# ESDII Lab 2- PWM Module

Aliana Tejeda *CPET*

## I. INTRODUCTION

THE purpose of this lab was to create an PWM module that allows the user to control the duty cycle and period of the module.

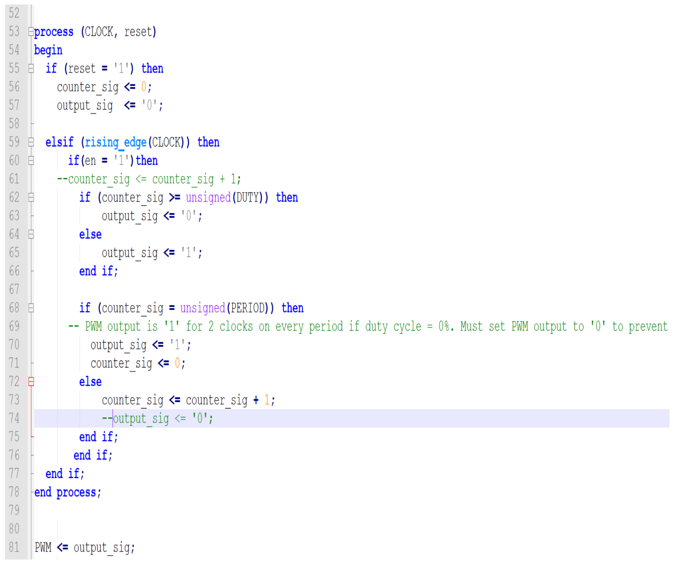
## II. CODE

### *A. Comparison & Conclusion*

The **Top.vhd** file is composed of two if statements that compare the duty count to the period count. The signal called counter-sig has the range that is from *"0 to 67108864"* which holds the count to count to 50Mhz. while the count_sig is increments if it hits the same count as the period, it restarts, as well as if it reaching the count of the duty it will restart. As for the output, as long as the count is less than the duty, the output will be a 1.

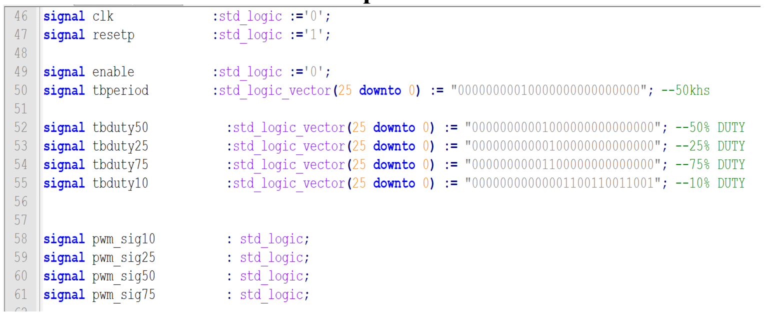
## III. CODE DISCUSSION

**The if statements that compare the duty and period counts**



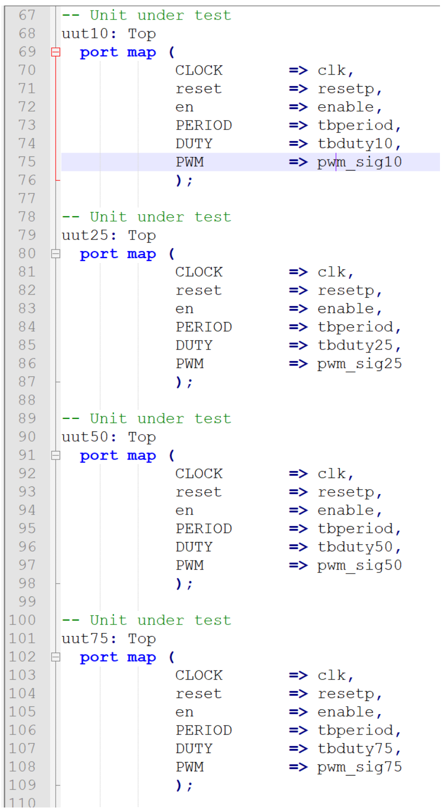
```vhdl

process (CLOCK, reset)
begin
  if (reset = '1') then
    counter_sig <= 0;
    output_sig  <= '0';

  elsif (rising_edge(CLOCK)) then
      if(en = '1')then
    --counter_sig <= counter_sig + 1;
        if (counter_sig >= unsigned(DUTY)) then
            output_sig <= '0';
        else
            output_sig <= '1';
        end if;

        if (counter_sig = unsigned(PERIOD)) then
      -- PWM output is '1' for 2 clocks on every period if duty cycle = 0%. Must set PWM output to '0' to prevent
          output_sig <= '1';
          counter_sig <= 0;
        else
            counter_sig <= counter_sig + 1;
            --output_sig <= '0';
        end if;
       end if;
  end if;
end process;


PWM <= output_sig;
```

**The signals set for the different duty cycles and the static period**



```vhdl
signal clk              :std_logic :='0';
signal resetp           :std_logic :='1';

signal enable           :std_logic :='0';
signal tbperiod         :std_logic_vector(25 downto 0) := "00000000010000000000000000"; --50khs

signal tbduty50           :std_logic_vector(25 downto 0) := "00000000001000000000000000"; --50% DUTY
signal tbduty25           :std_logic_vector(25 downto 0) := "00000000000100000000000000"; --25% DUTY
signal tbduty75           :std_logic_vector(25 downto 0) := "00000000001100000000000000"; --75% DUTY
signal tbduty10           :std_logic_vector(25 downto 0) := "00000000000001100110011001"; --10% DUTY


signal pwm_sig10          : std_logic;
signal pwm_sig25          : std_logic;
signal pwm_sig50          : std_logic;
signal pwm_sig75          : std_logic;
```

**The multiple port maps for sending in the different duty cycles to be tested**



```vhdl
-- Unit under test
uut10: Top
  port map (
             CLOCK        => clk,
             reset        => resetp,
             en           => enable,
             PERIOD       => tbperiod,
             DUTY         => tbduty10,
             PWM          => pwm_sig10
             );

-- Unit under test
uut25: Top
  port map (
             CLOCK        => clk,
             reset        => resetp,
             en           => enable,
             PERIOD       => tbperiod,
             DUTY         => tbduty25,
             PWM          => pwm_sig25
             );

-- Unit under test
uut50: Top
  port map (
             CLOCK        => clk,
             reset        => resetp,
             en           => enable,
             PERIOD       => tbperiod,
             DUTY         => tbduty50,
             PWM          => pwm_sig50
             );

-- Unit under test
uut75: Top
  port map (
             CLOCK        => clk,
             reset        => resetp,
             en           => enable,
             PERIOD       => tbperiod,
             DUTY         => tbduty75,
             PWM          => pwm_sig75
             );
```