ASSERT

The assertion statement checks a condition and reports a message with a severity level if the condition is not true.

Format:

```
ASSERT condition;

ASSERT condition REPORT “message”

ASSERT condition SEVERITY level;

ASSERT condition REPORT “message” SEVERITY
level;
```

Example:

```
ASSERT signal_input = ‘1’
  REPORT “Input signal_input is not 1”
    SEVERITY WARNING;
```

Severity levels are:

- **Note - general information**
- **Warning - undesirable condition**
- **Error - task completed, result wrong**
- **Failure - task not completed**

Simulators stop when the severity level matches or exceeds the specified severity level.

Simulators generally default to a severity level of “failure”

Assert

Assert statements may appear within:

- **concurrent statement areas**
- **sequential statement areas**
- **statement area of entity declaration**

Example:

```
ENTITY rs_flip_flop IS
  PORT(r, s   :  IN  std_logic;
       q, qn  :  OUT std_logic);
END rs_flip_flop;

ARCHITECTURE behav OF rs_flip_flop IS
BEGIN
  ASSERT NOT (r = ‘1’ AND s = ‘1’)
    REPORT “race condition!”
      SEVERITY FAILURE;
                 *
                 *
                 *
END behav;
```

<u>Remember</u>, the ASSERT statement triggers when the specified condition is *false*.