Bluespec Compiler, Bluesim, and Development Workstation

Release Notes

Version 2014.07

2014.07 Release Highlights

The 2014.07 release provides product fixes, performance improvements, and enhancements throughout the product set. New type classes, types, and functions have been added for compile-time string manipulation. Debugging has improved with flags providing more information and more control over the warnings and messages displayed. Positive resets are now supported through the Verilog library.

Documentation references are provided in parentheses. The following documents are referenced:

Reference Guide: rg

User Guide: ug

Compiler

- Improved debugging
    - -show-elab-progress: Directs the compiler to print trace statements as it expands a module. If the compiler appears to hang, this flag lets you see what the last trace was, which should indicate where in the design the hang occurs. (ug 7.14)
    - Elaboration error and warning messages now display hierarchy information. Even if you don't use the -show-elab-progress flag, the error message will now indicate the rule/submodule where the compiler was in elaboration when the error occurred. (ug 8.1.2)
    - -suppress-warnings: Provides control over which warnings are displayed by specifying a list of warnings to ignore. (ug 7.14)
    - -promote-warnings: Provides control over the severity of warnings by specifying a list of warnings to report as errors. (ug 7.14)
    - -demote-errors: Provides control over the severity of errors by specifying a list of errors to report as warnings (when possible). (ug 7.14)

    - New flag to help write importBVI statements: -show-method-bvi displays the method conflicts in a format that can be cut and pasted into an importBVI statement. (ug 7.14)
    - –show-method-conf flag is now off by default (ug 7.14)

- Scheduling attributes can now refer to methods in addition to rules. This only applies to method at synthesis boundaries. Attributes that refer to method names on non-synthesized modules are not supported, but you will get a warning saying the attribute is being ignored.
- Positive reset is now supported
    - There is a new Verilog macro BSV_POSITIVE_RESET. If this macro is set, the sense of reset is changed from asserted low to asserted high. This is a global switch; it is not possible to support mixed resets within a single design. (ug 4.3.3)
    - The name of the reset remains RST_N for generated modules. This name can be changed with a new bsc flag -reset-prefix. The flag has the same behavior as if the reset_prefix=<name> attribute were applied to all synthesized modules. (ug 7.9, 9.1.1)
    - To use a positive reset the following flags are recommended for both the compile and link stages: -reset-prefix "RESET_P" -D BSV_POSITIVE_RESET
    - The reset names in the Verilog library have been changed to a generic RST.
- New Verilog preprocessor macros (ug 4.3.3)
    - BSV_POSITIVE_RESET: Bluespec's Verilog primitives have been updated to allow for positive reset signals.
    - BSV_ASYNC_RESET: Allows an asynchronous reset for FIFO packages and packages using the Counter.v Verilog primitive
    - BSV_RESET_FIFO_HEAD: Allows reset on the head element of a FIFO
    - BSV_RESET_FIFO_ARRAY: Allows reset on array elements of a FIFO
- New environment variables BSC_OPTIONS, BLUETCL_OPTIONS to set flags for bsc, Bluetcl, Bluewish, or Development Workstation (ug A.3)
- Other new flags added
    - -verilog-filter to invoke a command or file post-process the Verilog file generated by the compiler. The flag can be used multiple times. (ug 7.3)
    - -systemc instructs the linking stage to generate a SystemC model instead of a Bluesim executable. (ug 7.4)

    - –fdir specifies the working directory for relative file paths during elaboration. This is used to specify where relative file paths should be based from in calls to openFile. (ug 7.7)

- The checking of always-enabled attributes on submodule methods is now performed prior to the backend split. Before, it was only checked in the Verilog backend; now, it is checked for both Verilog and Bluesim, and using an SMT solver (which improves analysis). When the solver can prove that the method is never enabled, this is reported as an error. This error can be demoted to a warning with the new -demote-errors flag.
- The compiler will display a warning if it tries to load a file from a search path and finds a file by that name in multiple directories.
- Fixed a bug in the scheduling analysis for the uncommon case of an ActionValue method whose return value is computed from an argument but whose body contains no action or only actions that do not conflict with each other.

Performance Improvements

- Improvements to elaboration and code-generation for array/vector indexing. Some corner case examples now compile significantly faster. You may also see cleaner Verilog code for dynamic array indexing that isn't a power of 2. The don't-care value for out-of-bounds indexing is retained through to code generation, where it can be X or whatever value specified with the –unspecified-to flag.
- Speed improvements in type check. Some corner case examples now compile significantly faster.
- Improved speed of scheduling analysis by replacing the BDD solvers with SMT solvers. The default solver is STP, which is supplied. You can also direct the compiler to use Yices, if you have access to it. The scheduling state should no longer be a common bottleneck for compile time; if you still have an example, please submit it to Bluespec support. (ug 7.13)
- Improved speed and memory use in the SystemVerilog preprocessor, on large files.

AzureIP Library

- All reset names in the Verilog library have been changed to a generic RST. This means local copies of Verilog libraries must be updated or they will not work.
- New Cntrs package which implements typed and untyped up/down counters. (rg C.8.4)
- Printf library that defines sprintf for generating formatted strings (rg C.8.18)
- New BuildVector package for advanced vector construction. (rg C.8.19)

- New CReg concurrent register modules (also known as EHR), for sequencing multiple actions on a register in a single clock cycle (rg B.4.2)
- Improved implementations of mkPipelineFIFO and mkBypassFIFO using the new CReg

Prelude Enhancements

- Compile-time string manipulation has been added
    - New String Literal data class (rg B.1.15)
    - New Char data type with functions for Char manipulation (rg B2.8)
    - Functions for String manipulation (rg B2.7)
    - Functions for getting the name of the package being compiled (genPackageName) and the module being synthesized (genModuleName) (rg B.6)
- Elaboration-time file IO operations have been added. You can open a file for reading, writing, or appending with openFile, which returns a Handle. A number of functions have been added for putting and getting strings and characters to the Handle. You can also control buffering and test for properties such as EOF. (rg B.7)
- FShow type class has been moved to the Prelude package and deriving FShow has been added. FShow for Maybe type is now consistent with other tagged unions by displaying tagged keyword. (rg B.1.14)
- Functions toGPClient and toGPServer added to the ClientServer package (rg C.7.3)
- New Array data type, as an alternative for explicitly giving the type of array values, which previously could only be expressed with [] syntax. (rg 8.1, B.2.13)

Bluesim

- Added support for multiple Bluespec models in a single SystemC simulation. The Bluesim schedule.cxx file has also been named model_<topmod>.cxx so that the files for multiple models can exist in the same directory without overwriting the schedule file each time. (ug 4.3.1)
- Added the ability to compile multiple Bluesim models in the same directory. Added the -parallel-sim-link command-line flag for forking several parallel calls to the C++ compiler during Bluesim linking, which can speed up that stage if the C++ compiler is taking a long time. (ug 7.4)
- Simulation speed improvements: Designs with signals wider than 64 bits may see a speedup (as much as 20%)

Bluetcl

- The bluetcl::flags set command now supports multiple arguments, separated by spaces. Any scripts using {} to provide multiple flags should be modified by removing the brackets. (ug B.4.2)

SceMi

- Added SceMi 2.1 Pipes Support (ug G)

Contrib Library

- A new directory, $BLUESPECDIR/BSVSource/Contrib has been added containing libraries which may be helpful to users, but are not supported or systematically tested by Bluespec. These libraries are provided on an as-is basis, and may include third-party contributions.

Documentation

- emVM changed to Emulation App
- Corrections and updates have been made in all manuals