Minor Corrections

1.1 Page 201 in Section 17.4

Change

There are certain restrictions on the expressions that can appear in concurrent assertions.

Expressions are allowed to include function calls, but certain semantic restrictions are imposed.

— Functions that appear in expressions may not cannot contain output or **ref** arguments (**const ref** are allowed).

— Functions should be automatic (or preserve no state information) and have no side effects.

1.2 Page 214 in Section 17.7.4

Change

The binary operator **and** is used when both operands ~~operand expressions~~ are expected to ~~succeed~~ match, but the end times of the operand ~~expressions~~ sequences can be different.

The two operands of **and** are sequences ~~sequence expressions~~. The requirement for the ~~success~~ match of the **and** operation is that both the operands ~~operand expressions~~ must ~~succeed~~ match. The operand ~~expressions~~ sequences start at the same time. When one of the operand sequences ~~expressions~~ ~~succeeds~~ matches, it waits for the other to ~~succeed~~ match. The end time of the composite sequence ~~expression~~ is the end time of the operand sequence ~~expression~~ that completes last.

When `te1` and `te2` are sequences, then the sequence ~~expression~~:

```
te1 and te2
```

— ~~Succeeds~~ matches if `te1` and `te2` ~~succeed~~ match.

1.3 Page 214 in Section 17.7.4

Change

The following example is ~~an expression~~ a sequence with ~~the and~~ operator **and**, where the two operands are ~~single~~ simple sequence evaluations.

1.4 Page 215 in Section 17.7.4

Change

3) Figure 17-5 shows the attempt to examine at clock tick 8 when both operand sequences start and ~~succeed~~ match. All five sequences for the first operand sequence match, as shown in a time window, at clock ticks 9, 10, 11, 12 and 13 respectively. The second operand sequence matches at clock tick 12.

1.5 Page 216 in Section 17.7.4

Change

If `te1` and `te2` are sampled ~~booleans~~ expressions (not sequences), the ~~expression~~ sequence (`te1` **and** `te2`) ~~succeeds~~ matches if `te1` and `te2` are both evaluated to be true.

1.6 Page 216 in Section 17.7.4

The binary operator **intersect** is used when both operand ~~expressions~~ sequences are expected to ~~succeed~~ match, and the end times of the operand ~~expressions~~ sequences must be the same.

1.7 Page 217 in Section 17.7.4

The two operands of **intersect** are sequences~~sequence expressions~~. The requirements for the success of the **intersect** operation are:

— Both the operands ~~expressions~~ must ~~succeed~~ match.

1.8 Page 250 in Section 17.13.4

A concurrent assertion statement can be used outside of a procedural context. It can be used within a module as

a *module_common_item*, an interface as a *module_common_item*, or a program as a *non_port_item*. A concurrent assertion statement is either an **assert** ,an **assume** or a **cover** statement.