



Hardware-Accelerated Computing Solutions

Software-to-Hardware Tools for an Accelerated World!

David Pellerin, CEO

Impulse Accelerated Technologies

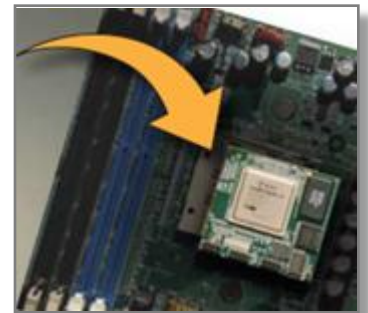
550 Kirkland Way, Suite 408

Kirkland, WA 98033

206-931-1540 (direct)

David.Pellerin@ImpulseAccelerated.com

www.ImpulseAccelerated.com



Impulse Software-to-FPGA Solutions

Software tools for FPGA programming

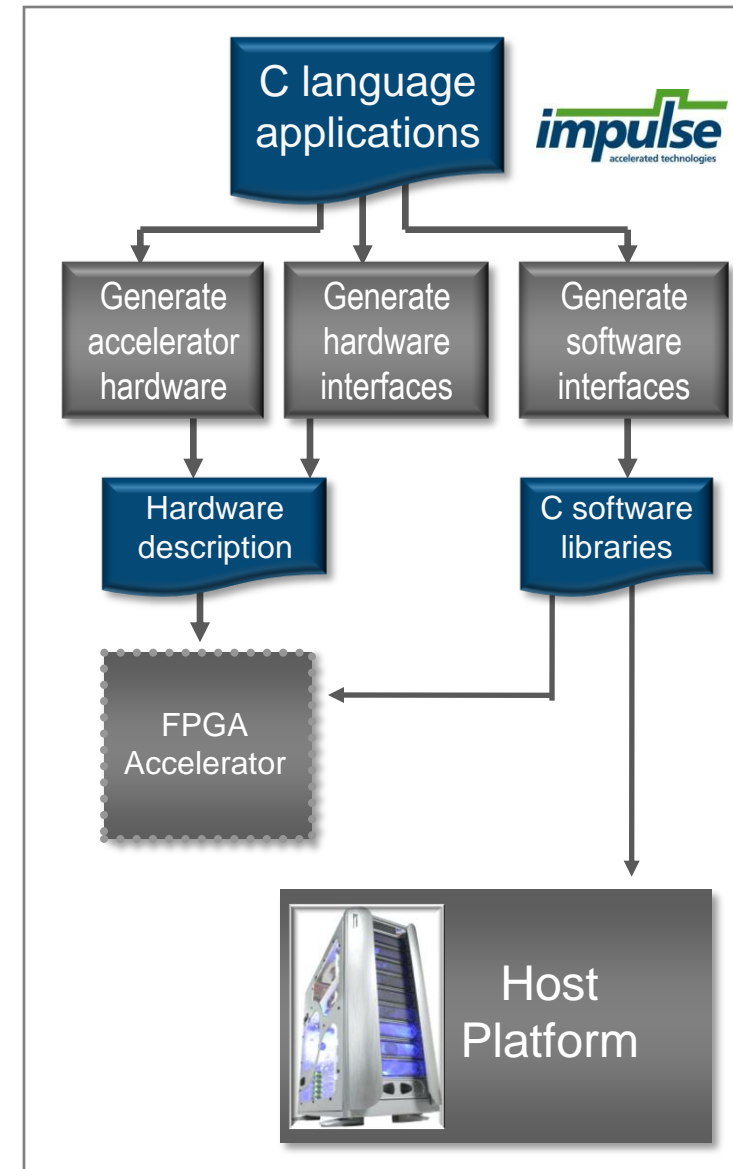
- ❑ For embedded and enterprise applications
- ❑ Allows programming in standard C language
- ❑ Enables hardware acceleration of software

CoDeveloper™ with Impulse C™

- ❑ Software-to-hardware compiler
- ❑ Optimizes C code for parallelism
- ❑ Generates standard FPGA design formats
- ❑ Also generates hardware/software interfaces

"We were able to achieve over 60X speedup of Monte-Carlo simulations, with just three months of effort. The Impulse support team has provided timely and relevant help along the way."

Rishi Khan, Research Engineer
ET International



Why Use Impulse C?

Reduced application development times

- ❑ Faster, more agile application development
- ❑ Faster time-to-prototype and reduced risk
- ❑ More opportunity for design optimization and experimentation

Reduced project costs

- ❑ Reduce or eliminate costly, high-risk hardware design phases
- ❑ Get your prototypes working faster in FPGA hardware

Example: image processing for defense/aerospace

- ❑ Advanced, embedded image processing algorithm for machine vision
- ❑ Customer saved an estimated three developer-months of effort
- ❑ Customer was able to try applications never before considered for an FPGA



From Software to FPGA Hardware

C-based design

- ❑ Emphasizing iterative methods of programming

Desktop simulation

- ❑ Using standard C tools

C-to-FPGA compilation

- ❑ VHDL or Verilog

Interactive optimization

- ❑ For high performance

The image displays three overlapping windows from the Impulse CoDeveloper software suite:

- Top Window (Impulse CoDeveloper Application Manager):** Shows a C source file named `msa_hw.c`. The code implements a sequence reading and processing function. Key features include a `#pragma CO PIPELINE` directive for performance optimization and a `for` loop that processes data in parallel using SIMD-like operations (e.g., `f0[i] = f1[i] = f2[i] = f3[i] = f4[i] = f5[i] = f6[i] = f7[i] = 0;`).
- Middle Window (Impulse CoDeveloper Application Monitor):** Shows a block diagram for the architecture named `msa`. It illustrates the data flow between a test block (`msa_test`) and the hardware block (`msa_hw`), with data and score signals.
- Bottom Window (Impulse CoDeveloper Application Monitor):** Shows a detailed hardware block diagram for `Block 21`. It features a pipeline structure with multiple 16-bit integer registers (`int16`) and adders (`+ #16`). A summary for `BLOCK 21` indicates it is a pipeline block with a latency of 12 and a rate of 12 cycles/result.

Real, Measurable Value

Increased productivity

- ❑ Supports agile, software-oriented methods of FPGA programming

Fewer errors

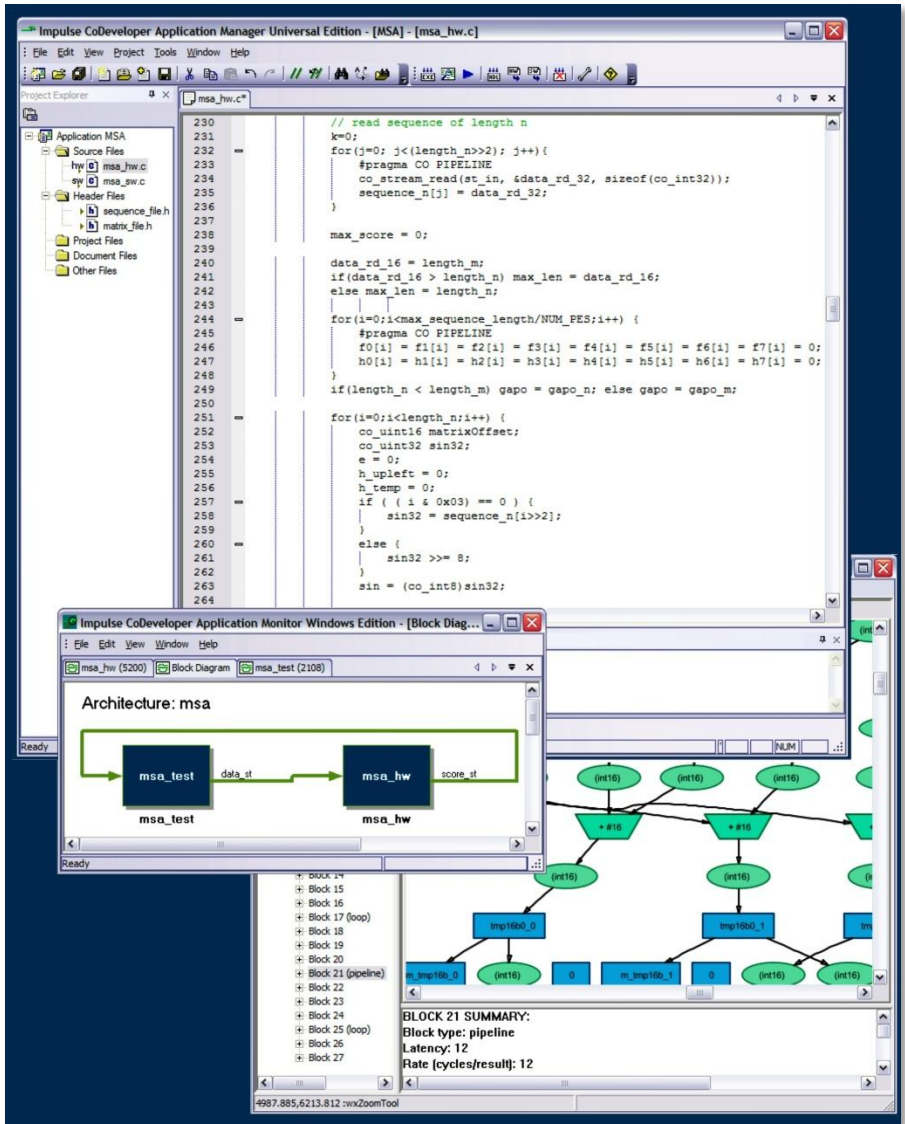
- ❑ Through faster time-to-prototype and faster design iteration

Better performance

- ❑ More computations per Watt!

“Being able to use C code, work with floating point types, and improve our performance so quickly has changed our way of designing FPGA logic.”

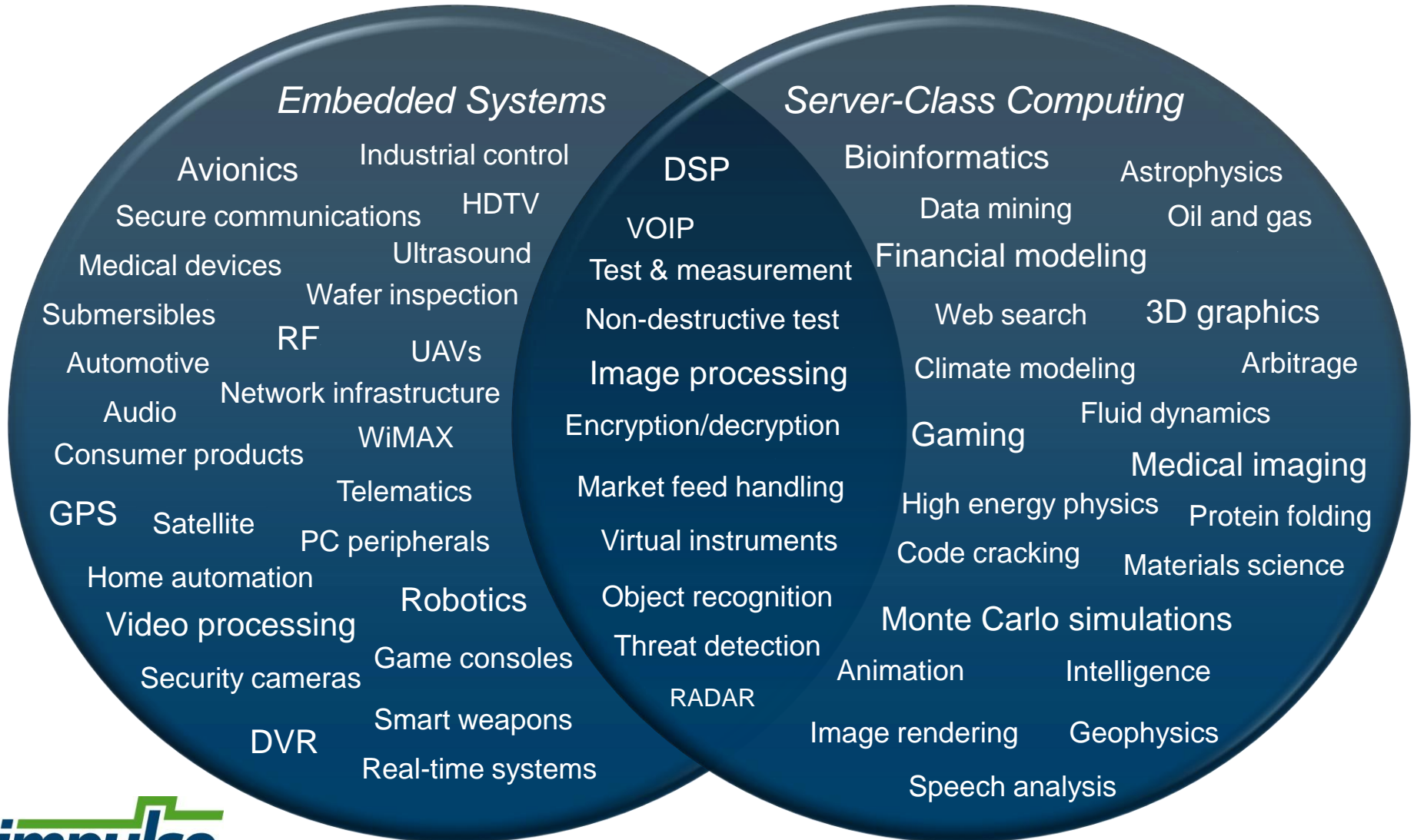
Francesco Ricci, Space Systems Developer
Xiphos Technologies, Inc.



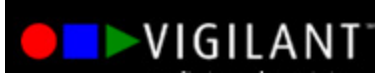
Impulse CoDeveloper™

A Wide Variety of Applications

Commercial, defense and scientific applications....



Impulse Customers Include...



National Security Technologies

Pacific Northwest National Laboratory