Enhanced PWM Circuit

Signature and Grading Sheet

Group #:_________ Name(s):____ .

Grading

- Section 4.1(a): VHDL code (45points):______.
  Attach code printout (with proper header and comment)
- Section 4.1(b): Block diagram (15 points):______.
  Attach diagram (can be manually drawn and photoed)
- Section 4.2(b) RT-level simulation waveform (20 points):______.
  Attach simulation waveform screen capture
- Section 4.3(e): post-synthesis simulation waveform (10 points): ______.
  Attach simulation waveform screen capture
- VHDL code format and comments (10 points):______.

Total points: .

Experiment
Enhanced PWM Circuit

1 Purpose

To design an enhanced PWM (Pulse Width Modulation) circuit

2 Reading

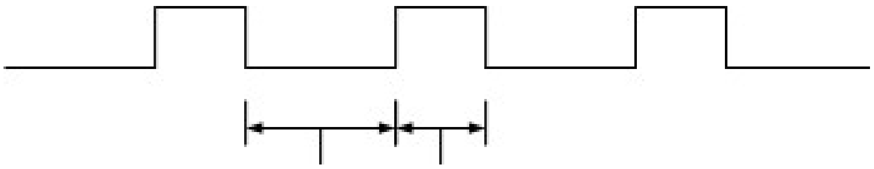
Chapter 5 of *FPGA Prototyping by VHDL Examples 2nd edition.*

3 Project specification

A PWM circuit has an output pulse whose duty cycle (i.e. the fraction of a period that is "high") is specified by a control signal. In the enhanced version, we want to control the exact time of the on and off intervals. The inputs and output of this circuit are specified as follows:

- `clk:` input clock signal. The period of `clk` is *Tc.*
- `d:` a 4-bit input signal. It is interpreted as an unsigned number and used to specify the "off-interval" of the `pulse` signal.
- `w:` a 4-bit input signal. It is interpreted as an unsigned number and used to specify the "on-interval" of the `pulse` signal.
- `pulse`: a 1-bit output.

The waveform of the `pulse` signal is shown below. Note that it can be considered as a squared wave with a period of *(d+w)*Tc* and with the *w*Tc* portion asserted.



The design must synchronous or 50% will be deducted.

4 Design Procedures

4.1 VHDL design

The entity declaration of this design is shown below:

```
entity epwm is
   port(
      clk, reset: in std_logic;
      d, w: in std_logic_vector(3 downto 0);
      pulse: out std_logic
     );
end epwm;
```

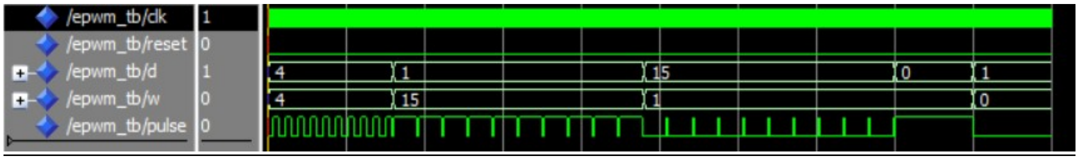
(a) Derive VHDL code.

(b) Draw the top-level conceptual diagram (up to 10 blocks) of your code.

4.2 Simulation

(a) Use the testbech (epwm_tb.vhd) to simulate your VHDL code. The testbench includes the test patterns of d=4, w=4; d=1, w=15; d=15, w=1; d=0, w=1; d=1, w=0.

(b) Develop a proper "layout" and "format" for the simulated waveform. To get full credits, <u>the input and output signals should be properly arranged and represented in a proper format so that the simulation result can be easily understood (like the one below).</u> Do a screen capture of the simulated result.



4.3 Post-synthesis ModelSim simulation

(a) Perform compiling (synthesis/placement and routing) and obtain the .vho file.

(b) Follow the format guideline to add a header to the file and verify the "structure" architecture body is generated.

(c) Revise the testbench to use the "structure" architecture for uut.

(d) Perform post-synthesis simulation

(e) Do a screen capture of the simulated result.

(f) Since the waveform is the same as the RT-level simulation, the snapshot should include the expanded **uut** on the left panel. <u>No point will be given if the **uut** unit is not expanded in the snapshot.</u>