Break Away with Intel® Atom™ Processors
A Guide to Architecture Migration

Lori M. Matassa
Max Domeika

Intel PRESS
Digital Edition

Digital Editions of selected Intel Press books are in addition to and complement the printed books.

Break Away with Intel® Atom Processors:
A Guide to Architecture Migration
By Lori M. Matassa and Max Domeika

Click the icon to access information on other essential books for Developers and IT Professionals
Visit our website at www.intel.com/intelpress
Notices and Disclaimers

The Intel® Integrated Performance Primitives (Intel® IPP) library contains functions that are more highly optimized for Intel microprocessors than for other microprocessors. While the functions in the Intel® IPP library offer optimizations for both Intel and Intel-compatible microprocessors, depending on your code and other factors, you will likely get extra performance on Intel microprocessors.

While the paragraph above describes the basic optimization approach for the Intel® IPP library as a whole, the library may or may not be optimized to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include Intel® Streaming SIMD Extensions 2 (Intel® SSE2), Intel® Streaming SIMD Extensions 3 (Intel® SSE3), and Supplemental Streaming SIMD Extensions 3 (Intel® SSSE3) instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors.

Intel recommends that you evaluate other library products to determine which best meets your requirements.

Intel® Hyper-Threading Technology (Intel® HT Technology) requires an Intel® HT Technology-enabled system, check with your PC manufacturer. Performance will vary depending on the specific hardware and software used. Not available on the Intel® Core™ i5-750. For more information, including details on which processors support HT technology, visit www.intel.com/technology/platform-technology/hyper-threading/index.htm.

Intel® Virtualization Technology requires a computer system with an enabled Intel® processor, BIOS, virtual machine monitor (VMM). Functionality, performance or other benefits will vary depending on hardware and software configurations. Software applications may not be compatible with all operating systems. Consult your PC manufacturer. For more information, visit http://www.intel.com/go/virtualization.

Intel® 64 architecture requires a computer system with a processor, chipset, BIOS, operating system, device drivers and applications enabled for Intel® 64 architecture. Processors will not operate (including 32-bit operation) without an Intel® 64 architecture-enabled BIOS. Performance will vary depending on your hardware and software configurations. See www.intel.com/info/em64i for more information including details on which processors support Intel® 64 architecture or consult with your system vendor for more information.
Contents

Notices and Disclaimers iii

Contents v

Foreword xi

Preface xiii

Acknowledgments xv

Chapter 1 Introduction 1
What Is Architecture Migration? 2
Motivation for This Book 2
Reasons to Migrate to Intel® Atom™ Processors 4
Intel® Embedded Design Center 5
Intel® Software Network 5
Intel® Atom™ Development Platform 5
What Are the Contents? 5

Chapter 2 Low Power Embedded Market Segments and Applications 9
Connected Services Gateway 10
Intel® Digital Security Surveillance 11
Gaming 16
Energy 16
Industrial 22
In-Vehicle Infotainment 26
Media Phone 29
Military, Aerospace, and Government 34
Print Imaging 34
Retail 35
Security 36
Storage 36
Summary 39

Chapter 3  
Intel® Atom™ Architecture for Software Developers 41
Intel® Atom™ Processor Architecture 43
Intel® Atom™ Processor Platform Architecture 54
Intel® Atom™ Processor Microarchitecture 58
Intel® Atom™ Processor Assembly Language Primer 78
Summary 87

Chapter 4  
Processor Architecture Differences, Software Implications, and Tools 89
Software Considerations Related to Processor Architecture Differences 90
Microprocessor Endian Architecture 95
Porting to the Target Operating System 115
IA-32 Interrupt and Exception Handling Model 116
Real-Time and Interrupt Latency 128
Real-Time Operation and Power Management 133
Device Drivers 134
Intel® Graphics Drivers 135
System Initialization Firmware 137
Migrating to the Intel® Architecture Instruction Set 137
Software Development Tools 139
Product Technologies 161
Chapter 5  Boot Loader Choices for Small and Fast System Initialization Requirements 169
    System Initialization Roles and Responsibilities 170
    Boot Loaders for Closed Box Designs 171
    Intel® Architecture System BIOS for Open Box Designs 172
    Intel® Platform Innovation Framework for EFI 175
    Intel® Atom™ Processor System Boot Flow 176
    Summary 210

Chapter 6  Choosing the Right Operating System 211
    Options in Choosing an Operating System 212
    System Requirements 214
    Option #1: Use the Same Operating System 226
    Option #2: Porting a Proprietary Operating System 227
    Option #3: Real-Time Operating Systems 231
    Option #4: Embedded Operating Systems 237
    Option #5: Desktop Operating Systems 241
    MeeGo 246
    Summary 251

Chapter 7  Performance Libraries for Embedded Systems 253
    Performance Library Basics 255
    Performance Library Ecosystem 268
    Intel® Integrated Performance Primitives Library 272
    Optimizing 3D Applications for Intel® Atom™ Platforms 294
    Hardware-Accelerated Video Decode 300
    Summary 301
Chapter 8  Performance Optimization 303
  Optimization Process 304
  Power and Performance Analysis Tools Overview 312
  Single Processor Core Tuning 330
  Multi-Core Tuning 342
  Power Optimization 358
  Summary 365

Chapter 9  Adopting Intel® Atom™
  Product Technologies 367
  Intel® Atom™ Processor Technologies 367
  Processor Core and Thread Features 369
  Intel® Multi-Core Technology Solutions 375
  Methods and Applications of System Virtualization Using Intel®
  Virtualization Technology 390
  Intel® G4 Architecture 405
  Summary 408

Chapter 10  Embedded Software Debugging 409
  Software Debug Overview 410
  Multi-Core Debugging for Intel® Atom™ Processor 428
  AMP versus SMP Support in an Embedded OS 434
  Debugger: Application and JTAG 437
  Whole Platform Signal and Event Debug 459
  Platform Simulation Best Practices 461
  System Bring-up 462
  Case Study: Moblin Kernel Bring-up and Debugging 466
  Summary 470

Chapter 11  Case Study: Migrating Software to
  the Intel® Atom™ Processor 471
  Ground Conditions of the Case Study 472
  Step 1: Port Code to the Target Operating System 475
Step 2: Execute Code Correctly on One Intel® Architecture Processor
Core 476
Step 3: Optimize the Code for Performance on One Intel® Architecture
Core 479
Step 4: Apply Platform-Specific Software Design Updates 483
Step 5: Optimize the Software Design for Multi-Core Intel®
Architecture Performance 499
Results 501
Summary 502

Glossary 503
References 513
Index 519
Break Away with Intel® Atom™ Processors: A Guide to Architecture Migration
Throughout the decades, Intel has taken a variety of approaches with processors and microcontrollers to target the embedded market. I was fortunate enough to have begun my career shortly after the company's introduction of the 8051, utilizing the microcontroller for one of my first engineering projects. Of course, since then, Intel has gone through many generations of embedded processors including variations of the 8086, 80186, 80386, i960, XScale, and so on. While each of these processor families has witnessed significant fame and glory, none seem to compare to the attention that the Intel® Atom™ processor is getting from the embedded industry as well as the media.

So why all the excitement about the Intel Atom processor? Unlike prior Intel architectures, the Intel Atom processor has a new architecture specially created to reduce power use. This helped the original Intel Atom processors target mobile Internet devices and is driving subsequent devices into many other areas of embedded where lower power is a key factor, as this book describes in Chapter 2, allowing these applications to expand their capabilities and scale between designs. However, this low power focus doesn't come without controversy, as Intel is targeting market segments where other processor architectures have dominated. Regardless, it will be very interesting to observe how Intel Atom processors evolve as they permeate the embedded market.

In writing the foreword to this book, I am intentionally not expressing my opinion on the qualities of the Intel Atom processor—that is for the reader to decide. On the other hand, through my work with the Multicore
Expo and Multicore Association, I have personally known and worked with the authors for many years and can testify to their due diligence on this book’s content. Furthermore, anyone undertaking an embedded design that will utilize an Intel Atom processor will benefit from reading this book, especially if you choose to go below the surface and take advantage of any of the Intel Atom processor’s optimized features.

—Markus Levy
Senior Analyst with the Multicore Insider
Twenty years ago a software developer might have asked “So what’s the big deal?” Back then architecture conversion was a routine process of designing and implementing the next “latest and greatest” product update. In my early years as a software developer, I worked for a high tech company that sold IBM-compatible mainframe and midrange computer system controllers and peripherals (terminals and printers). We started with products designed for the Zilog Z80 processor, and every line of code was written in Z80 assembly language. In the next generation of these products the C language made its debut, and all of the code was implemented again with enhanced features designed for the 68008. In the late 1980s those peripherals were extended to the early PC platform using an expansion card to emulate the peripheral. The emulated versions of these products rode the quickly advancing wave of Intel processors, catching the wave of the 8088 on the PC itself. The software would later be easily ported to the 80286, the Intel386™, and then finally to the Intel486™ processor in the early 1990s. In those days we didn’t balk at opportunities to design on new architectures. We just did it. We embraced the challenge like a breath of fresh air, realizing the prospect for innovation and extending the products’ capabilities each time. The benefit extended not only to the evolution of the end product’s offerings, but also to expanding our own engineering experience and knowledge. It was at this point of enthralment with Intel architecture that I actually became an Intel employee.
Today, with product cycles much shorter than in the old days, it's imperative to design for a computer architecture that will scale easily as industry trends advance. As with the architecture migrations that I engineered so many years ago, it's important to understand the goals and software implications before the port begins. This book is an opportunity to guide developers through the journey of migrating their software to Intel® Atom™ processor-based platforms, enabling their products to be well positioned for the future.

—Lori M. Matassa
June 2010
Acknowledgments

Architecture migration involves broad subject material and relies on domain experts for contributions and accuracy. This book involved contributions in various forms from a number of talented industry experts who we are honored to work with and who deserve acknowledgment.

For significant contributions to this book, we would like to thank Padma Apparao for power optimization information, Peter Barry for interrupts and exceptions information in Chapter 4 and Intel® architecture memory map and Master Boot Record information in Chapter 5, Pat Brouillette for SVEN information, Rajeshree Chabukswar for performance optimization information, Mark Charney for details on XED. Steve Daily for cache optimization details, Mylinh Gillen for Intel® Atom™ processor roadmap information, Sven Dummer of Wind River for operating system information, David R. Hillyard and Peter Brink for the byte swapping overhead information in Chapter 4, Praveen Jayakumar for Hardware-Accelerated Video Decode in Chapter 7, Drew Jensen for Intel Graphics driver information in Chapter 4, Sonia Leal of LynuxWorks for operating system information, Philippe Lecluse and Philippe David Verbeiren for the optimizing 3D applications for Intel® Atom™ platforms in Chapter 7, Eng Kean Lee for real-time interrupt latency information in Chapter 4, Alex Leidinger of FreeBSD for operating system information, David Levinthal for help with Intel® IPTU, Felix McNulty for the on-chip debugger tools overview in Chapter 4, Graham Morphey of Wind River for operating system information, David Randall of QNX for operating system information, Mudit Vats for the embedded pre-OS graphics information in Chapter 5, and Chris Weaver for architecture discussion and clarification.
We would especially like to thank these talented engineers for major chapter content and case studies contributions:

- Jenny M. Pelner and Jim A. Pelner – Chapter 6 topic Intel® Atom™ Processor System Boot Flow.
- Ishu Verma – Chapter 6 topic MeeGo.
- Paul A. Fischer – Chapter 7 topic Intel® Performance Primitives.
- Dave Kleidermacher – Chapter 9 case study: Methods and Applications of System Virtualization Using Intel® Virtualization Technology, and operating system information.
- Addicam V. Sanjay – Chapter 11 Case Study: Migrating Software to the Intel® Atom™ Processor.

No Intel Press book is published without peer review. For their careful reviews, constructive comments, and contributions we would like to thank Peter Brink, Peter Carlston, Garrett T. Drysdale, Peter Horn, Joshua Hort, Rob Mueller, Jim Nucci, Jayesh Patel, Karen Santa Cruz, Dale Taylor, and Michael Vierheilig, and all of the Intel Embedded and Communications Market Segment teams.

Max thanks his management, Joe Wolf and Kevin J. Smith, for support in this endeavor. Max thanks the lead author, Lori Matassa, in initially pitching the idea and receiving approval to proceed. It has been a productive and constructive partnership during the eight months of book development. Above all, Max is thankful for the support of his wife, Michelle, who was patient with the many nights and weekends consumed by this effort.

Lori thanks Intel’s Embedded and Communications Group management including Ton Steenman, Jonathan Luse, and Sam Lamagna for approving the project. A special admiration is extended to Max for his expertise and dedication, and for making this venture enjoyable and rewarding. Lori also thanks her family, especially her brothers Bob and Lance who influenced her technical curiosity and her cherished, faithful friends Sondra, Rosie, and others who provided enthusiasm and support along the way and have always been an inspiration.
About Intel Press

Intel Press is the authoritative source of timely, technical books to help software and hardware developers speed up their development process. We collaborate only with leading industry experts to deliver reliable, first-to-market information about the latest technologies, processes, and strategies.

Our products are planned with the help of many people in the developer community and we encourage you to consider becoming a customer advisor. If you would like to help us and gain additional advance insight to the latest technologies, we encourage you to consider the Intel Press Customer Advisor Program. You can register here:

www.intel.com/intelpress/register.htm

For information about bulk orders or corporate sales, please send email to bulkbooksales@intel.com

Other Developer Resources from Intel

At these Web sites you can also find valuable technical information and resources for developers:

www.intel.com/technology/rr  Recommended Reading list for books of interest to developers
www.intel.com/technology/itj  Intel Technology Journal
www.developer.intel.com    General information for developers
www.intel.com/software     Content, tools, training, and the Intel Early Access Program for software developers
www.intel.com/software/products     Programming tools to help you develop high-performance applications
www.intel.com/embedded     Solutions and resources for embedded and communications