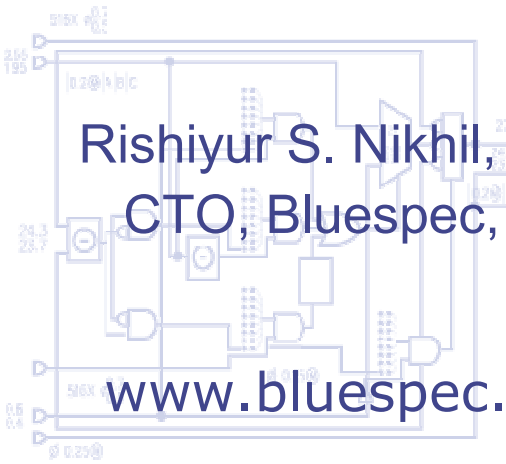
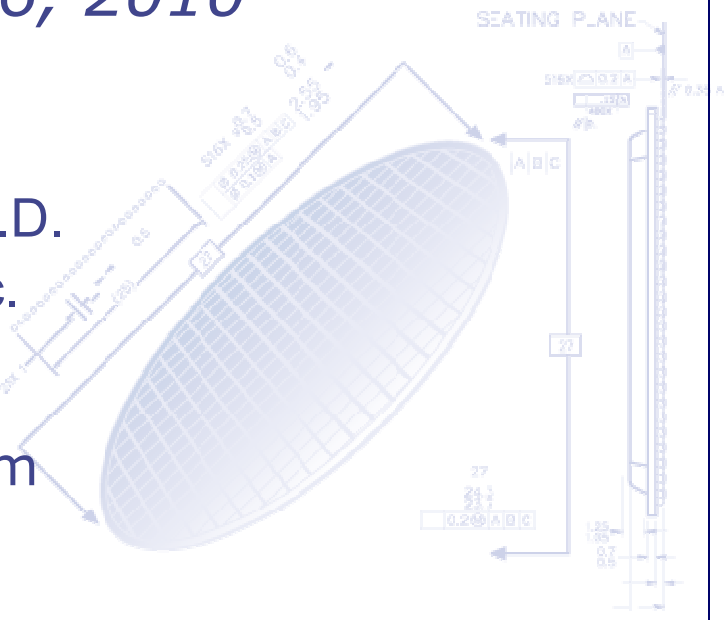
Proposals for SystemVerilog Enhancements

SystemVerilog Requirements Gathering Meeting
San Jose, February 26, 2010

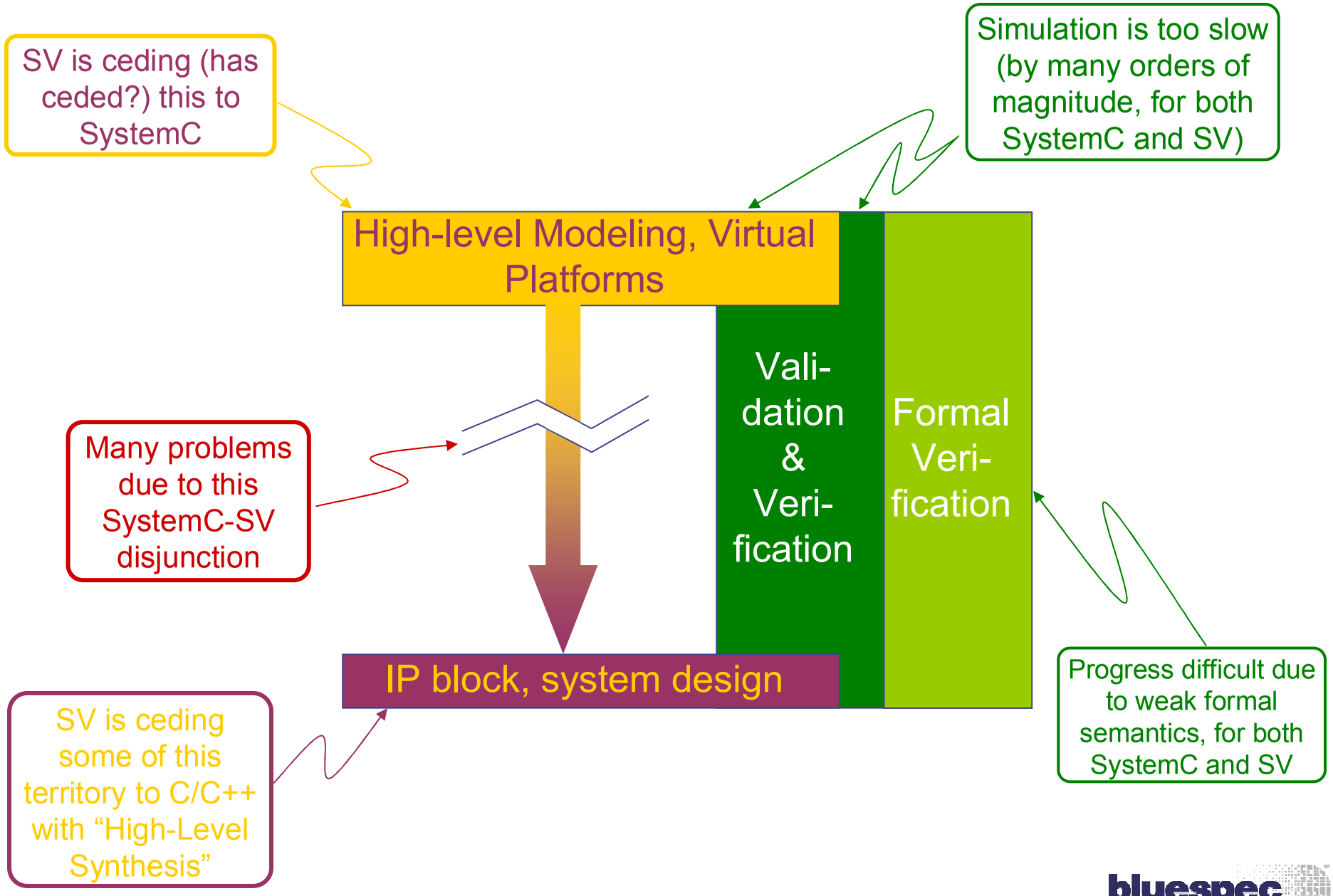


Rishiyur S. Nikhil, Ph.D.
CTO, Bluespec, Inc.

www.bluespec.com



The situation today



bluespec

Bluespec's proposal

Bluespec has enhancements ("BSV") that address all these issues (summary descriptions follow next slides)

These are not speculative—they have been tested in the field for many years with many top-tier customers

[Names available on request. Customers serve diverse markets:
Processors, Networks, Consumer Electronics, Search, and more.]

And several dozen top-tier universities and research labs: MIT, Carnegie-Mellon, UT Austin, U. Washington, BYU, U. Toronto, U.Tokyo, Seoul National U., Indian Inst. of Science, Indian Inst. of Technology, Oxford, Cambridge, MSFT Cambridge, IBM Research, RAMP project, and more

Bluespec will be happy to solicit testimonials and letters of support for these proposals from actual users in these communities

Bluespec is pleased to offer essentially all* the BSV enhancements to the P1800 standard

* All "untimed" enhancements; individual vendors make timing choices (for synthesis)

bluespec

Context behind BSV's central enhancement

Both hardware and software design are in crisis:
- SW: due to end of GHz scaling and the move to multi-cores
- HW: due to sheer size/complexity of designs due to Moore's Law

Complex parallelism/concurrency is at heart of this

Computer Science has one main potential "magic bullet" for this:

"I think we ultimately will see atomic transactions in most, if not all, languages. That's a bit of a guess, but I think it's a good bet."

Burton Smith
- *Microsoft Fellow (Parallel and High Performance Computing)*
- *ACM Eckert Mauchly Award (Computer Architecture)*
- *IEEE Seymour Cray Award (High Performance Computing)*
- *Founder/designer of seminal parallel computers (at HEP, Tera, Cray)*

BSV's central enhancement: *atomic transactional rules and interfaces*

BSV enhancements

- *Behavior*
  - "rules" with atomic transactional semantics
  - "methods" with same atomic transactional semantics for module interfaces, so atomicity composes/scales across module boundaries
  - Enhances abstraction because suppresses details about complex control logic and scheduling
  - Enhances correctness and formal reasoning because atomicity enables *invariants*

- *Structure*
  - More powerful types (lifting SV types to include functions, procedures, modules, ...)
  - Disciplined, user-extensible overloading
    - This is a fully worked out solution to what is still just a proposal in the C++ standards process ("concepts")
  - More powerful parametrization and Turing-complete static-elaboration ("generates"), including recursion

- *FSMs (using above behavior and structure capabilities)*
  - Parallel, nested, suspendable, abortable FSMs (comparable to Esterel)
  - User-written FSM generators

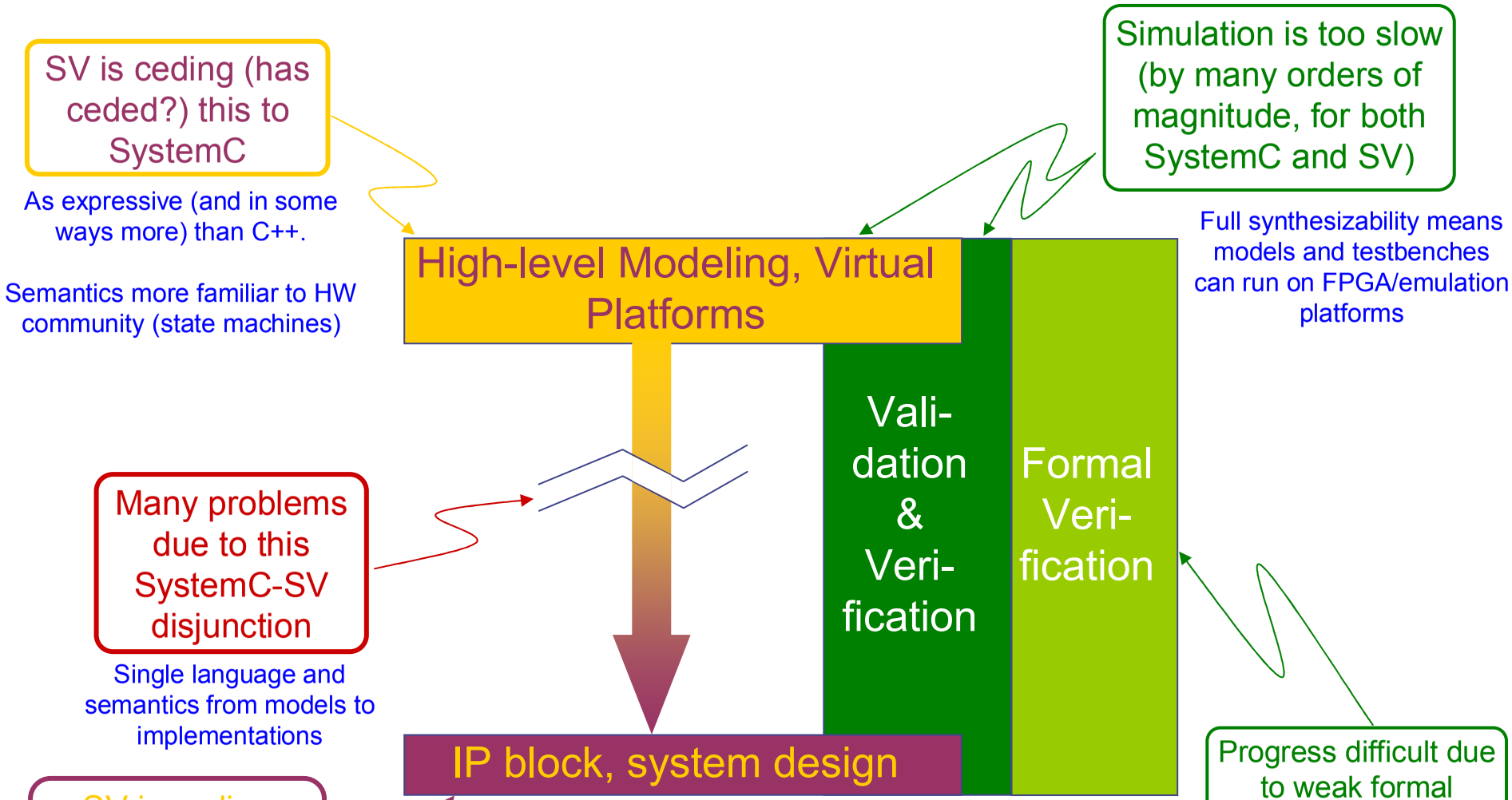
All the above are synthesizable to high-quality HW

Is it a good "fit" for SV?

- Absolutely!
  - All the proposed enhancements have been "grown organically" on existing SV syntax and semantics
  - All the proposed enhancements have been implemented and tested in the field
    - Delivered source code lines (~ millions)
    - Delivered ASIC gate designs (~ 100Ks to millions (conservative estimate))
    - Shipping ASIC gates (big multiples of delivered delivered gate designs)
  - Modules and subsystems with all the proposed enhancements have been successfully interoperated with existing HDL/HVL modules and environments
    - Verilog (1995 and 2001), SV, VHDL, 'e', ...
    - In all standard industry HDL simulators

Bluespec will work with P1800 to ensures that all these enhancements fit smoothly in SV

How do these enhancements address the problems?



In summary

Bluespec has a portfolio of enhancements to SV that

- that have been tried and tested by real and serious users
- that fit well into the semantics and "look and feel" of SV
- that fit well with existing SV code (and Verilog and VHDL)
- that opens the door to High Level Synthesis for SV

- that can dramatically improve the attractiveness of SV to
  - modelers (speed of model development, speed of execution, accuracy and refinability)
  - verifiers (speed of transactor and testbench development, speed of execution)

  in addition to raising the level of abstraction for designers

Bluespec is pleased to offer these enhancements to the P1800 standard, and to work with P1800 to ensure a smooth fit

bluespec

End

```
import FIFO::*;

typedef Bit#(32) DataT;

module ex_in1_out2_bs(Empty);

  Integer fifo_depth = 16;

  function Bit#(1) determine_queue(DataT val);
    return (val[0]);
  endfunction

  FIFO#(DataT) inbound1();
  mkSizedFIFO#(fifo_depth) the_inbound1(inbound1);
  FIFO#(DataT) outbound1();
  mkSizedFIFO#(fifo_depth) the_outbound1(outbound1);
  FIFO#(DataT) outbound2();
  mkSizedFIFO#(fifo_depth) the_outbound2(outbound2);

  rule enq1 (True);
      DataT in_data = inbound1.first;
      FIFO#(DataT) out_queue =
          determine_queue(in_data ) == 0 ? outbound1 : outbound2;
      out_queue.enq(in_data );
      inbound1.deq;
  endrule : enq1

endmodule : ex_in1_out2_bs
```