

IEEE Std 1364™-2001

(Revision of
IEEE Std 1364-1995)

Errata to IEEE Standard Verilog® Hardware Description Language

Sponsor

Design Automation Standards Committee

of the

IEEE Computer Society

Correction Sheet

Issued 26 November 2003

Copyright © 2003 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 2003. Printed in the United States of America.

This correction sheet may be freely reproduced and distributed in order to maintain the utility and currency of the underlying Standard. This correction sheet may not be sold, licensed or otherwise distributed for any commercial purposes whatsoever. The content of this correction sheet may not be modified.

About IEEE Std 1364-2001 Version C and the Errata

During the past two years, the IEEE 1364 Working Group's Errata Task Force has thoroughly reviewed the Standard and has identified and corrected a number of production and editorial errors that crept in between balloting and printing of the Standard. IEEE Std 1364-2001 Version C incorporates all of these corrections.

In addition, during the IEEE 1364 Working Group's review of the Standard, the Working Group and its Errata Task Force identified other areas where the standard had logical inconsistencies which were not the result of production problems. The Working Group developed an Errata document that identifies these and specifies the Working Group's statement as to the correct interpretation of the Standard.

Participants

At the time IEEE Std 1364-2001 Version C and the Errata were completed, the IEEE 1364 Working Group had the following membership:

Michael T. Y. (Mac) McNamara, *Chair*

Shalom Bresticker, *Editor*

Stefen Boyd, *Web Master*

Kurt Baty
Dennis Brophy
Clifford E. Cummings
Charles Dawson
Tom Fitzpatrick
Krishna Garlapati
Keith Gover
Ennis Hawk

Richard Ho
Atsushi Kasuya
Jay Lawrence
Andrew Lynch
James A. Markevitch
Dennis Marsa
Francoise Martinolle
Mehdi Mohtashemi

Anders Nordstrom
Karen Pieper
Brad Pierce
Steven Sharp
Alec Stanculescu
Stuart Sutherland
Chong Guan Tan
Gordon Vreugdenhil

The Errata Task Force had the following membership:

Karen Pieper, *Chair*

Stefen Boyd, *Vice Chair*

Kurt Baty
Shalom Bresticker
Dennis Brophy
Clifford E. Cummings
Charles Dawson
Ted Elkind
Tom Fitzpatrick
Jay Lawrence

Andrew Lynch
James A. Markevitch
Dennis Marsa
Francoise Martinolle
Michael T. Y. (Mac) McNamara
Elliot Mednick
Don Mills

Mehdi Mohtashemi
Anders Nordstrom
Brad Pierce
David Roberts
Steven Sharp
David Smith
Stuart Sutherland
Gordon Vreugdenhil

The Behavioral Task Force had the following membership:

Steven Sharp, *Chair*

Kurt Baty
Stefen Boyd
Dennis Brophy
Clifford E. Cummings
Tom Fitzpatrick
Ennis Hawk

Atsushi Kasuya
Jay Lawrence
Francoise Martinolle
Michael T. Y. (Mac) McNamara
Don Mills

Mehdi Mohtashemi
Karen Pieper
Brad Pierce
Alec Stanculescu
Stuart Sutherland
Gordon Vreugdenhil

The PLI Task Force had the following membership:

Charles Dawson, *Co-Chair*

Stuart Sutherland, *Co-Chair*

Steven Dovich
Dennis Marsa

Francoise Martinolle

Nisa Parikh
David Roberts

Clause 3

Page 25, subclause 3.5, paragraph 1:

The first dashed-list item

“—If an identifier is used in a port expression declaration, then an implicit net of type **wire** shall be assumed, with the vector width of the port expression declaration. See 12.3.3 for a discussion of port expression declarations.”

should be corrected to read

“—If an identifier is used in a port expression declaration, then an implicit net of default net type shall be assumed, with the vector width of the port expression declaration. See 12.3.3 for a discussion of port expression declarations.”

Clause 4

Page 42, subclause 4.1.2, Table 12:

Change row 1, column 1 from

“+ - ! ~ (unary)”

to

“+ - ! ~ & ~& | ~| ^ ~^ ^~ (unary)”

Change row 8, column 1, from

“& ~&”

to

“& (binary)”

Change row 9, column 1, from

“^ ~ ^~”

to

“^ ~ ^~ (binary)”

Change row 10, column 1, from

“| ~|”

to


“| (binary)”

In row 13, column 2, delete “Lowest precedence.”

Add a new bottom row (row 14)—add “{ } { } { }” in column 1, “Lowest precedence” in column 2.

Table 12 should appear as follows:

Table 12—Precedence rules for operators

+ - ! ~ & ~& ~ ^ ~ ^ ~ (unary)	Highest precedence
**	
* / %	
+ - (binary)	
<< >> <<< >>>	
< <= > >=	
== != === !==	
& (binary)	
^ ^~ ~^ (binary)	
(binary)	
&&	
?: (conditional operator)	
{ } { }	Lowest precedence

Page 60, subclause 4.4.1, Table 29, row 5, column 1:

The operators “&& ||” should be deleted, and a new row should be inserted for the operators stating that their operands are “self-determined” in column 3 (“Comments”). Table 29 should appear as follows:

Table 29—Bit lengths resulting from self-determined expressions

Expression	Bit length	Comments
Unsize constant number*	Same as integer	
Sized constant number	As given	
i op j, where op is: + - * / % & ^ ^~ ~^	max(L(i),L(j))	
op i, where op is: + - ~	L(i)	
i op j, where op is: === !== == != > >= < <=	1 bit	Operands are sized to max(L(i),L(j))
i op j, where op is: &&	1 bit	All operands are self-determined
op i, where op is: & ~& ~ ^ ~ ^ ~ !	1 bit	All operands are self-determined
i op j, where op is: >> << ** >>> <<<	L(i)	j is self-determined
i ? j : k	max(L(j),L(k))	i is self-determined

Table 29—Bit lengths resulting from self-determined expressions (Continued)

Expression	Bit length	Comments
{i,...j}	$L(i)+..+L(j)$	All operands are self-determined
{i{j,...k}}	$i * (L(j)+..+L(k))$	All operands are self-determined

*If an unsized constant is part of an expression that is longer than 32 bits, then if the most significant bit is unknown (X or x) or three-state (Z or z) the most significant bit is extended up to the size of the expression, otherwise signed constants are sign extended and unsigned constants are zero extended.

NOTE Multiplication without losing any overflow bits is still possible simply by assigning the result to something wide enough to hold it.

Clause 8

Page 110, subclause 8.1.6, Table 40, row 3, column 3:

The “Comment”

“Permitted in the input fields of all UDPs and in the current state field of sequential UDPs.”

should be corrected to read

“Permitted in the input and output fields of all UDPs and in the current state field of sequential UDPs.”

Page 115, subclause 8.6, Syntax 8-2, line 3:

The “[attribute_instance]” should be deleted. Syntax 8-2 should appear as follows:

```

udp_instantiation ::= (From Annex A- A.5.4)
    udp_identifier [ drive_strength ] [ delay2 ]
    udp_instance { , udp_instance } ;
udp_instance ::=
    [ name_of_udp_instance ] ( output_terminal , input_terminal
    { , input_terminal } )
name_of_udp_instance ::=
    udp_instance_identifier [ range ]
    
```

Clause 9

Page 120, subclause 9.2.1, Syntax 9-1, line 11:

The line “@ event_identifier” should be “@ hierarchical_event_identifier.” Syntax 9-1 should appear as follows:

```
blocking_assignment ::= (From Annex A - A.6.2)
    variable_lvalue = [ delay_or_event_control ] expression
delay_control ::= (From Annex A - A.6.5)
    # delay_value
    | # ( mintypmax_expression )
delay_or_event_control ::=
    delay_control
    | event_control
    | repeat ( expression ) event_control
event_control ::=
    @ hierarchical_event_identifier
    | @ ( event_expression )
    | @*
    | @ (*)
event_expression ::=
    expression
    | hierarchical_identifier
    | posedge expression
    | negedge expression
    | event_expression or event_expression
    | event_expression , event_expression
variable_lvalue ::= (From Annex A - A.8.5)
    hierarchical_variable_identifier
    | hierarchical_variable_identifier [ expression ] { [ expression ] }
    | hierarchical_variable_identifier [ expression ] { [ expression ] }
      [ range_expression ]
    | hierarchical_variable_identifier [ range_expression ]
    | variable_concatenation
```

Page 121, subclause 9.2.2, Syntax 9-2, line 11:

The line “@ event_identifier” should be “@ hierarchical_event_identifier.” Syntax 9-2 should appear as follows:

```
nonblocking_assignment ::= (From Annex A - A.6.2)
    variable_lvalue <= [ delay_or_event_control ] expression
delay_control ::= (From Annex A - A.6.5)
    # delay_value
    | # ( mintypmax_expression )
delay_or_event_control ::=
    delay_control
    | event_control
    | repeat ( expression ) event_control
event_control ::=
    @ hierarchical_event_identifier
    | @ ( event_expression )
    | @*
    | @ (*)
event_expression ::=
    expression
    | hierarchical_identifier
    | posedge expression
    | negedge expression
    | event_expression or event_expression
    | event_expression , event_expression
variable_lvalue ::= (From Annex A - A.8.5)
    hierarchical_variable_identifier
    | hierarchical_variable_identifier [ expression ] { [ expression ] }
    | hierarchical_variable_identifier [ expression ] { [ expression ] }
    | [ range_expression ]
    | hierarchical_variable_identifier [ range_expression ]
    | variable_concatenation
```

Page 137, subclause 9.7, Syntax 9-8, line 9:

The line “@ event_identifier” should be “@ hierarchical_event_identifier.” Syntax 9-8 should appear as follows:

```
delay_control ::= (From Annex A - A.6.5)
    # delay_value
    | # ( mintypmax_expression )
delay_or_event_control ::=
    delay_control
    | event_control
    | repeat ( expression ) event_control
event_control ::=
    @ hierarchical_event_identifier
    | @ ( event_expression )
    | @*
    | @ (*)
event_expression ::=
    expression
    | hierarchical_identifier
    | posedge expression
    | negedge expression
    | event_expression or event_expression
    | event_expression , event_expression
```

Page 142, subclause 9.7.7, Syntax 9-12, line 13:

The line “@ event_identifier” should be “@ hierarchical_event_identifier.” Syntax 9-12 should appear as follows:

```
blocking_assignment ::= (From Annex A - A.6.2)
    variable_lvalue = [ delay_or_event_control ] expression
nonblocking_assignment ::=
    variable_lvalue <= [ delay_or_event_control ] expression
delay_control ::= (From Annex A - A.6.5)
    # delay_value
    | # ( mintypmax_expression )
delay_or_event_control ::=
    delay_control
    | event_control
    | repeat ( expression ) event_control
event_control ::=
    @ hierarchical_event_identifier
    | @ ( event_expression )
    | @*
    | @ (*)
event_expression ::=
    expression
    | hierarchical_identifier
    | posedge expression
    | negedge expression
    | event_expression or event_expression
    | event_expression , event_expression
```


Clause 10

Page 153, subclause 10.2.1, Syntax 10-1, line 29:

The left bracket of the two surrounding “task_port_type” in the “tf_inout_declaration” should not be bold.

Syntax 10-1 should appear as follows:

```
task_declaration ::= (From Annex A - A.2.7)
    task [ automatic ] task_identifier ;
    { task_item_declaration }
    statement
endtask
| task [ automatic ] task_identifier ( task_port_list ) ;
    { block_item_declaration }
    statement
endtask
task_item_declaration ::=
    block_item_declaration
    | { attribute_instance } tf_input_declaration ;
    | { attribute_instance } tf_output_declaration ;
    | { attribute_instance } tf_inout_declaration ;
task_port_list ::=
    task_port_item { , task_port_item }
task_port_item ::=
    { attribute_instance } tf_input_declaration
    | { attribute_instance } tf_output_declaration
    | { attribute_instance } tf_inout_declaration
tf_input_declaration ::=
    input [ reg ] [ signed ] [ range ] list_of_port_identifiers
    | input [ task_port_type ] list_of_port_identifiers
tf_output_declaration ::=
    output [ reg ] [ signed ] [ range ] list_of_port_identifiers
    | output [ task_port_type ] list_of_port_identifiers
tf_inout_declaration ::=
    inout [ reg ] [ signed ] [ range ] list_of_port_identifiers
    | inout [ task_port_type ] list_of_port_identifiers
task_port_type ::=
    time | real | realtime | integer
block_item_declaration ::= (From Annex A - A.2.8)
    { 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
block_reg_declaration ::=
    reg [ signed ] [ range ]
    list_of_block_variable_identifiers ;
list_of_block_variable_identifiers ::=
    block_variable_type { , block_variable_type }
block_variable_type ::=
    variable_identifier
    | variable_identifier dimension { dimension }
```

Page 158, subclause 10.3.1, Syntax 10-3:

In line 9, “block_item_declaration { block_item_declaration }” should be “{ block_item_declaration }.”

In line 14, “| tf_input_declaration ;” should be “| {attribute_instance} tf_input_declaration ;”.

Syntax 10-3 should appear as follows:

```
function_declaration ::= (From Annex A - A.2.6)
    function [ automatic ] [ signed ] [ range_or_type ]
        function_identifier ;
        function_item_declaration { function_item_declaration }
        function_statement
    endfunction
| function [ automatic ] [ signed ] [ range_or_type ]
    function_identifier ( function_port_list ) ;
    { block_item_declaration }
    function_statement
    endfunction
function_item_declaration ::=
    block_item_declaration
    | {attribute_instance} tf_input_declaration ;
function_port_list ::=
    { attribute_instance } tf_input_declaration
    { , { attribute_instance } tf_input_declaration }
tf_input_declaration ::=
    input [ reg ] [ signed ] [ range ] list_of_port_identifiers
    | input [ task_port_type ] list_of_port_identifiers
range_or_type ::=
    range | integer | real | realtime | time
block_item_declaration ::= (From Annex A - A.2.8)
    { 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
block_reg_declaration ::=
    reg [ signed ] [ range ]
        list_of_block_variable_identifiers ;
list_of_block_variable_identifiers ::=
    block_variable_type { , block_variable_type }
block_variable_type ::=
    variable_identifier
    | variable_identifier dimension { dimension }
```

Page 161, subclause 10.3.5, paragraph 1, dashed list:

The fifth dashed-list item

“—The only system task that may be invoked is \$display, and it shall be ignored when invoked at elaboration time.”

should be deleted.

Clause 12

Page 167, subclause 12.1, Syntax 12-1, line 16:

The “range_expression” should be “constant_range_expression.” Syntax 12-1 should appear as follows:

```
module_declaration ::= (From Annex A - A.1.3)
    { 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

module_keyword ::= module | macromodule

module_parameter_port_list ::= (From Annex A -A.1.4
    # ( parameter_declaration { , parameter_declaration } ) )

list_of_ports ::= ( port { , port } )
list_of_port_declarations ::= ( port_declaration { , port_declaration } ) | ( )
port ::= [ port_expression ] . port_identifier ( [ port_expression ] )
port_expression ::= port_reference | { port_reference { , port_reference } }
port_reference ::= port_identifier | port_identifier [ constant_expression ]
| port_identifier [ constant_range_expression ]
port_declaration ::= { attribute_instance } inout_declaration
| { attribute_instance } input_declaration
| { attribute_instance } output_declaration

module_item ::= module_or_generate_item (From Annex A - A.1.5)
    port_declaration ;
    { attribute_instance } generated_instantiation
    { attribute_instance } local_parameter_declaration
    { attribute_instance } parameter_declaration
    { attribute_instance } specify_block
    { attribute_instance } specparam_declaration

module_or_generate_item ::= { attribute_instance } module_or_generate_item_declaration
    { attribute_instance } parameter_override
    { attribute_instance } continuous_assign
    { attribute_instance } gate_instantiation
    { attribute_instance } udp_instantiation
    { attribute_instance } module_instantiation
    { attribute_instance } initial_construct
    { attribute_instance } always_construct

module_or_generate_item_declaration ::= net_declaration
    reg_declaration
    integer_declaration
    real_declaration
    time_declaration
    realtime_declaration
    event_declaration
    genvar_declaration
    task_declaration
    function_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

parameter_override ::= defparam list_of_param_assignments ;
```

Page 185, subclause 12.3.1, Syntax 12-5, line 15:

The “range_expression” should be “constant_range_expression.” Syntax 12-5 should appear as follows:

```
list_of_ports ::= (From Annex A - A.1.4)
    ( port { , port } )
list_of_port_declarations ::=
    ( port_declaration { , port_declaration } )
    | ( )
port ::=
    [ port_expression ]
    | . port_identifier ( [ port_expression ] )
port_expression ::=
    port_reference
    | { port_reference { , port_reference } }
port_reference ::=
    port_identifier
    | port_identifier [ constant_expression ]
    | port_identifier [ constant_range_expression ]
port_declaration ::=
    { attribute_instance } inout_declaration
    | { attribute_instance } input_declaration
    | { attribute_instance } output_declaration
```

Clause 13

Page 202, subclause 13.2.1, Syntax 13-2:

The last line “**include** <file_path_spec> ;” should be “**include** file_path_spec ;”.

In line 9, add a closing bracket “]” between “]” and “;”.

Syntax 13-2 should appear as follows:

```
library_text ::= (From Annex A - A.1.1)
    { library_descriptions }
library_descriptions ::=
    library_declaration
    | include_statement
    | config_declaration
library_declaration ::=
    library library_identifier file_path_spec [ { , file_path_spec } ]
    [ -incdir file_path_spec [ { , file_path_spec } ] ] ;
file_path_spec ::=
    file_path
include_statement ::=
    include file_path_spec ;
```

Page 203, subclause 13.2.2, Syntax 13-3, last line:

The line “**include** <file_path_spec> ;” should be “**include** file_path_spec ;” Syntax 13-3 should appear as follows:

```
include_statement ::= (From Annex A - A.1.1)
    include file_path_spec ;
```

Clause 14

Page 214, subclause 14.2.2, Syntax 14-3, lines 16 and 20:

The “range_expression” should be “constant_range_expression.” Syntax 14-3 should appear as follows:

```
simple_path_declaration ::= (From Annex A - A.7.2)
    parallel_path_description = path_delay_value
    | full_path_description = path_delay_value
parallel_path_description ::=
    ( specify_input_terminal_descriptor [ polarity_operator ] =>
      specify_output_terminal_descriptor )
full_path_description ::=
    ( list_of_path_inputs [ polarity_operator ] *> list_of_path_outputs )
list_of_path_inputs ::=
    specify_input_terminal_descriptor { , specify_input_terminal_descriptor }
list_of_path_outputs ::=
    specify_output_terminal_descriptor { , specify_output_terminal_descriptor }
specify_input_terminal_descriptor ::= (From Annex A - A.7.3)
    input_identifier
    | input_identifier [ constant_expression ]
    | input_identifier [ constant_range_expression ]
specify_output_terminal_descriptor ::=
    output_identifier
    | output_identifier [ constant_expression ]
    | output_identifier [ constant_range_expression ]
input_identifier ::=
    input_port_identifier | inout_port_identifier
output_identifier ::=
    output_port_identifier | inout_port_identifier
polarity_operator ::= (From Annex A - A.7.4)
    + | -
```

Page 215, subclause 14.2.3, Syntax 14-4:

The lines 4–6

```
“parallel_edge_sensitive_path_description ::=
([ edge_identifier ] specify_input_terminal_descriptor =>
specify_output_terminal_descriptor [ polarity_operator ] : data_source_expression )”
```

should be corrected to read

```
“parallel_edge_sensitive_path_description ::=
([ edge_identifier ] specify_input_terminal_descriptor =>
( specify_output_terminal_descriptor [ polarity_operator ] : data_source_expression ) )”
```

and the lines 7-9

```
“full_edge_sensitive_path_description ::=
( [ edge_identifier ] list_of_path_inputs *>
  list_of_path_outputs [ polarity_operator ] : data_source_expression )”
```

should be corrected to read

```
“full_edge_sensitive_path_description ::=
( [ edge_identifier ] list_of_path_inputs *>
  ( list_of_path_outputs [ polarity_operator ] : data_source_expression ) )”
```

Syntax 14-4 should appear as follows:

```
edge_sensitive_path_declaration ::= (From Annex A - A.7.4)
  parallel_edge_sensitive_path_description = path_delay_value
  | full_edge_sensitive_path_description = path_delay_value
parallel_edge_sensitive_path_description ::=
  ( [ edge_identifier ] specify_input_terminal_descriptor =>
    ( specify_output_terminal_descriptor [ polarity_operator ] : data_source_expression ) )
full_edge_sensitive_path_description ::=
  ( [ edge_identifier ] list_of_path_inputs *>
    ( list_of_path_outputs [ polarity_operator ] : data_source_expression ) )
data_source_expression ::=
  expression
edge_identifier ::=
  posedge | negedge
```

Clause 15

Page 240, subclause 15.1, Syntax 15-2, line 30:

The “edge_control_specifier ::= **edge** [edge_descriptor [, edge_descriptor]]” should be “edge_control_specifier ::= **edge** [edge_descriptor { , edge_descriptor }].”

Syntax 15-2 should appear as follows:

```
checktime_condition ::= (From Annex A - A.7.5.2)
    mintypmax_expression
controlled_reference_event ::=
    controlled_timing_check_event
data_event ::=
    timing_check_event
delayed_data ::=
    terminal_identifier
    | terminal_identifier [ constant_mintypmax_expression ]
delayed_reference ::=
    terminal_identifier
    | terminal_identifier [ constant_mintypmax_expression ]
end_edge_offset ::= mintypmax_expression
event_based_flag ::= constant_expression
notify_reg ::= variable_identifier
reference_event ::= timing_check_event
remain_active_flag ::= constant_mintypmax_expression
stamptime_condition ::= mintypmax_expression
start_edge_offset ::= mintypmax_expression
threshold ::= constant_expression
timing_check_limit ::= expression
timing_check_event ::= (From Annex A - A.7.5.3)
    [timing_check_event_control] specify_terminal_descriptor [ &&& timing_check_condition ]
controlled_timing_check_event ::=
    timing_check_event_control specify_terminal_descriptor [ &&& timing_check_condition ]
timing_check_event_control ::= posedge | negedge | edge_control_specifier
specify_terminal_descriptor ::=
    specify_input_terminal_descriptor
    | specify_output_terminal_descriptor
edge_control_specifier ::= edge [ edge_descriptor { , edge_descriptor } ]
edge_descriptor* ::= 01 | 10 | z_or_x zero_or_one | zero_or_one z_or_x
zero_or_one ::= 0 | 1
z_or_x ::= x | X | z | Z
timing_check_condition ::=
    scalar_timing_check_condition
    | ( scalar_timing_check_condition )
scalar_timing_check_condition ::=
    expression
    | ~ expression
    | expression == scalar_constant
    | expression === scalar_constant
    | expression != scalar_constant
    | expression !== scalar_constant
scalar_constant ::= 1'b0 | 1'b1 | 1'B0 | 1'B1 | b0 | b1 | B0 | B1 | 1 | 0
```

*Embedded spaces are illegal.

Page 245, paragraph 9, line 2 of subclause 15.2.3 (paragraph 5, line 2 of the page):

The line

“(beginning of time window) < (timecheck time) <= (end of time window)”

should be corrected to read

“(beginning of time window) < (timestamp time) <= (end of time window)”

Page 249, paragraph 8, line 2 of subclause 15.2.6 (paragraph 4, line 2 of the page):

The line

“(beginning of time window) < (timecheck time) <= (end of time window)”

should be corrected to read

“(beginning of time window) < (timestamp time) <= (end of time window)”

Page 259, subclause 15.4, Syntax 15-15, line 2:

The “edge_control_specifier ::= **edge** [edge_descriptor [, edge_descriptor]]” should be “edge_control_specifier ::= **edge** [edge_descriptor { , edge_descriptor }].”

Syntax 15-15 should appear as follows:

```
edge_control_specifier ::= (From Annex A - A.7.5.3)
    edge [ edge_descriptor { , edge_descriptor } ]
edge_descriptor* ::=
    01
    | 10
    | z_or_x zero_or_one
    | zero_or_one z_or_x
zero_or_one ::= 0 | 1
z_or_x ::= x | X | z | Z
```

* Embedded spaces are illegal.

Clause 17

Pages 320–321, subclause 17.9.3:

The following chi_square function code

```
“static double
chi_square(seed,deg_of_free)
long *seed,deg_of_free;
{
    double x;
    long k;
    if(deg_of_free % 2)
    {
        x = normal(seed,0,1);
        x = x * x;
    }
    else
    {
        x = 0.0;
    }
    double log(),n;

    n = uniform(seed,0,1);
    if(n != 0)
    {
        n = -log(n) * mean;
    }
    return(n);
}”
```

should be corrected to appear as follows:

```
“static double
chi_square(seed,deg_of_free)
long *seed,deg_of_free;
{
    double x;
    long k;
    if(deg_of_free % 2)
    {
        x = normal(seed,0,1);
        x = x * x;
    }
    else
    {
        x = 0.0;
    }
    for(k = 2;k <= deg_of_free;k = k + 2)
    {
        x = x + 2 * exponential(seed,1);
    }
    return(x);
}”
```

Clause 19

Page 351, subclause 19.2, Syntax 19-1, line 3:

Add “**tri1**” between “| **tri0** |” and “| **wand** |.” Syntax 19-1 should appear as follows:

```
default_nettype_compiler_directive ::=
    'default_nettype net_type
net_type ::= wire | tri | tri0 | tri1 | wand | triand | wor | trior | trireg | none
```

Annex A

Page 761, subclause A.1.1:

The last line “include_statement ::= **include** <file_path_spec> ;” should be

“include_statement ::= **include** file_path_spec ;”.

In line 8, add a closing bracket “]” between “]” and “;”.

Subclause A.1.1 should appear as follows:

```
library_text ::= { library_descriptions }
library_descriptions ::=
    library_declaration
    | include_statement
    | config_declaration
library_declaration ::=
    library library_identifier file_path_spec [ { , file_path_spec } ]
    [ -indir file_path_spec [ { , file_path_spec } ] ];
file_path_spec ::= file_path
include_statement ::= include file_path_spec ;
```

Page 762, subclause A.1.4, line 15:

The “range_expression” should be “constant_range_expression.”

Subclause A.1.4 should appear as follows:

```
module_parameter_port_list ::= # ( parameter_declaration { , parameter_declaration } )
list_of_ports ::= ( port { , port } )
list_of_port_declarations ::=
    ( port_declaration { , port_declaration } )
    | ( )
port ::=
    [ port_expression ]
    | . port_identifier ( [ port_expression ] )
port_expression ::=
    port_reference
    | { port_reference { , port_reference } }
```

```

port_reference ::=
    port_identifier
    | port_identifier [ constant_expression ]
    | port_identifier [ constant_range_expression ]
port_declaration ::=
    {attribute_instance} inout_declaration
    | {attribute_instance} input_declaration
    | {attribute_instance} output_declaration

```

Page 766, subclause A.2.6:

In line 7, “block_item_declaration { block_item_declaration }” should be “{ block_item_declaration }.”

In line 12, “! tf_input_declaration ;” should be “! {attribute_instance} tf_input_declaration ;”.

Subclause A.2.6 should appear as follows:

```

function_declaration ::=
    function [ automatic ] [ signed ] [ range_or_type ] function_identifier ;
    function_item_declaration { function_item_declaration }
    function_statement
    endfunction
    | function [ automatic ] [ signed ] [ range_or_type ] function_identifier ( function_port_list ) ;
    { block_item_declaration }
    function_statement
    endfunction
function_item_declaration ::=
    block_item_declaration
    | {attribute_instance} tf_input_declaration ;
function_port_list ::= { attribute_instance } tf_input_declaration { , { attribute_instance } tf_input_declaration }
range_or_type ::= range | integer | real | realtime | time

```

Page 767, subclause A.2.7, line 28:

The left bracket of the two surrounding “task_port_type” in the “tf_inout_declaration” should not be bold.

Subclause A.2.7 should appear as follows:

```

task_declaration ::=
    task [ automatic ] task_identifier ;
    { task_item_declaration }
    statement
    endtask
    | task [ automatic ] task_identifier ( task_port_list ) ;
    { block_item_declaration }
    statement
    endtask

```

```

task_item_declaration ::=
    block_item_declaration
    | { attribute_instance } tf_input_declaration ;
    | { attribute_instance } tf_output_declaration ;
    | { attribute_instance } tf_inout_declaration ;
task_port_list ::= task_port_item { , task_port_item }
task_port_item ::=
    { attribute_instance } tf_input_declaration
    | { attribute_instance } tf_output_declaration
    | { attribute_instance } tf_inout_declaration
tf_input_declaration ::=
    input [ reg ] [ signed ] [ range ] list_of_port_identifiers
    | input [ task_port_type ] list_of_port_identifiers
tf_output_declaration ::=
    output [ reg ] [ signed ] [ range ] list_of_port_identifiers
    | output [ task_port_type ] list_of_port_identifiers
tf_inout_declaration ::=
    inout [ reg ] [ signed ] [ range ] list_of_port_identifiers
    | inout [ task_port_type ] list_of_port_identifiers
task_port_type ::=
    time | real | realtime | integer

```

Page 773, subclause A.6.5, line 12:

The line “@ event_identifier” should be “@ hierarchical_event_identifier.”

Subclause A.6.5 should appear as follows:

```

delay_control ::=
    # delay_value
    | # ( mintypmax_expression )
delay_or_event_control ::=
    delay_control
    | event_control
    | repeat ( expression ) event_control
disable_statement ::=
    disable hierarchical_task_identifier ;
    | disable hierarchical_block_identifier ;
event_control ::=
    @ hierarchical_event_identifier
    | @ ( event_expression )
    | @*
    | @ (*)
event_trigger ::=
    -> hierarchical_event_identifier ;

```

```

event_expression ::=
    expression
    | hierarchical_identifier
    | posedge expression
    | negedge expression
    | event_expression or event_expression
    | event_expression , event_expression
procedural_timing_control_statement ::=
    delay_or_event_control_statement_or_null
wait_statement ::=
    wait ( expression ) statement_or_null

```

Page 775, subclause A.7.3, lines 4 and 8:

The “range_expression” should be “constant_range_expression.”

Subclause A.7.3 should appear as follows:

```

specify_input_terminal_descriptor ::=
    input_identifier
    | input_identifier [ constant_expression ]
    | input_identifier [ constant_range_expression ]
specify_output_terminal_descriptor ::=
    output_identifier
    | output_identifier [ constant_expression ]
    | output_identifier [ constant_range_expression ]
input_identifier ::= input_port_identifier | inout_port_identifier
output_identifier ::= output_port_identifier | inout_port_identifier

```

Page 776, subclause A.7.4:

The lines 34–36

```

“parallel_edge_sensitive_path_description ::=
( [ edge_identifier ] specify_input_terminal_descriptor =>
    specify_output_terminal_descriptor [ polarity_operator ] : data_source_expression )”

```

should be corrected to read

```

“parallel_edge_sensitive_path_description ::=
( [ edge_identifier ] specify_input_terminal_descriptor =>
    ( specify_output_terminal_descriptor [ polarity_operator ] : data_source_expression ) )”

```

and the lines 37–39

```
“full_edge_sensitive_path_description ::=
([ edge_identifier ] list_of_path_inputs *>
  list_of_path_outputs [ polarity_operator ] : data_source_expression )”
```

should be corrected to read

```
“full_edge_sensitive_path_description ::=
([ edge_identifier ] list_of_path_inputs *>
  ( list_of_path_outputs [ polarity_operator ] : data_source_expression ) )”
```

Subclause A.7.4 should appear as follows:

```
path_delay_value ::=
  list_of_path_delay_expressions
  | ( list_of_path_delay_expressions )
list_of_path_delay_expressions ::=
  t_path_delay_expression
  | trise_path_delay_expression , tfall_path_delay_expression
  | trise_path_delay_expression , tfall_path_delay_expression , tz_path_delay_expression
  | t01_path_delay_expression , t10_path_delay_expression , t0z_path_delay_expression ,
  tz1_path_delay_expression , t1z_path_delay_expression , tz0_path_delay_expression
  | t01_path_delay_expression , t10_path_delay_expression , t0z_path_delay_expression ,
  tz1_path_delay_expression , t1z_path_delay_expression , tz0_path_delay_expression ,
  t0x_path_delay_expression , tx1_path_delay_expression , t1x_path_delay_expression ,
  tx0_path_delay_expression , txz_path_delay_expression , tzx_path_delay_expression
t_path_delay_expression ::= path_delay_expression
trise_path_delay_expression ::= path_delay_expression
tfall_path_delay_expression ::= path_delay_expression
tz_path_delay_expression ::= path_delay_expression
t01_path_delay_expression ::= path_delay_expression
t10_path_delay_expression ::= path_delay_expression
t0z_path_delay_expression ::= path_delay_expression
tz1_path_delay_expression ::= path_delay_expression
t1z_path_delay_expression ::= path_delay_expression
tz0_path_delay_expression ::= path_delay_expression
t0x_path_delay_expression ::= path_delay_expression
tx1_path_delay_expression ::= path_delay_expression
t1x_path_delay_expression ::= path_delay_expression
tx0_path_delay_expression ::= path_delay_expression
txz_path_delay_expression ::= path_delay_expression
tzx_path_delay_expression ::= path_delay_expression
path_delay_expression ::= constant_mintypmax_expression
edge_sensitive_path_declaration ::=
  parallel_edge_sensitive_path_description = path_delay_value
  | full_edge_sensitive_path_description = path_delay_value
parallel_edge_sensitive_path_description ::=
  ( [ edge_identifier ] specify_input_terminal_descriptor =>
    ( specify_output_terminal_descriptor [ polarity_operator ] : data_source_expression ) )
```

```

full_edge_sensitive_path_description ::=
    ( [ edge_identifier ] list_of_path_inputs *>
      ( list_of_path_outputs [ polarity_operator ] : data_source_expression ) )
data_source_expression ::= expression
edge_identifier ::= posedge | negedge
state_dependent_path_declaration ::=
    if ( module_path_expression ) simple_path_declaration
    | if ( module_path_expression ) edge_sensitive_path_declaration
    | ifnone simple_path_declaration
polarity_operator ::= + | -

```

Page 778, subclause A.7.5.3, line 12:

The “edge_control_specifier ::= **edge** [edge_descriptor [, edge_descriptor]]” should be “edge_control_specifier ::= **edge** [edge_descriptor { , edge_descriptor }].”

Subclause A.7.5.3 should appear as follows:

```

timing_check_event ::=
    [timing_check_event_control] specify_terminal_descriptor [ &&& timing_check_condition ]
controlled_timing_check_event ::=
    timing_check_event_control specify_terminal_descriptor [ &&& timing_check_condition ]
timing_check_event_control ::=
    posedge
    | negedge
    | edge_control_specifier
specify_terminal_descriptor ::=
    specify_input_terminal_descriptor
    | specify_output_terminal_descriptor
edge_control_specifier ::= edge [ edge_descriptor { , edge_descriptor } ]
edge_descriptor1 ::=
    01
    | 10
    | z_or_x zero_or_one
    | zero_or_one z_or_x
zero_or_one ::= 0 | 1
z_or_x ::= x | X | z | Z
timing_check_condition ::=
    scalar_timing_check_condition
    | ( scalar_timing_check_condition )
scalar_timing_check_condition ::=
    expression
    | ~ expression
    | expression == scalar_constant
    | expression === scalar_constant
    | expression != scalar_constant
    | expression !== scalar_constant
scalar_constant ::=
    1'b0 | 1'b1 | 1'B0 | 1'B1 | 'b0 | 'b1 | 'B0 | 'B1 | 1 | 0

```