## Open Issues in Verilog-AMS LRM2.1 version:

The following document lists all the open issues as identified in AnnexG of the LRM2.1 version and also includes other open items that have been raised by the users of the Verilog-AMS language.

**Category:** Each issue is identified by the following categories
 BNF - The Verilog-AMS BNF needs to be corrected - incomplete/incorrect syntax
 Enhancement - Extension of the current language features
 Ambiguity - Semantics have not been clearly stated
 Cleanup - General fixes, typos and other minor corrections

**Owner:** Responsible for correcting the LRM and proposing amendments/changes to address the issue. Currently ownership is idenitifed on a company basis and the active participants are Motorola, Cadence and National Semiconductors. The committee representatives from these groups are:
 Motorola - Graham, Sri
 Cadence - Martin, Jon
 National Semiconductors - Kevin

**Version:** The version in which the fix is going to be addressed. The table below has idenitified short term goals for LRM2.2 and the rest have been identified as future.

This table is just an initial proposal in terms of ownership and version and these would be discussed as part of the AMS Committee meetings. Additional issues would be added on to this table whenever they arise. This document will be updated on a regular basis depending on the feedback for particular issues to reflect the current status.

## Table 1: List of Open Issues

| Item | Description/Issue | Category | Owner | Version |
|---|---|---|---|---|
| 1 | **wreal** declaration and usage should be synchronized with System Verilog language.<br>The BNF allows a **wreal** declaration without any identifier, for example, **wreal;**<br>Change syntax from<br>`wreal_declaration := wreal [list_of_identifiers];`<br>to<br>`wreal_declaration := wreal list_of_identifiers;` | BNF | Motorola | LRM2.2 |
| 2 | Verilog-AMS has no language extensions to support back annotation. | Enhancement | National Semiconductors | LRM2.2 |
| 3 | Discipline Resolution Issues: Algorithm is based on net types rather than driver that appears on mixed net.<br>Related Issues:<br>• Driver-Receiver segregation<br>• placement of A/D converter<br>• empty disciplines, undeclared nets<br>• how to deal with leaf level wires<br>• no clear definition on OOMR<br>• What is the discipline of an expression when an expression is specified as an actual in a port connection? A driver might have to be created as an unnamed implicit function. | Ambiguity | National Semiconductors | Future |
| 4 | LRM does not clearly illustrate the MS simulation cycle and the initialization is not clearly defined. Illustration of IC analysis in AMS is non-existent.<br>• Which solver starts first<br>• Initialization mechanism<br>• Mixed-signal initialization: Verilog-D simulators are transient in operation and hence there is no mechanism defined for static/steady state simulation. | Ambiguity | -- | LRM2.2 |
| 5 | System tasks and function. Issue with **$random**. The syntax is defined in both digital (1364-2001) as well as analog (LRM 2.1) BNF. The syntax should be in sync. | BNF | Cadence | LRM2.2 |
| 6 | Issue with genvar. Currently genvar is defined both in Verilog-D 1364-2001 std as well as LRM 2.1, and they are used in different contexts. Digital genvar is used more in structural context and analog genvar is used in a behavioural (*analog_for*) context.<br>Should the syntax be deprecated in if/else blocks and switch statements and have analog genvar usage just in "analog_for" blocks to make migration and sync'ing up with digital standards later easier. | BNF | Motorola | LRM2.2 |

| Item | Description/Issue | Category | Owner | Version |
|---|---|---|---|---|
| 7 | • External module definition to support direct import of SPICE netlists in Annex E. Currently this is not clearly stated in the LRM.<br>• Augment "external module" and "macromodule" definitions within a "simulator class". Extend the "external module" syntax with parameters indicating the language used. Also, case sensitivity of SPICE simulators should be accounted for in the language. Should there be filters for support of foreign language? Implement something similar to "shell" which is used in digital simulators. | Enhancement | National Semiconductors | Future |
| 8 | Support for global variables using dynamic parameters in VerilogAMS. These parameters would be alterable between analysis runs. | Enhancement | Cadence | LRM2.2? |
| 9 | Ambiguities with *if-else-if* BNF syntax, when nested *if* statements are used, when the *if-condition* is statically evaluatable. | BNF | Motorola | LRM2.2 |
| 10 | Contribution statements:<br>• It is not clear whether contribution statements should be allowed as part of initial conditions like **initial_step**. The current BNF restricts the use of contribution statements in this context.<br>• Contribution statement inside loops. Should this be allowed? Current LRM disallows using contribution statements in loops. | Enhancement | Motorola | LRM2.2 |
| 11 | Analysis-dependent function should be clearly defined with use of tables to denote how they behave.<br>This should also clarify DC Sweep mechanism, which is not detailed in the LRM. Behaviour of initial/final step on DC Sweep is also not defined. | Ambiguity | Motorola | LRM2.2 |
| 12 | Cleanup on the analog syntax within the BNF to make it complete and consistent.<br>Syntax consistencies with 1364 in BNF snippets specified while describing the feature<br>Syntax 6-3, 6-4, 6-5 in Section 6.3, 6.4, and 6.5 are inconsistent with the BNF syntax defined in Annexure.The syntax snippets in the actual sections should be the same as the one specified in Annex.<br>Syntax 3-5 with regards to the discipline suffers from the same problem (not reflecting the Annex). All syntax snipets in all chapters to be made consistent. | BNF | Motorola | LRM2.2 |
| 13 | SPICE versus Verilog name conflict. There is an issue while instantiating two modules with the same name defined in different abstractions (SPICE versus Verilog). Currently, the LRM does not have any name scoping mechanisms. Look at the possibility of issuing an error when a name conflict occurs, or use the standard library methodology to pick the first match from the library. | Enhancement | National Semiconductors | Future |
| 14 | Switch branch syntax not defined in BNF, though it is explained in an example. Also, should indirect branch assignment be made illegal in conditional? This is not clearly defined in the LRM (direct contribution statements are allowed in *if* statements to model switch behavior). | BNF | Motorola | LRM2.2 |

| Item | Description/Issue | Category | Owner | Version |
|---|---|---|---|---|
| 15 | Discipline Compatibility:<br>How do you resolve disciplines with different abstol. Should there be a resolution function? Should the tighter tolerance apply? | Ambiguity | National Semiconductors | LRM2.2 |
| 16 | Discipline Compatibility:<br>Current LRM makes it illegal to have incompatible continuous disciplines on the same net. Should this be allowed? | Enhancement | National Semiconductors | Future |
| 17 | The meaning of 'analysis point' in the table explaining **initial_step**/**final_step** for all analysis is not clear (Section 6.7.4). | Ambiguity | Cadence | LRM2.2 |
| 18 | 'include does not support both <> and "" to include header files. This syntax could be used to differentiate system-defined header files and user-defined header files. | Enhancement | National Semiconductor | Future |
| 19 | The specification of roots for the Zi filter has changed from Z^-1 to Z. Poles and zeros are roots of Z^-1, but this changed in post 1.4 versions. | Ambiguity | Cadence | LRM2.2 |
| 20 | Should flow and potential etc be part of the global keyword list? Should there be a single global keyword list? | Enhancement | Cadence | Future |
| 21 | Coercion of string to real does not make sense and is not defined, but it is currently allowed. The LRM should not allow coercing of a string to real. This should be treated as an error in the analog context. | BNF | Cadence | LRM2.2 |
| 22 | The concept of associating variables based on which context they are assigned is apparently felt to be ambiguous. Can there be a better approach to identifying the domain of a variable in a mixed-signal context? | Enhancement | -- | Future |
| 23 | Can "„" (comma-comma) syntax be used for null arguments instead of "{}"? This can be useful for defining NULL numerator/denominator arguments in Laplace/Zi. Can this be explicitly allowed for these operators? | Enhancement | Motorola | LRM2.2 |
| 24 | How will we handle parameter override values specified in **defparam** versus instantiation precedence? If the module is pre-compiled, the compiler would not know about the last override value. (Apparently there was a discussion on Verilog-D not to support **defparam**.) | Enhancement | -- | Future |
| 25 | Light Weight Conversions: connectmodule only addresses automatic conversion of signals passed through ports (structural). What about OOMRs where behavioral code asks for values not passed through ports? These are just probes, which do not need resolution. There should be a mechanism in the LRM for treating these light weight conversions. | Enhancement | -- | Future |
| 26 | VPI issue: Nature and disciplines should not use param_assign, instead there should be a new object called attr_assign | Ambiguity | Cadence | Future |
| 27 | Support break/continue statement in behavioral syntax to break out of loops. Is this supported in 136-2001 std? How about SV3.1 standard? | Enhancement | Motorola | Future |

| Item | Description/Issue | Category | Owner | Version |
|------|-------------------|----------|-------|---------|
| 28 | Indexing of Named vector branches: What should be the index of a vector branch declared between two vector nodes:<br><br>`electrical [1:2] a;`<br>`electrical [0:1] b;`<br>`branch (a,b) br1 [1:2];`<br>`branch (a,b) br2; //??`<br><br>Should `br2` default to `[0:1]`? | BNF | Motorola | LRM2.2 |
| 29 | Bitwise **nand**/**nor** is mentioned in the operators precedence table, but there is no mention of these operators in the list of bitwise operators | BNF | Motorola | LRM2.2 |
| 30 | Currently, the digital syntax does not allow register declaration in named blocks mentioned in an @(initial) or @(always) block. Register and net declarations should be allowed as part of these named blocks | Enhancement | Cadence | LRM2.2 |
| 31 | Vector return values for UDF: LRM do not explicitly state that the return value of UDF should be scalar. This should be clarified. | Ambiguity | Motorola | LRM2.2 |
| 32 | Events inside loop statements: Events are used inside loops in references through examples (Section 5.2, dac), though this is disallowed as per the grammar. | Ambiguity | Motorola | LRM2.2 |
| 33 | Accessing nature and discipline attributes to be supported in the language. The language supports definition of these attributes but there is no syntax to access them. | Enhancement | Cadence | LRM2.2 |
| 34 | Support for system task $table_model in VerilogAMS | Enhancement | Cadence | Future |
| 35 | Support of wreal IEs and hanlding of user defined system types in SystemVerilog (SV allows user defined types through ports but there is no mechanism for resolving them). Also need to work out a mechanism on how the resolution is going to be done on these nets and the resolution of IEs. | Enhancement | -- | Future ?? |
| 36 | Reduction operators in Analog blocks | Enhancement | Cadence | LRM2.2 |
| 37 | Support in Verilog-AMS BNF for strobing both digital and analog nets | BNF | -- | LRM2.2 |
| 38 | Description of how $(f)strobe and $(f)display work within the analog context | Ambiguity | Cadence | LRM2.2 |
| 39 | LRM does not have support for $fwrite tho' there are tasks to open and close files | Ambiguity | Motorola | LRM2.2 |
| 39 | Support for a $fread function (reading from files) - sync up with 2001 standard | BNF | Cadence | LRM2.2 |
| 40 | Support for operator real2int() | Enhancement | -- | Future |
| 41 | Initialization of variables | Enhanement | -- | Future |

| Item | Description/Issue | Category | Owner | Version |
|---|---|---|---|---|
| 42 | Support for power operator ** - why not use existing pow syntax?? | Enhancement | -- | Future |
| 43 | timedelay argument for absdelay not properly specified in semantics. When maxdelay is specified time_delay can be dynamic during simulation (ie expression syntax). However if max_delay is not specified the time_delay argument should be a constant expression (cannot be changed during simulation). These syntax/semantics are not clearly specified in the LRM | Ambiguity | Motorola | LRM2.2 |
| 44 | Usage of implicit nets - LRM 1.4 stated that usage of implicit nets is forbidded in the behavioural part of the language (ie nets used in behaviour should be declared), implying they can be used only in the structural part as part of child instantiations. There restriction is no longer there. Does the restriction still apply? | Ambiguity | Motorola | LRM2.2 |
| 45 | Should "inf" be defined as a keyword that can be used as an expression value? Currently 'inf' and '-inf' are supported as keywords in parameter from/exclude range syntax specification. | Enhancement | Motorola | LRM2.2 |
| 46 | LRM2.1 states that last_crossing should return a negative value before the expression crosses zero for the first time. What should this negative value be? Is returning negative value correct or should last_crossing return zero before the expression crosses zero for the first time? | Ambiguity | Motorola | LRM2.2 |
| 47 | Should the section 10.3 be renamed as $rdist_functions instead of $dist_function | cleanup | Motorola | LRM2.2 |
| 48 | LRM2.1 does not have an index at the back, which is very useful for references. | cleanup | -- | LRM2.2 |
| 49 | Section 3.4.5 of LRM2.1 specifies that implicit nets are defined to be scalar with empty (domainless) discipline bound to it. This discipline is not defined. Should this be wire or nuetral or an interconnect? | Ambiguity | Cadence | LRM2.2 |
| 50 | The way 'module_items' are defined in Appendix A.1 what one may declare in the module definition. It restricts it to be either a collection of 'module_items' "**or**" an analog block. it disallows having a single analog block with a list of usual digital 'module_items'. The more correct syntax would be<br>module_items := {module_item} [analog_block] {module_item} instead of<br>module_items := {module_item}<br>               | analog_block | BNF | Motorola | LRM2.2 |