Yen-Kuang Chen,
Chaitali Chakrabarti,
Shuvra Bhattacharyya, and
Bruno Bougard

# Signal Processing on Platforms with Multiple Cores: Part 2–Applications and Design

Platforms with multiple cores are now prevalent everywhere from desktops and graphics processors to laptops and embedded systems. By adding more parallel computational resources while managing power consumption, multicore platforms offer better programmability, performance, and power efficiency. Signal processing systems of tomorrow will be and must be implemented on platforms with multiple cores. Writing efficient parallel applications that utilize the computing capability of many processing cores require some effort. Signal processing algorithm designers must understand the nuances of a multicore computing engine; only then can the tremendous computing power that such platforms provide be harnessed efficiently. To give a thorough perspective of the area, we have organized two special issues on this topic.

The first special issue, published in November 2009, provided an overview of multiple core systems along with some key methodologies. The articles provided coverage of key trends and emerging directions in architectures, design methods, software tools, and application development for the design and implementation of multicore signal-processing systems. There were three articles surveying the multicore architectures, from general-purpose processors and digital signal processors (DSPs) to multiprocessor system-on-chip. These were followed by four articles discussing software development methodology, including compilation tools that discover parallelism automatically, parallel programming languages where programmers can annotate the parallelism, and approaches that require programmer to explicitly express the parallelism. Part 1 of the two-part special issue ended with four design-example articles spanning fast Fourier transform (FFT), video processing, video coding, and speech recognition.

This special issue aims at 1) describing novel applications that can be enabled by platforms with multiple cores and 2) providing more extensive design examples to demonstrate useful techniques for developing efficient signal processing applications on platforms with multiple cores. Because multicore processors provide better programmability, performance, and power efficiency, many computationally demanding applications are now feasible at a much lower cost. To illustrate this point, we look at applications that can be enabled by multicore platforms. Furthermore, to provide more comprehensive examples on how applications can be realized, we review implementation details on a larger set of applications.

**SIGNAL PROCESSING SYSTEMS OF TOMORROW WILL BE AND MUST BE IMPLEMENTED ON PLATFORMS WITH MULTIPLE CORES.**

There are a total of ten articles in this special issue. They can be broadly classified into novel applications that can be enabled by platforms with multiple cores (articles one–four), and design examples illustrating useful techniques to enable efficient implementations on these platforms (articles five–ten). Some of the articles represent both categories because it is often difficult to demonstrate novel applications without providing some relevant implementation details. In such cases, we have categorized the articles according to whether they emphasize the enabled applications or focus more on implementation techniques.

The first article by Palkovic et al. shows that software-defined radio (SDR), an attractive solution for handling diverse and evolving wireless standards, makes effective use of multicore platforms. This article provides an overview of multicore architectures for SDR platforms along with the specifics of their mapping flows. The authors also show how this technology can be harnessed to handle more complex workloads of emerging wireless communication standards.

The second article by Rzeszutek et al. shows that object segmentation with interactive frame rates can be enabled by the computational capability of multicore platforms. Segmenting the boundaries of objects in a video sequence is a complex and time-consuming task. Furthermore, many objects of interest, such as people and animals, have highly complex and irregular shapes. This article presents a rotoscoping method that takes advantage of the ubiquitous multicore processors such as graphics processing units (GPUs) to assist artists.

The third article by Samsi et al. explores the acceleration of computationally intensive applications by applying commonly used tools to exploit multicore platforms. The authors show that with small changes to sequential MATLAB code, it is possible to effectively utilize today's multicore systems and reduce simulation time. Two signal processing kernels (FFT and convolution) and two full applications (synthetic aperture radar imaging and superconducting quantum interference devices) are used to illustrate the use of parallel MATLAB.

*Digital Object Identifier 10.1109/MSP.2009.935527*

The last article of the novel application category shows that medical imaging, which is highly computation intensive, can now be implemented on an easily available multicore platform. The focus is on medical image registration, which is an integral part of image-guided intervention and therapy systems. Shams et al. provide an extensive survey of image registration algorithms and their performance on various multiprocessor platforms, including cluster computers, GPUs, and CELL.

The first design example article by Plishker et al. is a specific design example of medical imaging on GPUs. It demonstrates that we must exploit hierarchical parallelism properly to get the best utilization of the platforms. Although multicore platforms offer significant performance potential, there are challenges in finding and exploiting the parallelism. The authors depict a synergistic approach that first organizes application parallelism into a domain-specific taxonomy and then structures the algorithm to target a set of multicore platforms.

The article by di Bisceglie et al. demonstrates the need to pay attention to the usage of resources (e.g., device memories, kernel functions, and synchronizations) and choose appropriate data transfer granularity to use the GPU resources efficiently. The authors show a design example of synthetic aperture radar on the GPU. While signal processing algorithms for synthetic aperture radar are becoming mature, it is a challenge to produce an accurate image in real time without a mainframe computer. This article provides an example on how to implement an important subset of focusing algorithms on general-purpose GPUs.

The next article by Cheung et al. advocates the need to structure the algorithm to expose as much data parallelism as possible to utilize the computational capability. The article illustrates this for video codecs running on GPUs. While the GPU offers high peak performance, it is challenging to achieve it because of the dependencies imposed by the codec. The authors demonstrate that with the proper algorithm redesign, the performance of the fast motion estimation algorithm, for example, on GPU can be improved by three to four times.

The article by Daudet also illustrates that we must reformulate the algorithm to expose the parallelism. He demonstrates this for the matching pursuit algorithm that is often used to solve very large sparse approximation problems. While matching pursuit is considered intrinsically sequential, a small modification of the algorithm can break the data dependencies and enable efficient implementation on multicore processors.

**THE GOAL OF THE TWO-PART SPECIAL ISSUE WAS TO CAPTURE STATE OF THE ART IN SIGNAL PROCESSING ON PLATFORMS WITH MULTIPLE CORES.**

The article by Kim et al. shows that we must consider the underlying architectural features in parallelizing workloads. Image processing applications have an abundance of parallelism and benefit significantly from multicore systems. However, simply exploiting parallelism is not enough to achieve the best performance. Optimization must take into account underlying architecture characteristics such as wide vector and limited bandwidth. The article presents techniques that can be used to optimize performance for multicore x86 systems on three key image processing kernels, FFT, convolution, and histogram.

The last article by van Nieuwpoort and Romein demonstrates that we must have different implementation and optimization strategies for multicore architectures with different performance characteristics. The authors choose correlation computation in radio astronomy signals to compare the performance, optimization, and programmability of multiple multicore platforms. The article shows how to predict what performance can be achieved on many-core platforms and where bottlenecks can be expected. The authors also provide guidelines for optimizing on the different platforms.

In short, this special issue showed that many novel applications can be enabled by platforms with multiple cores. These include image processing, video processing, rotoscoping, medical imaging, SDR, synthetic aperture radar, and radio astronomy signal processing. This special issue also demonstrated useful techniques to develop efficient signal processing applications on platforms with multiple cores. The era of signal processing on systems with multiple/many cores has just started. We hope that you enjoy the articles in this special issue of *IEEE Signal Processing Magazine* as much as those in the November 2009 issue and that you find the contents informative and useful.

We sincerely thank the authors for their valuable contributions, as well as the anonymous reviewers for their help in ensuring the quality of this special issue. The goal of the two-part special issue was to capture state of the art in signal processing on platforms with multiple cores. Many papers were not selected because they did not fit into the scope of the special issue, even though some of them would have been excellent matches for other publications. Furthermore, the high quality requirements of *IEEE Signal Processing Magazine* forced us to reject papers even after multiple rounds of reviews. In fact, of the 144 white papers that were submitted, only 21 articles were published in these two special issues. We would like to thank everyone who submitted articles for their effort and express our regret that due to limited space and the need for balanced coverage, not all high-quality submissions could be included.

Finally, we would like to thank Li Deng, Geri Krolin-Taylor, and Dan Schonfeld for their help and support in organizing the two-part special issue.

[SP]