

Rozbudowane projekty łączące maszyny stanów oraz moduły WE/WY (PS/2, RS232, etc.)

1 Treść zadania

Zadaniem do wykonania jest licznik rewersyjny 8-bitowy (00-FF) wyświetlany na 7-segmentowym, 4-cyfrowym wyświetlaczu. W dodatku z klawiatury PS/2 można wczytać znak, od którego licznik startuje działanie.

2 Wykonanie

2.1 Licznik

Wartość sygnału, w zależności od kierunku jest inkrementowana lub dekrementowana. Jeśli licznik dochodzi do wartości granicznych, czyli FF dla licznika idącego "w górę" wartość sygnału jest ustawiana na 00 oraz dla wartości 00 dla licznika idącego "w dół" na wartość FF . Aby móc wykonywać operacje inkrementacji i dekrementacji należy odpowiednio rzutować wektor na typ *unsigned*, dokonywać odpowiedniej operacji, a następnie rzutować na typ *std_logic_vector*.

2.2 Podłączenie klawiatury PS/2

Do podłączenia na płytę użyty został moduł *PS2_RX*, który podłączyliśmy z wygenerowanym symbolem zaprojektowanego licznika. Warto zauważyć, że moduł klawiatury oraz moduł licznika korzystają z jednego zegara *CLK_XT*, a moduł Display4x7S z zegara *CLK_LF*.

3 Pliki VHDL

3.1 Licznik

Plik VHDL przedstawiający logikę licznika 8-bitowego. Wszystko zawarte zostało w jednym procesie.

```
1 entity licznik is
2 Port(
3   Y : OUT std_logic_vector(15 downto 0);
4   blank : OUT std_logic_vector(3 downto 0) := "1100";
5   DIR : in STD_LOGIC;
6   CLK : in STD_LOGIC;
7   D0 : in std_logic_vector(7 downto 0);
8   CE : in std_logic
9 );
10 end licznik;
11 architecture Behavioral of licznik is
12 signal Ytemp : std_logic_vector(15 downto 0) := X"00" & D0;
13
14 begin
15 process1 : process(CLK, DIR, CE)
16 begin
17   if rising_edge(CLK) then
18     if CE = '1' then
19       Ytemp <= X"00" & D0;
20     else
21       if DIR = '0' then
22         if Ytemp = X"00FF" then
```

```
23         Ytemp <= X"0000";
24     else
25         Ytemp <= std_logic_vector(unsigned(Ytemp) + 1);
26     end if;
27 else
28     if Ytemp = X"0000" then
29         Ytemp <= X"00FF";
30     else
31         Ytemp <= std_logic_vector(unsigned(Ytemp) - 1);
32     end if;
33 end if;
34 end if;
35 end if;
36 end process process1;
37 Y <= Ytemp;
38 end Behavioral;
```

3.2 Plik .ucf

W pliku szablonie ZL-9572.ucf odkomentowano linijki odpowiadające kolejno za zegary (jeden dla licznika, drugi klawiatury), przyciski, wyświetlacz 7-segmentowy oraz klawiaturę PS/2.

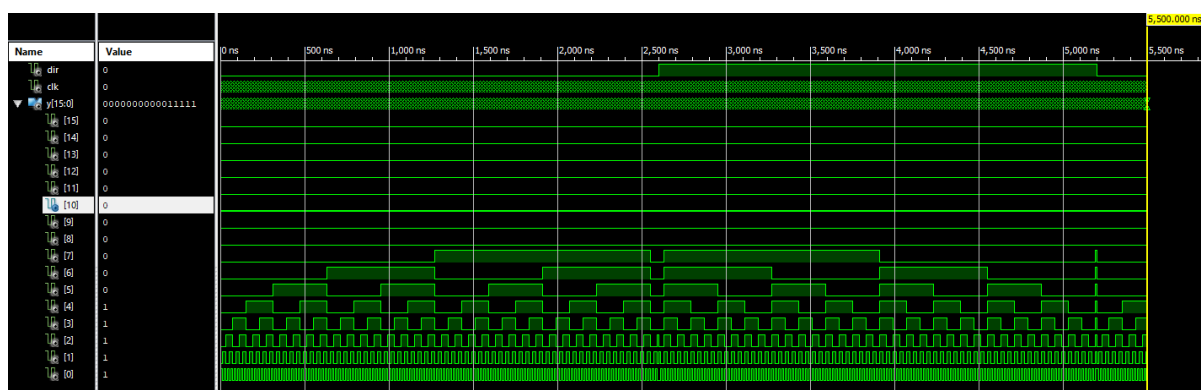
```
1 # Clocks
2 NET "CLK_LF" LOC = "P7" | BUFG = CLK;
3 NET "CLK_LF" PERIOD = 5ms HIGH 50%;
4
5 NET "CLK_XT" LOC = "P5" | BUFG = CLK;
6 NET "CLK_XT" PERIOD = 500ns HIGH 50%;
7
8 # Keys
9 NET "DIR" LOC = "P42";
10 NET "RESET" LOC = "P40";
11 #NET "Key<2>" LOC = "P43";
12 #NET "Key<3>" LOC = "P38";
13 NET "DP(0)" LOC = "P37";
14 NET "DP(1)" LOC = "P36"; # shared with ROT_A
15 NET "DP(2)" LOC = "P24"; # shared with ROT_B
16 NET "DP(3)" LOC = "P39"; # GSR
17
18 # DISPL. 7-SEG
19 NET "DOTOUT(0)" LOC = "P8" | SLEW = "SLOW";
20 NET "DOTOUT(1)" LOC = "P6" | SLEW = "SLOW";
21 NET "DOTOUT(2)" LOC = "P4" | SLEW = "SLOW";
22 NET "DOTOUT(3)" LOC = "P9" | SLEW = "SLOW";
23 NET "MAINOUT(0)" LOC = "P12"; # Seg. A; shared with LED<10>
24 NET "MAINOUT(1)" LOC = "P13"; # Seg. B; shared with LED<8>
25 NET "MAINOUT(2)" LOC = "P22"; # Seg. C; shared with LED<12>
26 NET "MAINOUT(3)" LOC = "P19"; # Seg. D; shared with LED<14>
27 NET "MAINOUT(4)" LOC = "P14"; # Seg. E; shared with LED<15>
28 NET "MAINOUT(5)" LOC = "P11"; # Seg. F; shared with LED<9>
29 NET "MAINOUT(6)" LOC = "P20"; # Seg. G; shared with LED<13>
30 NET "MAINOUT(7)" LOC = "P18"; # Seg. DP; shared with LED<11>
31
32 # PS/2
33 NET "PS2_Clk" LOC = "P3";
34 NET "PS2_Data" LOC = "P2";
```

4 Testbench

Dla licznika, jeszcze przed zawarciem elementów potrzebnych do wczytywania znaków z klawiatury oraz wyświetlania liczb na wyświetlaczu 7-segmentowym, wykonana została symulacja behawioralna aby sprawdzić poprawność procesu. Testbench przedstawia się następująco:

```
1
2 LIBRARY ieee;
3 USE ieee.std_logic_1164.ALL;
4 ENTITY tescik_licznik_2 IS
5 END tescik_licznik_2;
6
7 ARCHITECTURE behavior OF tescik_licznik_2 IS
8     COMPONENT licznik
9     PORT (
10         Y : OUT std_logic_vector(15 downto 0);
11         blank : OUT std_logic_vector(3 downto 0);
12         DIR : IN std_logic;
13         CLK : IN std_logic
14     );
15     END COMPONENT;
16     signal DIR : std_logic := '0';
17     signal CLK : std_logic := '0';
18     signal Y : std_logic_vector(15 downto 0);
19     signal blank : std_logic_vector(3 downto 0);
20     constant CLK_period : time := 10 ns;
21 BEGIN
22     uut: licznik PORT MAP (
23         Y => Y,
24         blank => blank,
25         DIR => DIR,
26         CLK => CLK
27     );
28     DIR <= '0', '1' after 2600 ns, '0' after 5200 ns;
29     CLK_process : process
30     begin
31         CLK <= '0';
32         wait for CLK_period/2;
33         CLK <= '1';
34         wait for CLK_period/2;
35     end process;
36     stim_proc: process
37     begin
38         wait for 100 ns;
39         wait for CLK_period*10;
40         wait;
41     end process;
42 END;
```

Symulacja przebiegła pomyślnie:



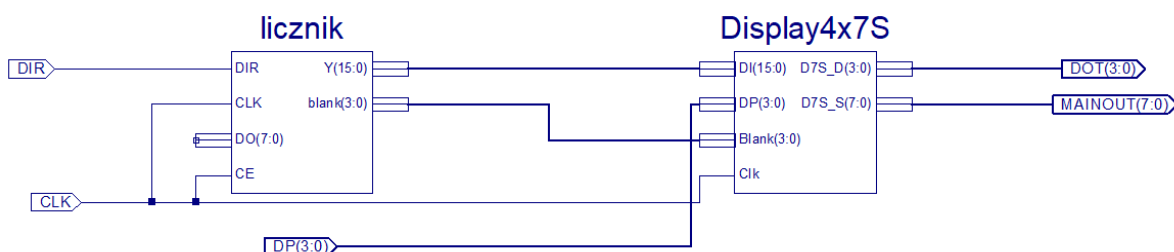
Symulacja behawioralna licznika 8-bitowego dwukierunkowego

Licznik poprawnie zapętlą się i reaguje na wejście DIR.

5 Projekt hierarchiczny

5.1 Licznik z wyświetlaczem

Po napisaniu licznika w języku VHDL, następnym krokiem było połączenie go z wyświetlaczem 7-segmentowym. Użyty został do tego gotowy moduł Display4x7S. Oba komponenty połączone zostały na schemacie w następujący sposób:

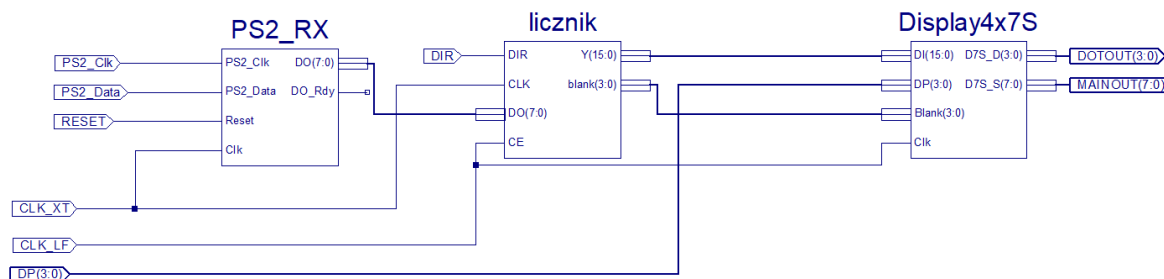


Schemat połączenia licznika z wyświetlaczem 7-segmentowym

Czas wykonywania został dobrany w ten sposób, aby było widać zapętlenie licznika w obie strony. Starsze 8 bitów wyjścia to zera, gdyż wyświetlacz przyjmuje 16-bitowe wejście (później dwie starsze cyfry są wyłączane, wyświetlane są tylko dwie młodsze).

5.2 Licznik z wyświetlaczem i klawiaturą

Po przeprowadzeniu sprawdzenia poprawności działania licznika na płycie, ostatnim elementem było dołączenie klawiatury PS/2 i zezwolenie na sterowanie za jej pomocą. Do poprzedniego schematu został dołożony moduł PS2_RX. Sposób połączenia pokazuje schemat:



Schemat połączenia licznika z wyświetlaczem 7-segmentowym i klawiaturą PS/2

6 Wgranie na płytę ZL-9572

Po wgraniu na płytę samego licznika sprawdzone zostały funkcjonalności, podobnie jak w testbenchu. Jednak, żeby przepływ był zauważalny trzeba było przesunąć przełączniki zegara na płycie. Po tej operacji dostrzegalne było przekręcenie licznika. Dzięki zadeklarowaniu wyjścia blank na 1100 dwie pierwsze cyfry zostały wyłączone. Kropki po cyfrach wyświetlacza były obsługiwane przyciskami 4-7.

Po zrobieniu projektu hierarchicznego, program również został wgrany na płytę. Jednak problemem było to, że układ nie reagował odpowiednio na klawiaturę. Zamiast startować od danej liczby, przyciski na klawiaturze działały jako "Clock Enable", tzn. musiał być wciśnięty dowolny przycisk, aby licznik działał.

7 Wnioski

Podczas laboratorium udało się sprawnie zrealizować układ licznika za pomocą inkrementacji i dekrementacji wartości sygnału. Użycie modułu *Display4x7S* nie sprawiło problemów. Z braku czasu nie udało nam się naprawić usterki związanej z użyciem klawiatury jako sygnału *CE*, zamiast wartości startowej licznika.