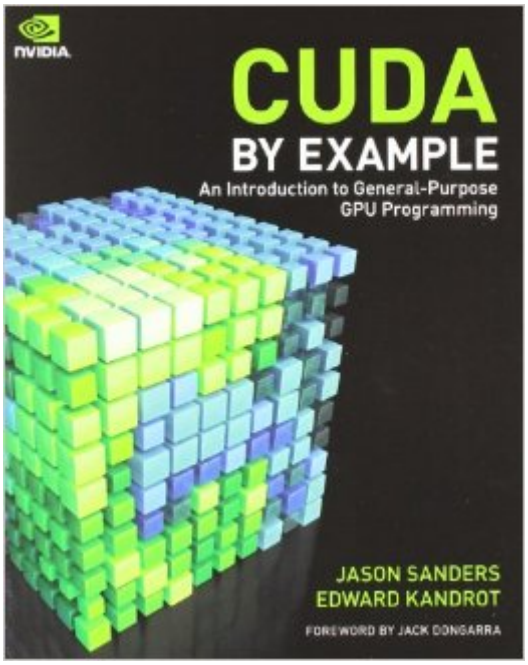
CUDA By Example: An Introduction To General-Purpose GPU Programming



Synopsis

â œThis book is required reading for anyone working with accelerator-based computing systems.â • â “From the Foreword by Jack Dongarra, University of Tennessee and Oak Ridge National Laboratory

Book Information

Paperback: 312 pages

Publisher: Addison-Wesley Professional; 1 edition (July 29, 2010)

Language: English

ISBN-10: 0131387685

ISBN-13: 978-0131387683

Product Dimensions: 7.3 x 0.9 x 9 inches

Shipping Weight: 1.2 pounds (View shipping rates and policies)

Average Customer Review: 4.2 out of 5 starsÂ Â See all reviewsÂ (60 customer reviews)

Best Sellers Rank: #67,427 in Books (See Top 100 in Books) #4 inÂ Books > Computers & Technology > Programming > Parallel Programming #32 inÂ Books > Computers & Technology > Hardware & DIY > Design & Architecture #138 inÂ Books > Computers & Technology > Programming > Introductory & Beginning

Customer Reviews

"CUDA by example: an introduction to general-purpose GPU programming" is a brand new text by Jason Sanders and Edward Kandrot, senior members of NVIDIA’s CUDA development team. This is basically the second introductory text to hit the market on general-purpose GPU programming, the first one being "Programming Massively Parallel Processors: A Hands-On Approach" by David Kirk and Wen-Mei Hwu.The Good: it is not very common to find a technical book in this price range that is not simply in greyscale. Perhaps unsurprisingly for an NVIDIA book there’s quite a bit of green, and this definitely enhances the reading experience. On a more substantive note: the authors really mean the "by example" part of "CUDA by example". From chapter 3 onward, all the main concepts are fleshed out by showing and dissecting lots of code -- probably more so than in Kirk & Hwu’s text, which includes application case studies, but also more extensive treatments of the CUDA architecture. As with any example-based book, it is important to run and modify the programs while reading through the text. Right now there are a few hiccups with the files Sanders & Kandrot were kind enough to provide (e.g. as of this writing README.txt and license.txt do not have the appropriate permissions set), but I’m pretty sure these are just teething troubles which will disappear

soon enough. The writing is cheerful (e.g. "For those readers who are more familiar with Star Trek than with weaving, a warp in this context has nothing to do with the speed of travel through space.", p. 106) and the explanations are for the most part clear, the language being pretty lucid -- once again, probably more so than in the Kirk & Hwu volume. This fact, along with the availability of lecture slides and lab materials for the latter book, points to the main difference between the two texts: Sanders & Kandrot are better-suited to a self-study of CUDA C, while the Kirk & Hwu book is more of a class textbook (and thus broader). Finally, I was pleased to see Sanders & Kandrot include a whole chapter (chapter 11) on working with multiple GPUs, a topic Kirk & Hwu relegate to a short section.The Bad: having color is a welcome addition, but I could not understand why the authors chose to simply follow the text editor’s default highlighting of keywords when they could have used color to highlight specific portions of the code. Similarly, a number of figures (e.g. Figs. 5.5 and 8.1) are described in the text as containing green, but they show up in greyscale. The book also contains quite a few minor typos, but that’s normal; what’s not normal is that every single section cross-reference outside the appendix is wrong (I counted 16 in total). Moving on to more consequential matters: Kirk & Hwu have a chapter on floating point topics; given that numerical computations are certainly part of general-purpose GPU programming, Sanders & Kandrot could have said more about them. On a different note, Kirk & Hwu have a chapter on the competing programming model OpenCL, while Sanders & Kandrot do not even have an index entry on it -- one might counter-argue here that they have knowingly put CUDA in the title. This brings me to my main gripe with this book: why didn’t the authors just call it "CUDA C by example"? I believe the answer is connected to their ambivalence toward C++. An illustrative example: new and delete are used in host code only once in the entire volume (on p. 82 and p. 84, respectively), but when the code snippets are shown again (on pp. 86-87) new and delete have been silently replaced by malloc and free! In the case of device code, the authors do not discuss CUDA-supported C++ constructs like default parameters, namespaces, function templates, not to mention compute capability 2.0 things like function objects. (Structures with member functions do not beget C++). In a nutshell, the book contains too much C++ for people who only know C, and not enough C++ for those who actually use that language.Despite these misgivings, I cannot ignore this book’s low selling price (especially on the Kindle), its practical focus on a multitude of code listings, and the fact that its explanations are generally clear. Thus, I think it is an appropriate buy for those interested in learning about CUDA C.Alex Gezerlis

This is an excellent introduction to CUDA. The prose and content are excellent: I read it

cover-to-cover in a single sitting and enjoyed every page.The authors clearly explain the basic CUDA paradigm starting with very simple code and working up to progressively more complex examples. The authors spend a considerable amount of time discussing different memory types and memory access styles, motivating when each style is appropriate. The code snippets are clean, clear and concise, providing a minimal yet complete introduction to each new language feature.Highly recommended!The book does not provide an HTML pointer to the source code used in the book. Edward Kandrot writes: "The Kindle version shipped a week too soon, it was supposed to ship next week when the physical book ships. Because of this, the website at NVIDIA wasn't done yet. Jason just spent the day making the website happen![...] is where the source code is currently located. I hope this helps. I wrote the examples to be specific for what is being covered, putting extras in the header files so as not to distract from the topic at hand. Only really works if the reader has the header files as well..."

I downloaded CUDA by Example on the Kindle and starting reading it. Sanders and Kandrot provide a nice step by step walk through of how to program with CUDA and the examples are really straight forward. It begins with the basic hello world introduction to the programming model, then dives deeper into the different API features with examples in each chapter.I would recommend this book to anyone who wants to get started using CUDA.(Found the source code online, not sure what the other review is about.)

I've done some work with CUDA and read a number of books and tutorials. This book does a very good job of relating the syntax and structure, but this book really doesn't go beyond showing you how to get your code to compile when using different features. It does not show you how to write efficient CUDA code (Getting a 7x speedup on a card running 960 threads simultaneously should *not" be considered very impressive. We've recently gotten >60x speedups, but using concepts that aren't covered in this text). I know the book industry doesn't turn on a dime, so I can certainly understand that no specific discussion is given to Fermi (though the book does list those cards), and there are (I think they claim) 200 million non-Fermi cards out there, so there is still more than enough reason to write apps that need to know how these "older" cards work. You really need a reference that will also discuss optimizing for register use, coalesced memory accesses, divergence, etc. in much greater depth.So, given the low price, it's a useful buy if you prefer a book instead of going through some online tutorials. But, if you want to write fast, efficient code, don't stop at this book.

CUDA by Example: An Introduction to General-Purpose GPU Programming CUDA Programming: A Developer's Guide to Parallel Computing with GPUs (Applications of Gpu Computing) CUDA Handbook: A Comprehensive Guide to GPU Programming, The Nathan Wallace's Delphi 3 Example Book (Programmer's Example Series) Programming Massively Parallel Processors: A Hands-on Approach (Applications of GPU Computing Series) CUDA for Engineers: An Introduction to High-Performance Parallel Computing Daily Inspiration for the Purpose Driven Life: Scriptures and Reflections from the 40 Days of Purpose Deep Belief Nets in C++ and CUDA C: Volume III: Convolutional Nets (Volume 3) Deep Belief Nets in C++ and CUDA C: Volume 1: Restricted Boltzmann Machines and Supervised Feedforward Networks Java: The Simple Guide to Learn Java Programming In No Time (Programming,Database, Java for dummies, coding books, java programming) (HTML,Javascript,Programming,Developers,Coding,CSS,PHP) (Volume 2) GPU Pro 7: Advanced Rendering Techniques GPU Pro 6: Advanced Rendering Techniques GPU Pro 5: Advanced Rendering Techniques Korn Shell Programming by Example Microsoft Access 2013 Programming by Example with VBA, XML, and ASP (Computer Science) Microsoft Access 2016 Programming By Example: with VBA, XML, and ASP MicrosoftÂ® AccessÂ® 2010 Programming By Example: with VBA, XML, and ASP (Computer Science) Programming Game AI By Example (Wordware Game Developers Library) Arachnomania: The General Care and Maintenance of Tarantulas and Scorpions (General Care and Maintenance of... Series) Clinical Anesthesia Procedures of the Massachusetts General Hospital: Department of Anesthesia, Critical Care and Pain Medicine, Massachusetts General Hospital, Harvard Medical School 8th (eighth) Edition published by Lippincott Williams & Wilkins (2010)