

Sprawozdanie 3: Liczniki i automaty

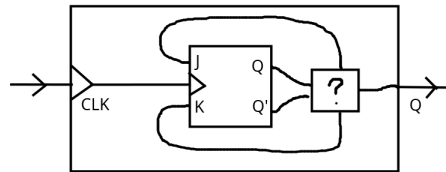
**Jakub Szymczak, Damian Tworek,
Maksymilian Sulima, Łukasz Wala**

*AGH, Wydział Informatyki, Elektroniki i Telekomunikacji
Technika Cyfrowa 2021/2022*

Kraków, 18 maja 2022

1 Ćwiczenie 3a

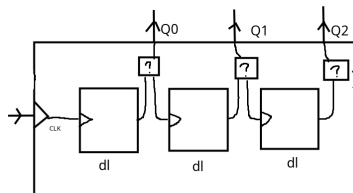
Ideą ćwiczenia jest zaprojektowanie, zrealizowanie i przetestowanie dwójki liczącej w oparciu o przerzutnik JK.



Rysunek 1: Dwójka licząca

Układ ma wejście zegara **CLK** oraz wyjście **Q**. W kolejnych taktach zegara stan na wyjściu **Q** zmienia się na przeciwny.

Następnie w oparciu o jeden z wariantów zaprojektowanej dwójki liczącej, zaprojektowany i przetestowany zostanie asynchroniczny licznik modulo 8.



Rysunek 2: Asynchroniczny licznik modulo 8

Wejście zegara oznaczono jako **CLK**, natomiast wyjścia to (w kolejności od najmniej istotnego bitu) **Q0**, **Q1**, **Q2**.

1.1 Rozwiązanie teoretyczne

Tabela prawdy dla dwójki liczącej wygląda następująco:

Q_n	Q_{n-1}
0	1
1	0

Rysunek 3: Tabela prawdy dla dwójki liczącej

Do zbudowania jej będzie potrzebny jeden przerzutnik JK:

J	K	Q_n
0	0	Q_{n-1}
0	1	0
1	0	1
1	1	\overline{Q}_{n-1}

Rysunek 4: Tabela prawdy dla przerzutnika JK

Na podstawie tych dwóch tabel można wywnioskować, które stany przerzutnika JK będzie można wykorzystać (z możliwych kombinacji wybrane zostały dwie):

J	K	Q_{n-1}	Q_n
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

J	K	Q_{n-1}	Q_n
0	1	1	0
1	0	0	1

J	K	Q_{n-1}	Q_n
1	1	0	1
1	1	1	0

Rysunek 5: Rozwiązanie teoretyczne

Tabele Karnaugh dla pierwszego przypadku:

J:

$\begin{matrix} Q_{n-1} & Q_n \\ \hline J & K \end{matrix}$	00	01	11	10
0	-	1	-	-
1	-	-	-	0

K:

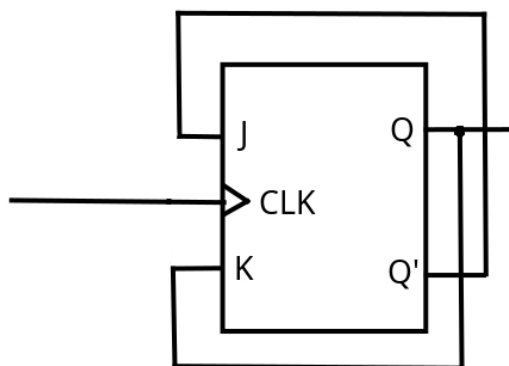
$\begin{matrix} Q_{n-1} & Q_n \\ \hline J & K \end{matrix}$	00	01	11	10
0	-	-	-	1
1	-	0	-	-

Rysunek 6: Tabela Karnaugh dla przypadku 1

Można zauważyć, że Q_{n-1} jest zawsze równe \bar{Q} , więc można skorzystać z wyjścia zanegowanego wyjścia Q w przerzutniku JK.

$$J = Q_{n-1} = \bar{Q}$$

$$K = Q_n$$



Rysunek 7: Schemat dla tabeli 1

Tabela Karnough dla drugiego przypadku:

$J:$

$\begin{array}{c} Q_{n-1} \\ \backslash \\ K \end{array} \begin{array}{c} Q_n \end{array}$	00	01	11	10
0	-	-	-	-
1	-	1	-	1

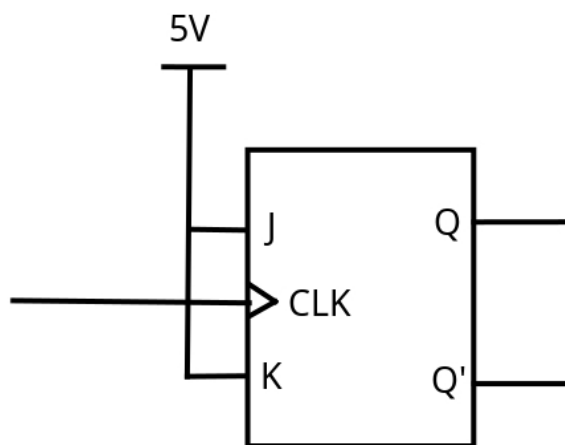
$K:$

$\begin{array}{c} Q_{n-1} \\ \backslash \\ J \end{array} \begin{array}{c} Q_n \end{array}$	00	01	11	10
0	-	-	-	-
1	-	1	-	1

Rysunek 8: Tabela Karnough dla przypadku 2

$$J = 1$$

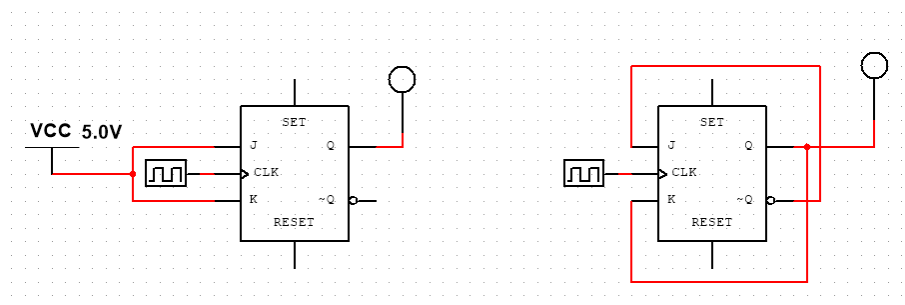
$$K = 1$$



Rysunek 9: Schemat dla tabeli 2

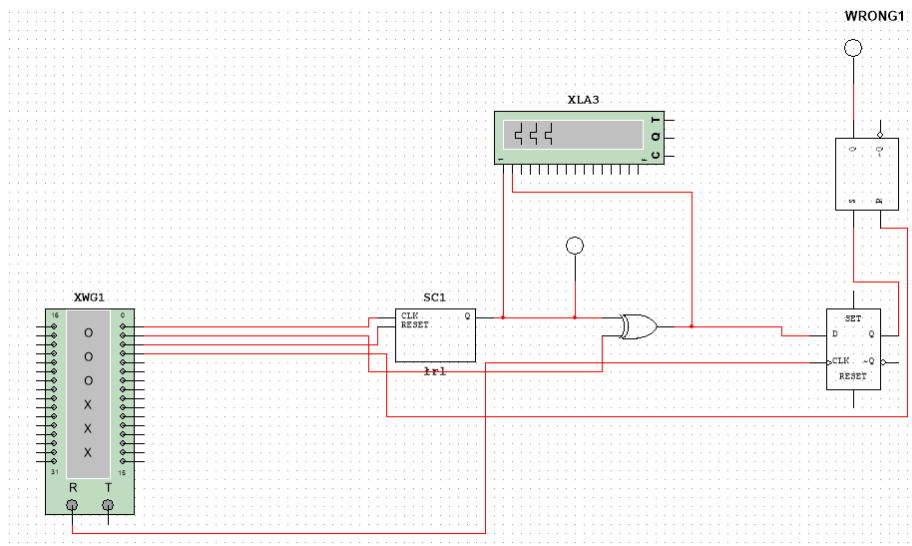
1.2 Implementacja w programie Multisim

Poniżej znajduje się implementacja obu wariantów układu:

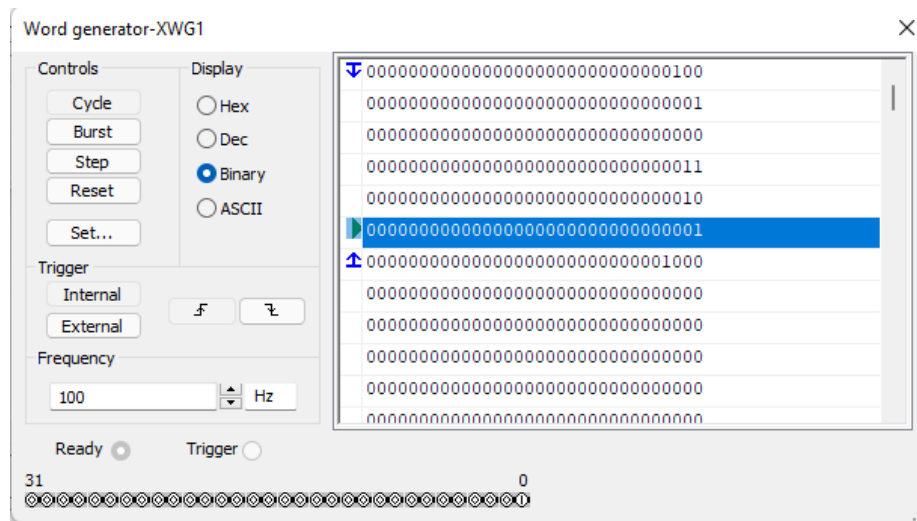


Rysunek 10: Implementacja dwójki liczącej

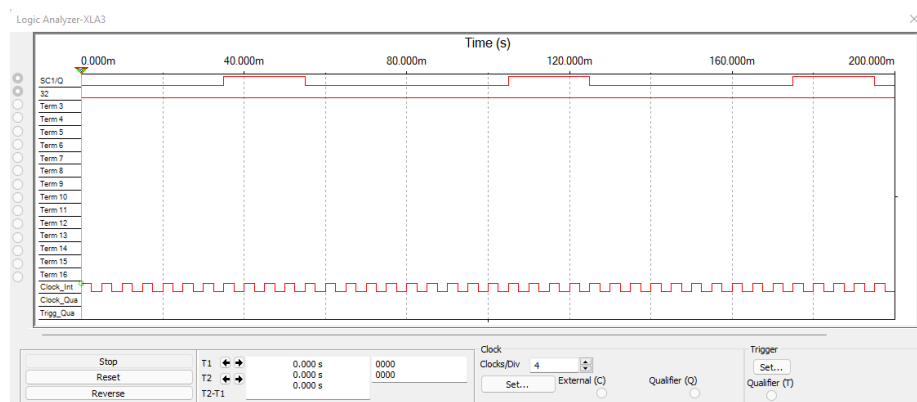
Układ testujący dla dwójki liczącej (używający wariantu po lewej stronie):



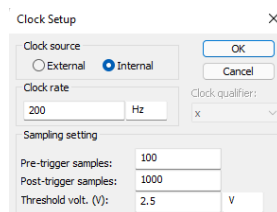
Rysunek 11: Układ testujący dla dwójki liczącej



Rysunek 12: Generator słów dla układu testującego



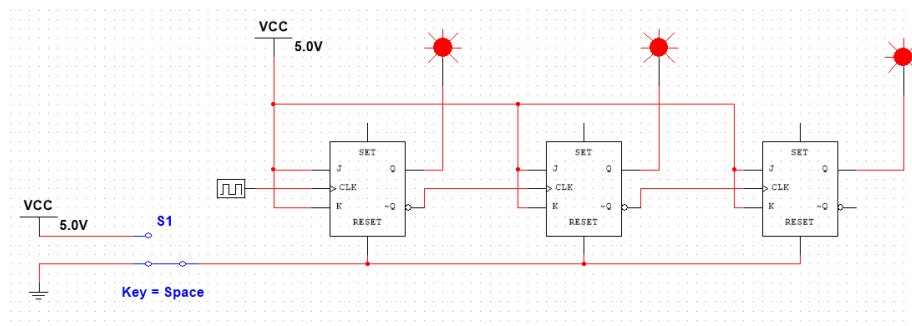
Rysunek 13: Analizator dla układu testującego



Rysunek 14: Ustawienia analizatora

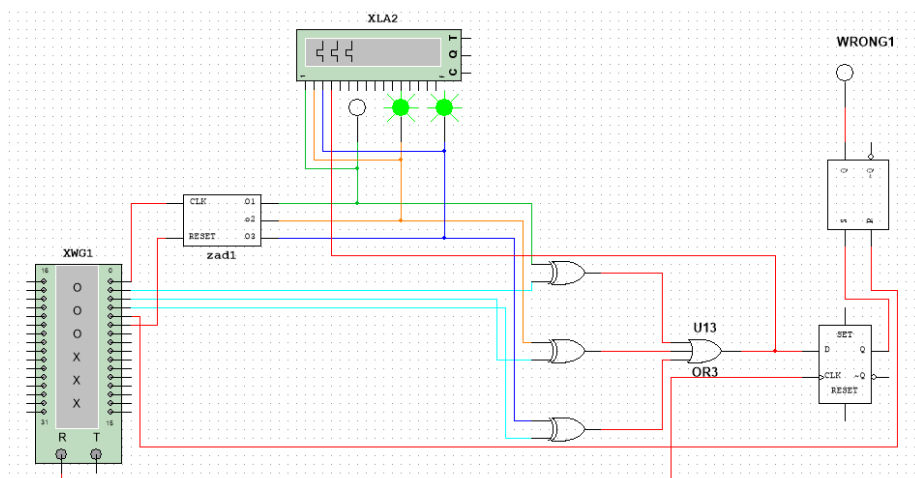
1.3 Asynchroniczny licznik modulo 8

Asynchroniczny licznik modulo 8 (odliczający do góry 0,1,2,...,7) można stworzyć używając trzech dwójek liczących, łącząc ich zanegowane wyjścia z wejściami zegara kolejnych (tylko, gdy użyty przerzutnik JK reaguje na wzrastający sygnał zegara):

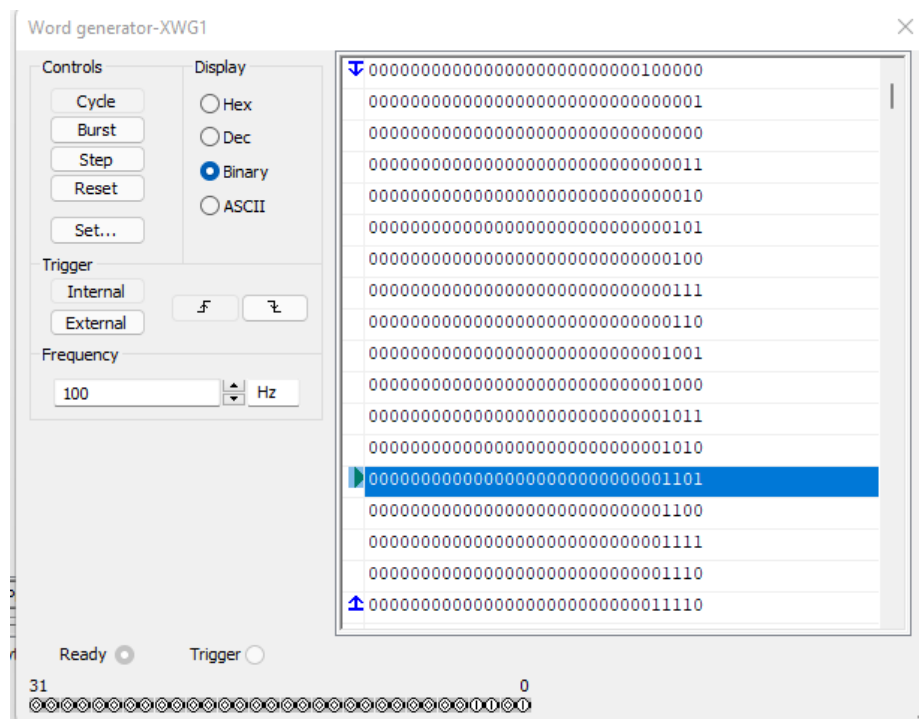


Rysunek 15: Implementacja asynchronicznego licznika modulo 8

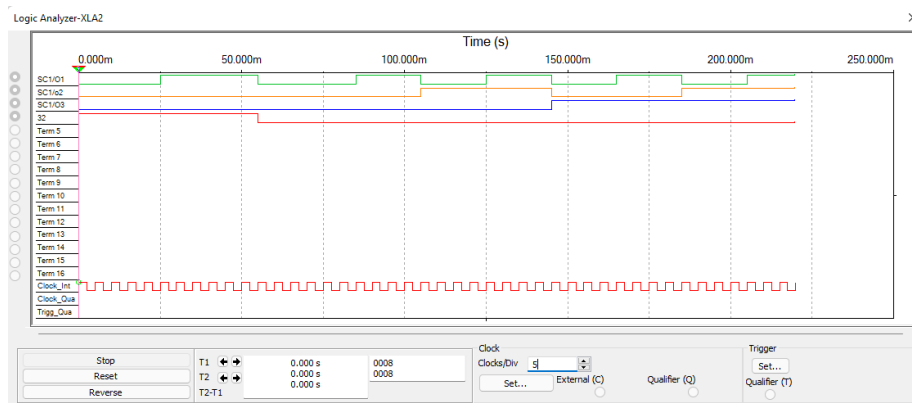
Poniżej układ testujący:



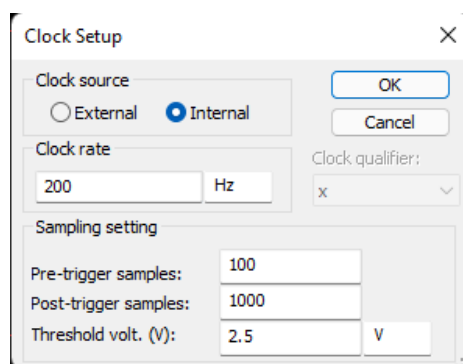
Rysunek 16: Układ testujący dla asynchronicznego licznika modulo 8



Rysunek 17: Generator słów dla układu testującego



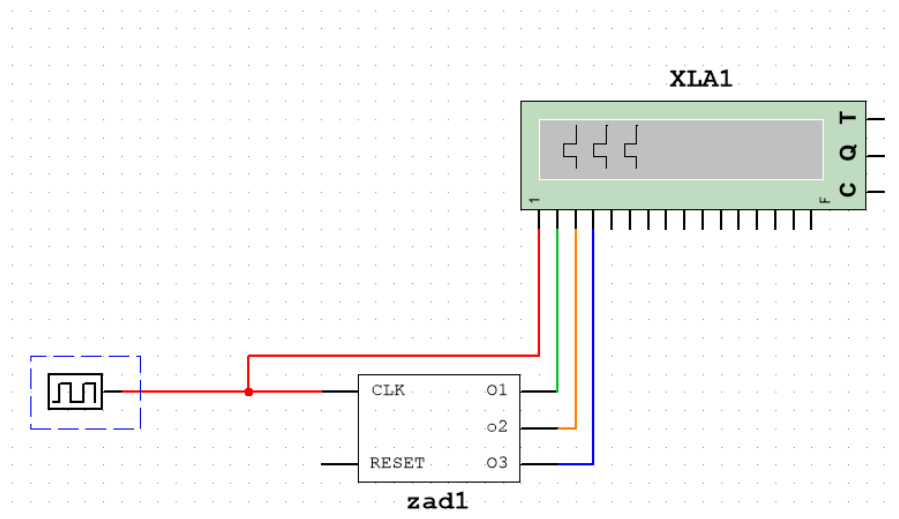
Rysunek 18: Analizator dla układu testującego



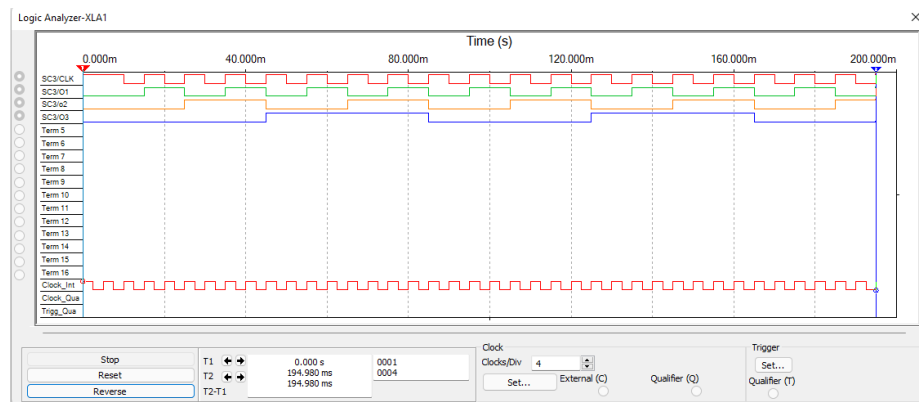
Rysunek 19: Ustawienia analizatora

1.4 Wnioski

- Liczniki asynchroniczne mają ograniczone zastosowanie, ponieważ, w związku z tym, że przerzutniki sterowane są wyjściami przerzutników poprzedzających, stan nie ustala się od razu, więc jeżeli impulsy zegarowe mają dużą częstotliwość i ich okres jest porównywalny z czasem propagacji przerzutnika, to sygnały wyjściowe licznika mogą podawać złe wartości. Rozwiązaniem tego problemu są liczniki synchroniczne.
- Dwójki liczącej oraz liczników w ogólności można użyć do zmniejszania częstotliwości zegara (jeżeli dwójka reaguje na zmianę z 0 na 1, wówczas częstotliwość na wyjściu zmniejszona jest dwukrotnie).



Rysunek 20: Układ zmniejszający częstotliwość

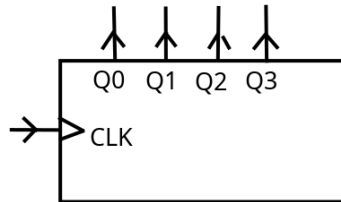


Rysunek 21: analizator układu zmniejszającego częstotliwość

Jak widać, częstotliwość na wyjści najmniej znaczącego bitu jest zmniejszona dwukrotnie, na drugim bicie czterokrotnie itd.

2 Ćwiczenie 3b

Ideą ćwiczenia jest, bazując na dowolnie wybranych przerzutnikach, zaprojektowanie, zbudowanie i przetestowanie synchronicznego czterobitowego licznika liczącego w kodzie Graya.



Rysunek 22: Synchroniczny licznik czterobitowy w kodzie Graya

2.1 Rozwiązanie teoretyczne

Wszystkie układy są synchroniczne i zakładają zmianę stanu przy zmianie sygnału zegarowego.

Licznik powinien liczyć w kodzie Graya, co oznacza, że dwa kolejne stany różnią się dokładnie jednym bitem:

Q3	Q2	Q1	Q0				
0	0	0	0	1	1	0	0
0	0	0	1	1	1	0	1
0	0	1	1	1	1	1	1
0	0	1	0	1	1	1	0
0	1	1	0	1	0	1	0
0	1	1	1	1	0	1	1
0	1	0	1	1	0	0	1
0	1	0	0	1	0	0	0

Rysunek 23: Kolejne stany licznika

W implementacji użyte zostaną przerzutniki D:

D	Q
1	1
0	0

Rysunek 24: Tabela przejść dla przerzutnika D

Wyjście przerzutnika D_i odpowiada Q_i :

				T_{+1}							
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	1	0	0	1	1
0	0	1	1	0	0	1	0	0	0	1	0
0	0	1	0	0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	1	0	1	1	1
0	1	1	1	0	1	0	1	0	1	0	1
0	1	0	1	0	1	0	0	0	1	0	0
0	1	0	0	1	1	0	0	1	1	0	0
1	1	0	0	1	1	0	1	1	1	0	1
1	1	0	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	0	1	1	1	0
1	1	1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	1	1	0	1	1
1	0	1	1	1	0	0	1	1	0	0	1
1	0	0	1	1	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0

Rysunek 25: Tabela prawdy dla licznika

Na podstawie tabeli prawdy licznika można skonstruować tabele Karnaugh oraz wzory dla D_0 , D_1 , D_2 , D_3 :

D0	Q1Q0	00	01	11	10
	Q3Q2	00	01	11	10
	00	1	1	0	0
	01	0	0	1	1
	11	1	1	0	0
10	0	0	1	1	

Rysunek 26: Tabela Karnaugh dla D_0

$$D_0 = \overline{Q_3} \overline{Q_2} \overline{Q_1} + \overline{Q_3} Q_2 Q_1 + \overline{Q_3} Q_2 \overline{Q_1} + \overline{Q_3} \overline{Q_2} Q_1$$

Ten wzór da się uprościć. Korzystając z rozdzielności koniunkcji względem alternatywy

$$D_0 = \overline{Q_3}(\overline{Q_2} \overline{Q_1} + Q_2 Q_1) + \overline{Q_3}(Q_2 \overline{Q_1} + \overline{Q_2} Q_1)$$

Tutaj natomiast można zauważyć, że zawartość pierwszego nawiasu to wzór na XNOR, a drugiego na XOR, więc

$$D_0 = \overline{Q_3}(\overline{Q_1 \oplus Q_2}) + \overline{Q_3}(Q_1 \oplus Q_2)$$

Ponownie, jest to wzór na XNOR Q_3 oraz zawartości nawiasu

$$D_0 = \overline{Q_1 \oplus Q_2 \oplus Q_3}$$

D1

Q1Q0 \ Q3Q2	00	01	11	10
00	0	1	1	1
01	0	0	0	1
11	0	1	1	1
10	0	0	0	1

Rysunek 27: Tabela Karnough dla D_1

$$D_1 = Q_1 \overline{Q_0} + \overline{Q_3} \overline{Q_2} Q_0 + Q_3 Q_2 Q_0$$

Tutaj można skorzystać z prawa rozdzielności koniunkcji względem alternatywy

$$D_1 = Q_1 \overline{Q_0} + Q_0 (\overline{Q_2} \overline{Q_3} + Q_2 Q_3)$$

Zawartość nawiasu to XNOR

$$D_1 = Q_1 \overline{Q_0} + Q_0 (\overline{Q_2 \oplus Q_3})$$

D2

Q1Q0 \ Q3Q2	00	01	11	10
00	0	0	0	1
01	1	1	1	1
11	1	1	1	0
10	0	0	0	0

Rysunek 28: Tabela Karnough dla D_2

$$D_2 = \underline{Q_1 \overline{Q_0} \overline{Q_3}} + \underline{Q_0 Q_2} + \underline{\overline{Q_1} Q_2}$$

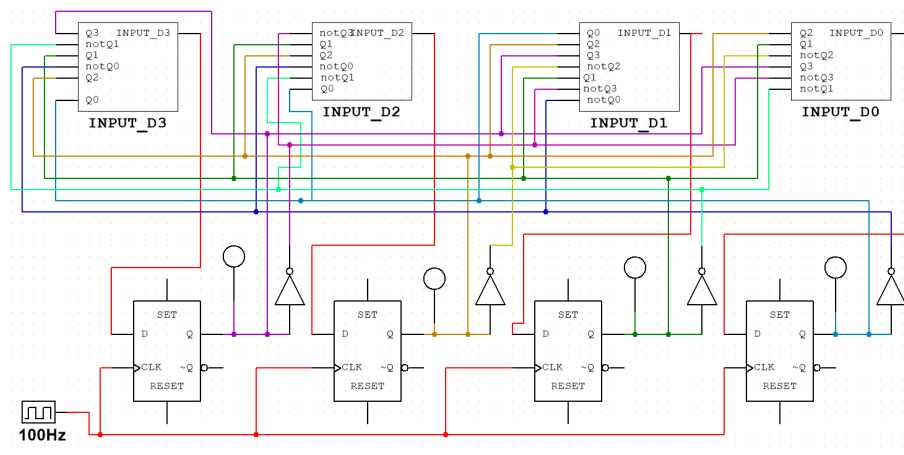
D3

Q1Q0 \ Q3Q2	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	1	1
10	0	1	1	1

Rysunek 29: Tabela Karnough dla D_3

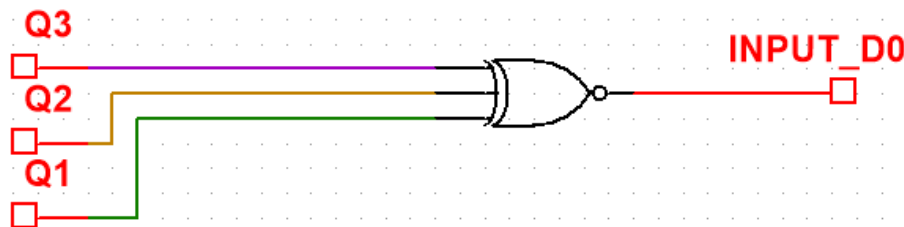
$$D_3 = \underline{Q_1 Q_3} + \underline{Q_3 \overline{Q_1} Q_0} + \underline{Q_2 \overline{Q_1} \overline{Q_0}}$$

2.2 Implementacja w programie Multisim



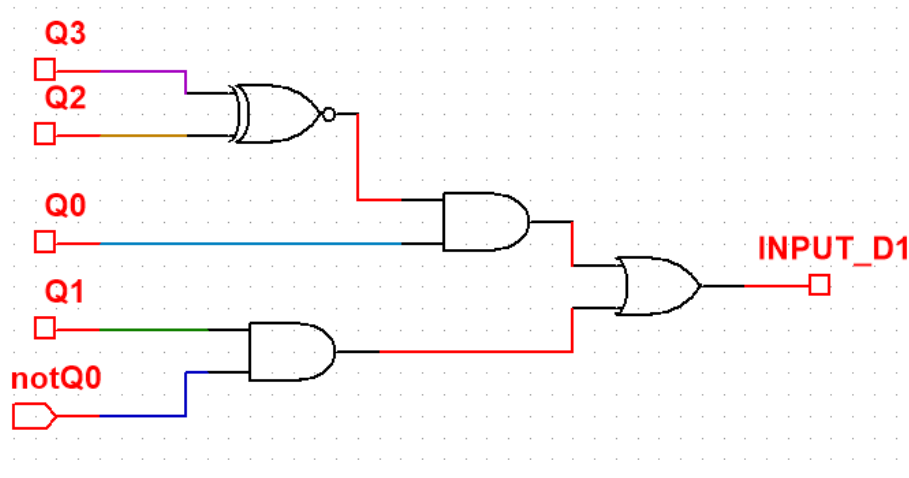
Rysunek 30: Implementacja licznika

$$D_0 = \overline{Q_1 \oplus Q_2 \oplus Q_3}$$



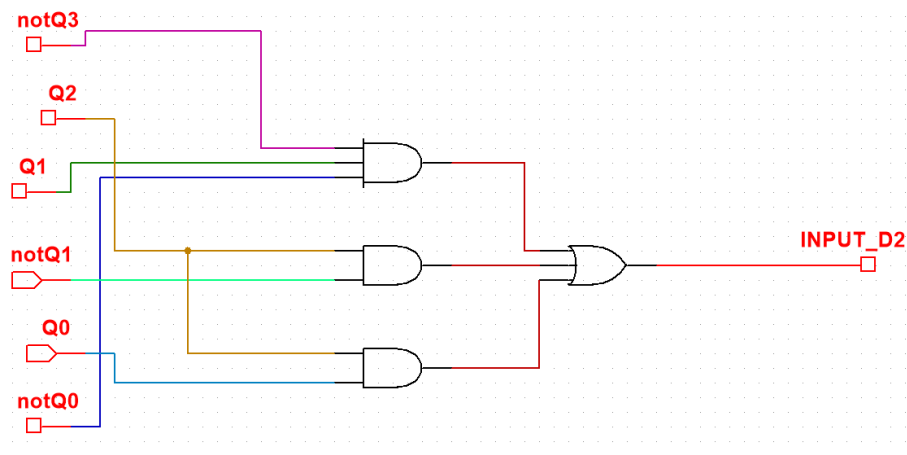
Rysunek 31: Podukład 0

$$D_1 = Q_1 \overline{Q_0} + Q_0 (\overline{Q_2} \oplus Q_3)$$



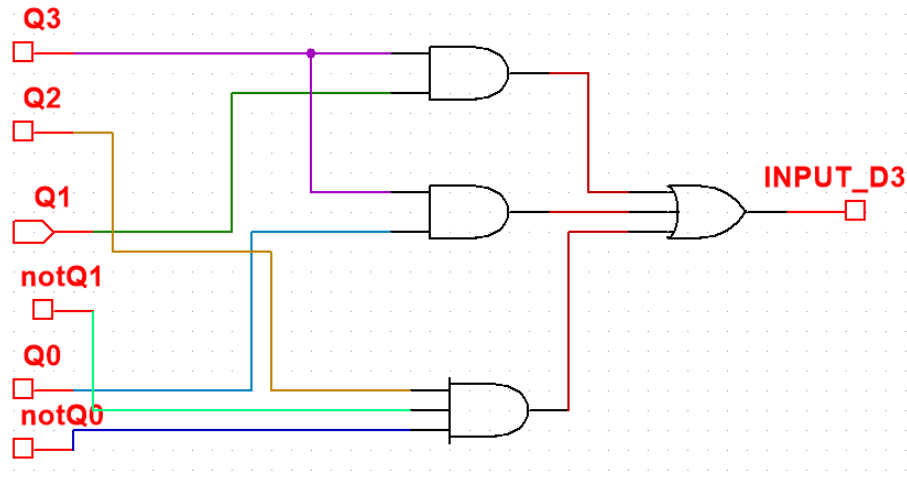
Rysunek 32: Podukład 1

$$D_2 = Q_1 \overline{Q_0} \overline{Q_3} + Q_0 Q_2 + \overline{Q_1} Q_2$$



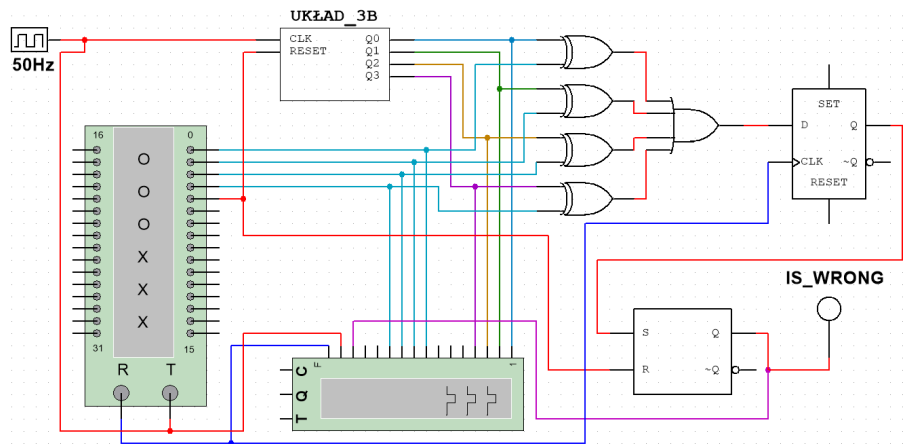
Rysunek 33: Podukład 2

$$D_3 = Q_1 Q_3 + Q_3 \overline{Q_1} Q_0 + Q_2 \overline{Q_1} \overline{Q_0}$$

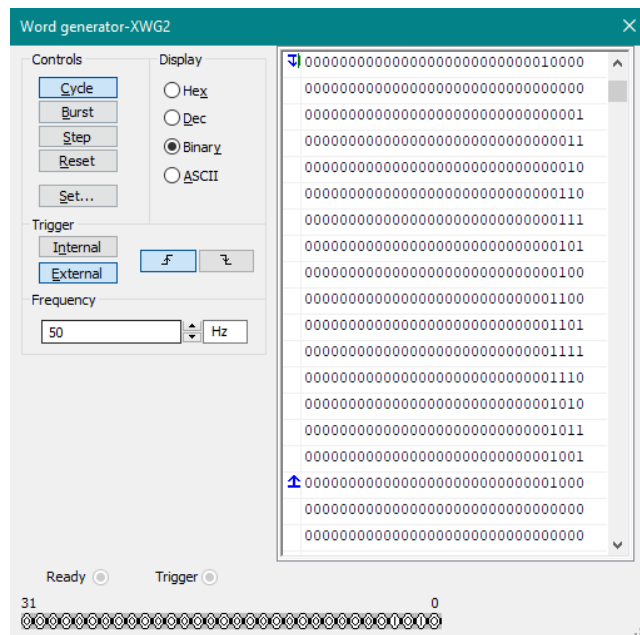


Rysunek 34: Podukład 3

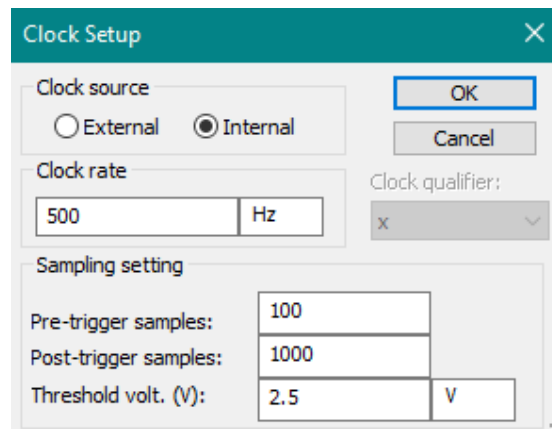
Poniżej układ testujący



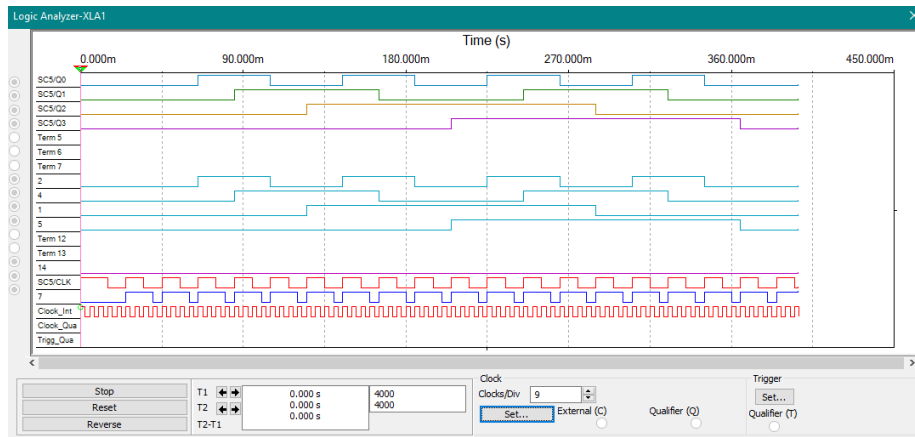
Rysunek 35: Układ testujący



Rysunek 36: Generator słów



Rysunek 37: Opcje analizatora



Rysunek 38: Analizator stanów logicznych

2.3 Wnioski

- Alternatywnym podejściem mogłoby być wykorzystanie innego typu przerzutników np. przerzutników JK (wówczas trzeba by jednak liczyć osiem zamiast czterech funkcji, bo bramka JK ma 2 wejścia) lub przerzutników typu T, które wydają się bardziej adekwatne. Podejście z przerzutnikiem T wymagałoby skonstruowania funkcji T_0, T_1, T_2, T_3 , np. tabela Karnough dla T_0 :

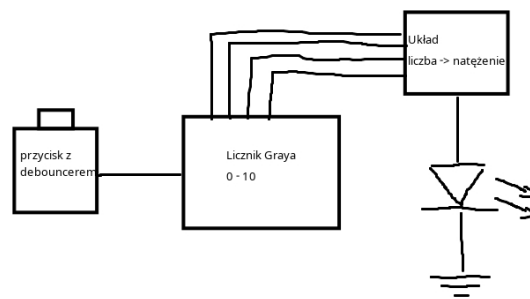
Q1Q0 \ Q3Q2	00	01	11	10
00	1	0	1	0
01	0	1	0	1
11	1	0	1	0
10	0	1	0	1

Rysunek 39: Tabela Karnough dla T_0

$$T_0 = \overline{Q_3} \overline{Q_2} \overline{Q_1} \overline{Q_0} + \overline{Q_3} \overline{Q_2} Q_1 Q_0 + \dots + Q_3 \overline{Q_2} Q_1 \overline{Q_0}$$

Jak widać, nie można zaznaczyć żadnej grupy większej niż jedna wartość, więc funkcja T_0 składałaby się z sumy ośmiu elementów, co wymaga większej liczby bramek i jest bardziej podatne na błędy w implementacji.

- Licznik synchroniczny w kodzie Graya, podobnie jak zwyczajny licznik binarny, może służyć jako dzielnik częstotliwości, jednak w tym przypadku częstotliwość sygnału na wyjściu najmniej znaczącego bitu będzie czterokrotnie mniejsza niż na wejściu, na wyjściu drugiego bitu ośmiokrotnie mniejsza itd.
- Przypuśćmy, że układ korzystający z licznika Graya służy do sterowania prądem przepływającym przez np. diodę LED. Użytkownik może naciskać przycisk tak żeby zwiększyć jasność diody (natężenie na diodzie = wyjście z licznika \cdot pewne natężenie bazowe). Po tym gdy użytkownik znajdzie się w najwyższej pozycji licznika (np. 10) licznika wraca do pozycji minimalnej.

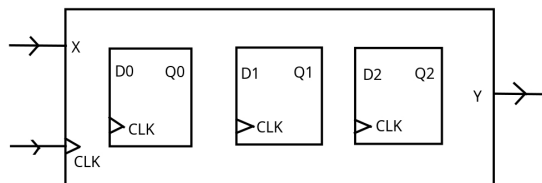


Rysunek 40: Przykład - układ sterujący jasnością diody

Gdyby używać licznika binarnego, wówczas istnieje ryzyko, że podczas zmiany stanów dioda chwilowo znajdzie się w stanie zabronionym (np. 0111 \rightarrow 1000 znajdzie się w stanie 1111), na diodę podane zostanie natężenie większe niż może przyjąć i dioda się przepali. Przy użyciu licznika w kodzie Graya podobne zagrożenie nie istnieje.

3 Ćwiczenie 3c

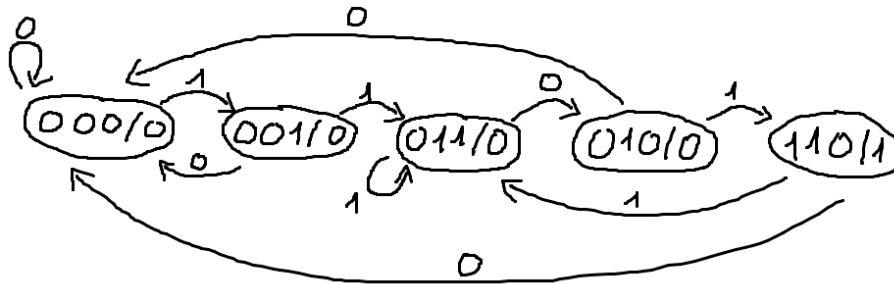
Ideą ćwiczenia jest, bazując na przerzutnikach "D", zaprojektowanie, zbudowanie i przetestowanie automatu realizującego detekcję litery Q przekazywanej alfabetem Morsa, czyli sekwencji bitów: "— — • —". Za kreskę należy przyjąć stan logiczny '1', a za kropkę stan logiczny '0'. Należy również zaproponować własny, ale skuteczny i praktyczny sposób wprowadzania do urządzenia sygnału wejściowego.



Rysunek 41: Układ 3c

3.1 Rozwiązanie teoretyczne

Sekwencja, którą trzeba wykryć, to "1101", na tej podstawie zaprojektować można automat wykrywający owe słowo:



Rysunek 42: Automat wykrywający słowo "1101"

Na podstawie automatu można stworzyć tabelę kolejnych stanów oraz tabelę dla Y:

X				
STAN	0	1	STAN	Y
000	000	001	000	0
001	000	011	001	0
011	010	011	011	0
010	000	110	010	0
110	000	011	110	1
111	—	—	111	—
101	—	—	101	—
100	—	—	100	—

Rysunek 43: Tabela kolejnych stanów automatu oraz tabela Y

Pozostaje skonstruować tabele Karnough ($STAN = Q_2Q_1Q_0$):

D_0 :

$Q_2Q_1 \backslash Q_0X$	00	01	11	10
00	0	1	1	0
01	0	0	1	0
11	0	1	—	—
10	—	—	—	—

Rysunek 44: Tabela Karnough dla D_0

$$D_0 = \underline{Q_0}X + \underline{X}\overline{Q_1} + \underline{X}Q_2$$

$D_1:$

$\begin{matrix} Q_0 X \\ Q_2 Q_1 \end{matrix}$	00	01	11	10
00	0	0	1	0
01	0	1	1	1
11	0	1	-	-
10	-	-	-	-

Rysunek 45: Tabela Karnough dla D_1

$$D_1 = \underline{Q_0 X} + \underline{X Q_1} + \underline{Q_0 Q_1}$$

$D_2:$

$\begin{matrix} Q_0 X \\ Q_2 Q_1 \end{matrix}$	00	01	11	10
00	0	0	0	0
01	0	1	0	0
11	0	0	-	-
10	-	-	-	-

Rysunek 46: Tabela Karnough dla D_2

$$D_2 = \underline{\overline{Q_0} X \overline{Q_2} Q_1}$$

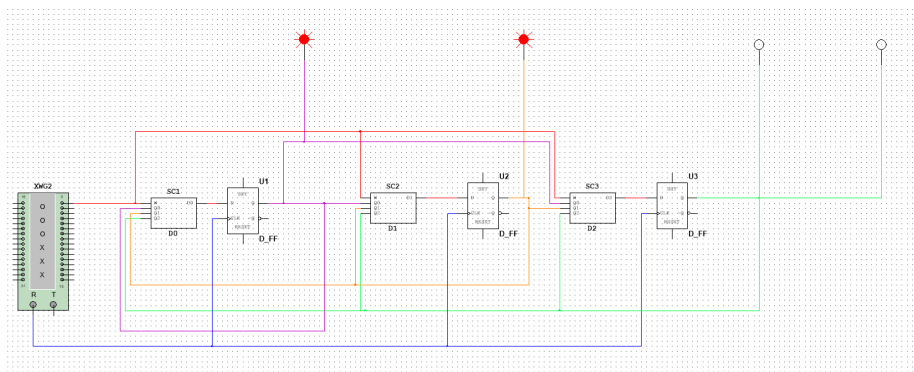
Oraz tabela Karnough dla Y:

$Q_2 Q_1$ \ Q_0	0	1
00	0	0
01	0	0
11	1	-
10	-	-

Rysunek 47: Tabela Karnough dla Y

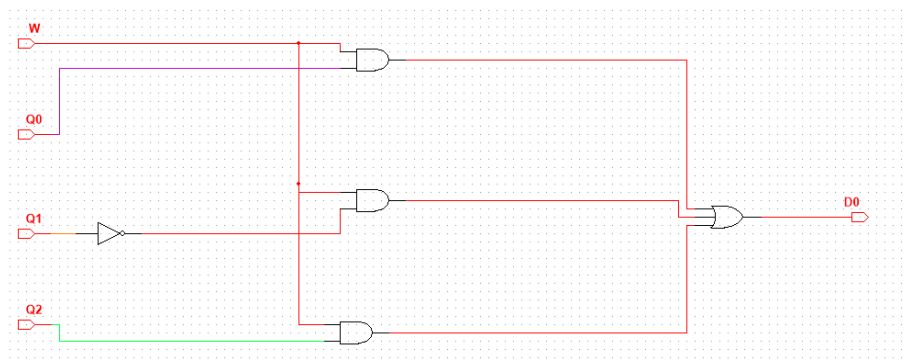
$$Y = \underline{Q_2}$$

3.2 Implementacja w programie Multisim



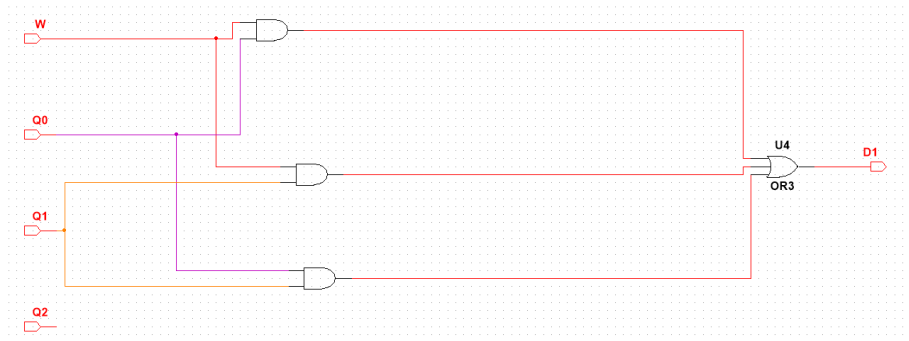
Rysunek 48: Implementacja w programie Multisim

$$D_0 = Q_0X + X\overline{Q_1} + XQ_2$$



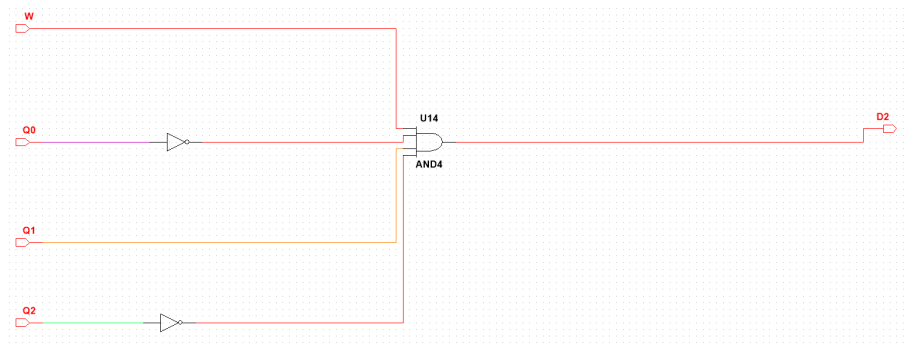
Rysunek 49: Podukład D_0

$$D_1 = Q_0X + XQ_1 + Q_0Q_1$$



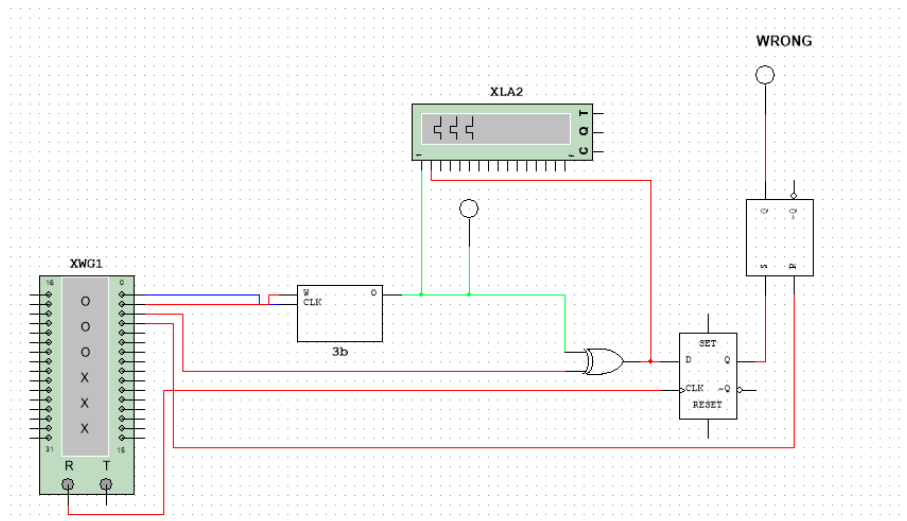
Rysunek 50: Podukład D_1

$$D_2 = \overline{Q_0}X\overline{Q_2}Q_1$$

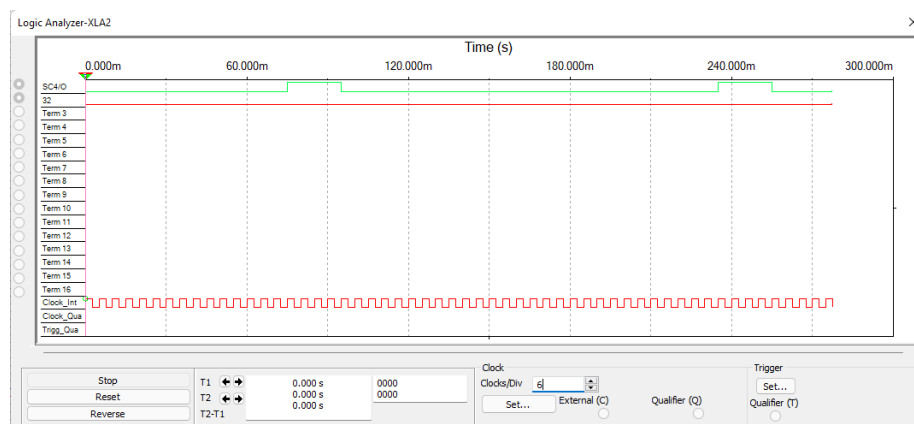


Rysunek 51: Podukład D_2

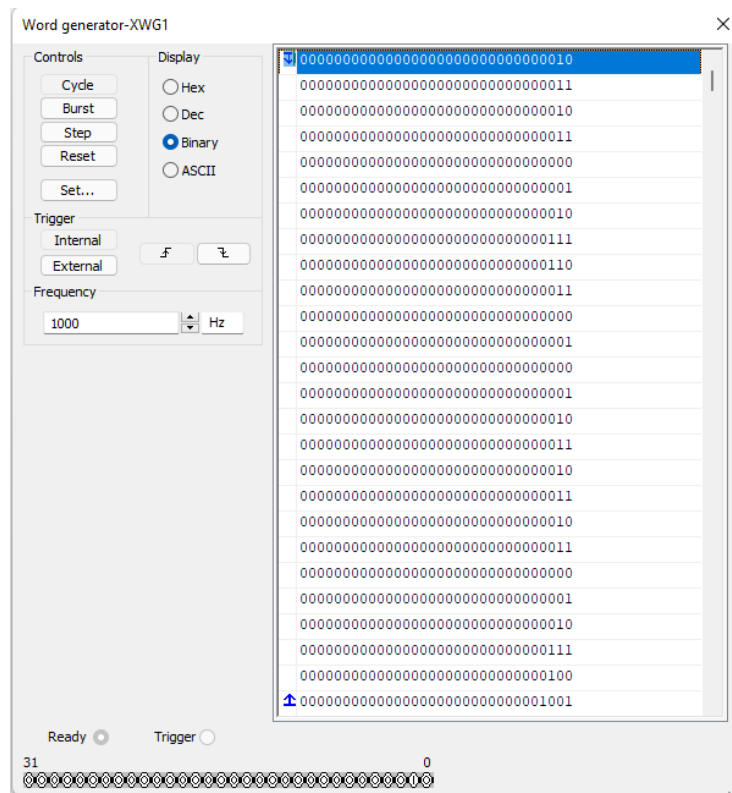
Poniżej układ testujący:



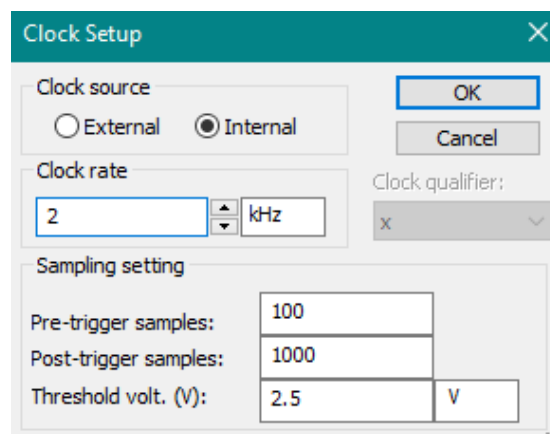
Rysunek 52: Układ testujący



Rysunek 53: Analizator stanów logicznych w układzie testującym



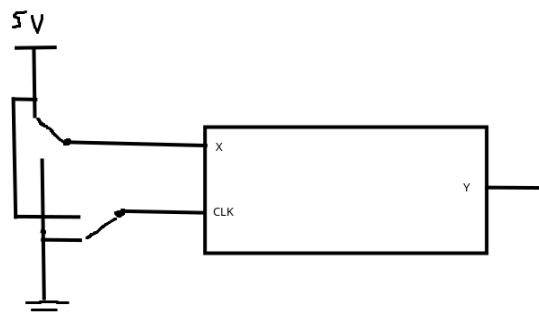
Rysunek 54: Generator słów w układzie testujm



Rysunek 55: Ustawienia analizatora stanów logicznych

3.3 Wnioski

- Prostym, a praktycznym sposobem podawania sygnału wejściowego byłoby zastosowanie przełącznika i przycisku: przełącznik określałby stan kolejnego symbolu (włączony - 1, wyłączony - 0), a drugi pełniłby rolę manualnego zegara.



Rysunek 56: Wprowadzanie sygnału

- W naszym podejściu kolejne stany oznaczone kolejnymi liczbami w kodzie Graya (przypadłość po ćwiczeniu 3b). Alternatywnie, można opisać by je kolejnymi liczbami zapisanymi binarnie (000, 001, 010, 011, 100). Wówczas tabele Karnough, a co za tym idzie, wzory byłyby inne niż obecnie.

X				
STAN	0	1	STAN	Y
000	000	001	000	0
001	000	010	001	0
010	010	011	010	0
011	000	100	011	0
100	000	010	100	1
101	-	-	101	-
110	-	-	110	-
111	-	-	111	-

Rysunek 57: Alternatywne oznaczenia stanów automatu

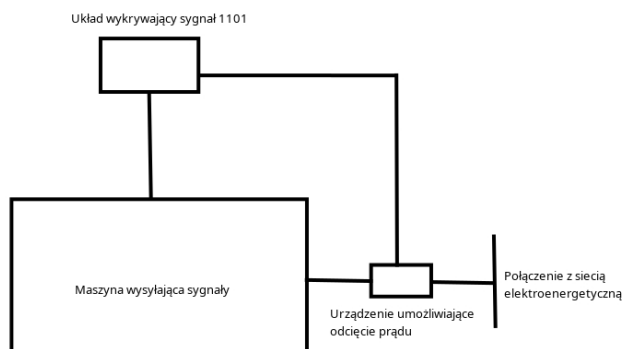
D_0 :

$Q_2 Q_1$ \ $Q_0 X$	00	01	11	10
00	0	1	0	0
01	0	1	0	0
11	-	-	-	-
10	0	0	-	-

Rysunek 58: Alternatywna tabela Karnough D_0

$$D_0 = \overline{Q_0} X \overline{Q_2}$$

- Hipotetycznym scenariuszem, w którym układ byłby przydatny, to wykrywanie sygnału wysłanego przez pewną maszynę. Przypuśćmy, że maszyna może wysyłać różne sygnały, np. o tym, że kończy się materiał, z którego korzysta. Jednym z sygnałów wysyłanych może być sygnał stop (1101), wysyłany w przypadku, gdy np. maszyna się przegrzewa. Układ wykrywa ten sygnał i odłącza maszynę od prądu.



Rysunek 59: Maszyna wysyłająca sygnały