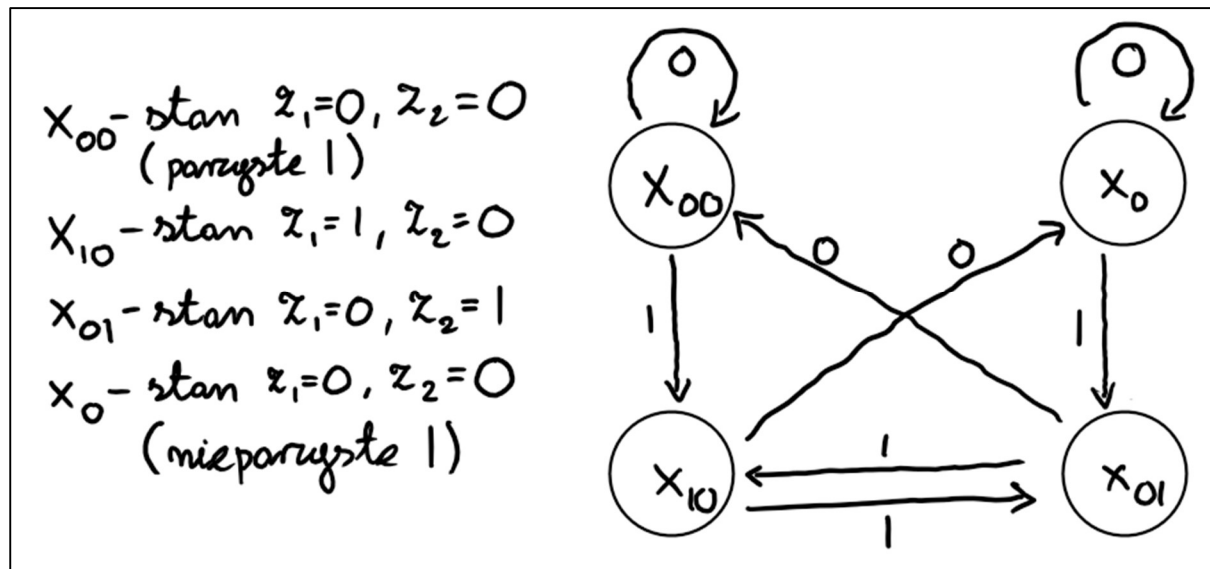


<h1>Sprawozdanie z układów logicznych</h1>	
	Ćwiczenie nr: 5
	Temat ćwiczenia: Analiza układu synchronicznego.
1. Imię i nazwisko – student 1: Wojciech Krzos	
2. Imię i nazwisko – student 2: Natalia Marszałek	
Grupa laboratoryjna nr (u prowadzącego): 5	Dzień tygodnia: Czwartek
Płyta montażowa nr (z tyłu zadajnika): NA	Godziny zajęć (od-do): 13:15 – 15:00

1 PRZEBIEG BADANIA

1.1 SYNTEZA AUTOMATU MOORE'A

1.1.1 Graf stanów-wyjść



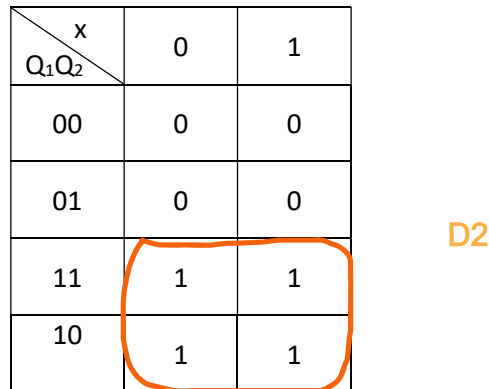
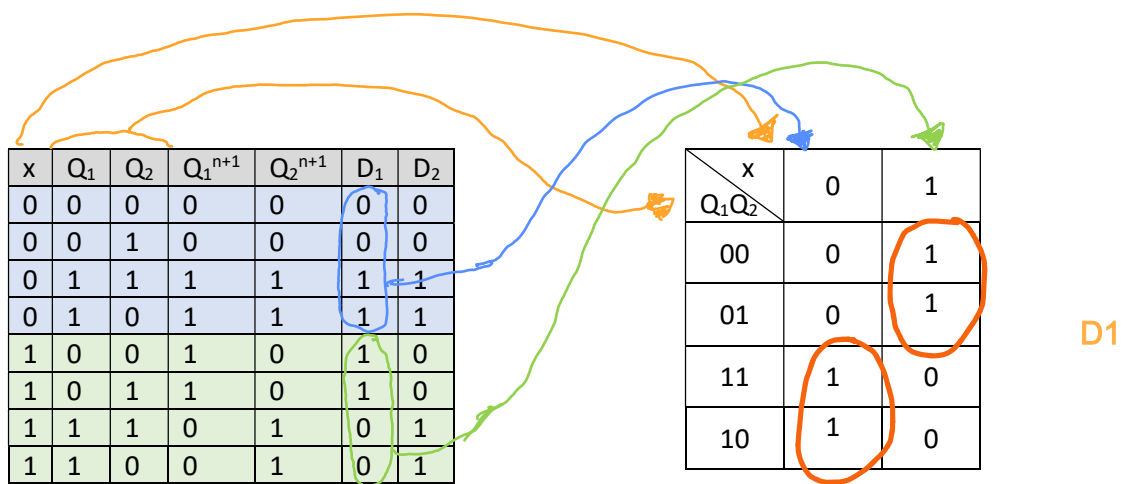
1.1.2 Tablica przejść-wyjść

$\begin{matrix} x \\ z_1 z_2 \end{matrix}$	0	1	z_1	z_2
X_{00}	X_{00}	X_{10}	0	0
X_{10}	X_0	X_{01}	1	0
X_{01}	X_{00}	X_{10}	0	1
X_0	X_0	X_{01}	0	0

wyjścia układowe

1.1.3 Kodowanie stanów, tablica

Następnie, zajmiemy się analizą przerzutników typu D względem powyżej tabeli i grafu. Warto zwrócić uwagę na fakt, że z racji na naturę przerzutnika typu D, następne stany jego wyjścia (odpowiednio Q_1 i Q_2 dla obu przerzutników) równe są wejściom danych (D_1 i D_2). Pomoże to w stworzeniu osobnych tablic, które następnie zostaną wykorzystane do stworzenia funkcji wzбудzającej przerzutniki.



1.1.4 Otrzymanie funkcji wzbudzającej wejścia przerzutników

Używając powyższych, możemy stworzyć funkcję wzbudzającą przerzutniki oraz stworzyć jej tabelę:

$$D_1 = x * \neg Q_1 + \neg x * Q_1 = x \oplus Q_1$$

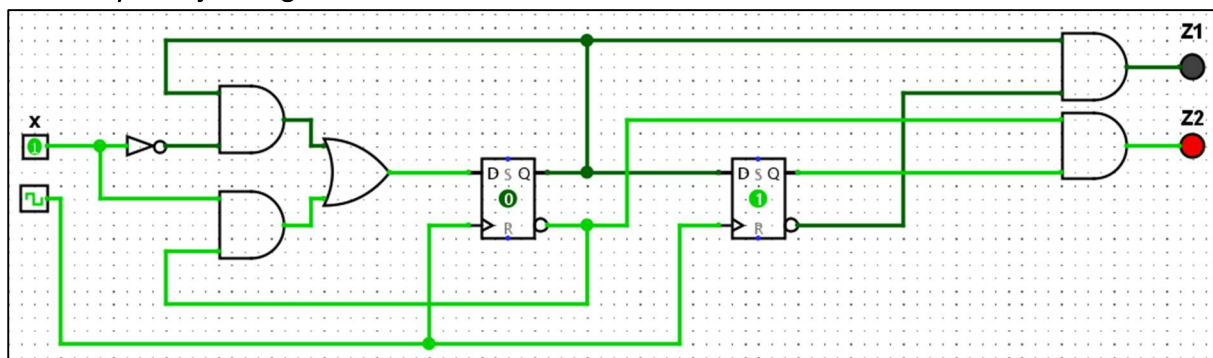
$$D_2 = Q_1$$

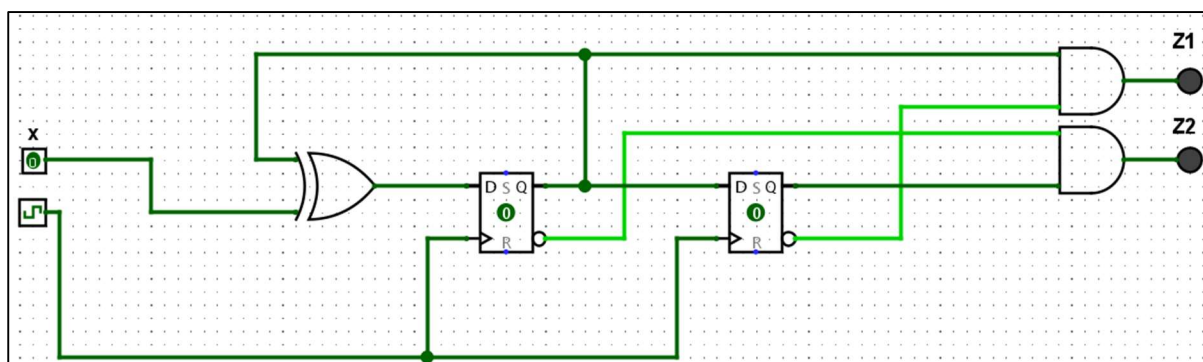
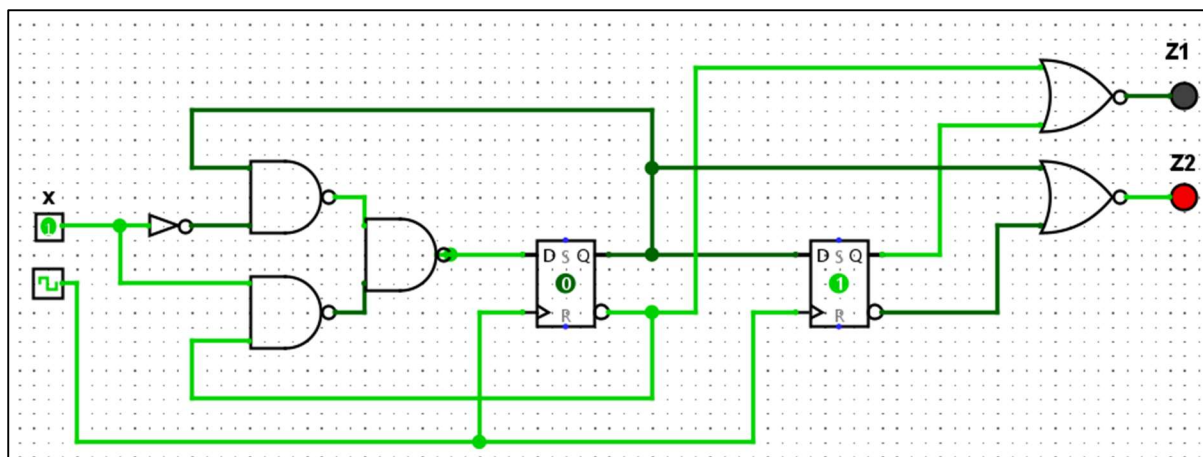
Zacieniowano wiersze, w których wyjście Z₁ lub Z₂ równe jest 1. Analizując, można dojść do wniosku, że dla Z₁ = 1, wejście Q₁ = 1 oraz Q₂ = 0; można również zauważyć, że dla Z₂ = 1: Q₁ = 0, a Q₂ = 1. Poniższy układ złożony z bramek AND i OR został następnie uproszczony do układu zawierającego bramki NAND i NOR.

Q ₁	Q ₂	Z ₁	Z ₂
0	0	0	0
0	1	0	1
1	1	0	0
1	0	1	0

Dodatkowo, może zostać użyta własność algebry Bool'a: $a \times \neg b + \neg a \times b = a \oplus b$. Jest to układ używający najmniejszej ilości bramek, do jakiego doszliśmy w trakcie badania.

1.1.5 Symulacja w LogiSim





1.1.6 Diagram czasowy

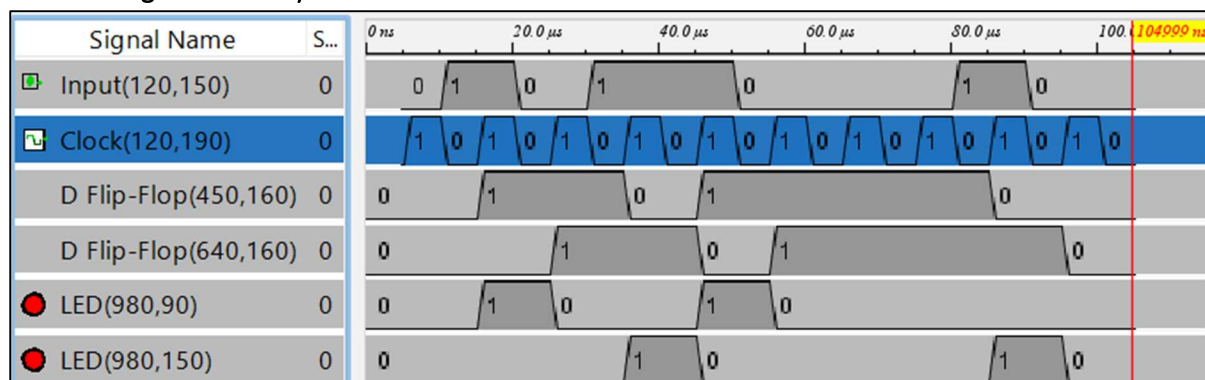
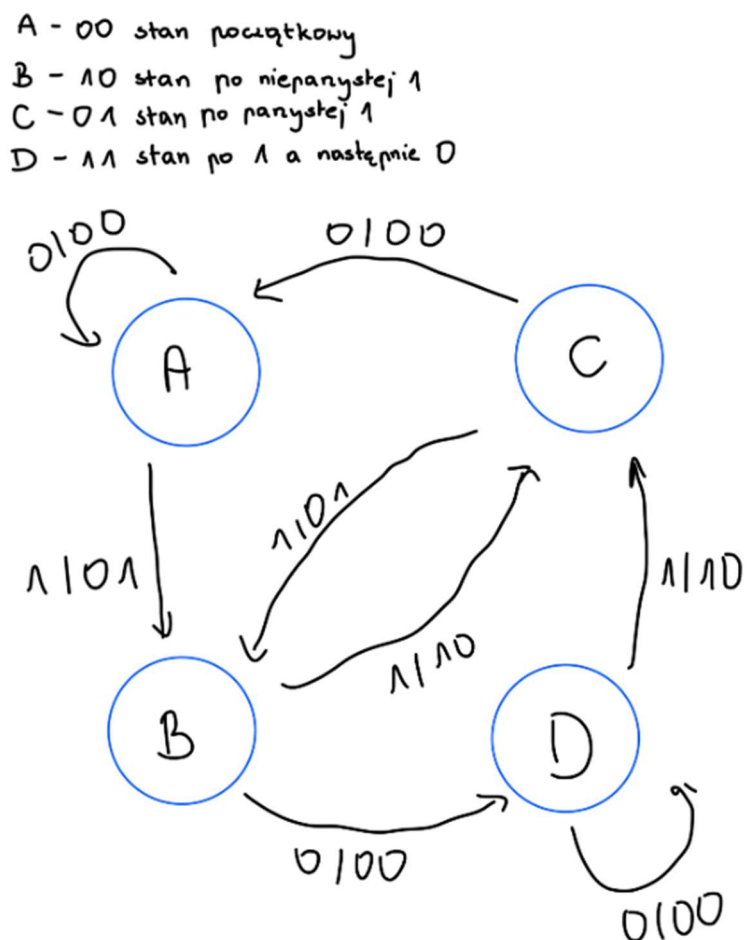


Figure 1: Sekwencja 0101100010

1.2 SYNTEZA AUTOMATU MEALY'EGO

1.2.1 Graf stanów-wyjść



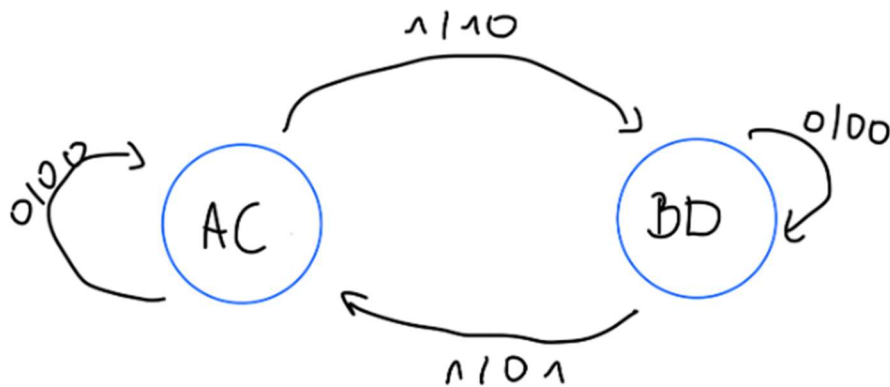
2.2.2 Tablica stanów-wyjść:

Stan\X	0	1
A	A	B
B	D	C
C	A	B
D	D	C

Jak widzimy, nasza tabela składa się z dwóch identycznych części, zatem możemy ją zminimalizować do postaci:

STAN\X	0	1
AC(0)	AC	BD
BD(1)	BD	AC

A graf do postaci:



2.2.3 Kodowanie stanów

Aby móc przekształcić naszą funkcję do postaci układu z przerzutnikiem D potrzebujemy zakodować stany :

STAN\X	0	1
0	0	1
1	1	0

Stwórzmy mapę Karnougha - dzięki której będziemy wiedzieli jakie dane należy przekazać do przerzutnika :

$$f(Q, x) = \bar{Q}x + Q\bar{x}$$

Korzystając z praw algebry Bool'a możemy uprościć funkcję do jednego operatora XOR:

$$f(Q, x) = Q \oplus x$$

1.2.2 Otrzymanie funkcji wzbudzającej wejścia przerzutników

Aby otrzymać tę funkcję musimy przeanalizować tabele wartości na wyjściach Z_1 oraz Z_2 .

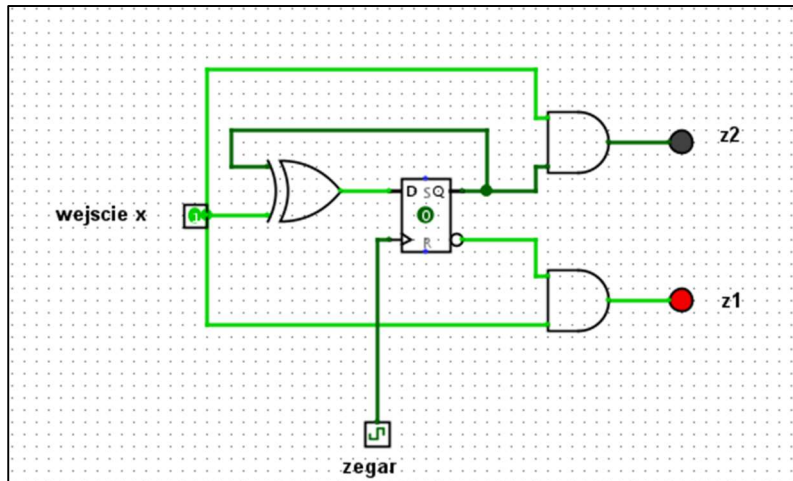
STAN\X	X	Z_1	Z_2
AC(0)	0	0	0
BD(1)	0	0	0
AC(0)	1	1	0
BD(1)	1	0	1

Otrzymaliśmy w ten sposób mapę Karnough, która pozwoli nam na pozyskanie potrzebnej funkcji dla każdego z wyjść :

$$\text{Dla } Z_1: f(Z_1) = x\bar{Q}$$

$$\text{Dla } Z_2: f(Z_2) = Q\bar{x}$$

1.2.3 Symulacja w LogiSim



1.2.4 Diagramy czasowe

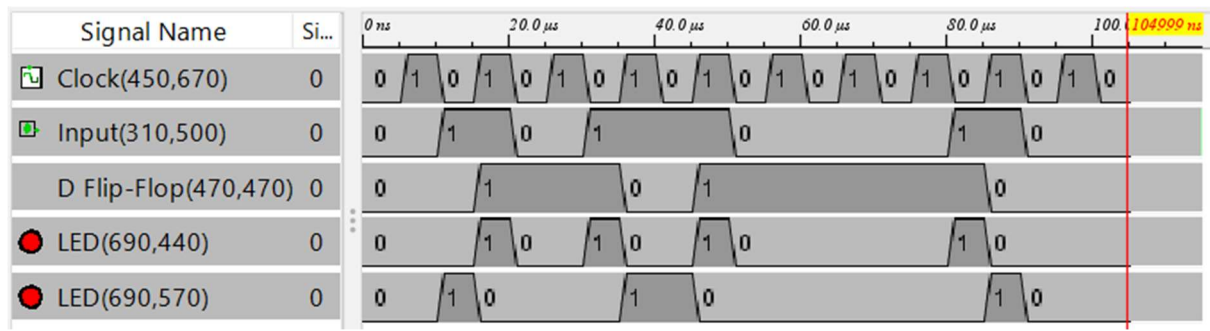


Figure 2: Diagram czasowy dla 0101100010

2 BIBLIOGRAFIA

1. Automat Moore'a. In: Wikipedia, wolna encyklopedia. 2019 [accessed 2023 Apr 19].
https://pl.wikipedia.org/w/index.php?title=Automat_Moore%E2%80%99a&oldid=57488148
2. Automata Moore Machine - Javatpoint. [accessed 2023 Apr 19].
<https://www.javatpoint.com/automata-moore-machine>
3. Finite State Machine Designer - by Evan Wallace. [accessed 2023 Apr 19].
<https://madebyevan.com/fsm/>
4. TC Moduł 7 – Studia Informatyczne. [accessed 2023 Apr 19].
https://wazniak.mimuw.edu.pl/index.php?title=TC_Modu%C5%82_7