ＶＨＤＬでのＦＰＧＡ開発について

VHDL 論理合成の環境で最初に注意すべき点は

シミュレーション・タイミング検証の為の HDL 記述と論理合成の為の HDL 記述とを明確に区別して意識することが必要であること、

ライブラリーには、予めテクノロジー・ライブラリとして提供されるデバイス・ファミリーに固有の構成要素（たとえばグローバル・クロック・ドライバーなど）があり、一方ではアプリケーション側の ＲＴＬ 記述によって生成される構成要素（たとえばパッケージ化された　カウンタなど）があり、共にライブラリと呼ばれ、ワーク・ディレクトリにコンパイルしてから使用されるので混同し易いことであるように思います。

また、シミュレータとして Ｖｓｙｓｔｅｍを使った場合のライブラリ／ワークの概念もはっきりと意識して使う必要があります。

小さなデバイスに多くを要求する場合

また、ＡＣＴＥＬ － ＡＣＴ ３ファミリーなど配線資源に種々の制約があるようなデバイスでは、タイミング特性をチューンする為に意図的に階層構造をフラットにして最適化を図った場合に生じる ＬＶＴ 資源の不足問題があります。

これは、ターゲット側のフィッタ／コンパイラが、資源としてセル・ピンの利用率などを認識してフィット可能と判断し、長時間の配置配線の実施に入った後でギブアップする状況になりますが、事前の予測はかなり困難と言えます。

階層構造を保持して LVT 資源の利用を制限する事により、ルーターのギブアップを回避する事が出来る場合がありますが、このような場面に直面した場合の基本的な解決手法としては、設計サイズを小さく抑える工夫に尽きるものと思われます。

ただし、これらの状況は、ターゲットデバイスのフィッターの改版や論理合成プログラムの改版によって大きくその様相が変化します。

旧版での対応策が新版では逆にパーフォーマンスを低下させる場合もありますので十分な注意が必要です。

これまでの基本的な設計方針として

子葉末節の部分について言えば、今回の私共の設計方針では、使用するライブラリを極めて限定し、型変換についてのパッケージなどについても極力、ieee 標準ライブラリを直接参照するようにしています。

これらの型変換関数などは自分独自のパッケージを作成し、その中で新しい関数を定義して使えばよいのですが、このような手法は緊密な連絡を取り合っているプロジェクトの中で有効とされているもので、疎な結合連絡形態で作業を進める場合には、今回のような方法で混乱を避けるほうが良いように思っております。

双方向性のバス信号については、初期のレオナルドで発生した不具合経験から、複数の内部バスを階層最上位のマルチプレクサに接続した後に std_logic_vector に変換してから外部バスへトライステート形式で接続するように統一してあります。

ＶＨＤＬ の処理系によっては（例えばアルテラ社）３２ビットの取扱いを行わないものがありますが、このような場合には一旦レオナルドで ＥＤＩＦ に落としてインポートすれば済む内容なので、わざわざパッケージを作ってオーバーライドするような事はしておりません。

使用開始初期のレオナルド 4.0x では、バスのメンバーの取扱いに問題があり、バス・メンバーの並び方向指定に downto 構文を使用しても、ターゲット側に正しく伝達解釈されず、ビットの並びが逆転してしまうなどの問題がありましたが、現在の版 4.2x ではこれらの問題はひととおり解決されています。

同様に、ＡＣＴＥＬ ターゲットでの負荷バランスの問題も現在のレオナルドでは解決されていますし、ＣＬＫＩＮＴ マクロの多重生成の障害も発生しなくなっています

ザイリンスクの場合についても同様にレオナルドの改版とザイリンクスのフィッタの能力向上により大幅な能力の向上がありました。

xact6.0.2 版（xnf ネットを取り込んでいた）の時に入り切らなかった規模の設計が、m1.4 版（edif 形式ネットの取り込みができる）では殆ど苦労せずに同一のデバイスにフィットさせることができるようになっています。

ただし、ファウンデーションの m1.4 版は基本的に回路図入力を前提にしており、ネットリストを全て ＥＤＩＦ 形式で吐き出すレオナルドのようなサードパーティ製品からのエントリーではプロジェクトファイルの作成や、ファイル名ルールに少し注意が必要です。（おそらくアライアンスでの m1.4 には不必要な事だろうという事ですが私共では検証できません）

この資料の構成

添付した資料は、これからレオナルド／Ｖｓｙｓｔｅｍを使って実際にシミュレーション／フィティングを行う為に必要な作業の流れと、余りマニュアルに書いていない ＴＩＰＳ を説明するもので、具体例としてアルテラの事例から入り、実例を示した上でザイリンスクでの事例を説明するようにしました。

設計の具体例として実例を利用するのには、余りにも大きな時間と労力が必要とされますので、

ＴＯＰ．ＶＨＤ　　という簡単なターゲットの例
ＴＢ．ＶＨＤ　　という、この ＴＯＰ 用のテストベンチ

を作成し、比較的詳しい解説資料を作成した上で、これを leonardo でコンパイルし、アルテラの maxplusII でフィットするまでの実例ファイルと、フロー解説をＡ ３の解説図に纏めてみました。

次に、同じ設計をザイリンクスに適用した場合のフロー＆ ＴＩＰＳ を Ａ ３の解説図に作成し、実例ファイルを作成してあります。

特に、Ｖｓｙｓｔｅｍ環境は ＣＲＴ 上での解析を目的としたもので、詳細な波形情報や、カーソルで検査したタイミングデータなどの設計資料的な内容の保存はできませんので（WAVE.PS ファイルは作成できますがアウトラインしか判りません）おそらく、このシステムの設計意図にある利用法は、設計者が ＣＲＴ 上で判断した事を証拠に残す必要はなく、そのような場合は TEXTIO を利用するものと想定しているのではないかと思われます。

この資料がお役に立つ事を願っております。

altera-top.vhd

```
1       ---------------------------------------------------------------------
        -- N-bit COUNTER COMPARATOR   for testbench lessons.
        ---------------------------------------------------------------------
        library IEEE;
5       use IEEE.std_logic_1164.all;
        use ieee.std_logic_ARITH.all ;
        use ieee.std_logic_signed.all ;


10      Entity TOP is
            port (DBUS  : inout Std_Logic_Vector(7 downto 0);
                  CLK   : in Std_Logic;
                  nRD   : in Std_Logic;
                  nWT   : in Std_Logic;
15                nCLR  : in Std_Logic;
                  CENBL : in Std_Logic;
                  RSEL  : in Std_Logic;
                  CVAL  : out Std_Logic_Vector(7 downto 1);
                  EQU   : out Std_Logic );
20      end TOP;

        Architecture RTL of  TOP is
          signal    DRO     : Std_Logic_Vector(7 downto 0);
          signal    CompReg : Std_Logic_Vector(7 downto 0);
25        signal    CntrReg : Std_Logic_Vector(7 downto 0);
          signal    Counter : Integer range 0 to 255;  --Integer'high;

        begin
            GEN1:for I in 0 to 7 GENERATE
30              DBUS(I)<=DRO(I) when nRD='0' else 'Z';
            end GENERATE GEN1;

            CntrReg<= Conv_Std_Logic_Vector(Counter,8);

35          DRO<= CompReg   when RSEL='0' else CntrReg;

            ---------------------------------------------------------------------------
            RegLdProc: Process(CLK,DBUS,nWT,CompReg)
            ---------------------------------------------------------------------------
40          begin
                if(CLK'EVENT and CLK='1') then
                    if(nWT='0') then    CompReg<= DBUS;
                    else                CompReg<= CompReg;
                    end if;
45              end if;
            end Process RegLdProc;
            ---------------------------------------------------------------------------


50          ---------------------------------------------------------------------------
            CountProc: Process(CLK,CENBL,nCLR,nWT,Counter,CompReg)
            ---------------------------------------------------------------------------
            begin
                if(nCLR='0' or nWT='0') then
55                              Counter<=0;
                elsif(CLK'EVENT and CLK='1') then
                    if(CENBL='1') then  Counter<= Counter+1;
                    else                Counter<= Counter;
                    end if;
60              end if;
                CVAL<= Conv_Std_Logic_Vector(Counter,7);
            end Process CountProc;

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

65          EQU<='1' when (Counter= Conv_Integer(Compreg))
                    else '0';
        end  RTL;
```

こういうパラメータはジェネリックとして与えるほうが良いかも知れません。

こういう記述は安易に使わないようにしたほうが無難でしょう。（きちんとした体系を作って設計するなら別）実際にこれを試してみると分かりますが深い階層下で起きると厄介なバグになりかねません（--;）

意図して遅延のある信号を取り出しています

altera tb.vhd

被テスト設計をコンポーネント宣言し
（実体は別にコンパイルされている）

```
Entity TestBench is
end;

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

Architecture TestMain of TestBench is
    component TOP   -- This is the TARGET design ----
    port (DBUS      : inout Std_Logic_Vector(7 downto 0);
          CLK       : in Std_Logic;
          nRD       : in Std_Logic;
          nWT       : in Std_Logic;
          nCLR      : in Std_Logic;
          CENBL     : in Std_Logic;
          RSEL      : in Std_Logic;
          CVAL      : out Std_Logic_Vector(6 downto 0);
          EQU       : out Std_Logic
          );
    end component;

    Type TestRecordT is record  -- TestVector format definition.
        nRD,nWt : Std_Logic ;
        CE,RSEL : Std_Logic ;
        DBUS    : Std_Logic_Vector(7 downto 0) ;
        EOV     : Std_Logic ;
    End record;
```

印加したいテストベクタ値を設定する

```
    Type TestArrayT is array(positive range <>) of TestRecordT;

    constant TestPattern : TestArrayT := (
    (nRD=>’1’,nWT=>’1’,CE=>’0’,RSEL=>’0’,DBUS => "00000000",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’0’,RSEL=>’0’,DBUS => "00000000",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’0’,RSEL=>’0’,DBUS => "00000000",EOV=>’0’),
    (nRD=>’1’,nWT=>’0’,CE=>’1’,RSEL=>’0’,DBUS => "00000101",EOV=>’0’),
    (nRD=>’0’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "00000000",EOV=>’0’),
    (nRD=>’0’,nWT=>’1’,CE=>’1’,RSEL=>’1’,DBUS => "ZZZZZZZZ",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "ZZZZZZZZ",EOV=>’0’),
    (nRD=>’1      プリンタドライバのミス
    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "ZZZZZZZZ",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "ZZZZZZZZ",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "ZZZZZZZZ",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "ZZZZZZZZ",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "ZZZZZZZZ",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "ZZZZZZZZ",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "ZZZZZZZZ",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "ZZZZZZZZ",EOV=>’0’),

    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "00000000",EOV=>’0’),
    (nRD=>’1’,nWT=>’0’,CE=>’1’,RSEL=>’0’,DBUS => "00001010",EOV=>’0’),
    (nRD=>’0’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "00000000",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "ZZZZZZZZ",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "ZZZZZZZZ",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "ZZZZZZZZ",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "ZZZZZZZZ",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "ZZZZZZZZ",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "ZZZZZZZZ",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "ZZZZZZZZ",EOV=>’0’),
    (nRD=>’1’,nWT=>’1’,CE=>’1’,RSEL=>’0’,DBUS => "ZZZZZZZZ",EOV=>’0’),
    (nRD=>’0’,nWT=>’1’,CE=>’1’,RSEL=>’1’,DBUS => "ZZZZZZZZ",EOV=>’1’) );
```

```
65          signal INDX: positive :=1;       -- Index of TestPattern(INDX) 0 not allowed

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

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

            signal DBUS     : Std_Logic_Vector(7 downto 0);
            signal CVAL     : Std_Logic_Vector(7 downto 1);
            signal CLK,nRD  : Std_Logic;
            signal nWt,nCLR : Std_Logic;
75          signal CE,RSEL  : Std_Logic;
            signal EQU      : Std_Logic;


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

80      Begin   -- Test Bench MAIN.  Connect target entity with test vector signals --
        --------------------------------------------------------------------------------
            nCLR    <= '0', '1' after 200 ns,'0' after 1 us,'1' after 1.2 us;


85          ------------------------------------------------------------------------
            CLK_P : Process     -- Clock pulse generator
            ------------------------------------------------------------------------
            Begin       CLK<='0';   wait for 25 ns;
                        CLK<='1';   wait for 25 ns;
90          End Process CLK_P;
            ------------------------------------------------------------------------

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

            CLKCOUNT_P : process(CLKCOUNT,CLK)  -- @CLKCOUNT=0 increment (INDX)
95          ------------------------------------------------------------------------
            -- You can modify TestStim process to control INDX increment timing NOT
            -- with CLKCOUNT=0 event BUT with some responce signal from the Target.
            -- This will provide a TestBench-Target handshaking.
            ------------------------------------------------------------------------

100         Begin
                if(CLK'EVENT and CLK='1') then
                    if(CLKCOUNT=MAXCOUNT) then  CLKCOUNT<=0;
                    else                        CLKCOUNT<=CLKCOUNT+1;
                    end if;
105             end if;
            End Process CLKCOUNT_P;
            ------------------------------------------------------------------------


            -- provide stimulus ------------------------------------------------------
110         TestStim: process(CLKCOUNT)     --   Read Test-Vectors into signals.
            ------------------------------------------------------------------------
            variable Vector : TestRecordT;
            variable EOV    : Std_Logic;
            Begin
115             if(CLKCOUNT=0) then
                    Vector  := TestPattern(INDX);
                    nRD     <= Vector.nRD;          -- apply the stimuls
                    nWT     <= Vector.nWT;
                    CE      <= Vector.CE;
120                 RSEL    <= Vector.RSEL;
                    DBUS    <= Vector.DBUS;
                    EOV     := Vector.EOV;
                    INDX    <= INDX+1;
                    if(EOV='1') then assert FALSE report "End.";
125                 end if;
                end if;
            End Process TestStim;
            ------------------------------------------------------------------------
```

クロックの発生例、当然ジェネリックで与えてもよい

ここではテストベクターを CLKCOUNT_P に合わせて順次読込んでいるが、任意の方法をとればよい。

altera tb.vhd

被テスト設計をインスタンス化してテストベンチの信号とを接続する。

```
130 ------------------------------------------------------------------------------
    UUT: TOP    -- Notice this is the sdf applicant REGION
    ------------------------------------------------------------------------------
        PORT MAP(   DBUS    =>DBUS,
                    CLK     =>CLK,
135                 nRD     =>nRD,
                    nWT     =>nWT ,
                    nCLR    =>nCLR,
                    CENBL   =>CE ,
                    RSEL    =>RSEL,
140                 CVAL    =>CVAL,
                    EQU     =>EQU
                     );
    END;
    ------------------------------------------------------------------------------
145
```

.sdfのデータを（個別に）適用する対象、レジョン名がＵＵＴとなる。

```
    ------------------------------------------------------------------------------
    configuration RTL of TestBench is
150     for TestMain
            for all: TOP
                use entity work.TOP(RTL);
            end for;
        end for;
155 end RTL;
```

アルテラのバック・アノテート出力 .vho の場合は、フィットしたデバイス名がアーキテクチャ名になるので、これを指定したコンフィギュレーションを作っておく。

```
    ------------------------------------------------------------------------------
    configuration  ALTERA_TEST of TestBench is
        for TestMain
160         for all: TOP
                use entity work.TOP(EPM7032SLC44_a10);    -- From Top.vho output
            end for;
        end for;
    end ALTERA_TEST;
165
```

```
    ------------------------------------------------------------------------------
    configuration XILINX_FIT of TestBench is
        for TestMain
            for all: TOP
170             use entity work.TOP(STRUCTURE);
            end for;
        end for;
    end XILINX_FIT;
```

ザイリンクスでのバックアノテートファイルは time_sim.vhd time_sim.sdf の２つになる（バイタル）
time_sim.vhd ：フィットしたデバイス内部のネットリスト
time_sim.sdf ：使用されているプリミティブの遅延データ

ｖｓｉｍするとき、エンティティを指定する方法と、コンフィギュレーションを指定する方法のどちらも使えるので、ワーク内にコンパイルしてあれば差はない。
エンティティから指定する方法を取った場合には、アーキテクチャ名も指定する事になる。

LEONARDO-ALTERA-VSYSTEM 簡略化作業フロー概念図

leonardoのワーク

！注意 top.vhdなどのように設計と同じ名前でラッパーを作成しないようにします

WRAPPER
最上位のエンティティをＳｔｄＬｏｇｉｃで再定義
目的はシミュレーション。手書きでも可。

F６K etc
テクノロジ
ライブラリ

パッケージ
@work

EXEMPLAR
leonardo

TOP.VHD
トップ
デザイン

LAYER.VHD
別階層の
デザイン

タイミング制約を与えオプティマイズし、故意にスラックさせ、REPORT_DELAYをＶｉｅｗＣｒｉｔｉｃａｌＰａｔｈチェックで実行してハイライトファイル（ex temp.hlt)しながらネットスコープ起動。ハイライトボタンで目的ネットを検索すると便利です。

netscope
ＥＤＩＦ
ビュワー

デザイン
ブラウザ

階層の組み替え（オプティマイズ対象の構成変更）
インスタンス・ポート名・プリミティブの確認
特に配線資源が微妙な場合に有効

TOP.EDF
EDIF出力

この段階で、すでにＲＴＬ記述ではなくなっている事に注意します

コンパイラ／インターフェース／EDIFネットリーダ設定でEXEMPLARを指定
SHOW LMF マッピングをチェックしておくとOPミスに気が付く事があります。

メニュー／コンパイラ／インターフェース
／VHDLネットライタの設定で参照される
バイタル・ライブラリ。
インストールしてあればよい（筈）です

アルテラの場合、バイタルは
maxplusII に読込ませて利用
できます

ALT_VTL.VHD
ALT_VTL.CMP

maxplus2
フィッター

コンパイラ／インターフェース／VHDLライタセッティング
でVHOを選択、同じくインターフェース／VHDLライタをセットしてコンパイルします。
アルテラの場合のシミュレーションはこのＶＨＯファイルだけで十分です

基本的にはWYSWYGでフィットさせます

POFファイル
SOFファイル

デバイス
ファイル

V8.3までの maxplusII のVHDLリーダはパーサが貧弱なので使用しないほうがよいと思いますが、コンパイラがハングした時は Vsystem やleonardoで調べると文法ミスなどのハング原因が判る事があります

実デバイス

アルテラでは同じデバイス・パッケージであっても速度ランクが異なる場合には再度コンパイルする事になるので、スマート・コンパイルオプションを使っておくと良いかも知れません。（ザイリンクスの場合はＰＡＲ以後の処理で済みます）

maxplusIIのワーク

TOP.VHO
VHOファイル
遅延データ内蔵

TOP.SDO
SDOファイル

実フィットデバイス名がアーキテクチャ名として出力されるので、ＴＢでは この新しいバインディングを使います。

このファイルは使用プリミティブの個別タイミングデータを返すものでネット情報は持っていません。
どうしても sdf を使いたい場合には、maxplusII/vhdl87/vital/v3_0でvlibしてworkを作成し、alt_vtlを全てvcomし、これをvlibでマッピングして使います。

ＳＤＦファイルを使う場合はVSIM/SDFで適用するREGIONを指定して使います。

WAVE.PS
波形出力
ファイル出力

VSYSTEMのワーク

ieee.std

TEXTIO

ライブラリマッピングしておく

TB.VHD
テスト
ベンチ

Vsystem
VCOM

RTLモデル

TOP(RTL)
ワーク内に
コンパイル

TB
ワーク内に
コンパイル

エンティティ・コンフィギュレーション
（ＲＴＬ)を指定してシミュレートしますが、論理合成とは関係なく処理されます。

VSIM

WAVE.PS
波形出力
ファイル出力

テキスト記述を修正してバインディングを切替えます。

外部テキストファイル(テストベクタ)取込み、アサート文による警告レベルデバイスとハンドシェークしたシミュレーション、ブレーク条件などが制御できます。 テストベクタなどの配列はインデックスが１から始まることに注意します。

GATEモデル

Vsystem
VCOM

ここで読み込むのはtop.VHOファイルです（top.vhdをVCOMするとRTLモデルになります）その次にtb.vhdをVCOMしてVSIMします。

VSIM

VSIMのwave/メニュー/signals/addtowaveformでsignals in designを選択すると全信号がモニタできます

TB.VHD
TOP.VHO

適用すべきバインディングはもはやＲＴＬではない事に注意します。
アルテラの場合、対象となるアーキテクチャ名＝デバイス名になります。

maxplusIIでは処理が困難な外部遅延素子（メモリなど）のモデル化が容易です。

ソース
記述依存

タイミング
データ包括

テスト
ベクタ

ソース記述
依存の構造

ゲートレベル
変換済の構造

leonardo-Xilinx-VSYSTEM 簡略化作業概念図

leonardoのワーク

Systems Workshop Inc 1998/7/25

Xe4K etc
テクノロジ
ライブラリ

パッケージ
@work

Wrapperとは最上位のエンティティを StdLogic で再定義したもの,目的はシミュレーションで利用価値は不明です。

！注意 top.vhdなどのように設計と同じ名前で
ラッパーを作成しないようにしましょう。

TOP.VHD
トップ
デザイン

LAYER.VHD
別階層の
デザイン

①設計を準備

EXEMPLAR
leonardo

netscope
ＥＤＩＦ
ビュワー

タイミング制約を与えてオプティマイズし、故意にスラックさせ、REPORT_DELAYをViewCriticalPathチェックで実行してネットスコープ起動。ハイライトファイル（temp.hlt)しながら目的ネットを検索すると便利です。

③ 詰込、ピン指定、タイミング負荷分散、階層組替え、等の処理。 ターゲット側からの資源制約もここで解決します

デザイン
ブラウザ

階層の組み替え（オプティマイズ対象の構成変更）インスタンス・ポート名・プリミティブの確認等。
特に配線資源が微妙な場合に有効です。

TOP.EDN
EDIF出力

④ＥＤＩＦ作成

xilinxの場合、leonardo出力ファイル名をTOP.EDNとしておきます

VSYSTEMのワーク

ieee.std
TEXTIO
ライブラリマッピングをしておきます

①動作環境を設計

TB.VHD
テスト
ベンチ

シミュレーションを停止させる為に意図的にエラーを発生させる事がよくあります。

Vsystem
VCOM

コンパイルには順序があります

TOP(RTL)
ワーク内に
コンパイル

ＴＢ
ワーク内に
コンパイル

WAVE.PS
波形出力
ファイル出力

VSIM

エンティティまたはコンフィギュレーションを指定してシミュレートしますが、実デバイスとは関係なく処理されます。ソースの文法チェックや、ＴＢの機能のチェックに非常に有用です。

予め記述を加工してバインディングを切替えられるようにしておくと楽です。

予めザイリンクスの SIMPRIMは
インストール・ディレクトリ内
にてvlibしてworkを作成し、
ＶＣＯＭしておき、
実作業時 vlib マッピング指定
で SIMPRIM ライブラリ名をフル
パスでこのディレクトリに指定
します。

②基本動作確認し、問題点を修正する為に①へ戻るループ（または③経由）で設計を完成します。

Xilinx のワーク

ファウンデーションでの処理の場合にはプロジェクトマネージャからメニューで/NEW PROJECT/としてフォルダを作成します、この操作でディレクトリ階層が１つ下がりますのでDOCUMENT/ADD で１つ上の階層の TOP.EDN ファイルを指定し、インプリメントアイコンでデザインマネージャを起動します。
この時、EDIFネットが SCHネットより古いから更新するか？と問い合わせて来ますので必ずＮＯとします。

FOUNDATION
PROJECT MGR
M1.4

FOUNDATION
Design MGR
M1.4

PROJECT MANAGER(.pdf)を使うとバイタルライブラリ SIMPRIMsが自動的に検索されますが、
フローエンジン／設定／オプション／タイミングシミュレーションデータの生成をチェックしておく事が必要です。
／オプション／コンフィギュレーション／テンプレートで、スタートアップ・タイ・コンフィギュレーション・CMOS/TTL等を設定します。

M1.3/1.4

必ず フローエンジン／設定／オプション／インプリメンテーション／テンプレート／インターフェースで、
フォーマットをvhdlに指定します。

デバイス指定は
デザインマネージャ
で再度行います

⑤シミュレーションファイルの作成

もし、ツールがうまく動作しない場合フローエンジンのコマンドヒストリをテキストで開き、同じ内容をＤＯＳ窓から実行すると処理できる場合があります（トランスレート／マップまで）
このような場合には、インストールの問題か、プロジェクトファイルの問題か、環境変数の設定が原因かをつきとめて対策とします。（当然、ルータはＤＯＳ窓からは実行 できません）
問題の発生した場所がトランスレートやマッピングの段階であれば、ログファイルをよく検討して、プリミティブの使い方などの基本的な間違いを検出できることがあります。

Top.Bit
ビットストリーム
ファイル

⑦デバイス作成

タイミングアナライザでの時間制約条件確認した後、電力計算に満足できれば、ＲＯＭの作成を行う為、以下の例のようなＤＯＳのコマンドを実行します（PROMに近いデバイスから順にビットファイルを列挙するだけですがバージョン管理には十分に注意します）

```
promgen -p mcs -o MXOpMX1 -s 512 -u 0 mx0_bd0.bit mx1.bit
```

MXOpMX1.MCS
書込み器に合わせたフォーマットのＲＯＭファイル

実デバイス
デバッグへ

time_sim.vhd
vhdファイル

time_sim.sdf
sdfファイル

エンティティ名は元のTOPですが、フィットされた結果、デバイス内部結線を反映したアーキテクチャを持つvhdファイルです。

アーキテクチャ名は元設計名とは無関係にstructure として出力されるのでＴＢでは新しいバインディングを使います。
このファイルは vhdl 形式のネットリストになっていますが、アルテラの.vhoと異なり、遅延情報は含まれていません。

このファイルは使用しているプリミティブの個別タイミングデータを返すもので、ネット情報は一切持っていません。

SIMPRIM
ライブラリ
マッピング

！適用すべきバインディングはもはやＲＴＬではない事に注意します。
ザイリンクスでは対象となるアーキテクチャ名は常に"structure" となります

⑥タイミング問題の発見
動作条件の確定

sdfファイルを使う時には VSIM/SDFで適用する REGION を指定して使います。
xilinxのシミュレーションでは、VSIMのメニューで以下の項目をチェックし
Use VITAL2.2b Mapping
VITAL Disable Timing Checks
同じくSpecify Genericsで設計外パラメタWIDTHを0以外の値に設定します。
等価コマンドラインは以下のとおりです

印刷して使うのは避けます
4.6Gでは印字バグあり
基本的にＣＲＴ上での解析のみの利用を考えたシステムです。 textioが有効。

ソース記述依存
タイミングデータ包括
テストベクタ
ソース記述依存の構造
ゲートレベル変換済の構造

```
vsim -Gwidth=1 +notimingchecks -vital2.2b -sdftype uut=d:\***\time_sim.sdf xilinx_fit（または testbench testmain）
```

意味：レジョンuutにtime_sim.sdfを適用し、コンフィギュレーション xilinx_fitをシミュレートせよ。
またはエンティティ指定で、（ファイルtb.vhdの）エンティティ名testbench内のアーキテクチャtestmainで、実デバイスに対応するレジョンuut（time_sim.vhdのtop）をシミュレートせよ。
どちらでも結果は同じになります。混在シミュレーションでは複数のレジョンに複数のsdfを指定します。

以下、wave * /view * / run -all などとします。assert文などtextioを利用しブレーク条件を使うとvhdlになります。

xilinx top.vhd

```
1       ----------------------------------------------------------------------
        -- N-bit COUNTER COMPARATOR   for testbench lessons.
        ----------------------------------------------------------------------
        library IEEE;
5       use IEEE.std_logic_1164.all;
        use ieee.std_logic_ARITH.all ;
        use ieee.std_logic_signed.all ;

        Entity TOP is
10          port (DBUS  : inout Std_Logic_Vector(7 downto 0);
                  P_CLK : in Std_Logic;
                  nRD   : in Std_Logic;
                  nWT   : in Std_Logic;
                  P_nCLR: in Std_Logic;
15                CENBL : in Std_Logic;
                  RSEL  : in Std_Logic;
                  CVAL  : out Std_Logic_Vector(7 downto 1);
                  EQU   : out Std_Logic );
        end TOP;
20
        Architecture RTL of  TOP is
        -- for Xilinx fitting ------------------------------------------------------
        COMPONENT BUFG  PORT( I: IN Std_Logic;  O: OUT  Std_Logic ); END COMPONENT ;
        ----------------------------------------------------------------------------
25        signal    DRO     : Std_Logic_Vector(7 downto 0);
          signal    CompReg : Std_Logic_Vector(7 downto 0);
          signal    CntrReg : Std_Logic_Vector(7 downto 0);
          signal    Counter : Integer range 0 to 255; -- Integer'high;
          signal    CLK,nCLR: Std_Logic;
30
        Begin
        -- for Xilinx fitting ------------------------------------------------------
            ux0: BUFG   PORT MAP( I => P_CLK, O => CLK  );
            ux1: BUFG   PORT MAP( I => P_nCLR,O => nCLR );
35      ----------------------------------------------------------------------------

            GEN1:for I in 0 to 7 GENERATE
                DBUS(I)<=DRO(I) when nRD='0' else 'Z';
            end GENERATE GEN1;
40
            CntrReg<= Conv_Std_Logic_Vector(Counter,8);
            DRO<= CompReg   when RSEL='0' else CntrReg;

            ----------------------------------------------------------------------
45          RegLdProc: Process(CLK,DBUS,nWT,CompReg)
            ----------------------------------------------------------------------
            begin
                if(CLK'EVENT and CLK='1') then
                    if(nWT='0') then    CompReg<= DBUS;
50                  else                CompReg<= CompReg;
                    end if;
                end if;
            end Process RegLdProc;
            ----------------------------------------------------------------------
55
            ----------------------------------------------------------------------
            CountProc: Process(CLK,CENBL,nCLR,nWT,Counter,CompReg)
            ----------------------------------------------------------------------
            begin
60              if(nCLR='0' or nWT='0') then
                                    Counter<=0;
                elsif(CLK'EVENT and CLK='1') then
                    if(CENBL='1') then  Counter<= Counter+1;
                    else                Counter<= Counter;
65                  end if;
                end if;
                CVAL<= Conv_Std_Logic_Vector(Counter,7);
            end Process CountProc;
            ----------------------------------------------------------------------
70
           EQU<='1' when (Counter= Conv_Integer(Compreg))
                    else '0';
        end  RTL;
```

```vhdl
Entity TestBench is
end;

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

Architecture TestMain of TestBench is

    component TOP   -- This is the TARGET design ----
    port (DBUS      : inout Std_Logic_Vector(7 downto 0);
          P_CLK     : in Std_Logic;
          nRD       : in Std_Logic;
          nWT       : in Std_Logic;
          P_nCLR    : in Std_Logic;
          CENBL     : in Std_Logic;
          RSEL      : in Std_Logic;
          CVAL      : out Std_Logic_Vector(6 downto 0);
          EQU       : out Std_Logic
          );
    end component;


    Type TestRecordT is record  -- TestVector format definition.
        nRD,nWt : Std_Logic ;
        CE,RSEL : Std_Logic ;
        DBUS    : Std_Logic_Vector(7 downto 0) ;
        EOV     : Std_Logic ;
    End record;

    Type TestArrayT is array(positive range <>) of TestRecordT;

    constant TestPattern : TestArrayT := (
    (nRD=>'1',nWT=>'1',CE=>'0',RSEL=>'0',DBUS => "00000000",EOV=>'0'),
    (nRD=>'1',nWT=>'1',CE=>'0',RSEL=>'0',DBUS => "00000000",EOV=>'0'),
    (nRD=>'1',nWT=>'1',CE=>'0',RSEL=>'0',DBUS => "00000000",EOV=>'0'),
    (nRD=>'1',nWT=>'0',CE=>'1',RSEL=>'0',DBUS => "00000101",EOV=>'0'),
    (nRD=>'0',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "00000000",EOV=>'0'),
    (nRD=>'0',nWT=>'1',CE=>'1',RSEL=>'1',DBUS => "ZZZZZZZZ",EOV=>'0'),
    (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "ZZZZZZZZ",EOV=>'0'),
    (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "ZZZZZZZZ",EOV=>'0'),
    (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "ZZZZZZZZ",EOV=>'0'),
    (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "ZZZZZZZZ",EOV=>'0'),
    (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "ZZZZZZZZ",EOV=>'0'),
    (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "ZZZZZZZZ",EOV=>'0'),
    (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "ZZZZZZZZ",EOV=>'0'),
    (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "ZZZZZZZZ",EOV=>'0'),
    (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "ZZZZZZZZ",EOV=>'0'),

    (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "00000000",EOV=>'0'),
    (nRD=>'1',nWT=>'0',CE=>'1',RSEL=>'0',DBUS => "00001010",EOV=>'0'),
    (nRD=>'0',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "00000000",EOV=>'0'),
    (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "ZZZZZZZZ",EOV=>'0'),
    (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "ZZZZZZZZ",EOV=>'0'),
    (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "ZZZZZZZZ",EOV=>'0'),
    (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "ZZZZZZZZ",EOV=>'0'),
    (nRD=>'1   (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "ZZZZZZZZ",EOV=>'0')
,
    (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "ZZZZZZZZ",EOV=>'0'),
    (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "ZZZZZZZZ",EOV=>'0'),
    (nRD=>'1',nWT=>'1',CE=>'1',RSEL=>'0',DBUS => "ZZZZZZZZ",EOV=>'0'),
    (nRD=>'0',nWT=>'1',CE=>'1',RSEL=>'1',DBUS => "ZZZZZZZZ",EOV=>'1') );

    signal INDX: positive :=1;       -- Index value of TestPattern(INDX)

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

```
70          -- Internal signals for Test-Bench operations -----------------------------

            signal DBUS     : Std_Logic_Vector(7 downto 0);
            signal CVAL     : Std_Logic_Vector(7 downto 1);
            signal CLK,nRD  : Std_Logic;
            signal nWt,nCLR : Std_Logic;
75          signal CE,RSEL  : Std_Logic;
            signal EQU      : Std_Logic;


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

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

            nCLR    <= '0', '1' after 200 ns,'0' after 1 us,'1' after 1.2 us;

85          --------------------------------------------------------------------------------

            CLK_P : Process     -- Clock pulse generator
            --------------------------------------------------------------------------------
            Begin       CLK<='0';   wait for 25 ns;
                        CLK<='1';   wait for 25 ns;
90          End Process CLK_P;
            --------------------------------------------------------------------------------

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

            CLKCOUNT_P : process(CLKCOUNT,CLK)  -- @CLKCOUNT=0 increment (INDX)
95          --------------------------------------------------------------------------------
            -- You can modify TestStim process to control INDX increment timing NOT
            -- with CLKCOUNT=0 event BUT with some responce signal from the Target.
            -- This will provide a TestBench-Target handshaking.
            --------------------------------------------------------------------------------

100         Begin
                if(CLK'EVENT and CLK='1') then
                    if(CLKCOUNT=MAXCOUNT) then  CLKCOUNT<=0;
                    else                        CLKCOUNT<=CLKCOUNT+1;
                    end if;
105             end if;
            End Process CLKCOUNT_P;
            --------------------------------------------------------------------------------


            -- provide stimulus ------------------------------------------------------
110         TestStim: process(CLKCOUNT)     --   Read Test-Vectors into signals.
            --------------------------------------------------------------------------------
            variable Vector : TestRecordT;
            variable EOV    : Std_Logic;
            Begin
115             if(CLKCOUNT=0) then
                    Vector  := TestPattern(INDX);
                    nRD     <= Vector.nRD;          -- apply the stimuls
                    nWT     <= Vector.nWT;
                    CE      <= Vector.CE;
120                 RSEL    <= Vector.RSEL;
                    DBUS    <= Vector.DBUS;
                    EOV     := Vector.EOV;
                    INDX    <= INDX+1;
                    if(EOV='1') then assert FALSE report "End.";
125                 end if;
                end if;
            End Process TestStim;
            --------------------------------------------------------------------------------

130     --------------------------------------------------------------------------------

        UUT: TOP    -- Notice this is the sdf applicant REGION
        --------------------------------------------------------------------------------
```

```
xilinx tb.vhd


        PORT MAP(   DBUS    =>DBUS,
                    P_CLK   =>CLK,
135                 nRD     =>nRD,
                    nWT     =>nWT ,
                    P_nCLR  =>nCLR,
                    CENBL   =>CE ,
                    RSEL    =>RSEL,
140                 CVAL    =>CVAL,
                    EQU     =>EQU --,
                     );

    END;
145 ------------------------------------------------------------------------------


    ------------------------------------------------------------------------------
    configuration RTL of TestBench is
150     for TestMain
            for all: TOP
                use entity work.TOP(RTL);
            end for;
        end for;
155 end RTL;

    ------------------------------------------------------------------------------
    configuration ALTERA_FIT of TestBench is
        for TestMain
160         for all: TOP
                use entity work.TOP(EPF6016TC144_a3);
            end for;
        end for;
    end ALTERA_FIT;
165
    ------------------------------------------------------------------------------
    configuration XILINX_FIT of TestBench is
        for TestMain
            for all: TOP
170             use entity work.TOP(STRUCTURE);
            end for;
        end for;
    end XILINX_FIT;
```