Implementation of H.264/AVC Decoder for Mobile Video Applications

Suh Ho Lee, Ji Hwan Park, Seon Wook Kim, Sung Jea Ko and Suki Kim

Department of Electronics and Computer Engineering
Korea University
Seoul, Korea 136-701
Tel : 82-2-927-1582
Fax : 82-927-2-1582
e-mail : lsh@ulsi.korea.ac.kr

Abstract - This paper presents an H.264/AVC baseline profile decoder based on a SoC platform design methodology. The overall decoding throughput is increased by optimized software and a dedicated hardware accelerator. We minimize the number of bus accesses and use macroblock (MB) level pipeline processing techniques to achieve a real time operation. We implemented and verified a prototype on a SoC platform with a 32-bit RISC CPU core and FPGA module. Our design can process up to 20 frames/sec with QCIF_(176x144). The proposed architecture can be easily applied to many mobile video application areas such as a digital camera and a DMB (Digital Multimedia Broadcasting) phone.

I. Introduction

Recently, H.264/AVC, a new video compression standard, has been developed in partnership with ITU-T and ISO MPEG. The primary achievement is significantly improved coding performance in low and high bit rates as compared with previous coding standards such as H.263, MPEG-2, and MPEG-4 [1][2]. However, H.264/AVC needs large amount of computation, which makes it difficult to achieve real time processing in software manner. As one of the solutions, in this paper, we propose a hardware accelerator for H.264/AVC decoder. H.264/AVC decoder profiling shows that motion compensation, de-blocking filter, inverse integer transform and quantization consumes about 70% of total decoding time [3][4]. We implemented theses components in hardware. Also, we used a parallel processing architecture with more efficient data transmission scheme to achieve real time processing. To speed up inverse quantization, inverse integer transform, de-blocking filter and context adaptive variable length decoding (CAVLD) are implemented in hardware and inter/intra prediction blocks are realized in software. Our experiment shows that our proposed hardware accelerator delivers about 30% speed up over only software execution.

II. Proposed Architecture

For reusability and performance, the decoder includes one RISC CPU, some dedicated image processing hardware blocks, AHB bus interface blocks and internal bus blocks. The RISC CPU is responsible for consulting all other hardware units to make co-operation between software and hardware. The image processing blocks include Inter/Intra prediction unit, CAVLD unit, inverse quantization, inverse 4x4 integer transform unit, and de-blocking filter unit. The bus interface blocks include AHB master/slave units and direct memory access (DMA) unit. Our proposed H.264/AVC decoder is accessed by control and status registers via an AHB slave interface. For data transfer, the H.264/AVC decoder data from/to external memory buffers with minimum CPU intervention according to control register settings. Fig.1 shows our proposed H.264/AVC decoder architecture. With the new dedicated hardware accelerator, the decoding throughput is increased by about 30% on the average.

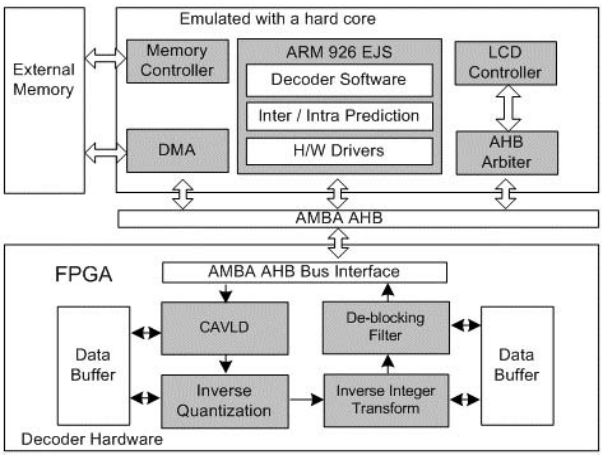

Fig.1. The proposed architecture of H.264/AVC decoder.

III. System Operation and Design

Decoding H.264/AVC video requires fast signal processing. Our designed hardware accelerators for H.264/AVC decoder consist of the following four components: CAVLD, IQ, IDCT and de-blocking filter. We have adopted MB level 4-stage pipeline architecture.

A. CAVLD

The whole CAVLD consist of the following parts: the input data buffer, coeff_token, level, total zeros, run of zeros decoder and VLC table. Together with a barrel shifter and controller, the pipeline architecture can CAVLD every syntax element in on clock cycle. The input data buffer of this CAVLD is to align the input bit stream for decoding the next code word.

0-7803-9451-8/06/$20.00 ©2006 IEEE.

B. IQ, IDCT

The block of IQ, IDCT that is composed of four main stages: input/output buffer stage, the inverse transform stage, the quantization processing stage and the quantization stage. MB data is initially stored in the register file. Then the 4x4 input is passed to the inverse transform block. This block consists of two cascaded sub-blocks. Each of them is responsible of multiplying two 4x4 matrices and is composed of four identical butterfly-adder blocks. Its operation is to perform a group of additions and shifts.

C. De-blocking filter

The architecture includes four components: low bs filter, high bs filter performs the horizontal/vertical filtering in a row-by row manner. 4x4 pixel data register buffers the intermediate results produced by low bs filter, high bs filter and acts as a transposed output buffer. Output buffer is used as a local buffer. Data flow control unit consist of a finite state machine which controls synchronization among low bs filter, high bs filter, 4x4 pixel data register and local output buffer.

IV. Experimental Results

Our design is implemented in the Verilog and synthesized using the Synopsys Design Compiler with Samsung 0.18um standard cell library. The results are shown in Table 1. Table 2 shows the encoder parameters for our designed decoder experiments and Table 3 shows the system performance comparison using the H.264/AVC decoder. In this system, the ARM926EJS CPU is running at 150MHz and the FPGA module is running at 50MHz. As compared to the software implementation, our performance gain mainly comes from the MB-level pipelining architecture and the hardware acceleration. Moreover, we verified our H.264/AVC baseline decoder using FPGA module. Fig.2 shows the photograph of our evaluation system.

TABLE I
Synthesis Results

Target Processing Capability	20 frames/sec @ 50 MHz QCIF(176x144)
Frequency	50 MHz
Power Consumption	113 mW
Functional Block	Gate Count
CAVLC decoder	12,900
IDCT/ Inverse Hadamard	7,380
Inverse Quantization	6,530
De-blocking Filter	24,580
Control Logic	3,400
Total Gate Counts	54,790

TABLE II
Encoder Parameters for Experiments

Frame Size	QCIF
Frame Rate	30fps
QP	I(28) P(31)
Reference Frame Number	5

TABLE III
System Performance Comparison Using H.264 Decoder at ARM926EJS

Sequnece	S/W	Our (S/W+H/W)	Throughput improvment
Foreman	7.56 fps	10.22 fps	35.18 %
News	11.40 fps	15.21 fps	33.42 %
Mother_daughter	13.12 fps	17.22 fps	31.25 %
Tennis	15.21 fps	20.15 fps	32.47 %

Fig.2. Photography of evaluation system.

V. Summary and Conclusions

In this paper, we have proposed a MB-level pipelining and our bus architecture to minimize the number of bus accesses in order to achieve a real time operation for a H.264/AVC baseline profile decoder. Our design methodology is based on task partitioning and scheduling in MB-level to enhance the overall decoding throughput. We control the hardware components in software and accelerate computationally intensive modules in hardware. The proposed MB-level pipelining improves the overall performance significantly. Our design could process up to 20 frames/sec with QCIF_(176x144), which is 30% more speed up than software only execution.

References

[1] Draft ITU-T Recommendation and Final Draft international Standards of Joint Video Specification (ITU-T Rec. H.264\ISO/IEC 14 496-10 AVC) Joint Video Team (JVT), Mar.2003, Doc. JVT-G050.

[2] Malavar, H.S., Hallapuro, A., Karczewicz, M., and Kerofsky, L.,"Low-complexity transform and quantization in H.264/AVC," IEEE Trans. Circuits Syst. Video Technol vol.13, no.7, pp.598-603, July 2003.

[3] T.Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp.560-576, July 2003.

[4] M. Horowitz, A. Joch, F. Kossentini, A. Hallapuro, "H.264/AVC baseline profile decoder complexity analysis," IEEE Trans. Circuits Syst. Video Technol, vol. 13, no. 7, pp.704-716, July 2003.