Restrict verification statement

Objectives:

Currently SVA supports the verification statements **`assert`**, **`assume`** and **`cover property`**. In formal verification, for the tool to converge on a proof of a property or to initialize the design to a specific state, it is often necessary to constrain the state space. For this purpose, the verification statement **`restrict property`** is introduced. It has the same semantics as assume property, however, in contrast to that statement, verification tools are not obliged to verify that the property holds.

Add in 16.14

— **`assert`** to specify the property as a checker to ensure that the property holds for the design
— **`assume`** to specify the property as an assumption for the environment
— **`cover`** to monitor the property evaluation for coverage
— **`restrict`** to constrain the environment

Insert after 16.14.3.

(Note to editor: please renumber clauses)

17.13.4 Restrict statement

In formal verification, for the tool to converge on a proof of a property or to initialize the design to a specific state, it is often necessary to constrain the state space. For this purpose, the verification statement **restrict property** is introduced. It has the same semantics as **assume property**, however, in contrast to that statement, the **restrict** statement is not verified in simulation.

The statement has the following form

restrict property (property_spec **) ;**

There is no action block associated with the statement.

Example:

Suppose that when a control bit `ctr` has a value 0, an ALU performs an addition, and when it is 1, it performs a subtraction. We want to formally verify that some behavior is correct when ALU does an addition (perhaps in another verification session we will do the same for subtraction). Thus, we can constrain the behavior using

```
restrict property (@(posedge clk) ctr == '0);
```

It does not mean that `ctr` cannot be 1 in any test case in the simulation, this is not an error.

Change Syntax 16-16

procedural_assertion_statement ::= *// from A.6.10*
  concurrent_assertion_statement
| immediate_assert_statement
concurrent_assertion_item ::= [block_identifier **:**] concurrent_assertion_statement *// from A.2.10*
concurrent_assertion_statement ::=
  assert_property_statement
| assume_property_statement
| cover_property_statement
| restrict_property_statement
assert_property_statement::=
  assert property (property_spec **)** action_block
assume_property_statement::=
  assume property (property_spec **)** action_block
cover_property_statement::=
  cover property (property_spec **)** statement_or_null
restrict_property_statement::=
  restrict property (property_spec **) ;**

Change Annex A.2.10 Assertion declaration

concurrent_assertion_statement ::=
  assert_property_statement
| assume_property_statement
| cover_property_statement
| expect_property_statement
| restrict_property_statement
assert_property_statement::=
  assert property (property_spec **)** action_block
assume_property_statement::=
  assume property (property_spec **)** action_block
cover_property_statement::=
  cover property (property_spec **)** statement_or_null
assume_property_statement::=
  restrict property (property_spec **) ;**
...

Change Annex J

From

```
/* concurrent assertions */
#define vpiAssert 686
#define vpiAssume 687
```

```
#define vpiCover 688
```

To

```
/* concurrent assertions */
#define vpiAssert 686
#define vpiAssume 687
#define vpiCover 688
#define vpiRestrict ?? (Editor please assign a code)
```