


[HOME](#) > [TECHNOLOGIES](#) > [EMBEDDED](#) > Q&A: A LOOK BACK AND AHEAD AT PARALLEL COMPUTING

## Q&A: A Look Back and Ahead at Parallel Computing

We take a look back at the progress made in parallel computing with James Reinders, Parallel Programming Models Architect at Intel, who also shares his thoughts on how developers can get involved today, and what's ahead.

 William Wong | *Electronic Design*

Dec 28, 2015

[EMAIL](#) [SHARE](#) [Tweet](#) [G+1](#) [Recommend](#) 12

COMMENTS 0

I wanted to find out where parallel computing stands today, so I talked with **James Reinders**, Parallel Programming Models Architect at Intel. He has been involved with high-performance computing (HPC) and is a well-known author with books like *High Performance Parallelism Pearls Volume One: Multicore and Many-Core Programming Approaches*, with James Jeffers.

### What does the parallel processing landscape look like these days?

**Reinders:** The landscape for parallel processing is both maturing and rapidly evolving because there are many innovations and uses to be developed. Parallel computing is everywhere, and is therefore both an opportunity and a challenge for every software developer. The challenge, I like to say, is to “think parallel.” That is the most important aspect of being a parallel programmer—the ability to think about where parallelism opportunities are in your application, and exploit them in a way to a parallel computer’s advantage.

#### RELATED CONTENT

- » [Performance, Power, and Security Top Digital Hardware Trends](#)
- » [64-Bit ARM Platforms Target the IoT Cloud](#)
- » [Parallel Programming Tools Give Tips to Programmers](#)

### How has this changed over the past 10 years and where is parallel processing heading in 2016? Over the next 10 years?

**Reinders:** It was just over a decade ago that multicore processors appeared on the market. In the intervening decade, we have seen a shift to all parallel (multicore) processors, ranging from cellphone processors to high-end processors for supercomputers. We are even seeing hardware designs for computing that no longer hold on to the legacy requirement to be able to run serial applications. This is a huge change in the landscape, and a trend that will continue.

We are finishing a decade I will call the “dawn of ubiquitous parallel computing.” A decade ago, I often observed that the toughest time for software developers would be the short era when single core *and* multicore performance both mattered. For most programmers (outside of supercomputers), parallelism in the past decade required attention to both single-core and multicore platforms. I think we are rapidly exiting that era, and software programs which *require* parallel computing are acceptable everywhere.

	 <b>MM7150</b> Motion Module <a href="#">LEARN MORE</a>
	 <b>MM7150</b> Motion Module <a href="#">LEARN MORE</a>
	 <b>MM7150</b> Motion Module <a href="#">LEARN MORE</a>

### ED Top Articles

1. [Electronic Design's Products of the Week \(1/3-1/9\)](#)
2. [The Difference Between Accuracy and Precision](#)
3. [The Internet of Things: Hype, Hope, or Hit?](#)
4. [Building Home or Very Small Office Electronic Circuit Prototypes, Part 3](#)
5. [Dual-Junction Solar Cell Breaks Efficiency Record](#)

### Commentaries and Blogs

[The Internet of Things: Hype, Hope, or Hit?](#)

by Lou Frenzel

Posted 1 week ago  
in [Communiqué](#)



[The Best of CES 2016: Day 3](#)

by William Wong

Posted 3 days ago  
in [alt.embedded](#)



[The Difference Between Accuracy and Precision](#)

by Patrick Mannion

Posted 1 week ago





We do not need to write both a parallel and a serial version of a program to ensure acceptance. We should celebrate shedding the burden of supporting non-parallel computers, and the ability to focus on parallel computing.

Therefore, the next 10 years will be the coming-of-age for ubiquitous parallel computing, and we have multiple hardware and software innovations coming together. Hardware is being designed for parallel computing with an increasing variety of innovations to make parallel computing more affordable and higher performance. Software efforts ranging from operating systems, to tools, to language standards have all seen a decade of aligning to support parallel programming, which we can now build upon. With the hardware and software to build parallel systems on, we are going to see an unleashing of compute power we have never seen before. This era will be defined by the combination of compute power from parallel computing, with access for everyone via the cloud, combined with an industry that is now equipped and oriented to take advantage of it.

### What is the most exciting thing that Intel is doing in the parallel processing space right now?

**Reinders:** Being a leader in the democratization of technical computing and high-performance computing (HPC) is what excites me the most these days. In these matters, supercomputers serve as a bellwether for the entire computing industry. History tells us that today's supercomputer advances will be commonplace computing within a decade and everyday computing in the decade after that. The trend of democratization of HPC is very exciting. As technologies mature, Intel is increasingly finding ways to make such high-performance systems more accessible. This in turn energizes more transformational science and engineering that utilize such computers.

Intel participates in the [OpenHPC community](#) and drives the [HPC Scalable System Framework](#) along with numerous partners. Our products include [Intel OmniPath Architecture](#), Intel Xeon processors, Intel Xeon Phi processors and software development tools in [Intel Parallel Studio](#), including support for [Data Analytics and Machine Learning](#) and high performance for [Python](#) applications. These tools and programs create a platform that is not only unprecedented in performance, but also unprecedented in terms of wide accessibility. That translates to a brighter future for all of us.

### What kinds of tools does Intel offer developers to help with parallel processing/HPC?

**Reinders:** Intel offers a wide variety of tools, and since they fit seamlessly into most current development environments, people find our tools are more like an upgrade than a radical change. Most of what we offer is in a single product called [Intel Parallel Studio](#), and we have free [community versions](#) of the libraries.

What our tools do is help with three things: First, building a parallel program via our compilers and libraries. Second, debugging a parallel program via our Intel Parallel Inspector XE. Finally, analyzing parallel programs to guide design (Intel Parallel Advisor) and development (Intel VTune Amplifier). Words do not do justice to how transformative the tools in each of these categories are for software development. A wealth of information about the tools is available on Intel's website, including webinars, tutorials, and articles showing how to get started and succeed using these tools. If software developers are doing parallel computing on Intel processors, I encourage them to use our libraries and tools, and invest in maximizing their ability to "think parallel."

### You just launched the Intel Modern Code Developer Community in July. What is that program and how have people responded so far?

**Reinders:** The response has been strong, which has reinforced our belief that parallel programming has an enthusiastic following that is eager to learn more. The appetite for useful information and dialog is insatiable. The [Intel Modern Code website](#) acts as a launch pad for a multitude of opportunities to learn more and engage others interested in parallel computing. The videos alone can pull you in for hours of education from experts, including videos from our recent Intel HPC Developers Conference held in Austin. We have many more conferences and training events that provide the opportunity to join our partners and us around the world for parallel programming oriented events. Key talks from these events are often recorded and available for viewing from our website.

This year, we held the [Intel Modern Code Challenge](#) for students to optimize a brain

in [Test.Pass](#)

## Contributing Technical Experts

### JESD204B Simplified

by **Richard F. Zarr**

Published on Oct. 15, 2015

in [ADC](#)



### 11 Myths About Hardware Emulation

by **Lauro Rizzatti**

Published on Dec. 28, 2015

in [EDA](#)



### User-Programmable FPGAs: The New Frontier of Instrumentation

by **David Hall**

Published on Nov. 18, 2015

in [Test & Measurement](#)



### Power Analyzer Measures UPS Runtime

by **Bob Zollo**

Published on Dec. 7, 2015

in [Test & Measurement](#)



### Does Inductor Ripple-Current Percentage Still Matter in Low-Power Step-Down Converters?

by **Chris Glaser**

Published on Nov. 23, 2015

in [Power](#)



## Guest Blogs

JAN 4, 2016

### Building Home or Very Small Office Electronic Circuit Prototypes, Part 3 5



NOV 16, 2015

### Building Home or Very Small Office Electronic-Circuit Prototypes, Part 2 6



OCT 7, 2015

### Building Home or Very Small Office Electronic-Circuit Prototypes, Part 1 10



## Electronic Design White Papers



**Addressing the Challenges of IoT Design**  
Sponsored by *Mentor Graphics*

[Download this white paper](#)

simulation model used at CERN, which concluded with a handful of winners. Mathieu Gravey, a 25-year-old university student from France, won the highly coveted grand prize of an internship at CERN awarded by the Intel Modern Code Developer Challenge, in partnership with CERN openlab. I was pleased to meet and talk with Mathieu at our Intel HPC Developers Conference in November when we announced that he won. When asked what tools he used when optimizing the code to run in under nine minutes instead of the original 45 hours, Mathieu said his “brain” was the most important tool he used. I love it! When I teach parallel computing, I always insist that the key is to “think parallel.” Mathieu definitely believes this, too. I hope we will be able to have more contests in the future.

I cannot emphasize enough how important code modernization (using parallel computing) is, and how relatively easy it is to get benefits from the start. In fields like data analytics and machine learning, where relatively little prior work has optimized for parallel computing—we routinely see enormous speed-ups (10X, 20X, sometimes 100X) when helping developers. We have also seen some incorrect attributions of speed-ups to computer designs, implying that it has provided some unique and large advantage. The reality is the almost all speed-ups come from code modernization and can be made widely applicable—something that rests upon the skills and work of software developers. Our Modern Code efforts, highlighted on our website, is our focal point in the education and promotion of code modernization to help software developers realize faster speeds on all systems through parallel programming.

**I understand you have authored a few books about parallel programming. Can you tell us a little about them, and what they have taught you?**

**Reinders:** I love to teach, and in particular to teach parallel programming skills. It is a large topic, and there are many possible approaches to teaching. I am very proud of our two most recent books: *High Performance Parallel Programming Pearls* (Volumes 1 and 2). Over a hundred experts contributed their experiences and code in a highly accessible collection of parallel programming examples. I learned a lot as I engaged these experts to bring these books together. The books are chapter after chapter of examples on modernizing code to utilize parallel computing. The examples are real-world codes, with explanation of how these experts considered options for parallelism, made their choices, and implemented them. The source code is freely downloadable as well as another learning tool. If you are programming in C, C++, or Fortran, and tackling parallel programming for technical and high-performance computing, these books are an incredible resource for learning from a wide variety of experts.

Another book I am very proud of is *Structured Parallel Programming*. It is useful for anyone interested in learning more about how to “think parallel.” Universities have used it to teach parallel programming, and the University of Oregon has shared their lessons, slides and teaching assignments online. Our approach is to teach patterns, where a pattern is defined as a recurring way that parallel programming experts solve particular types of problems. Rather than teach a language (like a book on OpenMP might do), we teach concepts that are essential to effective parallel programming. If you do not know what a stencil is, and how to use it in parallel programming, it is a pattern you should learn. The book covers stencils, map, reduce, and much more. If programmers can take the time to step back and learn these fundamentals, I think this book can be transformative to making sure your brain is your best tool for parallel programming.



**Identifying the Ideal Mechanical PCB Hardware for Electronic Systems**  
Sponsored by Pentair Schroff

[Download this white paper](#)



**Instrumentation Challenges: Seven Essentials - How the “Internet of Things” impacts today’s instrumentation challenges**

Sponsored by Fischer Connectors

[Download this white paper](#)

[View All](#)

## Upcoming Webcasts



**Mastering Power Integrity**

Thurs, January 28,  
1:00 pm EST

Sponsored by Keysight Technologies

[Register Now!](#)

[View All](#)

## Search Parts

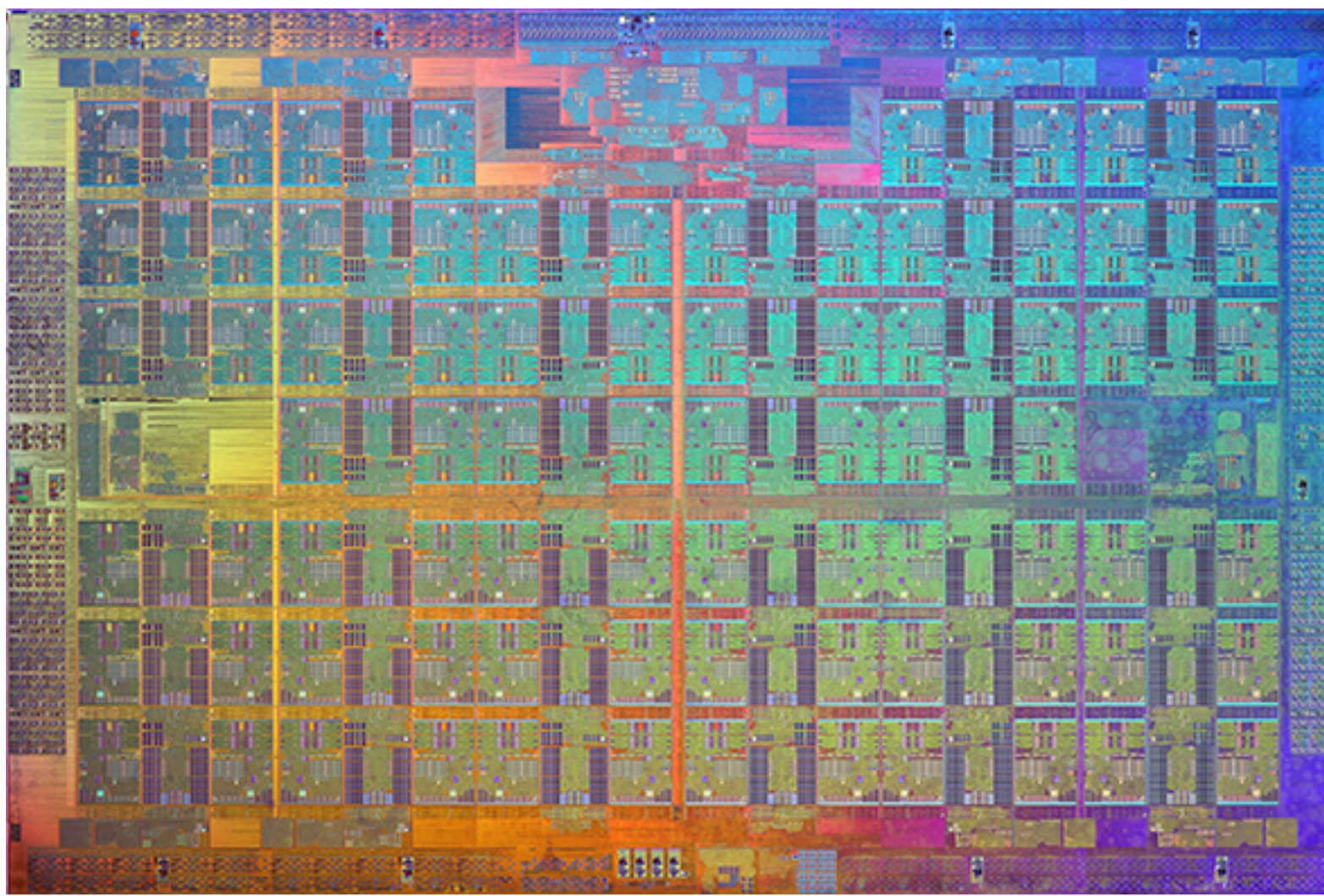


Powered by:



The Original Electronics Parts Search  
And Procurement Tool





The Xeon Phi packs in dozens of processing cores.

I am working to release a new book by the middle of 2016 focused on parallel programming for the 2nd Generation Intel Xeon Phi processor—code-named Knights Landing (*Fig. 1*). I am writing the book with Jim Jeffers, who co-wrote the first book with me on Intel Xeon Phi coprocessors, and with Avinash Sodani, the chief architect of Knights Landing. I also have the help of a dozen more people inside and outside of Intel who are working with early Knights Landing machines.

Together, these books have taught me a lot about the challenges of parallel computing. The two greatest lessons for me are that you need to make *yourself* your greatest tool by investing in learning to “think parallel,” and that parallel computing is coming of age. The tools, techniques, and systems available make parallel computing both essential and approachable in a way that is radically different than a decade ago. I see limitless opportunities ahead for software programmers who unlock the power of parallel computing for solving needs in all fields. That is what truly inspires me.

*James Reinders is an expert in the area of parallelism, Intel’s leading spokesperson on tools for parallelism, Intel’s parallel programming evangelist, and author of books on VTune (2005), TBB (2007), Structured Parallel Programming (2012), Intel Xeon Phi co-processor programming (2013), Multithreading for Visual Effects (2014), High Performance Parallelism Pearls, Volume One (2014) and Volume Two (2015).*

[PRINT](#) [REPRINT](#) [SAVE](#) [EMAIL](#) [SHARE](#) [Tweet](#) [G+1](#) [Recommend](#) 12

Please [Log In](#) or [Register](#) to post comments.

## Related Articles

[Q&A: Digi-Key’s Larson Looks Back—And Ahead](#) 2

[Q&A: James Kimery on LabVIEW Communications System Design Suite](#)

[Q&A: The Role of Big-Data Analytics in Semiconductor Production](#)

[Digital Sign Q&A With Logic Supply](#)

[FPGA Q&A: Xilinx](#)

## Search Parts

GO

powered by: [sourceesb](#)

## Newsletter Signup

Sign-up to receive our free newsletters

- Electronic Design Today** - (*Electronic Design Today offers up to date coverage of the Electronics Industry. This eNewsletter is delivered five days a week.*)
- ED Update: Power and Analog** - (*Weekly insight into critical news, products and design ideas in the power and analog space.*) [View Sample](#)
- Product Spotlight** - (*New product highlights for electronic design engineers.*) [View Sample](#)
- Engineering TV** - (*Travel the globe with ETV videos to see the latest innovative technologies in design engineering and electronics*) [View Sample](#)
- 3D Printing 360** - (*The complete source for additive manufacturing news, insights and innovations*) [View Sample](#)

**E-MAIL\***

**COUNTRY\***

Enter your email above to receive messages about offerings by Penton, its brands, affiliates and/or third-party partners, consistent with Penton’s [Privacy Policy](#).

**SUBSCRIBE**

## Newsletter Signup

Sign-up to receive our free newsletters

- Electronic Design Today** - (Electronic Design Today offers up to date coverage of the Electronics Industry. This eNewsletter is delivered five days a week.)
- ED Update: Power and Analog** - (Weekly insight into critical news, products and design ideas in the power and analog space.) [View Sample](#)
- Product Spotlight** - (New product highlights for electronic design engineers.) [View Sample](#)
- Engineering TV** - (Travel the globe with ETV videos to see the latest innovative technologies in design engineering and electronics) [View Sample](#)
- 3D Printing 360** - (The complete source for additive manufacturing news, insights and innovations) [View Sample](#)

E-MAIL\*

COUNTRY\*

Enter your email above to receive messages about offerings by Penton, its brands, affiliates and/or third-party partners, consistent with Penton's [Privacy Policy](#).

**SUBSCRIBE**

## Connect With Us



ElectronicDesign.com

[Technologies](#) [News](#) [Markets](#) [Learning Resources](#) [Community](#) [Companies](#) [Part Search](#)

[Site Features](#)

[Media Center](#)

[Newsletters](#)

[RSS](#)

[Site Archive](#)

[Sitemap](#)

[Submit Articles](#)

[View Mobile Site](#)

[Penton Corporate](#)

[About Us](#)

[Privacy Policy](#)

[Terms of Service](#)

[Contact Us](#)

Follow Us



Search ElectronicDesign.com



electronic  
design

Electronic Design Related Sites

[Machine Design](#) [SourceESB](#) [Microwaves & RF](#) [Power Electronics](#) [Hydraulics & Pneumatics](#) [Medical Design](#) [Engineering TV](#)

[Defense Electronics](#) [Global Purchasing](#) [Electronic Design Europe](#)



