# VHDL- 7 x

## 備忘録編　VHDL87 / 93 ついて

### この文書の目的

VHDL 言語について、その使用時の注意点幾つか気がついた事を断片的に備忘録として作成、保存しておきたいと思います。

### VHDL87

Vsystem/Leonardo では、コンパイル／シミュレーション時にオプションメニューから VHDL87 / 93 仕様についての選択指定が可能です。

基本的には、言語仕様に改定があった場合にはできるだけ新しい仕様を使うべきだとは思うのですが、市販の書籍で紹介されている構文例題は逆に、殆どが VHDL87 で記述されています。

VHDL87 での記述方法でそれなりの習熟を重ねてから VHDL93 仕様を積極的に使用してゆくステップを踏むのが（私にとっては）より良い在り方であるように思われますので、これまでの記述ではすべて VHDL87 で処理できるように注意してきました。

但し、textio を使う場合には VHDL87 の規定は非常に重い障壁となりますので、いつまでも VHDL87 にこだわりつづける事には意味がないように思います。

また、VHDL87 / 93 仕様とは無関係に注意すべき点としては、論理合成時、コンパイラによって（アルテラ社の VHDL リーダなど）はローカルなスタティックのみを認め、グローバルなスタティックを受け付けないものがあります。

ローカルにスタティックとは、コンパイル時にその値が確定できるものを指して言います。　これに対し、グローバルにスタティックとは、全てのエンティティがコンパイルされ、エラボレートされた（インスタンス化されて初期化された）時に、その値が確定するものを指して言います。

具体的なバグ発生の原因例としては、ジェネリックスの不用意な取扱いなどがあるように思います。

VHDL87 / 93 でコンパイルした場合、シミュレーション時に使えるバイタルライブラリが 2.2B になるか、VITAL95 を前提としているか、などの違いが出てくる場合がある点にも留意しておきます。

## その他の VHDL 処理系

ではどのような相違があるか、例題はザイリンクス社のファウンデーションのインストールディレクトリの以下のフォルダにありますので参照してください。

X : \ FNDTN \ ACTIVE \ VHDL \ EX \ PREP9.VHD

```
package typedef is
    subtype byte is bit_vector(7 downto 0);
end ;

use work.typedef.all;

entity prep9 is
    port    (clk,rst,as,ce : boolean; al,ah : byte;
             be : out boolean; q : out byte);
    end prep9;

architecture only_level of prep9 is
    procedure decoder( signal clk,rst,as : boolean;
                       signal al,ah : byte;
                       signal be : out boolean;
                       signal q : out byte) is
    begin
      if rst then   q <= x"00";
                    be <= false;
      elsif clk and clk'event then
      if as then
        be <= false;
        case ah & al is
            when x"f000" to x"ffff"     => q <= x"80";
            when x"efff" downto x"e800" => q <= x"40";
            when x"e7ff" downto x"e400" => q <= x"20";
            when x"e3ff" downto x"e300" => q <= x"10";
            when x"e2ff" downto x"e2c0" => q <= x"08";
            when x"e2bf" downto x"e2b0" => q <= x"04";
            when x"e2af" downto x"e2ac" => q <= x"02";
            when x"e2ab"                => q <= x"01";
            when others                 => q <= x"00";
                       be <= true;
          end case;
      else
          q <= x"00";
          be <= false;
      end if;
      end if;
    end;
    signal q1,q2 : byte;
begin
    one : decoder(clk,rst,as,al,ah,be,q);
end;
```

実際にこれらのファイルが Vsys でコンパイルできるか否かは別の次元の話として、このザイリンクス・ファウンデーションのシステムが提供しているライブラリは X: \ FNDTN\ ACTIVE \ VHDL \ VHDL_LIB にありますので今回の勉強の材料にしてみます。

正直、何も考えず、直接的にこういう記述が随所でできると設計が楽で良いとは思うのですが、私の現時点での水準では、なぜこれがコンパイルできるのか、理解できないでいます。

おそらく、何らかの上位のラッパーを作って使えるようにしているのだろうとは思うのですが、少なくともこの記述内に限定してこの設計を理解しようとすれば、ここで使用するパッケージの指定が特に記述されていませんので、この例題は文法的に間違えていると言えるように思います。

逆に言えば、ＶＨＤＬ ソース記述を見ただけでは、その設計記述の全てを解釈することはできず、Ｃ 言語でのメイクファイルによるプログラム作成手順のような構築過程・要素がトランスクリプト履歴にしか残らない／勝手なパッケージを作る事の怖さ／が理解できるのではないかと思います。

（　とは言いながらこの例題は、エンティティ宣言部の use 文節で他のパッケージを呼んでいないから、これはどうしても文法違反の例題だと思っていたりしますが。）

## この事例を基として

単純な CASE 構文について少し調べておきましょう。

case 構文の ＬＲＭ（出典：ＶＨＤＬ / D.Perry 3.3.2 ）を見ますと

when 文で使われる choises::=choise{ | choise} は、

choise ::= SIMPLE_expression | discrete_range | ELEMENT_simple_name | OTHERS

との事で、この discrete_range はサブタイプ指示かレンジ文で表現されるとなっているのですが、基本的には expression で示されている被比較値に対する比較レンジという関係ですので、位置または値による大小関係が成立している必要がある筈です。

オリジナルの例題では、expression 部では結合されたビット・ベクター

```
case ah & al is
```

を使い、そのレンジ比較部では文字列

```
when x"f000" to x"ffff"     => q <= x"80";
```

を与えていますので、これらの関係が明示的に規定されていない以上、レオナルドやＶｓｙｓ が次のようなエラーメッセージを出すのは至極、自然な事のように思われます。

```
# ###### PREP9ORG.VHD(36): case ah & ah is
#ERROR:PREP9ORG.VHD(36): Array case expression must have a static subtype.
# ###### PREP9ORG.VHD(37): when x"f000" to x"ffff" => q <= x"80";
# ERROR: PREP9ORG.VHD(37): Range must be a scalar type.
```

（注）レオナルドは　(36) で警告をださない。

これは、現在の設定で、レオナルドの std/ieee が Ｖｓｙｓ の std/ieee と違うファイルを使用している為に生じる相違と思いますが、大きな障害にはなっていないのでここでは無視しておきます。　Vsystem では ｖｃｏｍするときにダイアログボックスのオプションに

use explicit declarations only

チェックボックスがあります。　ポーティングを意識した時には、なるべく明示的な記述方法をとった（チェックする）ほうが良いように思っています。

適切なパッケージを作って、このようなビットベクターに対し、文字列でレンジを指定できるような関係を構築する事ができれば良いのですが、今の私の知識では、ＶＨＤＬ 言語の基本仕様である「スカラータイプ」のデータとしての整数を考え、これをそのまま使う方法しか思い付かないので、この方針で変更してみる事にします。

prep_9a.vhd として提示しているのは、前の例を

```
LIBRARY ieee ;
use ieee.Std_logic_1164.all ;
use ieee.Std_logic_unsigned.all ;
```

だけを使って書き直した例です。

次ページに、注意したい部分を抜粋します。

```
INTAHAL <= conv_integer(To_StdlogicVector('0' & ah & al));


x0:PROCESS(rst,clk,INTAHAL,ah,al)
  Begin
  if rst='1' then   q <= x"00";
  elsif (clk='1' and clk'event) then
    case INTAHAL  is
      when 16#000f#  =>   q <=  X"01";
      when 16#0010# to 16#003f#   =>  q <=  X"10";
      when 16#0040# to 16#007f#   =>  q <=  X"20";
      when 16#0100# to 16#7fff#   =>  q <=  X"40";
      when 16#8100# to 16#f100#   =>  q <=  X"80";
      when others=> q <= X"00";
    end case;
  end if;
end PROCESS x0;
```

変更の目的は、レオナルドやＶｓｙｓでCASE構文内のレンジ指定をオリジナルのPREP9.VHDのようにきれいな形で記述する方法を知る事で、全く同じ記述方法ではありませんが、この例題はレオナルド／Ｖｓｙｓでのコンパイルが可能ですし、合成結果も一応目的を達成しています。

この構文を、アーキテクチャに対してローカルなプロシージャに書き直したものがprep_9b.vhdで、パッケージにするよりも害は少ないかも知れません。

また、このようなプロシージャを作成した場合、これをパッケージとして使う場合を想定した例題をprep_9c.vhdに提示しましたので参照して下さい。

ここでは、デコーダ部分だけをdecpack.vhdにまとめ直し、これを予めワーク内にコンパイルしておいてから、prep_9c.vhd をコンパイルする手順を想定しています。

## まとめ

元々、プロシージャ構文は設計の大きな部分を抜き出して利用する為に使われるもので、その呼び出し方はプロセス文の場合とは当然異なります。

今回の例は必ずしも、パッケージ化／再利用 の目的にふさわしいものではないと思いますが、ＶＨＤＬ記述の本体部分をすっきりと見せる（理解し易くする）為にもかなり有効な方法になっている事が判ります。

ＶＨＤＬ ８７／９３の具体的な相違については、ペリー著の ＶＨＤＬ ／メンター翻訳の巻末付録に解説されています。　　具体的な９３仕様の適用についてはこのシリーズで「ＶＨＤＬ ９」以後の例題で説明しますので参照していただければ幸いです。

コンパイル時にエラーが発生したときには、自分のソースを疑う事の他に
基本仕様として８７／９３どちらを選択したかによる相違
パッケージとして何を使ったか、による相違
コンパイラ作成者のインプリメンテーリョンによる相違
についてチェックしておくと、その問題の状況把握が早くできるようになるかも知れません。

また、レオナルド４．２２には、未だつまらないバグ（？）が残っていまして、ソース・テキスト・ファイルの ＥＯＦ が ＣＲＬＦ を指していないと、永久にソースファイルを読み続ける (- - ;) 事が現実に起こります。
これを見つけるには、レオナルドのメッセージを見て、- read での実行時間が異様に長い場合、**エディタでソースの最終行をリターンで終了するように変更して**セーブしてから再度コンパイルし、結果を比較すると良いでしょう。

Ｃ 言語などのように枯れたコンパイラしか生き残っていない状況からの類推で何かおかしな事が起きると常に自分のソースを疑ってかかる習慣がついていると思います。
しかし、ＶＨＤＬ の場合はまだその段階に達していない処理系が多いように思いますので、できれば、不具合を起こしたソースについて、レオナルドと Ｖｓｙｓｔｅｍ の両方でのコンパイル結果を比較して焦点を絞り込むようにお勧めしたいと思います。

prep9_a.vhd

```
LIBRARY ieee ;
use ieee.Std_logic_1164.all ;
use ieee.Std_logic_arith.all ;
use ieee.Std_logic_unsigned.all ;

-------------------------------------------------------------------------------
Entity prep9_a is
-------------------------------------------------------------------------------
      port (   clk,rst : in  Bit;
               al,ah   : in  Bit_Vector( 7 downto 0);
               qout    : out Bit_Vector( 7 downto 0)
           );
end prep9_a;

-------------------------------------------------------------------------------
Architecture a of prep9_a is
-------------------------------------------------------------------------------
    signal  q : Bit_Vector( 7 downto 0);
    signal  INTAHAL: natural range 0 to 2**16 -1;

Begin

    INTAHAL <= conv_integer(To_StdlogicVector(ah & al));
    -------------------------------------------------------------------------------
    x0:PROCESS(rst,clk,INTAHAL,ah,al)
    -------------------------------------------------------------------------------

    Begin
        if rst='1' then                         q <= x"00";
        elsif (clk='1' and clk'event) then
            case INTAHAL  is
                when 16#000f#               => q <= X"01";
                when 16#0010# to 16#003f#   => q <= X"10";
                when 16#0040# to 16#007f#   => q <= X"20";

                when 16#0100# to 16#7fff#   => q <= X"40";
                when 16#8100# to 16#f100#   => q <= X"80";
                when others                 => q <= X"00";
            end case;
        end if;
    end PROCESS x0;
    -------------------------------------------------------------------------------
    qout <= q;

end a;
-------------------------------------------------------------------------------
```

prep9_b.vhd

```
LIBRARY ieee ;
use ieee.std_logic_1164.all ;
use ieee.std_logic_arith.all ;
use ieee.std_logic_signed.all ;

-------------------------------------------------------------------------------
Entity prep9_b is
-------------------------------------------------------------------------------
      port (   clock,reset : in bit;
               low,high    : in  bit_Vector(7 downto 0);
               qout        : out bit_Vector(7 downto 0)    );
end prep9_b;

-------------------------------------------------------------------------------
Architecture b of prep9_b is
-------------------------------------------------------------------------------
signal AHALINT: natural  range 0 to 2**16 -1;
           signal  INTAHAL: natural range 0 to 2**16 -1;
    -----------------------------------------------------------------------------
    PROCEDURE decoder(  signal clk : in bit;
                        signal rst : in bit;
                        signal INTAHAL: natural range 0 to 2**16 -1;
                        signal q : out  bit_Vector(7 downto 0) ) is
    -----------------------------------------------------------------------------

    Begin
        if rst='1' then
            q <= x"00";
        elsif (clk='1' and clk'event) then
            case INTAHAL  is
                when 16#000f#             => q <= X"01";
                when 16#0010# to 16#003f# => q <= X"10";
                when 16#0040# to 16#007f# => q <= X"20";

                when 16#0100# to 16#7fff# => q <= X"40";
                when 16#8100# to 16#f100# => q <= X"80";
                when others               => q <= X"00";
            end case;

        end if;
    end decoder;
    -----------------------------------------------------------------------------


-------------------------------------------------------------------------------

Begin

    AHALINT <= conv_integer(To_StdlogicVector( '0' & high & low) );

    mydecoder: decoder(clock,reset,AHALINT,qout);

end b;
-------------------------------------------------------------------------------
```

decpack.vhd

```
1       LIBRARY ieee ;
        use ieee.std_logic_1164.all ;
        use ieee.std_logic_unsigned.all ;

5       ------------------------------------------------------------------------------
        Package MyDec is
        ------------------------------------------------------------------------------
           PROCEDURE Decoder(
                INTAHAL     : in natural range 0 to 2**16 -1;
10              signal q    : out Bit_Vector(7 downto 0) );
        end MyDec;

        ------------------------------------------------------------------------------
        Package BODY  MyDec is
15      ------------------------------------------------------------------------------
            ------------------------------------------------------------------------------
            PROCEDURE Decoder(
                INTAHAL     : in natural range 0 to 2**16 -1;
                signal q    : out Bit_Vector(7 downto 0) ) is
20          ------------------------------------------------------------------------------

            Begin
                case INTAHAL  is
                    when 16#000f#             => q <= X"01";
                    when 16#0010# to 16#003f# => q <= X"10";
25                  when 16#0040# to 16#007f# => q <= X"20";

                    when 16#0100# to 16#7fff# => q <= X"40";
                    when 16#8100# to 16#f100# => q <= X"80";
                    when others               => q <= X"00";
                end case;
30          end Decoder;

        end MyDec;
        ------------------------------------------------------------------------------
```

```
use work.MyDec.all;     -- use MyLogicProject.MyDec.all;

LIBRARY ieee ;
use ieee.std_logic_1164.all ;
use ieee.std_logic_unsigned.all ;

--------------------------------------------------------------------------------
Entity prep9_c is
--------------------------------------------------------------------------------
      port ( clock     : in  Bit;
             low,high  : in  Bit_Vector(7 downto 0);
             qout      : out Bit_Vector(7 downto 0) );
end prep9_c;

--------------------------------------------------------------------------------
Architecture c of prep9_c is
--------------------------------------------------------------------------------
    signal INTAHAL: natural range 0 to 2**16 -1;
    signal qintnl: Bit_Vector(7 downto 0);
Begin

    INTAHAL <= conv_integer(To_stdlogicVector(high & low) );

    x0:Process(clock,INTAHAL)
    Begin
        if(clock'event and clock='1') then
            Decoder(INTAHAL,qintnl); -- qintnl is DFF'd.
--          qout <= qintnl;          -- This will cause double bufferd output
        end if;
    end Process x0;

    qout <= qintnl;

end c;
--------------------------------------------------------------------------------
```