

Section 32.21 Class object definition

REPLACE

NOTES

- 1) **ClassDefn** handle is a new concept. It does not correspond to any **vpiUserDefined** (class object) in the design. Rather it represents the actual type definition of a class.
- 2) Should not call **vpi_get_value/vpi_put_value** on the non-static variables obtained from the class definition handle.
- 3) The iterator to constraints returns only normal constraints and not inline constraints.
- 4) To get constraints inherited from base classes, it is necessary to traverse the extend relation to obtain the base class.
- 5) The **vpiDerivedClasses** iterator returns all the classes derived from the given class.
- 6) The relation to **vpiExtend** exists whenever a one class is derived from another class (refer to 12.12). The relation from extend to classDefn provides the base class. The iterators from extend to param assign and arguments provide the parameters and arguments used in constructor chaining (refer to 12.16 and 12.23)

WITH

- ~~1) **ClassDefn** handle is a new concept. It does not correspond to any **vpiUserDefined** (class object) in the design. Rather it represents the actual type definition of a class.~~
- 1) Should not call **vpi_get_value/vpi_put_value** on the non-static variables obtained from the class definition handle.
- 2) The iterator to constraints returns only normal constraints and not inline constraints.
- 3) To get constraints inherited from base classes, it is necessary to traverse the extend relation to obtain the base class.
- 4) The **vpiDerivedClasses** iterator returns all the classes derived from the given class.
- 5) The relation to **vpiExtend** exists whenever a one class is derived from another class (refer to 12.12). The relation from extend to classDefn provides the base class. The iterators from extend to param assign and arguments provide the parameters and arguments used in constructor chaining (refer to 12.16 and 12.23)