Accelerating System-Level Design Tasks using Graphics Processors

Udeepta D. Bordoloi
Advanced Micro Devices (AMD)
Sunnyvale, USA

Samarjit Chakraborty
TU Munich
Germany

Unmesh D. Bordoloi
Linköping University
Sweden

(Tutorial proposal for VLSI Design 2011)

Length of the tutorial: 3 hours (half-day tutorial) (can be extended to 6 hours if requested)

Summary: Recent years have seen the increasing use of graphics processing units (GPUs) for non-graphics related applications. Applications that have harnessed the computational power of GPUs span across numerical algorithms, computational geometry, database processing, image processing, astrophysics and bioinformatics. There are many compelling reasons behind exploiting GPUs for such general-purpose computing tasks. First, modern GPUs are extremely powerful. For example, high-end GPUs such as the NVidia GeForce GTX 480 and ATI Radeon 5870 have 1.35 TFlops and 2.72 TFlops of peak single precision performance, whereas a high-end general-purpose processor such as the Intel Core i7-960 has a peak performance of 102 Gflops. Additionally, the memory bandwidth of these GPUs are more than 5x greater than what is available to a CPU, which allows them to excel even in low compute intensity but high bandwidth usage scenarios. Second, GPUs are now commodity items as their costs have dramatically reduced over the last few years.

In spite of a wide variety of computationally expensive system-level design tasks (in the context of embedded systems design) that are regularly solved by software tools running on desktops and laptops equipped with high-end GPUs, the use of GPUs for accelerating such problems is still not a conventional practice within the design automation community. As a result, of late, there has been a lot of research interest in demonstrating the applicability of GPUs in accelerating design automation tasks. Some of tasks that have been accelerated using modern GPUs include schedulability/timing analysis, hardware/software partitioning, fault simulation, and verification of digital designs. In this tutorial we will describe techniques for programming GPUs for general purpose computing (i.e., non-graphics applications) and cover a number of case studies from the electronic design automation area. We will demonstrate how GPUs can lead to significant improvement in running times and hence the usability of the design tools that exploit them. In particular, we will start by introducing the graphics processor architecture and programming models for GPUs (OpenCL and CUDA). OpenCL is an open standard for programming GPUs (and also other modern processors) and is a cross-platform alternative to CUDA. It has been created by a consortium that includes AMD, Apple, IBM, Intel, and Nvidia. We will then discuss various examples of system-level design tasks and identify their computational kernels. Finally, we will present different case studies to illustrate how system-level design algorithms have to be suitably modified in order to map them onto GPUs.

Target audience: This tutorial will be specifically targeted towards an audience who has some background in embedded systems design but is new to GPU programming. The level of the tutorial will be from introductory to intermediate and no background in advanced system-level design techniques, GPU programming or GPU architecture will be assumed. The lectures will give an overview of the current state-of-the-art in GPGPUs (General Purpose Computing on Graphics Processor Units), will introduce the relevant background material in system-level design, and finally present concrete case studies on the use of GPUs. The material to be presented will be useful to researchers from the embedded systems domain, students, and software developers interested in the use of graphics processors for design automation tasks.

Keywords: Electronic Design Automation, General Purpose Computation on Graphics Processor Units (GPGPU), CUDA, OpenCL

Contact Information of the presenters:
Udeepta D. Bordoloi, Advanced Micro Devices (AMD), Sunnyvale, USA, udeepta.bordoloi@amd.com
Samarjit Chakraborty, TU Munich, Germany, samarjit@tum.de **(Coordinating presenter)**
Unmesh D. Bordoloi, Linköping University, Sweden, unmesh.bordoloi@liu.se

Tutorial objectives and outline: The main objective of this tutorial will be to give an overview of programming techniques for GPUs, with a particular focus on system-level design applications from the design automation domain. A tentative list of topics to be covered is given below:

- Introduction to GPUs and programming models [1 hour]
  - Modern GPU architecture
    - Overview of the latest ATI and Nvidia GPU architectures
    - Data parallelism in GPUs
    - Control flow and branching
    - Memory hierarchies
    - Inter-thread synchronization and data sharing
  - Programming Model (OpenCL)
    - Platform layer and runtime API (multiple vendors/devices)
    - Memory model
    - OpenCL C language for kernels
    - Asynchronous execution
    - C++ API
  - Comparison of CUDA and OpenCL
    - Similarities and mapping between the two API
  - Code example
- Introduction to system-level design tasks [1 hour]
  - High-level design of embedded systems
    - Embedded systems architecture and software
    - Modeling embedded systems
    - Timing and performance analysis
  - System-level design tasks and algorithms
    - Hardware/software partitioning and task mapping
    - Timing and schedulability analysis
    - Designing systems with reconfigurable architectures
    - Custom instruction selection problem and its complexity
    - System verification and simulations
- Accelerating system-level design tasks using GPUs (Case Studies) [1 hour]
  - Mapping timing analysis algorithms onto GPUs
  - Using GPUs to accelerate multi-objective custom instruction problems
  - Parallel simulation on GPUs

Selected Technical Bibliography

- U.D. Bordoloi, and S. Chakraborty. GPU-based Acceleration of System-Level Design Tasks. *International Journal of Parallel Programming* 38(3-4): 225-253*,* Springer, 2010
- U.D. Bordoloi and S. Chakraborty. Accelerating System-Level Design Tasks using Commodity Graphics Hardware: A Case Study. *22nd International Conference on VLSI Design and 8th International Conference on Embedded Systems*, New Delhi, January 2009
- J. Feng, S. Chakraborty, B. Schmidt, W. Liu, and U.D. Bordoloi. Fast Schedulability Analysis Using Commodity Graphics Hardware. *13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, Daegu, Korea, August 2007
- K. Gulati, and S. Khatri. Fault Table Generation using Graphics Processing Units, *International High Level Design Validation and Test Workshop*, San Francisco, November 2009

- D. Chatterjee, A. De Orio and V. Bertacco. GCS: High Performance Gate-Level Simulation with GP-GPUs*. Design Automation and Test in Europe (DATE)*, Nice, France, April 2009
- Khronos OpenCL Working Group: The OpenCL Specification, Version 1.1. www.khronos.org/opencl

Tutorial material to be provided: The lecture slides, annotated with detailed explanations where necessary, will be made available to the audience.

Presenters' Bio:

Udeepta D. Bordoloi is a Senior Member of Technical Staff at AMD working on GPGPU technology. He is an expert in high performance computation and GPUs, and has helped advance and optimize GPU utilization in multiple large-scale GPU based clusters in the top500 supercomputers list. He has published papers, tutorials and lectures promoting the state-of-the-art in GPUs, and collaborates with the academia and the industry on the newest features of the latest GPUs being developed at AMD. He obtained his MS degree in Electrical Engineering from Washington University in St. Louis and his PhD in Computer Science from The Ohio State University. Since then he has worked in the field of graphics and visualization with industry leading companies and also with bootstrapping startups. He has many academic publications and patents to his name and continues to collaborate with and contribute to the research community. Besides GPGPU and HPC, he has interests in Image Processing, Graphics and Visualization.

Samarjit Chakraborty is a Professor of Electrical Engineering at TU Munich in Germany, where he heads the Institute for Real-Time Computer Systems (RCS). Prior to joining TU Munich, from 2003 – 2008 he was an Assistant Professor of Computer Science at the National University of Singapore. He obtained his PhD in Electrical and Computer Engineering from ETH Zurich in 2003. His research interests cover all aspects of system-level design of real-time embedded systems and software, and his work has attracted funding from companies like Intel, General Motors, Bosch and BMW. Prof. Chakraborty regularly serves on the technical program committees of several conferences in the area of real-time and embedded systems, such as DATE, CODES+ISSS, ASP-DAC, RTSS and RTAS. He was the TPC Co-Chair of EMSOFT 2009, and is currently serving as the General Co-Chair of Embedded Systems Week 2010. Apart from more than 80 referred journal and conference articles, he has authored a number of patents and regularly gives invited talks and tutorials at various research labs and international conferences (such as ESWeek, VLSI Design, ACM Multimedia and ICME). For his Ph.D. thesis, he received the ETH Medal and the European Design and Automation Association's "Outstanding Doctoral Dissertation Award" in 2004. His work has also received a HiPEAC Paper Award in 2009 and Best Paper Award nominations at DAC 2005, CODES+ISSS 2006, ECRTS 2007, and CODES+ISSS 2008.

Unmesh D. Bordoloi is a post-doctoral researcher at Linköping University, Sweden. Prior to joining Linköping he spent one year as a post-doctoral researcher at the Verimag Laboratory in Grenoble, France. He obtained his PhD from the School of Computing at the National University of Singapore in 2008. He was awarded the "President's Graduate Fellowship" and the "Sun Microsystems Ph.D. Fellowship"' for his research work at the National University of Singapore. He received a Bachelor of Technology in Computer Science and Engineering from the National Institute of Technology, Rourkela in 2004. In 2007, he visited General Motors R&D, India Science Lab at Bangalore, as an intern. His research focuses on system-level design and analysis issues like schedulability analysis, hardware/software co-design, and fault-tolerance. His work has appeared in a number of top-tier conferences and journals such as DAC, RTAS and the ACM Trans. Embedded Computing Systems. He is currently investigating techniques for automatically mapping a large class of algorithms from the design automation domain onto GPUs.