (previous) macroblock data to local memory from external frame memory for searching matching region, it performs 2:1 sub-sampling for each macroblock data[3] because we use 2-step hierarchical(integer-) half-pixel) search algorithm. Therefore, current MB memory size is 8x8 from 16x16 and reference MB memory size is 24x24 from 48x48. IME calculates SAD by 8-processing elements(PEs), which can process an eight row unit of reference memory data at a time. If IME pre-fetches one sub-sampled 8x8(16x6) reference data at first time, after that time, only one third of reference data is patched for SAD calculation in search window. As the result of that, it reduces the memory bandwidth by one third compared with full memory access [3][9].

The average bit rate and quality (PSNR) with three resolutions are described in Table 3. ETRI dB is a case of 16x16 inter mode. Its PSNR is somewhat lower compared with JM. But considering hardware size, this distortion is negligible.

Table 3. The average bit rate and PSNR

Sequence	Resolution	Bit rate (bps)	JM (dB)	ETRI (dB)
CIF (foreman)	352x288	256K	34.26	33.00
	352x288	2M	43.49	42.67
VGA (myname_VGA)	640x480	256K	29.32	28.85
	640x480	2M	36.29	35.67
D1 (myname_D1)	720x480	256K	29.11	28.64
	720x480	2M	35.87	34.89

The logic gate count of each block and local memory is in Table 4.

Table 4. Logic gate count

Block	Gate count
VIM	15K
IME	70K
FME	80K
IP & DCTQ/IQIDCT	200K
DB	30K
CAVLC	20K
INTERNAL MEMORY	40K
TOTAL	455K

IV. Design of Application Specific Processor

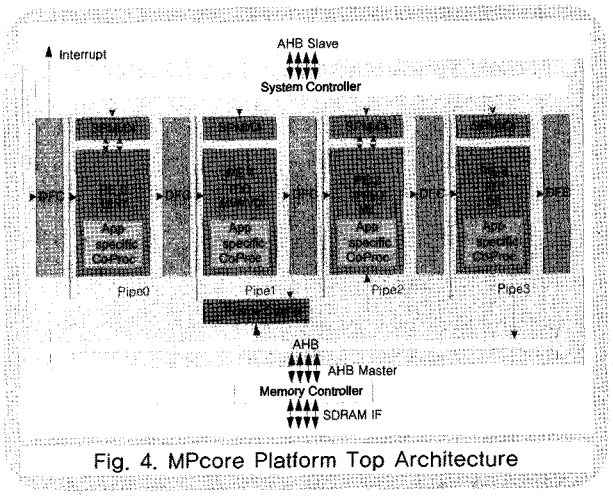

Fig. 4. MPcore Platform Top Architecture

Fig. 4 shows EMP that they consists of system controller, SPM(Scratch Pad Memory), Processing Elements, DFC(Data Flow Controller), Frame Cache, and memory controller. System Controller programs each ASIPs and inputs video stream operates as local memory transfers data between ASIPs and frame cache. The proposed EMP is 4 step pipelined architecture with parallel processing between processing elements and others processing element according to application, Four ASIPs(Application-Specific

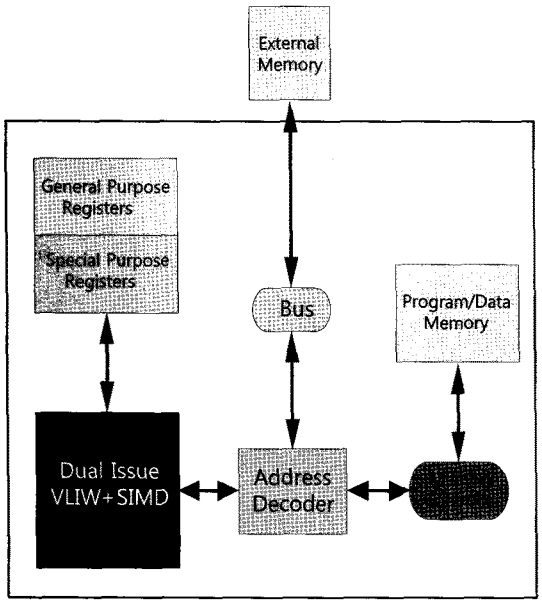

Fig. 5. Block diagram of ASIP

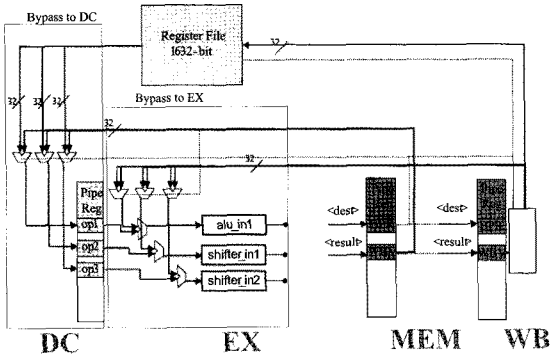

Fig. 6. Block diagram of bypass logic

Instruction-set Processor), DFC of self control method, frame cache controller for DMA access, AHB bus interface, and task parallelism of H.264, and verilog RTL design.

This pipeline architecture makes each ASIP perform a different part of video processing tasks, so that inputted data stream to the first DFC is sequentially processed through the ASIPs. Overall performance of the system increases linearly according to the number of pipelined ASIPs, if the jobs partition is well balanced. We partitioned H.264 decoder into four tasks as shown in the Fig. 4 and achieved 3.5 times performance more than that of single ASIP.

Fig. 5 shows the block diagram of application specific instruction-set processor consisting of dual issue VLIW(Very Long Instruction Word) core including SIMD(Single Instruction Multiple Data) instructions, special register file for SIMD operation and memory controller for 32-bit internal and external memory access.

ASIP is a powerful solution when the contradicting requirements such as performance and flexibility have to be jointly satisfied with a single task block.

The proposed application specific architecture based on 6-

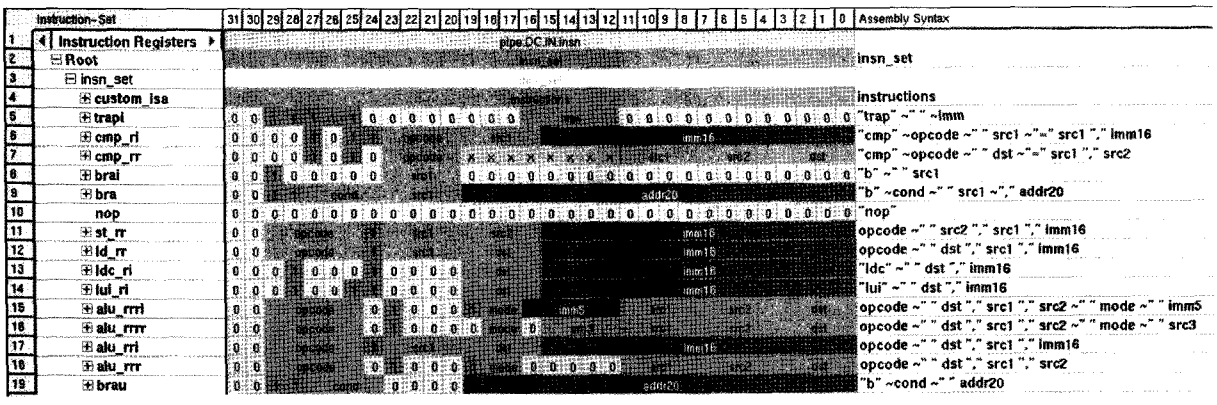

Fig. 7. Basic instruction-set

stage pipeline such as PF(PreFetch), FE(FEtch), DC(DeCode), EX(Execution), MEM(MEMory) and WB(WriteBack).

The Instruction-set of the proposed architecture consists of basic load/store, arithmetic, logic, branch and trap instructions, and custom instruction-sets which are specific to H.264 decoding as shown in Fig. 6 and Fig.7, respectively[8].

The LISA(Language for Instruction-Set Architecture) processor design platform offers the possibility to generate structured RTL model by grouping operations into functional units. Each functional unit in the LISA model represents an entity or a module in the HDL model. The generated verilog RTL code for application specific processor is synthesized using Synopsys Design Compiler based on TSMC cell library. Table 5 lists the performance of EMP.

Table 5. Performance of EMP. (CIF 3 frames)

Streams	Total Cycles
Foreman	44,018,569
Akiyo	33,041,968
Container	34,495,534
Table Tennis	34,447,175

V. Verification for hardware and software Platform

The top-down SoC design begins with the SoC requirement specification, followed by behavioral verification of SoC algorithms. However, analyzing the behavior of an ASIC may not be sufficient to detect all of the errors in the circuit. In such cases, the alternative is to use the register transfer level (RTL) of the gate-level description, which is more appropriate for pinpointing the errors related to the critical behavior of the SoC. We developed a design verification and methodology. Figure 5 shows a verification flow of H.264 SoC. It is from high level C to gate level simulation. We developed C models for major functional blocks of the H.264 video decoder, and the models performed high level simulation. Also, using hardware description language (HDL), we modeled an external environment made up of host interface and synchronous dynamic access memory (SDRAM). Simulation and testing for the software and hardware were carried out using a co-simulation environment. The test vectors for high-level simulation were used for verification from RTL-level HDL simulation through gate-level simulation. Before SoC fabrication, designed chips verified the board level using a field programmable gate array (FPGA), which is called logic emulation. Using test sequence files, RTL simulation obtained frame level verification, and board level simulation obtained 300-frame verification. We tested the board level for function level verification. The board consists of Xilinx FPGAs, an ARM9 chip, and testbench environments. The vertex3000 and vertex6000 series of Xilinx FPGAs were used. Gate level simulation was used for the NCVerilog tool.

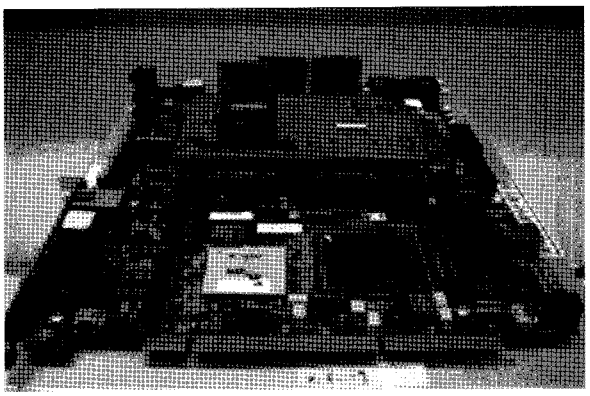

Fig. 8. Verification platform of H.264 encoder with Xilinx FPGA.

Implemented and verified the proposed H.264 encoder engine on Xilinx XC2V6000 FPGA. Test sequences were various file based ones and we also tested real time streams

with CMOS image sensor camera. We verified that JM S/W decoder and our having designed H/W decoder decode this encoded test streams. Fig.8 shows our developed H.264 encoder verification platform with Xilinx XC2V6000 FPGA. Fig.9 shows EMP hardware platform using Xilinx FPGA. The result of performance and gate count is 400MHz@30fps about 100k per core for H.264 decoder.

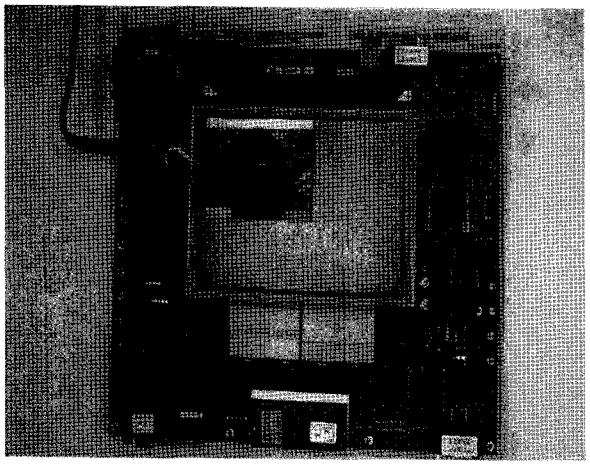

Fig. 9. EMP hardware platform

VI. Conclusions

We present a low-power design of H.264 Codec using dedicated[11] hardware and software. Design methodology is a top-down approach from algorithm level to architecture level. Algorithm level proposed to [12]parameters of H.264 encoder considering hardware cost as follows. The ideal scheme, we found is that search range is +/- 16.5, Hadamard is on, reference frame is one frame, block mode is 16x16, 16x8, 8x16, and 8x8, and rate distortion is off. On the architecture level, the design reduced a load of memory access scheme from external memory to internal memory for motion estimation reference frame windows. This scheme reduced computation load up to 66% compared to that of conventional methods. Gate Count of H.264 encoder is about 455k and performance is a 43Mhz@30fps with D1(720x480) for H.264 encoder. We designed EMP with ASIP that it [13]is SIMD, Dual Issue VLIW core, specified register file for SIMD, internal memory and data memory access for memory controller, 6 step pipeline, and 32 bits bus width. Performance and the gate count are 400MHz@30fps and about 100k per core for H.264 decoder with CIF.

References

[1] ISO/IEC 14496-10 International Standard (ITU-T Rec. H.264).

[2] Alexis Michael Tourapis, "Direct Mode Coding for Bipredictive Slices in the H.264 Standard," IEEE Trans. on Circuits Syst. for Video Technol., vol. 15, no. 1, Jan. 2005, pp. 119-126.

[3] Seongmo Park, A MPEG-4 video codec chip with low power scheme for mobile application, IEICE Transaction Fundamentals, Vol.E86-A, No.6, pp1353-1363,June.2003

[4] Seongmo Park, VLSI implementation of H.264 video decoder for mobile multimedia application, ETRI Journal,Vol.28, pp525-528.Aug.2006

[5] S. M. Park et al., "A Single-Chip Video/Audio Codec for Low Bit Rate Application," ETRI J., vol. 22, no. 1, Mar. 2000, pp. 20-29.

[6] Takashi, H. et al., "A 90mW MPEG4 Video Codec LSI with the Capability for Core Profile," ISSCC Digest of Technical Papers, Feb. 2001, pp.140-141.

[7] Jun-Young, et al., "A 100Mhz ASIP for CAVLD of H.264 decoder," ISCAS2008 Proceeding, May. 2008, pp. 3462-3465.

[8] Jae-Jin Lee, et al., "Design of Application Specific Processor and Compiler for H.264 CAVLC Decoding", ITC-CSCC 2008 Proceeding, July. 2008, pp. 189-192.

[9] Sukho Lee, et al., "A 40Mhz dedicated hardware

H.264/AVC video encoder with the reducing memory access scheme" ISCE 2008 Proceeding, April 2008.

약 력

Seongmo Park

Seongmo Park received the B.S, M.S., and Ph.D. degrees in electronics engineering from Kyungpook National University, Taegu, Korea, in 1985,1987, and 2006 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 was with the Electronics and Telecommunications Research Institute (ETRI), Taejon, Korea and joined in development of SoC design. He currently engaged in research on SoC design, image compression algorithm and SoC architecture design. He is now a team leader of mobile multimedia platform team and professor of University of Science and Technology. He has published over 30 technical papers throughout the international journals and conference proceedings. His main research interests are video coding, image compression, Multi Processor design and low power SoC architecture design.

Sukho Lee

Sukho Lee was born in Seoul Korea on July 12, 1969. He received the B.S degrees in electrical engineering from the Kyunghee University in 1995. He received the M.S degrees in electrical engineering from the Korea University in 1999. From 1995 to 1997, he was a VLSI design engineer in Samsung Electronics, Kihung Korea. He is currently a senior member of engineering staff in Electronics and Telecommunications Research Institute (ETRI), Daejeon Korea. His research interests are DSP, video coding, image compression and Multimedia SoC design.

KyoungSeon Shin

KyoungSeon Shin was born in Jeollanam-do, Korea on May 28, 1966. He received the B.S and M.S. degrees in electronics engineering from Chonbuk National University, Jeonju, Korea, in 1989 and 1991, respectively. From 1991 to 1999, he was with the LG semiconductor company, Cheongju, Korea, where he worked on MCU design and application. In 1999, he joined Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea, where he currently works in multimedia SoC platform development and multimedia SoC design.

Jae-Jin Lee

Jae-Jin Lee received the Ph.D. degree in computer engineering from Chungbuk National University in 2007. He is with MMP (Multimedia Mobile Platform) development team, ETRI (Electronics and Telecommunication Research Institute). His research interests include the areas of application-specific instruction-set processor and compiler.

약 력

Moo-Kyoung Chung

Moo-Kyoung Chung received the BS degree in Electrical Engineering from Korea University, Korea, in 1999, and the MS and PhD degrees in Electrical Engineering and Computer Science from KAIST, Korea, in 2001 and 2006, respectively. After graduation from KAIST, he worked at Dynalith Systems, Korea, from January 2006 to June 2007. In July 2007, he joined the Department of SoC Research at ETRI, where he is now a senior member of engineering staff. His current research interests include microprocessor/DSP/multiprocessor architecture, chip design and video processing.

Jun Young Lee

Jun Young Lee received the B.S degree in electronics engineering from Ajou University, Suwon, Korea in 2006. From 2006 to now, he is with the Electronics and Telecommunication Research Institute (ETRI) as a research associate and as a Ph.D candidate in University of Science and Technology in Korea. His research interests are in the areas of MPCore-SoC(Multi-core SoC) , multi-media codec and signal processing.

Nak-Woong Eum

Nak-Woong Eum received the MS and PhD degrees in Electrical Engineering and Computer Science from KAIST, Korea, in 1987 and 2001, respectively. In 1987, he was with ETRI, Taejon, Korea and joined in the field of electronic design automation and SoC design. He is now the director of SoC research department. His main research interests include Multi Processor design and low power SoC architecture design.