

Licznik synchroniczny zliczający w górę i dół (dodatkowe wejście Up) modulo n pracujący w kodzie binarnym z wpisem synchronicznym L i zezwoleniem zegara. Liczbę n ustalamy za pomocą zewnętrznego parametru.

1. Kod źródłowy (zał. counter.vhd).

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_unsigned.ALL;
use ieee.numeric_std.all;

entity counter is
    generic (n : STD_LOGIC_VECTOR := "0111"
            );
    Port ( Clk, Reset, L, Up, Enable : in  STD_LOGIC;
           wpis : in  STD_LOGIC_VECTOR (n'length-1 downto 0);
           Q : out STD_LOGIC_VECTOR (n'length-1 downto 0)
            );
end counter;

architecture Behavioral of counter is
    signal count : STD_LOGIC_VECTOR (n'length-1 downto 0) :=(others=>'0') ;
begin
    process(Clk, Reset, L, Up, Enable)
    begin
        if rising_edge(Clk) and Enable= '1' then
            if Reset = '1' then
                count<= (others=>'0');
            elsif L='1' and wpis <= (n-'1') then
                count<= wpis;
            elsif L='1' and wpis > (n-'1') then
                count<= (n-'1');
            elsif Up='1' and count /= (n-'1') then
                count<= count +'1';
            elsif Up='1' and count = (n-'1') then
                count<= (others=>'0');
            elsif Up='0' and not(count=0) then
                count<= count -'1';
            elsif Up='0' and count = 0 then
                count<= n-'1';
            else null;
            end if;
        end if;
    end process;
    Q<=count;
end Behavioral;
```

2. Test bench (zał. test8_tb.vhd).

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--USE ieee.numeric_std.ALL;

ENTITY test8_tb IS
END test8_tb;

ARCHITECTURE behavior OF test8_tb IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT counter
    PORT(
        Clk : IN std_logic;
        Reset : IN std_logic;
        L : IN std_logic;
        Up : IN std_logic;
        Enable : IN std_logic;
        wpis : IN std_logic_vector(3 downto 0);
        Q : OUT std_logic_vector(3 downto 0)
    );
    END COMPONENT;

    --Inputs
    signal Clk : std_logic := '0';
    signal Reset : std_logic := '0';
    signal L : std_logic := '0';
    signal Up : std_logic := '1';
    signal Enable : std_logic := '0';
    signal wpis : std_logic_vector(3 downto 0) := "0011";

    --Outputs
    signal Q : std_logic_vector(3 downto 0);

    -- Clock period definitions
    constant Clk_period : time := 10 ns;

BEGIN

    -- Instantiate the Unit Under Test (UUT)
    uut: counter PORT MAP (
        Clk => Clk,
        Reset => Reset,
        L => L,

```

```

    Up => Up,
    Enable => Enable,
    wpi => wpi,
    Q => Q
);

-- Clock process definitions
Clk_process :process
begin
    Clk <= '0';
    wait for Clk_period/2;
    Clk <= '1';
    wait for Clk_period/2;
end process;

-- Stimulus process
stim_proc: process
begin
    -- hold reset state for 100 ns.
    wait for 100 ns;

    Enable<='1';
    wait for 100 ns;
    L<='1';
    wait for Clk_period;

    L<='0';
    Up<='0';
    wait for 100 ns;
    Up<='1';
    wait for 100 ns;
    Reset <='1';

    wait for Clk_period*10;

    -- insert stimulus here

    wait;
end process;

END;
```

3. Opis projektu.

Projekt posiada wejścia:

- Clk – zegarowe licznika,
- Reset – resetujące licznik,
- L – powodujące wpisanie na wyjście licznika wcześniej ustalonej wartości,

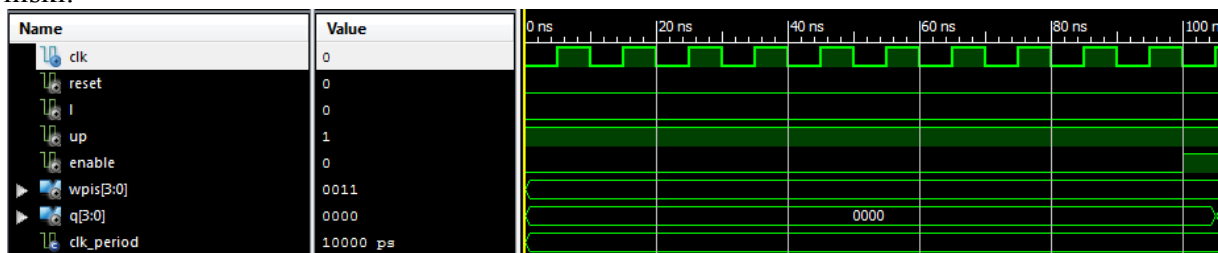
- Up – decydujące o kierunku zliczania licznika,
- Enable – zezwalające na zliczanie,
- wpis – wartość mogąca zostać wpisana na wyjście licznika.

Posiada także wyjście Q, na które podawana zostaje wartość zliczeń. Układ zlicza (mod n) od 0 do n-1, lub od n-1 do 0. Wszystko odbywa się w kodzie binarnym. Licznik pracuje synchronicznie, więc wszystkie jego zmiany mogą się wydarzyć tylko gdy wykryte zostanie zbocze narastające zegara i wejście zezwalające Enable przyjmie wartość 1. Jeśli spełnione są te dwa warunki to:

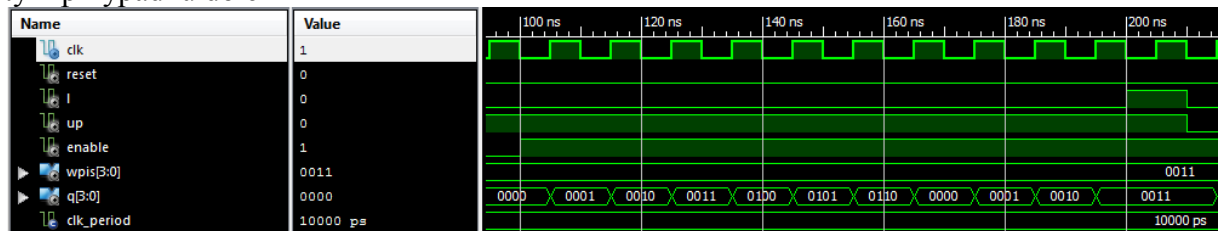
- licznik zeruje się kiedy przy zliczaniu do góry osiągnie wartość n-1, lub na wejście reset podany zostanie stan wysoki.
- licznik przyjmuje wartość podaną na wejście wpis, jeśli na wejście L podany zostanie stan wysoki i wartość wpis jest mniejsza lub równa n-1,
- licznik przyjmuje wartość n-1, jeśli na wejście L podany zostanie stan wysoki i wartość wpis jest większa od n, a także jeśli przy zliczaniu w dół osiągnie wartość 0,
- licznik zlicza do góry, kiedy na wejście Up podany jest stan wysoki,
- licznik zlicza w dół, kiedy na wejście Up podany jest stan niski,
- wykonywana jest instrukcja pusta gdy, wystąpi jakiś nieprzewidziany stan

4. Symulacja.

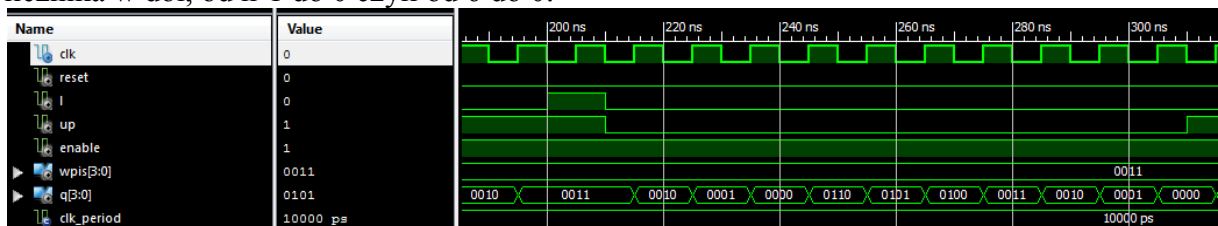
Na początku wartość parametru Enable ustalona jest na 0, więc na wyjściu utrzymuje się stan niski.



Po 100 ns Enable przyjmuje wartość 1, więc licznik zaczyna zliczać od 0 do n-1, a więc w tym przypadku do 6



Po kolejnych 100 ns L przyjmuje na czas okresu zegara wartość 1, więc na wyjście licznika podana jest wartość wektora wpis, czyli 3. Dzieje się to w momencie gdy wartość 3 miała zmienić się na 4. W efekcie wartość 3 utrzymuje się przez 2 okresy. Następnie Up przyjmuje wartość 0, w tym samym czasie gdy L również przyjmuje wartość 0, co skutkuje zliczaniem licznika w dół, od n-1 do 0 czyli od 6 do 0.



Po kolejnych 100ns Up wraca do wartości 1 i licznik znów zlicza do góry, aby po kolejnych 100ns przyjąć na wyjściu wartość 0 do końca symulacji, przez to że reset przyjmuje 1.

