

27.14 Variables (supersedes 26.6.7 and 26.6.8 of IEEE Std 1364)

ADD

- z) The **vpiParent** transition shall be allowed on all variable objects. It will return one of the types of objects listed below, representing one of its prefix objects (similar to the field select prefix or indexing select prefix as described in 8.11), or NULL, depending on whether certain criteria are met. For purposes of defining **vpiParent**, a prefix object is the object obtained from successively removing the rightmost index or identifier from a compound or indexed/multi-dimensional object name. Consider the following **vpiArrayVar** objects:

```
logic [1:0][2:3] mda [4:6][6:8];
struct { int i1; bit [1:0][2:3] bvec[4:5]; } spa [9:11][12:13];
```

`mda[6][8][1][3]` is a **vpiVarBit**, `mda[6][8][1]` is its first prefix object (a 2-bit **vpiLogicVar** vector), and `mda[6][8]` is its second prefix object (a 2 x 2 **vpiLogicVar** packed array), etc. The `spa[9][12].bvec[4]` object is a **vpiBitVar** (a 2 x 2 **vpiBitVar** packed-array), and `spa[9][12].bvec` is its first prefix object (a **vpiArrayVar** struct member), and `spa[9][12]` is the second prefix object (the **vpiStructVar** containing the `bvec` member), etc.

For a variable object with valid prefix objects (**vpiValid** property == **vpiValidTrue** for all prefix objects), the **vpiParent** transition will return one of the following prefix objects, whichever comes first in prefix order (rightmost to leftmost):

- Struct, union, or class variable
- Struct or union member variable, or class variable data member
- The largest containing packed-array object
- The largest containing unpacked-array object

If there is no prefix object (or a prefix is not valid), or some prefix object does not meet at least one of the above criteria, **vpiParent** will return NULL.

Using the declarations above, the **vpiParent** of `mda[6][8][1][3]` is `mda[6][8]`, the **vpiLogicVar** representing the largest containing packed-array prefix; the **vpiParent** of `mda[6][8]` is `mda`, the **vpiArrayVar** representing the largest containing unpacked-array prefix. Likewise, the **vpiParent** of `spa[9][12].bvec[4][0]` is `spa[9][12].bvec[4]` (the largest containing packed-array); the **vpiParent** of `spa[9][12].bvec[4]` is `spa[9][12].bvec` (struct member), and applying **vpiParent** again yields `spa[9][12]`, the struct variable for member `bvec`. The **vpiParent** of `spa[9][12]` is `spa`, the largest containing unpacked-array of the struct variable; **vpiParent** of `spa` (or `mda`) would return NULL.

Class variables (as mentioned in the prefix object types above) shall be returned as parent objects only when they are valid and explicitly used to reference corresponding class data members in the design. A VPI handle to a data member that does not correspond to such an explicit reference in the design (e.g. a VPI handle to a data member derived from iterations on its **vpiClassObj** or **vpiClassDefn**) shall have a NULL parent.

27.13 Nets (supersedes 26.6.6 of IEEE Std 1364)

ADD

- ab) The **vpiParent** transition shall be allowed on all net objects. It will return one of the types of objects listed below, representing one of its prefix objects (field select prefix or indexing select prefix as described in 8.11), or NULL, depending on whether certain criteria are met. For purposes of defining **vpiParent**, a prefix object is the object obtained from successively removing the rightmost index or identifier from a compound or indexed/multi-dimensional object name. Consider the following **vpiArrayVar** objects:

```
wire logic [1:0][2:3] mda [4:6][6:8];
wire struct { int i1; logic[1:0][2:3]bvec[4:5]; } spa [9:11][12:13];
```

`mda[6][8][1][3]` is a **vpiLogicNet**, `mda[6][8][1]` is its first prefix object (a 2-bit **vpiLogicNet** vector), and `mda[6][8]` is its second prefix object (a 2 x 2 packed array **vpiLogicNet**), etc. The `spa[9][12].bvec[4]` object is a **vpiLogicNet** (a 2 x 2 packed-array **vpiLogicNet**), and `spa[9][12].bvec` is its first prefix object (a **vpiArrayNet** struct member), and `spa[9][12]` is the second prefix object (the **vpiStructNet** containing the `bvec` member). etc.

For a variable object with valid prefix objects (**vpiValid** property == **vpiValidTrue** for all prefix objects), the **vpiParent** transition will return one of the following prefix objects, whichever comes first in prefix order (rightmost to leftmost):

- Struct, or union net
- Struct or union member net
- The largest containing packed-array net object
- The largest containing unpacked-array net object

If there is no prefix object (or a prefix is not valid), or some prefix object does not meet at least one of the above criteria, **vpiParent** will return NULL.

Using the declarations above, the **vpiParent** of `mda[6][8][1][3]` is `mda[6][8]`, the **vpiLogicNet** representing the largest containing packed-array prefix; the **vpiParent** of `mda[6][8]` is `mda`, the **vpiArrayNet** representing the largest containing unpacked-array net prefix. Likewise, the **vpiParent** of `spa[9][12].bvec[4][0]` is `spa[9][12].bvec[4]` (the largest containing packed-array net); the **vpiParent** of `spa[9][12].bvec[4]` is `spa[9][12].bvec` (struct member), and applying **vpiParent** again yields `spa[9][12]`, the struct net for member `bvec`. The **vpiParent** of `spa[9][12]` is `spa`, the largest containing unpacked-array of the struct net; **vpiParent** of `spa` (or `mda`) would return NULL.