

Układy cyfrowe i systemy wbudowane

SPRAWOZDANIE NR 6

Licznik Sterowany w VHDL-u

PROWADZĄCY:

DR INŻ. JACEK MAZURKIEWICZ

AUTORZY:

MATEUSZ GRUSZKA 249448

BARTOSZ RUDNIK 248893

Grupa: C

TERMIN ZAJĘĆ:

WT 9:15 TP

1. Cel ćwiczenia

Celem ćwiczenia było zaimplementowanie synchronicznego licznika 8-bitowego, pracującego w NKB przy użyciu języka programowania VHDL. Licznik ten powinien mieć dostępną opcję załadowania wartości startowej zliczania, powinien sygnalizować przekroczenie połowy zakresu zliczania na linii wyjściowej HALF, powinien obsługiwać sterowanie przy pomocy pojedynczych linii START, STOP, PAUZA, LOAD oraz RESET. Dodatkowym założeniem przez nas przyjętym jest wykonanie licznika działającego jako licznik pozytywny.

2. Implementacja VHDL

W pierwszej kolejności zadeklarowane zostały wszystkie sygnały potrzebne do uzyskania pełnej funkcjonalności układu, następnie zadeklarowane zostały również dwa sygnały pomocnicze: startValue2, który użyty został do zliczania oraz sygnał play, który określa na podstawie otrzymanego sterowania czy kontynuować zliczanie czy należy je zatrzymać. W przypadku sygnałów startValue, startValue2, oraz output zastosowane zostały sygnały typu UNSIGNED, użycie tego typu jest możliwe po zaimportowaniu biblioteki IEEE.NUMERIC_STD.ALL. Biblioteka ta pozwala na używanie wektorów liczb typu UNSIGNED oraz SIGNED, a także na przeprowadzenie operacji arytmetycznych na wektorach wyżej wymienionych typów. Działanie licznika powiązane jest wzrastającym zboczem zegara, zliczanie odbywa się pod warunkiem, że sygnał play ma wartość wynoszącą '1'. Wartość sygnału pomocniczego play jest modyfikowana przy pomocy linii sterujących START, STOP oraz PAUZA. Aktualna wartość licznika znajduje się w wektorze startValue2 ponieważ przedstawiany przez nas licznik działa jako licznik pozytywny to wartość zliczania jest inkrementowana przy każdym wzrastającym zboczu zegara i pod warunkiem, że pomocniczy sygnał play ma wartość '1'. Wartość zliczania jest zerowana w przypadku gdy osiągnięty zostanie limit wartości dla licznika 8-bitowego, a także w przypadku wystąpienia wzrastającego zbocza na linii sterującej RESET. Wartość zliczania może być także zmodyfikowana przez linię sterującą LOAD, w przypadku wystąpienia wzrastającego zbocza na linii LOAD jako nową wartość zliczania przyjmuje się wartość wektora startValue podaną w pliku testowym. Ostatnim obsługiwanym elementem jest sygnalizacja przekroczenia połowy zakresu zliczania. Funkcja ta zrealizowana jest poprzez wykonanie porównania aktualnej wartości zliczania z wartością będącą połową zakresu zliczania, jeśli aktualna wartość jest większa od porównywanej wartości to jest to sygnalizowane na linii HALF poprzez ustawienie tam sygnału '1', w przeciwnym wypadku linia HALF przyjmuje wartość '0'.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity Licznik_vhdlModule is
    Port ( clk : in  STD_LOGIC;
          Start : IN std_logic;
          Stop : IN std_logic;
          Pauza : IN std_logic;
          Load : IN std_logic;
          Reset : IN std_logic;
          startValue : IN  unsigned(7 downto 0);

          output : out unsigned (7 downto 0);
          half : out STD_LOGIC
    );
end Licznik_vhdlModule;
```

```

architecture Behavioral of Licznik_vhdlModule is
    signal startValue2 : unsigned(7 downto 0) := "00000000";
    signal play : std_logic := '0';
begin
    process1 : process (clk, Reset, Load, Start, Stop, Pauza)
    begin
        if rising_edge(clk) then
            if (play = '1' AND startValue2 < "11111111") then
                startValue2 <= startValue2 + 1;
            end if;
            if (startValue2 = "11111111") then
                startValue2 <= "00000000";
            end if;
        end if;
        if rising_edge(Start) then
            play <= '1';
        end if;
        if rising_edge(Stop) then
            play <= '0';
        end if;
        if rising_edge(Pauza) then
            play <= '0';
        end if;
        if rising_edge(Reset) then
            startValue2 <= "00000000";
        end if;
        if rising_edge(Load) then
            startValue2 <= startValue;
        end if;
        if (startValue2 >= "10000000") then
            half <= '1';
        end if;
        if (startValue2 < "10000000") then
            half <= '0';
        end if;
    end process process1;
    output <= startValue2;
end Behavioral;

```

Listing 1 Implementacja licznika w VHDL

W pliku testowym umieszczone zostały pobudzenia dla linii sterujących licznikiem.

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
ENTITY Licznik8Bit_TestBench IS
END Licznik8Bit_TestBench;
ARCHITECTURE behavior OF Licznik8Bit_TestBench IS
    COMPONENT Licznik_vhdlModule
    PORT(
        clk : IN std_logic;
        Start : IN std_logic;
        Stop : IN std_logic;
        Pauza : IN std_logic;
        Load : IN std_logic;
        Reset : IN std_logic;
        startValue : IN unsigned (7 downto 0);

        output : OUT unsigned (7 downto 0);

```

```

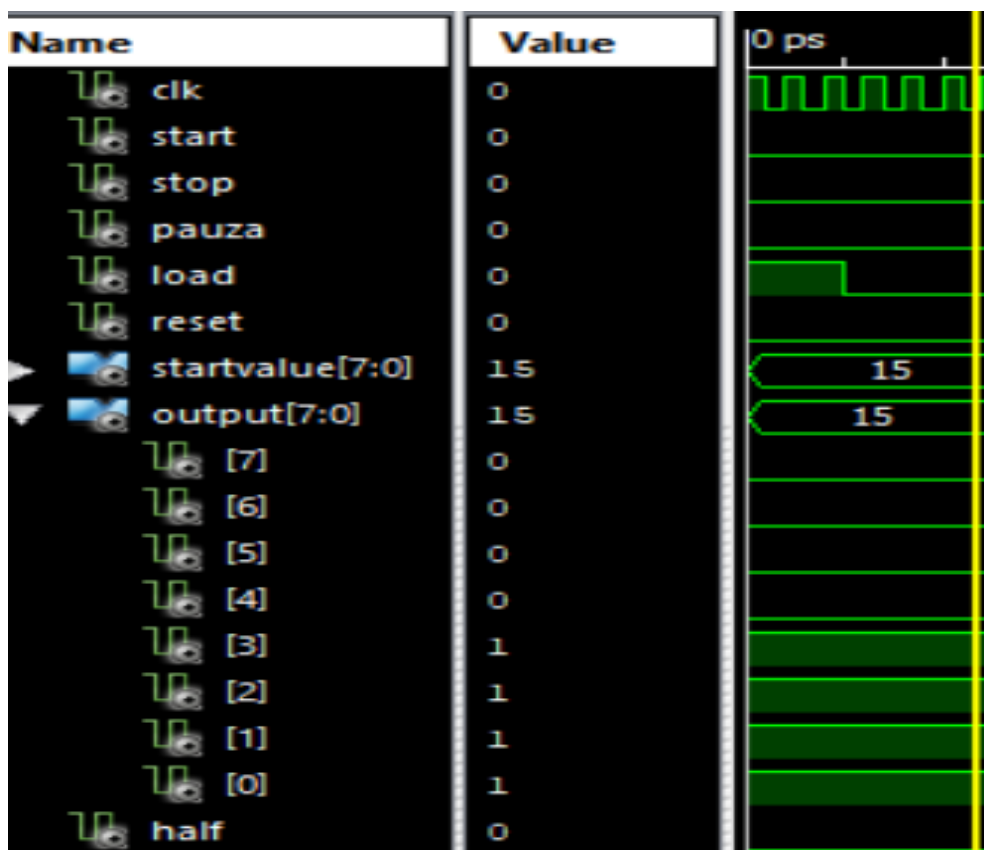
        half : OUT std_logic
    );
END COMPONENT;
signal clk : std_logic := '1';
signal Start : std_logic := '0';
signal Stop : std_logic := '0';
signal Pauza : std_logic := '0';
signal Load : std_logic := '0';
signal Reset : std_logic := '0';
signal startValue : unsigned (7 downto 0) := "00001111";
signal output : unsigned(7 downto 0);
signal half : std_logic;
BEGIN
    uut: Licznik_vhdlModule PORT MAP (
        clk => clk,
        Start => Start,
        Stop => Stop,
        Pauza => Pauza,
        Load => Load,
        Reset => Reset,
        startValue => startValue,
        output => output,
        half => half
    );
    clk <= not clk after 20 ns;
    Load <= '1' after 0 ns, '0' after 100 ns, '1' after 320 ns;
    startValue <= "11001100" after 300 ns;
    Reset <= '1' after 260 ns, '0' after 720 ns, '1' after 980 ns;
    Start <= '1' after 560 ns, '0' after 580 ns, '1' after 1020 ns;
    Stop <= '1' after 900 ns;
END;

```

Listing 2 Pobudzenia dla linii sterujących licznikiem

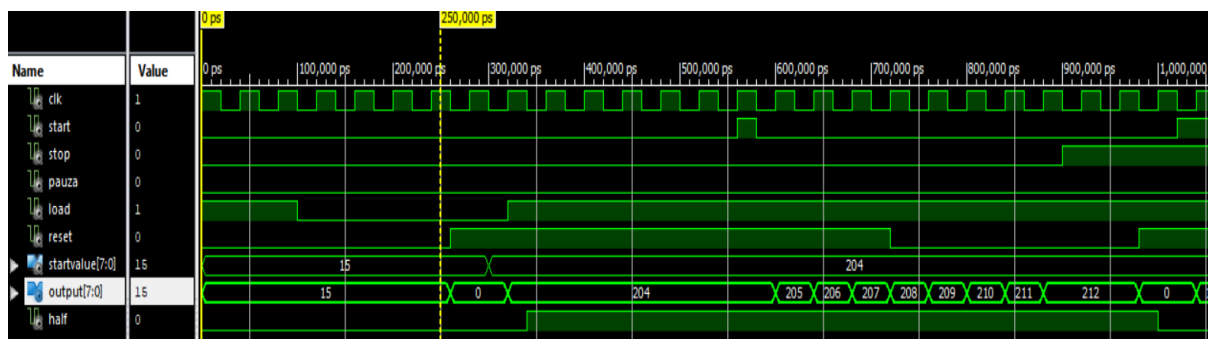
3. Symulacja

Jednym z wymagań przedstawionych w instrukcji do laboratorium było zapewnienie możliwości śledzenia stanu licznika w trakcie jego pracy. W tym celu w trakcie przeprowadzania symulacji należy nacisnąć strzałkę znajdująca się obok wektora przechowującego stan licznika. Po naciśnięciu strzałki możliwe jest sprawdzenie poszczególnych bitów wektora, w celu ułatwienia obserwacji stanu licznika dla człowieka obserwującego symulację zdecydowaliśmy się też ustawić wyświetlanie stanu licznika w systemie unsigned decimal. Rezultaty zastosowanie tych opcji umieszczone są na poniższym obrazku.



Rysunek 1 Sprawdzanie bieżącego stanu licznika

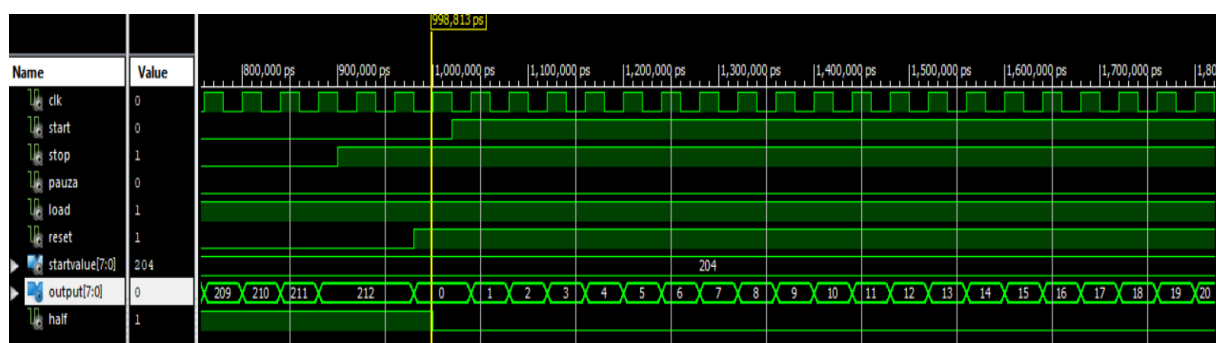
Na poniższym obrazku możemy zaobserwować wpływ linii sterujących na zachowanie licznika. Pierwszą linią sterującą, która została użyta była linia LOAD służąca do załadowania wartości początkowej zliczania. Kolejną wywołaną linią była linia RESET, której skuteczne działanie możemy zauważyć na przeprowadzonej symulacji. Następnie ponownie użyta jest linia LOAD ładująca nową wartość początkową. Jako, że nowa wartość początkowa przekracza połowę zakresu zliczania to zasygnalizowane to jest na linii HALF. Następną użytą linią sterowania jest linia START, po użyciu której możemy zauważyć, że licznik rozpoczął zliczanie. Następnie na obrazku widoczne jest zastosowanie linii STOP zatrzymujące działanie licznika i ponowne użycie linii RESET.



Rysunek 2 Wpływ linii sterujących na zachowanie licznika, część 1

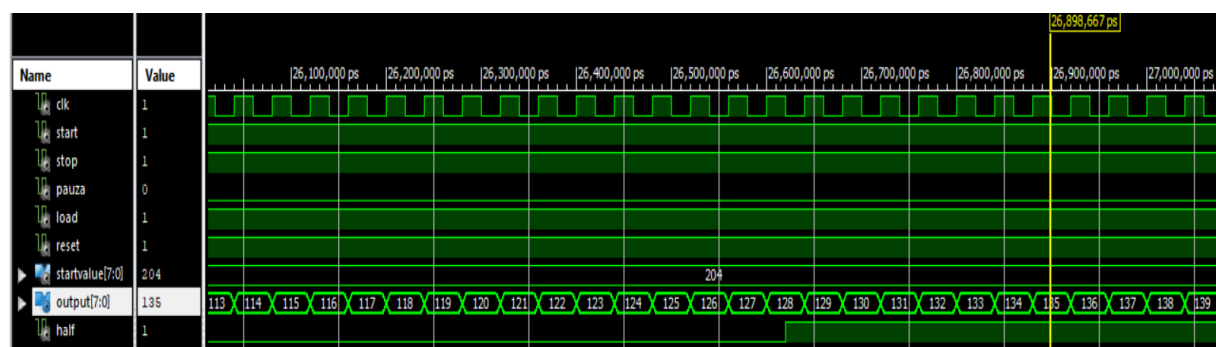
Na poniższym obrazku, będącym kontynuacją poprzedniego obrazka możemy zauważyć, że ponowne użycie linii START rozpoczyna działanie zresetowanego i zatrzymanego wcześniej licznika. Jednocześnie możemy zauważyć też, że po resecie zliczania linia HALF sygnalizująca

przekroczenie połowy zakresu zliczania przyjęła wartość '0'.



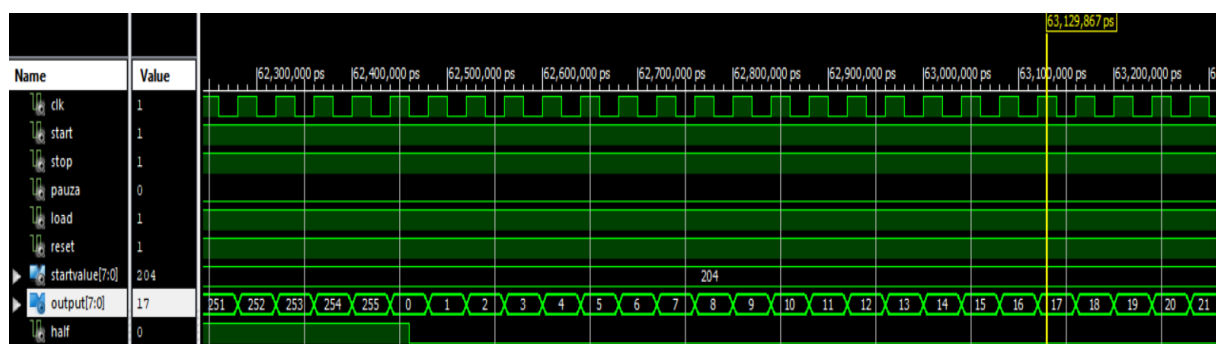
Rysunek 3 Wpływ linii sterujących na zachowanie licznika, część 2

Na załączonym poniżej obrazku możemy zaobserwować przekroczenie połowy zakresu zliczania i sygnalizację tego faktu przy pomocy linii HALF.



Rysunek 4 Przekroczenie połowy zakresu zliczania

Na poniższym obrazku przedstawione jest przekroczenie całego zakresu zliczania i ponowne rozpoczęcie zliczania od wartości 0.



Rysunek 5 Przekroczenie zakresu zliczania

4. Wnioski

W trakcie zajęć laboratoryjnych udało nam się w pełni zrealizować przedstawione zadanie. Kluczowa dla naszej implementacji licznika okazała się biblioteka IEEE.NUMERIC_STD.ALL pozwalająca przeprowadzać operacje arytmetyczne na wektorach. Przeprowadzone przez nas testy symulacyjne działania układy pokazały, że wszystkie linie sterujące układu działają zgodnie z oczekiwaniami. Ponownie bardzo pomocne okazało się środowisko Xilinx ISE pozwalające na elastyczne modyfikowanie systemu liczbowego w jakim przedstawiane są wyniki symulacji. Ułatwiło to nam śledzenie bieżącego stanu wykonanego licznika.