SynthWorks

VHDL Training Experts

Scoreboard Package Quick Reference

1. Generic Package Interface

```
package ScoreboardGenericPkg is
generic (
  type ExpectedType ;
  type ActualType ;
  function Match(Actual : ActualType ;
     Expected : ExpectedType)
     return boolean ;
  function expected_to_string(
     A : ExpectedType) return string ;
  function actual_to_string(
     A : ActualType) return string
) ;
```

ExpectedType is the input to the scoreboard. ActualType is the value to be checked against the oldest ExpectedType value. ExpectedType and ActualType may be the same type. Match is a function that compares ExpectedType with ActualType. Match is often mapped to "=". expected_to_string converts ExpectedType to a string value (for reports). actual_to_string converts ActualType to a string.

2. Package Instance

```
library ieee ;
  use ieee.std_logic_1164.all ;
  use ieee.numeric_std.all ;

package ScoreBoardPkg_slv is new
work.ScoreboardGenericPkg
  generic map (
    ExpectedType => std_logic_vector,
    ActualType   => std_logic_vector,
    Match              => std_match,
    expected_to_string  => to_hstring,
    actual_to_string    => to_hstring
) ;
```

Note, that ExpectedType and ActualType do not need to be constrained types.

3. Compatibility Package

ScoreboardPkg_slv_c.vhd contains equivalent subtypes and aliases to work like the above package instance.

4. Basic Operations

4.1 Create a ScoreboardID

A Scoreboard ID is a handle to the scoreboard object that is created internal to an instance of ScoreboardGenericPkg.

```
signal SB : ScoreBoardIDType ;
```

If using more than one scoreboard package instance, disambiguate the type using a fully selected name.

```
signal SB_int :
 work.ScoreBoardPkg_int.ScoreBoardIDType;
```

4.2 Construct the Scoreboard

NewID allocates and initializes the scoreboard data structure. It must be called before any other scoreboard operations are used.

```
SB <= NewID("UART_SB") ;
```

4.3 Push

Add expected value (ExpectedType) to the scoreboard.

```
Push(SB, ExpectedVal) ;
```

4.4 Check

Check a received value (ActualType) with value in scoreboard. Reports PASSED/ERROR via AffirmIf.

```
Check(SB, ReceiveVal) ;
```

4.5 Pop

Get oldest value from FIFO and remove it. Uses an out mode variable parameter of ExpectedType.

```
Pop(SB, FifoVal) ;
```

Can also be called as a function:

```
FifoVal := Pop(SB) ;
```

4.6 Peek

Get oldest value from FIFO, but do not remove it. Uses an out mode variable parameter of ExpectedType.

```
Peek(SB, FifoVal) ;
```

Can also be called as a function:

```
FifoVal := Peek(SB) ;
```

4.7 GetAlertLogID

Get the AlertLogID from the scoreboard internals.

```
SB_ID := GetAlertLogID(SB) ;
```

4.8 Empty

Check if the Scoreboard is empty.

```
if not Empty(SB) then …
```

4.9 Gettting Error Counts

Errors are recorded in the AlertLog data structure. Retrieve the error count using the AlertLogID:

```
Err := GetAlertCount(GetAlertLogID(SB));
```

4.10 Find

Return the ItemNumber of the oldest expected value that matches the received value. Find + Flush support systems that drop items before they are synchronized.

```
ItemNum := Find(SB, ReceiveVal) ;
```

4.11 Flush

Quietly drop all values whose item number is less than the specified item number. Find + Flush support systems that drop items before they are synchronized.

```
Flush(SB, ItemNum) ;
```

4.12 GetPushCount

Get number of items pushed into the scoreboard.

```
print("…" & to_string(GetItemCount(SB)));
```

4.13 GetCheckCount

Get number of items checked by the scoreboard.

```
print(to_string(GetCheckCount(SB)));
```

4.14 GetPopCount

Get number of items popped out of the scoreboard.

```
print("…" & to_string(GetPopCount(SB)));
```

4.15 GetDropCount

Get number of items dropped by the scoreboard.

```
print("…" & to_string(GetDropCount(SB)));
```

4.16 GetFifoCount

Get number of items in the FIFO inside the scoreboard.

```
print("…" & to_string(GetFifoCount(SB)));
```

SynthWorks

VHDL Training Experts

5. Tagged Scoreboards

Tagged Scoreboards are used for systems that allow transactions to execute out of order.

Tags are represented as string values (since most types convert to string using to_string). A tag value is specified as the first value in the calls to push, check, and pop, such as shown below. In all examples, ExpectedVal has the type ExpectedType, and ReceiveVal has the type ActualType.

```
Push(SB, "WriteOp", ExpectedVal) ;
Check(SB, "WriteOp", ReceiveVal) ;
Pop(SB, "WriteOp", ExpectedVal) ;

if Empty(SB, "MyTag")  then   …
```

For Check (and Pop), the item checked (or returned) is the oldest item with the matching tag.

```
ItemNum := Find(SB, "ReadOp", RxVal);
Flush(SB, "ReadOp", ItemNum) ;
```

For Flush, only items matching the tag are removed. In some systems, it may be appropriate to do the Find with the tag and the flush without the tag.

6. Indexed Scoreboards

Indexed scoreboards are for systems, such as a network switch that have multiple scoreboards that are most conveniently represented as an array.

ScoreboardIDType is an ordinary type, so normal VHDL methods can be used to create arrays of it.

6.1 1D and 2D Array Types

ScoreboardGenericPkg defines the following types:

```
-- 1D array type
type ScoreboardIdArrayType  is array (integer
range <>) of ScoreboardIdType ;

-- 2D array type
type ScoreboardIdMatrixType is array (integer
range <>, integer range <>)
    of ScoreboardIdType ;
```

6.2 1D Arrays of Scoreboards

The following operations are appropriate for any array of scoreboards. Procedures and functions not documented here are from AlertLogPkg.

```
signal SB :
  ScoreboardIDArrayType(1 to 3);
. . .

-- Create 3 indexed scoreboards
SB <= NewID("SB", 3);
wait for 0 ns ;

-- Push values into scoreboards
Push(SB(1), X"11") ;
Push(SB(2), X"2222") ;
Push(SB(3), X"3") ;

-- Check values using scoreboards
Check(SB(3), X"3") ;
Check(SB(2), X"2222") ;
Check(SB(1), X"01") ;

-- Includes Scoreboard errors
ReportAlerts ;
```

6.3 1D Arrays of Scoreboards Again

The previous example used a simple form of NewID since the left array index is 1. If it is not 1, such as below, the second form of NewID is needed. This is shown below.

```
signal SB :
  ScoreboardIDArrayType(5 to 7);
. . .

SB <= NewID("SB", (5, 7));
```

6.4 2D Arrays of Scoreboards

The following demonstrates the creation of 2D arrays of scoreboards. Note in SB1, the left array indices are 1 and a simpler form of NewID can be used.

```
signal SB1 :
  ScoreboardIDArrayType(1 to 3, 1 to 5);
signal SB2 :
  ScoreboardIDArrayType(1 to 3, 5 to 7);. . .

SB1 <= NewID("SB1", 3, 5 );
SB2 <= NewID("SB", (1, 3), (5,7) );
```


SynthWorks Design Inc.
VHDL Design and Verification Training
11898 SW 128th Ave. Tigard OR 97223 (800)-505-8435
http://www.SynthWorks.com jim@synthworks.com