

# VLSI Implementation of H.264 Video Decoder for Mobile Multimedia Application

Seong Mo Park, Miyoung Lee, Seungchul Kim, Kyoung-Seon Shin, Igkyun Kim, Hanjin Cho, Heebum Jung, and Dukdong Lee

*ABSTRACT*—In this letter, we present a design of a single chip video decoder called advanced mobile video ASIC (A-MoVa) for mobile multimedia applications. This chip uses a mixed hardware/software architecture to improve both its performance and its flexibility. We designed the chip using a partition between the hardware and software blocks, and developed the architecture of an H.264 decoder based on the system-on-a-chip (SoC) platform. This chip contains 290,000 logic gates, 670,000 memory gates, and its size is 7.5 mm×7.5 mm (using 0.25 micron 4-layers metal CMOS technology).

*Keywords*—H.264, decoder, VLSI, hardware and software.

## I. Introduction

The new H.264 [1] video coding standard has gained more and more attention recently, mainly due to its higher coding efficiency over previous standards [2]. Digital signal processing (DSP) 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 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].

This letter presents an H.264 video decoder with hybrid architecture and cycle analysis. The proposed architecture achieves both high performance and high flexibility. Performance of the chip is indicated by 30 frames/s of the decoder in the common intermediate format (CIF) at 54 MHz.

Section II describes an architectural overview of the H.264 video decoder. In sections III and IV, cycle estimation in the H.264 video decoder is discussed. Conclusions are presented in section V.

## II. Architecture of H.264 Decoder

Figure 1 shows a block diagram of H.264. Synchronous DRAM, accumulated image data, and the microprocessor are controlled by the host parameters. The decoder consists of a

Manuscript received Jan. 19, 2006; revised Apr. 13, 2006.

Seong Mo Park (phone: +82 42 860 5203, email: smpark@etri.re.kr), Miyoung Lee (email: sharav@etri.re.kr), Seungchul Kim (email: skimc@etri.re.kr), Kyoung-Seon Shin (email: shinks@etri.re.kr), Igkyun Kim (email: ikkim@etri.re.kr), Hanjin Cho (email: hjcho@etri.re.kr), and Heebum Jung (email: hbjung@etri.re.kr) are with IT Convergence & Components Laboratory, ETRI, Daejeon, Korea.

Dukdong Lee (email: ddlee@ee.knu.ac.kr) is with the School of Electrical Engineering & Computer Science, Kyungpook National University, Daegu, Korea.

32-bit RISC processor and dedicated hardware engines. The dedicated engines are implemented for fixed functions in H.264, such as input stream control (ISC), entropy decoder, inverse quant (IQ), inverse transform (IT), reconstruction (REC), intra-frame prediction (IPRED), inter-frame prediction and motion compensation (MC) prediction, deblocking filter (DB), host interface (HIF), and clock generator (CLKGEN).

The local memories (LM) reduce accessibility to the external memory and power dissipation rate for real-time application.

The DMA controller can transfer data among local memories and access external memory. The RISC is a 32-bit ARM7TDMI processor with three-stage pipeline at the decoding mode. The RISC processor acts as a scheduler and high entropy decoder, and executes syntax synthesis and syntax parsing. Direct memory access and external memory interface blocks perform a data transfer between internal memory and external frame memory. The data transfer of the chip has one-cycle operation between local memory and external frame memory. The DMA supports several mode operations with

Advanced Microcontroller Bus Architecture, Advanced High-performance Bus specifications.

A DMA controller has special features. It can interface with all internal modules with only one channel that consist of programmable DMA and support both burst block mode and packet mode for data transfer. On the other hand, it has the architecture of a dual-addressed DMA without buffered memory. In a dual-addressed DMA transfer, the explicit addresses are required for selecting the right destinations.

Figure 2 shows the pipeline decoder operation at 30 frames per second. Decoder timing needs 33,333,333 ns per frame, and the macroblock cycle has 4,550 cycles (1,801,801/396 macroblocks) for CIF. Operations in real-time application need an extra cycle margin for system control and firmware.

Figure 3 shows a macroblock-level pipeline flow by the controller. The pipeline flow consists of three steps of decoding. Each stage must take less than 3,600 cycles in decoding in order to code 3,600 macroblocks in one second.

Figure 4 shows the performance of the H.264 decoder for the critical path. A configurable embedded processor has been recently developed by Tensilica. We evaluated cycles of the critical path for the H.264 decoder using ARM7, Tensilica, and hardware with and without DMA optimization.

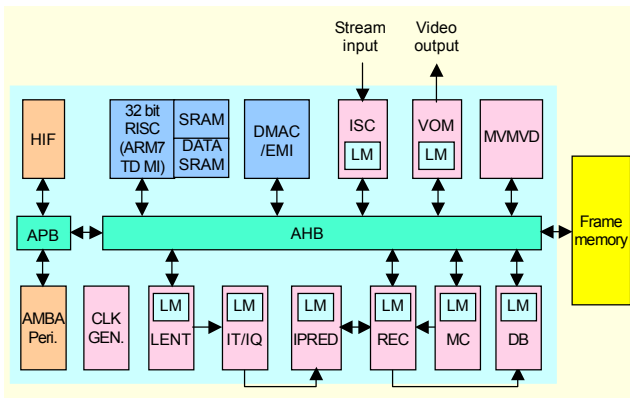


Fig. 1. Block diagram of the H.264 decoder.

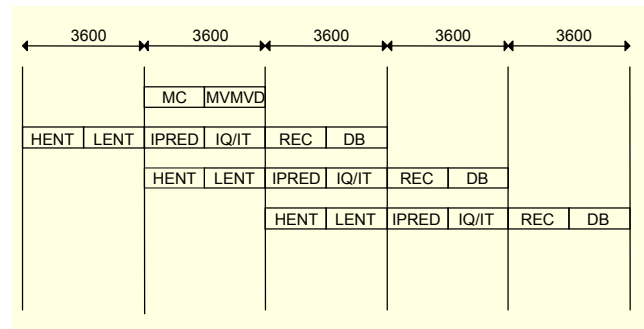


Fig. 3. Pipeline of the H.264 decoder.

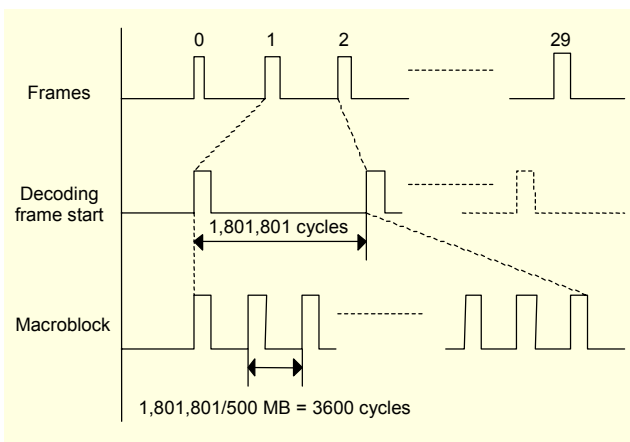


Fig. 2. Macroblock cycle of the H.264 decoder.

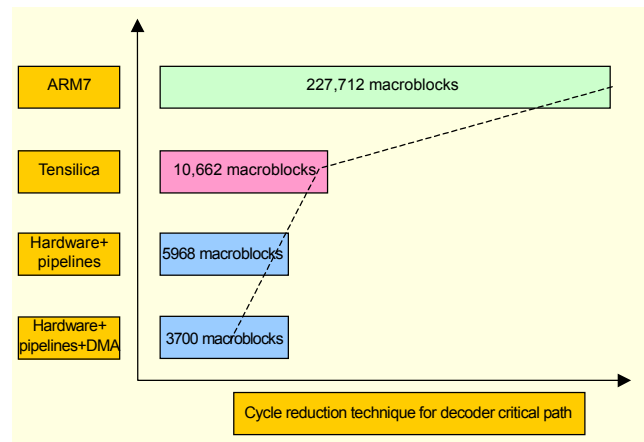


Fig. 4. Cycle reduction of critical path.

### III. Hardware Module Design

#### 1. Intra Prediction

For the luma signal, there are nine intra prediction modes labeled 0, 1, 2, 3, 4, 5, 6, 7, and 8 for vertical prediction, horizontal prediction, DC prediction, diagonal down/left prediction, diagonal down/right prediction, vertical-left prediction, horizontal-down prediction, vertical-right prediction, and horizontal-up prediction.

Examples of intra prediction for an Intra\_16 × 16 macroblock type luma block are vertical prediction, horizontal prediction, DC prediction, and plane prediction.

Prediction in intra coding of chroma blocks includes vertical prediction, horizontal prediction, DC prediction, and plane prediction.

#### 2. Integer Transform Inverse Quantization (ITIQ)

H.264 uses three transforms depending on the type of residual data that are coded in the bitstream: transformation for the 4 × 4 array of luma DC coefficients in intra macroblocks (predicted in 16 × 16 mode), transformation for the 2 × 2 array of chroma DC coefficients (in any macroblocks), and transformation for all other 4 × 4 blocks in the residual data. Therefore, for the implementation of ITIQ, the control flow should be dependent on the macroblocks and is more complex than ISO/IEC 13818-2 MPEG-4 IS (International Standard).

#### 3. Deblocking Filter

Conditional filtering shall be applied to all macroblocks of the pictures. This filtering is done on a macroblock basis, with macroblocks processed in raster-scan order throughout the picture. For luma, as the first step, the 16 samples of the four vertical edges of the 4 × 4 raster shall be filtered from the left edge to the right edge. Filtering of the four horizontal edges (vertical filtering) follows in the same manner, or from the top edge. The same ordering is applied to chroma filtering, with the exception that two edges for eight samples each are filtered in each direction. This process also affects the boundaries of the reconstructed macroblocks above and to the left of the current macroblock. Picture edges are not filtered.

#### 4. Motion Compensation

Motion compensation is divided into several macroblocks and 8 × 8 partitions. Accuracy of motion compensation is expressed in units of one quarter of the distance with luma and chroma pixels.

This chip designed using a hybrid architecture consisting of an ARM7TMI and dedicated hardware engine. This chip utilizes H.264 Base Profile Level 2 (BP@L2) support. It operates at 54 MHz. Table 1 shows the result of a synthesis

Table 1. Synthesis results.

	Gate count	Local memory
IPRED	32,190 gates	864 bits
DB	29,326 gates	5120 bits
REC	8,792 gates	3904 bits
LENT	52,688 gates	2176 bits
ITIQ	9,337 gates	7040 bits
MVMVD	29,817 gates	
MC	80,555 gates	4096 bits

with each module. Power consumption is 250 mW.

### IV. Verification Methodology

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

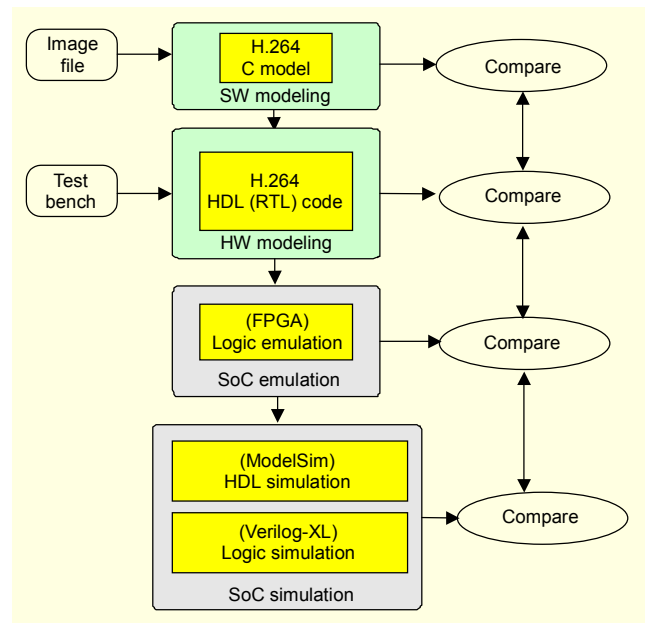


Fig. 5. Verification flow.

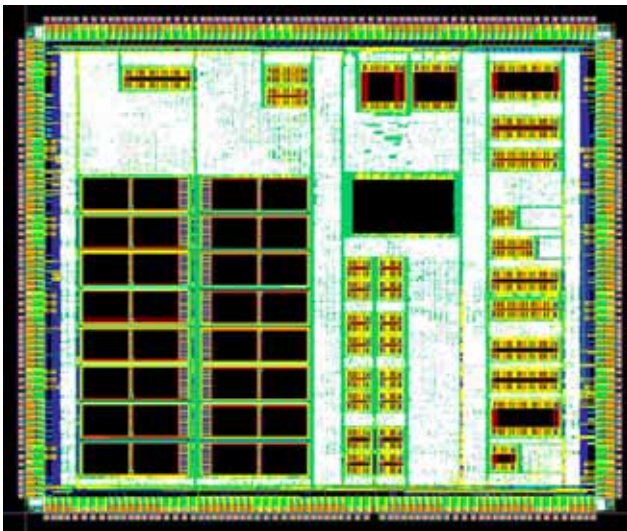


Fig. 6. H.264 chip layout photography.

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 3-frame 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 ARM7TDMI chip, and testbench environments. The vertex3000 and vertex6000 series of Xilinx FPGAs were used. Gate level simulation was used for the NCVerilog tool. Figure 6 shows a microphotograph of a H.264 chip.

## V. Conclusion

We implemented the design of a single chip video decoder for mobile multimedia application. This chip has a mixed hardware and software architecture to improve both performance and flexibility. It has low power consumption and high flexibility. We designed the chip using a partition between the hardware and software blocks. We developed the architecture of an H.264 decoder based on the SoC platform. This chip contains 290,000 logic gates and 670,000 memory gates. The chip size was 7.5 mm × 7.5 mm, and was fabricated using 0.25 micron 4-layer metal CMOS technology. The performance of the chip is indicated by 30 frames/s of the decoder in the common intermediate format (CIF) at 54 MHz.

## 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. Circuits Syst. for Video Technol.*, vol. 15, no. 1, Jan. 2005, pp. 119-126.
- [3] Iain E.G. Richardson, *H.264 and MPEG-4 Video Compression*, John Wiley & Sons, Jan. 2003, p. 270.
- [4] M. Horowitz, A. Joch, F.Kossentini, and A. Hallapuro, "H.264/AVC Baseline Profile Decoder Complexity Analysis," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 13, July 2003, pp. 704-716.
- [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.