Proposed BNF Fix for Verilog-2001 Parameter Errata

We (I) made a mistake in the Verilog-2001 BNF concerning the definition of the module_parameter_port_list and vendors are starting to implement these parameter port lists incorrectly. We need to fix this in the Verilog-2001 Standard (issue an errata notice), but we also need to make sure that the errata does not propagate to the IEEE Verilog-Synthesis (1364.1) and SystemVerilog (Accellera) standards to compound the problem.

The problem is that section A.1.3 shows that modules can be declared with a module_parameter_port_list, section A.1.4 shows the port list to be a comma-separated list of parameter_declarations (so far so good), section A.2.1.1 shows that each parameter_declaration ends with a semicolon (this is the problem).

Now vendors are requiring semicolons in the module_parameter_port_list, which was never our intent. We expected parameter port lists to be comma-separated (see the generic_fifo example in section 12.2 of the Verilog-2001 Standard). Unfortunately, we may have only included one example in the entire Verilog-2001 Standard showing the correct usage (not a good move on our part).

The incorrect BNF (**WAS**) and the proposed BNF correction (**PROPOSED**) sections are shown below.

Please comment ASAP.

PROPOSED: immediately replace the **WAS** section with the **PROPOSED** section in the IEEE Verilog Synthesis and Accellera SystemVerilog BNF sections and that the corrections be listed as an errata-fix by the Verilog Standards Group at it earliest convenience.

I believe these changes safely remove semicolons from a module_parameter_port_list while still requiring semicolons when parameters are declared outside of a module_parameter_port_list.

Regards - Cliff Cummings

WAS:

A.1.3 Module and primitive source text

module_declaration ::=
{ attribute_instance } module_keyword module_identifier [module_parameter_port_list]
[list_of_ports] **;** { module_item }
endmodule
| { attribute_instance } module_keyword module_identifier [module_parameter_port_list]
[list_of_port_declarations] **;** { non_port_module_item }
endmodule

A.1.4 Module parameters and ports

module_parameter_port_list ::= **#** **(** parameter_declaration { **,** parameter_declaration } **)**

A.1.5 Module items

module_item ::=
module_or_generate_item
| port_declaration **;**
| { attribute_instance } generated_instantiation
| { attribute_instance } local_parameter_declaration
| { attribute_instance } parameter_declaration

| { attribute_instance } specify_block
| { attribute_instance } specparam_declaration

non_port_module_item ::=
{ attribute_instance } generated_instantiation
| { attribute_instance } local_parameter_declaration
| { attribute_instance } module_or_generate_item
| { attribute_instance } parameter_declaration
| { attribute_instance } specify_block
| { attribute_instance } specparam_declaration

A.2.1.1 Module parameter declarations

parameter_declaration ::=
parameter [**signed**] [range] list_of_param_assignments **;**
| **parameter integer** list_of_param_assignments **;**
| **parameter real** list_of_param_assignments **;**
| **parameter realtime** list_of_param_assignments **;**
| **parameter time** list_of_param_assignments **;**

A.2.3 Declaration lists

list_of_param_assignments ::= param_assignment { **,** param_assignment }

A.2.8 Block item declarations

block_item_declaration ::=
{ attribute_instance } block_reg_declaration
| { attribute_instance } event_declaration
| { attribute_instance } integer_declaration
| { attribute_instance } local_parameter_declaration
| { attribute_instance } parameter_declaration
| { attribute_instance } real_declaration
| { attribute_instance } realtime_declaration
| { attribute_instance } time_declaration

PROPOSED:

A.1.3 Module and primitive source text (no change)

module_declaration ::=
{ attribute_instance } module_keyword module_identifier [module_parameter_port_list]
[list_of_ports] **;** { module_item }
endmodule
| { attribute_instance } module_keyword module_identifier [module_parameter_port_list]
[list_of_port_declarations] **;** { non_port_module_item }
endmodule

A.1.4 Module parameters and ports (no change)

module_parameter_port_list ::= **#** **(** parameter_declaration { **,** parameter_declaration } **)**

A.1.5 Module items (added semicolons)

module_item ::=
    module_or_generate_item
  | port_declaration **;**
  | { attribute_instance } generated_instantiation
  | { attribute_instance } local_parameter_declaration
  | { attribute_instance } parameter_declaration **;**
  | { attribute_instance } specify_block
  | { attribute_instance } specparam_declaration

non_port_module_item ::=
    { attribute_instance } generated_instantiation
  | { attribute_instance } local_parameter_declaration
  | { attribute_instance } module_or_generate_item
  | { attribute_instance } parameter_declaration **;**
  | { attribute_instance } specify_block
  | { attribute_instance } specparam_declaration

A.2.1.1 Module parameter declarations (deleted semicolons)

parameter_declaration ::=
    parameter [**signed**] [range] list_of_param_assignments
  | **parameter integer** list_of_param_assignments
  | **parameter real** list_of_param_assignments
  | **parameter realtime** list_of_param_assignments
  | **parameter time** list_of_param_assignments

A.2.3 Declaration lists (no change)

list_of_param_assignments ::= param_assignment { **,** param_assignment }

A.2.8 Block item declarations (added semicolon)

block_item_declaration ::=
    { attribute_instance } block_reg_declaration
  | { attribute_instance } event_declaration
  | { attribute_instance } integer_declaration
  | { attribute_instance } local_parameter_declaration
  | { attribute_instance } parameter_declaration **;**
  | { attribute_instance } real_declaration
  | { attribute_instance } realtime_declaration
  | { attribute_instance } time_declaration