

この文書の目的

前回の VHDL-3 ではメモリーを「実デバイスへの実装」する時の注意点についての説明をしてみました。今回は、再び Vsys に戻って TextIO についての説明をしてみたいと思います。

アルテラなどのシミュレータでも、ベクターを利用することは当然可能なのですが、以前にも説明したとおり、FPGA の外部に設置した遅延を含む要素を取り込んだ場合のシミュレーション (maxplus2 では、全ての要素をアルテラ社のデバイスとして作成すれば可能です) に対しては基本的に対応がとれません。

今回の例題では、Vsys に外部ベクターファイルを読みませ、シミュレーションを実行してその結果をテキスト出力ファイルに吐き出すまでを説明します。

シミュレーションの過程でクロック信号にドリフトを加えていますが、Vsys での取扱いでは、このような処理が比較的自由に行える事が特徴といえそうです。

準備です

前回同様、作業用ディレクトリを作成します。

今回は `d:\uoo` として、添付してある `mem_tb.vhd` と `mem_tb.vec` との2つのファイルをコピーしておいて下さい。

Vsys を起動したらこのディレクトリ `d:\uoo` に移動して、プロジェクト・ファイル `vsystem.ini` を作成してから、前の例題のように作業ディレクトリ `work` を設定します。(アイコンのショートカットのプロパティを編集して、作業フォルダに `d:\uoo` を指定しておく、何回も起動してテストを繰り返すときには少しだけ楽ができます) 参考までに、私の場合の `vsystem.ini` 内容の一部を下に示します。

```
[library]
std = c:/vsystem/std
IEEE = c:/vsystem/ieee
SIMPRIM = D:/xilinx/vhdl/src/simprims/work
ALT_VTL = C:/MAXPLUS2/VHDL87/VITAL/V3_0/WORK
[vcom]
Keep Dialog Open=1
Show_source = true
Explicit = false
VHDL93 = false
以下略
```

Vsim での取り扱いについて

殆ど何も新しい説明はありませんが、作業ディレクトリ名がもし正確に `d:\uoo` となっていない場合は、ソースに変更を加える（読み込みベクター・ファイルをフルパス名で指定している為です）必要があります。このような場合には、ソースファイルの 130,132 行目を、適切なディレクトリ名に変更して下さい。

いつものようにサンプルファイル `mem_tb.vhd` を VCOM します。

次に、VSIM ボタンを押すと、今回の例題ではデザインファイルが 2 つ、エンティティとしてコンパイルされている事が判ります。この内容は被試験物の ram と、その試験環境 testbench の 2 つですので、テストベンチの方を選んでアーキテクチャ testmain を指定し add して ok して下さい。

トランスクリプト画面で（別に表示されていなくても構わないのですが）

```
view wave
```

```
wave *
```

```
としてから
```

```
run -all
```

で、タイミングのドリフトを持ったクロックによってメモリがテストされている事が波形表示されます。このドリフト操作はソースファイルの行 165 で処理された結果です。

vsim する時に、前の例題のように "add" してからジェネリックを vhdl メニューで変更していないのに、シミュレーション中に警告がでていますが、これはソースファイルの 73 行目で発行されたメッセージです。

Vsys を終了し、ファイルのロックを外してから、`d:\uoo` ディレクトリの内容を見ると、新しい `mem_tb.out` という名前のファイルが作成されている事が判ります。これは、`mem_tb.vhd` ソースファイルの行 183..190 出力文で作成されたテキストファイルですので、適切なエディタで開いてみてください。

まとめ

市販の書籍の参考例題は、必ずしもそのままでは vsim できない場合がありますが、特に textio を使用した場合に多いのではないかと思いますので、今回の例題でどのように引数を渡しているかを見て参考にしてください。

```

1  -----
-- VHDL-4   TextIO step for Vsystem           1998/8/5 Systems Workshop Inc.
-- -----
-- Please be noticed that, if you have installed Vsystem/VHDL into your PC,
5  -- you will find that several examples described in Tutorial books doesn't work
-- at all.   It is especially true when you have experimented with textio
-- applications.
--
-- Here described is a simple example for Vsystem applications. After you have
10 -- got well experienced, it will become much easier to overload those library
-- dependent in-compatibleness.
--
-- Please find a TEXTIO.VHD in /Vsystem/vhdl_src/std/ directory and a package
-- STD_LOGIC_TEXTIO in textio.vhd at your /Vsystem/vhdl_src/synopsys/ also.
15 --
-- -----
-- TARGET DESIGN
-- -----

20 use std.standard.all;

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

25 Entity Ram is
    Generic( INSTANCE_ID: string ;
             adr_bits   : integer;
             data_bits  : integer );
    port(Adr_in      :in Std_Logic_vector( adr_bits-1 downto 0);
30     DIN          :in Std_Logic_vector(data_bits-1 downto 0);
        DOUT       :out Std_Logic_vector(data_bits-1 downto 0);
        CS         :in Std_Logic;
        nWRITE     :in Std_Logic
        );

35     constant nwords :integer := 2 **adr_bits;

        subtype word is Std_Logic_vector(data_bits-1 downto 0);
        type ram_type is array(0 to nwords-1)of word;
40 End;

-- -----
Architecture RTL of Ram is
-- -----

45 BEGIN

-- -----
RamUnit:Process (CS)
-- -----

50     variable RamArray : ram_type;

        variable address : integer;
        variable cs_rise,cs_fall, cs_width: TIME :=0 ns;

        Begin
55         if rising_edge(CS) then
            cs_rise := NOW;
            address := conv_integer (Adr_in);
            if (nWRITE = '0' and CS='1') then
                RamArray(address):=DIN;
60             end if;
            if(nWRITE='1' and CS='1') then
                DOUT <= RamArray(address) after 3 ns;
            else
                DOUT <= "ZZZZZZZZ";
            end if;
65         End if;

            if falling_edge(CS) then
                cs_fall := NOW;
            end if;

```

```

70         cs_width :=(cs_fall - cs_rise);

        Assert not (CS='0' and (cs_rise /= 0 ns) and (cs_width > 48 ns ))
            report "cs_pulse width " & "@" & INSTANCE_ID
75             severity WARNING;
end Process RamUnit;

End RTL;
-----

80
-----
-- TEST BENCH EX2
-----

85 use std.textio.all;

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

-----

Entity TestBench is      Generic(clocktic : time:=5 ns);
95 end;
-----

-----
Architecture TestMain of TestBench is
-----
100 Component Ram      -- This is the TARGET design deccaration in testbench.
    Generic( INSTANCE_ID : string;
            adr_bits      : integer ;
            data_bits     : integer );
105     Port(Adr_in :in Std_Logic_vector( adr_bits-1 downto 0);
            DIN   :in Std_Logic_vector(data_bits-1 downto 0);
            DOUT  :out Std_Logic_vector(data_bits-1 downto 0);
            CS    :in Std_Logic;
            nWRITE :in Std_Logic
110             );
end Component;

-- Internal signals for Test-Bench operations -----

115 signal CLK      : Std_Logic;
signal CLKCS      : Std_Logic :='0'; -- If not initialized,see what happens.
signal tnWRITE    : Std_Logic ;
signal tAdr_In    : Std_Logic_Vector(7 downto 0);
signal tDIN       : Std_Logic_Vector(7 downto 0);
120 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

Type VectRecord is Record -- TestVector format definition.
125     Adr : Std_Logic_vector(7 downto 0); -- For Z/U char in text file
     Din : Std_Logic_vector(7 downto 0);
     nW  : Std_Logic;
End Record;

130 File TestPatFile : TEXT is in "d:\uoo\Mem_Tb.vec"; -- Drive vector.
File TestDroFile : TEXT is out "d:\uoo\Mem_Tb.out"; -- Logging output.

-----

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

```

UUT: Ram

```

-----
140   Generic Map(
        INSTANCE_ID => "MyRamModel-2",
        adr_bits   => 8 ,
        data_bits  => 8
    )
145   PORT Map(
        Adr_in  => tAdr_In,  -- Connect test vector signals to the target.
        DIN     => tDIN,
        DOUT    => tDRO,
        CS      => CLKCS,
150        nWRITE => tnWRITE
    );

-----

155   Process
-----
        variable CLI,CLO : Line ;           -- TextIO Line.
        variable CV : VectRecord;         -- Current Vector.
        variable Drift,Clocktime : Time;
-----
160   Begin
        Clocktime := Clockctic;
        Drift := 1 ns;
165   Tbloop:loop
        Clocktime:=Clocktime+Drift;

        CLK<='0';  wait for Clocktime ;           -- When you used a "wait for"

        CLK<='1';  wait for Clocktime; -- you can't use "CLK'EVENT" any more.

170   if(CLKCOUNT=MAXCOUNT) then
        CLKCS  <='1';
        CLKCOUNT<= 0;
        ReadLine( TestPatFile, CLI );
        Read( CLI, CV.Adr );           -- Direct connection from file to "signa
1"
        Read( CLI, CV.Din );           -- is not allowed. (must be a variable)
175   Read( CLI, CV.nW );
        tAdr_In <= CV.Adr;
        tDIN    <= CV.Din;
        tnWRITE <= CV.nW;

180   else
        CLKCS<='0';
        CLKCOUNT<=CLKCOUNT+1;
        write(CLO,NOW,RIGHT,8,ns);
        write(CLO,'>',RIGHT,1);
185   write(CLO,to_bitvector(tAdr_In),RIGHT,10);
        write(CLO,tDIN,RIGHT,10);
        write(CLO,to_bit(tnWRITE),RIGHT,4);
        write(CLO,':',RIGHT,1);
        write(CLO,tDRO,RIGHT,10);
190   writeline( TestDroFile,CLO );  -- Here you have wrote a new line.

        end if;

        exit when ENDFILE(TestPatFile );
195   end loop Tbloop;

        assert False
            report "File End "
200            severity Error;

        end process;
-----

```

205 **END;**

d:\uoo\mem_tb.vec

1	00000000	00000001	0
	00000001	00000010	0
	00000010	00000100	0
	00000011	00001000	0
5	00000100	00010001	0
	00000101	00100011	0
	00000110	01000111	0
	00000111	10001111	0
	00000000	ZZZZZZZZ	1
10	00000001	ZZZZZZZZ	1
	00000010	ZZZZZZZZ	1
	00000011	ZZZZZZZZ	1
	00000100	ZZZZZZZZ	1
	00000101	ZZZZZZZZ	1
	00000110	ZZZZZZZZ	1
15	00000111	ZZZZZZZZ	1

d:\uoo\mem_tb.out

1	12	ns>	00000000	UUUUUUUU	0:	UUUUUUUU
	26	ns>	00000000	UUUUUUUU	0:	UUUUUUUU
	42	ns>	00000000	UUUUUUUU	0:	UUUUUUUU
	80	ns>	00000000	00000001	0:	ZZZZZZZZ
5	102	ns>	00000000	00000001	0:	ZZZZZZZZ
	126	ns>	00000000	00000001	0:	ZZZZZZZZ
	180	ns>	00000001	00000010	0:	ZZZZZZZZ
	210	ns>	00000001	00000010	0:	ZZZZZZZZ
10	242	ns>	00000001	00000010	0:	ZZZZZZZZ
	312	ns>	00000010	00000100	0:	ZZZZZZZZ
	350	ns>	00000010	00000100	0:	ZZZZZZZZ
	390	ns>	00000010	00000100	0:	ZZZZZZZZ
15	476	ns>	00000011	00001000	0:	ZZZZZZZZ
	522	ns>	00000011	00001000	0:	ZZZZZZZZ
	570	ns>	00000011	00001000	0:	ZZZZZZZZ
	672	ns>	00000100	00010001	0:	ZZZZZZZZ