Laboratorium Teorii Automatów	
Wykorzystanie technologii CPLD do projektowania układów logicznych z użyciem funkcji Ex-OR	
Grupa 4b (wtorek 17.15)	Sonia Wittek, Katarzyna Wątorska, Bartłomiej Mróz

Wstęp teoretyczny

Celem naszego ćwiczenia było zaprojektowanie układów cyfrowych z wykorzystaniem programowanych układów logicznych typu CPLD firmy XILINX. Układy kombinacyjne zrealizowaliśmy głownie za pomocą funkcji logicznej EX-OR.

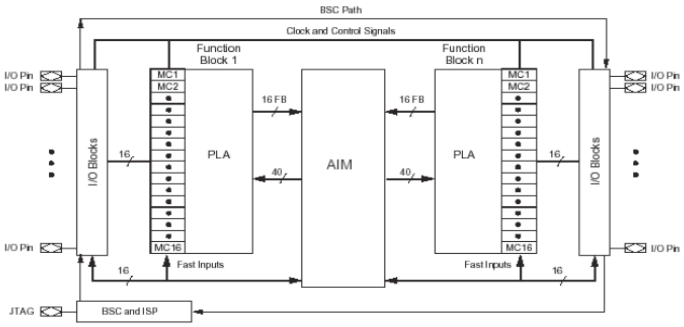
Układy cyfrowe możemy podzielić na dwie zasadnicze kategorie:

- układy standardowe (projektowane do uniwersalnych zastosowań);
- układy specjalizowane (projektowane w większym lub mniejszym stopniu indywidualnie).

Układy standardowe to ogólnie dostępne, nieprogramowalne układy o standaryzowanych oznaczeniach typów, wykonywane w różnych technologiach i często przez wielu wytwórców.

Układy specjalizowane (ASIC), do których należą między innymi układy CPLD, są natomiast wytwarzane lub programowalne indywidualnie, zgodnie z potrzebami wytwórców sprzętu.

Elektroniczne układy CPLD (Complex Programmable Logic Device) to układy wytwarzane i sprzedawane jak układy standardowe, lecz ich użycie w konkretnym celu wymaga wcześniejszego zaprogramowania funkcji przez użytkownika. CPLD posiadają architekturę hierarchiczną opartą na makrokomórkach logicznych. Pojemność takiego układu właśnie od liczby tych makrokomórek a ich liczba waha się od kilkudziesięciu do kilkuset. Z reguły te komórki połączone są w większe bloki funkcjonalne, które mogą realizować funkcje kombinacyjne jak i sekwencyjne. Wiele takich bloków jest zaś połączona za pomocą matrycy połączeniowej kluczy, której zdolność łączeniowa określa, w jakim stopniu układ jest programowalny.



PLA – Programmable Logic Array, **AIM** – Advanced Interconnect Matrix, **MC** - MacroCell

Rys. 1. Schemat blokowy układu CPLD rodziny XC2Cxxx

Funkcja EX-OR nazywana jest alternatywą wykluczającą lub sumą mod 2. Jest to dwuargumentowa funkcja boolowska, której wynik jest prawdą tylko wtedy gdy dokładnie jedno ze zdań jest prawdziwe. Aby wyznaczyć funkcję logiczne zrealizowane przez nas w układzie CPLD skorzystaliśmy z poniższych tożsamości:

$$0 \oplus 0 = 0$$
$$1 \oplus 1 = 0$$
$$0 \oplus 1 = 1$$
$$0 \oplus A = A$$
$$1 \oplus A = \bar{A}$$
$$A \oplus \bar{A} = 1$$
$$A \oplus A = 0$$
$$A \oplus B = B \oplus A$$
$$A \oplus \bar{B} = \bar{A} \oplus B = \overline{A \oplus B}$$
$$\overline{A \oplus B} = 1 \oplus A \oplus B$$
$$A\bar{B} = A \oplus AB = A(1 \oplus B)$$
$$\overline{A\bar{B}} = 1 \oplus A \oplus B \oplus AB$$
$$A \oplus B = A\bar{B} \vee \bar{A}B$$
$$\overline{A \oplus B} = AB \vee \overline{A}\overline{B}$$
$$A \vee B = A \oplus B \oplus AB = A \vee (A \oplus B)$$
$$A \oplus (B \oplus C) = AB \oplus AC$$
$$A \oplus (B \oplus C) = (A \oplus B) \oplus C = A \oplus B \oplus C$$
$$\bar{A} \oplus B \oplus C = A \oplus \bar{B} \oplus C = A \oplus B \oplus \bar{C} = \overline{A \oplus B \oplus C} = 1 \oplus A \oplus B \oplus C$$
$$\bar{A} \oplus \bar{B} \oplus C = A \oplus \bar{B} \oplus \bar{C} = \bar{A} \oplus B \oplus \bar{C} = A \oplus B \oplus C$$
$$\underbrace{A \oplus A \oplus \ldots \oplus A}_{n \text{ razy}} = \begin{cases} A & \text{dla } n \text{ nieparzystych} \\ 0 & \text{dla } n \text{ parzystych} \end{cases}$$

Przebieg laboratorium

Zadanie 1

Korzystając z funkcji logicznej Ex-OR zrealizowano prostą logikę: konwerter 4-ro bitowego kodu binarnego na 4-ro bitowy kod Grey'a.

Kod binarny

L.p.	B3	B2	B1	B0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Kod Grey'a

L.p.	G3	G2	G1	G0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1
14	1	0	0	1
15	1	0	0	0

Projektowanie dla G3:

B1B0\B3B2	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	1	1
10	0	0	1	1

$$G_3 = B_3$$

Projektowanie dla G1:

B1B0\B3B2	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	1	0	0	1
10	1	0	0	1

$$G_1 = B_2 \oplus B_1$$

Projektowanie dla G2:

B1B0\B3B2	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

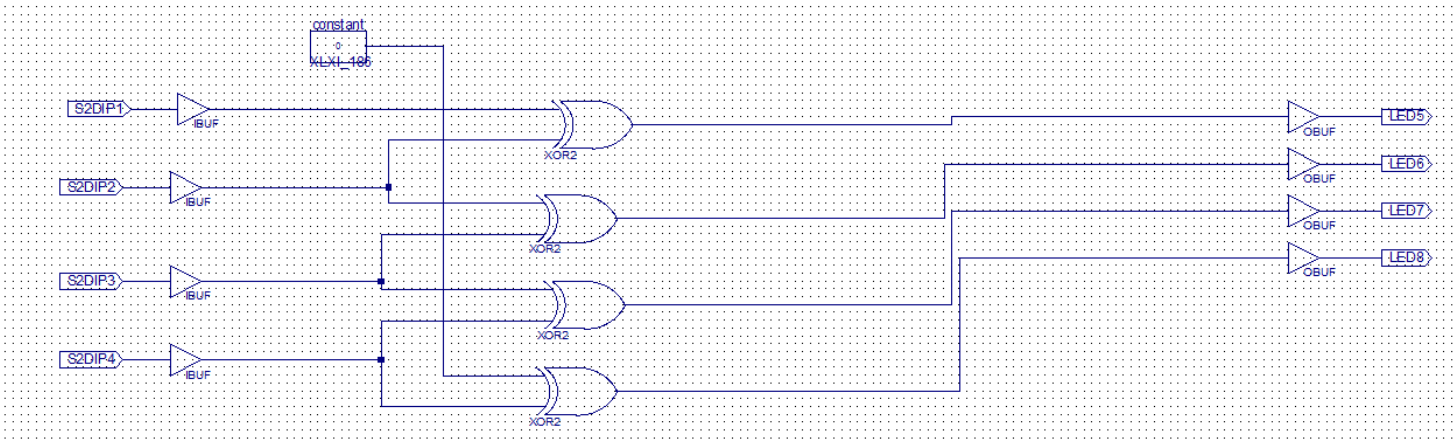
$$G_2 = B_3 \oplus B_2$$

Projektowanie dla G0:

B1B0\B3B2	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

$$G_0 = B_1 \oplus B_0$$

Wyznaczone funkcje G zrealizowano za pomocą poniższego schematu na bramkach EX-OR; jako wejścia $B0 \div B3$ podłączono mikroprzełączniki systemu ewaluacyjnego, natomiast jako wyjścia $G0 \div G3$ diody LED.



Zadanie 2

Korzystając z funkcji logicznej Ex-OR zrealizowano prostą logikę: konwerter 4-ro bitowego kodu Grey'a na 4-ro bitowy kod binarny.

Kod Grey'a

L.p.	G3	G2	G1	G0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1
14	1	0	0	1
15	1	0	0	0

Kod binarny

L.p.	B3	B2	B1	B0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Projektowanie dla B3:

G1G0\G3G2	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	1	1
10	0	0	1	1

$$B_3 = G_3$$

Projektowanie dla B2:

G1G0\G3G2	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

$$B_2 = G_3 \oplus G_2$$

Projektowanie dla B1:

G1G0\G3G2	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	1	0	1	0
10	1	0	1	0

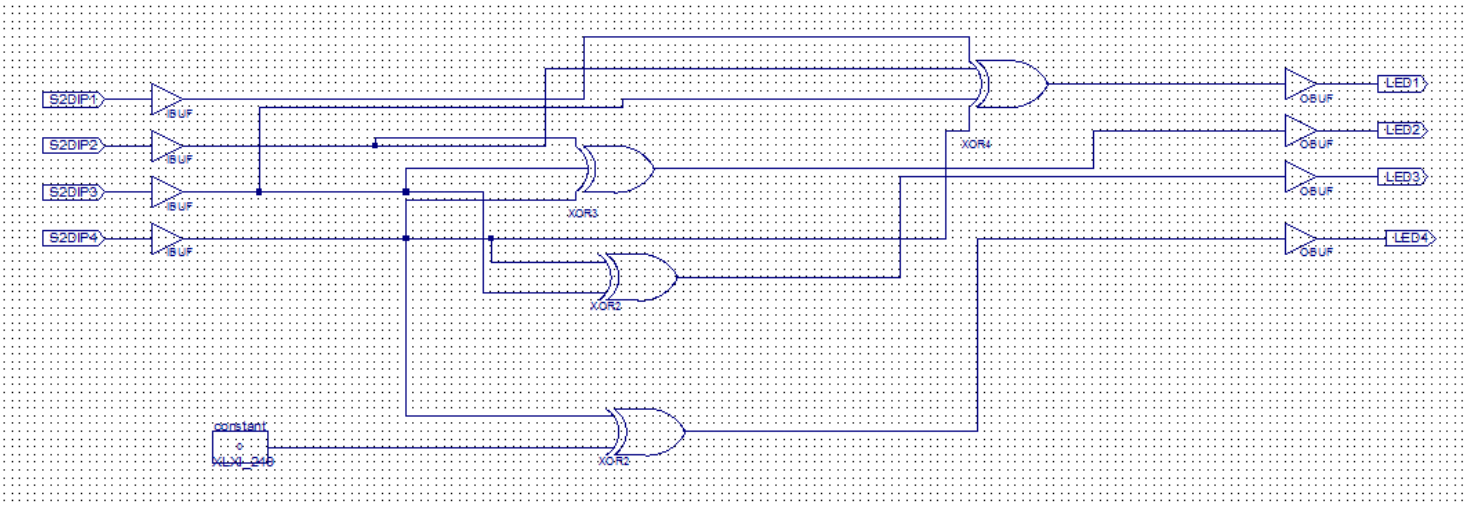
$$B_1 = G_3G_2 \oplus \overline{G_3}\,\overline{G_2} \oplus \overline{G_1}$$
$$= G_3G_2 \oplus \overline{G_3} \oplus \overline{G_3}G_2 \oplus \overline{G_1}$$
$$= G_3G_2 \oplus 1 \oplus G_3 \oplus G_2 \oplus G_3G_2 \oplus 1 \oplus G_2 = G_3 \oplus G_2 \oplus G_1$$

Projektowanie dla B0:

G1G0\G3G2	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

$$\begin{aligned} \boldsymbol{B_3} &= \boldsymbol{\overline{G_3}G_2 \oplus \overline{G_2}G_3 \oplus \overline{G_1}G_0 \oplus \overline{G_0}G_1} \\ &= \boldsymbol{G_2 \oplus G_2G_3} \\ &\boldsymbol{\oplus G_3 \oplus G_2G_3} \\ &\boldsymbol{\oplus G_0 \oplus G_0G_1} \\ &\boldsymbol{\oplus G_1 \oplus G_0G_1} \\ &= \boldsymbol{G_2 \oplus G_3 \oplus G_0 \oplus G_1} \end{aligned}$$

Wyznaczone funkcje B zrealizowano za pomocą poniższego schematu na bramkach EX-OR; jako wejścia $G0 \div G3$ podłączono mikroprzełączniki systemu ewaluacyjnego, natomiast jako wyjścia $B0 \div B3$ diody LED.



Zadanie 3

Korzystając z dowolnych funkcji logicznych zrealizowano dekoder 3-bitowego naturalnego kodu binarnego (NKB) na kod „1 z 8”.

NKB

B2	B1	B0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Kod „1 z 8”

Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0

Projektowanie dla Q7:

B2B1\B0	0	1
00	0	0
01	0	0
11	0	1
10	0	0

$Q_7 = B_2 \wedge B_1 \wedge B_0$

Projektowanie dla Q4:

B2B1\B0	0	1
00	0	0
01	0	0
11	0	0
10	1	0

$Q_4 = B_2 \wedge \overline{B_1} \wedge \overline{B_0}$

Projektowanie dla Q1:

B2B1\B0	0	1
00	0	1
01	0	0
11	0	0
10	0	0

$Q_1 = \overline{B_2} \wedge \overline{B_1} \wedge B_0$

Projektowanie dla Q6:

B2B1\B0	0	1
00	0	0
01	0	0
11	1	0
10	0	0

$Q_6 = B_2 \wedge B_1 \wedge \overline{B_0}$

Projektowanie dla Q3:

B2B1\B0	0	1
00	0	0
01	0	1
11	0	0
10	0	0

$Q_3 = \overline{B_2} \wedge B_1 \wedge B_0$

Projektowanie dla Q0:

B2B1\B0	0	1
00	1	0
01	0	0
11	0	0
10	0	0

$Q_0 = \overline{B_2} \wedge \overline{B_1} \wedge \overline{B_0}$

Projektowanie dla Q5:

B2B1\B0	0	1
00	0	0
01	0	0
11	0	0
10	0	1

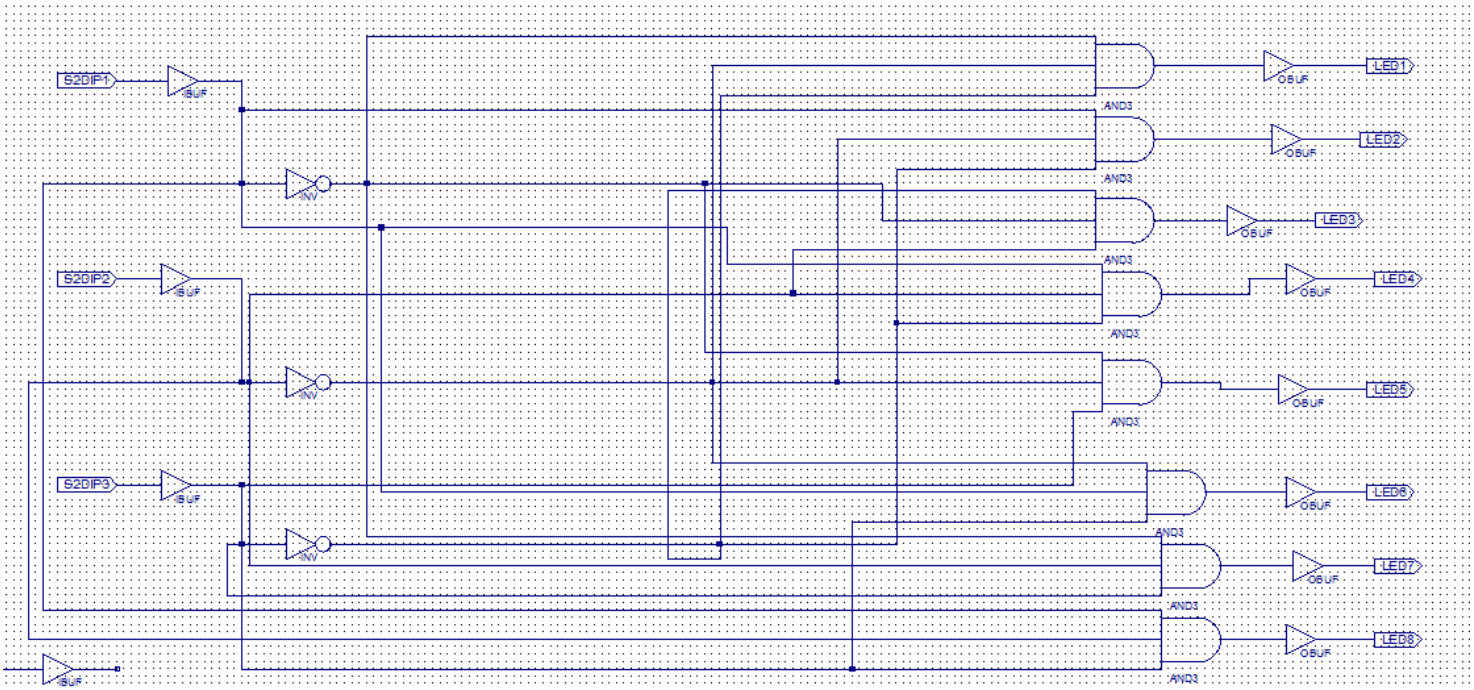
$Q_5 = B_2 \wedge \overline{B_1} \wedge B_0$

Projektowanie dla Q2:

B2B1\B0	0	1
00	0	0
01	1	0
11	0	0
10	0	0

$Q_2 = \overline{B_2} \wedge B_1 \wedge \overline{B_0}$

Wyznaczone funkcje Q zrealizowano za pomocą poniższego schematu; jako wejścia $B0 \div B2$ podłączono mikroprzełączniki systemu ewaluacyjnego, natomiast jako wyjścia $Q0 \div Q7$ diody LED.



Podsumowanie

Na zajęciach zapoznaliśmy się ze sposobem programowania układów logicznych typu CPLD. Mieliśmy możliwość przećwiczyć otrzymywanie kanonicznej postaci funkcji wielomianowej na podstawie tabel Karnaugha oraz projektowanie układów z wykorzystaniem bramek EX-OR.