早稲田大学大学院情報生産システム研究科

# 博 士 論 文 概 要

## 論 文 題 目

# SIMD Based Multicore Processor for Image and Video Processing

申 請 者
HE, Xun

情報生産システム工学専攻
マルチメディアシステム研究

2012 年 2 月

Continuous improvements in image and video processing require high computational power to deal with the increasing complexity of algorithms and higher definition video. Meanwhile, development in VLSI technology allows the integration of more cores on a single chip for achieving higher performance. There are three levels of parallelism in applications: task level, data level and instruction level. Typically, multicore architecture can obtain high task level parallelism, SIMD can achieve high data level parallelism, and VLIW architecture can achieve high instruction level parallelism. However, hardware costs and performance gains of the three methods are quite different for video processing. Multicore can achieve almost double performance gains with double hardware costs for a video encoder. SIMD can achieve double data level parallelism with less than double gates. VLIW needs more than double gates for double instruction level parallelism. Based on these technologies, we design and fabricate a 32 cores processor, and also evaluate its performance. The results show that this processor can achieve very high performance for our target applications.

**Chapter 1 [Introduction]** presents a background introduction of parallel architectures and challenges in multicore processor.

**Chapter 2 [SIMD based Core Architecture]** presents the SIMD based core architecture. For maximizing on-chip parallelism, multicore and 128 bits SIMD architecture is applied. SIMD instruction set is optimized for multimedia applications. The core is consisted by one small 32 bits RISC core, and SIMD pipelines. RISC is based on open source project: OR1200. SIMD is divided into two parts for executing two instructions at the same

time. Comparing with CELL processor's SIMD, the proposed work can reduce by 29% cycle count for video applications.

**Chapter 3 [Application Specified Cache Coherence Protocol]** proposes a manually controlled invalid cache coherence protocol (MCI). In multicore system, the cache coherence problem becomes more and more important with core number. In conventional snooping based protocol, coherence transaction broadcasts to all cache monitors. M. Ekman's Experiments in WDDD2002 show that more than 70% percentage of all snooping operations misses in other caches for conventional MOESI protocol. This means that most of the snooping induced tag-lookups just waste the power. In the proposed MCI protocol, memory space can be dynamically or statically defined as shared or private space. The shared spaces can be shared by all cores, clusters or several cores. A new programming model is proposed for defining the sharing patterns. A snooping unit is added for each core, which keeps the sharing configurations and sends out invalid messages automatically. For MCI, the cost for coherence is mainly determined by data sharing method, not the number of cores. Large scale SMP processor can also achieve good performance in MCI. In our experiments, MCI is compared with Jetty which is proposed for reduced snooping operations by A. Moshovos in HPCA2001. For a 32-core processor, snooping operations of MOESI protocol costs more than 50% of L1 cache's power. Jetty can reduce about 24.6% snooping operations for MOESI. Snooping operations in the proposed MCI protocol is about 42.7% less than MOSEI with Jetty.

**Chapter 4 [Communication Network]** describes a Core Interconnection Bus (CIB) for connecting eight cores. In this processor, there are four

clusters, and each cluster is consisted by 8 cores and one shared cache. Data sharing within cluster is supported by the sharing cache. However, it's inefficient to use the shared cache for data communication between SIMD cores, as it needs a write back and read operation to L2.To enhance the data communication ability for SIMD core, CIB is designed for achieving very low latency data transfer between cores. Vector cores directly exchange data through CIB. In CIB, every flit has routing header, and they share the links in time-division multiplexing method. Data transmissions are divided into independent flits. Each flit can route in CIB independently. CIB can broadcast one flit to all cores, which is not supported in channel based NoC. Comparing with S. R. Vangal's NoC design in JSSC08, CIB doesn't store data flits, which can save a lot of buffer resources. Thus CIB can achieve more than 10 GB/s BW for small size vector transfers, which is more than 2.5 times better than S. R. Vangal's work. When the injection rate is less than 8%, CIB's latency is less than 4 cycles in average.

**Chapter 5 [Chip implementation and Performance Evaluation]** shows the chip implementation and performance evaluation of the proposed 32 cores processor. Applying the proposed technology together, a 32 cores processor has been fabricated and verified in SMIC 65 nm CMOS. This chip is consisted by 32 cores (286KGates per core), 256KB L2 cache, two PLL, and one 64-bit DDR PHY, and the die is about 25 mm2.This processor can achieve a maximum speed of 750 MHz at 1.2 V core power. The whole chip can achieve a peak performance of 375 GMACs, or 750 GOPS of 8-bit data operations. It can achieve 1.9 times higher GMACs performance than D. N. Truong's 167 RISC cores chip in JSSC09. For SAD (Sum of Absolute Difference) and

Matrix Multiply kernels, the proposed work's cycle count is reduced by 37% than TI C6415 DSP.

**Chapter 6 [Extended Processor with Hardware Accelerator]** presents a sharing hardware accelerator for extending the proposed SIMD processor. This multicore platform is designed for high performance multimedia applications by maximizing on-chip data level and task level parallelism. However, there are limitations for high parallel system: sequential functions with less parallelism become the bottleneck. Hardware accelerators are added for resolving these problems. As the usage rate of accelerators is very low, it's unnecessary to add the same accelerator for every core of multicore processor. Sharing resources and reducing the cost becomes a hot topic recently for multicore processor. The low latency CIB network makes it possible for sharing hardware accelerators within a cluster. Based on our previous works on 4Kx2K@60fps H.264 Decoder, an intra decoder is added in the multicore platform as a shared hardware accelerator. It can satisfy eight channels parallel HD decoding at 31 MHz.

**Chapter 7 [Conclusion]** summaries the proposals. Based on this design, the proposed 32-core SIMD processor can be applied for a lot of multimedia applications, such as video decoding or image processing. The dual-issue SIMD cores can guarantee very high performance for processing 8-bit pixels. Together with hardware accelerators, the application fields can be wider.