

36.16 Variables

ADD

- 30) The **vpiDecompile** and **vpiFullName** properties for variable objects which are members of structs, unions, or class vars shall include their struct, union, or class var name prefixes. Such prefixes shall include all nested levels of **vpiParent** objects sufficient to identify the respective member element in an expression. The **vpiName** property for these objects shall not include such prefixes. The **vpi_handle_by_name** function shall require the **vpiDecompile** form of the name to properly resolve it for any non-top-level scope context, and the **vpiFullName** form shall be required for the top level. If the object is an indexed element or indexed sub-array (slice) of another object, those indices shall be included in **vpiDecompile**, **vpiName**, and **vpiFullName** properties for the object in order to distinguish it from its **vpiParent** object. Examples:

```
module top;
  bit [7:0] arr1 [1:4][9:15];
  struct {
    integer i1;
    logic[1:4]vec[5:8];
    struct {
      shortint j1;
      byte b1;
    } inner1;
  } str1;

  class cdef;
    int cvInt;
  endclass
  cdef cv = new;
endmodule

// Objects from above declarations-
vpiFullName : top.arr1[1][9]
vpiDecompile : arr1[1][9]
vpiName      : arr1[1][9]

vpiFullName : top.str1.i1
vpiDecompile : str1.i1
vpiName      : i1

vpiFullName : top.str1.inner1.j1
vpiDecompile : str1.inner1.j1
vpiName      : j1

vpiFullName : top.str1.vec[5]
vpiDecompile : str1.vec[5]
vpiName      : vec[5]

vpiFullName : top.cv.cvInt
vpiDecompile : cv.cvInt
vpiName      : cvInt
```

36.15 Nets

ADD

- 29) The **vpiDecompile** and **vpiFullName** properties for net objects which are members of structs shall include their struct name prefix. Such prefixes shall include all nested levels of **vpiParent** objects sufficient to identify the respective member element in an expression. The **vpiName** property for these objects shall not include such prefixes. The **vpi_handle_by_name** function shall require the **vpiDecompile** form of the name to properly resolve it for any non-top-level scope context, and the **vpiFullName** form shall be required for the top level. If the object is an indexed element or indexed sub-array (slice) of another net object, those indices shall be included in **vpiDecompile**, **vpiName**, and **vpiFullName** properties for the object in order to distinguish it from its **vpiParent** object. Examples:

```
module top;
  wire logic[7:0] warr1 [1:4][9:15];
  wire struct {
    integer i1;
    logic[1:4]vec[5:8];
    struct {
      time t1;
      integer j1;
    } inner1;
  } str1;
endmodule
```

// Objects from above declarations-

```
vpiFullName : top.warr1[1][9]
vpiDecompile : warr1[1][9]
vpiName      : warr1[1][9]
```

```
vpiFullName : top.str1.i1
vpiDecompile : str1.i1
vpiName      : i1
```

```
vpiFullName : top.str1.inner1.j1
vpiDecompile : str1.inner1.j1
vpiName      : j1
```

```
vpiFullName : top.str1.vec[5]
vpiDecompile : str1.vec[5]
vpiName      : vec[5]
```