```
------------------------------------------------------------
-- CWRUCutter_PWMOutput.vhd
-- EJ Kreinar
--
-- Creates a PWM signal
--
-- Inputs:
--   POS_PULSE_LEN: # of Ticks that the PWM output should be high
--   CYC_PULSE_LEN: # of Ticks in a period
--
-- Outputs:
--   PWM_OUT: The PWM output signal
--
-- Notes:
--
-- History
-- 9/19: ejk43- Created
------------------------------------------------------------

Library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity CWRUCutter_PWMOutput is
    port (
        CLK           : in  std_logic;
        aRESET        : in  std_logic;
        POS_PULSE_LEN : in std_logic_vector(31 downto 0) := (others => '0');
        CYC_PULSE_LEN : in std_logic_vector(31 downto 0) := (others => '0');
        PWM_OUT       : out  std_logic
      );
end CWRUCutter_PWMOutput;

architecture rtl of CWRUCutter_PWMOutput is
    signal pwm_count: unsigned(31 downto 0);
    signal pwm_reset: std_logic;
    signal pos_int  : std_logic_vector(31 downto 0);
    signal cyc_int  : std_logic_vector(31 downto 0);
    -- signal pos_reg: unsigned(32 downto 0);
    -- signal cyc_reg: unsigned(32 downto 0);
begin
    process(aRESET, CLK) begin
      if(aRESET = '1') then
        PWM_OUT       <= '0';
        pwm_count     <= (others => '0');
        pwm_reset     <= '0';
        pos_int       <= (others => '0');
        cyc_int       <= (others => '0');

      elsif rising_edge(clk) then

        -- RESET
        -- Reset at end of each cycle AND at beginning
        --  of the loop (because cyc_int = '0')
        if std_logic_vector(pwm_count) = cyc_int then
            pwm_reset <= '1';
        else
            pwm_reset <= '0';
        end if;

        -- COUNTER
        if pwm_reset = '1' then
            pwm_count <= (others => '0');
            pos_int   <= POS_PULSE_LEN;
            cyc_int   <= CYC_PULSE_LEN;
        else
            pwm_count <= pwm_count + 1;
        end if;

        -- OUTPUT
        if pwm_reset = '1' then
            PWM_OUT <= '1';
        elsif std_logic_vector(pwm_count) = pos_int then
            PWM_OUT <= '0';
        end if;

      end if;
    end process;
end rtl;
```