

Technika Cyfrowa.
Ćwiczenie 2.

Maciej Pieta

Piotr Koproń
Rafał Piwowar

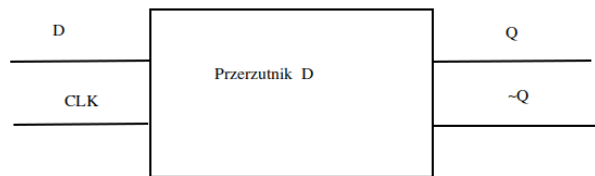
Jakub Woś

Marzec 2023

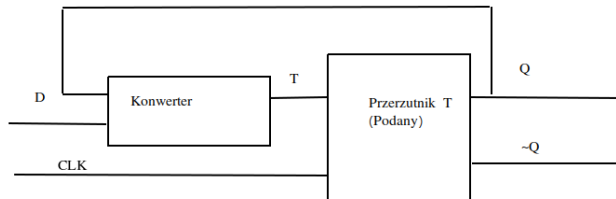
1 Zadanie 2a

Treść zadania Na podstawie dostępnych tabel prawdy, zaprojektować i praktycznie zrealizować synchroniczny przerzutnik D w oparciu o dostępny synchroniczny przerzutnik T, po czym proszę jednoznacznie przetestować poprawność jego działania w programie Multisim.

1.1 Ogólna idea rozwiązania



Jako że realizacja ma opierać się o synchroniczny przerzutnik T, to schemat przyjmuje postać:



W celu wyznaczenia bramek logicznych zastosujemy następujący algorytm:

1. Wyznamy wzory przejścia dla przerzutników D oraz T.
2. Nadamy równoważność wzorom przejścia.
3. Otrzymamy zależność między sygnałami D, T, oraz Q.
4. Przekształcimy otrzymaną zależność do funkcji T od D i Q.

Wzory przejścia Dla przerzutnika T:

T	Q	Q_T^+	
0	0	0	
0	1	1	\implies Z definicji xor otrzymujemy $Q_T^+ = T \text{ xor } Q$. (1)
1	0	1	
1	1	0	

Dla przerzutnika D:

D	Q	Q_D^+	
0	0	0	
0	1	0	\implies Bezpośrednio otrzymujemy $Q_D^+ = D$. (2)
1	0	1	
1	1	1	

Z (1) i (2), podstawiając $Q_T^+ = Q_D^+$ otrzymujemy $D = T \text{ xor } Q$ (3).

Przekształcenie do funkcji Chcemy utworzyć funkcję T od D i Q, tak aby (3) zawsze było spełnione. Tworzymy tabelę, gdzie po lewej stronie będziemy mieć wartości niezależne, w środku - wyrażenie wymuszające, po prawej - wyrażenia zależne.

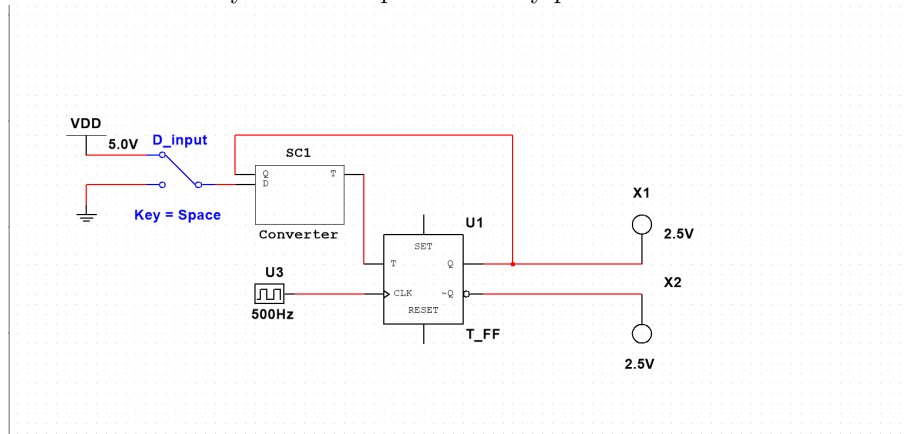
D	Q	$D = T \text{ xor } Q$	$T \text{ xor } Q$	T
0	0	1	0	0
0	1	1	0	1
1	0	1	1	1
1	1	1	1	0

Usuując kolumny ' $D = T \text{ xor } Q$ ' i ' $T \text{ xor } Q$ ' z powyższej tabeli, otrzymujemy:

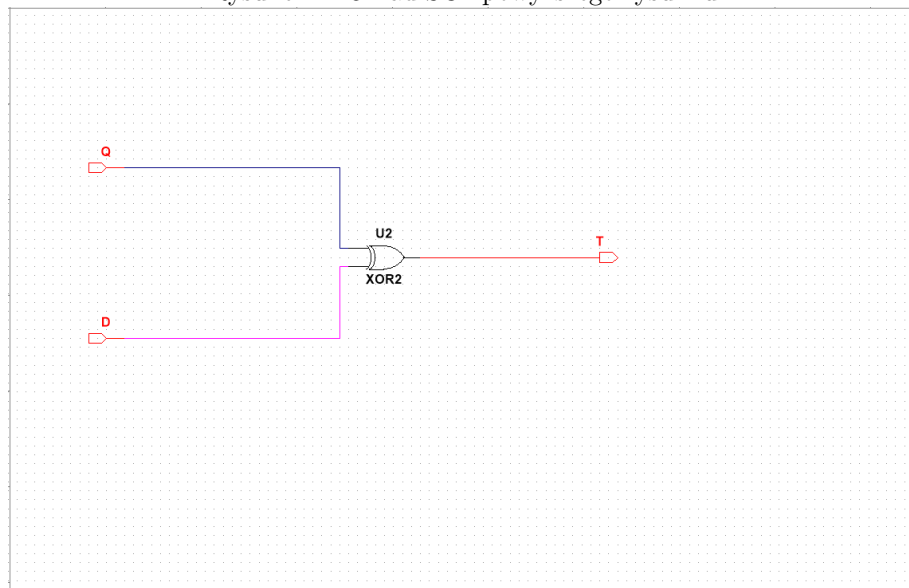
D	Q	T	
0	0	0	
0	1	1	\implies Z definicji xor otrzymujemy $T = D \text{ xor } Q$.
1	0	1	
1	1	0	

1.2 Implementacja w programie Multisim

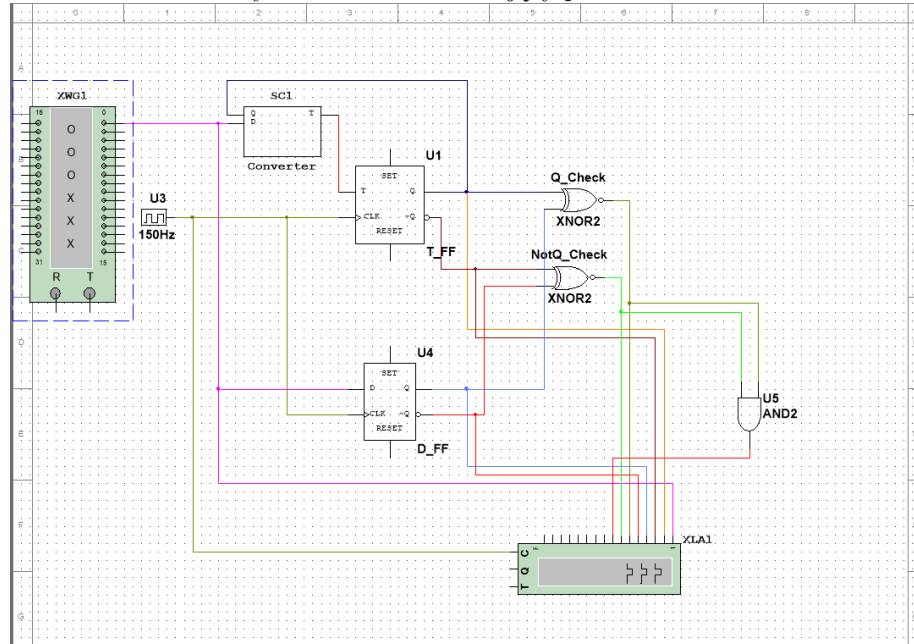
Rysunek 1: Implementowany przerzutnik D.



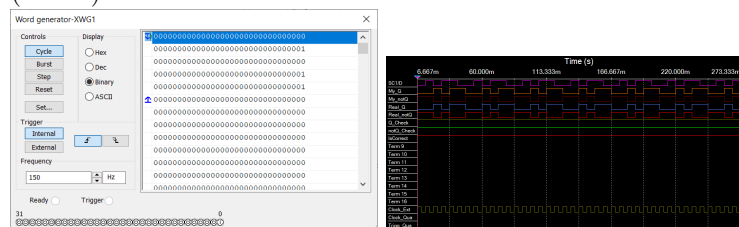
Rysunek 2: Układ SC1 powyższego rysunku.



Rysunek 3: Układ testujący przerzutnik.

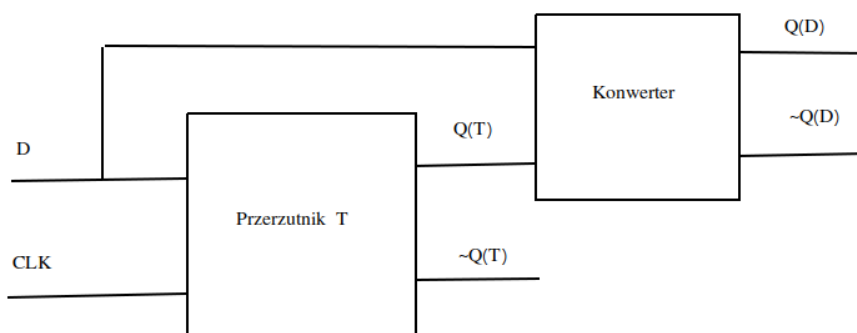


Rysunek 4: Ustawienia generatora słów (XWG1) i rezultat analizatora logicznego (XLA1).



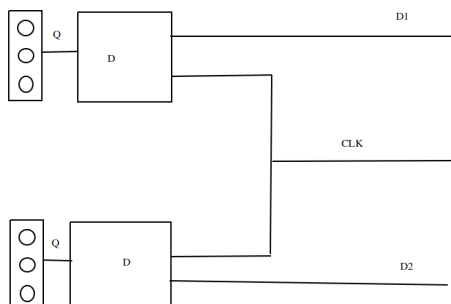
1.3 Wnioski

Alternatywne koncepcje Rozważaliśmy podejście alternatywne, w którym zamiast przekształcać sygnał wejściowy, przekształcalibyśmy sygnał wyjściowy.



Szybko doszliśmy jednak do wniosku że takie podejście wymagałoby zastosowania drugiego przerzutnika, co miałooby się z celem zadania, więc skupiliśmy się na opisanym wyżej podejściu.

Zastosowania Przerzutniki typu D mogą być stosowane na przykład w rejestrach, co zademonstrujemy w dalszej części sprawozdania - w którym zbudowaliśmy czterobitowy rejestr PISO w oparciu właśnie o przerzutniki D. Alternatywnie, przerzutnik typu D może być zastosowany w sygnalizacji świetlnej, do synchronizacji zmian świateł między pasami, zachowując możliwość indywidualnej zmiany świateł.



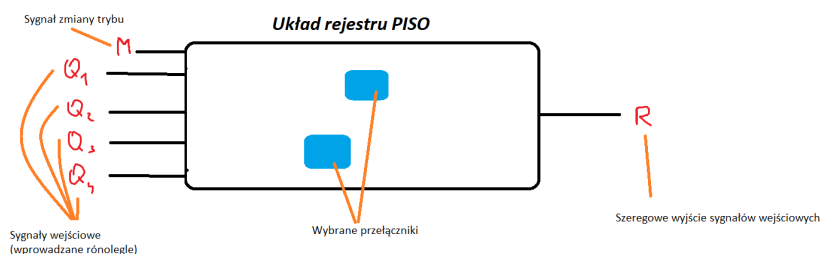
Rysunek 5: Fragment przedstawiający kontrolę żółtego światła.

Jeżeli chcemy zmienić światło tylko na dolnym sygnalizatorze, to wystarczy zmienić sygnał D2. Jeżeli chcemy zmienić wszystkie światła na raz, to synchronizacja gwarantuje nam że nawet jeżeli sygnały D1 i D2 ustawią się w nieco innym momencie, to faktyczne światła zmieniają się równocześnie.

2 Zadanie 2b

Treść zadania Korzystając z wybranych przerzutników, proszę zbudować czterobitowy rejestr PISO. Tak jak w przypadku pozostałych zadań, proszę skutecznie przetestować działanie układu. Następnie proszę zbudować praktyczny układ, który za pomocą przełączników binarnych pozwoli ustawić żądaną czterobitową wartość, a następnie przy pomocy piątego przełącznika uruchomi szeregową transmisję odczytywanej wartości.

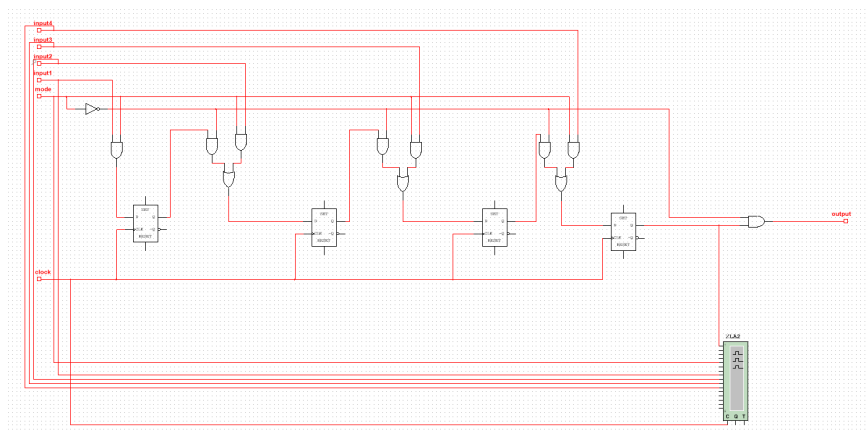
Wykorzystane zostały synchroniczne przerzutniki D, jak zostało wspomniane wyżej.



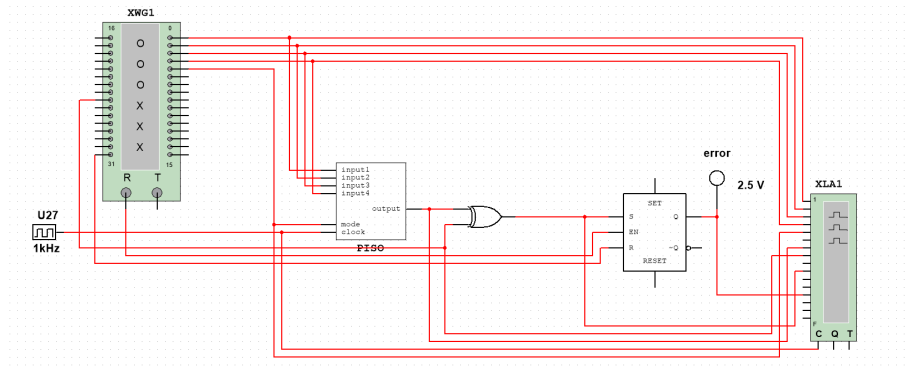
Implementacja W celu zapamiętania czterech bitów - korzystamy z czterech przerzutników D.

W trybie zapisu, zapisują one wartości odpowiednio input 1-4.

W trybie odczytu, przez cztery cykle zegara na wartość output przekazywana jest wartość ostatniego przerzutnika, wszystkie poprzednie wartości są przesuwane do kolejnych przerzutników. (1 do 2, 2 do 3, 3 do 4) Do pierwszego przerzutnika wpisujemy 0. W ten sposób po zakończeniu transmisji otrzymujemy stały sygnał 0, taki sam jak w trybie zapisu.



Układ testujący licznik. Lampka błędu wskazuje od razu czy wystąpił jakikolwiek błąd, analizator logiczny podaje szczegółowy obraz.



Generator słów XWG1 został uruchomiony z automatyczną serią testów wygenerowaną przez poniższy program języka Python:

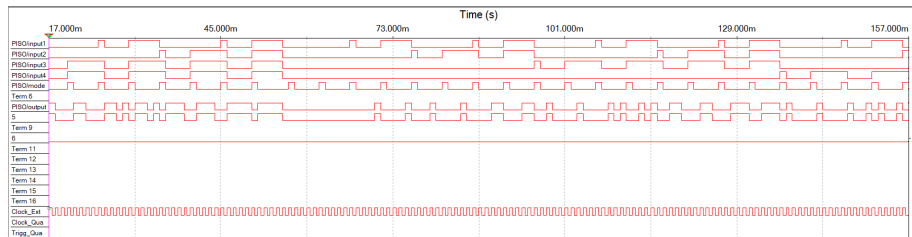
```
output = []
def toBinary(val):
    res = []
    while val != 0:
        res.insert(0, val % 2)
        val = val // 2
    while len(res) < 4:
        res.insert(0, 0)
    return res
def hex4(val):
    res = hex(val)[2:]
    while len(res) < 4:
        res = "0" + res
    return res
output.append("Data:\n")
output.append("80000000\n")
for i in range(16):
    output.append('0000001' + hex(i)[2:] + '\n')
    binary = toBinary(i)
    for k in range(4):
        output.append('0' + str(binary[k]) + '000000' + '\n')
    output.append('0000001' + hex(i)[2:] + '\n')
    for k in range(4):
        output.append('0' + str(binary[k]) + '00000' + hex(i)[2:] + '\n')
output.append("Initial:" + '\n')
output.append("0000" + '\n')
```



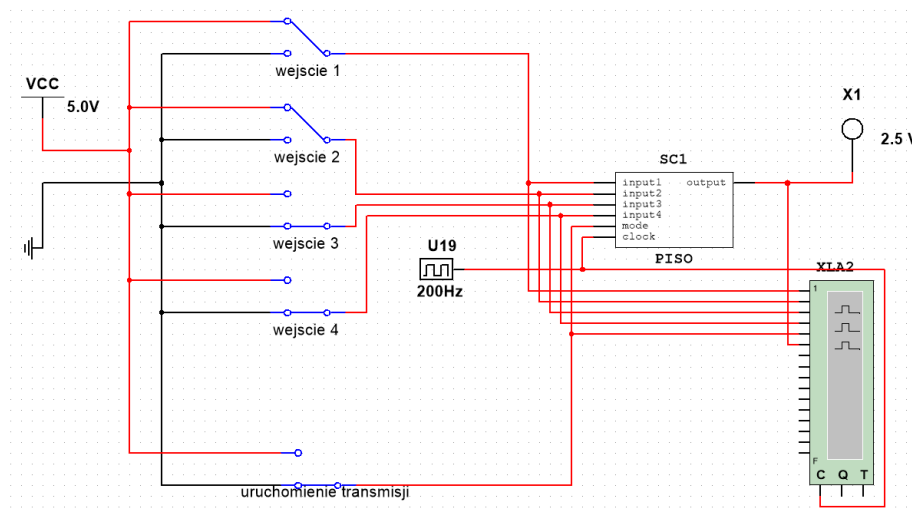
```
output.append(" Final:" + '\n')
output.append(hex4(161) + '\n')
```

```
with open('output.dp', 'w') as outputFile:
    outputFile.writelines(output)
```

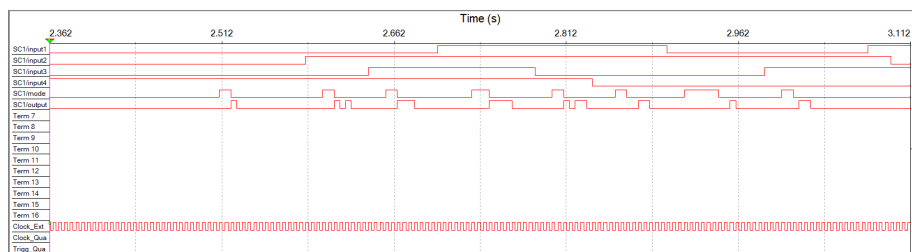
Rezultat z analizatora logicznego (XLA1).



Układ na przełącznikach. Lampka X1 przedstawia szeregowe wyjście.



Rezultat ręcznych testów zapisany za pomocą dołączonego analizatora logicznego (XLA2).



Słowem komentarza twórczego Podczas tworzenia rejestru, pierwotnie użyliśmy przerzutnika asynchronicznego zamiast synchronicznego. Układ w takim stanie oczywiście nie działał. Jednym z możliwych "rozwiązań" okazało się zamiana każdego z przerzutników na dwa identyczne, z pośrednim połączeniem $Q_1 = D2$. Zdziwiająco, rozwiązanie takie działało... czasami. Czasami jednak produkowało wyniki niewytłumaczalne przez autorów. Ostatecznie zauważyliśmy jednak nasz błąd i wymieniliśmy pary przerzutników asynchronicznych na pojedyncze - synchroniczne.