Microelectronics Journal 40 (2009) 1705–1713

Contents lists available at ScienceDirect

Microelectronics Journal

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

A versatile linear insertion sorter based on an FIFO scheme

Roberto Perez-Andrade, Rene Cumplido *, Claudia Feregrino-Uribe, Fernando Martin Del Campo

Department of Computer Science National Institute for Astrophysics, Optics and Electronics, Puebla, Mexico

ARTICLE INFO

Article history:
Received 24 June 2008
Received in revised form
20 August 2009
Accepted 26 August 2009
Available online 16 September 2009

Keywords:
Hardware sorters
Linear sorters
FIFO

ABSTRACT

A linear sorter based on a first-in first-out (FIFO) scheme is presented. It is capable of discarding the oldest stored datum and inserting the incoming datum while keeping the rest of the stored data sorted in a single clock cycle. This type of sorter can be used as a co-processor or as a module in specialized architectures that continuously require to process data for non-linear filters based on order statistics. This FIFO sorting process is described by four different parallel functions that exploit the natural hardware parallelism. The architecture is composed of identical processing elements; thus it can be easily adapted to any data lengths, according to the specific application needs. The use of compact identical processing elements results in a high performance yet small architecture. Some examples are presented in order to understand the functionality and initialization of the proposed sorter. The results of synthesizing the proposed architecture targeting a field programmable gate array (FPGA) are presented and compared against other reported hardware-based sorters. The scalability results for several sorted elements with different bits widths are also presented.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Sorting is one of the most important operations performed by computers. Given their practical importance, algorithms for sorting data have been the focus of extensive research, resulting in several algorithms proposed to address specific problems. First, serial sorting algorithms were investigated. Then parallel sorting algorithms became a very active area of research, and several models of parallel computations have been considered and developed, deriving in sorting algorithms that later on were implemented in hardware. All developed serial algorithms implemented in software are evaluated by their time complexity and other properties such as the time-memory trade-offs (the amount of additional memory required to run the algorithm and the memory for storing the initial sequence), stability, and sensitivity to the initial distribution of the data (best and worst cases). In parallel processing, when processors share a common memory, the idea of contiguous memory locations is identical to that in serial processors. Therefore, this situation can be analyzed identically as the serial case.

When the processors do not share memory and they communicate with each other through an interconnection network, the time complexity is expressed in terms of parallel comparisons and exchanges between adjacent processors [1,2].

For certain applications, like median filters, asynchronous transfer mode (ATM) switching, order statistics filtering and, in general, continuous data processing, sometimes software-only implementations of sorting algorithms do not achieve the required processing speed [3]. In order to speed up the sorting operation, some custom hardware architectures have been proposed in recent years. The relatively simple logic required for sorting and the inherent concurrency of the algorithms have allowed exploring a number of custom architectures. Hardware sorters are evaluated according to area requirements (number of flip–flops, comparators, control logic, gates, and LUTs), processing time, including latency and maximum operating frequency, and power consumption. Hardware sorters can be grouped into two kinds of architectures: sorting networks, including some systolic architectures, and linear arrays. The main idea behind sorting networks is to sort a block of data passing through a network of processing elements (PE) connected in such way that a datum takes its corresponding place. Linear sorters are based on the idea that data to be sorted come in a continuous stream, one datum at a time; each datum is inserted into its corresponding place in a register group (sorting array) at the same time that one of the stored data is deleted. Fig. 1.a represents the sorting network idea, where data are firstly stored and then sorted by a sorting network in a parallel fashion [4]. The gray blocks represent the first and the last stored datum. The first stored datum is the first element leaving the file register, i.e. like in an FIFO scheme. Fig. 1.b represents the linear sorter idea, where data are always sorted,

* Corresponding author.
E-mail addresses: j_roberto_pa@inaoep.mx (R. Perez-Andrade), rcumplido@inaoep.mx (R. Cumplido), cferegrino@inaoep.mx (C. Feregrino-Uribe), fmartin@inaoep.mx (F. Martin Del Campo).

0026-2692/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mejo.2009.08.006

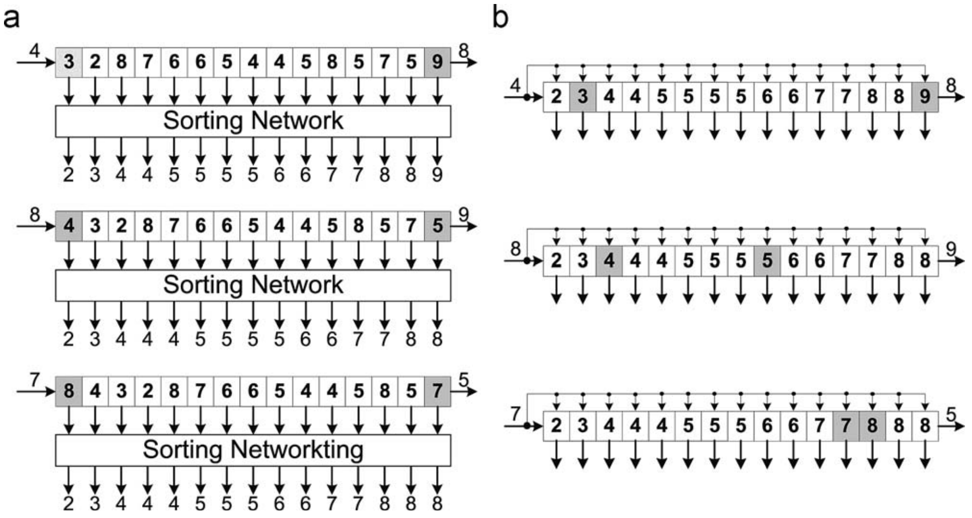

Fig. 1. Sorting network and linear sorter.

thus the first and the last datum are merged inside the sorting array. On these sorters, a deleting mechanism must be used in order free space for incoming data. Some examples of these mechanisms are deleting the oldest datum, selecting one datum or deleting the greatest or the smallest one.

This work is based on the idea of sorting the data as they are introduced into the sorting array, discarding the oldest datum in the sorting array while maintaining the data sorted, all that in a single clock cycle. This FIFO scheme can be used in applications that are continuously processing data in serial fashion like non-linear filters such as the rank-order filters, weighted order statistics (WOS) filters, and stack filters. These non-linear filters are based on order statistics, thus require to access an ordered list of the random variables $X_1, X_2, X_3, \ldots, X_n$. An ascending sequence can be represented as follows:

$$X_1 \le X_2 \le X_3 \le,\ \ldots,\ X_n \tag{1}$$

where indexes indicate the rank-order number. The idea on rank-order filters is to select a value X_i, where $i \in \{1, 2, 3, \ldots, n\}$ from the sequence in Eq. (1) and then to use this X_i value as a sample of the sorted data.

Several signal processing applications based on order statistics require a sorter with an FIFO-like behavior. For example, in image processing, non-linear filters such as: rank order, max/min, mean, morphological, and adaptive trimmed mean are commonly used as they offer benefits such as edge preservation, robustness, adaptation to noise statistics, and preservation of image details [5,6]. Other applications can be found in radar and sonar systems where the detection procedures involve the comparison of the received signal amplitude to a threshold. This threshold is obtained by using a constant false alarm ratio (CFAR) algorithm, which requires keeping sorted the incoming echo samples [7]. More examples of applications in signal processing are: smoothing of time-series, maximum likelihood estimation, and one-dimensional non-linear filtering [8].

These applications require accessing a value from a specific position within a sorted array, more than one value simultaneously, or even the whole set of values in the array to perform parallel operations, thus making traditional FIFO memories with a single output port unsuitable. The proposed architecture for the insertion sort algorithm has an FIFO-like behavior, i.e. it discards the oldest datum when a new one arrives, while allowing flexible access to its contents.

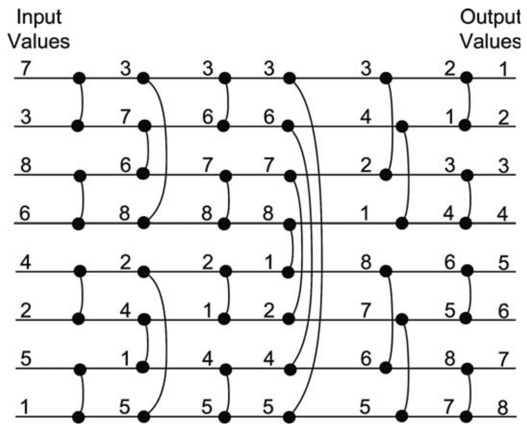

Fig. 2. Sorting network example.

2. Related work

As mentioned earlier, several hardware architectures for performing sorting algorithms have been proposed. These architectures can be grouped in two families according to the algorithm they use: sorting networks and linear sorters. The sorting networks are based on a network constituted by several PEs, which consists of a comparator and are located in the nodes of the network. The goal of each PE is sorting two input data in ascending (or descending) order by placing the larger (or smaller) datum in a specific output [4]. This technique supposes that a block of data is available for being sorted in parallel fashion. Sorter networks can be pipelined in order to reduce their critical path and latency, thus resulting in a better throughput. The disadvantage of this approach is that the network can potentially require a large number of PEs and, depending on the algorithm, several clock cycles for sorting the whole block of data. Besides, even if only one input datum changes, the whole block of data must be resorted. The efficiency of these sorters can be measured by its total size (numbers of PEs) and by its depth (maximum number of PEs from input to output). Both metrics are highly dependant on the number of data the architecture can sort. Fig. 2 shows an eight elements input sorting network example, whose size is of 24 PEs and has a depth of 6 stages. Each PE is represented by two interconnected nodes.

دریافت فوری ← متن کامل مقاله

ISIArticles
مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات