Accelerated Computing
Using C-to-FPGA tools

David Pellerin, CTO
Impulse Accelerated Technologies

www.ImpulseC.com
Why Software-to-FPGA Tools?

Moving applications to FPGAs
- Desktop/server applications
- Embedded processing
- Financial processing
- Image processing
- Etc


- Systems on FPGA
- DSP
- Control Logic
- Glue Logic

Software Coprocessing

Hardware accelerated processing from C-language

www.ImpulseC.com
A Wide Range of Applications

Embedded Systems
- Avionics
- Secure communications
- Medical devices
- Submersibles
- Automotive
- Audio
- Consumer products
- GPS
- Satellite
- Home automation
- Video processing
- Security cameras
- DVR
- Industrial control
- HDTV
- Ultrasound
- Wafer inspection
- RF
- UAVs
- Network infrastructure
- WiMAX
- Telematics
- PC peripherals
- Robotics
- Game consoles
- Smart weapons
- Real-time systems

High Performance Computing
- Bioinformatics
- Data mining
- Astrophysics
- Oil and gas
- Financial modeling
- Web search
- 3D graphics
- Arbitrage
- Gaming
- High energy physics
- Protein folding
- Medical imaging
- Code cracking
- Materials science
- Monte Carlo simulations
- Animation
- Intelligence
- Image rendering
- Geophysics
- Speech analysis
- Speech analysis
- Real-time systems

DSP
- VOIP
- Test & measurement
- Non-destructive test
- Image processing
- Encryption/decryption
- Market feed handling
- Virtual instruments
- Object recognition
- Threat detection
- RADAR
- Image rendering
- Geophysics
- Speech analysis

www.ImpulseC.com
Moving C to FPGAs

1. Develop and profile using standard C tools

2. Partition into multiple parallel processes
   - Arrays of processes for spatial parallelism
   - Pipelines of processes for temporal parallelism

3. Verify using standard C debuggers

4. Compile to FPGA hardware/software
   - Generate FPGA bitmap
   - Generate host libraries

5. Optimize the application to increase performance
   - Use interactive pipeline optimizer
   - Balance logic depths and clock speeds
What is Impulse C?

- C-language for FPGA programming
  - For embedded and HPC applications
  - Supports standard C development tools
  - Supports multi-process partitioning
  - Used with or without an embedded or host processor

- A software-to-hardware compiler
  - Optimizes C code for parallelism
  - Generates HDL, ready for FPGA synthesis
  - Also generates hardware/software interfaces

- Purpose
  - Describe hardware accelerators using C
  - Move compute-intensive functions to FPGAs
Perspectives

**Software-to-FPGA tools are powerful enablers**

- Appropriate for embedded applications as well as workstation and server acceleration
- Compatibility with standard programming tools and languages is critical for acceptance

**The future: what will make this even better?**

- Ongoing improvements in compiler technology
- Faster FPGA placement and routing tools
- Dynamic/partial reconfiguration including OS support
- Larger FPGAs with more multipliers