1.

Record Nr.	UNINA9910427050903321
Autore	Kusswurm Daniel
Titolo	Modern arm assembly language programming : covers Armv8-A 32-bit, 64-bit, and SIMD / / Daniel Kusswurm
Pubbl/distr/stampa	[Place of publication not identified] : , : Apress, , [2020] ©2020
ISBN	1-4842-6267-0
Edizione	[1st ed. 2020.]
Descrizione fisica	1 online resource (XIX, 465 p. 67 illus.)
Disciplina	005.265
Soggetti	Assembly languages (Electronic computers)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Chapter 1 – Armv8-32 Architecture -- Chapter 2 – Armv8-32 Core Programming – Part 1 -- Chapter 3 – Armv8-32 Core Programming – Part 2 -- Chapter 4 – Armv8-32 Core Programming – Part 3 -- Chapter 5 – Armv8-32 Floating-Point Architecture -- Chapter 6 – Armv8-32 Floating-Point Programming -- Chapter 7 – Armv8-32 SIMD Architecture -- Chapter 8 – Armv8-32 SIMD Integer Programming -- Chapter 9 – Armv8-32 SIMD Floating-Point Programming -- Chapter 10 – Armv8-64 Architecture -- Chapter 11 – Armv8-64 Core Programming – Part 1 -- Chapter 12 – Armv8-64 Core Programming – Part2 -- Chapter 13 – Armv8-64 Floating-Point Programming -- Chapter 14 – Armv8-64 SIMD Integer Programming -- Chapter 15 – Armv8-64 SIMD Floating-Point Programming -- Chapter 16 – Armv8-64 Advanced SIMD Programming -- Chapter 17 –Optimization Strategies and Techniques -- Appendix A – Source Code and Software Development Tools -- Appendix B – References and Additional Resources.
Sommario/riassunto	Gain the fundamentals of Armv8-A 32-bit and 64-bit assembly language programming. This book emphasizes Armv8-A assembly language topics that are relevant to modern software development. It is designed to help you quickly understand Armv8-A assembly language programming and the computational resources of Arm’s SIMD platform. It also contains an abundance of source code that is structured to accelerate learning and comprehension of essential Armv8-A assembly

language constructs and SIMD programming concepts. After reading this book, you will be able to code performance-optimized functions and algorithms using Armv8- A 32-bit and 64-bit assembly language. Modern Arm Assembly Language Programming accentuates the coding of Armv8-A 32-bit and 64-bit assembly language functions that are callable from C++. Multiple chapters are also devoted to Armv8-A SIMD assembly language programming. These chapters discuss how to code functions that are used in computationally intense applications such as machine learning, image processing, audio and video encoding, and computer graphics. You will: See essential details about the Armv8-A 32-bit and 64-bit architectures including data types, general purpose registers, floating-point and SIMD registers, and addressing modes Use the Armv8-A 32-bit and 64-bit instruction sets to create performance-enhancing functions that are callable from C++ Employ Armv8-A assembly language to efficiently manipulate common data types and programming constructs including integers, arrays, matrices, and user-defined structures Create assembly language functions that perform scalar floating-point arithmetic using the Armv8-A 32-bit and 64-bit instruction sets Harness the Armv8-A SIMD instruction sets to significantly accelerate the performance of computationally intense algorithms in applications such as machine learning, image processing, computer graphics, mathematics, and statistics. Apply leading-edge coding strategies and techniques to optimally exploit the Armv8-A 32-bit and 64-bit instruction sets for maximum possible performance .