LABORATORIUM **Teoria Automatów**	
Temat Ćwiczenia: **Wykorzystanie Technologii CPLD do projektowania układów logicznych z użyciem funkcji EX-OR**	
Grupa laboratoryjna: **1a, wtorek 11^{00}**	
L.p	Nazwisko i Imię
1 2 3	**Aleksandrowicz Maciej** **Krzyszczuk Michał** **Marczewski Marcin**
Data wykonania ćwiczenia : **28.11.2017r**	

Spis treści

---

1) Wstęp teoretyczny

Cyfrowe układy scalone są wytwarzane w dwóch zasadniczych kategoriach: układy standardowe(do uniwersalnych zastosowań), oraz jako układy specjalizowane. Przykładem takich układów są rozbudowane programowalne układy elektroniczne CPLD (ang. Complex Programmable Logic Device).Układy CPLD posiadają architekturę hierarchiczną opartą na makrokomórkach logicznych.

Funkcja Ex-OR (ang. Exclusive OR) nazywana inaczej alternatywą wykluczającą (lub sumą modulo 2), to dwuargumentowa funkcja boolowska, której wynik jest prawdą wtedy i tylko wtedy, gdy dokładnie jedno ze zdań jest prawdziwe.

Kod Graya(kod refleksyjny) – dwójkowy kod, który charakteryzuje się tym, że dwa kolejne słowa kodowe różnią się tylko stanem jednego bitu. Jest to ważna cecha, która stawia kod Graya jako alternatywę dla kodu binarnego na przykład w realizacji sterowania ramieniem robota, wykorzystującego czujnik pomiaru kąta, przy zastosowaniu kodu binarnego, a bity z czujnika nie zmieniają się równocześnie - układ pomiarowy ma luzy. Z tego powodu w pewnych sytuacjach mogą pojawić się nieprawidłowe dane z czujnika.

2) Zadania do wykonania

Ćwiczenie składało się z trzech zadań do wykonania:

- Zrealizować konwerter 4 bitowego kodu binarnego na 4-bitowy kod Graya
- Zrealizować konwerter 4 bitowego kodu Graya na 4-bitowy kod binarny
- Wykorzystać wyświetlacz 7 segmentowy do wyświetlenia wartości liczbowej 4-bitowego kodu binarnego otrzymanego z konwersji 4 bitowego kodu Graya

3) Analiza teoretyczna

Do dwóch pierwszych ćwiczeń będzie potrzebne zestawienie kodu Graya oraz kodu binarnego. Zostało ono zaprezentowane w poniższej tabeli 1:

B4	***B3***	***B2***	***B1***	***G4***	***G3***	***G2***	***G1***
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	1	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Tabela 1 *- Zestawienie 4 bitowego kodu binarnego i kodu Graya*

Na podstawie tabeli 1 dokonano analizy wyjść poszczególnych bitów przy konwersji z **kodu binarnego** w **kod Graya**:

G0				
B1 B0 \ B3 B2	*00*	*01*	*11*	*10*
00	0	0	0	0
01	**1**	**1**	**1**	**1**
11	0	0	0	0
10	**1**	**1**	**1**	**1**

Tabela 2 - Wyjście ***G0*** konwertera

Funkcję logiczną stanu wyjścia ***G1*** wyprowadzono z pomocą metody rozłącznych pokryć:

G0				
B1 B0 \ B3 B2	*00*	*01*	*11*	*10*
00	0	0	0	0
01	**1**	**1**	**1**	**1**
11	0	0	0	0
10	**1**	**1**	**1**	**1**

Tabela 3 - Metoda rozłącznych pokryć dla wyjścia ***G0***

$G0 = \neg B1 \wedge B0 \oplus \neg B0 \wedge B1$ //wzór (11) z instrukcji

$G0 = B0 \oplus (B1 \wedge B0) \oplus B1 \oplus (B1 \wedge B0)$ //wzór (7) z instrukcji

$G0 = B0 \oplus B1$

G1				
B1 B0 \ B3 B2	*00*	*01*	*11*	*10*
00	0	**1**	**1**	0
01	0	**1**	**1**	0
11	**1**	0	0	**1**
10	**1**	0	0	**1**

Tabela 4 - Wyjście ***G1*** konwertera

Funkcję logiczną stanu wyjścia ***G1*** wyprowadzono z pomocą metody rozłącznych pokryć:

G1				
B1 B0 \ B3 B2	*00*	*01*	*11*	*10*
00	0	**1**	**1**	0
01	0	**1**	**1**	0
11	**1**	0	0	**1**
10	**1**	0	0	**1**

Tabela 5 - Metoda rozłącznych pokryć dla wyjścia ***G1***

$G1 = \neg B1 \wedge B2 \oplus \neg B2 \wedge B1$ //wzór (11) z instrukcji

$G1 = B2 \oplus (B1 \wedge B2) \oplus B1 \oplus (B1 \wedge B2)$ //wzór (7) z instrukcji

$G1 = B1 \oplus B2$

G2				
B1 B0 \ B3 B2	*00*	*01*	*11*	*10*
00	0	**1**	0	**1**
01	0	**1**	0	**1**
11	0	**1**	0	**1**
10	0	**1**	0	**1**

Tabela 6 - Wyjście ***G2*** konwertera

Funkcję logiczną stanu wyjścia ***G2*** wyprowadzono z pomocą metody rozłącznych pokryć:

G2				
B1 B0 \ B3 B2	*00*	*01*	*11*	*10*
00	0	**1**	0	**1**
01	0	**1**	0	**1**
11	0	**1**	0	**1**
10	0	**1**	0	**1**

Tabela 7 - Metoda rozłącznych pokryć dla wyjścia ***G2***

$G2 = \neg B3 \wedge B2 \oplus \neg B2 \wedge B3$ //wzór (11) z instrukcji

$G2 = B2 \oplus (B3 \wedge B2) \oplus B3 \oplus (B3 \wedge B2)$ //wzór (7) z instrukcji

$G2 = B2 \oplus B3$

G3				
B1 B0 \ B3 B2	*00*	*01*	*11*	*10*
00	0	0	**1**	**1**
01	0	0	**1**	**1**
11	0	0	**1**	**1**
10	0	0	**1**	**1**

Tabela 8 - Wyjście ***G3*** konwertera

Funkcję logiczną stanu wyjścia ***G3*** wyprowadzono z pomocą metody rozłącznych pokryć:

G3				
B1 B0 \ B3 B2	*00*	*01*	*11*	*10*
00	0	0	**1**	**1**
01	0	0	**1**	**1**
11	0	0	**1**	**1**
10	0	0	**1**	**1**

Tabela 9 - Metoda rozłącznych pokryć dla wyjścia ***G3***

$$G3 = B3$$

Tym samym otrzymano równania pozwalające na konwersję każdego z czterech bitów kodu binarnego na kod Graya:

1. $G0 = B0 \oplus B1$
2. $G1 = B1 \oplus B2$
3. $G2 = B2 \oplus B3$
4. $G3 = B3$

Konwerter 4 bitowego kodu Graya na 4-bitowy kod binarny

Na podstawie tabeli 1 dokonano analizy wyjść poszczególnych bitów przy konwersji z **kodu binarnego** w **kod Graya**:

B0				
G1 G0 \ G3 G2	*00*	*01*	*11*	*10*
00	0	**1**	0	**1**
01	**1**	0	**1**	0
11	0	**1**	0	**1**
10	**1**	0	**1**	0

Tabela 10 - Wyjście ***B0*** konwertera

Funkcję logiczną stanu wyjścia ***B0*** wyprowadzono z wielomianowej postaci kanonicznej funkcji EXOR:

$$
\begin{aligned}
B0 \ = \ & a_0 \oplus a_1 G0 \oplus a_2 G1 \oplus a_3 G2 \oplus a_4 G3 \oplus a_5 G0 \wedge G1 \oplus a_6 G0 \wedge G2 \oplus \\
& a_7 G0 \wedge G3 \oplus a_8 G1 \wedge G2 \oplus a_9 G1 \wedge G3 \oplus a_{10} G2 \wedge G3 \oplus \\
& a_{11} G0 \wedge G1 \wedge G2 \oplus a_{12} G0 \wedge G1 \wedge G3 \oplus a_{13} G0 \wedge G2 \wedge G3 \oplus \\
& a_{14} G2 \wedge G3 \wedge G4 \oplus a_{15} G1 \wedge G2 \wedge G3 \wedge G4
\end{aligned}
$$

G3	***G2***	***G1***	***G0***	***Wyznaczenie współczynników***
0	0	0	0	$a_0 = 0$
0	0	0	1	$0 \oplus a_1 = 1 \rightarrow a_1 = 1$
0	0	1	0	$0 \oplus a_2 = 1 \rightarrow a_2 = 1$
0	1	0	0	$0 \oplus a_3 = 1 \rightarrow a_3 = 1$
1	0	0	0	$0 \oplus a_4 = 1 \rightarrow a_4 = 1$
0	0	1	1	$0 \oplus 1 \oplus 1 \oplus a_5 = 0 \rightarrow a_5 = 0$
0	1	0	1	$0 \oplus 1 \oplus 1 \oplus a_6 = 0 \rightarrow a_6 = 0$

1	0	0	1	$0 \oplus 1 \oplus 1 \oplus a_7 = 0 \rightarrow a_7 = 0$
0	1	1	0	$0 \oplus 1 \oplus 1 \oplus a_8 = 0 \rightarrow a_8 = 0$
1	0	1	0	$0 \oplus 1 \oplus 1 \oplus a_9 = 0 \rightarrow a_9 = 0$
1	1	0	0	$0 \oplus 1 \oplus 1 \oplus a_{10} = 0 \rightarrow a_{10} = 0$
0	1	1	1	$0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus a_{11} = 1 \rightarrow a_{11} = 0$
1	0	1	1	$0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus a_{12} = 1 \rightarrow a_{12} = 0$
1	1	0	1	$0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus a_{13} = 1 \rightarrow a_{13} = 0$
1	1	1	0	$0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus a_{14} = 1 \rightarrow a_{14} = 0$
1	1	1	1	$0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus a_{15} = 1 \rightarrow a_{15} = 0$

Tabela 11 **-** Odczytanie z tabeli 10 współczynników funkcji logicznej wyjścia ***B0***

Na podstawie tabeli 11 uzupełniono współczynniki funkcji logicznej:

$$B0 = G0 \oplus G1 \oplus G2 \oplus G3$$

B1				
G1 G0 \ G3 G2	*00*	*01*	*11*	*10*
00	0	**1**	0	**1**
01	0	**1**	0	**1**
11	**1**	0	**1**	0
10	**1**	0	**1**	0

Tabela 12 - Wyjście ***B1*** konwertera

Funkcję logiczną stanu wyjścia ***B1*** wyprowadzono z pomocą metody rozłącznych pokryć:

B1				
G1 G0 \ G3 G2	*00*	*01*	*11*	*10*
00	0	**1**	0	**1**
01	0	**1**	0	**1**
11	**1**	0	**1**	0
10	**1**	0	**1**	0

Tabela 13 - Metoda rozłącznych pokryć dla wyjścia ***B1***

$$B1 = (G1 \wedge \neg G2 \wedge \neg G3) \oplus (\neg G1 \wedge G2 \wedge \neg G3) \oplus$$
$$(G1 \wedge G2 \wedge G3) \oplus (\neg G1 \wedge \neg G2 \wedge G3)$$

$$(G1 \wedge \neg G2 \wedge \neg G3) = (G1 \oplus G1 \wedge G2) \wedge \neg G3 =$$
$$(G1 \oplus G1 \wedge G2) \oplus G3(G1 \oplus G1 \wedge G2) =$$
$$G1 \oplus G1 \wedge G2 \oplus G1 \wedge G3 \oplus G1 \wedge G2 \wedge G3$$

$$(\neg G1 \wedge G2 \wedge \neg G3) = (G2 \oplus G1 \wedge G2) \wedge \neg G3 =$$
$$(G2 \oplus G1 \wedge G2) \oplus (G2 \oplus G1 \wedge G2)G3 =$$
$$G2 \oplus G1 \wedge G2 \oplus G2 \wedge G3 \oplus G1 \wedge G2 \wedge G3$$

$$(\neg G1 \wedge \neg G2 \wedge G3) = \neg G1 \wedge (G3 \oplus G2 \wedge G3) =$$
$$(G3 \oplus G2 \wedge G3) \oplus G1(G3 \oplus G2 \wedge G3) =$$
$$G3 \oplus G2 \wedge G3 \oplus G1 \wedge G3 \oplus G1 \wedge G2 \wedge G3$$

$$B1 = G1 \oplus (G1 \wedge G2) \oplus (G1 \wedge G3) \oplus (G1 \wedge G2 \wedge G3) \oplus$$
$$G2 \oplus (G1 \wedge G2) \oplus (G2 \wedge G3) \oplus (G1 \wedge G2 \wedge G3) \oplus (G1 \wedge G2 \wedge G3) \oplus$$
$$G3 \oplus (G2 \wedge G3) \oplus (G1 \wedge G3) \oplus (G1 \wedge G2 \wedge G3)$$

$$(G1 \wedge G2) \oplus (G1 \wedge G2) = 0$$

$$B1 = G1 \oplus (G1 \wedge G3) \oplus (G1 \wedge G2 \wedge G3) \oplus G2 \oplus (G2 \wedge G3) \oplus (G1 \wedge G2 \wedge G3) \oplus$$
$$(G1 \wedge G2 \wedge G3) \oplus G3 \oplus (G2 \wedge G3) \oplus (G1 \wedge G3) \oplus (G1 \wedge G2 \wedge G3)$$

$$(G1 \wedge G3) \oplus (G1 \wedge G3) = 0$$

$$B1 = G1 \oplus (G1 \wedge G2 \wedge G3) \oplus G2 \oplus (G2 \wedge G3) \oplus (G1 \wedge G2 \wedge G3) \oplus$$
$$(G1 \wedge G2 \wedge G3) \oplus G3 \oplus (G2 \wedge G3) \oplus (G1 \wedge G2 \wedge G3)$$

$$(G1 \wedge G2 \wedge G3) \oplus (G1 \wedge G2 \wedge G3) = 0$$

$$B1 = G1 \oplus G2 \oplus (G2 \wedge G3) \oplus (G1 \wedge G2 \wedge G3) \oplus G3 \oplus (G2 \wedge G3) \oplus$$
$$(G1 \wedge G2 \wedge G3)$$

$$(G2 \wedge G3) \oplus (G2 \wedge G3) = 0$$

$$B1 = G1 \oplus G2 \oplus (G1 \wedge G2 \wedge G3) \oplus G3 \oplus (G1 \wedge G2 \wedge G3)$$

$$(G1 \wedge G2 \wedge G3) \oplus (G1 \wedge G2 \wedge G3) = 0$$

$$B1 = G1 \oplus G2 \oplus G3$$

B2				
G1 G0 \ G3 G2	*00*	*01*	*11*	*10*
00	0	**1**	0	**1**
01	0	**1**	0	**1**
11	0	**1**	0	**1**
10	0	**1**	0	**1**

Tabela 14 - Wyjście ***B2*** konwertera

Funkcję logiczną stanu wyjścia ***B2*** wyprowadzono z pomocą metody rozłącznych pokryć:

B2				
G1 G0 \ G3 G2	*00*	*01*	*11*	*10*
00	0	**1**	0	**1**
01	0	**1**	0	**1**
11	0	**1**	0	**1**
10	0	**1**	0	**1**

Tabela 15 - Metoda rozłącznych pokryć dla wyjścia ***B2***

$$B2 = (G2 \wedge \neg G3) \oplus (\neg G2 \wedge G3)$$
$$B2 = G2 \oplus G2 \wedge G3 \oplus G3 \oplus G2 \wedge G3$$
$$B2 = G2 \oplus G3$$

B3				
G1 G0 \ G3 G2	*00*	*01*	*11*	*10*
00	0	0	**1**	**1**
01	0	0	**1**	**1**
11	0	0	**1**	**1**
10	0	0	**1**	**1**

Tabela 16 - Wyjście ***B3*** konwertera

Funkcję logiczną stanu wyjścia ***B3*** wyprowadzono z pomocą metody rozłącznych pokryć:

B3				
G1 G0 \ G3 G2	*00*	*01*	*11*	*10*
00	0	0	**1**	**1**
01	0	0	**1**	**1**
11	0	0	**1**	**1**
10	0	0	**1**	**1**

Tabela 17 - Metoda rozłącznych pokryć dla wyjścia ***B3***

$$B3 = G3$$

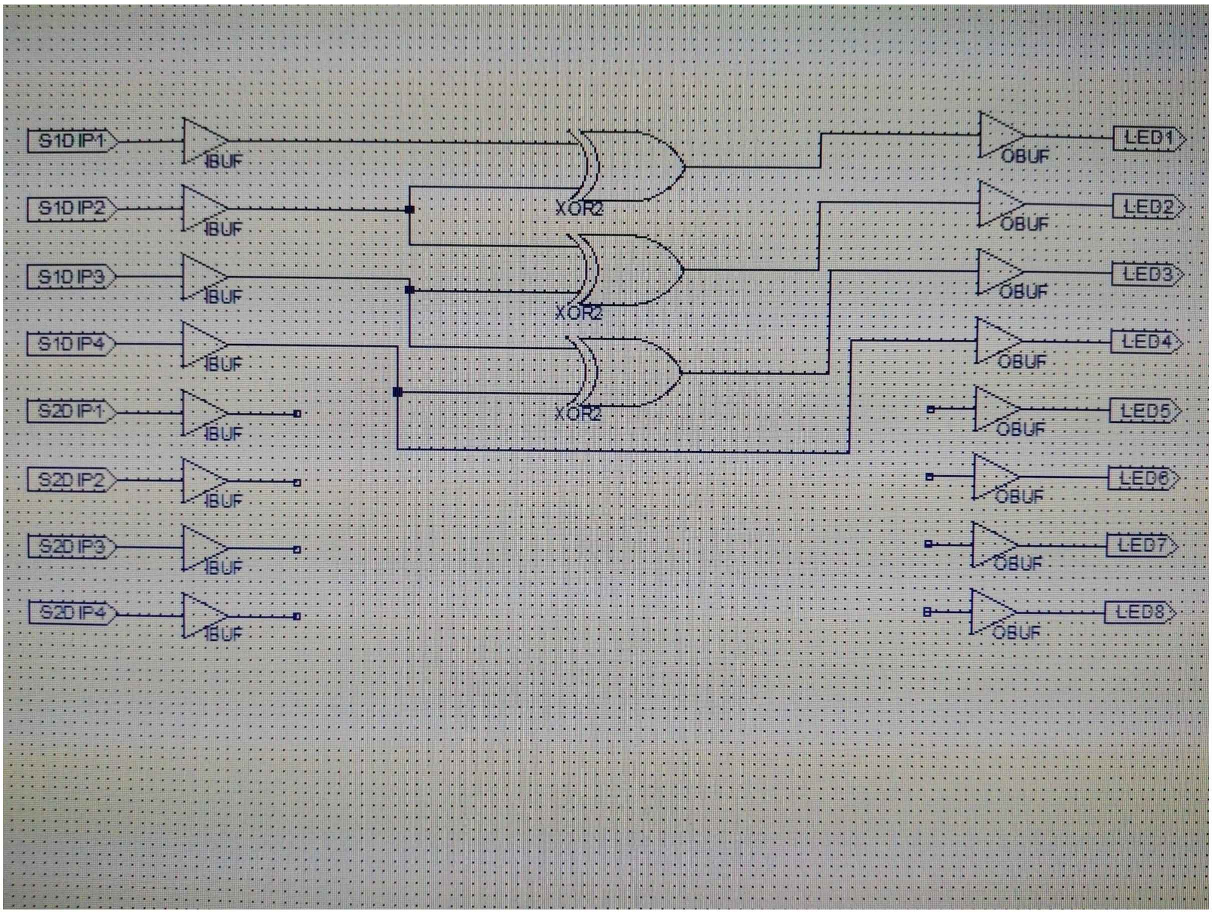
Tym samym otrzymano równania pozwalające na konwersję każdego z czterech bitów kodu Graya na kod binarny:

1. $B0 = G0 \oplus G1 \oplus G2 \oplus G3$
2. $B1 = G1 \oplus G2 \oplus G3$
3. $B2 = G2 \oplus G3$
4. $B3 = G3$

4) Sposób realizacji zadania w praktyce

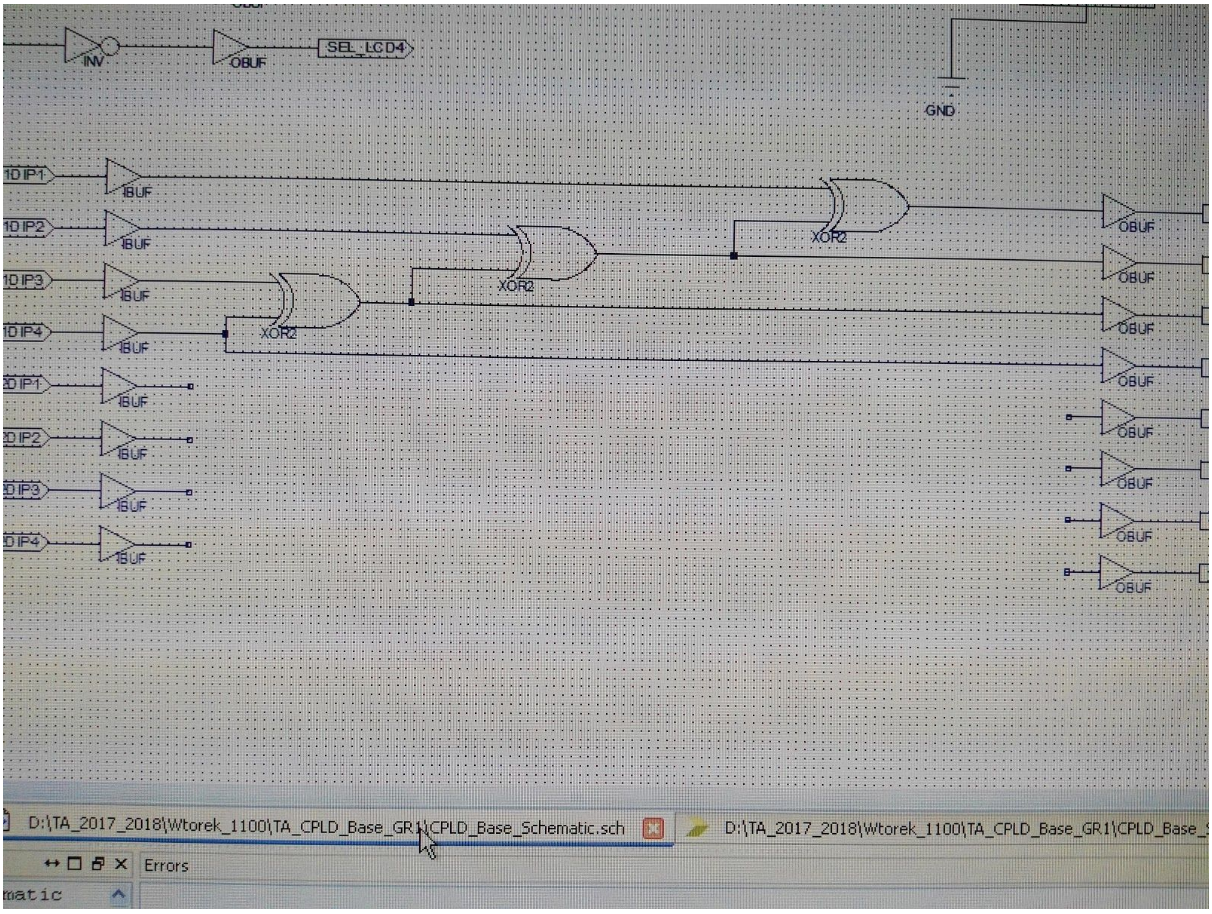
Po przeprowadzeniu analizy teoretycznej zapoznano się z instrukcją podłączenia układu CPLD do komputera oraz programu do konfiguracji tego połączenia. Następnie grupa upewniła się, że układ jest poprawnie podłączony oraz czy istnieje możliwość zaprogramowania go programem bazowym.

Przystąpiono do realizacji pierwszego ćwiczenia, czyli zbudowania 4 bitowego konwertera kodu binarnego na kod Graya.



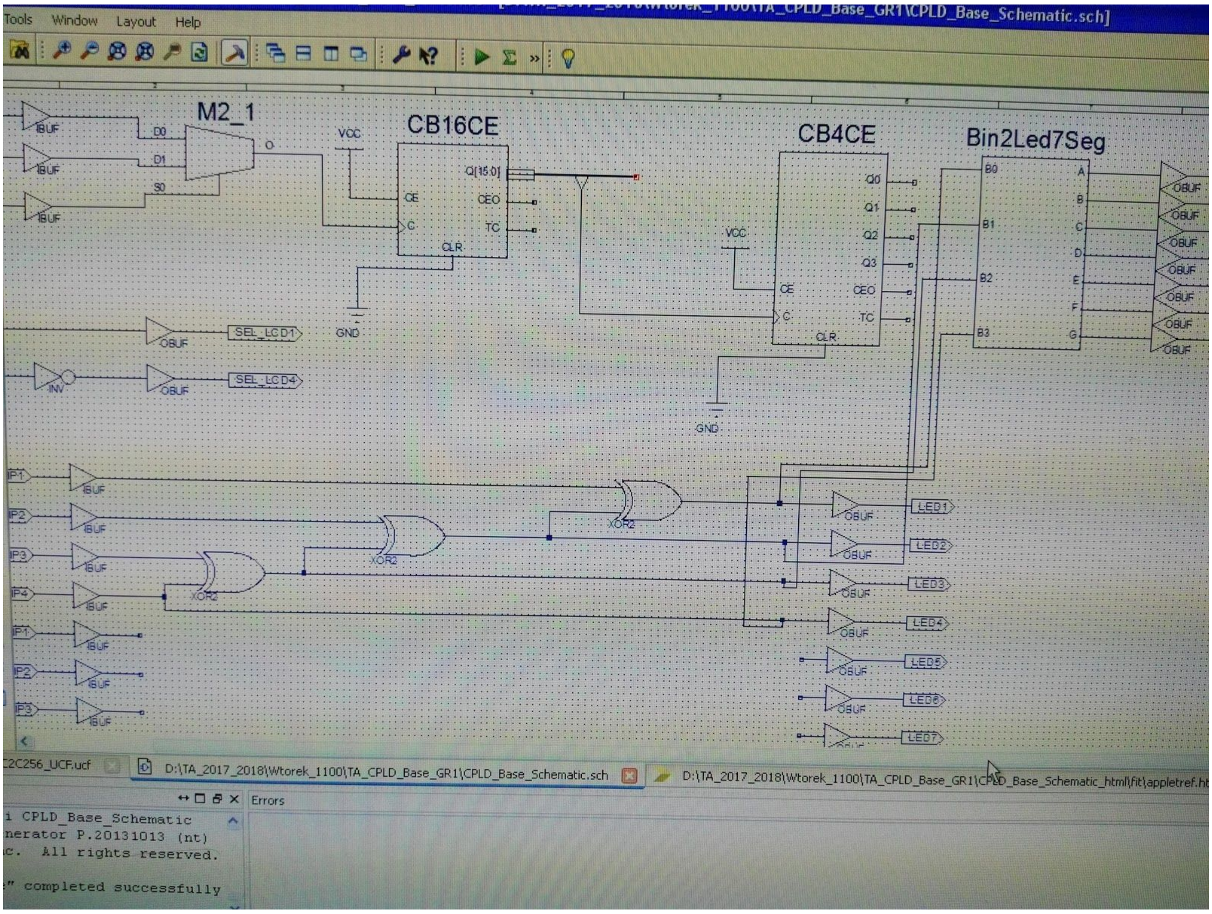
Zdjęcie 1 - Układ logiczny konwertera BIN->GRAY

Kolejnym ćwiczeniem było zbudowania konwertera odwrotnego, to znaczy kodu Graya na kod binarny.

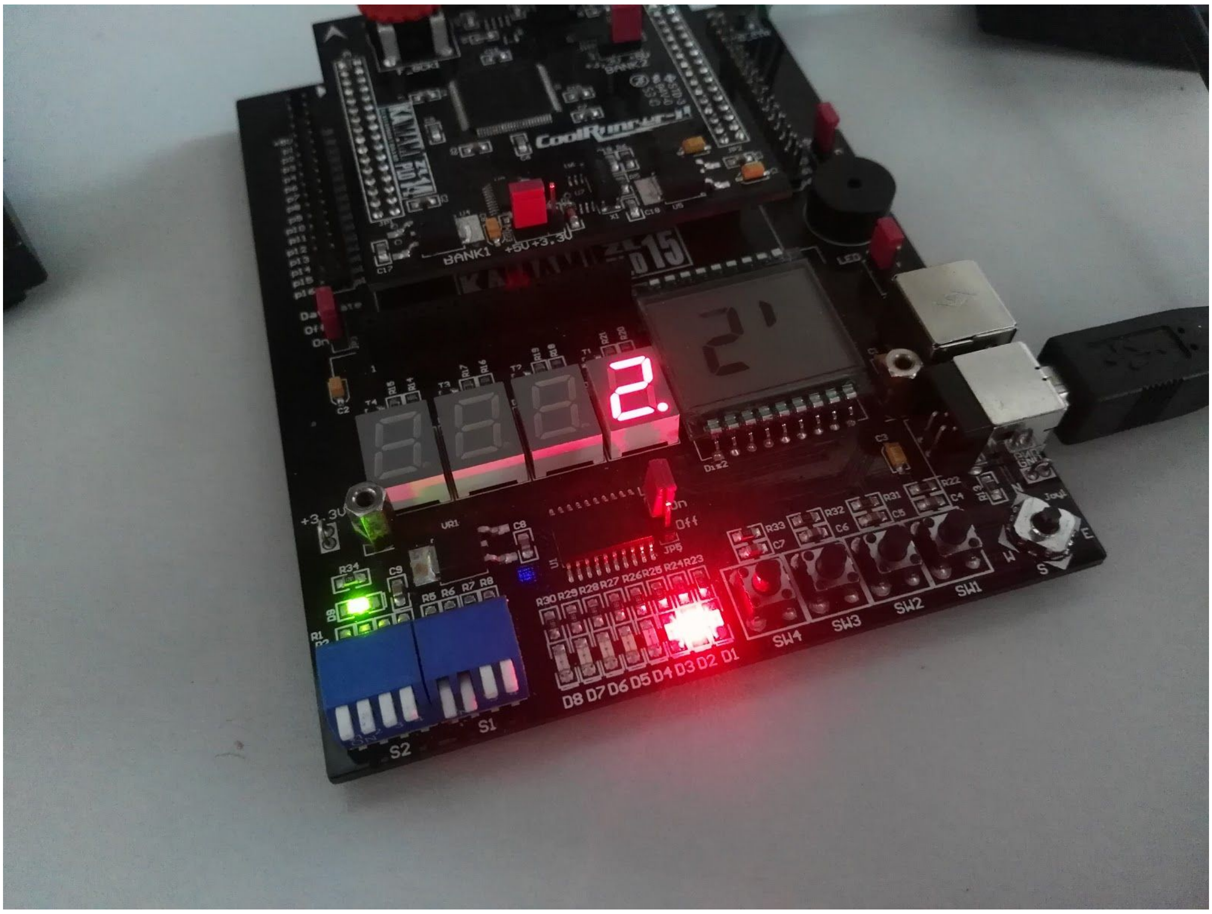


Zdjęcie 2 - Układ logiczny konwertera GRAY->BIN

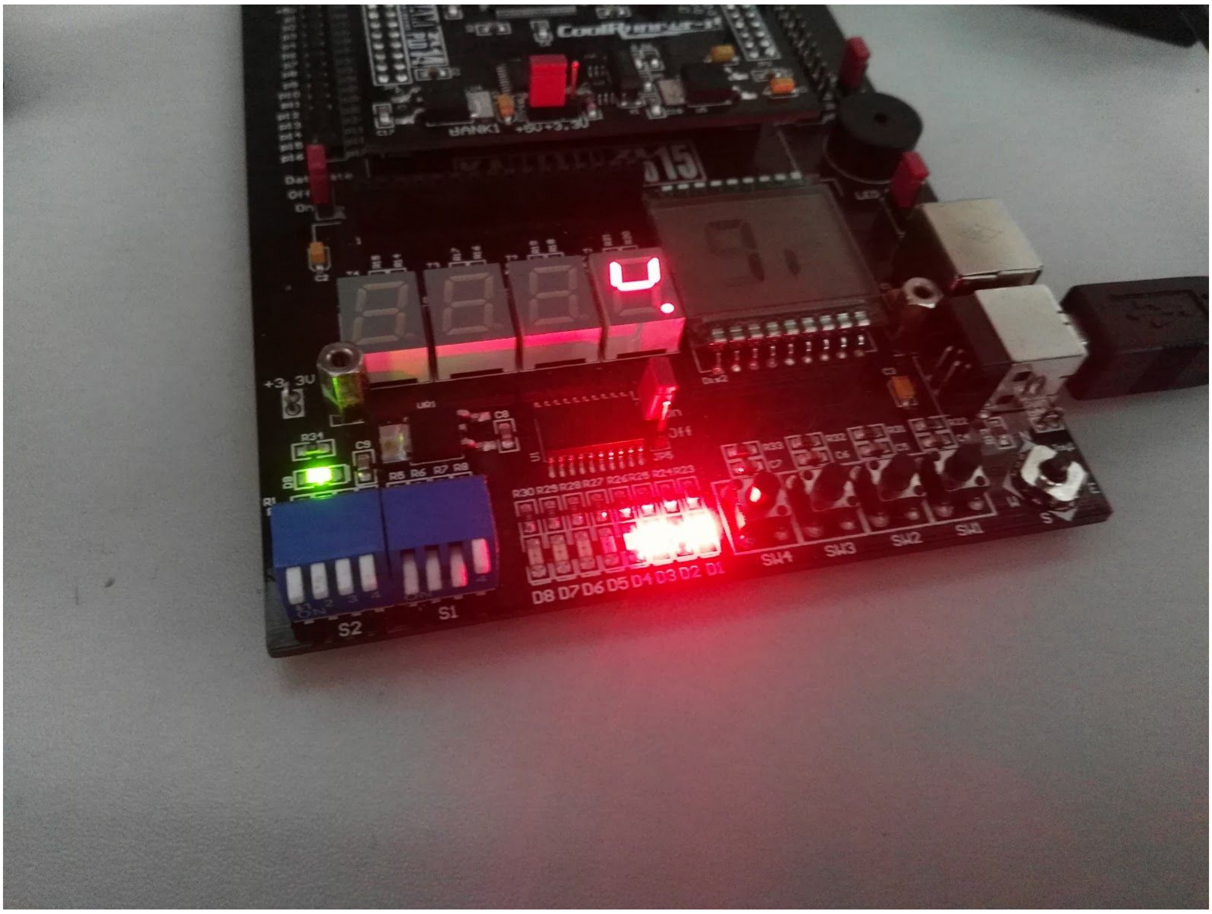
Ostatnim ćwiczeniem było podłączenie wyjścia konwertera kodu Graya na kod binarny do wyświetlacza siedmiosegmentowego. W tym celu wykorzystano gotowy moduł sterownika wyświetlacza, przyjmującego na wejściu kod binarny. Po zaprogramowaniu układu okazało się, że wyświetlacz siedmiosegmentowy nie działa w pełni poprawnie.



Zdjęcie 3 - Podłączenie wyjść konwertera GRAY->BIN do kontrolera wyświetlacza siedmiosegmentowego



Zdjęcie 4 **-** Poprawnie zdekodowana i wyświetlona liczba 2



Zdjęcie 5 - Błędnie wyświetlana wartość liczbowa

5) Wnioski

- Algebra boolowska funkcji logicznych może być bardzo czasochłonna. Czas poświęcony na analizę teoretyczną jest jednak kompensowany przez znacząco prostsze funkcje logiczne.
- Konwersję kodu binarnergo na kod Greya (oraz vice versa) można realizować w prosty sposób poprzez odpowiednie połączenia takiej samej ilości bramek EXOR, co ilość bitów danego kodu.
- Układ CPLD udostępniony na zajęciach laboratoryjnych nie był w pełni sprawny.