Abstract:

We propose a novel sorting algorithm that sorts input data integer elements on-the-fly without any comparison operations between the data — comparison-free sorting. We present a complete hardware structure, associated timing diagrams, and a formal mathematical proof, which show an overall sorting time, in terms of clock cycles, that is linearly proportional to the number of inputs, giving a speed complexity of the order of $O(N)$. Our hardware-based sorting algorithm precludes the need for SRAM-based memory or complex circuitry, such as pipelining structures, but rather uses simple registers to hold the binary elements and the elements’ associated number of occurrences in the input set, and uses matrix-mapping operations to perform the sorting process. Thus, the total transistor count complexity is on the order of $O(N)$. We evaluate an application-specified integrated circuit design of our sorting algorithm for a sample sorting of $N = 1024$ elements of size $K = 10$-bit using Xilinx software. Results verify that our sorting requires approximately $4–6 \, \mu s$ to sort the 1024 elements with a clock cycle time of 0.5 GHz, consumes 1.6 mW of power, and has a total transistor count of less than 750,000.

Existing system

Sorting algorithms have been widely researched for decades due to the ubiquitous need for sorting in many application domains. Due to the ever-increasing computational power of parallel processing on many core CPU- and GPU-based processing systems, much research has focused on harnessing the computational power of these resources for efficient sorting. Additionally, there is no clear dominate sorting algorithm due to many factors including the algorithm’s percentage utilization of the available CPU/GPU resources, the specific data type being sorted, amount of data being sorted. To address these challenges, much research has focused on architecting customized hardware designs for sorting algorithms in order to fully utilize the hardware resources and provide custom, cost-effective hardware processing. However, due to the inherent complexity of the sorting algorithms, efficient hardware implementation is challenging. Furthermore, these structures are not inherently scalable due to the complexity of integrating and...
combining the data path and control logic within the processing units, thus potentially requiring a full redesign for different data sizes, as well as complex connective wiring with high fan-out and fan-in in addition to coupling effects, thus circuit timing issues are challenging to address. Additionally, if multiple processors are used along with pipelining stages and global memories, the data must be globally merged from these stages to output the complete final sorted data set.

Proposed system

We propose a new sorting algorithm targeted for custom, IC-designed applications that sort small- to moderate-sized input sets, such as graphics accelerators, network routers, and video Processing DSP chips. In videoprocessing, fast computation is required for small matrices in a frame in order to increase the resolution using digital filters that leverage sorting algorithms. Even though we present our design based on these scenarios, our design also supports processing large input sets by subsequently processing the data in multiple, smaller input sets (i.e., in sets of $N < 100,000$) using fast computations, and then merging these sets. However, since applications with larger input sets (on the order of millions) are usually embedded into systems with large computational resources, such as data mining and database visualization applications running on high-performance grid computing and GPU accelerators, these applications can harness those powerful resources for sorting.

Applications

1) Communications
2) Digital signal processing

Advantages

Area, delay and power reduced

System Configuration:

In the hardware part a normal computer where Xilinx ISE 14.3 software can be easily operated is required, i.e., with a minimum system configuration

HARDWARE REQUIREMENT
Processor - Pentium –III

Speed - 1.1 GHz

RAM - 1 GB (min)

Hard Disk - 40 GB

Floppy Drive - 1.44 MB

Keyboard - Standard Windows Keyboard

Mouse - Two or Three Button Mouse

Monitor - SVGA

SOFTWARE REQUIREMENTS


❖ Front End : Modelsim 6.3 for Debugging and Xilinx 14.3 for Synthesis and Hard Ware Implementation

This software’s where Verilog source code can be used for design implementation.