博士論文概要

論文題目
Studies on Hardware/Software Codesign Methodology for Video Encoders.
ビデオエンコーダを対象とした
ハードウェア/ソフトウェア協調設計方法に関する研究。

申請者
崔錦求
Jinku Choi

電子・情報通信学専攻 システムLSI研究

2003年1月
Advances in semiconductor technology are playing a fundamental role in multimedia technologies, and have resulted in highly integrated, more complex large scale integration (LSI) systems. Although system design is complicated, time consuming, and expensive, market demands now require a shorter time-to-market, lower costs, lower power consumption, and higher throughput.

These systems require efficient design within the hardware and software domain for optimization of the overall system. System design has concentrated on embedded systems. Applications' designs impose constraints on the hardware and software components of a system and involve complex tradeoffs among design metrics.

Design productivity and quality are becoming increasingly important as complex new applications are associated with more difficult system design. New system design methods are needed, both to deal with such complex systems and to meet market demands.

Hardware/software codesign is emerging as a promising approach to deal with these challenges.

The codesign approach involves the concurrent design of hardware and software, which is tightly coupled throughout the design process. One of the goals of codesign is to shorten the time-to-market, low cost, and achieve better designs and products, while reducing the effort and cost of designing a product to meet strict design constraints.

One of the important aspects of this approach is to find the best system architecture, including the right partition between the hardware and software components.

The video coding is a core technology in multimedia applications. The objective of video coding is to optimize video quality at or over a given bit rate, yielding optimum quality under complexity constraints. Quality and complexity are both important issues in designing video applications, and designers have to make tradeoffs between these parameters.

In this thesis, we firstly proposed algorithm and hardware architecture for the motion estimation, and then completed video encoder with hardware/software codesign approach.

We presented an optimization method of a video encoder based on hardware/software codesign and architecture template.

In chapter 1, we gave the background, motivation, and overview of our research.

In chapter 2, we proposed a novel motion estimation algorithm and hardware architecture. In general, video coding technologies take advantage of data redundancies in order to reduce the complexity.

Motion estimation is widely used to reduce temporal redundancy that may exist within a video sequences. It has a marked effect on performance, and is therefore an important component of video encoding systems. However, motion estimation introduces a great deal of computational complexity into the video encoding process, and usually requires specialized hardware. It therefore becomes a bottleneck in real-time video applications.
Our proposed algorithm determines motion type, whether small or large motion and then selects adapted searching pattern between CBS (Center-Biased Search) and NCBS (Noncenter-Biased Search) for different kinds of motion sequences.

The motion type is determined by comparing the motion vectors measure of each macroblock.

We also proposed CBS and NCBS search strategy. The CBS performs better in center-biased images or in searching small motion, whereas the NCBS gives better performance when searching large motion or noncenter-biased images.

We analyzed the performance of the algorithms and presented adapted algorithm. Our proposed algorithm improves performance and reduces computation complexity by combining the ability of the noncenter-biased algorithm to search for large motion sequences with the center-biased nature and search speed of the center-biased algorithm.

The quality of our algorithm is better than that of the TSS and the BBGDS algorithm, or comparable to the performance of the better of the two algorithms, and the computational complexity of our algorithm is significantly less than that of the TSS algorithm.

We also proposed hardware architecture for realizing our algorithm. We implemented the flexible hardware architecture by using address generator unit, delay unit, and parameters and by using the hardware description language (VHDL) and the SYNOPSYS synthesis design tools.

Our architecture is applicable to objects coding and low cost real-time video encoding applications.

In chapter 3, we implemented an MPEG-4 encoder with hardware/software codesign approach.

We presented design strategies for two types of solutions, a hardware implementation and software implementation. We dealt first with the fundament concepts of digital video, image and video compression, MPEG-4 standards for video coding, and the key components of video encoder in detail.

As with many aspect of video encoder hardware/software design, there are usually tradeoffs between algorithm and computational complexity, quality. The relationship between algorithm, computational complexity and quality is discussed.

We described efficient and effective feedback between algorithm and architecture design, and provided the tradeoffs that can be effectively analyzed at a pre-implementation level before decisions have to be made.

In our hardware/software codesign approach, we firstly decomposed functional module from design specifications. We searched bottleneck module in the video encoder, and therefore focused on this bottleneck module to optimize and improve the video encoder.

Typically, there are several algorithms for the implementation of a function module in the video encoder.

In addition, there are different architectures for a single algorithm. When an algorithm is found, the architecture is determined and optimized, and the algorithm is transformed into hardware. Designers must make many decisions concerning the algorithm, system architecture, and whether and how each
component is implemented, as hardware or as software.

Our approach provided the effective decisions and presented an optimization method for an MPEG-4 encoder.

In chapter 4, we described the optimization of an MPEG-4 encoder based on hardware/software codesign methodology and system architecture template. We implemented the MPEG-4 encoder using hardware and software based codesign flow, and searched for the bottleneck module constraining the system; this was the motion estimation module, which we focus on it. After investigating the computational complexity and quality of the well-known motion estimation algorithms and the simplicity of hardware implementation, we chose the best algorithm among motion estimation algorithms and implemented it with hardware, then optimized. Various hardware architectures can be used to implement an algorithm. We based our implementation on a parallel PE array for optimization of hardware architecture.

We presented relations the performance, required memory bandwidth, and chip area according to the number of PEs.
In this manner, we effectively optimized the motion estimation module and overall system.

We proposed an architecture template which is based on RISC processor, the AMBA bus, shared memory, and hardware functional modules with local memory. Using an architecture template helps to reduce design costs and shorten the design time by reducing design decisions.

The important point in the system design process is the exact effect of design decisions on the requirement of each design process step. The decisions must be based on quantitative simulations, estimations, and evaluation results. However, design decisions are often inefficient or unoptimized.

We presented an implementation method for an MPEG-4 encoder based on codesign of the system algorithm and architecture using the system architecture template. We mapped the MPEG-4 encoder onto target architecture template and calculated the total number of instruction cycle per macroblock and the chip area for each module.
Our approach can be easily expanded to other image applications and multi-standard video encoder systems, and provides a flexible, scalable system using a standard system bus.
In this manner, we effectively optimized the motion estimation module and overall system.
The evaluations resulted in effective optimization of the motion estimation module and better tradeoffs that optimized the overall video encoder system.

Chapter 5 presented a discussion of the results obtained using the proposed algorithm and architecture for motion estimation, implemented an MPEG-4 encoder, codesign methodology, and conclusions of the thesis.