Bluespec Compiler (BSC) Release Notes

2024.07 Release

Changes since release 2024.01:

Documentation

- Fix the type of `continuousAssert` in the Libraries Reference Guide (GitHub PR#686)
- Minor typo fix in the BH Reference Guide (GitHub PR#708)

Compiler

- Update source code to compile with GHC 9.10.1 (GitHub PR#705)

Libraries

- Add BuildList library, analogous to BuildVector (GitHub PR#723)

Bluetcl

- Resolve a potential compilation warning by removing the use of K&R C syntax that is deprecated in newer C standards (GitHub PR#703)

Bluesim

- Add braces to some if-statements in generated C++ modules to avoid dangling-else warnings (GitHub Issue#442, PR#691)
- Resolve a warning during compilation of the Bluesim kernel by fixing a call to `bk_clock_name` in code that is unused except by developers for debugging (GitHub Issue#698, PR#702)
- Resolve a compilation error with newer C++ compilers by updating the source code to not use a feature that is deprecated since the C++20 standard (GitHub Issue#698, PR#701)

Utilities

- Update BSV mode for `emacs` to work with newer versions (GitHub PR#697)

General

- Clean up how the `tcllibs` flags are computed in `platform.sh` (GitHub PR#703)

- This adds the version number to the flag for macOS (from `-ltcl` to `-ltcl8.5`)

Test Suite

- Add support for querying the `MACHTYPE` so that tests can support different behavior on, say, `arm64` vs `x86_64` (GitHub Issue#688, PR#690)

Internal

- Releases now built with GHC 9.6.6 (previously 9.4.8) (GitHub PR#705, PR#728)

- Updates to GitHub CI (continuous integration)

  - Retire the CI for macOS 11 (GitHub PR#700)

  - Add CI for macOS 14 (GitHub PR#690)

  - Add CI for Ubuntu 24.04 (beta) (GitHub PR#700)

  - Expand the number of GHC versions that are tested besides the version for releases — previously only a single "latest" version was being tested (GitHub PR#705)

    - Continue testing with older GHC 9.4.8, which GHCUP still labels as recommended

    - Continue testing with GHC 9.8 (updated to the latest 9.8.2)

    - Add testing with the new GHC 9.10.1

  - Support leaving the `hls_version` field blank to indicate that the HLS testing step should be skipped (PR#703)

    - This allows for testing newer GHC installations that don't yet have HLS support in GHCUP

  - Ensure that `brew` and `apt-get` are updated before installing, to avoid failures due to old GitHub runner images (GitHub PR#687)

2024.01 Release

Changes since release 2023.07:

Documentation

- Updates to the READMEs

  - Add Gentoo to the list of OSes with BSC packages

  - Update the URL for Icarus Verilog

  - Additional links to documentation

  - Link to Repology search for Bluespec packages

- Document string types in the BH and BSV Reference Guides

- Document the Generics feature and associated types and type class in the Libraries Reference Guide

- Minor cleanups and typo fixes in the BH and BSV Reference Guides and Libraries Reference Guide

Compiler

- Fix a bug in scheduling inference for action methods with arguments, when an argument is used to conditionally execute an SBR action and inference doesn't otherwise determine the method to conflict with itself (GitHub issue #641)

- Change the derived `Generic` instance for enums and tagged unions to use a balanced binary tree

    ◦ This significantly improves the speed and memory use of the compiler on designs with large enums (GitHub issue #334, discussion #414)

- Remove the warning on uses of `Generic` that the feature is experimental, now that the implementation has settled

- Use UTF-8 instead of ISO-8859-1 for all text-like I/O (GitHub PR #601)

    ◦ Source files can now include non-Latin characters!

- In BH, disallow ° and ´ in identifiers and instead parse them as symbols (GitHub PR #610)

- In BH, support more clock and reset pragmas available to BSV (GitHub issues #616, #224)

    ◦ Specifically: `gate_input_clocks`, `clock_family`, `clock_prefix`, `gate_prefix`, `reset_prefix`

- In BH, add missing features to the `properties` pragma (GitHub PR #626)

- In BH. fix the parsing of `arg_names` interface pragma to allow uppercase identifiers and disallow qualified names (GitHub issue #654)

- Fix the typecheck of struct update/literal/pattern (GitHub issue #628)

    ◦ This mostly fixes issues with qualified field names in BH, but may improve position information in error messages for BSV

- Fix the BSV parsing of naked expressions in statement and case-expression contexts (GitHub issue #646)

- Fix a failure to satisfy provisos during typecheck (GitHub issue #678)

- Update source code to compile with GHC 9.8

Libraries

- In BH, use the unicode ring operator (∘) for function composition (GitHub PR #601)

- Improve provisos in the `FixedPoint` package (GitHub PR #249)

    ◦ `FixedPoint` does not support an integer compoment with bit width less than one and this is now enforced with provisos

    ◦ The `epsilon` function also requires at least two bits in the representation

    ◦ Polymorphic uses of the `FixedPoint` type may need to add `Min` provisos (see GitHub PR #634

for example updates in the testsuite)

- Add a CShow generic instance for higher-rank fields

Bluetcl

- New `version ghc` subcommand for querying the version of GHC that the BSC tools were compiled with

Bluesim

- Fix the use of named sephamores, so that if Bluesim crashes before unlinking a semaphore, it won't cause a failure the next time Bluesim runs with the same process ID and attempts to link the same name (GitHub issue #611)

- Resolve `-Wformat-truncation` warning from GCC (GitHub PR #649)

Test Suite

- When creating an archive of log files (`archive_logs.sh`), include the C++ compiler output for SystemC tests

    ◦ The GitHub CI uses this script to upload an artifact when there is a failure

- The GHC version used to build the BSC tools is available in the test infrastructure (as `$ghc_version`), for use when the expected behavior of a test differs depending on the GHC version

- Fix the value of `$bsc_version`

Internal

- Releases now built with GHC 9.4.8 (previously 9.2.8)

- Updates to GitHub CI (continuous integration)

    ◦ Reorganization to support building and testing with a variety of GHC versions; for now, test with the version for release and with the latest version (9.8.1)

    ◦ Explicitly specify the Haskell Language Server (HLS) version to use, that is known to support the specified GHC version

    ◦ Turn off fast-fail, so that a failure for one OS version won't kill the processes testing other versions

    ◦ Support macOS VMs that don't have ghcup installed

    ◦ Support macOS VMs where the SystemC library is compiled with an unpredictable C++ standard

2023.07 Release

Changes since release 2023.01:

Documentation

- Fix the syntax for struct patterns in the BSV Reference Guide
- Update the build instructions
  - Document the `STP_STUB` and `YICES_STUB` options
  - Clarify the options for testing
- Update the test suite README
  - Add sections explaining the testing infrastructure and how to diagnose failures
  - Document how to provide additional options to BSC
  - Document how to specify the location and C++ options for SystemC

Compiler

- Improvements to VPI wrapper locations for designs with imported C functions (import-BVI) that are compiled and linked for Verilog (GitHub discussion #575, PR #576)
  - VPI wrappers are written to the same directory as the Verilog files in all cases; previously, they would be written to the current directory when the `-vdir` flag is not specified
  - BSC linking will look for VPI wrappers in the `-vsearch` path; previously, BSC would look only in the `-vdir` directory if specified or the current directory if not
- Fix the parsing of `for` loop control in the `Stmt` sublanguage, to allow register assignment with array and field selection (GitHub issue #586)
- Source code cleanups
  - Update to compile with GHC 9.6
  - Resolve most incomplete pattern warnings, enabled in GHC 9.2 (GitHub issue 469)

Libraries

- Lower the precedence of the `:=` operator in BH to match the precedence of `$` (GitHub discussion #567)
- Add a complex conjugate function (`cmplxConj`) to the `Complex` package

Verilog

- Update the Verilator link script
  - Support version 5, which requires the `--no-timing` flag
  - Remove the work directory when done, since it is not reused

- Update the Icarus Verilog link script to not generate `sft` files for newer versions (11+) as it is deprecated

Bluesim

- Remove uses of `sprintf` and replace with the safer `snprintf` or `asprintf`, to resolve warnings when building with some compilers (such as on macOS 13)

Utilities

- Improve indentation in the BSV mode for `vim`

General

- Replace deprecated `egrep` with `grep -E` as recommended by the POSIX standard, for greater portability

Test Suite

- Update to pass with Icarus Verilog versions 12 and 13
- Add an option for specifying C++ flags to use with SystemC (`TEST_SYSTEMC_CXXFLAGS`)
- Update to invoke the C++ compiler in the same way that BSC does
  - Use `c++` and not `g++`
  - Use `CXXFLAGS` from the environment (but not yet `BSC_CXXFLAGS` as BSC does)
- Additional testing and small cleanups

Internal

- Releases now built with GHC 9.2.8 (previously 9.0.2)
- Updates to GitHub CI (continuous integration)
  - Retire the CI for Ubuntu 18.04 and macOS 10.15
  - Add CI for macOS 13

2023.01 Release

Changes since release 2022.01:

Documentation

- Addition of the BSV Language Reference Guide with updates

- Addition of the BH (Bluespec Haskell/Classic) Reference Guide with updates

- Fixes in the Libraries Reference Guide, for the `Cntrs`, `Clocks`, and `BRAMCore` libraries

Compiler

- Fix to the pretty-printing of BH syntax for `letrec` and `letseq`

- Removed use of `-fpermissive` when compiling C/C++ files, which eliminates warnings when compiling with foreign imports

- For macOS 12 (XCode 14) and later, disabled chained fixups in the C++ compiler when generating shared objects, which resolves a warning about chained fixups not working with dynamic lookup

- Miscellaneous small optimizations

Libraries

- Fixes to the interface schedule for `mkSizedBypassFIFOF` (in `SpecialFIFOs`)

- Cleanup to `mkBRAMAdapter` (in `BRAM`)

- Addition of `getEvalPosition` to `Prelude`, which can be used similarly to `getStringPosition` to add position information to function error messages when a `String` argument is not available

Verilog

- Improved portability of Verilator linking by removing `bash`-isms from the shell script

Internal

- Improvements to CI (continuous integration)

- Releases now built with GHC 9.0.2 (previously 9.0.1)

For Developers

- Added support for using Haskell Language Server (HLS) on the BSC source code, via files provided in the `util` directory

---

2022.01 Release

This release supports building and running on more systems, such as CentOS 7.9, FreeBSD, Arm-based Macs, systems with Tcl 8.5, and macOS when Tcl-Tk is installed via Homebrew.

This release also includes initial support for DPI instead of VPI (for imported C functions) and support for automatic linking with Verilator (using `-vsim verilator`). Feedback on both of these

features is welcome!

Changes since release 2021.07:

General

- Update the install instructions
  - Show how to use Bluetcl to programmatically retrieve the BSC version
  - Show how to use Cabal `v2-install`
  - Show how to build a release without Asciidoctor
- Support building and running on more systems

Documentation

- Fix typos in the `MIMO` library documentation
- Document new `-use-dpi` flag
- Document Verilator as a new option for `-vsim`

Compiler

- Support optional use of DPI instead of VPI, for imported C functions (BDPI)
  - This is draft support; feedback welcome!
  - Size-polymorphic import-BDPI functions are not yet supported
  - A new flag, `-use-dpi`, must be provided when compiling and linking
- Checkout the Yices submodule at an official tagged version, 2.6.4
- Udpate the source to compile with GHC 9.2
  - Note that BSC triggers a bug in GHC 9.2.1 (#20639), which has been fixed in 9.2.2

Libraries

- Fix the modules in the `Divide` library
  - Fix bug when iterations-per-cycle is greater than one
  - Fix scheduling issues at the interface
  - Improve the provisos
- Fix divide and square root modules in the `FloatingPoint` library, to not require `-aggressive-conditions` flag for correct behavior
- Fix `Prelude` function `hexDigitToInteger`

Bluesim

- Eliminate error on exit when running on systems with Tcl 8.5

Verilog

- Support automatic linking with Verilog, using `-vsim verilator`
  - This is draft support; feedback welcome!
  - The `-use-dpi` flag is needed for designs with imported C, since Verilator does not support our VPI implementation
  - Designs with generated clocks may not link; ultimately, BSC may need a Verilator backend (separate from Verilog and Bluesim) to support arbitrary designs

---

2021.07 Release

Welcome to the first release of open BSC! Thank you and congratulations to everyone involved!

We have decided on the convention YYYY.MM for naming releases. And we have decided on a release schedule of twice a year, in January and July. Therefore, this first release is 2021.07 and users can expect a next release, 2022.01, in six months. Patch releases, if needed, will be named 2021.07.1, etc.

This release has some incompatibilities with prior proprietary releases, but for the most part remains the same. Hopefully all projects using prior releases should find it accessible to migrate to this open release. But users should expect that more incompatible changes may be coming in future releases. Examples of changes to expect include:

- Renaming and reorganizing of directories in the release
- Renaming of Verilog primitives (for example, to start with a unique prefix such as `__BSC_`)
- Renaming of preprocessor macros (for example, changing the prefix `BSV_` to `BSC_`)
- New preprocessor macros (for example, rather than having Vivado-specific versions of Verilog primitives in a separate directory, they may coexist in one file and users may need to define a macro such as `VIVADO`, to select for the target tool)
- Use of newer Verilog features (rather than restricting primitives and generated Verilog to the Verilog95 standard as much as possible)

The changes in this release are highlighted below. In addition, it is worth acknowledging the logistical and community changes. Most communication around open BSC happens on GitHub; however, we also now have mailing lists, hosted at Groups.io.

- To receive announcements about BSC and related projects, subscribe to b-lang-announce
- For questions and discussion about BSC source, subscribe to the developers' mailing list bsc-dev
- For any questions or discussion about Bluespec HDLs, using BSC, or related projects, subscribe

to

Only the core BSC tools have been included in the open BSC project (compiler, standard libraries, Bluesim, and Bluetcl). Some libraries have been released in a separate GitHub repository, `bsc-contrib`. And BDW, the Bluespec Development Workstation GUI, has been released as its own GitHub project, `bdw`. Other features from the proprietary release (such as BlueNoC, SCE-MI, and other emulation tools and transactor libraries) have not been released.

Highlights since proprietary release 2019.05:

Licensing

- FlexLM licensing has been removed from BSC and Bluesim, along with related flags
- Source is provided under the BSD-3-Clause license, except for some components where specified (under other open/copyleft licenses)

Documentation

- The documentation for standard libraries, that was previously found in the BSV Language Manual, has been collected into a stand-alone document, now residing in the `bsc` repo so that it can be updated as the libraries are updated
- BDW documentation has been removed from the User Guide and placed in its own document in the `bdw` repo; the remainder of the User Guide resides in the `bsc` repo where hopefully it can be updated as features are updated

General

- Users no longer need to set `BLUESPECDIR` — the executables will expect the directory to sit at a known location relative to the executables
- The locations for C++ libraries (SAT, VPI, Bluesim) are no longer under a CXXFAMILY directory (for example, `g++4_64`)
- Version information no longer includes a date, just a build number (usually a git hash) and a version name (now reported as a single string instead of three separate fields)

Compiler

- Removed unnecessary library requirements (X11, Tcl/Tk)
  - Previously, the BSC executable required dynamic linking with Tcl, Tk, and X11 libraries — which were legitimately needed for Bluetcl and Bluewish, but not for BSC
- Removed `Prelude` directory and consolidated all the libraries into the `Libraries` directory
- Flags and special support for BlueNoC/SCE-MI have been removed
- New flags `-show-timestamps` and `-show-version`
- New flag `-quiet` and its short form `-q`

- Yices is now the default SMT solver and the library is now included
  - Support is updated to the latest version (2.6.2)
  - Bugs have been fixed in BSC's use of Yices
- Support for CUDD solver removed, along with associated flags for scheduler effort and BDD cache size
- Better code generation for tagged unions and for enums that are non-consecutive or non-zero-based
  - Pack-unpack of types results in pure wires in more cases
  - More optimized code should occur in other situations, with fewer unnecessary case-statements
- Improved the handling of struct/union fields (in patterns, selection, and value construction)
  - The BSV parser now accepts a pattern syntax for matching structs
  - BSV syntax for struct vs tagged union can no longer be used interchangeably (users may need to add or remove the `tagged` keyword in existing code); this also means that clash between namespaces is no longer a problem
  - BH/Classic still uses the same syntax for both structs and constructors with named fields, so the type checker still uses heuristics to decide which is intended — this process has been improved
  - Parsing/type-checking is now more strict about when named vs unnamed fields can be used
  - Empty braces (without any listed fields) are disallowed in BSV syntax in situations where this does not make sense
- Record updates are now allowed on interfaces
- In BH/Classic, `prefix` is no longer a reserved keyword, and is now supported as an alternate to `prefixs` port renaming pragma
- Type-level strings are now supported, as a new string kind (alongside numeric and star kinds)
  - The pseudo-function `stringOf` exists for converting a string type to a string value (along the lines of `valueOf` for numeric types)
- Fixed some `combsched` internal errors in scheduling
- Fixed an internal error on mutually recursive type class instances
- Fixed an issue where parallel calls to BSC would conflict if they used the C preprocessor, because it created a temporary file with a hard-coded name (fixed to use a unique name now)
- Fixed a bug in static evaluation of SLE/SLT on 0-width values
- Other efficiency improvements, error message improvements, and bug fixes
  - Releases are also built with newer GHC versions, which ought to improve performance

Libraries

- Experimental support in the Prelude for datatype-generic functions, based on GHC's Generics: https://hackage.haskell.org/package/base/docs/GHC-Generics.html

- New `CShow` library (implemented with Generics), which provides a `CShow` typeclass that acts similar to `FShow` but prints values in BH/Classic syntax

- An instance of `FShow` is derived for `Either`

- The `DefaultValue` typeclass is now in `Prelude`, so it is automatically available and does not require importing a separate package

- The `guarded` parameter on FIFO primitives was fixed to be of type `Bool` rather than `Integer`

- Fix to `SquareRoot` library

Bluetcl

- The executable links with the locally installed Tcl/Tk and Itlk/Itk (rather than being compiled with source snapshots for specific versions) which also means that any locally installed Tcl libraries are available for use in Bluetcl

- The separate `bluewish` executable has been removed — now that local libraries are used, Bluetcl users can `require` the local Tk package, to pull in Tk/X11 support

- Removed unnecessary library requirements (X11, Tk)

  ◦ Bluetcl can be run on systems where Tk/X11 is not available, as long as the Bluetcl commands don't request it

- `TCLLIBPATH` and `BLUETCLLIBPATH` environment variables are supported, for listing directories to add to the search path for packages

Bluesim

- Fixed code generation for conditionally called ActionValue methods/tasks

- Improved a scaling issue in Bluesim linking

- Handles `SIGPIPE` the same as Ctrl-C

Verilog

- Fixed typos in the Quartus versions of the Verilog primitives for BRAMs

- Fixed BSC linking for Icarus Verilog, so that the Verilog search path is also used for finding preprocessor include files

- BSC linking now supported for Questa (using `-vsim questa`)

- BSC linking for ModelSim updated to remove deprecated flag