

A Bandwidth and Memory Efficient MPEG-4 Shape Encoder

Kun-Bin Lee, Nelson Yen-Chung Chang, Hao-Yun Chin, Hui-Cheng Hsu, and Chein-Wei Jen

Department of Electronics Engineering

National Chiao Tung University

HsinChu, Taiwan, R.O.C., 30010

Tel : +886-3-5731925

Fax : +886-3-5710580

e-mail : {kblee, ycchang, hychin, huijane, cwjen}@twins.ee.nctu.edu.tw

Abstract - We have developed a shape encoder hardware for MPEG-4 video coding. On the one hand, the alpha component is compressed and therefore, the size and memory access of alpha frame memory can be reduced to 50% and 56.25% respectively. On the other hand, an efficient data transfer scheme combining the run length coding and addressing mode can reduce average data transfer time to 9.39% and accelerate the shape encoding process. The shape encoder can support MPEG-4 Main Profile at Level 4 in real-time. In addition, verification and testing methods are also considered.

I. Introduction

MPEG-4's object-based scene description allows the transmission of arbitrarily shaped video objects [1][2]. The purpose of using shape is to promote a better subjective picture quality, a higher coding efficiency as well as more user interaction. These advantages make this standard best suited for the needs of mobile applications or browsing multimedia databases on the Internet. Therefore shape coding can be utilized in lots of consumer electronics devices, such as video telephony, PDA, and video surveillance. However, these flexible and high-efficient coding features are based on the complex decision process and high computational tasks that demand the high computing and high data traffic properties. For example, the analysis of MPEG-4 Core Profile at Level 2 performed on a Ultra Sparc RISC indicated that MPEG-4 shape encoding requires giga scale operations and hundred mega byte scale memory access per second. To meet the stringent requirements on low cost and low power for the embedded system market, there is a clear need of the optimized VLSI architecture for MPEG-4 shape encoding.

Some works [3][4] have been presented on hardware designs of MPEG-4 shape encoder. In this work, we present a more efficient hardware design for the MPEG-4 shape encoder in terms of bandwidth and memory usage. In addition, the proposed design can also support higher level of MPEG-4 standard. We have successfully implemented the hardware shape encoder and ported the design on the ARM-based platform. The chip is fabricated using UMC 0.18 μ m CMOS 1P6M process. The design can support MPEG-4 Main Profile at Level 4, i.e., 489,600MB/sec with a frame size of 1920 \times 1088.

II. Shape Encoder Hardware Architecture

MPEG-4 video encoding is based on the video object plane (VOP) encoder. The alpha component of a VOP is encoded using a binary shape encoder while the color components are encoded

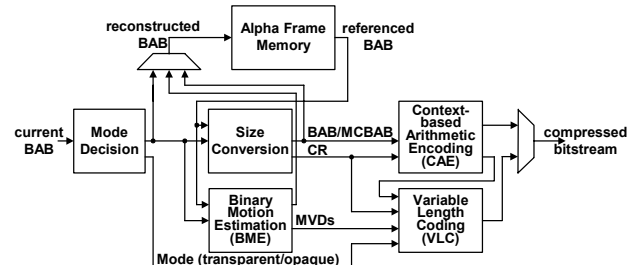


Fig. 1. Block diagram of shape coding system.

using motion estimation and compensation followed by DCT-based texture coding. The binary alpha data are grouped into what are called binary alpha blocks (BABs) to have the same dimensions as a macroblock, i.e., 16 \times 16. These BABs then undergo shape coding using a binary shape encoder. Fig. 1 shows the block diagram of our shape encoder design. In general, the procedure of binary shape coding can be as follows.

Step 1: Mode decision for boundary and non-boundary BAB. If the BAB is a non-boundary one, go to Step 5. In contrast, if the BAB is a boundary one, go to Step 2 or 3 for interframe or intraframe coding respectively.

Step 2: Perform BME to remove temporal redundancy. If a qualified motion vector is found, go to step 5. Otherwise, go to Step 3.

Step 3: Perform size conversion to obtain a subsampled version of current BAB under the predefined quality. Go to Step 4.

Step 4: Perform intra or inter mode CAE for intraframe or interframe coding respectively. Go to Step 5.

Step 5: Perform VLC for BAB coding mode and other related information.

Our shape encoder is designed to have VSIA VCI handshake protocol as its intrinsic interface and wrapped with ARM AMBA AHB interface. We have presented an efficient data transfer scheme that can significantly reduce data transfer time to 9.39% [5]. The reduced data transfer time can keep the shared system bus as free as possible, which in turn leads to increasing the potentialities of improvement on system performance. Additionally, this data transfer scheme also helps in the processing of both the mode decision for shape mode coding and the BAB class decision for index table access.

We have also presented a distributed tile-based memory organization for the alpha frame memory to efficiently fetch the required data and support the time-variant VOP sizes. This

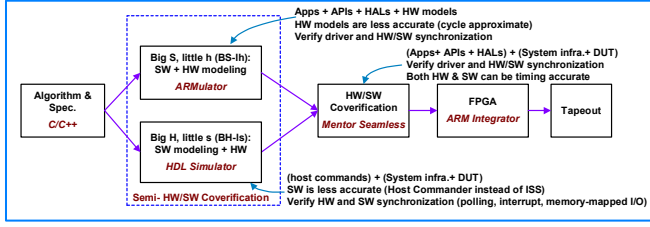


Fig.2. Design and verification flow.

memory organization makes both the memory access more efficient and the corresponding address generator more simple and flexible. Besides, we also demonstrated that a small, local index buffer can be used to reduce the size and the memory access of the alpha frame memory to 56.25% and 50% respectively in [5]. The proposed cost-effective multi-symbol CAE that encodes two symbols within each clock cycle without sacrificing the clock rate can achieve a speedup of 1.47 in comparison with traditional CAE architectures [6].

III. Experimental Results

Fig. 2 shows the design and verification flow of the shape encoder. The first stage is to determine the algorithm and spec. of the shape encoder by using Microsoft MPEG-4 verification C++ model. In addition, constrained random test patterns, such as patterns to cover the whole search range of motion vectors, are also generated at this stage. The second stage, semi-HW/SW co-verification, is split into two sub-stages to enable parallel design of software and hardware. At i big S, little h sub-stage, the major task is to verify software with less accurate hardware models. In this sub-stage, virtual prototyping of the ARM-based SoC system and the shape encoder hardware model written in C++ are built in ARMis ARMulator environment. The whole MPEG-4 program and the driver for shape encoder are developed at this sub-stage. At i big H, little s sub-stage, hardware design is exhaustively verified. In addition, the behavior of the driver, including the interrupt handling, is also modeled. Verilog RTL and gate-level simulations are verified using transaction-based and coverage-driven verification. The AMBA AHB protocol is verified by using in-house AHB bus functional model capable of interrupt handling [7] and qualified through having the 100% coverage report from Synopsys AMBA AHB monitor. Both statement and branch coverage are 100% under TransEDA Verification Navigator. The design is synthesized and inserted with three scan chains by using Synopsys DC Ultra and DFT Compiler respectively. The fault coverage is 95.5% when using 1,387 test patterns generated from TetraMax ATPG tool. In addition, test circuitry is added to enable the testing of the individual sub-module of the whole shape encoder. At the third stage, we integrate the shape encoder into ARM EASY platform with an ARM920T processor under Mentor Graphics' Seamless co-verification environment to complete HW/SW co-verification.

At the fourth stage, our design is implemented on the Xilinx XCV2000E FPGA of ARMis Logic Module and integrated with other modules of ARM Integrator rapid prototyping system, including an ARM920T, AHB sub-system, SDRAM sub-system, etc. The hardware resources utilization of FPGA is shown in TABLE I. The total equivalent gate count is 231,488. Finally, The design is implemented using the UMC 0.18μm CMOS

TABLE I. Xilinx XCV2000E FPGA Utilization @13.12 MHz

Resource	Available	Used	Resource	Available	Used
Slices	19,200	43%	Bonded IOBs	512	39%
Flip Flops	38,400	9%	Block RAMs	160	3%
4 Input LUTs	38,400	39%	GCLKs	4	25%

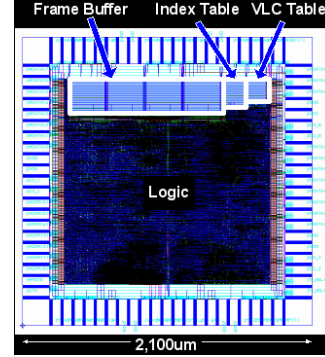


Fig. 3. Chip specification.

technology and the cell-based design flow. As shown in Fig. 3, the chip has an area of 2.1×2.1 mm² (pad limited.) The design can achieve 83 MHz at worst-case post-layout simulation. When running at 78.47 MHz, the design can support the performance of MPEG-4 Main Profile at Level 4, i.e., 489,600 MB/sec.

IV. Conclusions

We have developed a bandwidth and memory efficient MPEG-4 shape encoder with the performance capable of MPEG-4 Main Profile at Level 4. The size and memory access of alpha frame memory can be reduced to 50% and 56.25% respectively. In addition, the average data transfer time through system bus is reduced to 9.39%.

Acknowledgements

This work was supported by the National Science Council, Taiwan, R.O.C. under Grant NSC-91-2215-E-009-033. The shape encoder was fabricated by NSC Chip Implementation Center (CIC.) The authors express sincere thanks to the help of physical design from CIC.

References

- [1] ISO/IEC 14496-2, i Information technology 6 Coding of audio-visual objects, i 2nd edition, Switzerland, Dec. 2001.
- [2] N. Brady, i MPEG-4 standardized methods for the compression of arbitrarily shaped video objects, i *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 9, pp. 1170ñ1189, Dec. 1999.
- [3] D. Gong and Y. He, i Computation complexity analysis and VLSI architectures of shape coding for MPEG-4, i in *Proc. SPIE VCIPi2000*, vol. 4067, Jun. 2000, pp. 1459ñ1470.
- [4] Y.-C. Wang, H.-C. Chang and L.-G. Chen, i Efficient architecture of binary motion estimation for MPEG-4 shape coding, i in *Proc. SPIE VCIPi 2001*, San Jose, California, Jan. 2001.
- [5] K.-B. Lee, Nelson Y.-C. Chang, H.-Y. Chin, H.-C. Hsu, and C.-W. Jen, i Optimal frame memory and data transfer scheme for MPEG-4 shape coding, i in *Inter. Conf. on Consumer Electronics (ICCE)*, Los Angeles, CA, Jun. 2003, pp. 164-165.
- [6] K.-B. Lee, J.-Y. Lin and C.-W. Jen, i A multi-symbol context-based arithmetic coding architecture for MPEG-4 shape coding, i *IEEE Trans. Circuits Syst. Video Technol.*, accepted.
- [7] K.-B. Lee, H.-L. Wu, Weber Chen, and C.-W. Jen, i Design alternatives for testbench authoring, i in *SNUGi03*, Taiwan.