

この文書の目的

前回の例題でバックアノテートされた実デバイスのシミュレーションを簡単に説明しているのですが、今回の例題では、メモリというごくありふれたターゲットについて、そのシミュレーションを行う為の環境と操作とを、GENERIC を使いながら説明しようと思います。

モデル・テクノロジー社のウェブから単純なメモリモデルをダウンロードし、若干の修正を加えた上で、これを駆動するテストベンチを作成し、Vsystem のシミュレータ環境での操作と動作を解説しますが、あくまでこれはシミュレーション用のモデルである事(シンセシスも一応可能ですが)を十分に認識しておいてください。

ここでメモリを取り上げたのは、幾つかの入門書で類似したモデルをシグナルを使って解説したりしているのに対し、モデルテクノロジー社がそのホームページでメモリアレイのようなものはバリエーションで処理すべきであると力説している点に注目したからです。

この理由の詳細については、同社 WWW を見ていただくとして、シグナルというものは1ビットの扱いに数十バイトを必要とする事を、シミュレーターのユーザーに認識させたいと言う意図があるように見受けられます。

しかし、今回の例題のようにバリエーションを使って記述したモデルであっても、"adr_bits" や "data_bits" に大きな値を与えてleonardoでコンパイルすると、殆どハングアップに似た処理時間が必要になります。

FPGA をターゲットにした合成では一般に、メモリセルはレジスタ・DFF として実現されますので、例えばアルテラの10K シリーズでのEAB に対してメモリを作り込む場合は別のテクニックを必要とします。

Vsystem の使用方法を練習する時、以前から何か簡単なオール・イン・ワンの例題が1つでもあれば非常に楽にVHDL 全体をマスターできる筈だと思っていたのですが、メーカーが提供する例題は余りに初歩的過ぎて参考にもならないものか、あるいは、提供された形のままでシミュレーションの実行ができないような不十分な例題ばかりだったのではないかと思います。

前回と異なり、この例題ではパッケージ・ターゲット・テストベンチの全てを1つのファイルにまとめてしまっているのですが、これは本来のVHDL の使われ方からすると変則的な構成だろうと思います。

この意図は、このファイル1つで Vsystem のシミュレーション方法を一通りなぞることができるようにしておきたい(ファイル構成などは後から慣れれば済む)というところにあります。

特にテストベンチの記述で、そのエンティティにジェネリックを記述してある所あたりは、私の一人よがりかも知れませんが、他の説明書には書いていないように思われます。

実際にこれを使うことで、Vsystem シミュレータが基本機能として持っているジェネリックスの外部設定ダイアログボックスの利便がかなり高まるように思います。

説明主文のオリジナルはソースファイルに英文で表記してありますので、そちらも合わせて参照してください。

前提

Vsystem のインストールは予め終わっているものと想定します。基本ライブラリである STD や IEEE は、Vsystem のディレクトリ内にコンパイルされているものとし、プロジェクトファイル内で定義されるライブラリマッピングについては、すでにその取扱い方法を知っているものとします。

最初のステップ

予めフォルダを作成して今回の例題 MEMORY.VHD ファイルをコピーしておいてください。例えば D:\Foo が目的のディレクトリとします。

アイコンをダブルクリックして V s y s t e m を起動してください。

2 番目に

Menu/file/directry を使い、予め準備の終わっているフォルダー、D:\Foo に作業ディレクトリを変更します。このディレクトリには他にはなにも入っていない事が確認できます。

3 番目に

VSYSYSTEM.INI という名前でプロジェクトファイルを作成します。

Menu/project で、フルパス名 d:\foo\vsystem.ini とキー入力します。

この段階でフォルダ内に vsystem.ini という名前のファイルが作成されています。std や ieee のマッピングを確認しておいてください。

4 番目に

`Menu/library/new` からワークを作成します。

これはレオナルドで言うワークとは別の物で、実体は新しい1つのフォルダなのですが、新しく `Vsystem` でコンパイルされた結果はすべて(特別に指定しなければ)このワーク内に貯えられ、`info` ファイルに登録されます。

くどいようですが(正直いうと私自身がかつて混乱した事なので)この段階で、`Menu/library/Mapping` を実行したときに、少なくとも `std`、`ieee` ライブラリがここで確認できなければなりません。

できれば代表的な `FPGA` のメーカーのライブラリが下記の例のようにマッピングされているように処理してしまいましょう。

```
SIMPRIM  d:/xilinx/vhdl/src/simprims/work
ALT_VTL  c:/maxplus2/vhdl87/vital/v3_0/work
```

5 番目に

準備の確認が終わりました。

`Vsystem` の `VCOM` ボタンを押すと、ダイアログボックスが現れますのでその中に `"Memory.vhd"` というファイルがあることを確認し、`"COMPILE"` します。

トランスクリプト・ウインドウ内にメッセージが流れ、その最後でエンティティ名 `"Memory"` がロードされた事が示されますので、`DONE` として終了させます。

この段階でフォルダ内部にどのようなファイルが作成されているかを調べておくとおおいに参考になります。

6 番目は

シミュレーションの実行可能な段階になりました。

メニューから `VSIM` を選択すると新しいダイアログボックスが現れ、その中には今コンパイルした `VHD` ファイルのエンティティ名 `"Memory"` と、`"testbench"` とが表示され、シミュレーションが可能となっている事を示しています。`"testbench"` を選択すると、この中に含まれる全てのアーキテクチャ名が表示されます。

今は `"testmain"` がただ一つの選択になりますので、`"ADD"` して `"OK"` とします。

ようやく完了です

結果を見る為にキーボードから `"wave *"` 次に `"run -all"` としてみます。

Wave ウィンドウには、ここで作ったメモリのバス信号がテストベンチの信号に応答している状態表示されていますので、カーソルを追加して移動し、多くのタイミング情報が簡単に得られる事を確認してください。

これまでの手順で、シミュレーションの実行の為に GUI から信号の設定を一切しないで済んだ事にはもうお気づきでしょう。

ここで、VSIM を選択し、現れたダイアログボックスから VHDL を選択して “Specify Generic” で “name” に clk_{tic}、“value” に 2 (2 n s ではなく) を与えて再度シミュレート実行をさせると、トランスクリプトウィンドウに、この設計で仕掛けておいたタイミング違反の警告メッセージが流れるのが確認できます。

蛇足ながら

Vsystem では、ワーク内にコンパイルしたファイルは、明示的に削除しない限り保存されています。

削除すべきときに削除されなかったファイルが思わぬいたずらをする事がありますので十分な注意が必要 (これは私の事でありすが) だと思います。

以上、VHDL - 2 の説明とさせていただきます。

```

1  -----
--  VHDL-2
-----
--          Memory.VHD          1998/8/1 Systems Workshop Inc.
5  -----
--          An introduction to Vsystem/VHDL simulation
-----
--
-- NOTICE  This is a sample of a imaginal Memory model with timing delay.
10 --         Basic model is downloaded from www site of ModelTech.
--         Some modificatoins are made and Testbench added by our site for
--         first step lesson of Vsystem enviroment.
--
--         This is a sample to generate a simple simulation model. It is NOT
15 --         intended to be used for synthesis pourpose(thogh it works).
--
--         I predict giving some idea about a usage of such imaginal models
--         with some actual FPGA back-annotated models.
--         When you get familier with vsys operation, it will get much easy
20 --         and helpfull tool for you.
--
-- Caution; Please DO NOT compile this file in leonardo. It is not a
--         synthesis model. Especially, in leonardo, setting a large
--         values to "adr_bits" or "data_bits", it will cause a hang-
25 --         up-like compilation time.
--         In some VHDL-books, similar but "signal" based memory models
--         are described and explained precisely, however, ModelTech do
--         recommend to use "variable" insted of "signal" for memory
--         array definitions; even if it is limited for simulation only
30 --         models.
--
-- Assumption:  You have installed a vsystem properly. And have compiled
--         basic libraries into each work directry. You know how to
--         "map" your libraries into your "projectfile".
35 --
-- The 1st step is to invoke Vsystem in your system by double clicking the
--         Vsystem icon.
--
-- 2nd  Please change your directry by Menu/file/directry to a prepared
40 --         space where you have copied this Memory.vhd, say d:\foo.
--         Your current directry is "foo" now. You can confirm that there's
--         no other files beside this Memory.vhd in this folder.
--
-- 3rd  Let's build a new projectfile here, say "vsystem.ini".
45 --
--         From Menu/project, please type in a full path name of it.
--         Let's assume it  d:\foo\vsystem.ini.
--         If you look into this folder with winfile.exe or something, now you
--         will find a new file named vsystem.ini in it.
50 --
-- 4th  Next,from Menu/library/new, you make a new "work" to put new things.
--
--         At this instance, it will be better to confirm that you have "Mapped"
55 --         some standard libraries in your project already.
--
--         From Menu/library/Mapping, you must see names of "std" and "ieee" at
--         least.
--         ( Hopefully,      SIMPRIM at d:/xilinx/vhdl/src/simprims/work, and
--                       ALT_VTL at c:/maxplus2/vhdl87/vital/v3_0/work  also.)
60 --         If it is not yet, please consult with your vsystem installation directry
--         and/or to your original vsystem.ini, or dial to your Vsystem dealer.
--
--
-- 5th,  You are now ready to go.  When you push a VCOM button, a dialog box
65 --         appears showing a filename "Memory.vhd" in it.
--         Just "COMPILE" it.
--         In a Transcript window, many messages flow, and at its end it will tell
--         you that a entity "Memory" is loaded.
--         Push "DONE" button.

```

```

70  --      It will be a good idea to look into your current folder now.
--
-- 6th, Here we come to a simulation phase.
--      From Menu, select Vsim.
--      Then you will have a new dialog box showing names of your design entity
75  --      "Memory"and "testbench" are now ready for simulation.
--
--      Selecting "testbench", the dialog box shows a list of available names
--      of architectures which are binded to this "testbench" entity.
--      In this case, a "testmain" will be the only one to go.
80  --      Select it with "ADD" and "OK" button.
--
-- At Last,it is completed (^)/.
--      Let's see what happend in your design.
--      From keyboard, type "wave *" then "run -all".
85  --      Look into Wave window,tdro bus signals are responding to your testbench.
--
--      With a back-annotated vhdl files, you are capable of testing real FPGA's
--      ,involving such imaginal external devices,in this easiness (^)/
--      ( Although it really is a tough-enough work (--i) )
90  --
--      Adding cursor,you get much timing infomations from Vsim. A Wave window
--      thus display the results.
--      Please remind that with this testbench, you didn't forced any signals no
r
--      clocks from Vsim GUI interface.
95  --
--
-- Watch:  Modifying a Generic parameter "clocktic" from default 5 to a value
--          of 2, you will see warning messages appear in transcript window.
--          Now you know why and how to control your model parameters.
100 --
-- Final notice
--      It is important to notice that your "work", which is assigned by current
--
--      project file "vsystem.ini", keeps previous contents untill you delete it
--      explicitly.
105 --      Anytime you are to begin your session, you have to start from 2nd step
--      described in this manual.
--
-- Next coming is
--
110 --      File based TestVector loading and result logging example.
--
-- start here
-----
115 -- PACKAGE EXAMPLE      When you use a "natural", this pakage will be a help.
-----

use std.standard.all;

library ieee;
120 use ieee.std_logic_1164.all;

Package conversions is
    Function sulv_to_natural(x :Std_Logic_vector)return natural;
    Function natural_to_sulv(n,bits :natural)return Std_Logic_vector;
125 End conversions;

Package body conversions is
Function sulv_to_natural(x :Std_Logic_vector)return natural is
    variable n :natural :=0;
130     variable failure :boolean :=false;
    Begin
        Assert (x'high -x'low +1)<=31
            report "Range of sulv_to_natural argument exceeds natural range"
            severity error;
135     for i in x'range loop

```

```

    n :=n *2;
    case x(i)is
        when '1'|'H'=>n :=n +1;
        when '0'|'L'=>>null;
140         when others =>failure :=true;
    End case;
End loop;

    Assert not failure
145     report "sulv_to_natural cannot convert indefinite Std_Logic_vector"
        severity error;
    if failure then
        return 0;
    else
150         return n;
    End if;
End sulv_to_natural;

Function natural_to_sulv(n,bits :natural)return Std_Logic_vector is
155     variable x :Std_Logic_vector(bits -1 downto 0):=(others =>'0');
    variable tempn :natural :=n;
    Begin
        for i in x'reverse_range loop
            if (tempn mod 2)=1 then
160                 x(i):='1';
            End if;
            tempn :=tempn / 2;
        End loop;
        return x;
165     End natural_to_sulv;

End conversions;
-----
170
-----
-- TARGET DESIGN (An IMAGINAL Memory MODEL)
-----
175     use std.standard.all;

    library ieee;
    use ieee.std_logic_1164.all;
    use work.conversions.all;

180     entity Memory is
        generic( INSTANCE_ID: string ;
                adr_bits   : integer;
                data_bits  : integer
185                );    -- A ";" is needed here.  This is not a GENERIC_MAP.

        port(add_in       :in  Std_Logic_vector(adr_bits-1  downto 0);
              data_in     :in  Std_Logic_vector(data_bits-1 downto 0);
              data_out    :out Std_Logic_vector(data_bits-1 downto 0);
              cs          :in  Std_Logic;
190              mwrite    :in  Std_Logic
                );

        subtype word is Std_Logic_vector(data_bits-1 downto 0);
        constant nwords :integer :=2 **adr_bits;
195         type ram_type is array(0 to nwords-1)of word;

    End;

-----
200     architecture style_87 of Memory is
-----
        BEGIN
-----
        Memory:Process (cs)

```

```

205 -----
-----
variable ram :ram_type;    -- "Why variable?" explained @www MODELTECH
-----
----- -- Recall this is a simulation model.

210 variable address :natural;
variable cs_rise : TIME :=0 ns;
variable cs_fall : TIME :=0 ns;
variable cs_width: TIME :=0 ns;

215 Begin
    if rising_edge(cs) then
        cs_rise := NOW;
        address :=suly_to_natural(add_in);
        if (mwrite ='1') then
220             ram(address):=data_in;
        End if;
        if(CS='1') then data_out <= ram(address) after 3 ns;
        else             data_out <= "UUUUUUUU";
        end if;
225 End if;

        if falling_edge(cs) then
            cs_fall := NOW;
            data_out <= "UUUUUUUU" after 2 ns;
230 end if;

        cs_width :=(cs_fall - cs_rise);

        Assert not (CS='0' and (cs_rise /= 0 ns) and (cs_width < 8 nS ))
235 -- "(cs_width < 8 nS)" mightl be replaced to a device specific constant
        .
        report "cs_pulse width " & "@" & INSTANCE_ID & " :msg from Memory.vh
d"
        severity WARNING;
        --
        -- Actually,Vsim always put msg with a corresponding indtance name. St
ill,
240 -- this could be a good help of understanding how GENERIC parameters are
        -- transfferd to each instances.
        --
        End Process Memory;

245 End style_87;
-----

250 -- TEST BENCH EXAMPLE
-----
Entity TestBench is
    generic(clocktic : time:=5 ns);
    -- You can over-ride this generics value via Menu/Vsim/VHDL/SpecifyGeneric
255 -- dialogbox. (ex) Name:clocktic value:2 (not 2 ns)
    --
    -- Unit "ns" or "ps" depends on your setting in Simulator resolution dialog.
    --
end;

260 library IEEE;
use IEEE.std_logic_1164.all;
use ieee.std_logic_ARITH.all ;
use ieee.std_logic_signed.all ;

265 -----
Architecture TestMain of TestBench is
-----

```



```

Component Memory      -- This is the TARGET design deccration in testbench.
270   generic( INSTANCE_ID : string;
          adr_bits   : integer ;
          data_bits  : integer );
          port(add_in   :in  Std_Logic_vector(adr_bits-1 downto 0);
              data_in   :in  Std_Logic_vector(data_bits-1 downto 0);
275   data_out  :out Std_Logic_vector(data_bits-1 downto 0);
          cs       :in  Std_Logic;
          mwrite   :in  Std_Logic
          );
end Component;

280
Type TestRecordT is Record  -- TestVector format definition.
  adr : Bit_vector(7 downto 0);
  din : Bit_vector(7 downto 0);
  mw  : Bit;
285  EOVS : Bit;
End Record;

Type TestArrayT is array(natural range <>) of TestRecordT;
290 -- If you use "positive" instead of "natural" here, a INDEX value
-- should start from 1 not from 0.
  constant TestPattern : TestArrayT := (
    (adr=>X"00",din=>X"01",mw =>'1',EOV=>'0'),
    (adr=>X"01",din=>X"02",mw =>'1',EOV=>'0'),
295   (adr=>X"02",din=>X"04",mw =>'1',EOV=>'0'),
    (adr=>X"03",din=>X"08",mw =>'1',EOV=>'0'),
    (adr=>X"04",din=>X"10",mw =>'1',EOV=>'0'),
    (adr=>X"05",din=>X"20",mw =>'1',EOV=>'0'),
    (adr=>X"06",din=>X"40",mw =>'1',EOV=>'0'),
300   (adr=>X"07",din=>X"80",mw =>'1',EOV=>'0'),

    (adr=>X"00",din=>X"00",mw =>'0',EOV=>'0'),
    (adr=>X"01",din=>X"00",mw =>'0',EOV=>'0'),
    (adr=>X"02",din=>X"00",mw =>'0',EOV=>'0'),
305   (adr=>X"03",din=>X"00",mw =>'0',EOV=>'0'),
    (adr=>X"04",din=>X"00",mw =>'0',EOV=>'0'),
    (adr=>X"05",din=>X"00",mw =>'0',EOV=>'0'),
    (adr=>X"06",din=>X"00",mw =>'0',EOV=>'0'),
    (adr=>X"07",din=>X"00",mw =>'0',EOV=>'0'),
310   (adr=>X"08",din=>X"80",mw =>'0',EOV=>'1')
  );

  -- Internal signals for Test-Bench operations -----
315  signal INDX      : natural :=0;
  signal CLK        : Std_Logic;
  signal CLKCS     : Std_Logic :='0';  -- If not initialized,see what happens.
  signal tmwrite   : Std_Logic ;
  signal tadr_in   : Std_Logic_Vector(7 downto 0);
320  signal tdata_in : Std_Logic_Vector(7 downto 0);
  signal tdro     : Std_Logic_Vector(7 downto 0);

  signal  CLKCOUNT : INTEGER:= 0; -- CLOCK counter to increment INDEX value.
  constant MAXCOUNT : INTEGER:= 3; -- for Patern-Gen repeat count value
325  -----

Begin  -- Test Bench MAIN.  Connect target entity with test vector signals --
-----

330  -----

  CLKProc : Process      -- Clock pulse generator
  -----
  Begin  CLK<='0';  wait for clocktic ; -- Given via GENERIC default is 5 ns

```

```

335         CLK<='1'; wait for clocktic ;
End Process CLKProc;
-----

ClkCountProc : process(CLKCOUNT,CLK)    -- @CLKCOUNT=0 increment(INDX)
340 -----
Begin
    if(CLK'EVENT and CLK='1') then
        if(CLKCOUNT=MAXCOUNT) then CLKCOUNT<=0;
345         else CLKCOUNT<=CLKCOUNT+1;
        end if;
        if(CLKCOUNT=0) then CLKCS<='1'; -- here generate CLKCS signal.
        else CLKCS<='0';
        end if;
    end if;
350 End Process ClkCountProc;
-----

RdVector: process(CLKCOUNT)    -- Read Test-Vectors into test signals.
355 -----
variable Vector : TestRecordT;
variable tEOV   : Std_Logic;
Begin
360     if(CLKCOUNT=0) then
        Vector := TestPattern(INDX);
        tadr_in <= to_stdlogicvector(Vector.adr);
        tdata_in<= to_stdlogicvector(Vector.din);
365         tmwrite <= to_stdulogic (Vector.mw);
        tEOV := to_stdulogic (Vector.EOV);
        INDX <= INDX+1;
        Assert not tEOV='1' report "End of Vector table."
        SEVERITY Failure ;
        -- You can trap this with Vsim/Menu/option/simulation_option/assert
370         -- "break_on_assestion" setting.
    end if;
End Process RdVector;
-----

375 -----

-- Unit Under Test : Instanciate the TARGET and connect it with test signals
-----

UUT: Memory -- Remind this label UUT is called as "REGION" in sdf dialog-box.
380 -----
Generic Map(
    INSTANCE_ID =>"MyModel_1",
    adr_bits => 8 ,
    data_bits => 8
) -- Here, You should not put any ";". This is a GENERIC=MAP.
385 PORT Map(
    add_in => tadr_in, -- Connect test vector signals to the target.
    data_in => tdata_in,
    data_out=> tdro,
    cs => CLKCS,
390    mwrite => tmwrite
);
-----

END;
```