

パワーエレクトロニクス技術者向けFPGAセミナー 実習資料

2019年4月21日
Mywayプラス株式会社

演習0. Vivadoインストール

受講者は事前にVivado HIL WebPACKのインストールを行い、PCをご持参下さい（64bit OSのみ）

インストール方法および動作環境はXilinx社により随時更新されるとともに、処理系（お客様PC）に依存します。Mywayプラス社からは対応できかねますのでご了承下さい。

（参考）Vivadoダウンロードリンク

<https://japan.xilinx.com/support/download/index.html/content/xilinx/ja/downloadNav/vivado-design-tools.html>

※本セミナーで使用するソースコード・設計情報等は受講者の学習用途に限定いたします。Mywayプラス社の許可なく、その他の2次利用を禁止いたします。

演習 1. 基本操作およびシミュレーション

ゴール：基本的な開発環境の使用方法に慣れる、VHDLコード・テストベンチの構成を理解する。

1. 開発環境Vivado立上げ
2. ソースコード説明
3. テストベンチ説明
4. シミュレーションの実行

下記のプロジェクトファイルをクリックしてVivadoを立ち上げて下さい

project_training_1.xpr

プロジェクトの起動

- ・ Vivado起動

下記のプロジェクトファイルをクリック

project_training_1.xpr

- ・ ソースコードを見るには

Project Manager 内の Sources窓のHierachy(デフォルト) 画面内で、pwm_ifをクリック

- ・ テストベンチコードを見るには

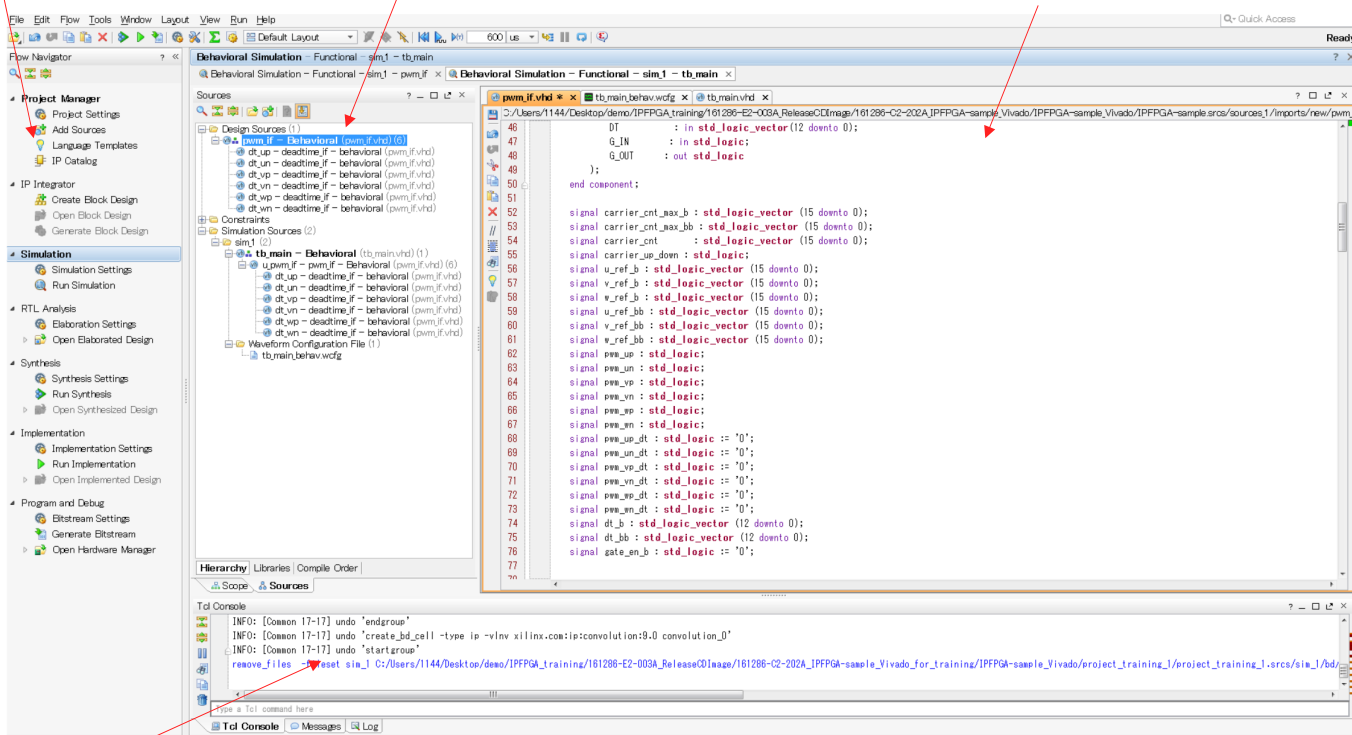
上記Hierachy画面内でSimulation Sorucesをクリックしてtb_mainをクリック

ウィンドウ構成 (例)

基本メニュー

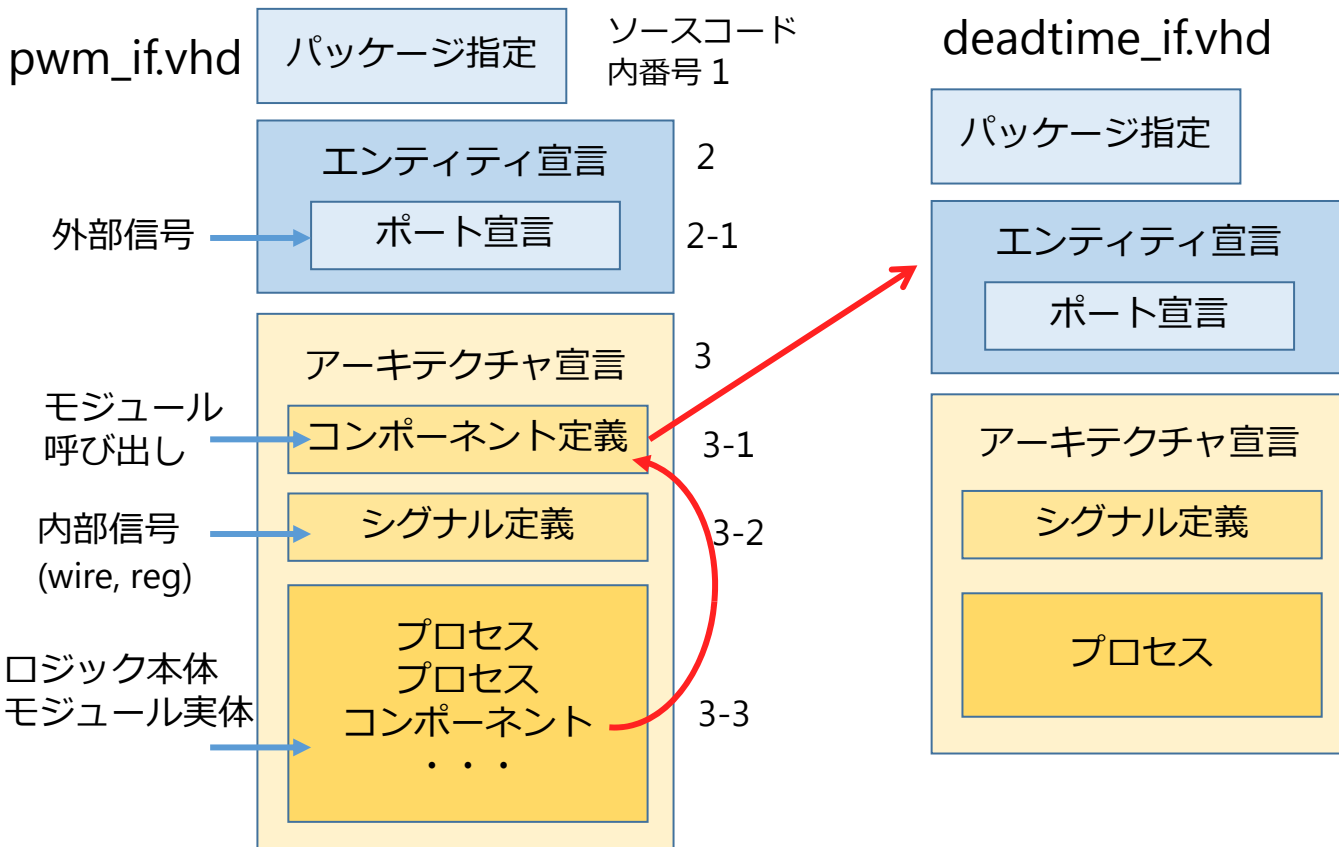
ファイル構成

中身 (ソースコード・波形等)



エラーメッセージ等

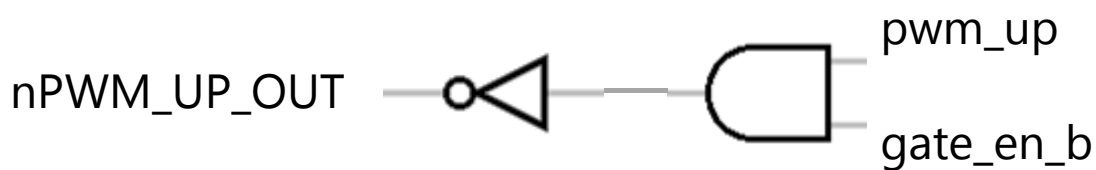
VHDLソースコード構成



組合せ回路

-- 3-3-1 Logic description -----

```
nPWM_UP_OUT <= not (pwm_up and gate_en_b);
nPWM_UN_OUT <= not (pwm_un and gate_en_b);
```



順序回路

-- 3-3-2 Process description -----

```
process(CLK_IN)
```

```
begin
```

```
  if CLK_IN'event and CLK_IN = '1' then
```

```
    if RESET_IN = '1' then
```

```
      u_ref_b <= X"0000"; -- initial
```

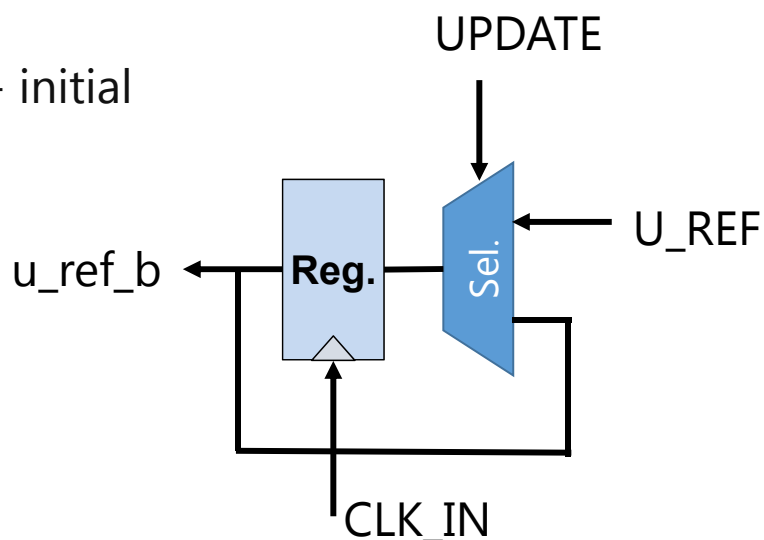
```
    elsif UPDATE = '1' then
```

```
      u_ref_b <= U_REF;
```

```
    else
```

```
      u_ref_b <= u_ref_b;
```

```
  end if;
```



キャリアカウンター（三角波生成用）

-- 3-3-2-3 carrier counter increment or decrement ---

```

1.初期化  if RESET_IN = '1' then
            carrier_cnt <= X"0000";
2.上り    elsif carrier_up_down = '1' then
            carrier_cnt <= carrier_cnt + 1;
3.下り    else
            carrier_cnt <= carrier_cnt - 1;
            end if;
    
```

三角波の山と谷の処理

-- 3-3-2-4 reference, carrier count and dead time -----

if RESET_IN = '1' then

（省略）

elsif carrier_cnt = X"0001" and carrier_up_down = '0' then

carrier_up_down <= '1';

谷の時

elsif carrier_cnt >= (carrier_cnt_max_bb -1) and carrier_up_down = '1'

carrier_up_down <= '0';

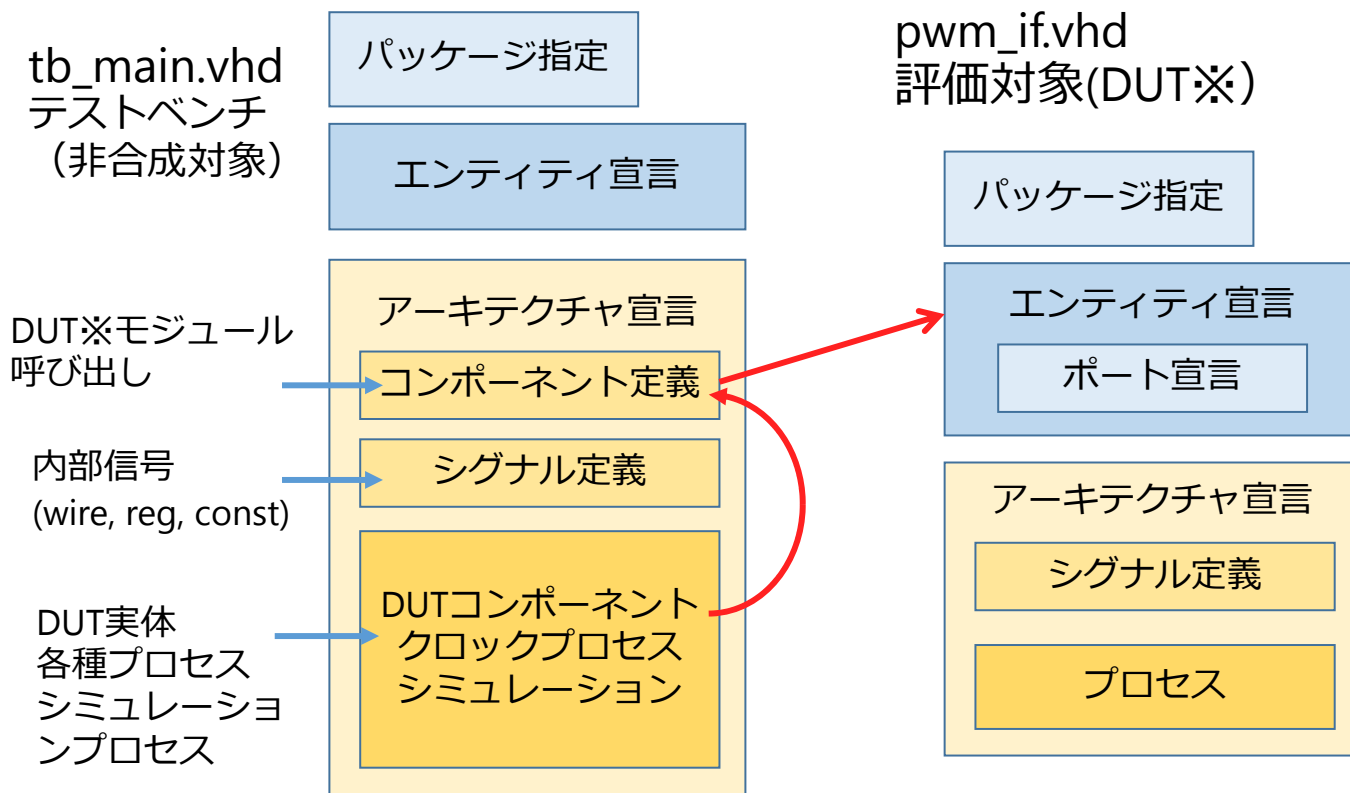
山の時

carrier_cnt_max_bb <= carrier_cnt_max_b;

end if;

山・谷以外のときは前状態保持

シミュレーション構成



※ DUT: Design Under Test

シミュレーションの実行

左側のFlow Navigator内のSimulation下のRun Simulationをクリック

Run Behavioral Simulationをクリック

画面上部の▶(T)マーク (時間を300usに指定)を押してさらに実行

波形を確認

観測したい信号はObjects窓内の信号をドラッグして波形表示窓にドロップ

演習2-1 シミュレーションとデバッグ

ゴール：コンパイルエラーから、記述ミスを見つけて修正できるようになる。シミュレーション結果から不具合を見つけてコードを修正できるようになる。

(3つのコンパイルエラーを修正)

1. 演習2プロジェクト project_training_2.xprを開いて下さい。
2. Sources窓のMessageの“critical warnings”をクリック、下部のMessages窓のメッセージを参考に、デバッグ(pwm_if.vhdを修正してfile->saveにて保存する)
3. シミュレーションを実行→エラーが出たらCritical Message -> Open Message View -> Tcl Consoleのメッセージを下から上に見てゆく
ERRORメッセージを参考に、pwm_if.vhdを修正→save
4. 上記をエラーがなくなるまで繰り返す

演習2-1 シミュレーションとデバッグ

(ヒント)

3つのコンパイルエラー

69行、207行、272行

コメントアウトされている隣の行を参照

演習2-2 シミュレーションとデバッグ

1. シミュレーションを実行し、V相のゲート制御波形 `tb_pwm_vp_n`, `tb_pwm_vn_n`の波形を表示して下さい。
2. (ヒント) Objects窓から`tb_pwm_up_n`と`tb_pwm_un_n`をドラッグして波形の窓の信号名が並んでいる領域にドロップ
3. U層のpwm波形を確認
4. U相と同様に正しいPWMパルスが出るように`pwm_if.vhd`を修正して下さい。
(ヒント) 189行

演習3 デッドタイムロジック

デバッグ済みの演習2のロジックに、デッドタイムロジックを追加して下さい

(ヒント)

223行のデッドタイムモジュールの呼び出しを利用
(現状はコメントアウト)

96行~101行のゲート出力(`nPWM_UP_OUT`等)の入力信号をデッドタイム付の信号に変更

pwm_if.vhd (抜粋) その1

```

-----
-- 1. Package declaration -----
-----

library ieee;
-----

-- 2. Entity declaration "pwm_if" module -----
-----

entity pwm_if is
-- 2-1. Port declaration used by caller -----
port (
    CLK_IN      : in std_logic;
    RESET_IN    : in std_logic;
    nPWM_UP_OUT : out std_logic; --nUSER_OPT_OUT(0)
    nPWM_UN_OUT : out std_logic; --nUSER_OPT_OUT(1)
    UPDATE      : in std_logic;
    CARRIER    : in std_logic_vector (15 downto 0);
    U_REF       : in std_logic_vector (15 downto 0);
    V_REF       : in std_logic_vector (15 downto 0);
    W_REF       : in std_logic_vector (15 downto 0);
    DEADTIME    : in std_logic_vector (12 downto 0);
    GATE_EN     : in std_logic
);
end pwm_if;

-----
-- 3. Architecture description of "pwm_if" module -----
-----

architecture Behavioral of pwm_if is
-- 3-1. Component declaration used in this module -----
component deadline_if is
    Port (
        CLK_IN      : in std_logic;
        RESET_IN    : in std_logic;
        DT          : in std_logic_vector(12 downto 0);
        G_IN        : in std_logic;
        G_OUT       : out std_logic
    );
end component;

-- 3-2. Signal declaration used in this module -----
signal carrier_cnt_max_b : std_logic_vector (15 downto 0);
signal carrier_cnt_max_bb : std_logic_vector (15 downto 0);
signal carrier_cnt       : std_logic_vector (15 downto 0);
signal carrier_up_down   : std_logic;
signal u_ref_b           : std_logic_vector (15 downto 0);

```

pwm_if.vhd (抜粋) その2

```

-- 3-3 Logic description -----
begin
-- 3-3-1 Logic description -----
    nPWM_UP_OUT <= not (pwm_up and gate_en_b);
    nPWM_UN_OUT <= not (pwm_un and gate_en_b);
-- 3-3-2 Process description -----
    process(CLK_IN)
    begin
        if CLK_IN'event and CLK_IN = '1' then
-- 3-3-2-1 GATE_EN signal registered -----
            if RESET_IN = '1' then
                gate_en_b <= '0';
            else
                gate_en_b <= GATE_EN;
            end if;
-- 3-3-2-2 reference, carrier count and dead time are conditionally updated -----
            if RESET_IN = '1' then
                carrier_cnt_max_b <= X"0000"; -- X"1388"=10kHz
            elsif UPDATE = '1' then
                carrier_cnt_max_b <= CARRIER;
                carrier_cnt <= carrier_cnt;
                u_ref_b <= U_REF;
                dt_b <= DEADTIME;
            else
                carrier_cnt_max_b <= carrier_cnt_max_b;
                carrier_cnt <= carrier_cnt;
                u_ref_b <= u_ref_b;
                dt_b <= dt_b;
            end if;
-- 3-3-2-3 carrier counter increment or decrement -----
            if RESET_IN = '1' then
                carrier_cnt <= X"0000";
            elsif carrier_up_down = '1' then
                carrier_cnt <= carrier_cnt + 1;
            else
                carrier_cnt <= carrier_cnt - 1;
            end if;
-- 3-3-2-4 reference, carrier count and dead time are conditionally updated -----
            if RESET_IN = '1' then
                carrier_up_down <= '1';
                carrier_cnt_max_bb <= X"1388"; -- or X"1388"
            elsif carrier_cnt = X"0001" and carrier_up_down = '0' then -- at bottom
                carrier_up_down <= '1';
                carrier_cnt_max_bb <= carrier_cnt_max_bb;
            elsif carrier_cnt >= (carrier_cnt_max_bb - 1) and carrier_up_down = '1' then -- at top
                carrier_up_down <= '0';
                carrier_cnt_max_bb <= carrier_cnt_max_b;
            else
                carrier_up_down <= carrier_up_down;
                carrier_cnt_max_bb <= carrier_cnt_max_bb;
            end if;
        end if;
    end process;

```

pwm_if.vhd (抜粋) その3

```

-- 3-3-3 Process description -----
process(CLK_IN)
begin
  if CLK_IN'event and CLK_IN = '1' then
    if RESET_IN = '1' then
      u_ref_bb <= X"0000"; -- m = X"09C4" = 0.5
    elsif carrier_cnt = (carrier_cnt_max_bb -1) and carrier_up_down = '1' then
      u_ref_bb <= u_ref_b;
    end if;
    if RESET_IN = '1' then ----- U phase pulse gen
      pwm_up <= '0';
      pwm_un <= '0';
    elsif carrier_cnt >= u_ref_bb then
      pwm_up <= '0';
      pwm_un <= '1';
    else
      pwm_up <= '1';
      pwm_un <= '0';
    end if;

    if RESET_IN = '1' then----- V phase pulse gen

    if RESET_IN = '1' then----- W phase pulse gen

    end if;
  end process;

-- 3-3-4 Process description -----
process(CLK_IN)
begin
  if CLK_IN'event and CLK_IN = '1' then
    if RESET_IN = '1' then
      dt_bb <= '0' & X"190"; -- 4us
    elsif carrier_cnt = (carrier_cnt_max_bb -1) then
      dt_bb <= dt_b;
    end if;
  end if;
end process;

-- 3-3-5 Component instantiation -----
-- dt_up : deadline_if port map (CLK_IN => CLK_IN, RESET_IN => RESET_IN, DT => dt_bb, G_IN => pwm_up,
G_OUT => pwm_up_dt);
-- dt_un : deadline_if port map (CLK_IN => CLK_IN, RESET_IN => RESET_IN, DT => dt_bb, G_IN => pwm_un,
G_OUT => pwm_un_dt);
-- dt_vp : deadline_if port map (CLK_IN => CLK_IN, RESET_IN => RESET_IN, DT => dt_bb, G_IN => pwm_vp,
G_OUT => pwm_vp_dt);
-- dt_vn : deadline_if port map (CLK_IN => CLK_IN, RESET_IN => RESET_IN, DT => dt_bb, G_IN => pwm_vn,
G_OUT => pwm_vn_dt);
-- dt_wp : deadline_if port map (CLK_IN => CLK_IN, RESET_IN => RESET_IN, DT => dt_bb, G_IN => pwm_wp,
G_OUT => pwm_wp_dt);
-- dt_wn : deadline_if port map (CLK_IN => CLK_IN, RESET_IN => RESET_IN, DT => dt_bb, G_IN => pwm_wn,
G_OUT => pwm_wn_dt);

end Behavioral;
-- End of entity -----

```