VHDL- 9

総括編　シミュレータでのパッケージの利用

この文書の目的

これまでにＶＨＤＬ開発環境についてはかなりの解説をしてきたつもりなのですが、開発初期段階では論理的な整合を見る為に機能シミュレーションだけを手軽に簡単にやっておきたい場合があるのではないかと思います。

このような初期段階では、ＶＨＤＬでの回路記述について少し工夫をしないと、エンティティ部分の接続記述の部分が多くの行をとりすぎて全体像を判読しにくくするソースコードになるように思います。

すでに良くご存知の事かも知れませんが、シミュレーションだけを考えた時には、複数のエンティティ・アーキテクチャ・ペアを構造記述によって互いに接続する方法の他に、共有パッケージによって互いの接続を行うことができます。

この共有パッケージを使う方法で、初期設計で行うシミュレーション用記述を整理する事を今回の例題にしてみたいと思います。（論理合成はできませんので誤解されないようお願いします）

共有パッケージ

という呼び方が正しいか否かは正直言ってわからないのですが、１つのデバイスの中に、複数の、ある意味を持ってまとめた機能モジュールを組込む事はよく起きる事だと思います。

直接にデバイスのピンへ繋ぐ信号については「名前による接続」が普通の方法として使われていると思いますが、同じ方法を内部の複数のモジュール間だけで行われる接続にも適用しますと、どうしても非常に長くて読みにくいソースファイルになってしまいます。

ここではエンティティ部でそれらの局所的配線を記述せず、あるパッケージを作成してこの中に号を記述し、これらの複数のモジュール毎に use 文節によってそのパッケージを共有させる方法によって、ローカルな接続を実現する事を考えてみます。

今回のソースファイルは３つの部分に分かれています

１、パッケージ

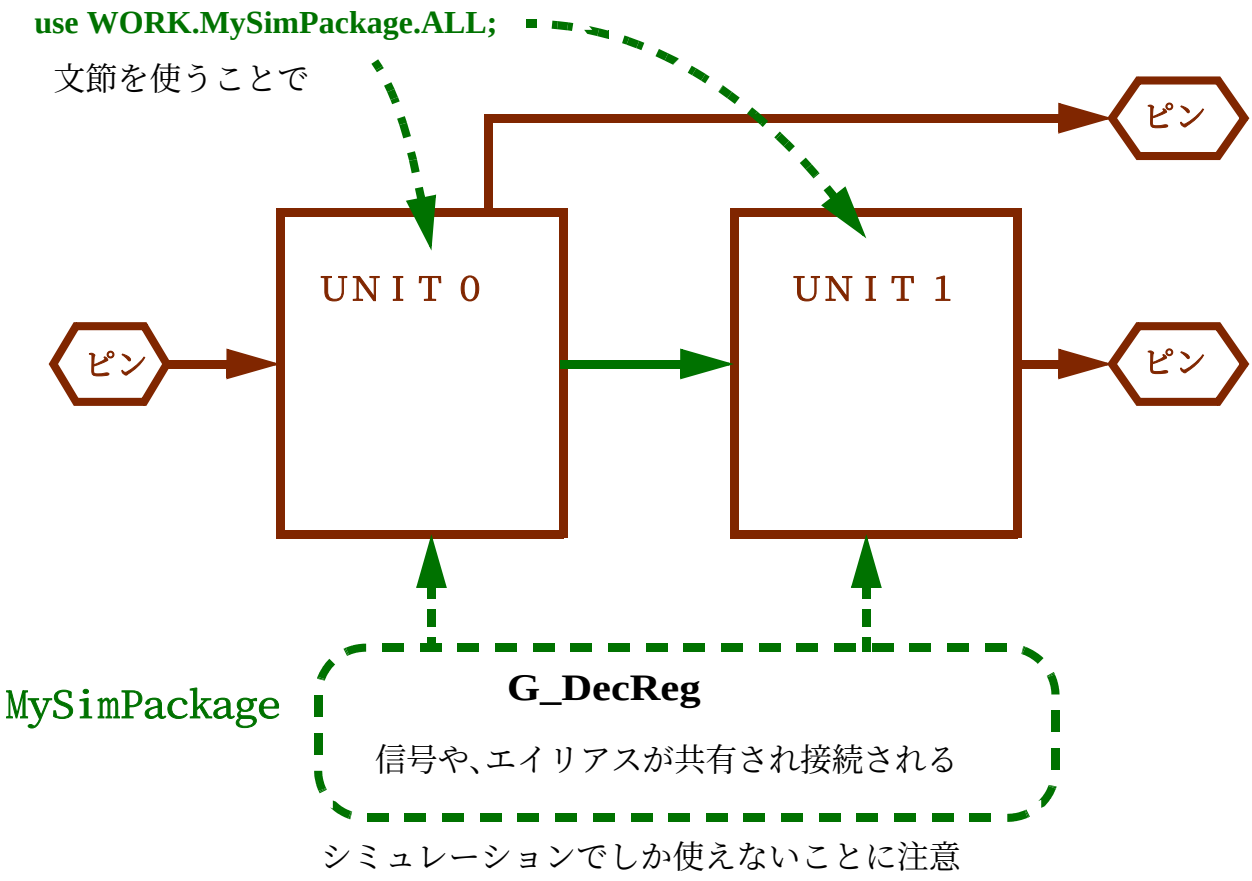
２、本文

３、テスト・ベンチ

これらを１つのディレクトリ（soo）にコピーしておいてください。

イメージ図

この例題で考えているイメージは以下のようになります。



Ｖｓｙｓ の操作は

初めにメニュー / FILE / DIRECTRY からディレクトリを soo に設定し、メニュー / PROJECT / NEW で現在ディレクトリに VSYSTEM.INI を作ります。

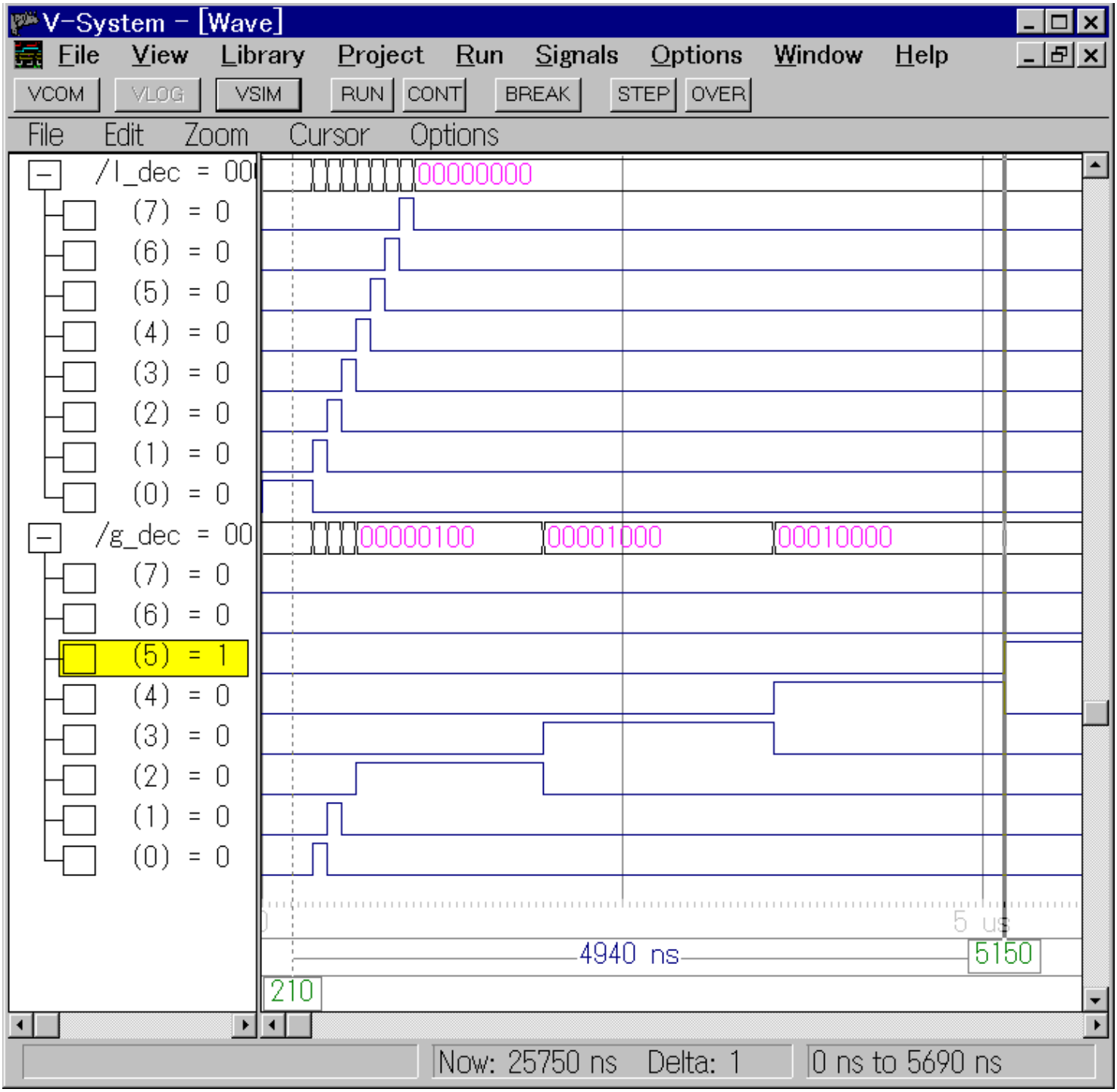
メニュー / FILE / Libray / NEW で WORK を作成しておいて下さい。

以下、ＶＣＯＭ にて

最初に パッケージファイル SOOPACK.VHD を読込み、

例題ファイル SOO.VHD を読込み

最後にテストベンチ SOOTB.VHD を読込んでください。



コンパイルが完了したら、メニューバーのボタンからＶＳＩＭし、現われたダイアログボックスでエンティティ tb_unit のアーキテクチャ testbench を指定して view wave し、run -all としてください。

ＶＳＩＭの結果は上図に示すようになります。(適当にズームなどを使ってみて下さい)

今回の例では、ウインドウズのプリント・スクリーン機能を使ってＯＬＥ対応のエディタにＶｓｙｓの画面を貼り付けてあります。

今のところ、この方法以外にＶｓｙｓ のカーソル情報などを保存する手段が見つかりません。 このような貼り付け操作の可能なワープロとしては、ＭＳ ワードなどが、あることはよくご存知の通りです。

まとめ

これを maxplus2/ leonardo でコンパイルすることはできません。

この例では内部接続を共通のパッケージを使う事でシミュレータ用に実現しており、グローバルなシグナルを使う事で記述を簡便にしています（もっと大きな設計ではこの効果がはっきりとでてきます）が、ブロック図を検討しているような段階では ＲＴＬ 合成の必要はそれほど高くないので、このような局面での「一応使えるテクニック」としてご紹介するにとどめておきたいと思います。

ただ、そのような場合であっても、今回の例題のように ＲＴＬ 論理合成に使う目的のパッケージと、シミュレーション用のパッケージとは別の名前（ファイル名という意味ではなく）で明確に分離しておかないと大きな混乱を招くことになるように思われます。

SooPack.vhd

```
-- PACKAGE DEF ------------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;
--------------------------------------------------------------------------------
package MyPackage is

subtype BITVEC7 is  BIT_VECTOR(7 downto 0);

constant BIOSEL0: INTEGER:=0;
constant BIOSEL1: INTEGER:=1;
constant BCS0:    INTEGER:=2;
constant BCS1:    INTEGER:=3;
constant BCS2:    INTEGER:=4;
constant BCS3:    INTEGER:=5;
constant BMSEL0:  INTEGER:=6;
constant BMSEL1:  INTEGER:=7;

constant A_IOSEL0: INTEGER:= 16#00#;
constant A_IOSEL1: INTEGER:= 16#01#;
constant A_CS0   : INTEGER:= 16#03#;
constant A_CS1   : INTEGER:= 16#10#;
constant A_CS2   : INTEGER:= 16#20#;
constant A_CS3   : INTEGER:= 16#30#;
constant A_CSend : INTEGER:= 16#3F#;

constant A_MSEL0 : INTEGER:= 16#40#;
constant A_MSEL1 : INTEGER:= 16#80#;
constant A_MEMend: INTEGER:= 16#CF#;

FUNCTION Decoder7( INPUT: BIT_VECTOR(7 downto 0); INDEX:INTEGER ) return BIT;
FUNCTION BitV2Int( INPUT: BIT_VECTOR(7 downto 0) )          return INTEGER;

FUNCTION SETBIT_7( INDEX: INTEGER) return BITVEC7;

end MyPackage;
--------------------------------------------------------------------------------

-- PACKAGE BODY ----------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all ;
--------------------------------------------------------------------------------
package body MyPackage is

    FUNCTION BitV2Int( INPUT: BIT_VECTOR(7 downto 0)) return INTEGER is
    Begin   return CONV_INTEGER( TO_STDLOGICVECTOR('0' & INPUT) );
    end BitV2Int;

    FUNCTION Decoder7(INPUT: BIT_VECTOR(7 downto 0);
                      INDEX: INTEGER  ) return BIT is
        variable TEMP: BIT;
    Begin
        if( INDEX = BitV2Int(INPUT) )then
                TEMP:= '1';
        else    TEMP:= '0';
        end if;
        return TEMP;
    end Decoder7;

    FUNCTION SETBIT_7(INDEX: INTEGER) return BITVEC7 is
        variable TEMP: BIT_VECTOR(7 downto 0);
    Begin
        for I in 0 to 7 loop
            if(I = INDEX) then  TEMP(I):='1';
            else                TEMP(I):='0';
            end if;
        end loop;
        return TEMP;
    end SETBIT_7;
```

SooPack.vhd

```
70      end MyPackage;
        ------------------------------------------------------------------------------



75      ------------------------------------------------------------------------------
        package MySimPackage is

        -- SIMULATION ONLY( GLOBAL SIGNALS )----------------
        SIGNAL G_DecReg     : BIT_VECTOR(7 downto 0);
80      SIGNAL G_DecArry    : BIT_VECTOR(7 downto 0);
            ALIAS IOSEL0    : BIT is G_DecArry(0);
            ALIAS IOSEL1    : BIT is G_DecArry(1);
            ALIAS CS0       : BIT is G_DecArry(2);
            ALIAS CS1       : BIT is G_DecArry(3);
85          ALIAS CS2       : BIT is G_DecArry(4);
            ALIAS ALIAS CS3     : BIT is G_DecArry(5);
            ALIAS MSEL0     : BIT is G_DecArry(6);
            ALIAS MSEL1     : BIT is G_DecArry(7);
        ------------------------------------------------------
90      end MySimPackage;
        ------------------------------------------------------------------------------
```

soo.vhd

```
-- UNIT0 ---------------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use WORK.MyPackage.ALL;
-------------------------------------------------------------------------------
ENTITY UNIT0 is
PORT (  CLK,CLR : in    BIT;
        DIN     : in    BIT_VECTOR( 7 downto 0);
        L_DECOUT: out   BIT_VECTOR( 7 downto 0);
        DOUT    : out   BIT_VECTOR( 7 downto 0) );
end UNIT0;

-------------------------------------------------------------------------------
use WORK.MySimPackage.ALL;
-------------------------------------------------------------------------------
ARCHITECTURE RTL of UNIT0 is
    signal L_DECREG : BIT_VECTOR(7 downto 0);
Begin
    Process (CLK,CLR,DIN)
    Begin
        if(CLR='1') then
            L_DECREG <=X"00";
            G_DecReg <=X"00";
        elsif(CLK'EVENT and CLK='1') then
            L_DECREG <= DIN;
            G_DecReg <= DIN;            -- REG into G_DecReg / Package
        end if;
        DOUT <= L_DECREG;

        L_DECASGN:for I in 0 to 7 loop
            L_DECOUT(I)<= Decoder7(L_DECREG,I);  --  "Decoder7" func package
        end loop L_DECASGN;

    end Process;
end RTL;

-- UNIT1 ---------------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use WORK.MyPackage.ALL;
-------------------------------------------------------------------------------
ENTITY UNIT1 is
PORT (  G_DECOUT: out   BIT_VECTOR( 7 downto 0);  -- Note: No input signals.
        LED0    : OUT   BIT;
        LED1    : OUT   BIT );
end UNIT1;

-------------------------------------------------------------------------------
use WORK.MySimPackage.ALL;
-------------------------------------------------------------------------------
ARCHITECTURE RTL of UNIT1 is
Begin
    Process (G_DecReg)      -- REG'd signal from G_DecReg / Package
    Begin
        CASE BitV2Int(G_DecReg) is
            when A_IOSEL0             => G_DecArry<=SETBIT_7(BIOSEL0);
            when A_IOSEL1             => G_DecArry<=SETBIT_7(BIOSEL1);
            when A_CS0 to (A_CS1 - 1) => G_DecArry<=SETBIT_7(BCS0);
            when A_CS1 to (A_CS2 - 1) => G_DecArry<=SETBIT_7(BCS1);
            when A_CS2 to (A_CS3 - 1) => G_DecArry<=SETBIT_7(BCS2);
            when A_CS3 to  A_CSend    => G_DecArry<=SETBIT_7(BCS3);
            when A_MSEL0 to A_MSEL1-1 => G_DecArry<=SETBIT_7(BMSEL0);
            when A_MSEL1 to A_MEMend  => G_DecArry<=SETBIT_7(BMSEL1);
            when OTHERS =>               G_DecArry<=X"00";
        end CASE;
        G_DECOUT <= G_DecArry;
    end Process;

    LED0 <=   IOSEL0;
    LED1 <=   IOSEL1;
```

soo.vhd

```
    end RTL;
    ---------------------------------------------------------------------------

75  -- TOPUNIT -----------------------------------------------------------------
    library ieee;
    use ieee.std_logic_1164.all;
    use WORK.MyPackage.ALL;
    ---------------------------------------------------------------------------
80  ENTITY TOPUNIT is
    PORT (  P_CLK   : in    BIT;
            P_CLR   : in    BIT;
            P_DIN   : in    BIT_VECTOR( 7 downto 0);
            P_L_DEC : out   BIT_VECTOR( 7 downto 0);
85          P_DOUT  : out   BIT_VECTOR( 7 downto 0);
            P_G_DEC : out   BIT_VECTOR( 7 downto 0);
            P_LED0  : OUT   BIT;
            P_LED1  : OUT   BIT );

90  end TOPUNIT;

    ---------------------------------------------------------------------------
    use WORK.MySimPackage.ALL;
    ---------------------------------------------------------------------------
95  ARCHITECTURE RTL of TOPUNIT is

    component UNIT0     -------------------------------------------------------

    port(   CLK : in    BIT;
            CLR : in    BIT;
100         DIN     : in    BIT_VECTOR( 7 downto 0);
            L_DECOUT: out   BIT_VECTOR( 7 downto 0);
            DOUT    : out   BIT_VECTOR( 7 downto 0)  );
    end component;

105 component UNIT1     -------------------------------------------------------

    port(   G_DECOUT: out   BIT_VECTOR( 7 downto 0);
            LED0    : OUT   BIT;
            LED1    : OUT   BIT  );
    end component;
110
    Begin
        UUT0: UNIT0
        PORT MAP(   CLK     =>  P_CLK,
                    CLR     =>  P_CLR,
115                 DIN     =>  P_DIN,
                    L_DECOUT=>  P_L_DEC,
                    DOUT    =>  P_DOUT   );

        UUT1: UNIT1
120     PORT MAP(   G_DECOUT=>  P_G_DEC,
                    LED0    =>  P_LED0,
                    LED1    =>  P_LED1      );

    end RTL;
125 ---------------------------------------------------------------------------
```

soo TB

```
-- TB ---------------------------------------------------------------------------
entity TB_UNIT is     end;
-------------------------------------------------------------------------------

library IEEE;
use IEEE.std_logic_1164.all;
use ieee.std_logic_ARITH.all ;
use ieee.std_logic_unsigned.all ;

-------------------------------------------------------------------------------
ARCHITECTURE TESTBENCH OF TB_UNIT is
-------------------------------------------------------------------------------

component TOPUNIT   ----------------------------------------------------------

port(   P_CLK   : in    BIT;
        P_CLR   : in    BIT;
        P_DIN   : in    BIT_VECTOR( 7 downto 0);
        P_L_DEC : out   BIT_VECTOR( 7 downto 0);
        P_DOUT  : out   BIT_VECTOR( 7 downto 0);
        P_G_DEC : out   BIT_VECTOR( 7 downto 0);
        P_LED0  : OUT   BIT;
        P_LED1  : OUT   BIT  );
end component;

SIGNAL CLOCK,RESET  : BIT;
SIGNAL DATAI,DATAO  : BIT_VECTOR(7 DOWNTO 0);
SIGNAL L_DEC,G_DEC  : BIT_VECTOR(7 DOWNTO 0);
SIGNAL LED0, LED1   : BIT;

-------------------------------------------------------------------------------
Begin
    ---------------------------------------------------------------------------
    UUTOP: TOPUNIT
    PORT MAP(   P_CLK   =>  CLOCK,
                P_CLR   =>  RESET,
                P_DIN   =>  DATAI,
                P_L_DEC =>  L_DEC,
                P_G_DEC =>  G_DEC,
                P_DOUT  =>  DATAO,
                P_LED0  =>  LED0,
                P_LED1  =>  LED1    );

    RESET <= '1', '0' after 200 ns;

    CLK_GEN : Process
        Begin   CLOCK<='0'; wait for 50 ns;
                CLOCK<='1'; wait for 50 ns;
        end Process CLK_GEN;

    ADRSPUT: Process(CLOCK)
    variable ADRVALUE : INTEGER RANGE 0 to 16#FF# ;
        Begin
            if(RESET='1') then
                ADRVALUE :=0;
            elsif(CLOCK'EVENT and CLOCK='1') then
                ADRVALUE := ADRVALUE +1;
            end if;
        DATAI <= TO_BITVECTOR(CONV_STD_LOGIC_VECTOR(ADRVALUE,8));
    end Process ADRSPUT;

end TESTBENCH;
-------------------------------------------------------------------------------
```