



# VHDL/Verilog-HDL テストベンチ・サンプル記述集

鳥海佳孝, 田原迫仁治, 横溝憲治

本連載の第1回(本誌2000年8月号, pp.66-78に掲載)では、「テストベンチとはなにか」、そして「HDLによるLSI設計におけるテストベンチの重要性」を簡単に説明しました。今回はもう一步踏み込んで、実際に検証を行う上で必要となる考え方を紹介してみようと思います。なお本記事で紹介する記述はModelSim, VeriLogger Proなどで動作を確認しています。

## 1. VHDL テストベンチの記述法(基礎編)～承前～ ——検証結果をモニタに表示する方法

前回は、`assert FALSE report STRING severity LEVEL`の使い方まで解説しましたが、今回はこの続きから始めます。復習のため、もう一度簡単に説明しておきましょう。VHDLにおけるモニタ上へのメッセージ出力には、

- `assert ... report` 文を用いること
  - VHDL93 では `report` 文を直接利用できること
- などを解説しました。

〔リスト1〕 `std_logic_vector` から `string` への変換 (VHDL)

```
function stdv2str(vec:std_logic_vector) return string is
variable str: string(vec'left+1 downto 1);
begin
for i in vec'reverse_range loop
if(vec(i)='U') then
str(i+1):='U';
elsif(vec(i)='X') then
str(i+1):='X';
elsif(vec(i)='0') then
str(i+1):='0';
elsif(vec(i)='1') then
str(i+1):='1';
elsif(vec(i)='Z') then
str(i+1):='Z';
elsif(vec(i)='W') then
str(i+1):='W';
elsif(vec(i)='L') then
str(i+1):='L';
elsif(vec(i)='H') then
str(i+1):='H';
else
str(i+1):='-';
end if;
end loop;
return str;
end;
```

①  
②  
③  
④  
⑤  
⑥

さて、`report` 文に続くテキストの文字列は" "で囲んで出力しますが、`string` タイプしか扱えないのが `report` 文の最大の欠点です。実際には、シミュレーション中の信号の状態など、さまざまなデータ・タイプ(`sdt_logic` など)の情報をモニタに表示させたいのが、一般的だと思います。

### ● `std_logic(_vector)` のモニタ表示

ここでは、データ・タイプ `std_logic_vector` や `std_logic` を例に、モニタに出力する方法を解説します。`report` 文が `string` データ・タイプしか扱えないわけですから、方法としては `std_logic_vector` や `std_logic` を `string` データ・タイプに変換した後、`report` 文に渡すことが考えられます。これには、サブプログラムの `function` 文を使います。リスト1を見てください。

①から、`function` 名が `stdv2str`、入力パラメータが `vec` であることがわかります。また、データ・タイプは `std_logic_vector` で、戻り値が `string` になります。

②では、変数 `str` を宣言し、データ・タイプ `string` と範囲を指定しています。`vec'left+1` ですが、アトリビュートを用いて、`std_logic_vector` の範囲に関係なく `function` 文が使えるように、`std_logic_vector` の左の値と連動させています。また、ここで `+1` しているのは、`string` の範囲として0が使えないためです。`downto 0` とは書けないので、`string` の左値にも右値にも `+1` しています。

③では、ベクタの長さの分だけ処理をループさせます。`'reverse_range` というアトリビュートで `1 to vec'left+1` の順になっていることに注意してください。

④以降の `if` 文で、`std_logic(_vector)` がとりうる値をすべて、キャラクタに置き換えます。⑤でループを閉じ、⑥で戻り値を与えて終わりです。これで、`std_logic_vector` から `string` への変換が可能になります。

さて、では実際に、このサブプログラムでデータ・タイプ `std_logic_vector` を表示してみましょう。サブプログラムを使う方法は、2通りあります。一つは `architecture` 文の `architecture` と `begin` の間に記述する方法、もう一つは

〔リスト2〕 変換のサブプログラムを architecture 内に記述した例 (VHDL)

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> library IEEE; use IEEE.std_logic_1164.all ; use IEEE.std_logic_unsigned.all ;  entity SFTRG_TEST is end SFTRG_TEST;  architecture SIM1 of SFTRG_TEST is  component SFTRG   port (     CLK : in std_logic;     RST : in std_logic;     SEL : in std_logic;     S : in std_logic;     P : in std_logic_vector(3 downto 0);     Q : out std_logic_vector(3 downto 0)   ); end component;  constant CLK_CYCLE : time := 100 ns; signal CLK_I : std_logic; signal RST_I : std_logic:= '1'; signal SEL_I : std_logic:= '0'; signal S_I : std_logic:= '1'; signal P_I : std_logic_vector(3 downto 0) := "1111"; signal Q_0 : std_logic_vector(3 downto 0); </pre> | <pre> --- function ----- function stdv2str(vec:std_logic_vector)   return string is   variable str: string(vec'left+1 downto 1);   begin     for i in vec'reverse_range loop       if(vec(i)='U') then         str(i+1):='U';       elsif(vec(i)='X') then         str(i+1):='X';       elsif(vec(i)='0') then         str(i+1):='0';       elsif(vec(i)='1') then         str(i+1):='1';       elsif(vec(i)='Z') then         str(i+1):='Z';       elsif(vec(i)='W') then         str(i+1):='W';       elsif(vec(i)='L') then         str(i+1):='L';       elsif(vec(i)='H') then         str(i+1):='H';       else         str(i+1):='-';       end if;     end loop;     return str;   end; </pre> | <pre> ----- begin   U0: SFTRG port map (CLK=&gt;CLK_I, RST=&gt;     RST_I, SEL=&gt;'1', S=&gt;S_I,     P=&gt;P_I, Q=&gt;Q_0);    process begin     CLK_I &lt;= '1';     wait for CLK_CYCLE/2;     CLK_I &lt;= '0';     wait for CLK_CYCLE/2;   end process;   process   begin     wait for 100 ns;     assert FALSE --VHDL87       report "P_I=" &amp; stdv2str(P_I)         severity NOTE; ⑦     assert FALSE -- VHDL87       report "Finished." severity NOTE;     wait;   end process; end SIM1; configuration CFG_SFTRG_TEST of SFTRG_TEST is   for SIM1   end for; end CFG_SFTRG_TEST; </pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

〔リスト3〕 変換のサブプログラムを package 文で記述した例 (VHDL)

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> library IEEE; use IEEE.std_logic_1164.all ;  package PCNV is   function stdv2str(vec:std_logic_vector) return string; ⑧ end PCNV; package body PCNV is   -- std_logic_vector to string   function stdv2str(vec:std_logic_vector) return string is     variable str: string(vec'left+1 downto 1);     begin       for i in vec'reverse_range loop         if(vec(i)='U') then           str(i+1):='U';         elsif(vec(i)='X') then           str(i+1):='X';         elsif(vec(i)='0') then           str(i+1):='0';         elsif(vec(i)='1') then           str(i+1):='1';         elsif(vec(i)='Z') then </pre> | <pre>           str(i+1):='Z';         elsif(vec(i)='W') then           str(i+1):='W';         elsif(vec(i)='L') then           str(i+1):='L';         elsif(vec(i)='H') then           str(i+1):='H';         else           str(i+1):='-';         end if;       end loop;       return str;     end;   end PCNV; </pre> |
| <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;">         テストベンチ側へのライブラリ宣言の追加       </div> <pre> library IEEE; use IEEE.std_logic_1164.all ; use IEEE.std_logic_unsigned.all ; use WORK.PCNV.all ; ⑨ </pre>                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                            |

package 文に記述する方法です。前者は、当然のことながら、同一の architecture 内でしか使うことができません。リスト2に architecture 内に記述した例、リスト3に package 文で記述した例を示します。

リスト2のなかでは、⑦の report 文で function を使用し、std\_logic\_vector の信号 P\_I の値を表示しています。リスト2を実行したときの出力例は、以下のようになります。

```
# ** Note: P_I=1111
```

リスト3は package 文を使用した例です。⑧でサブプログラムを宣言し、package body のなかに function 文を記述します。⑨では、WORK lib に置いたパッケージ PCNV を使用することを宣言しています。この場合、リスト2と違って、⑨の宣言を記述することによって、どの architecture 内でも stdv2str function を使用することができます。リスト4に std\_logic の function の例をあげておきます。

ただし、このようなめんどろなことをしなければならないのは VHDL87 の場合です。VHDL93 では、'IMAGE'、'VALUE' という新しいアトリビュートが用意されています。構文は、以下のようになります。

```
T' IMAGE (X)
```

T: Data Type

X: Value of Expression

たとえば、

```
report std_logic' IMAGE (信号名);
report Time' IMAGE (NOW);
```