High-Level Synthesis Tools

Best-in-Class, General Purpose High-Level Synthesis (HLS) Tools

Employed by the world’s leading semiconductor and systems companies, Bluespec is the only general-purpose, high-level synthesis toolset for:

- Any use model: models, verification IP, transactors & production IP
- Any design type: control, datapath & interconnect

Our silicon-proven tools synthesize high-level descriptions of all of these to either Verilog RTL or SystemC executable models. Bluespec’s patented concurrency technology makes all types of hardware design significantly faster. The industry’s only hardware concurrency abstraction above RTL simplifies hardware design expression and enables automatic generation of control logic, accelerating development and eliminating many errors. Unlike other approaches, this makes Bluespec architecturally transparent, which is fundamental to achieving quality of results, debugging designs, working with generated Verilog RTL, leveraging emulation and predictably closing designs.

Bluespec provides the only solution that closes the gap between models and RTL implementations. Bluespec synthesizable models interoperate with RTL, can be incrementally and selectively refined to a full implementation, and allow high-speed emulation at all stages of complex IP development.

Bluespec HLS toolset comprises the following:

- **BSV language and BSC compiler** – generates synthesizable Verilog RTL from high-level models, verification IP, transactors and production IP
- **Bluesim Simulator** – delivers high-speed simulation of BSV designs at a source-level or with SystemC executables
- **Bluespec Development Workstation (BDW)** – provides an integrated GUI for the design, analysis and debug of BSV designs
- **AzureIP™ Libraries** – accelerates design with an extensive family of parameterized IP modules, types and functions
The BSV language and BSC compiler can be used to generate synthesizable Verilog RTL from high-level models, verification IP, transactors and production IP.

BSV provides a high-level of abstraction for hardware design by leveraging atomic transactions, the single most powerful tool for managing complex concurrency.

Key Features
- Synthesizable Verilog RTL or SystemC executable output
- Atomic transactions for simple expression of concurrency and automatic control logic generation
- Extreme parameterization: powerful, control-adaptive parameterization on almost any dimension: features, dimensions, types, functions, behaviors and micro-architectures
- Automated formal, transactional interfaces: correct connectivity and protocol
- Comprehensive static verification of designs to eliminate problems before simulation
- StmtFSM: high-level specification of parallel, nested sequential FSMs
- Hierarchical synthesis for unlimited capacity

Key Benefits
- General purpose, high-level abstraction and synthesis across any use (models, verification IP, transactors and production IP) and any design type (control, datapath, interconnect)
- Rapid changes for architectural exploration, feature changes, bug fixes, design-by-refinement and more
- 100% architectural transparency enables predictable, accelerated development and higher quality of results (power, area, timing, latency, performance) – and much faster exploration and debug
- Orders-of-magnitude fewer bugs for higher quality designs and less verification. One customer benchmarked a design at 1/6th the number of bugs of an equivalent RTL implementation.
- Up to and exceeding 10X productivity gains with no compromise in quality of results
- Extreme reuse through powerful parameterization, interface specifications, and much safer changes
- Ability to attack problems previously too complex for hardware description languages
- Succinct, easy-to-understand code for executable specifications, better design reviews, and higher quality

Bluesim Simulator

Bluesim delivers high-speed simulation of BSV designs at a source-level or with SystemC executables.

Key Features
- High-speed, source-level or SystemC executable simulation of a BSV high-level design
- 100% cycle accurate with Verilog RTL
- Generates standard VCD files

Benefits
- Accelerated simulation
- Verification turnarounds up to and exceeding 10X faster

Bluespec Development Workstation (BDW)

BDW provides an integrated GUI for the design, analysis and debug of BSV designs.

Key Features
- Source-level waveform viewer integration: integrates Bluespec type analysis and module hierarchy with the Verdi and GTKWave waveform viewers. Users debug with waves at the source level, including structures, bit fields, vectors and enumerated types. Keeps debug at the same level of abstraction as BSV design entry.
Simplifies the setup and management of projects
Accelerates design analysis and debug with powerful analysis tools and source-level views
For new users, simplifies tool usage and lowers learning curve
For advanced users, moves design and debug to a system level

AzureIP™ Libraries

Bluespec’s AzureIP streamlines design with a rich IP library of design building blocks, interfaces, and types, highly parameterized and full of capabilities more advanced than possible in existing design. The Foundation Library is included with the compiler. Additional libraries are optionally available.

AzureIP is self-documenting and offers interfaces for its use that allow designers to use it as a black box. AzureIP cannot be used incorrectly. It affords no ambiguity in its interfaces, but allows sufficient resilience so that the IP can mold itself to its design context. Interface methods formally guarantee that the correct interface protocol and connectivity is in place at instantiation. AzureIP is parameterized on micro-architecture, function and type, such as polymorphism. AzureIP automatically generates the control logic that is required for the IP to fit within its context. This correct-by-construction instantiation allows designers to use IP effortlessly without burdening verification. While the AzureIP Foundation Library provides a rich family of design building blocks, designers and IP vendors can augment the library with their own building blocks and benefit from the capabilities listed above.

Foundation Library Examples:

- Module connectivity and interconnects (interfaces, protocols and wiring)
- Storage (FIFOs and registers)
- Math (fixed point and complex)
- Aggregation (vectors and lists)
- Finite state machine language for sequential, parallel, condition and loop structures
- Multi-clock domain (clock synchronizers)
- Utility blocks (e.g. completion buffer)

Additional libraries are optionally available:

- PAClib (Pipelined Architecture Composers library) is a family of plug-and-play building blocks for the specification and modeling of pipelined architectures. Starting with the basic computational kernel functions of an application, one builds complex pipeline architectures in an easy plug-and-play manner by systematically composing powerful constructors from BSV’s PAClib (Pipelined Architecture Composers library). Although high level, it enables transparent and precise specification of the desired pipeline architecture, converging quickly and predictably to high quality implementations. Very powerful parameterization permits writing a single, compact source code to encompass a wide choice of architectures for a wide choice of area, speed and power.
- The AMBA library is a set of IP for developing with AHB, AXI and APB. The library includes bus transactors, interfaces and other supporting IP.

Key Benefits

- Faster time to market
- Rapid design composition, including customization and reuse
- Increased quality and decreased verification costs