

# Annex A

## Syntax

This annex contains the formal syntax definition of Verilog-AMS HDL. The conventions used are described in Section 1. For completeness the *IEEE 1364-1995 Verilog HDL* grammar has been included with the Verilog-AMS grammar.

### A.1 Source text

### A.2 Natures

### A.3 Disciplines

### A.4 Declarations

### A.5 Module instantiation

### A.6 Mixed-signal

## A.7 Behavioral statements

## A.8 Analog expressions

## A.9 Expressions

### A.9.1 Concatenations

```

constant_concatenation ::=
  { constant_expression { , constant_expression } }
constant_multiple_concatenation ::=
  { constant_expression constant_concatenation }
ams_constant_concatenation ::=
  { ams_constant_expression { , ams_constant_expression } }
constant_array_expression ::=
  { constant_array_init_element { , constant_array_init_element } }
constant_array_init_element ::=
  ams_constant_expression
  | integer_ams_constant_expression {ams_constant_expression }

```

### A.9.2 Expressions

```

ams_constant_expression ::=
  ams_constant_primary
  | ams_unary_operator ams_constant_primary
  | ams_constant_expression binary_operator ams_constant_expression
  | ams_constant_expression ? ams_constant_expression : ams_constant_expression
  | string
  | constant_array_expression
  | attribute_reference
  | constant_built_in_function_call
constant_expression ::=
  constant_primary
  | unary_operator constant_primary
  | constant_expression binary_operator constant_expression
  | constant_expression ? constant_expression : constant_expression
  | string

```

### A.9.3 Function calls

```

attribute_reference ::=
    net_or_port_identifier . pot_or_flow . attribute_identifier
constant_built_in_function_call ::=
    built_in_function ( ams_constant_expression { , ams_constant_expression } )
built_in_function ::=
    ln | log | exp | sqrt | min | max | abs | pow | ceil | floor
    | sin | cos | tan | asin | acos | atan | atan2
    | sinh | cosh | tanh | asinh | acosh | atanh | hypot

```

### A.9.4 Primaries

```

ams_constant_primary ::=
    number
    | parameter_identifier
    | ams_constant_concatenation
constant_primary ::=
    number
    | parameter_identifier
    | constant_concatenation
    | constant_multiple_concatenation

```

### A.9.5 Expression left-side values

### A.9.6 Operators

```

ams_unary_operator ::=
    + | - | ! | ~
unary_operator ::=
    + | - | ! | ~ | & | ~& | | | ~ | ^ | ~^ | ^~
binary_operator ::=
    + | - | * | / | % | == | === | != | !== | && | ||
    | < | <= | > | >= | & | | | ^ | ^~ | ~^ | >> | <<

```

### A.9.7 Numbers

```

number ::=
    decimal_number
    | octal_number
    | binary_number
    | hex_number
    | real_number
decimal_number ::=
    [ sign ] unsigned_number
    | [ size ] decimal_base unsigned_number
sign ::=
    + | -

```

```

size ::=
    unsigned_number
unsigned_number ::=
    decimal_digit { _ | decimal_digit }

binary_number ::=
    [ size ] binary_base binary_digit { _ | binary_digit }
octal_number ::=
    [ size ] octal_base octal_digit { _ | octal_digit }
hex_number ::=
    [ size ] hex_base hex_digit { _ | hex_digit }
decimal_base ::=
    'd' | 'D'
octal_base ::=
    'o' | 'O'
hex_base ::=
    'h' | 'H'
binary_base ::=
    'b' | 'B'
decimal_digit ::=
    0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
binary_digit ::=
    x | X | z | Z | 0 | 1
octal_digit ::=
    x | X | z | Z | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7
hex_digit ::=
    x | X | z | Z | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f | A | B | C | D | E | F
real_number ::=
    [ sign ] unsigned_number . unsigned_number
    | [ sign ] unsigned_number [ . unsigned_number ] e [ sign ] unsigned_number
    | [ sign ] unsigned_number [ . unsigned_number ] E [ sign ] unsigned_number
    | [ sign ] unsigned_number [ . unsigned_number ] scale_factor
scale_factor ::=
    T | G | M | K | k | m | u | n | p | f | a

```

## A.10 General