

JPEG, MPEG-4, and H.264 Codec IP Development

Chung-Jr Lian, Yu-Wen Huang, Hung-Chi Fang, Yung-Chi Chang and Liang-Gee Chen
Graduate Institute of Electronics Engineering and
Department of Electrical Engineering
National Taiwan University, Taipei 10617, Taiwan, R.O.C.
lgchen@cc.ee.ntu.edu.tw

Abstract

This paper summarizes our design experiences of various image and video codec IPs. The design issues and methodology of custom video codecs are discussed. The design methodology can be summarized as four stages, system analysis, algorithm optimization, architecture exploration, and code development. Based on these guidelines, several design cases are presented, including the proposed JPEG, MPEG-4, and H.264 architectures.

1. Introduction

Image and video codec IPs play an important role in today's highly demanding multimedia appliances. Most codecs are implemented as dedicated architectures for the high computational complexity under real-time constraints. To have a high performance and cost efficient architecture, designers must have an insightful understanding of the characteristics of video data and coding algorithms first, and then apply architecture design techniques to achieve highly parallel designs with smooth data flow and high hardware utilization. In the following sections, design methodology is discussed, followed by some design cases and a conclusion.

2. Design Methodology

The design methodology discussed here are partitioned into four stages:

- 1) system analysis,
- 2) algorithm optimization,
- 3) architecture exploration, and
- 4) design coding and verification.

System analysis is the first step to identify the critical problem of the system under design. Profiling tools are used for complexity analysis, characteristics understanding, and

bottleneck identification. In a codec system, some modules are computation-intensive, while others are control-intensive. The bottleneck may be computation, memory size, or bandwidth. For video encoders, the profiling data show the bottleneck is the motion estimation (ME). This module is therefore always implemented as a highly parallelized array processor with carefully designed I/O considerations and local buffer allocation. Data reuse techniques can be applied further to reduce memory bandwidth and share computations. As for the bitstream parsing in a decoder, the characteristic is bit-wise processing. Though it is not computationally complicated, a custom architecture is necessary for efficient bit-level operations. Based on the analysis, the design goal is to map each module in a codec to an efficient processing element architecture.

Hardware-oriented optimization at algorithmic level is crucial in architecture design. The optimization at a higher level always has a greater impact on the entire system. Classic examples are the discrete cosine transform (DCT) and fast ME algorithm optimization considering the hardware costs, processing speeds, and power issues. Besides, hardware-feasibility is another issue in some modules, since some software-based algorithms, such as recursive processing, may not be suitable for dedicated implementations and need to be modified.

There are various methodologies and techniques [1][2] in mapping algorithms to hardware architectures. An architecture is highly related to design specifications, such as area, speed, power, and functions to be provided. Due to the tough real-time constraint, pipelining and parallelizing are the most frequently used techniques in codec designs. Inherent parallelism in an algorithm is extracted and efficiently mapped to multiple processing elements. System pipelining and scheduling must be carefully designed to minimize the inter-module buffer size and increase the hardware utilization.

Coding rules and simulation approaches are important in the Verilog code development stage. Disciplined coding styles help prevent inconsistencies between pre- and

post-synthesis, and are also beneficial for design maintenance. Commercial source code linting tools are used for our code checking. During the code development stage, large amount of simulations are necessary for various parameters and conditions. Fast simulation and efficient error diagnosis are the key to shorten the development time. FPGA-PC co-simulation can help speed up the intermediate verification of each module, and it is also a platform for final emulation and demo of the entire system.

3. Proposed Codec IPs

In this section, design results and experiences of several codec IPs are presented, including JPEG, MPEG-4, and H.264.

JPEG is widely used in digital imaging applications and video surveillance systems. Digital still camera (DSC) is the most typical application. The proposed hardwired JPEG engine [3][4] can easily support both high speed still image and motion-JPEG processing at a very low clock frequency. The most computation-intensive module, DCT/IDCT, is based on a compact row-column decomposition architecture. The other modules are designed to be cascaded seamlessly such that no extra buffer is required for inter-module data flow smoothing. It is because the fully pipelined smooth data flow, high throughput and compact design are achieved.

A codec IP is usually expected to be a stand-alone processor. In this case, the master processor only has to fire up the IP, and then wait for output data ready. The proposed JPEG engine meets the requirement and is an entirely custom design supporting complete JPEG coding and decoding, including file syntax handling, bit-packing of variable length codes, and Huffman decoding for user-defined tables. Experiences show that although these tasks are not so computation-critical as shown in software run-time profile, they usually need more design effort in coding and debugging than the DCT/IDCT module, which is with regular processing elements and easy control.

In our MPEG-4 encoder [5], the platform-based approach is adopted for the system architecture. The prototype supports real-time encoding of MPEG-4 Simple Profile Level 3 at 40 MHz. The system mainly consists of a RISC processor, an embedded SRAM, a DMA unit, a memory interface, wrappers for dedicated units, two signal buses, and dedicated accelerators of ME/MC, block engine, and variable length coder (VLC). The JPEG design experiences of modules such as DCT/IDCT, quantization/inverse quantization and VLC can be transferred to the MPEG-4 design. However, a poor scheduling of modules in the coding loop of MPEG-4 will involve large buffer and cost. Therefore, an interleaving DCT/IDCT scheduling is proposed. For the decoder, a programmable bitstream processor [6] is pro-

posed to efficiently handle bit-level tasks.

Emerging H.264 is much more complicated than all previous standards. The computational load is higher, and there are many modes to be processed and then selected. The optimal data flow and pipelining stages are therefore different from previous MPEG algorithms. After analysis, a four-stage macroblock pipelining architecture [7] is proposed. The four stages are integer motion estimation, fractional motion estimation, intra prediction engine, and entropy coding and de-blocking engines. The Lagrangian mode decision is also optimized for dedicated hardware feasibility. The processing capability is HDTV/720p 30 frames/s with one reference frame and $H \pm 64/V \pm 32$ at 108 MHz.

4. Conclusion

In this paper, image and video codec IP design experiences are given. Design concepts, algorithm analysis and optimization, parallelism exploration, and efficient mapping are discussed. FPGA development platform is adopted to cope with the high verification complexity of codec IP designs. By following the design methodology, many high performance image and video codec IPs have been developed and successfully transferred to third party for mass production.

References

- [1] S. Y. Kung. *VLSI Array Processors*. Prentice-Hall, Englewood Cliffs, NJ, 1998.
- [2] K. K. Parhi. *VLSI Digital Signal Processing Systems: Design and Implementation*. Wiley-Interscience, New York, NY, 1999.
- [3] C.-J. Lian, L.-G. Chen, H.-C. Chang, and Y.-C. Chang. Design and implementation of JPEG encoder IP core. In *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC'01)*, pages 29–30, Yokohama, Japan, Jan. 2001.
- [4] C.-J. Lian, H.-C. Chang, K.-F. Chen, and L.-G. Chen. A JPEG decoder IP core supporting user-defined Huffman table decoding. In *Proc. International Symposium on Integrated Circuits, Devices and Systems (ISIC'01)*, pages 497–500, Singapore, Sept. 2001.
- [5] Y.-C. Chang, W.-M. Chao, C.-W. Hsu, and L.-G. Chen. Platform-based MPEG-4 SOC design for video communication. *Journal of VLSI Signal Processing Systems*, submitted for publication.
- [6] Y.-C. Chang, C.-C. Huang, W.-M. Chao, and L.-G. Chen. An efficient embedded bitstream parsing processor for MPEG-4 video decoding system. *Journal of VLSI Signal Processing Systems*, submitted for publication.
- [7] Y.-W. Huang, T.-C. Chen, C.-H. Tsai, C.-Y. Chen, T.-W. Chen, C.-S. Chen, C.-F. Shen, S.-Y. Ma, T.-C. Wang, B.-Y. Hsieh, H.-C. Fang, and L.-G. Chen. A 1.3TOPS H.264/AVC single-chip encoder for HDTV applications. In *Proc. IEEE International Solid-State Circuits Conference (ISSCC'05)*, San Francisco, California, USA, Feb. 2005.