

1. Record Nr.	UNINA9910810195503321
Titolo	Heterogeneous computing with OpenCL 2.0 // David Kaeli [and three others]
Pubbl/distr/stampa	Amsterdam, [Netherlands] : , : Morgan Kaufman, , 2015 ©2015
ISBN	0-12-801649-3 0-12-801414-8
Edizione	[Third edition.]
Descrizione fisica	1 online resource (330 p.)
Disciplina	005.275
Soggetti	Parallel programming (Computer science) OpenCL (Computer program language)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references at the end of each chapters and index.
Nota di contenuto	Front Cover; Heterogeneous Computing with OpenCL 2.0; Copyright; Contents; List of Figures; List of Tables; Foreword; Acknowledgments; Chapter 1: Introduction; 1.1 Introduction to Heterogeneous Computing; 1.2 The Goals of This Book; 1.3 Thinking Parallel; 1.4 Concurrency and Parallel Programming Models; 1.5 Threads and Shared Memory; 1.6 Message-Passing Communication; 1.7 Different Grains of Parallelism; 1.7.1 Data Sharing and Synchronization; 1.7.2 Shared Virtual Memory; 1.8 Heterogeneous Computing with OpenCL; 1.9 Book Structure; References; Chapter 2: Device architectures; 2.1 Introduction 2.2 Hardware Trade-offs 2.2.1 Performance Increase with Frequency, and its Limitations; 2.2.2 Superscalar Execution; 2.2.3 Very Long Instruction Word; 2.2.4 SIMD and Vector Processing; 2.2.5 Hardware Multithreading; 2.2.6 Multicore Architectures; 2.2.7 Integration: Systems-on-Chip and the APU; 2.2.8 Cache Hierarchies and Memory Systems; 2.3 The Architectural Design Space; 2.3.1 CPU Designs; Low-power CPUs; Mainstream desktop CPUs; Server CPUs; 2.3.2 GPU Architectures; Handheld GPUs; At the high end: AMD Radeon R9 290X and NVIDIA GeForce GTX 780; 2.3.3 APU and APU-like Designs; 2.4 Summary Chapter 3: Introduction to OpenCL; 3.1 Introduction; 3.1.1

The OpenCL Standard; 3.1.2 The OpenCL Specification; 3.2 The OpenCL Platform Model; 3.2.1 Platforms and Devices; 3.3 The OpenCL Execution Model; 3.3.1 Contexts; 3.3.2 Command-Queues; 3.3.3 Events; 3.3.4 Device-Side Enqueuing; 3.4 Kernels and the OpenCL Programming Model; 3.4.1 Compilation and Argument Handling; 3.4.2 Starting Kernel Execution on a Device; 3.5 OpenCL Memory Model; 3.5.1 Memory Objects; Buffers; Images; Pipes; 3.5.2 Data Transfer Commands; 3.5.3 Memory Regions; 3.5.4 Generic Address Space 3.6 The OpenCL Runtime with an Example 3.6.1 Complete Vector Addition Listing; 3.7 Vector Addition Using an OpenCL C++ Wrapper; 3.8 OpenCL for CUDA Programmers; 3.9 Summary; Reference; Chapter 4: Examples; 4.1 OpenCL Examples; 4.2 Histogram; 4.3 Image Rotation; 4.4 Image Convolution; 4.5 Producer-Consumer; 4.6 Utility Functions; 4.6.1 Reporting Compilation Errors; 4.6.2 Creating a Program String; 4.7 Summary; Chapter 5: OpenCL runtime and concurrency model; 5.1 Commands and the Queuing Model; 5.1.1 Blocking Memory Operations; 5.1.2 Events; 5.1.3 Command Barriers and Markers 5.1.4 Event Callbacks 5.1.5 Profiling Using Events; 5.1.6 User Events; 5.1.7 Out-of-Order Command-Queues; 5.2 Multiple Command-Queues; 5.3 The Kernel Execution Domain: Work-Items, Work-Groups, and NDRanges; 5.3.1 Synchronization; 5.3.2 Work-Group Barriers; 5.3.3 Built-In Work-Group Functions; 5.3.4 Predicate Evaluation Functions; 5.3.5 Broadcast Functions; 5.3.6 Parallel Primitive Functions; 5.4 Native and Built-In Kernels; 5.4.1 Native kernels; 5.4.2 Built-in kernels; 5.5 Device-Side Queuing; 5.5.1 Creating a Device-Side Queue; 5.5.2 Enqueuing Device-Side Kernels; Dynamic local memory Enforcing dependencies using events

Sommario/riassunto

Heterogeneous Computing with OpenCL 2.0 teaches OpenCL and parallel programming for complex systems that may include a variety of device architectures: multi-core CPUs, GPUs, and fully-integrated Accelerated Processing Units (APUs). This fully-revised edition includes the latest enhancements in OpenCL 2.0 including:

- Shared virtual memory to increase programming flexibility and reduce data transfers that consume resources
- Dynamic parallelism which reduces processor load and avoids bottlenecks
- Improved imaging support and integration with OpenGL

Designed to work on multiple platforms, OpenCL will help you more effectively program for a heterogeneous future. Written by leaders in the parallel computing and OpenCL communities, this book explores memory spaces, optimization techniques, extensions, debugging and profiling. Multiple case studies and examples illustrate high-performance algorithms, distributing work across heterogeneous systems, embedded domain-specific languages, and will give you hands-on OpenCL experience to address a range of fundamental parallel algorithms. Updated content to cover the latest developments in OpenCL 2.0, including improvements in memory handling, parallelism, and imaging support Explanations of principles and strategies to learn parallel programming with OpenCL, from understanding the abstraction models to thoroughly testing and debugging complete applications Example code covering image analytics, web plugins, particle simulations, video editing, performance optimization, and more
