Computers and Electrical Engineering 39 (2013) 2100–2111

Contents lists available at SciVerse ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Efficient sorting design on a novel embedded parallel computing architecture with unique memory access ☆

Wenbiao Zhou *, Zhaoyun Cai, Ruiqiang Ding, Chen Gong, Dake Liu

ASIP Institute, School of Information and Electronic, Beijing Institute of Technology, Beijing, China

ARTICLE INFO

Article history:
Received 23 October 2012
Received in revised form 21 June 2013
Accepted 21 June 2013
Available online 17 July 2013

ABSTRACT

Embedded Parallel computing architecture with Unique Memory Access (ePUMA) is a domain-specific embedded heterogeneous 9-core chip multiprocessor, which has a unique design with low power and high silicon efficiency for high-throughput DSP in emerging telecommunication and multimedia applications. Sorting is one of the most widely studied algorithms, more embedded applications also need efficient sorting. This paper proposes an efficient bitonic sorting algorithm eSORT for the novel ePUMA DSP. eSORT algorithm consists of two parts: an in-core sorting algorithm and an intra-core sorting algorithm. Both algorithms are adapted to the novel architecture and take advantage of the ePUMA platform. This paper implemented and evaluated the eSORT for variable datasets on ePUMA multi-core DSP and compared its performance with the Cell BE processors with the same SIMD parallelization structure. Results show that bitonic sort on ePUMA multi-core DSP has much better performance and scalability. Compared with optimized bitonic sort on Cell BE, the in-core sort is 11 times faster and intra-core sort is 15 times faster in average.

1. Introduction

Sorting is one of the most widely studied algorithmic problems in computer science, its importance has led to lots of efficient sorting algorithms design on a variety of parallel architectures. Many researches have demonstrated that multi-core architecture with Single Instruction Multiple Data (SIMD) instructions can perform high performance sorting [1–4]. Besides making extensive use of sorting operations in database system, many applications that will be used in embedded system also need efficient sort routines. For example, a sorting process is often essential in the construction of spatial data structures, sparse matrix multiplication [5] and parallel programming patterns like MapReduce [6]. It is therefore important to provide efficient sorting routines on any practical programming platform, and as computer architectures evolves, there is a continuing need to explore efficient sorting techniques on emerging architectures.

There is a large body of prior research [7] on sorting for a variety of multi-core and parallel architecture, but ePUMA [8–10], a multi-core Digital Signal Processing (DSP) with some unique features makes it valuable that revisit the fundamental problem once again. ePUMA is a domain-specific embedded heterogeneous 9-core chip multiprocessor, which has a unique design with low power and high silicon efficiency, it aims for high-throughput digital signal processing and image processing computations in emerging telecommunication and multimedia applications. It uses parallel memory access with permutation and orthogonal instruction sub-set to promote the performance, but not the clock frequency.

☆ Reviews processed and recommended for publication to Editor-in-Chief by Associate Editor Dr. Eduardo Cabal-Yepez.

* Corresponding author. Address: 5 South Zhongguancun Street, Haidian District Beijing, Beijing 180001, China. Tel.: +86 10 68918279.
E-mail addresses: zhouwenbiao@bit.edu.cn, wenbiaozhou2010@gmail.com (W. Zhou), zyc@bit.edu.cn (Z. Cai), rqd@bit.edu.cn (R. Ding), cheng@bit.edu.cn (C. Gong), dake@isy.liu.se (D. Liu).

0045-7906/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compeleceng.2013.06.007

This paper proposes an efficient bitonic sorting algorithm ePUMA Sort (eSORT) for the novel ePUMA multi-core DSP. eSORT algorithm consists of two parts: an in-core sorting algorithm and an intra-core sorting algorithm. Both algorithms can be adapted to the novel architecture and take advantage of the ePUMA platform. It implements and evaluates the eSORT for variable datasets on ePUMA multi-core DSP and compares their performance with bitonic on the Cell BE.

The rest of the paper is organized as follows. Section 2 presents a related work for sorting on the SIMD multi-core processor. Section 3 gives a brief overview of the ePUMA DSP. Section 4 describes ePUMA sort kernel design. Experimental results are presented in Section 5. Section 6 is conclusion and future work.

2. Related work

With the development of multi-core processor, lots of optimized sorting algorithms have been proposed on multi-core architecture in the recent decades. This section mainly concentrates on the related sorting work on the SIMD machine since ePUMA multi-core processor is also a SIMD architecture.

Bitonic Sort [11] is one of the earliest parallel sorting algorithms, which is represented by a sorting network consisting of multiple butterfly stages of increasing size. Basic bitonic sorting algorithm is based on repeatedly merging two bitonic sequences to form a larger bitonic sequence. Lots of proofs show that the bitnoic sort is well suited for SIMD processors [1–4].

Cell Broadband Engine (Cell BE) is a 64-bit single-chip multi-processor with 9 processing elements:1 general purpose processing element called PPE and 8 special purpose co-processors called SPEs [12]. CellSort [1] is a distributed bitonic merge with a data parallel bitonic sorting kernel based on Cell BE, it uses a three tiered sort to structure the CellSort and has a good performance. It fully utilizes the high bandwidth for cross-SPE communication by executing as many iterations of sorting as possible while the data is in the local memory.

Graphical Processing Units (GPUs) are a kind of high performance many-core processors. Highly parallel GPU architectures can support thousands of concurrent threads for optimal utilization of computational resources. NVIDIA's GPU [13] provides a Compute Unified Device Architecture (CUDA) programming model and this model provides the means for a developer to map a computing problem to such a highly parallel processing architecture. Article [14] presents a high-performance in-place implementation of Batcher's bitonic sorting networks for CUDA-enables GPUs, they adapt bitonic sort for arbitrary input length and assigned compare/exchange-operations to threads in a way that decrease low performance's global memory access and thereby increase the performance of implementation. Article [15] proposes a new adaptive bitonic algorithm, the results show that it is a fastest sort for NVIDIA's GPU. An analysis of parallel and sequential bitonic, odd-even and rank-sort algorithms on different GPU and CPU architectures is presented in [16].

Besides bitonic sort on SIMD multi-core processor, there are also other optimized sorting algorithms that have been implemented on multi-core processor. AA-sort [2], implemented on a PowerPC, exploits both the SIMD instructions and thread-level parallelism's performance. It consists of two algorithms, the in-core sorting algorithm and the out-of-core sorting algorithm. AA-sort firstly divides all data into blocks that fit into the cache of the processor and sort each block with the in-core sorting algorithm, then merges the sorted blocks with the out-of-core algorithm. It is not a bitonic sort, it uses comb-sort in in-core and the innovative odd-even merge algorithm in out-core. Merge-sort is a comparison based sorting algorithm, article [17] presents an efficient implementation and analysis of Merge-Sort on Intel processor with Streaming SIMD Extensions (SSE) instructions.

There also have some specialized hardware architectures that speed-up the sort process. A high-throughput low latency sorting units are proposed for large data-set sorting [18]. A dynamically reconfigurable architecture is proposed to adopt the well-known bitonic sorting method [19], it reduces the hardware area cost while keeping almost the same performance with the static bitonic sorter.

3. ePUMA platform

Most DSP algorithms exhibit data-independent data access patterns. This gives an opportunity to design an Application Specific Instruction set Processor (ASIP) that exploits the predictable computing properties of DSP algorithms, ePUMA's application domain is specified in multimedia, game and communications. As most of data access and control flow is static, one of the core ideas behind ePUMA is to separate control, addressing and the data-path. Besides SIMD instructions, the SIMD cores also have Single Instruction Multiple Tasks (SIMT) instructions that operate on long vectors in parallel to the main control thread. These instructions handle control and addressing as well as computation, which keeps the overhead to a minimum. Any SIMD instruction is also available as a SIMT instruction; a SIMD instruction is a simple SIMT instruction used in a conventional way, i.e. where the input and the output are single 128-bit words located in register files and no control is specified. Each SIMD core can operate on 128-bit vectors consisting of eight 16-bit elements, placed in wide registers or on-chip-memory. The on-chip memory is divided into three Local Vector Memories (LVM), a Constant Memory (CM) and the Program Memory (PM), see Fig. 1right). The PM is used to store the actual binary SIMD program. The CM is used for storing algorithm constants such as coefficients, configuration parameters and the permutation table, CM cannot be altered at run time. The LVMs are used for data. Only two of the vector memories are accessible at the same time from the SIMD unit, the remaining one can instead be used for simultaneous Direct Memory Access (DMA) transactions. 3 LVMs can also be connected to the data-path supporting butterfly computing for Fast Fourier Transform (FFT). 16 Multiply ACcumulate (MAC)