Proposal for Real number enhancement in SV for analog behavioral modeling

Tapan Halder, Arturo Salz - Synopsys

Introduction

There is increased need for real numbers modeling to support the development of analog or mixed signal functional models that can take advantage of the simulation performance offered by digital simulators. This type of modeling is very useful to investigate system characteristics before transistor level implementation, and to verify the connectivity of the entire system, thus, enabling a top-down design methodology. Real valued models that are simulated entirely within the event-driven (digital) simulator offer tremendous speed-up, thus, enables designers to do intensive verification in a short period of time. In addition, use of digital simulators enables more advanced methodologies to be put to use, such as dynamic random testbenches, assertions, and functional coverage. However, since there is a trade-off between performance and accuracy, engineers would like to swap models with different performance/accuracy characteristics. For example, once a spice model is available, engineers will typically swap a real valued model for its Spice model to check at a higher level of accuracy, but they would like to reuse the same testbench.

SystemVerilog's existing real number capabilities are not powerful enough to model common analog behavior and require enhancements. This document describes the key changes needed in the SV language to support real valued models natively.

1. **Support Multiple Drivers and Resolved Real Type**

   This is a must have feature for proper modeling. Although VHDL language supports user defined resolution function, these do not exist in SV. Instead, the language supports several predefined resolved net type (wire, wand, wor etc.). Built-in resolution functions have performance advantages, and are generally easier to use (by end-users) since all the rules that guide inter-type interactions are predefined, and are easily portable. In keeping with this approach, it is best to introduce a few new built-in types such as **voltage**, **current**, and **frequency**, which can be used to model a single analog wire with predefined resolution semantics.

2. **Support X and Z state with voltage/current**

   In addition to the regular real (floating point) values, there is a need to support 4-state values for analog signals, that is **X** (unknown, undetermined), and **Z** (tri-state, high impedance). In particular, the Z value is needed for proper modeling of multiple voltage drivers. It is not possible to change the existing SV real data-type without creating backward incompatibles. Hence, we propose the introduction of new 4-state floating-point types, along with the specification of the rules that govern interactions between all real data-types – the new 4-state types and the existing 2-state **real** type. Note that this extension will normalize the language across integral and floating-point types: Currently SV's integral types support built-in 4-state and 2-state representations (i.e., logic and bit) with pre-defined rules for converting one to the other.

3. **Allow INOUT ports with voltage/current resolved type**

   SV real type cannot be an INOUT port. This limitation is resolved by providing for a Z value that allows switching the signal direction. For proper modeling, ports with all three directions need to be supported: IN, OUT, and INOUT.

4. **Allow SV real and voltage/current connection**

   Specify the rules for using 4-state voltage/current in conjunction with the existing SV real type, which can be used to model analog variable. Since the 2-state **real** type does not support X and Z information, implicit type conversion functions between the 2-state and 4-state types will be provided. For example, X and Z can be mapped to 0.0 or infinity.

   Note that for proper modeling, both variables and signals (wires) need to support 4-sate as well as 2-state information, otherwise, X and Z information will be lost when assigning those values to variables. We propose the use of the **analog** type to represent a generic 4-state floating-point variable – which has no resolution semantics. All mathematical operators will be defined for handling analog the type, which includes X and Z values.

5. **Support for real complex types**

   In a manner consistent with the language defined for integral type, real types must be supported by all user-defined aggregate types, such as arrays and structs.

6. **Support voltage/current to Spice and Spice to voltage/current connection in mixed signal verification**

   Note that with just 4-state real number, it is difficult to distinguish between voltage and current. Explicit voltage/current type will allow correct semantics between voltage and current and ensure that the proper interface element is inserted in the analog/digital boundaries. Need to enhance the connect rule to allow usage of type (besides discipline) in choosing the correct connect module.

7. **Support $table model in digital domain**

   Enhance the $table model to allow real/voltage/current as arguments.

8. **Initialization of voltage/current**

   Define the initial value of 4-state analog wires and variables. Variables of type real are initialized to 0.0, but variables of type analog are initialized to X, voltage wires are initialized to Z, and current wires are initialized to 0.

9. **Usage of voltage/current (4-state real) in real expression**

   All math operators and functions must support the new 4-state real types. Typically, arithmetic expressions involving X and Z states will result in an X value, in a manner consistent with the digital behavior. However, if a certain mathematical operation can be reduced (as the logical digital operators) then a proper value – other than X will be defined.

   Signed numbers as well as the $realtobits and $bitstoreal must properly support the new 4-state types.

10. **Usage of 4-state real in operators**

    Allow the following relational operators:

    - >, <, ==, ===, >=, <=
    - 4-state real and SV real numbers can be compared with signed and unsigned integers
    - These operations involving ‘X’ or ‘Z’ values will result in ‘X’ similar to integers
    - The ternary operator must be supported
    - The logical operators &&, || may need to be supported similar to integers.

11. **Semantic check for illegal usage of operators**

    - Bitwise operators: &, |, ^, ~, ~&, ~|, ~^
    - Reduction operators: &, |, ^, ~, ~&, ~|, ~^
    - Shift operators: >>, <<, >>>, <<<
    - casex and casez

12. **Voltage/current and wire/tri connection**

    Don’t allow direct connection. Better to use manual realToLogic and LogicToReal interface element between these types at port boundary. A real value cannot be represented in a single bit wire.

13. **VCD Support**

    Enhance VCD to support 4-state real

14. **Force/Release/$deposit on 4 state real (voltage/current)**

15. **VPI enhancement for 4state real (vpi_put_value/vpi_get_value)**