RISC-V at Bluespec

Presentation @ RISC-V workshop, Jan 14-15, 2015, Monterey CA

Rishiyur S. Nikhil and Darius Rad

www.bluespec.com

Briefly …

- Bluespec, Inc. is opening a new business front involving tightly-coupled hardware-software apps (embedded, IoT, some HPC)
  - Hardware accelerators for speed and energy-efficiency, especially for new, high-complexity algorithms (e.g., HD video, new crypto)
  - CPU/SoC essential (but raw CPU performance not crucial, at least initially)

- Our "in-house" processor components will be RISC-V
  - (Although, by using standard AXI for interconnect, customers can substitute other CPUs if they prefer.)

- We expect many of our components to be open-source
  - Our business model is focused more on pre-tested, pre-certified, pre-integrated, curated systems (similar to Red Hat model)
  - Off-the-shelf offerings, but also custom on request

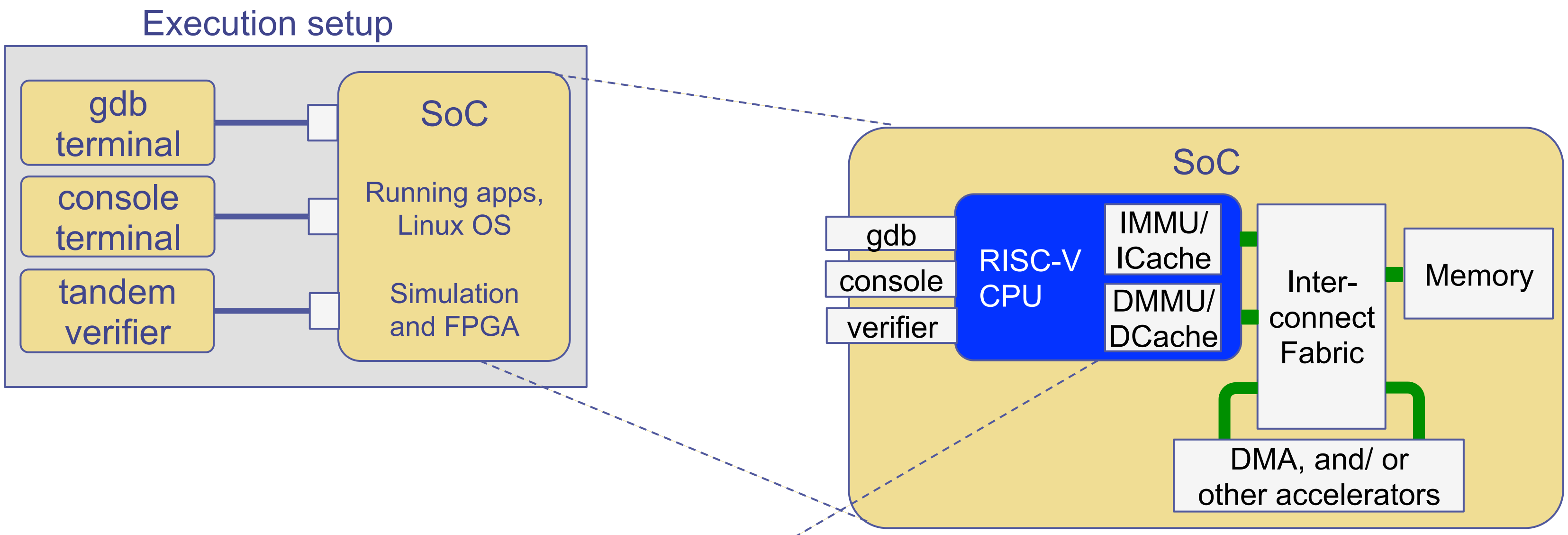
Today at the RISC-V workshop:

- This talk: quick overview of RISC-V development at Bluespec
  - Cissr simulator, BluROCS synthesizable simulator, Flute pipelined implementation
  - SoC structure around the CPU
  - GDB support, Tandem Verification

- Posters/demo later today: Showing some of these in action

bluespec

Overview of Bluespec's RISC-V components



Cissr	BluROCS	Flute
C ISS (instruction set simulator)	BSV ISS (instruction set simulator)	BSV Pipeline

(Details follow)

Current Bluespec RISC-V CPU implementations

Implementation	Cissr	BluROCS	Flute
Source language	C (some C++)	BSV	BSV
Type of implementation	ISS (instruction set simulator)	ISS	6-stage pipeline, single-issue, in-order, branch-prediction
Synthesizable? (FPGA-able?)	No	Yes	Yes
SoC connectivity	No	Yes	Yes
Linux boot time in simulation	Few seconds	Many hours	Many hours
Linux boot-time on FPGA	N/A	Seconds	Fewer seconds

All three components:
- Run ELF binaries
- Boot Linux kernel
  - Handle Exceptions
  - Have MMUs
  - Have su/user modes
- Have direct GDB support
- Support RV32 and RV64

Both run on FPGAs

Trivia:
"Cissr" = "C Instruction Set Simulator for RISC-V"
"BluROCS" = "Bluespec RISC-V Order Code[1] Simulator"
"Flute" = "A pipe that makes beautiful music"

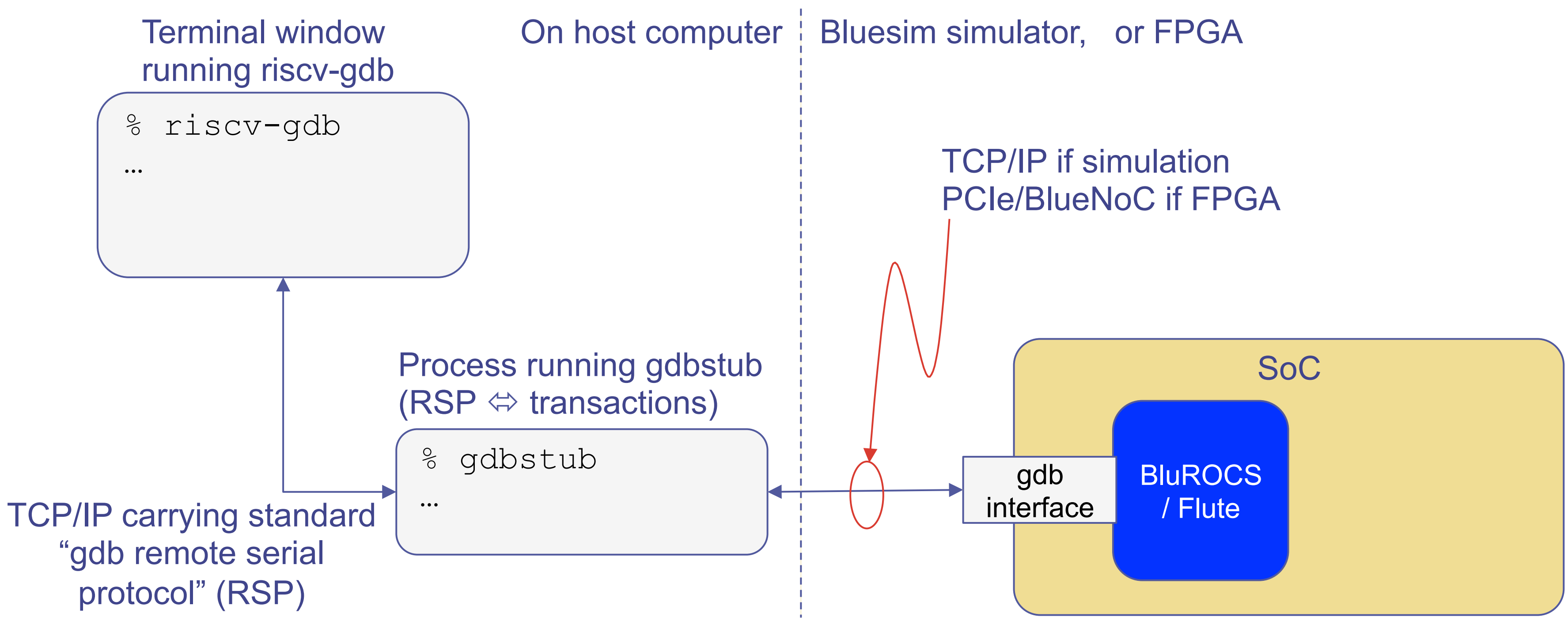
[1] Early British Computers (1940s-1950s) used the term ``Order Code'' for what we call an Instruction Set.

bluespec

Direct GDB support

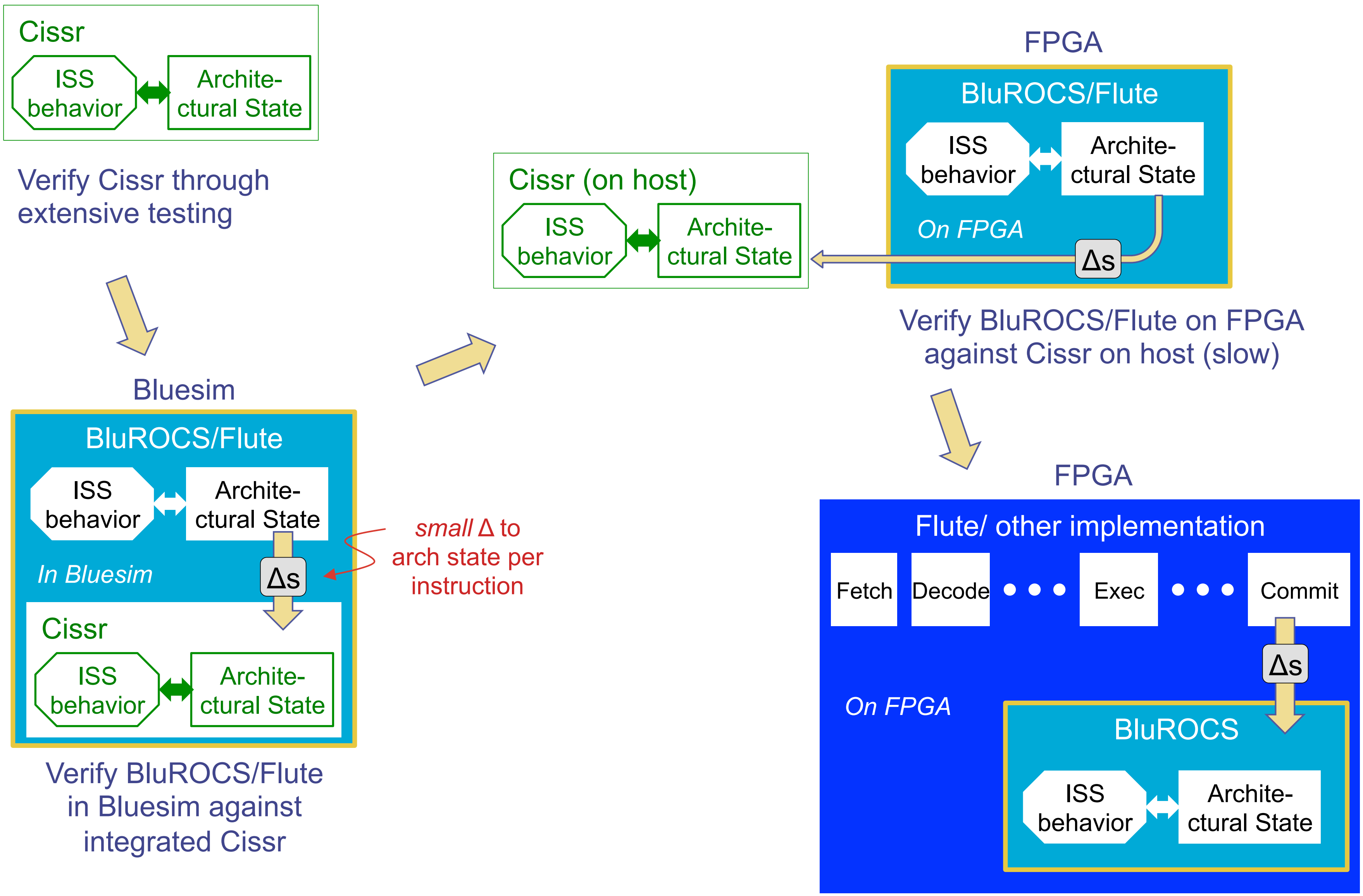
- GDB can (standardly) be run remotely, i.e., can debug a remote process
  - Uses a remote ``gdb stub'' that serves standard GDB RSP (Remote Serial Protocol) (r/w registers, r/w mem, set/rm breakpoints, continue, step, …)
  - This is all pure software
  - Perfectly adequate if your CPU/HW system/OS kernel are all stable
- But what if you're developing a CPU/system/OS kernel that are not yet stable?

- We've developed a ``gdb stub'' in hardware hooking directly into the CPU (BluROCs/Flute)
  - "Rock to stand on, while you investigate shifting sands'"



bluespec

Tandem Verification Support

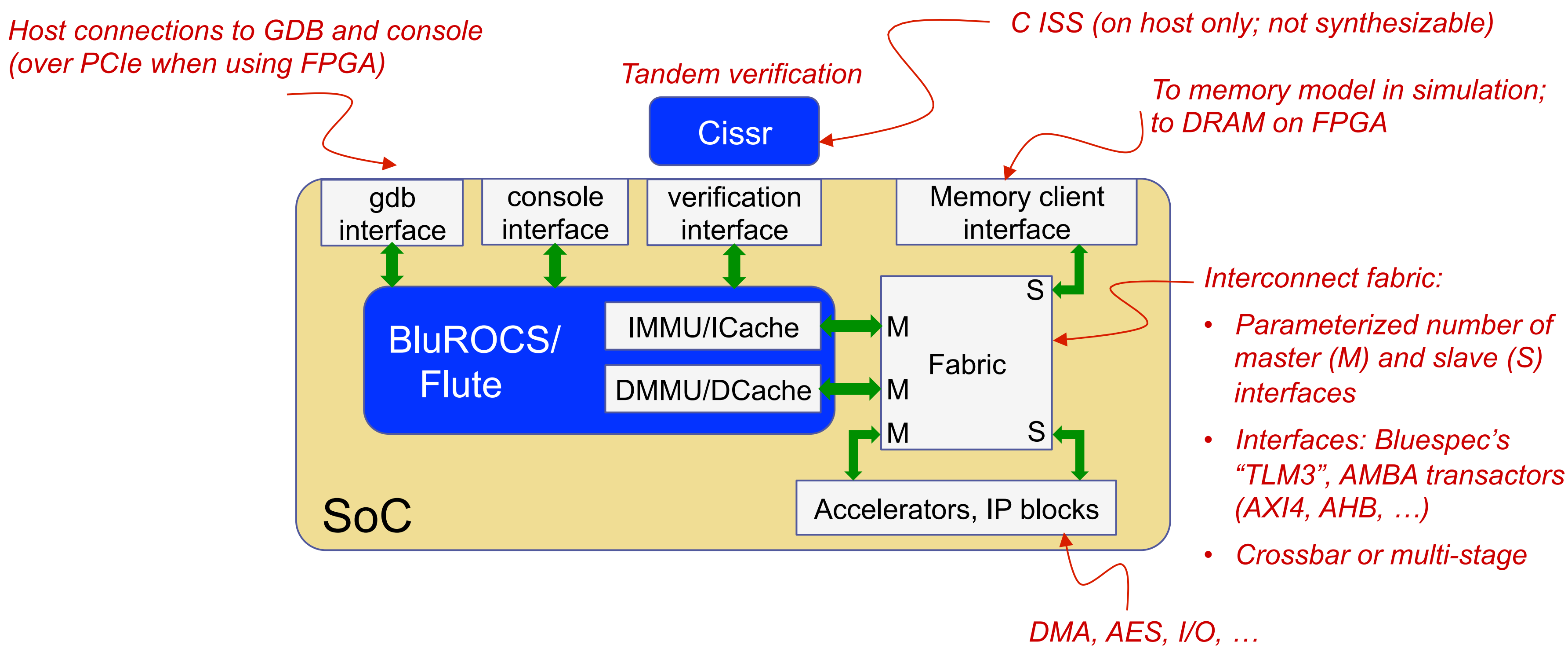


Verify Flute (or other implem) on FPGA against BluROCS on FPGA (fast)

Tandem verification catches divergence immediately and is built into BluROCS and Flute

bluespec

BluROCS/Flute and SoC structure



Written for high reuse / adaptation/ evolution

- Written entirely in BSV: excellent track record of reuse /adaptation /evolution due to
  - Very strong typing and parameterizability (Haskell-like)
  - Scalable concurrency (globally atomic transactions) with object-oriented modularity
- BluROCS and Flute have *exactly* the same HW interface (completely plug-compatible)
- Flute is written in ***latency-insensitive*** style
  - Conceptually, the pipeline stages form a "distributed system"
  - Inter-stage communication is logically "message-passing"; latency does not affect correctness
  - Easy to replace/change/stretch stages
  - Enables: evolution, physical silicon layout, and modular formal verification

Exploring Formal Verification

- For:
  - Flute (and future pipelined implementations of RISC-V)
  - Un-core components: MMUs, Caches, interconnects, coherence
  - Accelerators and accelerated computations
- Leveraging:
  - BSV rule semantics
  - Latency-insensitive style (which we believe can enable modular and therefore scalable proofs)
  - Existing past work in using FV of CPU pipelines and cache coherence protocols using BSV predecessors (with same rule semantics)
- Have had initial conversations with at least two interested parties in academia/research-labs
  - Please let us know if you have an interest in studying something like this

bluespec

RISC-V software activities

- Already made several contribs to the open-source repos:
  - riscv-gcc: mods for “soft-float” (emulated floating point) and no-atomic-memory-ops
  - riscv-gdb: port of gdb for RISC-V
  - Components of RV32 support in riscv-gnu-toolchain and riscv-linux repos
- Adapted Linux kernel to run on Cissr/BluROCS/Flute
- Next:
  - Will also be looking beyond Linux kernel to a full-blown Linux distro (Debian? Android?)
    - Also to be contributed into the open-source repo
    - Coordinate with anyone else doing this

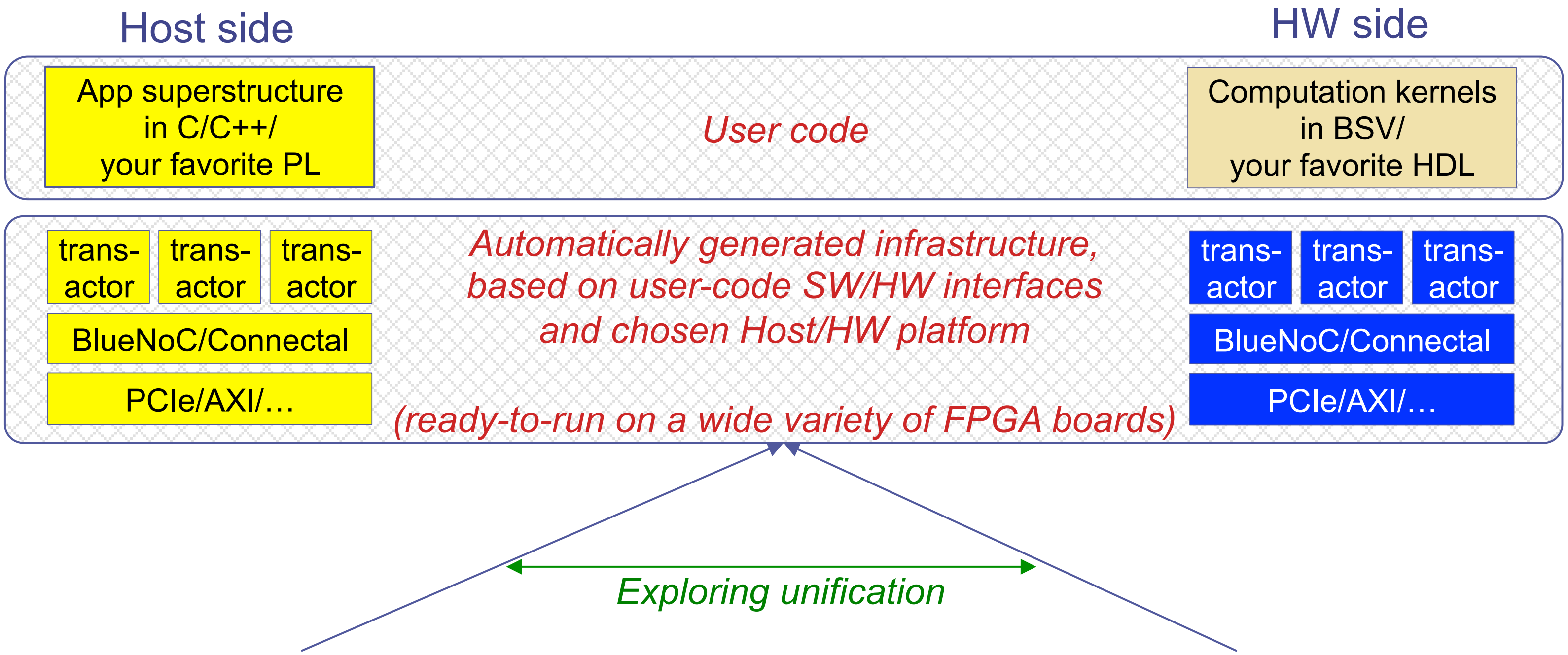
RISC-V evangelism

- Bluespec advocated and helped persuade the upcoming "India Microprocessor Project" to base itself on RISC-V (we've been talking to them about this since March 2013).
- Bluespec suggested to IIT Chennai, India, to adopt RISC-V (see Neel Gala's talk in this workshop)
- Bluespec has other partners, both commercial and R&D, with whom also we are making this case

bluespec

Next: Infrastructure for tightly coupled HW accelerators

Goal: rapid development of tightly-coupled HW accelerators for new, complex algorithms, for speed and energy efficiency



Bluespec infrastructure:

- Transactors (TLM, Get/Put, …)
- Clock control, HW breakpoints and visibility
- BlueNoC host-FPGA multipoint network
- PCIe support for many FPGA boards

Connectal: open-source BSV from Quanta

- Efficient RPC and direct memory access
- https://github.com/cambridgehackers/connectal
- *FPGA'15,* February 22–24, 2015, Monterey, CA

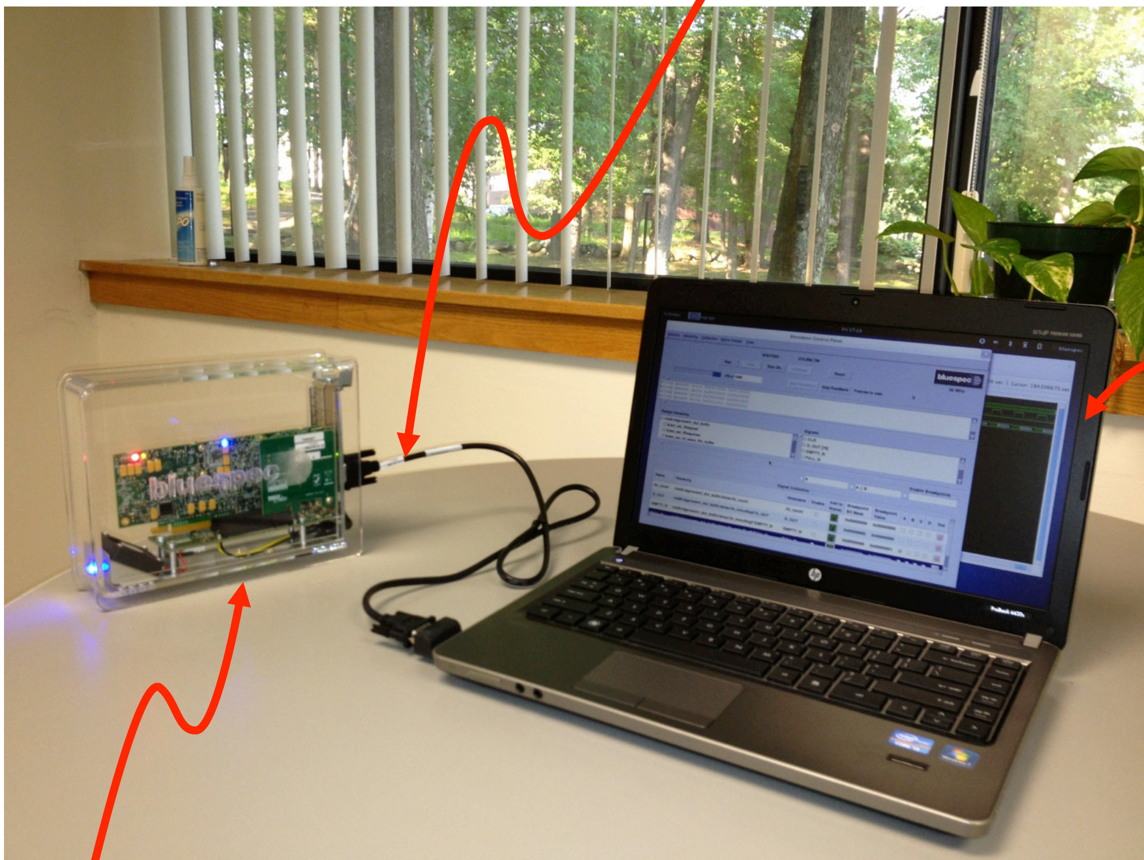
Software-Driven Hardware Development

Myron King, Jamey Hicks, John Ankcorn
Quanta Research Cambridge
{myron.king,jamey.hicks,john.ankcorn}@qrclab.com

End

System setup for today's demo is similar to this

PCIe link



Host running Linux

FPGA boards: KC705/VC707/DINI/HyperSilicon/TBD
Pictured: DINI Xilinx Kintex-7 410T ~3M ASIC gates

Today's Demo: Lenovo Thinkpad, Xilinx VC707 board
We're connecting remotely using VNC.

bluespec

Demo: BluROCS/Flute running on FPGA