

Motivation

Current LRM is not clear regarding the argument of bit vector system functions (`$onehot`, `$onehot0`, `$countones`, and `$isunknown`). Their argument is called *expression*. Unpacked data types are expressions according to the BNF. However, the current description does not elaborate the subtleties related to the unpacked data types in the context of the bit vector functions. This proposal explicitly states that the unpacked aggregate types may be arguments of bit vector functions and accurately defines the value of these functions in this case.

Suggested Solution

16.12 System functions

REPLACE

Assertions are commonly used to evaluate certain specific characteristics of a design implementation, such as whether a particular signal is “one-hot”. The following system functions are included to facilitate such common assertion functionality:

- `$onehot (<expression>)` returns true if only 1 bit of the expression is high.
- `$onehot0 (<expression>)` returns true if at most 1 bit of the expression is high.
- `$isunknown (<expression>)` returns true if any bit of the expression is x or z. This is equivalent to `^(<expression>) === 1'bx`.

All of the above system functions have a return type of **bit**. A return value of `1'b1` indicates true, and a return value of `1'b0` indicates false.

Another useful function provided for the Boolean expression is `$countones`, to count the number of ones in a bit vector expression.

```
$countones ( expression)
```

A bit with value X or Z is not counted towards the number of ones.

WITH

Assertions are commonly used to evaluate certain specific characteristics of a design implementation, such as whether a particular signal is “one-hot”. The following system functions are included to facilitate such common assertion functionality:

- `$onehot (<expression>)` returns true if only 1 bit of the expression is high.
- `$onehot0 (<expression>)` returns true if at most 1 bit of the expression is high.
- `$isunknown (<expression>)` returns true if any bit of the expression is x or z. This is equivalent to `^(<expression>) === 1'bx`.

All of the above system functions have a return type of **bit**. A return value of `1'b1` indicates true, and a return value of `1'b0` indicates false.

Another useful function provided for the Boolean expression is `$countones`, to count the number of ones in a bit vector expression.

`$countones (expression)`

A bit with value X or Z is not counted towards the number of ones.

If the argument `expression` of the above functions has an unpacked aggregated data type then the functions are defined recursively as follows:

- `$countones (expression)` returns the sum of `$countones` results applied to all expression elements.
- `$onehot (<expression>)` returns true if for only one element of the expression `$onehot` returns true, and `$countones` for all other elements returns 0.
- `$onehot0 (<expression>)` returns true if for at most only one element of the expression `$onehot0` returns true, and `$countones` for all other elements returns 0.
- `$isunknown (<expression>)` returns true if `$isunknown` returns true for some element of the expression.

If the expression has an unpacked aggregated data type its elements shall not contain unpacked unions.