Maciej Byczko	Prowadzący:	Numer ćwiczenia
Bartosz Matysiak		
DN 10.50 TD	Temat ćwiczenia:	Ocensi
F N 10.50 1F	dr inż. Jacek Mazurkiewicz 4 Temat ćwiczenia: Ocena:	Ocena:
Grupa:	Data wykonania:	
В	8 Listopada 2021r.	

Spis treści

1.1 Polecenie 1.2 Rozwiązanie 1.2.1 Schemat stanów 1.2.2 Schematy układu 1.2.3 Kod VHDL 1.2.4 Symulacja 1.3 Fizyczna implementacja 1.3.1 Kod UCF 2 Zadanie 2 2.1 Polecenie 2.2 Rozwiązanie 2.3 Uwagi wstępne 2.3.1 Tabele prawdy 2.3.2 Siatki Karnaugh 2.3.3 Schemat układu 2.3.4 Kod VHDL 2.3.5 Symulacja 2.4 Fizyczna implementacja 2.4.1 Kod UCF 3 Zadanie 3 3.1 Polecenie 3.2 Rozwiązanie 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3.1 Kod UCF	Za	ndanie 1
1.2.1 Schemat stanów 1.2.2 Schematy układu 1.2.3 Kod VHDL 1.2.4 Symulacja 1.3 Fizyczna implementacja 1.3.1 Kod UCF 2	1.1	Polecenie
1.2.2 Schematy układu 1.2.3 Kod VHDL 1.2.4 Symulacja 1.3 Fizyczna implementacja 1.3.1 Kod UCF 2 Zadanie 2 2.1 Polecenie 2.2 Rozwiązanie 2.3 Uwagi wstępne 2.3.1 Tabele prawdy 2.3.2 Siatki Karnaugh 2.3.3 Schemat układu 2.3.4 Kod VHDL 2.3.5 Symulacja 2.4 Fizyczna implementacja 2.4.1 Kod UCF 3 Zadanie 3 3.1 Polecenie 3.2 Rozwiązanie 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 2.3.1 Kod UCF 3 Zadanie 3 3.1 Polecenie 3.2 Rozwiązanie 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF 4 Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.2 Tabela prawdy 4.2.1 Schemat stanów 4.2.1 Schemat stanów 4.2.2 Tabela prawdy	1.2	Rozwiązanie
1.2.3 Kod VHDL 1.2.4 Symulacja 1.3 Fizyczna implementacja 1.3.1 Kod UCF 2 Zadanie 2 2.1 Polecenie 2.2 Rozwiązanie 2.3 Uwagi wstępne 2.3.1 Tabele prawdy 2.3.2 Siatki Karnaugh 2.3.3 Schemat układu 2.3.4 Kod VHDL 2.3.5 Symulacja 2.4 Fizyczna implementacja 2.4.1 Kod UCF 3 Zadanie 3 3.1 Polecenie 3.2 Rozwiązanie 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF		1.2.1 Schemat stanów
1.2.3 Kod VHDL 1.2.4 Symulacja 1.3 Fizyczna implementacja 1.3.1 Kod UCF 2 Zadanie 2 2.1 Polecenie 2.2 Rozwiązanie 2.3 Uwagi wstępne 2.3.1 Tabele prawdy 2.3.2 Siatki Karnaugh 2.3.3 Schemat układu 2.3.4 Kod VHDL 2.3.5 Symulacja 2.4 Fizyczna implementacja 2.4.1 Kod UCF 3 Zadanie 3 3.1 Polecenie 3.2 Rozwiązanie 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF		1.2.2 Schematy układu
1.2.4 Symulacja 1.3 Fizyczna implementacja 1.3.1 Kod UCF 2 Zadanie 2 2.1 Polecenie 2.2 Rozwiązanie 2.3 Uwagi wstępne 2.3.1 Tabele prawdy 2.3.2 Siatki Karnaugh 2.3.3 Schemat układu 2.3.4 Kod VHDL 2.3.5 Symulacja 2.4 Fizyczna implementacja 2.4.1 Kod UCF 3 Zadanie 3 3.1 Polecenie 3.2 Rozwiązanie 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.7 Fizyczna implementacja 3.8 Fizyczna implementacja 3.9 Fizyczna implementacja 3.1 Kod UCF		
1.3.1 Kod UCF		
1.3.1 Kod UCF	1.3	B Fizyczna implementacja
2.1 Polecenie 2.2 Rozwiązanie 2.3 Uwagi wstępne 2.3.1 Tabele prawdy 2.3.2 Siatki Karnaugh 2.3.3 Schemat układu 2.3.4 Kod VHDL 2.3.5 Symulacja 2.4 Fizyczna implementacja 2.4.1 Kod UCF 3 Zadanie 3 3.1 Polecenie 3.2 Rozwiązanie 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF 4 Polecenie 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.1 Schemat stanów 4.2.2 Tabela prawdy		· · ·
2.2 Rozwiązanie 2.3 Uwagi wstępne 2.3.1 Tabele prawdy 2.3.2 Siatki Karnaugh 2.3.3 Schemat układu 2.3.4 Kod VHDL 2.3.5 Symulacja 2.4 Fizyczna implementacja 2.4.1 Kod UCF Zadanie 3 3.1 Polecenie 3.2 Rozwiązanie 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.2 Tabela prawdy	Za	adanie 2
2.3 Uwagi wstępne 2.3.1 Tabele prawdy 2.3.2 Siatki Karnaugh 2.3.3 Schemat układu 2.3.4 Kod VHDL 2.3.5 Symulacja 2.4 Fizyczna implementacja 2.4.1 Kod UCF Zadanie 3 3.1 Polecenie 3.2 Rozwiązanie 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.2 Tabela prawdy	2.1	Polecenie
2.3 Uwagi wstępne 2.3.1 Tabele prawdy 2.3.2 Siatki Karnaugh 2.3.3 Schemat układu 2.3.4 Kod VHDL 2.3.5 Symulacja 2.4 Fizyczna implementacja 2.4.1 Kod UCF Zadanie 3 3.1 Polecenie 3.2 Rozwiązanie 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.2 Tabela prawdy	2.2	Rozwiązanie
2.3.1 Tabele prawdy 2.3.2 Siatki Karnaugh 2.3.3 Schemat układu 2.3.4 Kod VHDL 2.3.5 Symulacja 2.4 Fizyczna implementacja 2.4.1 Kod UCF	2.3	3 Uwagi wstępne
2.3.2 Siatki Karnaugh 2.3.3 Schemat układu 2.3.4 Kod VHDL 2.3.5 Symulacja 2.4 Fizyczna implementacja 2.4.1 Kod UCF 3 Zadanie 3 3.1 Polecenie 3.2 Rozwiązanie 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF 4 Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.2 Tabela prawdy		· · · ·
2.3.3 Schemat układu 2.3.4 Kod VHDL 2.3.5 Symulacja 2.4 Fizyczna implementacja 2.4.1 Kod UCF 3 Zadanie 3 3.1 Polecenie 3.2 Rozwiązanie 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF 4 Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.2 Tabela prawdy		- v
2.3.4 Kod VHDL 2.3.5 Symulacja 2.4 Fizyczna implementacja 2.4.1 Kod UCF 8 Zadanie 3 3.1 Polecenie 3.2 Rozwiązanie 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF 1 Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.2 Tabela prawdy		
2.3.5 Symulacja 2.4 Fizyczna implementacja 2.4.1 Kod UCF 8 Zadanie 3 3.1 Polecenie 3.2 Rozwiązanie 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF 8 Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.2 Tabela prawdy		
2.4 Fizyczna implementacja 2.4.1 Kod UCF Zadanie 3 3.1 Polecenie 3.2 Rozwiązanie 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.2 Tabela prawdy		
2.4.1 Kod UCF Zadanie 3 3.1 Polecenie 3.2 Rozwiązanie 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.2 Tabela prawdy	2.4	<i>y</i>
3.1 Polecenie 3.2 Rozwiązanie 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.2 Tabela prawdy	_,_	v i
3.2 Rozwiązanie . 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie . 4.2.1 Schemat stanów 4.2.2 Tabela prawdy	Za	adanie 3
3.2 Rozwiązanie . 3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie . 4.2.1 Schemat stanów 4.2.2 Tabela prawdy		
3.2.1 Schemat stanów 3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.2 Tabela prawdy	3.2	
3.2.2 Tabela prawdy 3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.2 Tabela prawdy		·
3.2.3 Siatki Karnaugh 3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.2 Tabela prawdy		
3.2.4 Schemat układu 3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.2 Tabela prawdy		I I
3.2.5 Kod VHDL 3.2.6 Symulacja 3.3 Fizyczna implementacja 3.3.1 Kod UCF Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.2 Tabela prawdy		0
3.2.6 Symulacja		
3.3 Fizyczna implementacja 3.3.1 Kod UCF Zadanie 4 4.1 Polecenie 4.2 Rozwiązanie 4.2.1 Schemat stanów 4.2.2 Tabela prawdy		
3.3.1 Kod UCF 4 Zadanie 4 4.1 Polecenie	3.3	
4.1 Polecenie	3.3	
4.2 Rozwiązanie	Za	adanie 4
4.2 Rozwiązanie	4.1	Polecenie
4.2.1 Schemat stanów 4.2.2 Tabela prawdy		
4.2.2 Tabela prawdy	-	
1 ,		
		1 /

\mathbf{W}_{1}	nioski																
	4.3.1	Kod UCF .			 •						•		•	•			•
4.3	Fizycz	na implemen	tacja														
	4.2.6	Symulacja .															
		Kod VHDL															
	4.2.4	Schemat uk	ładu														

Sprawozdanie

Strona 2

1 Zadanie 1

1.1 Polecenie

Detektor 2-znakowej sekwencji słów 8-bitowych: wejścia 2 znaków 8-bitowych, 1 wyjście 1-bitowe – sekwencja rozpoznana / sekwencja błędna. Źródło danych: początkowo "guziki" przystawki, potem klawiatura PC via terminal.

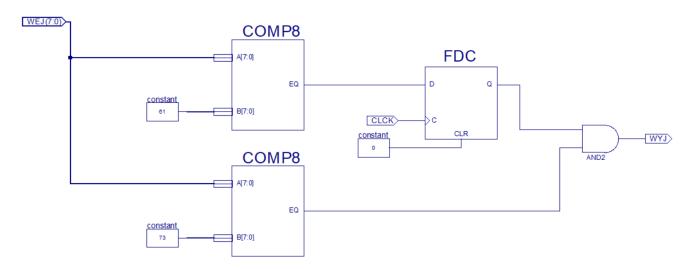
1.2 Rozwiązanie

Do rozwiązania problemu wymagane jest od nas podłączenie dwóch komparatorów 8-bitowych (COMP8), które po pobraniu wartości od użytkownika kolejno po sobie sprawdzają wprowadzone słowa. W zależności od wymagania można wprowadzić także obowiązkowe wprowadzanie wartości w odpowiedniej kolejności.

1.2.1 Schemat stanów

1.2.2 Schematy układu

Schemat dla wersji z przyciskami jako inputem:



Schemat wersji wykorzystującej klawiaturę jako wejście:

1.2.3 Kod VHDL

```
LIBRARY ieee;
  USE ieee.std_logic_1164.ALL;
  USE ieee.numeric_std.ALL;
  LIBRARY UNISIM;
  USE UNISIM. V components. ALL;
  ENTITY scheme scheme sch tb IS
  END scheme_scheme_sch_tb;
  ARCHITECTURE behavioral OF scheme_scheme_sch_tb IS
9
     COMPONENT scheme
10
                       STD_LOGIC_VECTOR (7 DOWNTO 0);
     PORT( WEJ : IN
            WYJ : OUT STD_LOGIC;
                  : IN
                         STD LOGIC);
            CLCK
13
     END COMPONENT;
14
```

```
15
      SIGNAL WEJ: STD_LOGIC_VECTOR (7 DOWNIO 0);
16
      SIGNAL WYJ : STD_LOGIC;
17
      SIGNAL CLCK : STD_LOGIC := '0';
18
19
   BEGIN
20
21
      UUT: scheme PORT MAP(
22
       WEJ \implies WEJ,
       WYJ \implies WYJ,
       CLCK \Rightarrow CLCK
25
      );
26
27
     CLCK <= not CLCK after 50 ns;
28
     WEJ <= B"0000_0000", B"0110_0001" after 200 ns, B"0010_0011" after
29
         300 ns, B"0110_0001" after 400 ns, B"0111_0011" after 500 ns;
  END;
```

1.2.4 Symulacja



1.3 Fizyczna implementacja

1.3.1 Kod UCF

Kod dla wersji z przyciskami jako inputem:

```
# Clocks
  NET "CLCK" LOC = "P7" | BUFG = CLK | PERIOD = 5ms HIGH 50%;
  # Keys
  NET "WEJ(0)" LOC = "P42";
  NET "WEJ(1)" LOC = "P40";
  NET "WEJ(2)" LOC = "P43";
  NET "WEJ(3)" LOC = "P38";
  NET "WEJ(4)" LOC = "P37";
  NET "WEJ(5)" LOC = "P36";
                             # shared with ROT_A
  NET "WEJ(6)" LOC = "P24";
                             # shared with ROT B
  NET "WEJ(7)" LOC = "P39";
                             # GSR
13
  # LEDS
14
  NET "WYJ"
            LOC = "P35";
```

Schemat wersji wykorzystującej klawiaturę jako wejście:

2 Zadanie 2

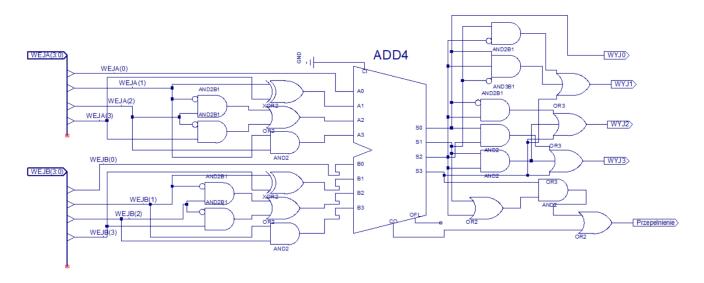
2.1 Polecenie

Układ arytmetyczny pracujący na dwóch argumentach 4-bitowych wyrażonych w kodzie Aikena i generujący stosowny wynik w tymże kodzie.

2.2 Rozwiązanie

2.3 Uwagi wstępne

- 2.3.1 Tabele prawdy
- 2.3.2 Siatki Karnaugh
- 2.3.3 Schemat układu



2.3.4 Kod VHDL

```
LIBRARY ieee;
  USE ieee.std_logic_1164.ALL;
  USE ieee.numeric_std.ALL;
  LIBRARY UNISIM;
  USE UNISIM. Vcomponents. ALL;
  ENTITY aikenAdderScheme_aikenAdderScheme_sch_tb IS
  END aikenAdderScheme aikenAdderScheme sch tb;
  ARCHITECTURE behavioral OF aikenAdderScheme_aikenAdderScheme_sch_tb
     IS
9
     COMPONENT aikenAdderScheme
     PORT( WYJ0 : OUT STD LOGIC;
11
                         STD_LOGIC_VECTOR (3 DOWNTO 0);
             WEJA
                   : IN
12
                         STD_LOGIC_VECTOR (3 DOWNTO 0);
                   : IN
13
             Przepelnienie : OUT STD_LOGIC;
14
                   : OUT STD LOGIC;
             WYJ1
15
                   : OUT STD_LOGIC;
             WYJ2
16
             WYJ3
                   : OUT STD LOGIC);
     END COMPONENT;
```

```
19
      SIGNAL WYJ0 : STD_LOGIC;
20
                     : STD_LOGIC_VECTOR (3 DOWNTO 0);
      SIGNAL WEJA
21
                     : STD LOGIC VECTOR (3 DOWNTO 0);
      SIGNAL WEJB
      SIGNAL Przepelnienie : STD_LOGIC;
                     : STD LOGIC;
      SIGNAL WYJ1
24
      SIGNAL WYJ2
                      : STD LOGIC;
25
      SIGNAL WYJ3
                      : STD_LOGIC;
26
   BEGIN
28
29
      UUT: aikenAdderScheme PORT MAP(
30
       WYJ0 \implies WYJ0,
31
       WEJA \implies WEJA,
32
       WEJB \implies WEJB,
33
        Przepelnienie => Przepelnienie,
34
       WYJ1 \implies WYJ1,
       WYJ2 \implies WYJ2,
36
       WYJ3 \implies WYJ3
37
      );
38
39
     \label{eq:WEJA} W\!E\!J\!A <= "0000", "0010" \ after \ 100 \ ns \,, "0010" \ after \ 200 \ ns \,, "1011"
40
         after 300 ns, "0010" after 400 ns, "1011" after 500 ns, "1111"
         after 600 ns;
     \label{eq:WEJB} <= "0010", "0001" after 100 ns, "0100" after 200 ns, "0011"
41
         after 300 ns, "1101" after 400 ns, "1100" after 500 ns, "1111"
         after 600 ns;
42
  END;
43
```

2.3.5 Symulacja



2.4 Fizyczna implementacja

2.4.1 Kod UCF

```
# Keys
NET "WEJA(0)" LOC = "P42";
NET "WEJA(1)" LOC = "P40";
NET "WEJA(2)" LOC = "P43";
NET "WEJA(3)" LOC = "P38";
NET "WEJB(0)" LOC = "P37";
NET "WEJB(1)" LOC = "P36"; # shared with ROT_A
NET "WEJB(2)" LOC = "P24"; # shared with ROT_B
NET "WEJB(3)" LOC = "P39"; # GSR
```

```
10
11 # LEDS
12 NET "WYJ0" LOC = "P35";
13 NET "WYJ1" LOC = "P29";
14 NET "WYJ2" LOC = "P33";
15 NET "WYJ3" LOC = "P34";
16 NET "Przepelnienie" LOC = "P28";
```

3 Zadanie 3

3.1 Polecenie

Konwerter cyfry szesnastkowej zapisanej na czterech bitach od 0 do 9, A do F na kod ASCII tej cyfry – wyjście 8-bitowe. Prezentacja wyniku na diodach przystawki, potem na wyświetlaczu 7-segmentowym.

3.2 Rozwiązanie

- 3.2.1 Schemat stanów
- 3.2.2 Tabela prawdy
- 3.2.3 Siatki Karnaugh
- 3.2.4 Schemat układu
- 3.2.5 Kod VHDL
- 3.2.6 Symulacja

3.3 Fizyczna implementacja

3.3.1 Kod UCF

4 Zadanie 4

4.1 Polecenie

Komparator dwóch 4-bitowych cyfr: 2 wejścia po 4 bity, 3 wyjścia 1-bitowe: mniejszy, większy, równy pracujący w kodzie Aikena.

4.2 Rozwiązanie

- 4.2.1 Schemat stanów
- 4.2.2 Tabela prawdy
- 4.2.3 Siatki Karnaugh
- 4.2.4 Schemat układu
- 4.2.5 Kod VHDL
- 4.2.6 Symulacja
- 4.3 Fizyczna implementacja
- 4.3.1 Kod UCF
- 5 Wnioski