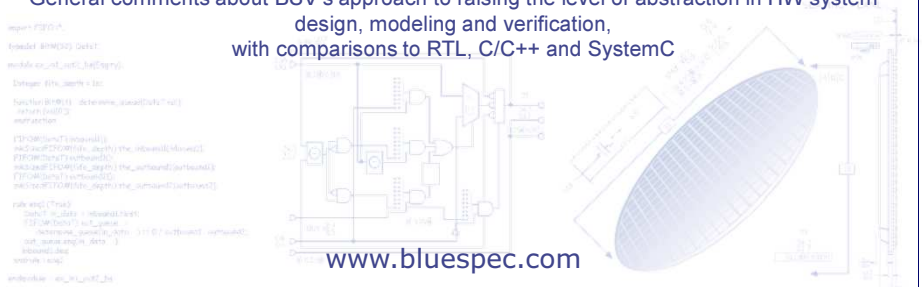
bluespec

BSV Training

Section: Introduction

General comments about BSV's approach to raising the level of abstraction in HW system design, modeling and verification,
with comparisons to RTL, C/C++ and SystemC



www.bluespec.com

© Bluespec, Inc., 2012

Bluespec: company overview

Company

- Founded in 2003
- HQ and engineering in Framingham, Massachusetts; worldwide presence
- Patented technology: proven, mature, and shipping since 2005
  - Technology roots in MIT research (atomic rule synthesis) and Haskell, a modern functional programming language (for types, parameterization, static elaboration)

Customer examples

Panasonic ideas for life, HITACHI Inspire the Next, Texas Instruments, IBM (Large networking company), Fujitsu, Alcatel·Lucent, Large search company, Samsung, ST, Large microprocessor company

And, many more!

Vibrant research partnerships

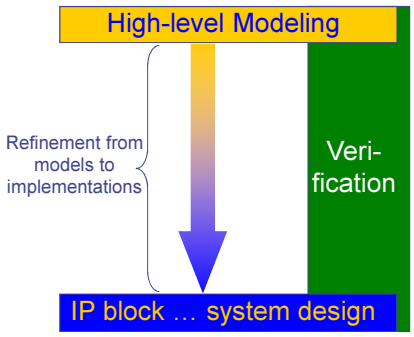
DARPA, SRI International, Open CPI, Over 50 universities teaching/doing research with Bluespec!, UCSD, Stanford, The University of Texas at Austin, Carnegie Mellon University, MIT, University of Washington

© Bluespec, Inc., 2012

bluespec

Today's key challenges in HW design

Accurate, fast models (including whole-system models) are crucial:
- *For system architectural exploration and evaluation*
- *For IP architectural exploration ("algotecture")*
- *For early start on SW development (firmware, drivers, OS, apps)*
- *For early and continuous verification during the design flow*



System and complex IP verification, too, require system models

Very complex IPs

Accuracy + speed for whole-system pre-silicon execution:
- *Necessary for system architecture evaluation (running real OSs, apps)*
- *Necessary for verification*

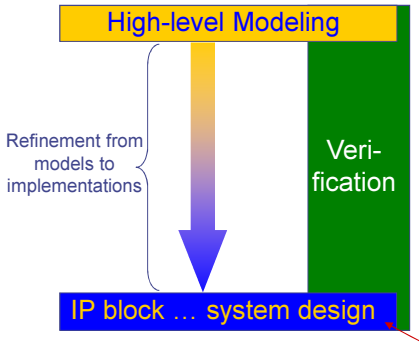
Pre-silicon execution used to mean Verilog simulation, and, for some years, SystemC simulation; today, these are 1000-10,000x too slow

Today: only FPGA-based emulation can deliver size+accuracy+speed

© Bluespec, Inc., 2012
bluespec

Bluespec's high-level synthesizable language BSV

BSV is used for modeling:
- *E.g., processor architecture models in BSV include MIPS, Sparc, x86, Itanium, ARM, PowerPC, JVM, and some newer research models*
- *All synthesized and running real apps and OSs on FPGAs*



BSV is used for verification of complex IPs:
- *E.g., transactors for PCIe Gen 3, multi-core cache-coherent processor interconnect and AXI*
- *All synthesized, and running on FPGAs*

BSV is used for complex IPs:
- *E.g., IPs in commercial mobile devices (phones/ tablets) and set-top boxes, involving both high-speed datapaths and complex control*

Other high-level synthesis tools are only used for IP design, and only for signal-processing IPS (datapath, not control)

© Bluespec, Inc., 2012
bluespec

Why not use C/C++ for High Level Synthesis?

Bad for modeling hardware parallelism
(massive, fine-grained, *heterogeneous* parallelism)

But

- C and C++ were designed and optimized for von Neumann machines (sequential execution, single large flat memory)
- High-level synthesis from C/C++ is based on automatic parallelization from analysis of Control and Data Flow Graphs (CDFGs), a technology that began with vectorization compilers in the 1970s
- 40 years of experience shows that good automatic parallelization for C/C++ only works for simple loop-and-array codes (fine-grain, *homogeneous*, SIMD-like parallelism)
  - Can't use many C/C++ high-level features in synthesizable code

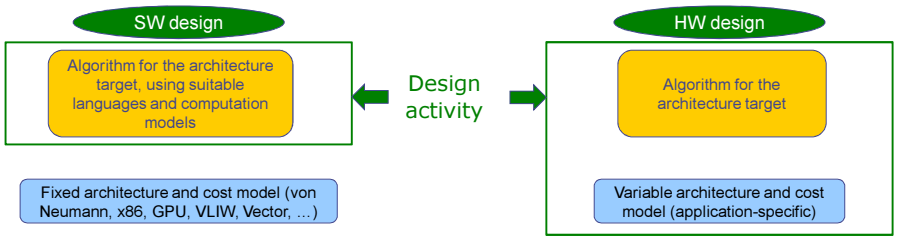
Note: SystemC adds thread parallelism to C++, but:
- The threads model or parallelism is notoriously error-prone (see E.A.Lee, "The Problem with Threads", IEEE Computer, 39:5, May 2006)
- CDFG analysis becomes *much* more complicated in the presence of threads
- SystemC is mostly used for abstract modeling, not for IP design (Transaction Level Modeling, or TLM) or HLS of IPs

© Bluespec, Inc., 2012
bluespec

Why not use C/C++ for High Level Synthesis?

Bad for describing "algotecture" (algorithm/architecture)

SW design is quite different from HW design



In HW design, algorithm and architecture are deeply linked ("algotecture").

Both need human skill and experience, and languages should support their clear expression.

C/C++ are great for sequential algorithm design on von Neumann architectures;
(not good for specifying new parallel algorithm/architecture systems)

© Bluespec, Inc., 2012
bluespec

Bluespec products overview

Core Technology
BSV language (Bluespec SystemVerilog)
and High-Level Synthesis

New approach to HLS, starting with Atomic Transactional Rewrite Rules.

Benefits:
- Universal (all kinds of digital designs, control and datapath, not just signal processing)
- IP implementation of course, ***but also***
  - Synthesizable models
  - Synthesizable testbenches
  - Synthesizable virtual platforms

Emulation
for modeling and verification
on off-the-shelf FPGA platforms
(with or without BSV)

- Full host-FPGA communications, and DUT multi-clock control
- Parameterized, reusable transactors
- SoC IP (e.g., AXI, AHB, UARTs, Memories, gdb, …)
- Signal Probes
- Hot-swap co-sim (FPGA with Verilog sim)
- Portable across FPGA platforms

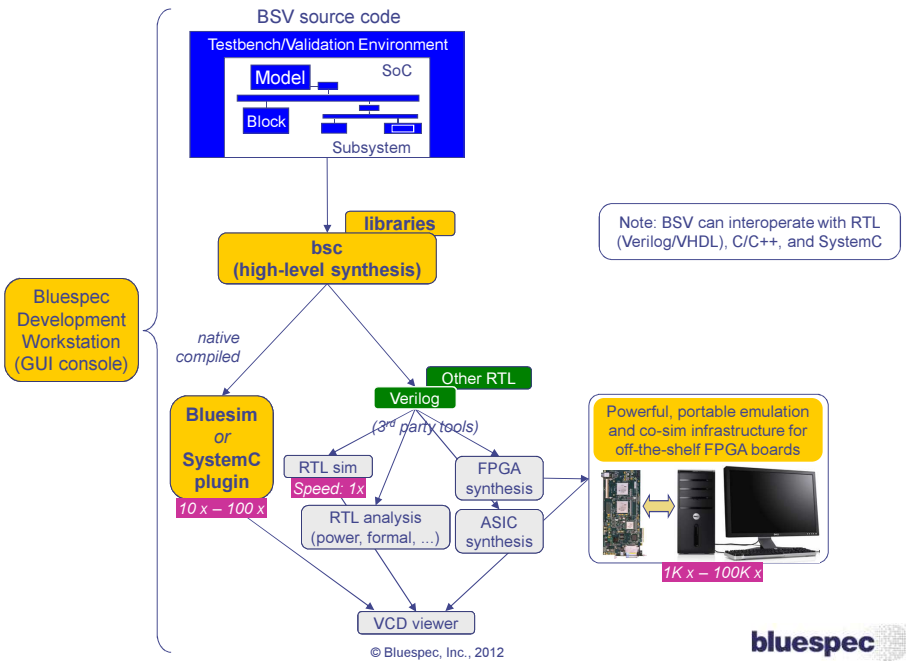
Synthesizable Virtual Platforms (SVPs)
(with or without BSV)

ARM9 | ARM Cortex | • • • | other ISAs

1000x-100000x speedups by running on FPGA platforms, compared to traditional simulation

© Bluespec, Inc., 2012
bluespec

BSV flow



© Bluespec, Inc., 2012
bluespec

bluespec

End