Extended Abstracts of the 2005 International Conference on Solid State Devices and Materials, Kobe, 2005, pp.312-313

D-6-1 (Invited)

Design and Architecture Exploration for Image and Video Coding Systems

Chao-Tsung Huang and Liang-Gee Chen

DSP/IC Design Lab, Graduate Institute of Electronics Engineering and
Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
E-mail:{cthuang, lgchen}@video.ee.ntu.edu.tw

1. Introduction

Low cost and low power hardware with sufficiently high performance is extremely essential for image and video coding applications to be popular. Thus, efficient hardware implementations in VLSI are of vital importance. However, image and video algorithms usually require very high computational complexity. In this paper, the design methodologies for existing and future hardware architectures of image and video coding are explored. The general design methodologies will be presented first, which include computational analysis, data access analysis, and rate-distortion-complexity analysis. Then, hardware architecture design concepts, including system architectures and module design, will be illustrated by use of a real design example for H.264/AVC encoder. Last of all, new challenges and possible future directions for image and video coding implementation will be discussed, which mainly come from rate-constrained coding flow, open-loop temporal prediction, and scalable coding.

2. General Design Methodologies

The optimization of hardware design for image and video coding systems can be achieved by considering the two different levels of architecture design: system design and module design. The former decides the whole system architecture and the relationship between modules. The latter is to optimize each module according to the allocated resource and constraints.

System design

The system architecture is usually designed by use of computation analysis and data access analysis that consider computing and data issues, respectively. The computation analysis is to classify the coding tools in the adopted coding algorithm into different level computational characteristics and choose the suitable implementation types, which includes characteristic and complexity analysis.

Fig. 1 shows the computational characteristic analysis, which categorizes computation into three different levels of operations. The low-level operation represents highly regular computation and predictable computational flow. It is suitable to be implemented by dedicated hardware because its complexity is usually very high and the regular computation can be accelerated via parallel processing. The high-level operation represents highly irregular computation and unpredictable computational flow. However, its complexity is usually much lower than the low-level operation. Thus, it is suitable to be implemented using programmable design. Between these two extremes, the medium-level operation is preferred to be implemented by use of configurable architecture that can be on-the-fly adapted according to data-dependent decisions.

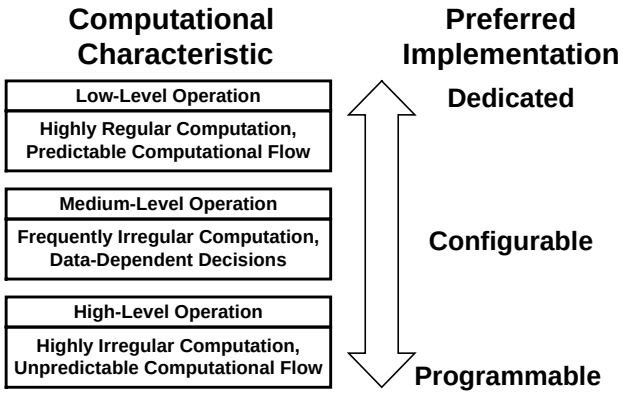

Fig. 1 Computational characteristic analysis and the corresponding preferred implementation.

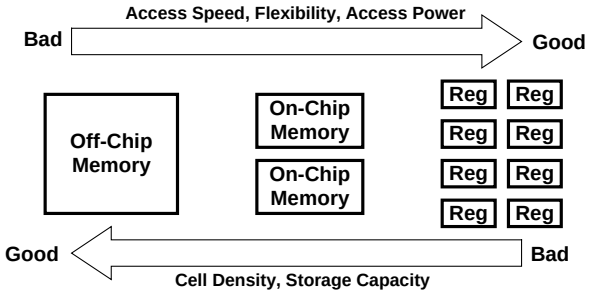

Fig. 2 General memory hierarchy: off-chip memory, on-chip memory, and registers.

The data access analysis is used to decide how data are transferred in the system. It analyzes how data should be stored for access and how data are transferred between modules. The storage and access issue is to optimize and balance the adopted memory hierarchy as shown in Fig. 2. Off-chip memory can provide highest cell density and capacity but suffers larger access power and lower access speed. On-chip memory is usually used as a cache to reduce the access of off-chip memory, but it may occupy significant chip area. Registers can be used for the fastest and most flexible storage elements. On the other hand, the interconnect issue is to decide how to allocate global bus and dedicated connection. The global bus provides flexible configuration and saves the interconnect area. The dedicated interconnect is to provide high throughput and efficient communication between highly related modules.

Module design

After the system architecture is defined, every module can be designed by use of algorithm-level and architecture-level optimization. The algorithm-level optimization is mainly to optimize the rate-distortion quality under given complexity constraints, or vice versa. More computation power results in better visual quality. The architecture-level optimization is to perform data flow smoothing and to balance scheduling and timing control for determined algorithms.

3. Case Study - H.264/AVC Encoder

A single-chip design for H.264/AVC encoder [1] is taken as an example to embody the design methodology. By computational characteristic analysis, the H.264 encoder system can be separated into four different levels of operations: low-level Integer Motion Estimation (IME), mixed low- and medium-level Fractional Motion Estimation (FME), medium-level Intra Prediction (IP), and high-level entropy coding and deblocking filter. The system is designed as a four-stage macroblock pipelining architecture accordingly. Due to the huge amount of data access of H.264 encoding algorithm, many on-chip memories are used to reduce off-chip memory access. Besides, interconnects between modules are decided to be global or dedicated according to the data transmission amount and regularity.

For the low-level and computation-hungry IME design, three algorithm-level techniques are applied to reduce the complexity but still maintain the superior encoding performance after rate-distortion-complexity analysis is evaluated. They are 1/2 computation subsampling, pixel truncation, and adaptive moving window. The architecture-level optimization is achieved by array parallel processing, snake scan data flow, and reuse of overlapped search area. The first one is for real-time processing of HDTV sequences. The second one can make the processing array fully utilized and provide low on-chip memory bandwidth. The third one is adopted to save 80% on-chip memory area and 87.5% off-chip memory bandwidth.

For the medium-level IP design, partial distortion elimination algorithm is applied to reduce the computation complexity. As for the architecture-level optimization, four-parallel reconfigurable computing elements are designed for resource sharing of thirteen intra prediction modes. The 4x4 and 16x16 interleaving schedule is used to eliminate the processing bubble cycles for higher hardware utilization.

4. New Design Challenges

Although hardware design for image and video coding has been developed more than one decade, some new design challenges still exist because of higher coding performance and demanded scalability. In the following, three new design challenges are introduced: rate-constrained coding flow, open-loop temporal prediction, and scalable coding.

For increasing coding performance, the rate-distortion optimization is becoming more and more important for image and video coding systems. Especially, JPEG2000 Tier-2 rate-distortion optimization and rate-constrained mode decision used in H.264 reference software are well-known techniques to boost the coding gain. However, The rate-constrained coding flow results in a sequential processing nature, which indeed becomes a problem for parallel processing. Good algorithm-level modification of the rate-constrained coding flow is required for efficient hardware implementation.

The hybrid texture and motion-compensated scheme becomes the mainstream of video coding algorithm development and international standardization in this decade. However, the close-loop prediction scheme is hard to provide efficient scalable coding because there will be a serious drift error if any mismatch occurs between encoder and decoder. Motion-Compensated Temporal Filtering (MCTF) is a recent breakthrough of video coding scheme, which breaks the close loop for efficient scalable coding. The Scalable Video Coding [2] is currently under MPEG standardization process, which adopts MCTF as the core structure. The open-loop temporal prediction scheme requires a group-of-picture (GOP) level operation. It results in longer coding delay, larger off-chip storage, and higher off-chip memory bandwidth.

Besides higher coding performance, the scalability is also demanded by many multimedia applications. The scalable entropy coding requires higher flexibility and more data access than fixed data rate coding. For example, JPEG2000 can provide spatial and quality scalable bitstream, and the reference software performs rate-distortion optimization in an image-level access. These two requirements will make hardware acceleration more difficult without any algorithm-level modification.

5. Conclusions

In this paper, general design methodologies for image and video coding systems are explored in two directions: system and module. Following these methodologies, efficient hardware implementation can be derived systematically. The case study of H.264 encoder chip design is given to illustrate each step of design methodologies. Three new design challenges are also presented according to current development of image and video coding systems.

References

[1] Yu-Wen Huang and *et. al.*, "A 1.3TOPS H.264/AVC Single-Chip Encoder for HDTV Applications," in *IEEE International Solid-State Circuits Conference*, 2005, pp. 128-129.

[2] J. Reichel, H. Schwarz, and M. Wien, "Working Draft 1.0 of 14496-10:200x/AMD1 Scalable Video Coding," ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Doc. N6901, Jan. 2005.