|  |  |
| --- | --- |
| Sprawozdanie z układów logicznych |  |
| 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 |

# Przebieg badania

## Synteza automatu Moore’a

### Graf stanów-wyjść

Obraz zawierający diagram

Opis wygenerowany automatycznie

### Tablica przejść-wyjść

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x Z1Z2 | 0 | 1 | Z1 | Z2 |
| X00 | X00 | X10 | 0 | 0 |
| X10 | X0 | X01 | 1 | 0 |
| X01 | X00 | X10 | 0 | 1 |
| X0 | X0 | X01 | 0 | 0 |

### 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 Q1 i Q2 dla obu przerzutników) równe są wejściom danych (D1 i D2). Pomoże to w stworzeniu osobnych tablic, które następnie zostaną wykorzystane do stworzenia funkcji wzbudzającej przerzutniki.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | x | Q1 | Q2 | Q1­n+1 | Q2­n+1 | D1 | D2 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | 0 | 1 | 1 | 1 | 1 | 1 | 1 | | 0 | 1 | 0 | 1 | 1 | 1 | 1 | | 1 | 0 | 0 | 1 | 0 | 1 | 0 | | 1 | 0 | 1 | 1 | 0 | 1 | 0 | | 1 | 1 | 1 | 0 | 1 | 0 | 1 | | 1 | 1 | 0 | 0 | 1 | 0 | 1 | | |  |  |  | | --- | --- | --- | | x  Q1Q2 | 0 | 1 | | 00 | 0 | 1 | | 01 | 0 | 1 | | 11 | 1 | 0 | | 10 | 1 | 0 | |

|  |  |  |
| --- | --- | --- |
| x  Q1Q2 | 0 | 1 |
| 00 | 0 | 0 |
| 01 | 0 | 0 |
| 11 | 1 | 1 |
| 10 | 1 | 1 |

### Otrzymanie funkcji wzbudzającej wejścia przerzutników

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

|  |  |  |  |
| --- | --- | --- | --- |
| Q1 | Q2 | Z1 | Z2 |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |

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. Dodatkowo, może zostać użyta własność algebry Bool’a: . Jest to układ używający najmniejszej ilości bramek, do jakiego doszliśmy w trakcie badania.

### Symulacja w LogiSim

Obraz zawierający diagram

Opis wygenerowany automatycznie

Obraz zawierający diagram, wykres

Opis wygenerowany automatycznie

### Diagram Description automatically generated

### Diagram czasowy

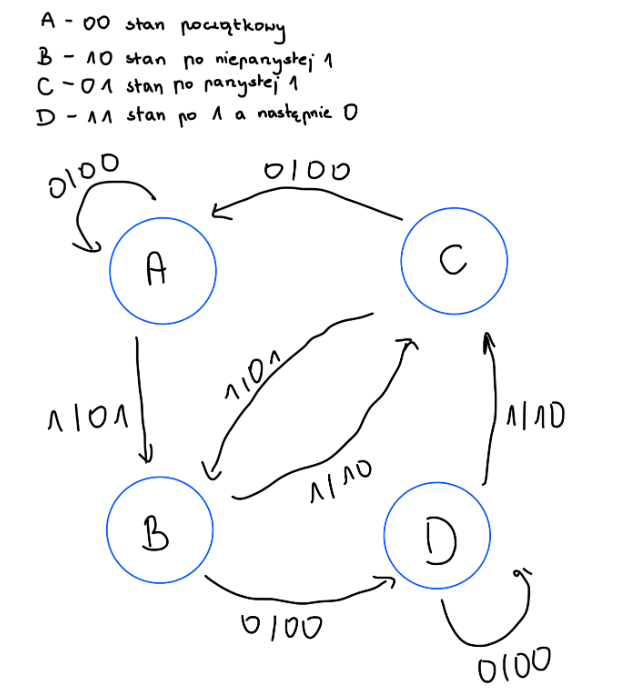
Graphical user interface, application, table, Excel

Description automatically generated

Figure : Sekwencja 0101100010

## Synteza automatu Mealy’ego

### Graf stanów-wyjść



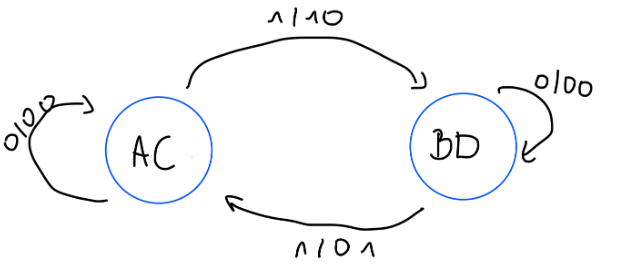
### 2.2.2 Tablica stanów-wyjść:

|  |  |  |
| --- | --- | --- |
| 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 fukcję do postaci układu z przerzutnikiem D potrzebujemy zakodować stany :

|  |  |  |
| --- | --- | --- |
| STAN\X | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 1 | 0 |

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

Korzystając z praw algebry Bool’a możemy uprościć funcję do jednego operatora XOR:

### Otrzymanie funkcji wzbudzającej wejścia przerzutników

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

|  |  |  |  |
| --- | --- | --- | --- |
| STAN\X | X |  |  |
| 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 Karnought, która pozwoli nam na pozyskanie potrzebnej fukcji dla każdego z wyjść :

Dla :

Dla :

### Symulacja w LogiSim

Obraz zawierający diagram

Opis wygenerowany automatycznie

### Diagramy czasowe

# 

Figure : Diagram czasowy dla 0101100010

# 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>