Układy Elektroniczne - Logika sekwencyjna

Grzegorz Litarowicz
Piotr Moszkowicz

7 czerwca 2019

Spis treści

1 Cel i zakres ćwiczenia

Celem ćwiczenia było zapoznanie się z logiką sekwencyjną oraz utworzenie prostych układów (liczników).

2 Wstęp teoretyczny

2.1 Logika sekwencyjna

Logika, w której wyjście zależy nie tylko od wejścia, ale również od stanu poprzedniego, który zwany jest stanem wewnętrznym i jest pamiętany w rejestrze. Dzięki tej właściwości, możemy tworzyć układy, które nie tylko reagują na wejście, ale również zapamiętują swój stan, co jest bardzo powszechnie wykorzystywane - w naszym przypadku przy konstrukcji liczników.

2.2 Kod Grey'a

Kod dwójkowy, który charakteryzuje się tym, iż kolejne wartości (słowa kodowe) różnią się między sobą stanem tylko jednego bitu. Ostatni i pierwszy wyraz tego kodu również spełnia to wymaganie.

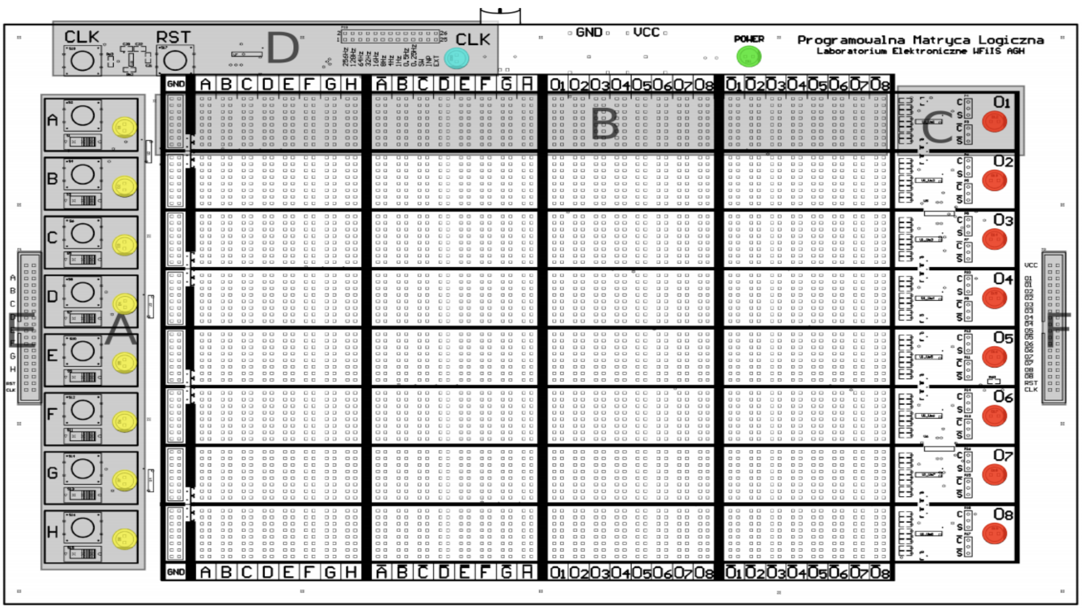
2.3 Przesunięcie bitowe

Operacja, którą wykonujemy na liczbach binarnych. Polega na zamianie miejscami bitów danej liczby, zależnie w którą stronę wykonujemy przesunięcie. Poniżej pokazany jest przykład przesunięcia bitowego w prawo:

Wejście: 1011 Wyjście: 1101

Jak widać powyżej każdy bit "przeskoczył" w prawo, natomiast ostatni wszedł na miejsce pierwszego.

3 Schemat płytki PLD



4 Pomiary i wyniki

4.1 4 bitowy licznik modulo 14

Jako pierwsze ćwiczenie mieliśmy zaprojektować licznik 4 bitowy, który wykonywał operację modulo 14. Poniżej można zaobserwować zeskanowane tabele Karnaugh'a oraz wzory 4 funkcji, które wykorzystaliśmy do zbudowania tegoż licznika.

	Q_4	Q_3	Q_2	Q_1	nQ_4	nQ_3	nQ_2	nQ_1
0 -	0	0	0	0	0	0	0	1
1 -	0	0	0	1	0	0	1	0
2 -	0	0	1	0	0	0	1	1
3 -	0	0	1	1	0	1	0	0
4 -	0	1	0	0	0	1	0	1
5 -	0	1	0	1	0	1	1	0
6 -	0	1	1	0	0	1	1	1
7 -	0	1	1	1	1	0	0	0
8 -	1	0	0	0	1	0	0	1
9 -	1	0	0	1	1	0	1	0
10 -	1	0	1	0	1	0	1	1
11 -	1	0	1	1	1	1	0	0
12 -	1	1	0	0	1	1	0	1
13 -	1	1	0	1	0	0	0	0
14								
15								

nQ_1

Q_4Q_3 \ Q_2Q_1	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	x	x
10	1	0	0	1

nQ_2

Q_4Q_3 \ Q_2Q_1	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	0	x	x
10	0	1	0	1

$$nQ_1 = \overline{Q_2}\,\overline{Q_1} + Q_2\overline{Q_1}$$

$$nQ_2 = \overline{Q_4}\,\overline{Q_2}Q_1 + \overline{Q_3}\,\overline{Q_2}Q_1 + \cancel{Q_3 Q_2 \overline{Q_1}} + Q_2\overline{Q_1} + \cancel{Q_4 Q_2 \overline{Q_1}}$$

O_4':

O_4O_3 \ O_2O_1	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	1	0	X	X
10	1	1	1	1

$$O_4' = O_4\overline{O_3} + O_4\overline{O_2}\,\overline{O_1} + O_3 O_2 O_1$$

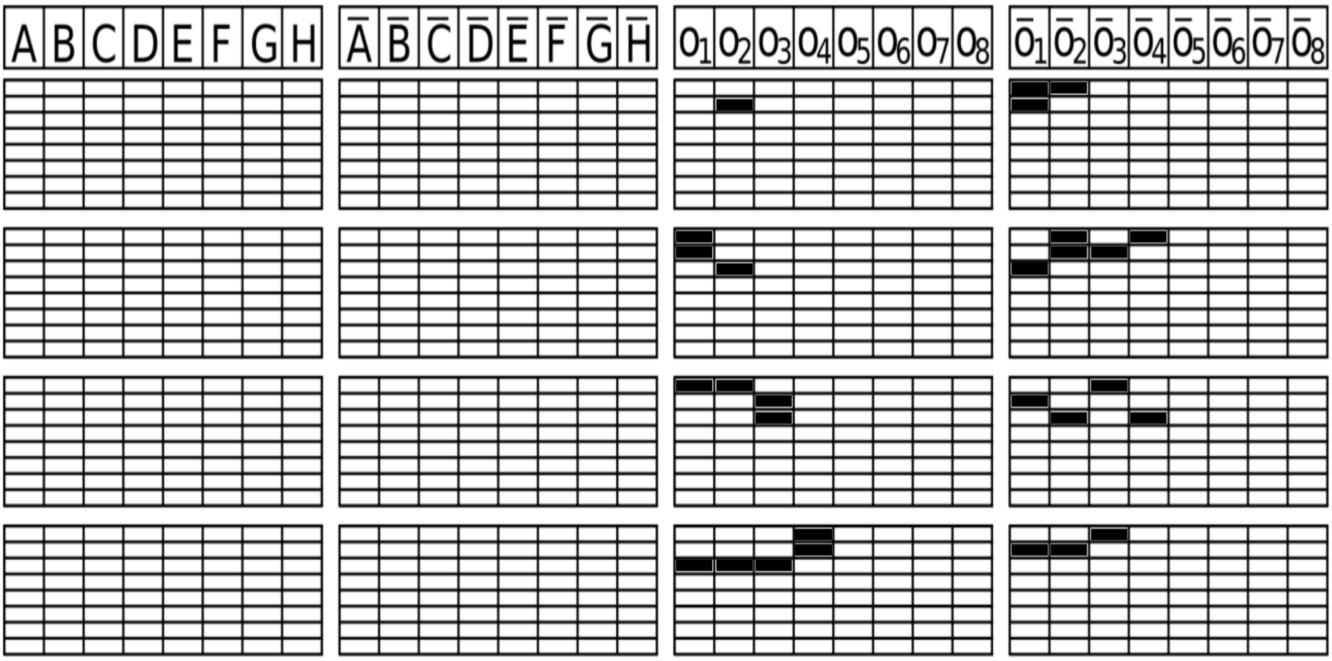
O_3'

O_4O_3 \ O_2O_1	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	1	0	X	X
10	0	0	1	0

$$O_3' = \cancel{\overline{O_4}\,\overline{O_3}\,O_2O_1} + \cancel{O_4O_3} \quad \cancel{\overline{O_3}\,O_2O_1 + \overline{O_1} + \overline{O_4}\,O_3\overline{O_2}}$$

$$\overline{O_3}\,O_2O_1 + O_3O_1 + O_4O_2O_2$$

Natomiast na poniższej tabeli zaprezentowana jest zworkowa realizacja licznika.



4.2 3 bitowy licznik sterowany sygnałem A w kodzie Grey'a

Kolejnym poleceniem było wykonanie 3 bitowego licznika w kodzie Grey'a (punkt 2.2), który w zależności od sygnału A wykonywał poniższe funkcje:

- A = 0 => Licznik zmienia się od 1 -> 6 w górę
- A = 1 => Rotate right - przesunięcie bitowe w prawo. (punkt 2.3)

Poniżej znajdują się zeskanowane tabele Karnaugh'a, które przedstawiają trzy konieczne funkcję, które składają się na ten układ.

nQ_1

$AQ_3 \backslash Q_2Q_1$	00	01	11	10
00	1	1	0	0
01	X	1	1	1
11	0	0	1	1
10	1	0	1	1

$\vee \bar{O} O_3$

$$nQ_1 = \left(\bar{A}\,\bar{Q}_2 + Q_3 Q_2 + A\,Q_2 + \cancel{\bar{Q}_1 A \bar{Q}_3} \right) B = 0$$

	B	Q_3	Q_2	Q_1
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
	0	1	0	1
	0	1	0	0

Q_3

$BQ_3 \backslash Q_2Q_1$	00	01	11	10
00	0	0	0	0
01	1	1	1	1
~~11~~				
~~10~~				

O_3':

$A\,O_3 \backslash O_2 O_1$	00	01	11	10
00	0	0	0	1
01	X	0	1	1
11	0	1	1	0
10	0	1	1	0

$O_3' = \cancel{\bar{A}\bar{O}_3 O_2 \bar{O}_1 + \bar{A} O_3 O_2 + \ldots}$

$\bar{A}\bar{O}_3 O_2 \bar{O}_1 + \bar{A} O_3 O_2 + A O_2$

O_2':

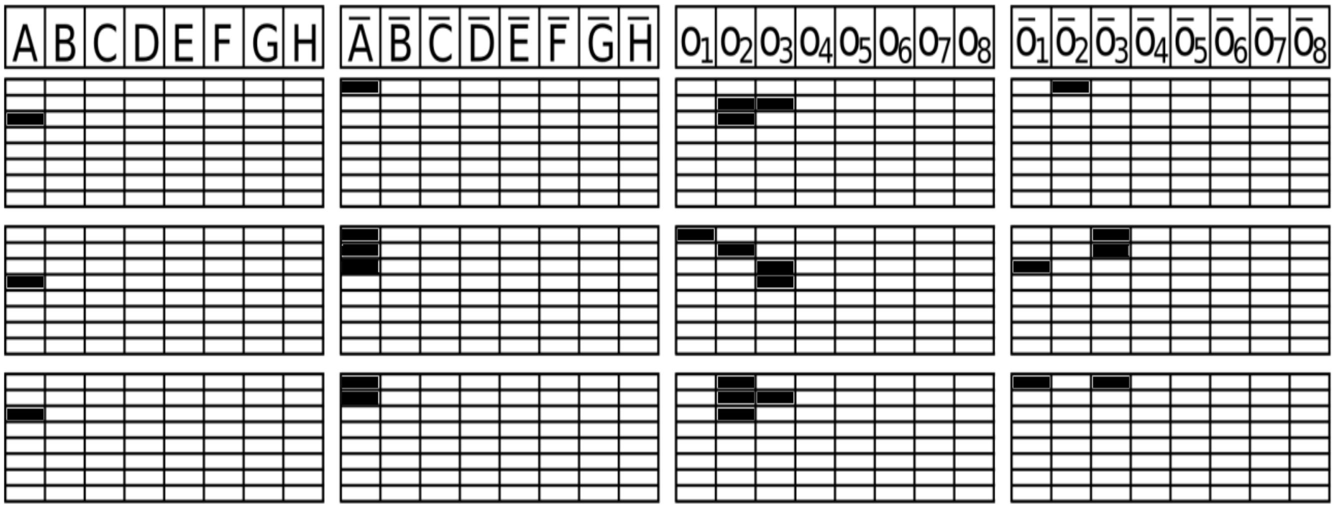
$A\,O_3 \backslash O_2 O_1$	00	01	11	10
00	0	1	1	1
01	X	0	0	1
11	1	1	1	1
10	0	0	0	0

$\cancel{O_2' = \bar{A}\bar{O}_3 \bar{O}_2 O_1 + \bar{A}\bar{O}_3 O_2 + \bar{A} O_3 \bar{O}_1 + A O_3}$

$O_2' = \cancel{\bar{A}\bar{O}_3 O_1 + \bar{A} O_3 \quad \bar{A} O_3 \bar{O}_1 + A O_3}$

$\bar{A}\bar{O}_3 O_1 + \bar{A}\bar{O}_3 O_2 + \bar{A} O_3 \bar{O}_1 + A O_3$

Natomiast poniżej można zaobserwować reprezentację zworkową tychże funkcji na matrycy PLD.



4.3 3 bitowy licznik sterowany sygnałem A w kodzie Grey'a, które zapamiętuje stan

Ostatnim poleceniem było przerobienie w taki sposób licznika z punktu 4.2, aby licznik w momencie, gdy sygnał B = 0 zatrzymywał się. Okazało się, że jedynym krokiem, jaki musimy zrobić to dodać B do aktualnych członów funkcji oraz dodać OR zanegowane B, aktualna funkcja.

Finalnie poniżej znajdują się zapisane funkcje:

$O_1' = B(\overline{AO_2} + O_3O_2 + AO_2) + \overline{B}O_1$

$O_2' = B(\overline{AO_3}O_1 + \overline{AO_3}O_2 + \overline{AO_1}O_3 + AO_3) + \overline{B}O_2$

$O_3' = B(\overline{AO_3O_1}O_2 + \overline{A}O_3O_2 + AO_2) + \overline{B}O_3$

Poniżej znajduje się reprezentacja zworkowa finalnego licznika: