Design and Implementation of H.264-based Video Decoder for Digital Multimedia Broadcasting

Sung-Kyu Choi, Jong-Gu Jeon, Woo-Sung Shim, Won-Kap Jang, and Victor H. S. Ha,
Samsung Electronics Co., Ltd.
Digital Media R&D Center, Suwon, Korea
Tel: +82-31-200-3746, Fax: +82-31-200-3147
sim1@samsung.com

Abstract—In Korea, a nation-wide Digital Multimedia Broadcasting (DMB) service will be launched in year 2004 to provide high quality digital audio, video and data broadcasting to mobile, portable, and fixed receivers. We have developed the world’s first FPGA implementation of the portable DMB receiver complete with an RF receiver, a 6.4-inch LCD display, and audio/video/data decoders. In this paper, we present the real-time video decoder designed and implemented for this DMB receiver. The video coding in DMB is based on the newest international video coding standard known as MPEG-4 Part 10 AVC/H.264. We describe the low-power high-performance design of the H.264-based DMB video decoder and illustrate its performance based on the FPGA implementation.

I. INTRODUCTION

In Korea, a new national standard for the Digital Multimedia Broadcasting (DMB) service has been developed [1]. The Korean DMB system adopts as its base the European Digital Audio Broadcasting (DAB) system known as Eureka-147 [2]. The Korean DMB system adds to Eureka-147 an additional set of tools such as Reed Solomon (RS) coding, MPEG-2 Transport Stream (TS) packet, MPEG-4 SL packet, MPEG-4 Core2D@Level1, MPEG-4 Bit Sliced Arithmetic Coding (BSAC) audio codec, and MPEG-4 AVC/H.264 video codec to process multimedia contents at the overall bit error rate (BER) of 10^{-9}. The DMB is expected to provide high-quality digital broadcasting services to fixed, mobile, and portable receivers with a nation-wide coverage.

In this paper, we introduce the video decoder part of the Korean DMB receiver. The video coding in the DMB system is based on the newest international standard known as MPEG-4 Part 10 AVC/H.264 [3]. This new video coding standard improves coding efficiency over other existing video coding standards such as MPEG-2 and MPEG-4 Advanced Simple Profile (ASP). H.264 is also “network friendly” and has a large set of applications ranging from low bit-rate conversational services to high definition (HD) digital video broadcasting. The application of H.264 to the portable DMB video decoder raises a number of design issues such as low power consumption and high performance. In this paper, we focus on these issues and present our solutions in the design and implementation of the DMB video decoder.

The paper is organized as follows. In section II, we give a brief overview of the H.264 video coding standard and the DMB system in Korea. In section III, we introduce the H.264-based video decoder for DMB. We discuss the design issues and present the hardware architecture that reflects them. We also demonstrate the performance of the video decoder in terms of the average decoding time and the system’s operation clock frequency. Section IV summarizes the paper.

II. BACKGROUND

This section provides a brief overview of the H.264 video coding standard and the Korean Digital Multimedia Broadcasting (DMB) system. Readers are referred to [4], [5], [6] and references therein for further information.

A. H.264 Video Coding Standard

MPEG-4 Part 10 AVC/H.264 is the state-of-the-art international video coding standard developed by Joint Video Team (JVT). The new standard has recently been approved by ITU-T as Recommendation H.264 and by ISO/IEC as International Standard 14496-10 (MPEG-4 part 10) Advanced Video Coding (AVC). H.264 is composed of two main parts: (i) Video Coding Layer (VCL) that efficiently represents the video contents and (ii) Network Abstraction Layer (NAL) that provides “network friendliness”. H.264 significantly improves coding efficiency over prior video coding standards, but at an increased complexity.

In H.264, the overall gain in coding efficiency results from a plurality of small improvements in the Video Coding Layer (VCL). The important improvements in VCL are (a) enhanced motion-prediction, (b) 4 × 4 integer transform, (c) adaptive deblocking filter, and (d) enhanced entropy coding. The H.264 video codec enhances the motion-prediction by employing a new set of techniques such as variable block-size motion compensation, quarter-sample-accurate motion vectors, multiple reference pictures, and spatial prediction for intra coded pictures. Due to these highly sophisticated prediction techniques, the transform coding in H.264 is simplified to a 4 × 4 integer transform. This new transform is closely related to the Discrete Cosine Transform (DCT) but is implemented only with 16-bit integer arithmetic operations. The conditional application of a deblocking filter effectively removes the artifacts across block boundaries. Finally, the entropy coding is improved by introducing new context adaptive entropy coding tools such as Context Adaptive Variable Length Coding.
(CAVLC) and Context Adaptive Binary Arithmetic Coding (CABAC).

The goal of Network Abstraction Layer (NAL) is to provide “network friendliness” by formatting video contents and appending appropriate header information for the transport and storage of VCL processed video data. There are a number of highlighted features of NAL. The parameter set structure separates handling of important but infrequently changing information such as sequence parameter sets and picture parameter sets for robust and efficient conveyance of header information. NAL units provide a generic format for use in both packet-oriented and bitstream-oriented transport systems. Flexible slice sizes allow customized packaging of compressed video data appropriate for each specific network. A slice in H.264 is the smallest unit that can be encoded or decoded independently. A picture is composed of one or more slices. Other NAL-related techniques that provide robustness to data errors and losses include Flexible Macroblock Ordering (FMO), Arbitrary Slice Ordering (ASO), Redundant Pictures (RP), Slice Data Partitioning, and SP/SI pictures.

Both VCL and NAL of the H.264 video coding standard offer a wide selection of tools for the compression and transmission of video contents. Clearly, not all of these tools are needed in every implementation of H.264-based video coding systems. To deal with this issue, the notions of profiles and levels are introduced. Profiles specify a set of application-dependent algorithmic features to be supported by the decoder. Levels set limits on performance-related parameter values (maximum picture size, frame rate, etc.) corresponding to the processing and memory capabilities of the video coders and decoders.

B. Digital Multimedia Broadcasting in Korea

The Digital Multimedia Broadcasting is the next generation digital broadcasting service for indoor and outdoor users. When the DMB service is launched in year 2004, its users will enjoy CD-quality stereo audio services and real-time video/data streaming services anywhere in the nation while moving at the speed of 200 km/h. The Korean DMB system is based on Eureka-147, the European DAB system. To support digital video and data services, the DMB standard incorporates additional MPEG-2 and MPEG-4 tools including the following set of codecs:

- Video : MPEG-4 AVC (ISO/IEC 14496-10) / H.264
- Audio : MPEG-4 (ISO/IEC 14496-3) BSAC
- Data : MPEG-4 (ISO/IEC 14496-1) Core2D @ Level 1

In this section, we discuss the design and implementation of the real-time video decoder based on H.264. The video decoder is incorporated into the portable DMB receiver that has been verified and demonstrated successfully. The video decoder presented here is composed of about 520 thousand gates and decodes 30 frames per second at the operating clock frequency of 14.5 MHz. The section starts with a discussion on the design issues. Then, the hardware architecture is described. Performance of the video decoder is presented based on the FPGA implementation.

A. Design Issues

There are a number of design issues that are specific to the DMB video decoder. These issues, listed below, are reflected in our FPGA implementation of the DMB video decoder.

- Bandwidth: The DMB standard specifies the maximum bit-rate for transmitting a MPEG-2 TS packet at different protection levels. The video portion in the overall bit-rate ranges roughly from 500 Kbps to 1.5 Mbps. Considering the portability and mobility of the DMB receivers, the broadcasting service providers favor the “protection level 2-A” with the bit-rate of 572 Kbps allocated to the video service.
- Low power: The DMB video decoder is a portable device and requires a low power consumption level. In our design of the DMB video decoder, we achieve low power consumption on an architectural level by reducing the number of memory access and employing a low system clock frequency.
- High performance: The real-time operation of the video decoder at 30 frames per second requires that the decoder processes each picture within the maximum processing time of 33 msec. We minimize the number of bus access and use pipeline processing techniques to achieve a real-time operation at the lowest system/bus clock frequency possible.
B. Architecture

The goal in the design of the DMB video decoder is to come up with a low-power high-performance architecture. This goal is achieved in our implementation by operating the decoder at a low clock frequency. The main features of the architecture reflecting this are

- One Clock System
- Dual Bus System
- Two-Level Pipeline Processing

We discuss each of these features next.

1) One Clock System: The DMB video decoder is designed as a simple one-clock system. That is, the codec unit and the buses operate at the same clock frequency. This results in a dual-bus two-level pipeline architecture as discussed later. In Figure 1, we show an overall block diagram of the DMB video decoder. The functional blocks are divided largely into 3 parts—parser, decoder, and display parts.

The first group of blocks, the parser part, consists of the channel, RISC processor, and parser units that connect the RF channel to the SDRAM A on the A bus. The channel unit receives the RF transmission of the DMB video stream and stores it into SDRAM A connected to the A bus. The RISC processor unit then decodes the sequence parameter set, picture parameter set, and slice layer header information of the H.264 video stream while managing the system level control signals. The parser unit processes the remaining syntax elements at the slice data layer and below. The decoded syntax elements are written back to SDRAM A.

The second group of blocks, the decoder part, decodes the video stream stored in SDRAM A and writes the decoded pictures in SDRAM B. First, the entropy unit reads the CA VLC syntax elements from SDRAM A, generates an array of transform coefficients, and sends to the transform/quantization (T/Q) unit. The T/Q unit performs an inverse integer transform and dequantization to obtain the residue data. The prediction unit reconstructs the image block by computing intra/inter prediction values and adding them to the residue data from the T/Q unit. Finally, the deblock filter unit filters the reconstructed image blocks. The reconstructed images are stored into SDRAM B on the B bus.

The third group of blocks, the display part, performs the functions that are related to displaying the decoded and reconstructed video images to the LCD screen. Each unit connected to the SDRAM is equipped with a DMA (Direct Memory Access) unit. The bus arbiter units control the bus access requests and the SDRAM controller units manage the SDRAMs.

2) Dual Bus System: The DMB video decoder in Figure 1 is a dual-bus system. The two buses, A bus and B bus, are both 32-bit wide and operate at the same bus clock frequency. The parser and the decoder parts are connected to the A bus while the decoder and the display parts are connected to the B bus. During the decoding process, most of the bus access requests arise for the B bus. The traffic on the A bus is thus relatively lighter. Therefore, connecting the RISC processor to the A bus allows a stable operation of the processor running other system applications. In our implementation, the parser part is separated out from the rest of the decoder parts. The parser part communicates with the decoder part through SDRAM A connected to the A bus. This helps the decoder part to perform independently of the parser part that is affected strongly by the actual bit-rate of the incoming video stream. Since the A bus is not used as much as the B bus, the bus accesses to the A bus by the parser part are accommodated easily. Also, both buses can operate at a lower clock frequency by splitting the total bus traffic to two separate buses.

3) Two-Level Pipeline Processing: We employ a two-level pipeline architecture in the DMB video decoder. First, the parser part and the decoder part form a slice-level pipeline structure. Then, the decoder part operates under a macroblock-level pipeline structure. The pipeline architecture reduces the overall decoding time by removing the idle intervals in the parser and decoder parts. As a result, the video decoder can operate at a lower system clock frequency.

In the slice-level pipeline process, each of the parser and decoder parts becomes a pipeline stage. In the parser stage, the parser part decodes the incoming slice data and writes the resulting syntax elements to SDRAM A. In the decoder stage, the decoder part processes the syntax elements read from SDRAM A. The two pipeline stages are executed concurrently and thus SDRAM A maintains two slices of video data, one for each pipeline stage, during the decoding process. Each of the pipeline stages has a variable execution time. Thus, the pipeline processing is delayed slightly until the previous stage is completed.

The second level pipelining is carried out on a macroblock-level in the decoder part. Each tool unit of the decoder part does not have the same processing time. If the macroblock-level pipeline were not used, some tool units would stay idle, waiting for the previous tool unit to complete. We reduce these idle intervals by allowing each tool unit to start processing the next macroblock ahead of time while transmitting the already-processed macroblock data to the subsequent tool unit.
To realize this, each tool unit of the decoder part maintains internal buffering for two macroblocks. While one macroblock is being processed and written to an internal buffer, the macroblock data in the other buffer (that has already been processed) is transmitted to the subsequent tool unit.

C. Performance

The DMB video decoder has been built and tested on an FPGA with the ARM920T processor and an interface to a 6.4-inch LCD screen. The decoding process is divided into 3 steps: (1) software parsing, (2) hardware parsing, and (3) hardware decoding. In the software parsing step, the sequence parameter set, picture parameter set, and slice header information are parsed and entropy decoded by the ARM processor. In the hardware parsing step, the slice data layers including macroblock and sub-macroblock layers are parsed and entropy decoded by hardware. The resulting syntax elements values are written to SDRAM A. Finally, in the hardware decoding step, the syntax elements are processed by the CAVLC decoder, Transform & Quantization, Prediction, and Deblocking filter blocks. The resulting video pictures are stored in SDRAM B for display. Figure 2 shows the proportion of the decoding time needed by each of the three steps to process one slice. The decoding time of the hardware decoding step takes longer than both of the parsing steps summed together. That is, the parsing steps can complete parsing of the next slice before the decoding step finishes processing the current slice. We therefore compute the total decoding time by measuring the decoding time of the hardware decoding step.

The table below shows the average decoding time of the DMB video decoder. The real-time decoding operation has been verified using different sequences at the bit-rates ranging from 572 Kbps to 1.5 Mbps. In this paper, we show two sets of test sequences. The first set of sequences are generated at the bit-rate of 572 Kbps corresponding roughly to the highest bit-rate that the DMB standard allocates for the video stream. The video decoder was built based on the Baseline profile for the Digital Multimedia Broadcasting system in Korea. The portable DMB receiver performed successfully at the speed of 100 km/h.

IV. Conclusion

We presented the world’s first real-time video decoder for the Digital Multimedia Broadcasting system in Korea. The video decoder was built based on the Baseline profile of MPEG-4 Part10 AVC/H.264 at Level 1.3. Our FPGA implementation of the DMB video decoder consisted of 520K gates and operated at the low clock frequency of 14.5 ± 15.5 MHz.

The gate count of the DMB video decoder has been estimated using Samsung Semiconductor’s Standard Cell Library (STDL130) implemented in the 0.18 µm L18L process technology. The total count is estimated to be 520,000 gates. The DMB receiver and its video decoder presented here have been verified by a series of conformance tests. The tests were conducted in cooperation with the participating national broadcasting corporations in Korea. The portable DMB receiver performed successfully at the speed of 100 km/h.

V. Acknowledgement

Authors would like to thank the members of Mobile Solution Lab in the Digital Media R&D Center, Samsung Electronics and the members of ETRI (Electronics and Telecommunications Research Institute) in Korea.

References