

# **Sprawozdanie przerzutniki**

**Technika Cyfrowa 2021/22**

***Dominik Grzesik, Sebastian Kozak, Szymon Słota,  
Marcin Mikuła, Jakub Łubkowski***

# Spis treści

<b>Sprawozdanie przerzutniki</b>	<b>1</b>
1. Zadanie 2 a)	3
1.1. Wprowadzenie	3
1.2. Wyprowadzenie	4
1.3. Spostrzeżenia	7
1.4. Implementacja układu w programie Multisim	7
1.5. Generator słów	8
1.6. Implementacja układu testującego w programie Multisim	8
1.7. Analizator słów	9
1.8. Wnioski	10
1.9. Przykładowe zastosowania	10
2. Zadanie 2 b)	12
2.1. Wprowadzenie	12
2.2. Rejestr w programie Multisim	13
2.3. Rejestr w programie Multisim	
2.8. Wnioski	19
2.9. Przykładowe zastosowania	20

## 1. Zadanie 2 a)

Za pomocą dwóch bramek NAND zaprojektować oraz zbudować asynchroniczny przerzutnik RS.

### 1.1. Wprowadzenie



Rysunek 1. Poglądowy rysunek przerzutnika RS.

Właściwości bramki NAND:

A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

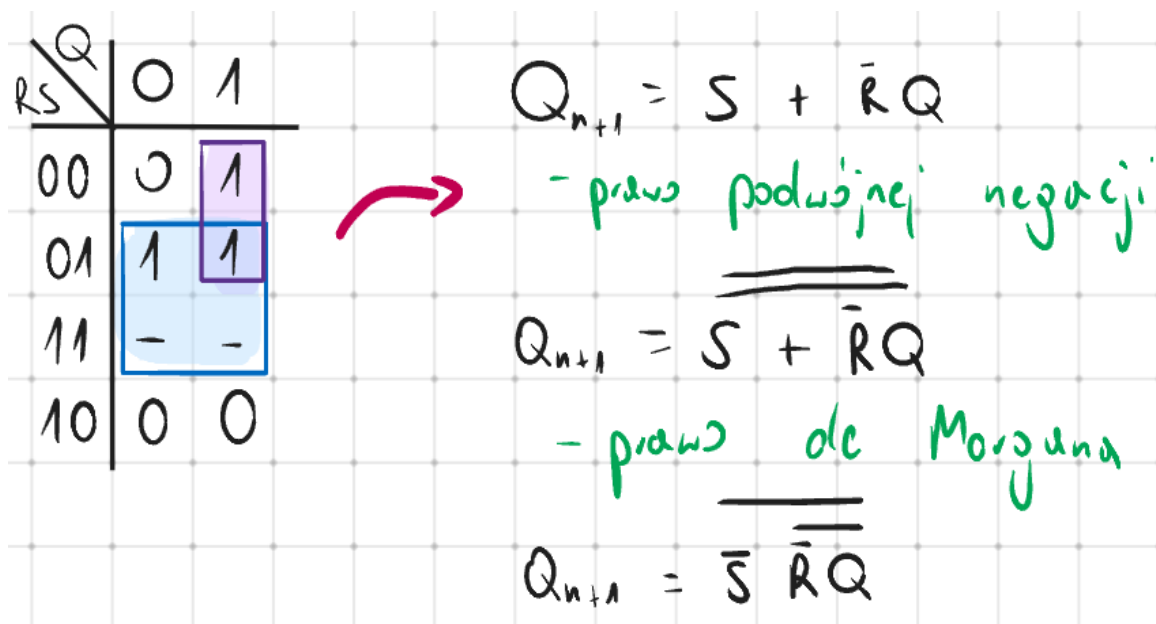
Tabela 1. Tablica bramki NAND.

## 1.2. Wyprowadzenie

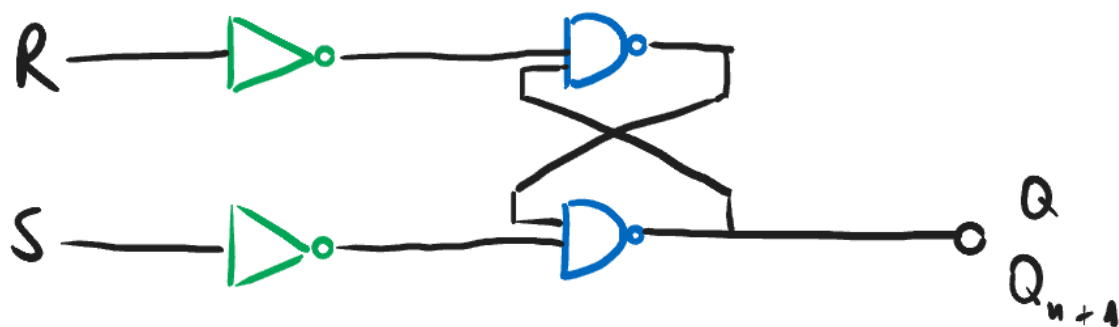
S	R	$Q_{n+1}$
0	0	$Q_n$
0	1	0
1	0	1
1	1	X

Rysunek 2. Tabela stanów dla przerzutnika RS.

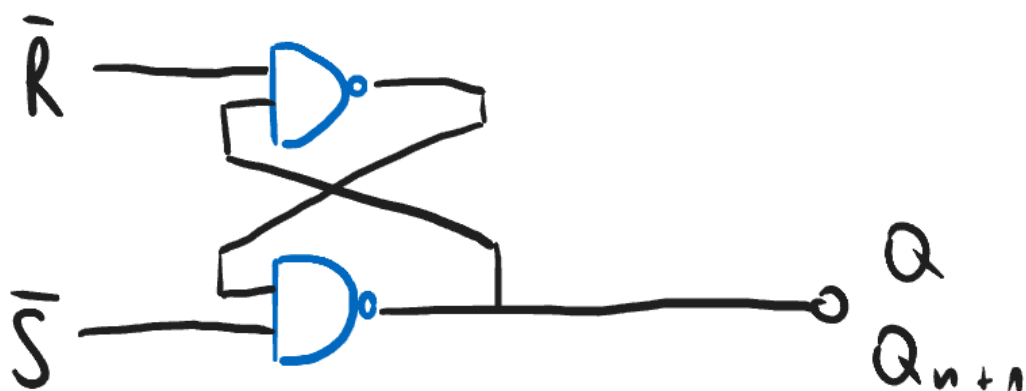
Analizując powyższą tabelę, możemy zauważyć, iż gdy  $S=R=1$  nie istnieje wartość  $Q_{n+1}$ . Jest to tzw. stan zabroniony, zakładamy, że on nie zachodzi oraz nie rozważamy wyjścia bramki.



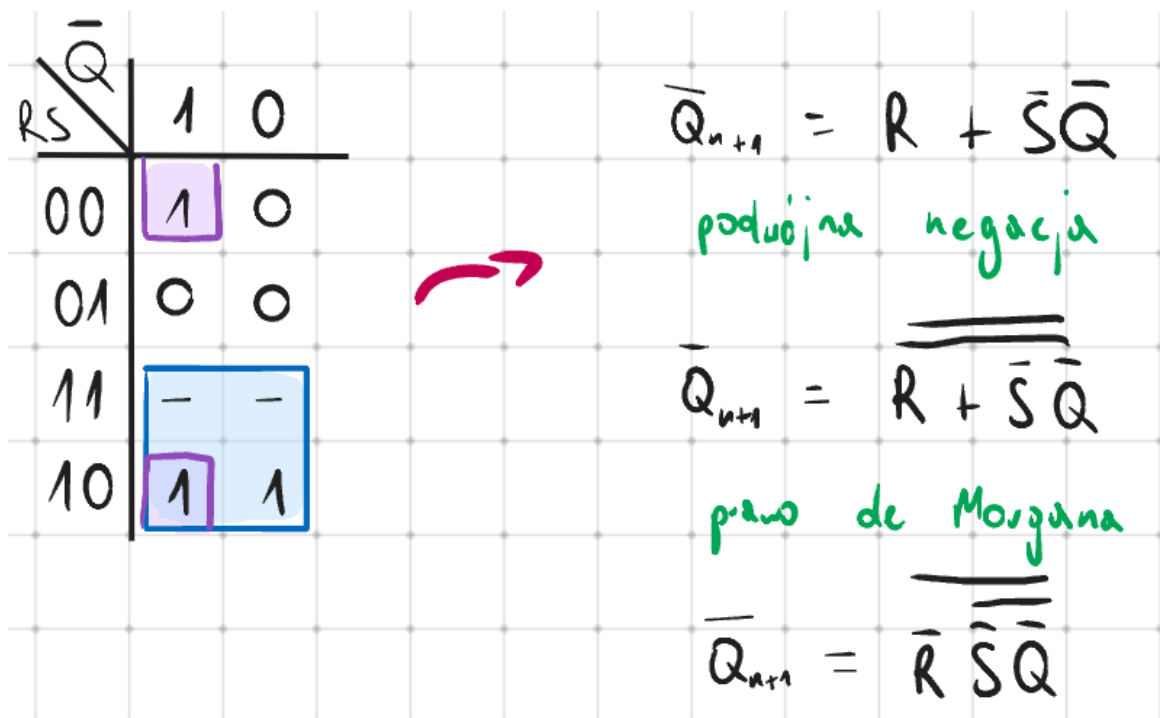
Rysunek 3. Tabela Karnaugh dla Q, oraz wyprowadzenie wzoru na Q.



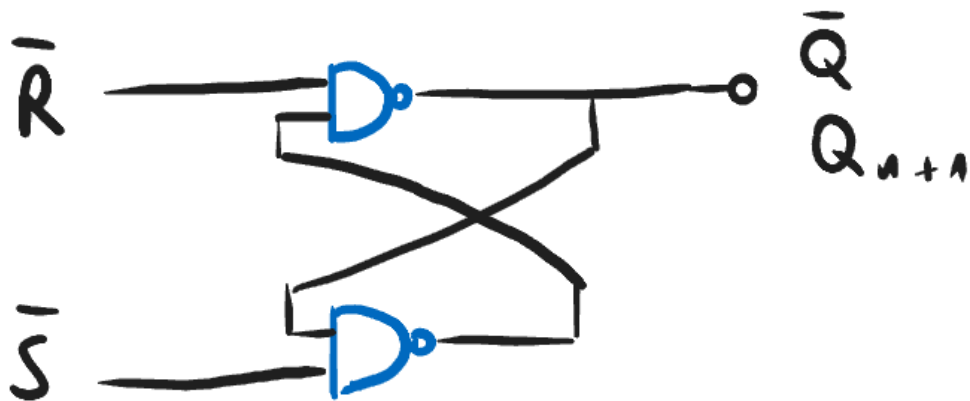
Rysunek 4. Poglądowy rysunek układu na podstawie funkcji logicznej Q.



Rysunek 5. Poglądowy rysunek bez bramek NOT z zanegowanymi wejściami.

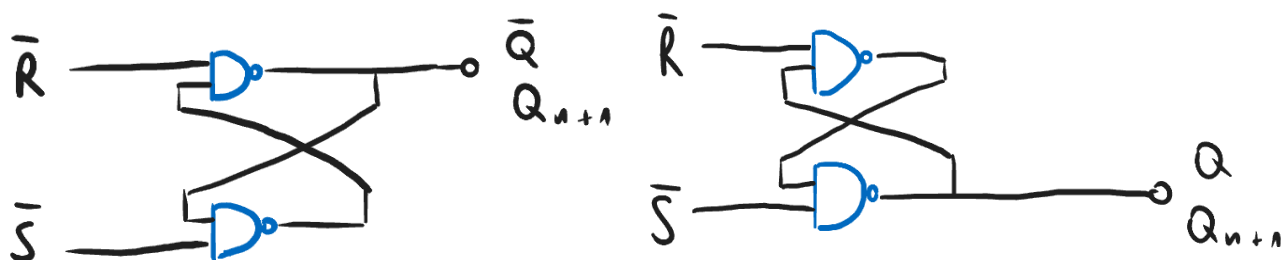


Rysunek 6. Tabela Karnaugh dla  $\bar{Q}$ , oraz wyprowadzenie wzoru na  $\bar{Q}$ .



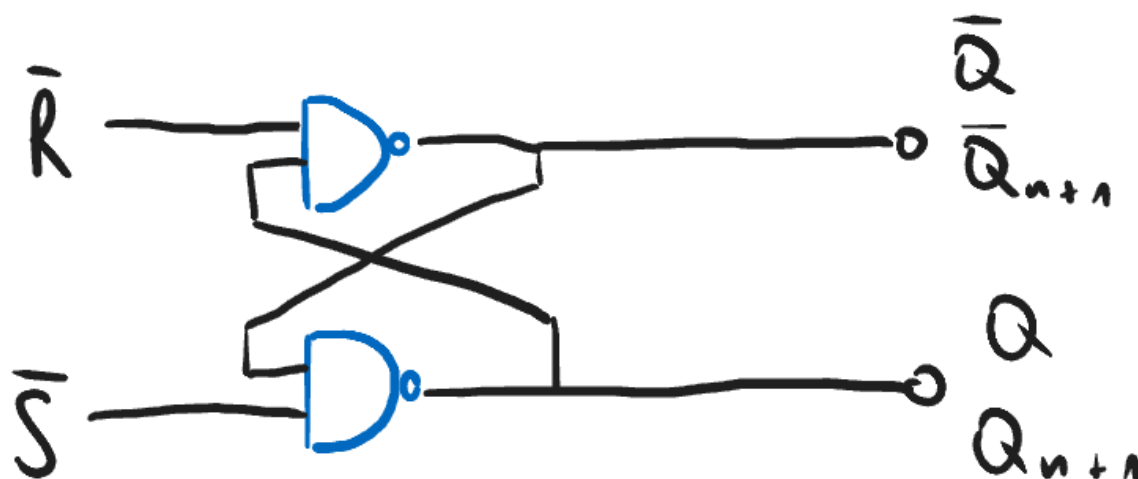
Rysunek 7. Poglądowy rysunek układu na podstawie funkcji logicznej  $\bar{Q}$ .

### 1.3 Spostrzeżenia



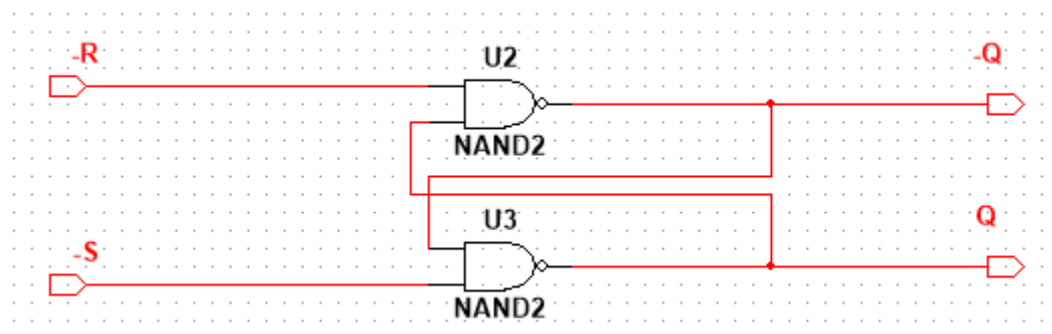
Rysunki 5 i 7.

Zauważamy, że oba układy powyżej są jednakowe, dzięki czemu możemy stworzyć jeden układ dla  $Q$  i  $\bar{Q}$ :



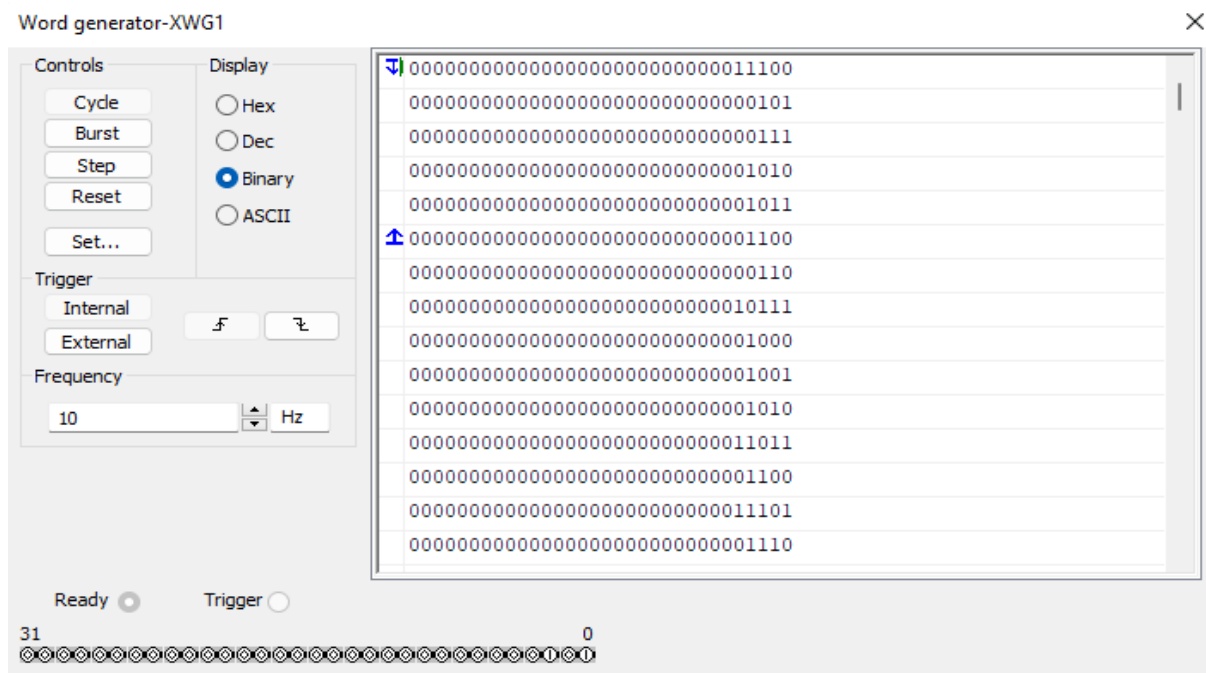
Rysunek 8. Poglądowy rysunek układu.

### 1.4 Implementacja układu z bramkami NAND w programie Multisim



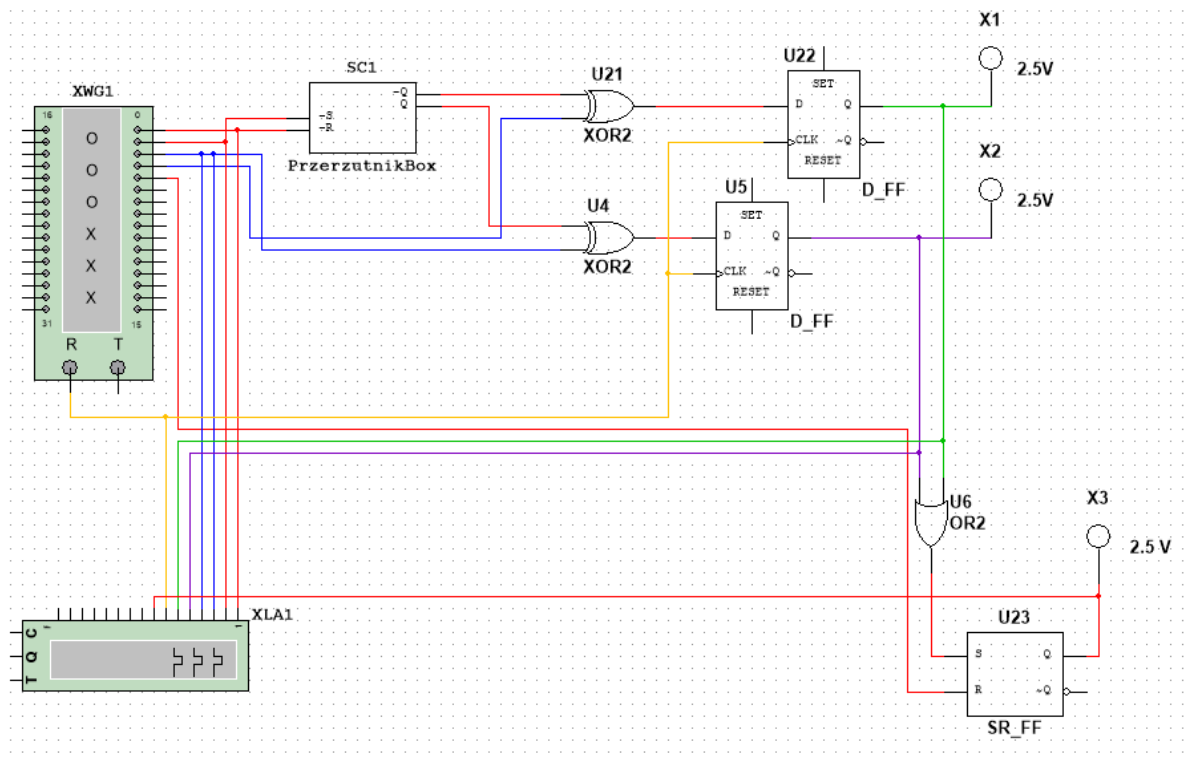
Rysunek 9. Zaprojektowany układ w Multisimie.

## 1.5 Generator słów



Rysunek 10. Generator słów.

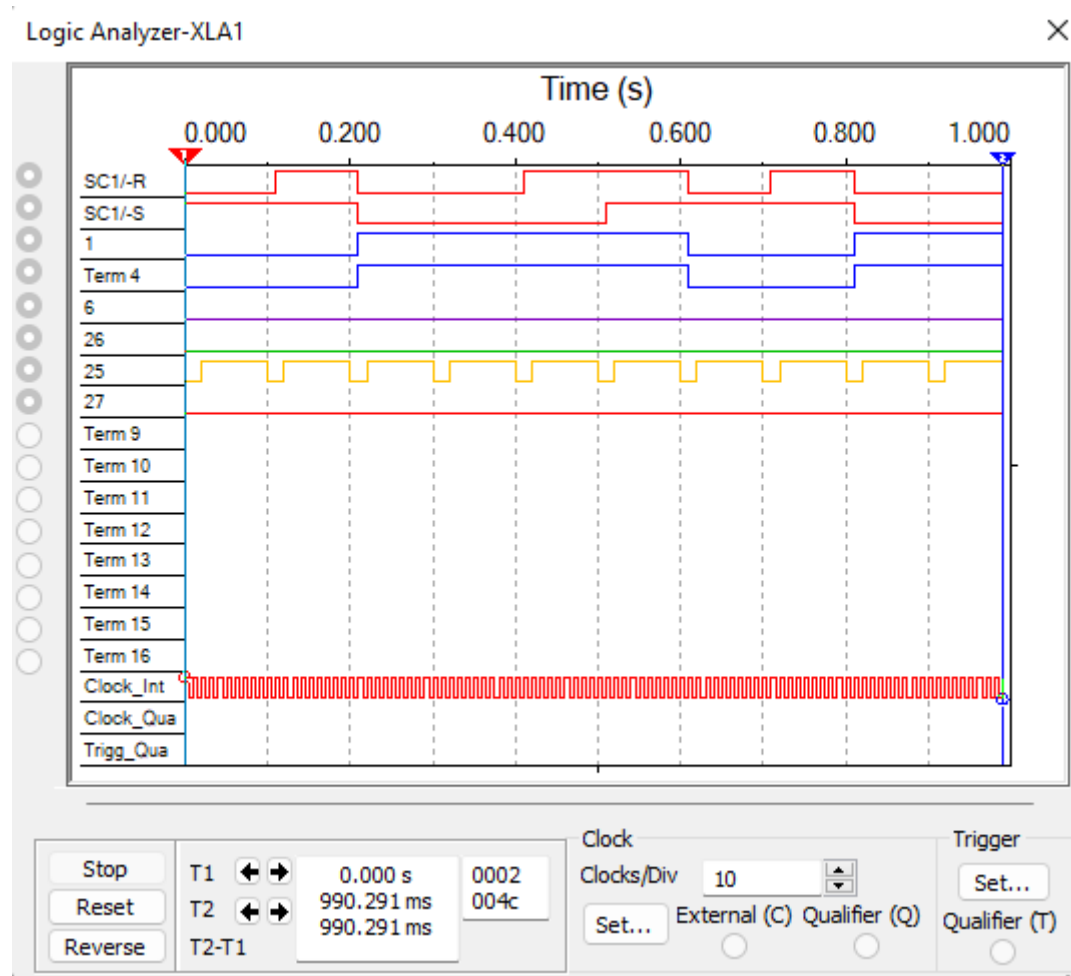
## 1.6 Implementacja układu testującego w programie Multisim



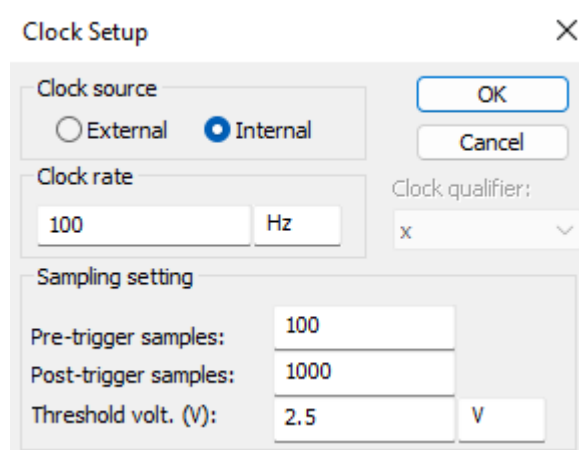
Rysunek 11. Zaprojektowany układ testowy w Multisimie



## 1.7 Analizator słów



Rysunek 12. Analizator słów



Rysunek 13. Ustawienia analizatora słów

## 1.8 Wnioski

1. Przerzutnik RS pozwala zapamiętywać informacje o poprzednim stanie.
2. Poprawne działanie przerzutnika RS zbudowanego z dwóch bramek NAND zapewnia nam zanegowanie wejścia R i S.
3. Zadanie można było zrealizować przy użyciu bramek NOR, lecz zgodnie z treścią polecenia skorzystaliśmy z bramek NAND.

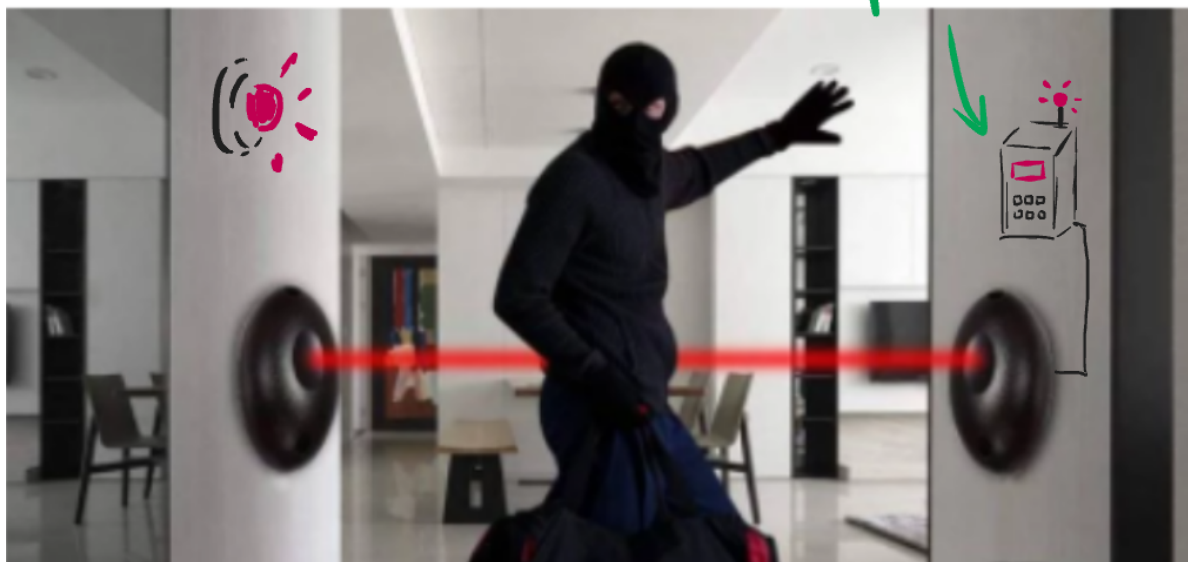
## 1.9 Przykładowe zastosowania

1. Przerzutnik RS można użyć do stworzenia włącznika w czajniku, która włącza się po wciśnięciu **start**, a wyłącza się po wciśnięciu **stop** lub gdy woda się zagotuje.



**Rysunek 14.** Zastosowanie przerzutnika RS w czajniku.

2. Alarm w banku w Szwajcarii, który zostaje aktywowany przez czujnik laserowy, a wyłączenie go jest możliwe tylko poprzez zresetowanie go 10-cyfrowym kodem przez dyrektora banku.



**Rysunek 15.** Zastosowanie przerzutnika RS w alarmie.

## 2. Zadanie 2 b)

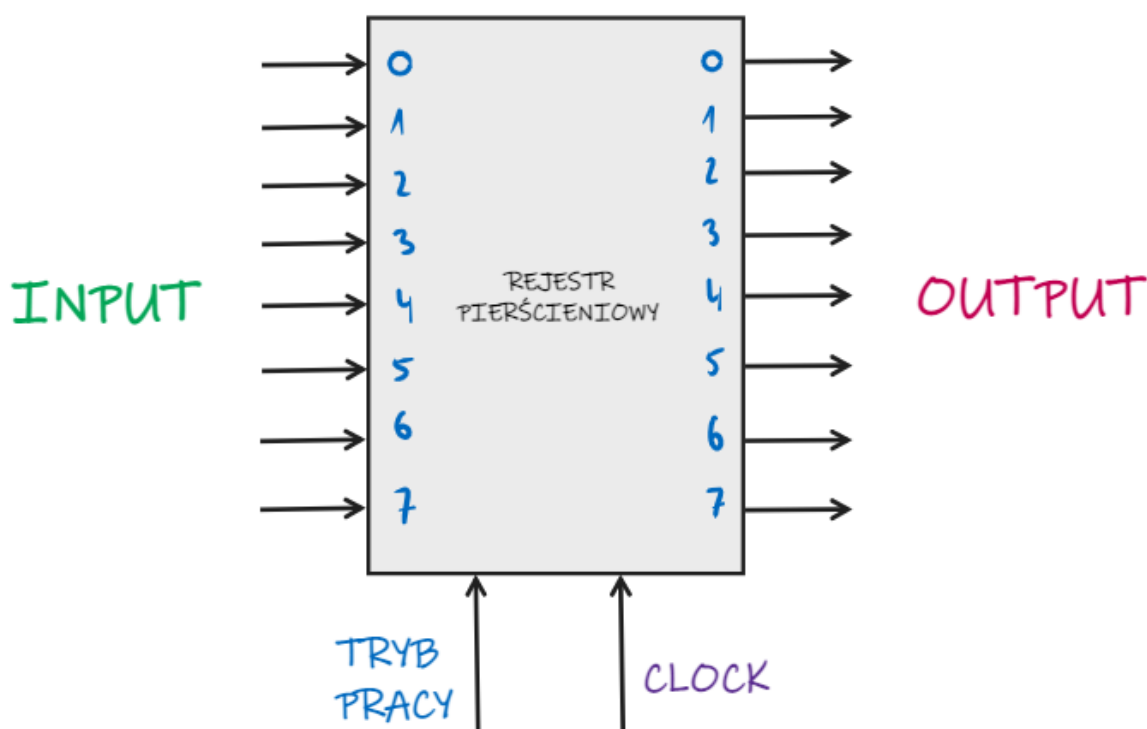
Za pomocą wybranych przerzutników, zbudować rejestr pierścieniowy. Rejestr powinien realizować dwie podstawowe funkcje, wybierane przy pomocy pojedynczego przełącznika posiadającego dwa tryby:

tryb 1 - ładowanie danych do rejestru.

tryb 2 - krążenie danych w rejestrze.

### 2.1. Wprowadzenie

Tworzymy układ przerzutników, rejestr pierścieniowy, jako zapętłony rejestr przesuwający:



Rysunek 16. Poglądowy rysunek rejestru pierścieniowego.

## 2.2 Wyprowadzenie

Do skonstruowania rejestru, ze względu na wygodę, użyliśmy przerzutników D, dzięki którym otrzymaliśmy oczekiwany rezultat. Na wyjściu Q każdego z przerzutników ustalamy wartość, która jest zapisana w komórce rejestru. Multiplexer z jednym bitem wyboru, decyduje jakie wartości będą wpisane w poszczególne komórki.

A	D <sub>0</sub>	D <sub>1</sub>	Y
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1

A \ D <sub>0</sub> D <sub>1</sub>	00	01	11	10
1	0	0	1	1
0	0	1	1	0

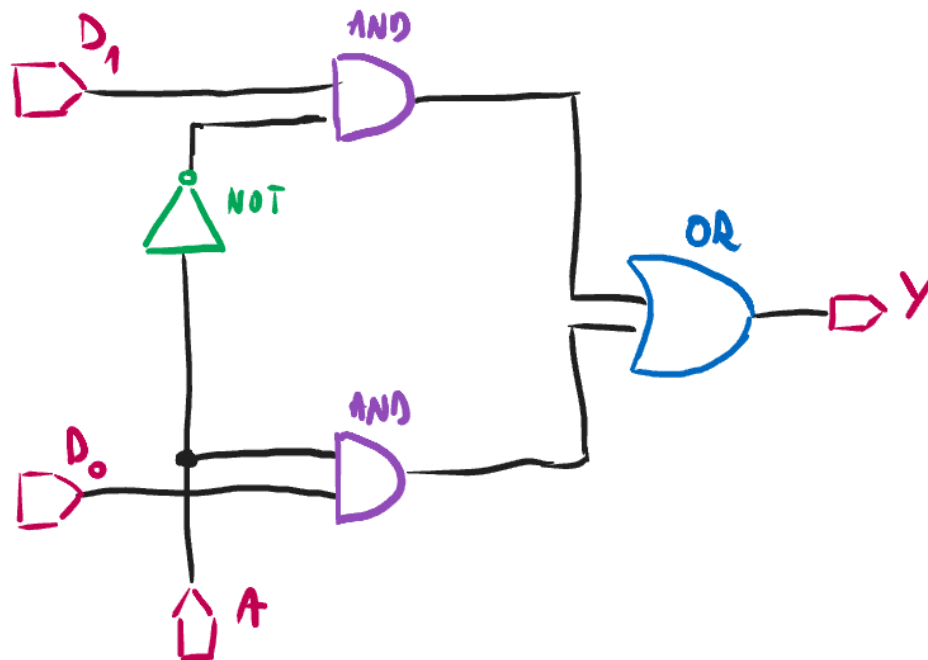
  

$$Y = \underline{AD_0} + \underline{\bar{A}D_1}$$

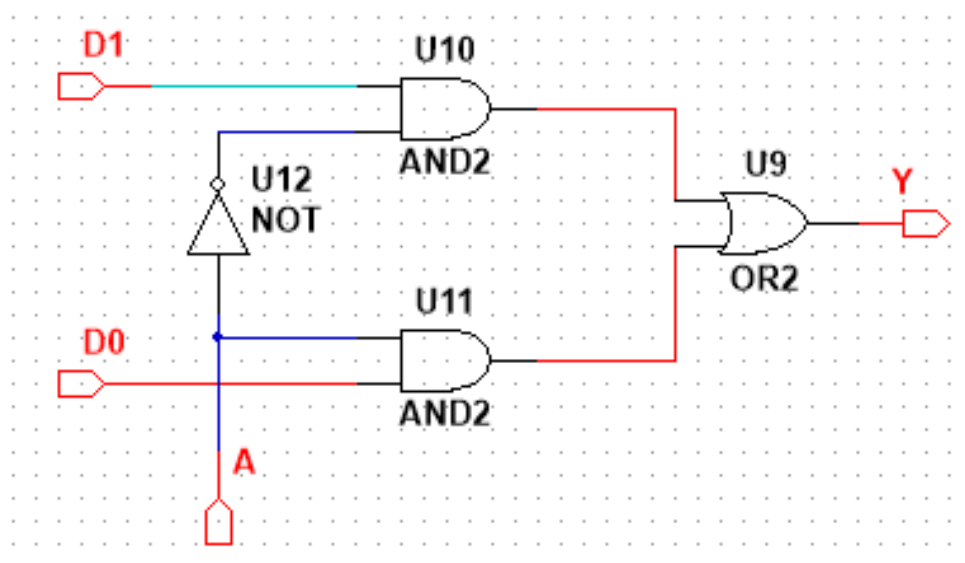
Rysunek 17. Działanie Multiplexera dla dwóch wejść.

$$Y = \underline{AD_0} + \underline{\bar{A}D_1}$$

Aby to zrealizować stworzyliśmy multiplexer z jednym bitem wyboru oraz dwoma bitami wejściowymi.

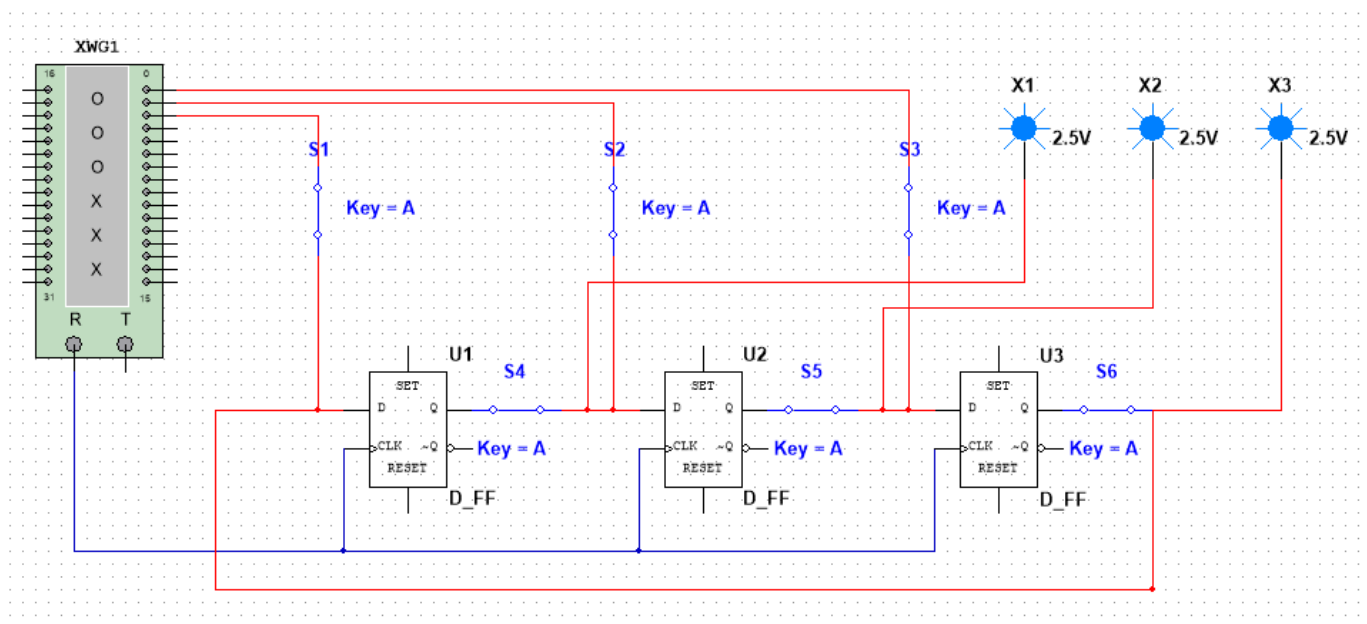


**Rysunek 18.** Poglądowy rysunek multiplexera.



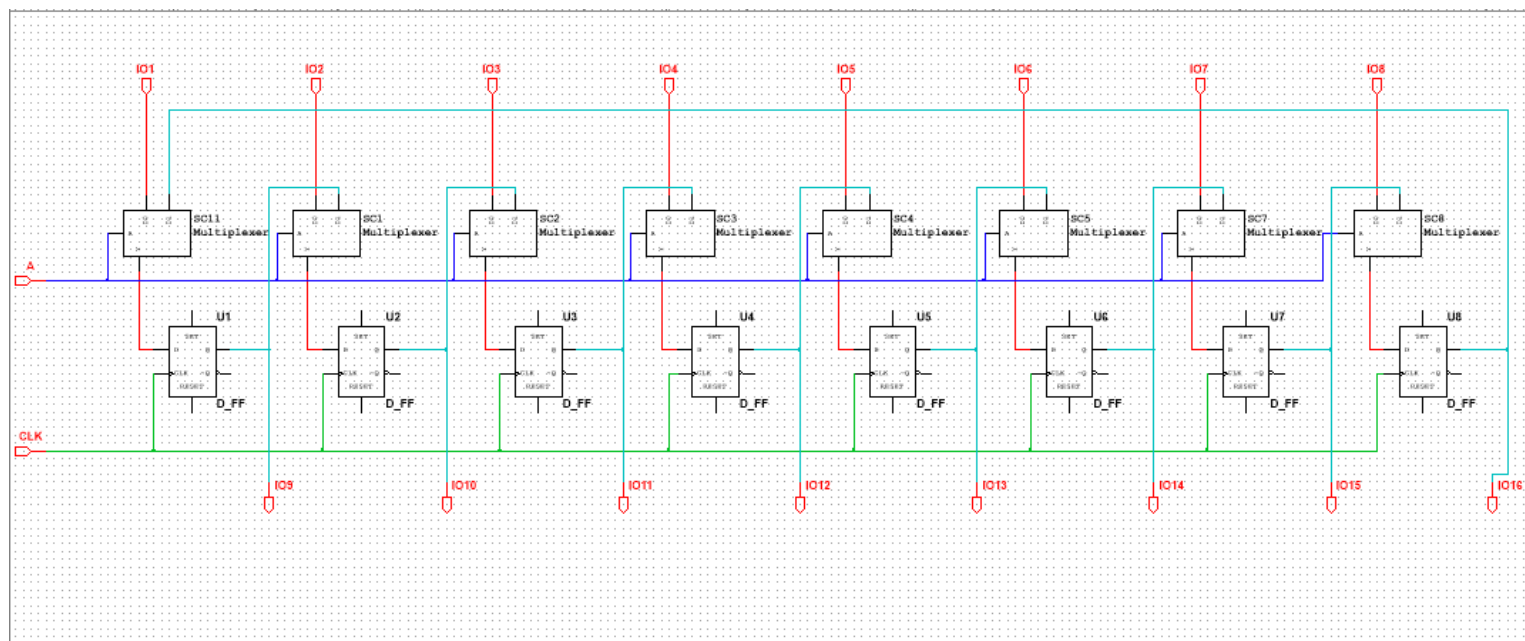
**Rysunek 19.** Multiplexer w multisimie.

## 2.3 Wstępny model



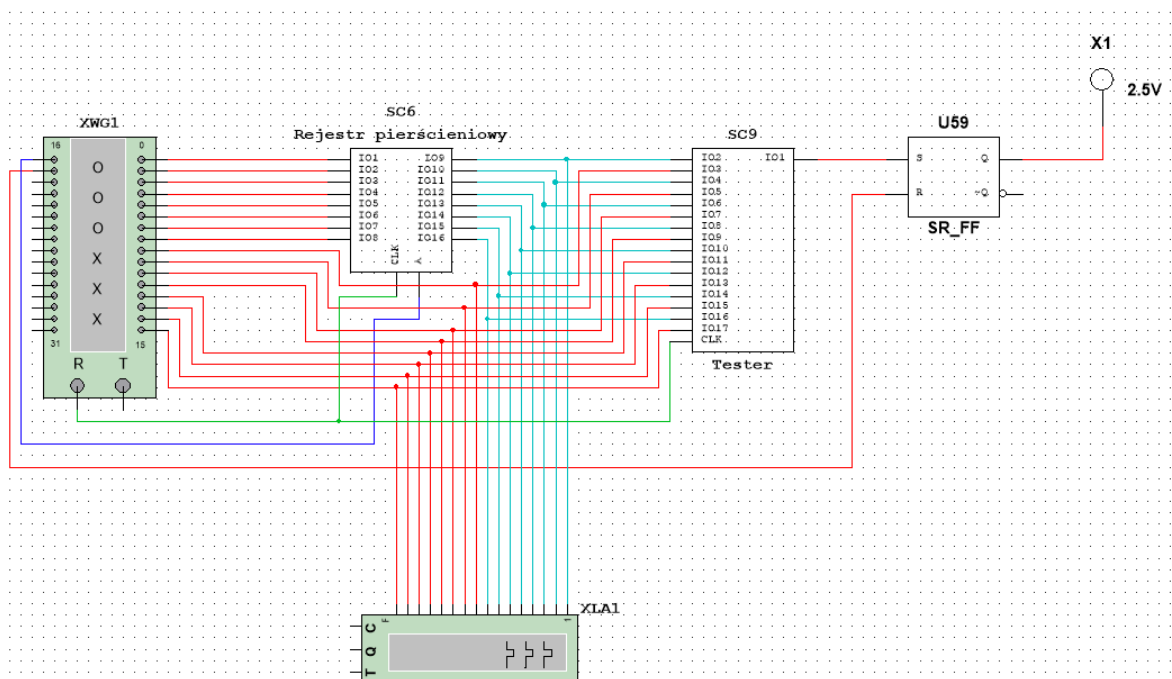
Rysunek 20. Wstępny rejestr pierścieniowy na mniejszą skalę w programie Multisim z wykorzystaniem przerzutników D.

## 2.4 Implementacja układu - rejestru pierścieniowy



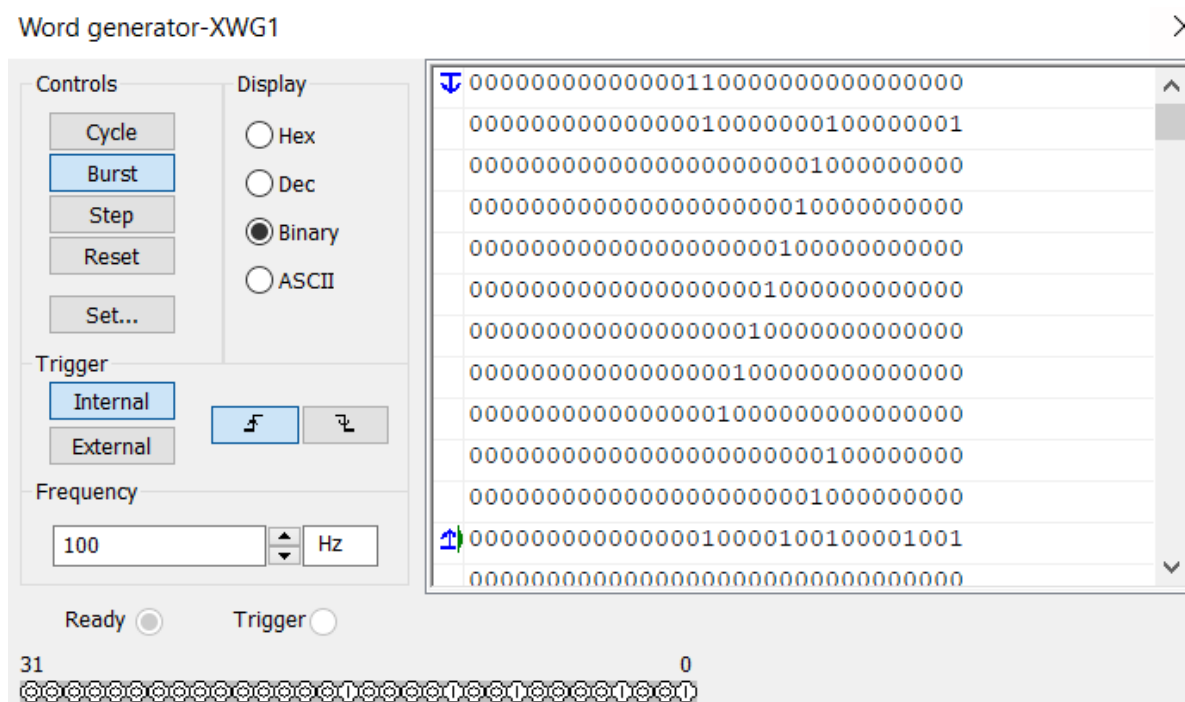
Rysunek 21. Rejestr pierścieniowy w programie Multisim.

## 2.5 Układ testujący rejestr pierścieniowy w programie Multisim



Rysunek 22. Implementacja rejestru pierścieniowego w programie Multisim..

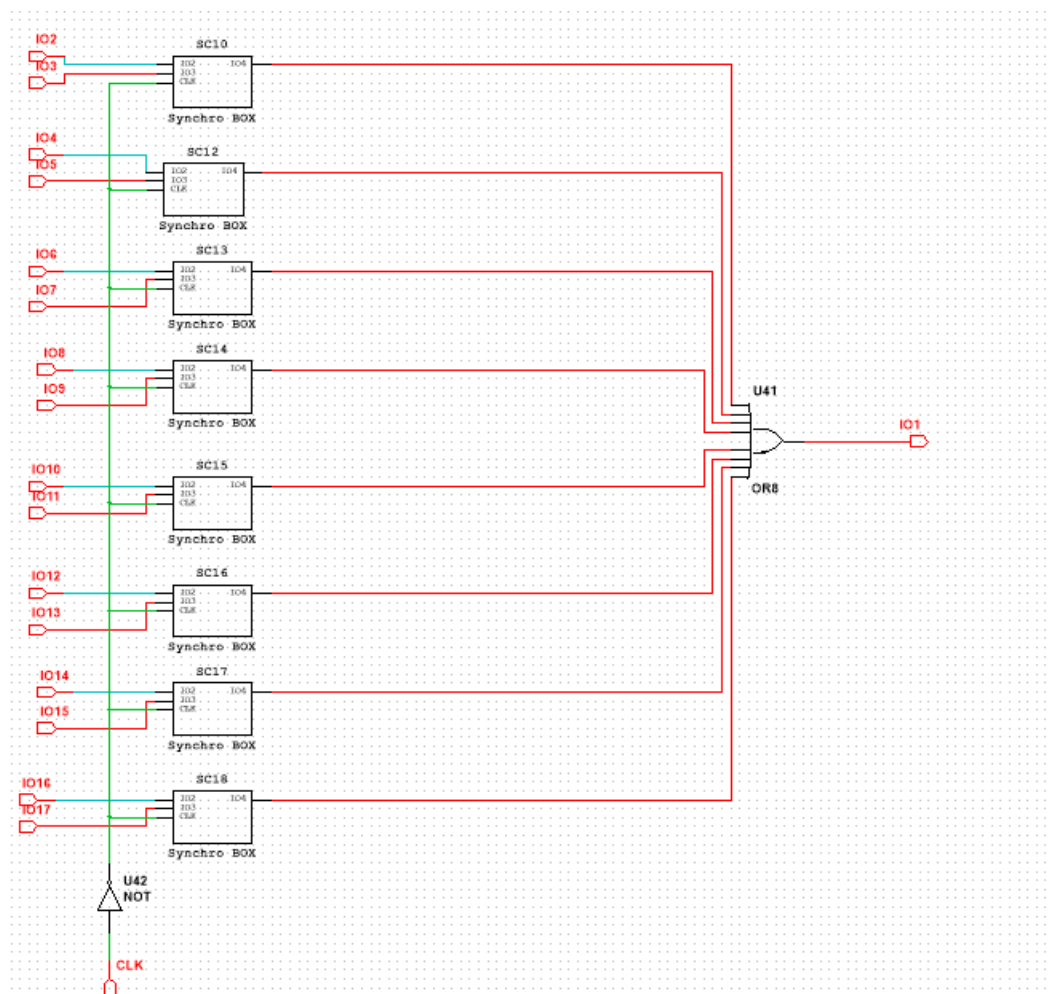
## 2.6 Generator słów



Rysunek 23. Ustawienia generatora słów.

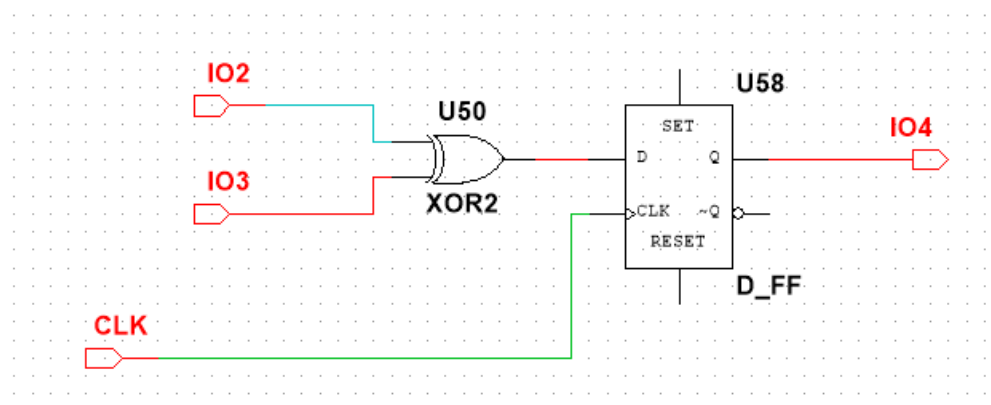


## 2.7 Tester



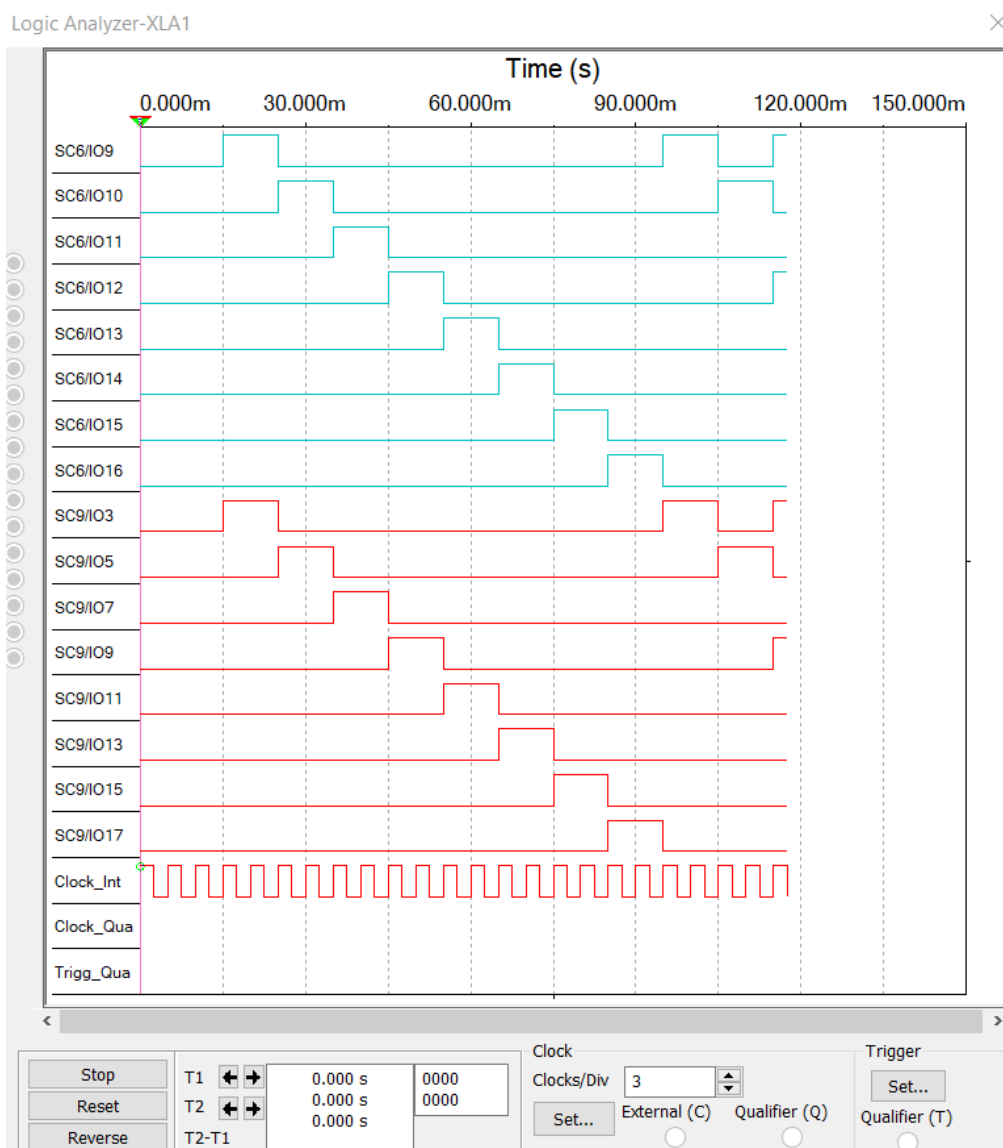
Rysunek 24. Układ podukładów testujących.

Każdy z Synchro BOX'ów (podukład testujący z zaimplementowanym synchronizatorem) w środku zawiera układ:



Rysunek 25. Podukład testujący pojedynczy bit.

## 2.8 Analizator słów



Rysunek 26. Logic analyzer.

Clock Setup

Clock source  
☐ External ☒ Internal

Clock rate  
 200 Hz

Clock qualifier:  
 x

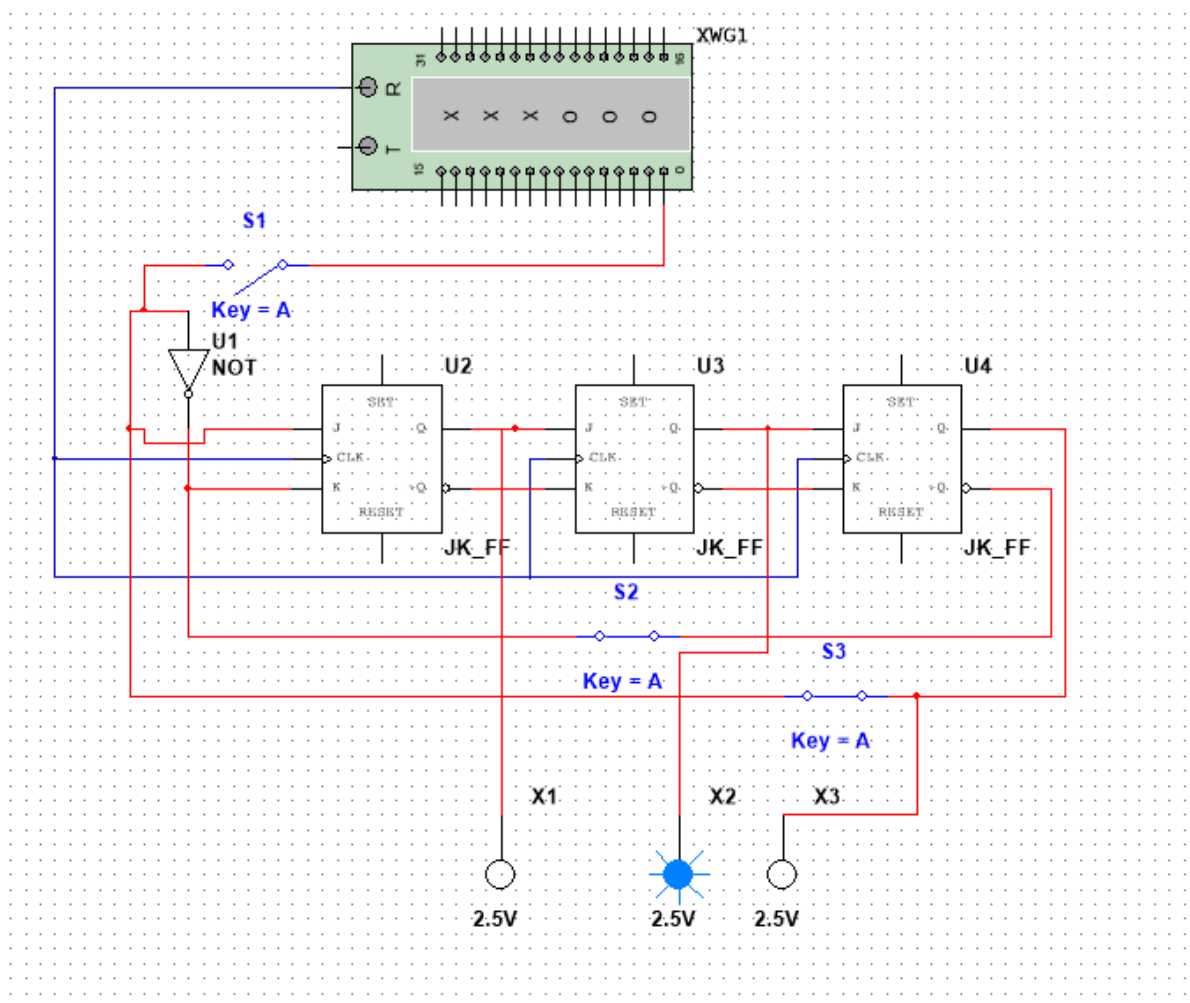
Sampling setting  
 Pre-trigger samples: 100  
 Post-trigger samples: 1000  
 Threshold volt. (V): 2.5 V

OK  
Cancel

Rysunek 27. Konfiguracja logic analyzera.

## 2.9 Wnioski

1. Rejestry służą do przechowywania i odtwarzania informacji, dlatego można je wykorzystać w różnego rodzaju migających lampach i banerach.
2. Jest to prosta w użyciu technologia, która stwarza możliwość szybkiego i łatwego rozszerzenia na większą liczbę zapamiętanych bitów.
3. W programie Multisim udało nam się stworzyć dodatkowo model rejestru pierścieniowego za pomocą przerzutników JK:



**Rysunek 28.** Wstępny rejestr pierścieniowy na mniejszą skalę w programie Multisim z wykorzystaniem przerzutników JK.

Zadanie w związku z tym można było rozwiązać za pomocą tego typu przerzutników, ale implementacja ich jest znacznie bardziej skomplikowana z powodu dwóch wejść.

## 2.10 Przykładowe zastosowania

1. Rejestr mógłby znaleźć zastosowanie w maszynie do lodów, gdzie dane krążące w rejestrze determinują, kiedy oraz który smak lodów nakładać. Tryb zapisywania pełni formę przyjmowania zamówień klientów.



2. Rejestr mógłby zostać użyty w migaczach policyjnych. Od pozycji bitów podczas krążenia danych zależy wtedy, które lampy mają świecić w danym momencie cyklu.

