

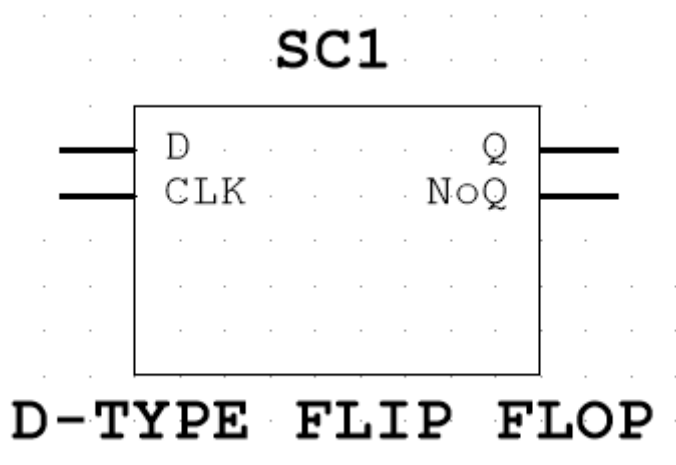
Sprawozdanie 2

Krzysztof Gryboś, Piotr Kądziela, Adrian Madej, Filip Piskorski

1. Zadanie 2a

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. Rozwiązanie teoretyczne



Rysunek 1. Koncepcja układu

1.2. Tabele prawdy

Tabela prawdy przerzutnika T

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Tabela 1. Tabela prawdy przerzutnika T

Tabela prawdy przerzutnika D

D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

Tabela 2. Tabela prawdy przerzutnika D

Tabela prawdy przerzutnika D przy wykorzystaniu przerzutnika T

D	Q_n	Q_{n+1}	T
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Tabela 3. Tabela prawdy przerzutnika D przy wykorzystaniu przerzutnika T

1.3 Tabela Karnaugh

$D \backslash Q_n$	0	1
0	0	1
1	1	0

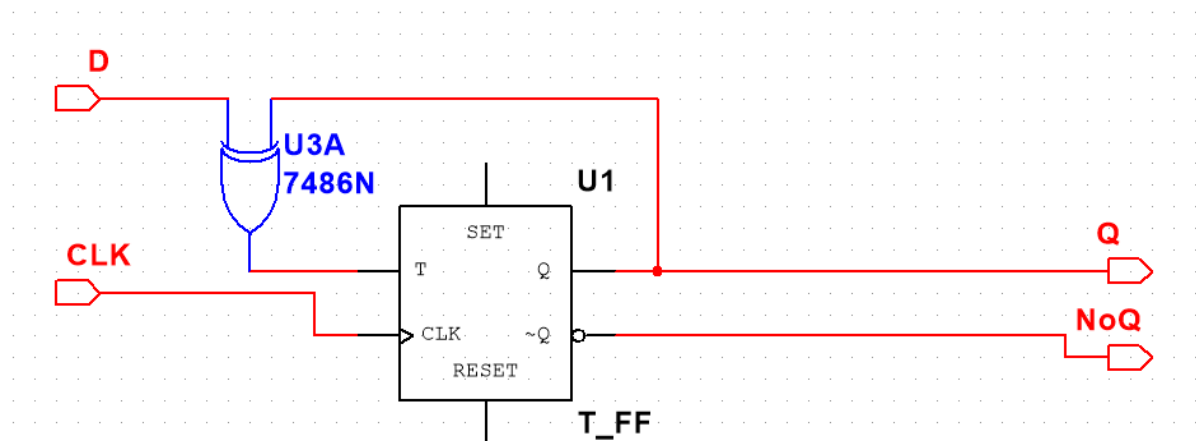
Tabela 4. Tabela prawdy przerzutnika D przy wykorzystaniu przerzutnika T

Otrzymujemy:

$$T = \underline{D}Q_n + D\underline{Q_n} = D \text{ xor}(Q_n)$$

1.4 Implementacja w programie Multisim

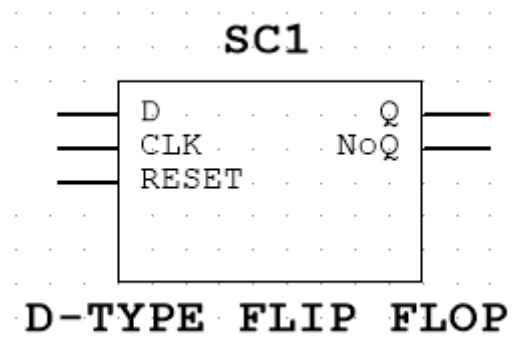
Na podstawie powyższej tabeli konstruujemy układ



Rysunek 2. Przerzutnik D

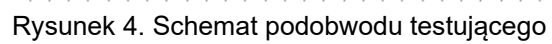
1.5 Testowanie

Do wykonania testu dodaliśmy dodatkowe wyjście do podobwodu

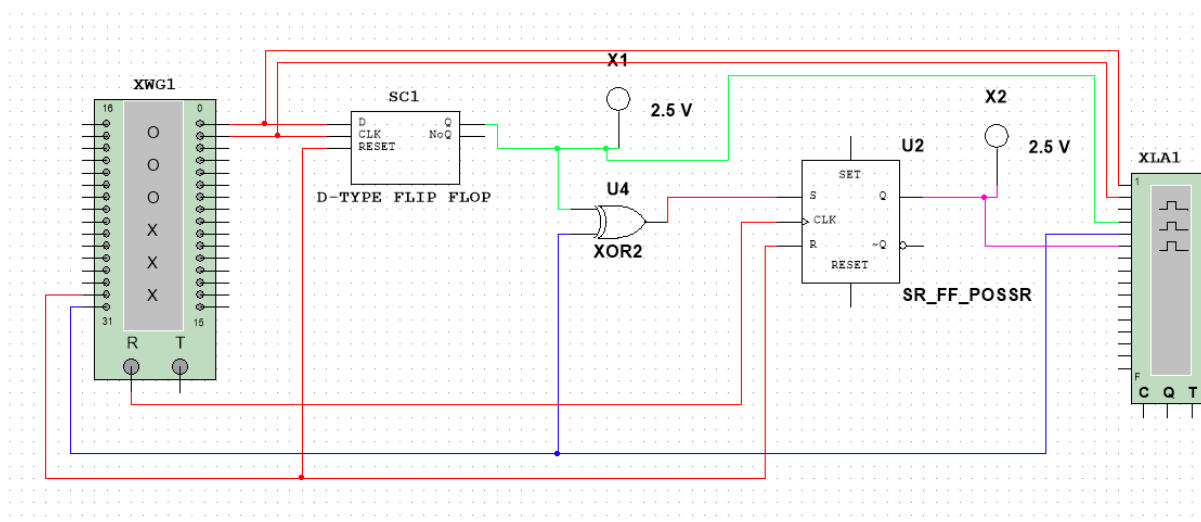


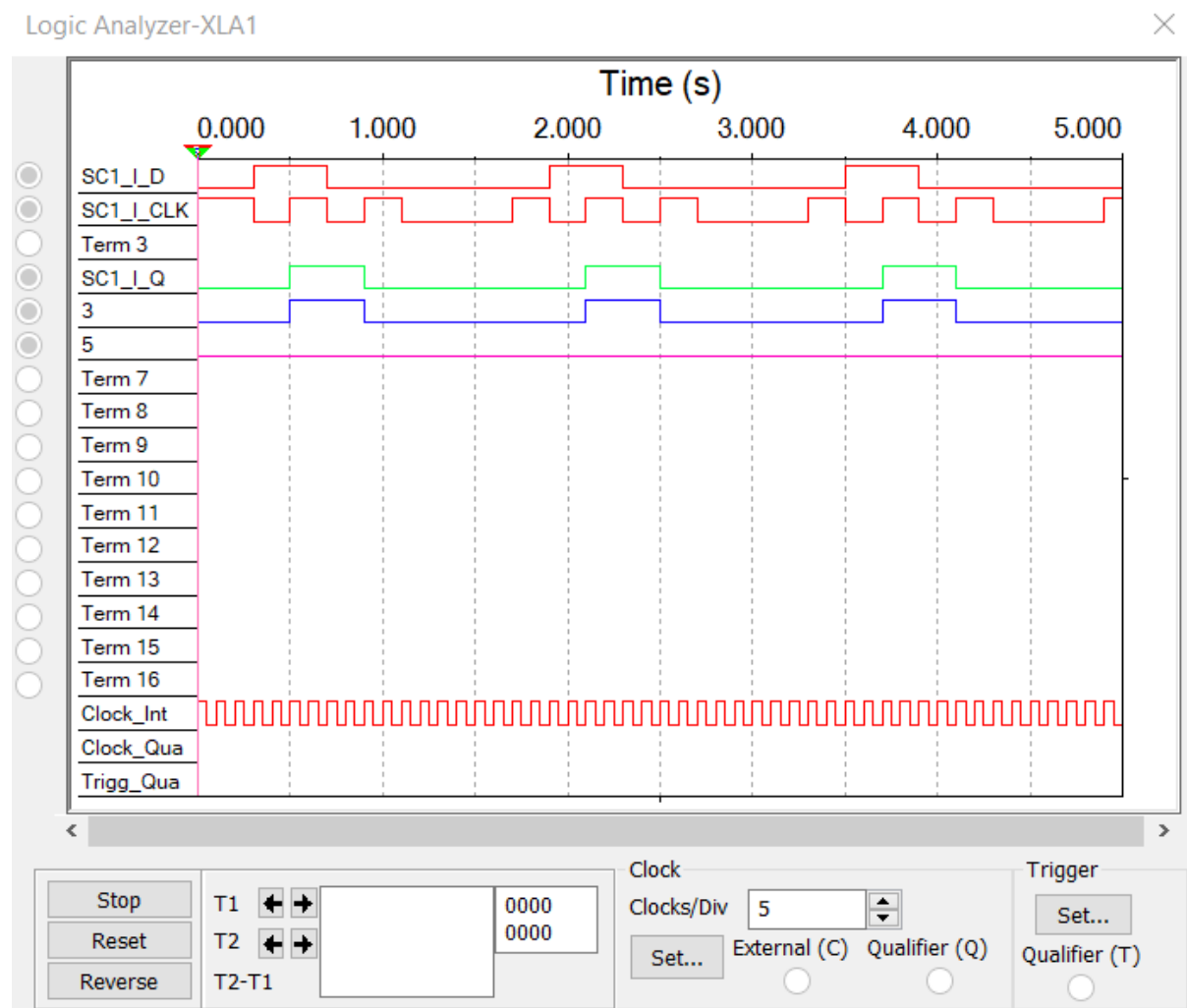
Rysunek 3. Koncepcja podobwodu testującego

Po modyfikacji układ wygląda następująco



Rysunek 5. Schemat testujący



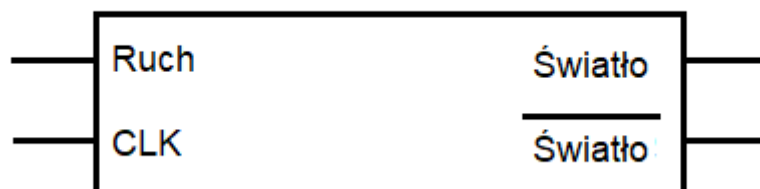


Rysunek 7. Analizator logiczny

Kolorem zielonym, niebieskim i fioletowym oznaczyliśmy kolejno: wynik, spodziewany wynik, różnicę symetryczną. Jak widać różnica zawsze jest równa 0, dioda się nie zapala, co dowodzi skuteczności przedstawionego układu.

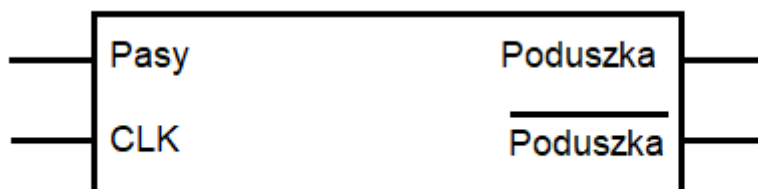
1.6 Wnioski

1. Niektóre przerzutniki można w łatwy sposób realizować za pomocą innych przerzutników, np. przerzutnik D za pomocą T lub JK
2. Przerzutniki typu D są podstawowym elementem liczników binarnych służących do zliczania impulsów.
3. Przerzutniki typu D znajdują zastosowanie w rejestrach przesuwnych.
4. Przerzutnik typu D może służyć jako sygnalizator stanu wejścia. Przykładowo światło które zapala się w przypadku wykrycia ruchu

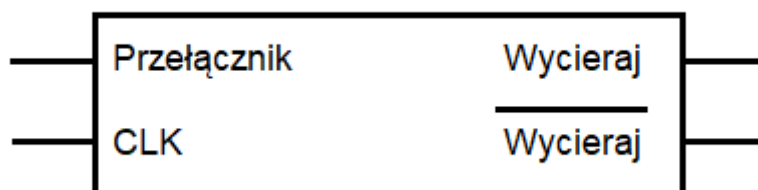


Rysunek 8. Koncepcja obwodu służąca do zapalania światła

lub uzbrojenie poduszki powietrznej w samochodzie gdy zapięto pasy pasażera

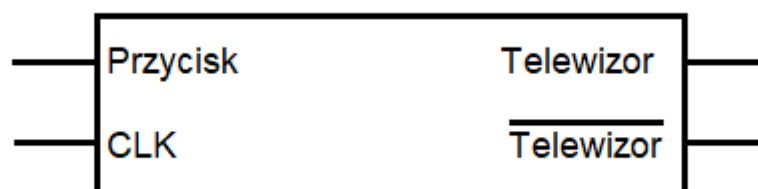


Rysunek 9. Koncepcja obwodu służąca do uzbrajania poduszki powietrznej



Rysunek 10. Koncepcja obwodu służąca do włączania wycieraczek

5. Przerzutnik typu C może zostać użyty jako pilot do włączania i wyłączania telewizora, lub innych urządzeń.

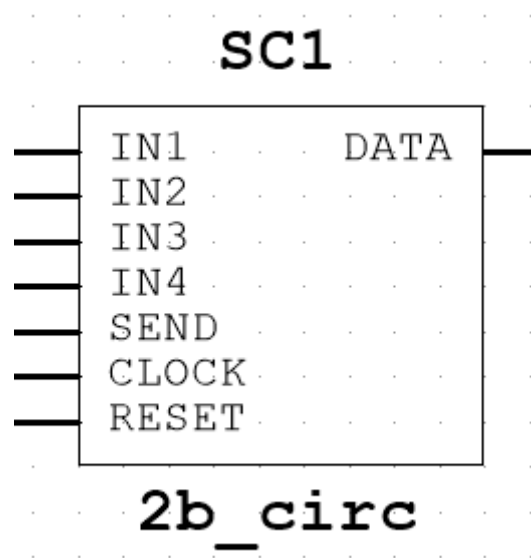


Rysunek 11. Koncepcja obwodu służąca do wyłączania/włączania telewizora

2. Zadanie 2b

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.

2.1 Rozwiązanie teoretyczne



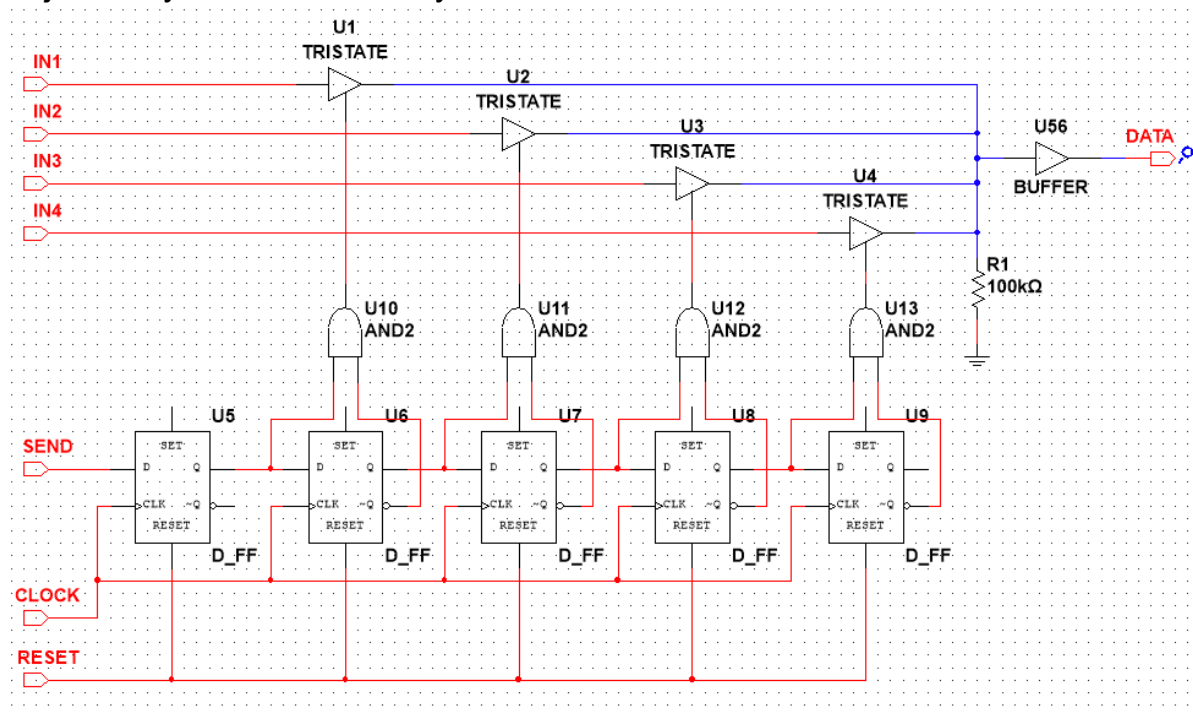
Rysunek 12. Koncepcja obwodu układu

Układ ma 7 wejść.

IN1, IN2, IN3, IN4 to wejścia pozwalające ustawić żądaną wartość, za pomocą send sterujemy trybem pracy rejestratora, clock to zegar, reset resetuje wszystkie przerzutniki.

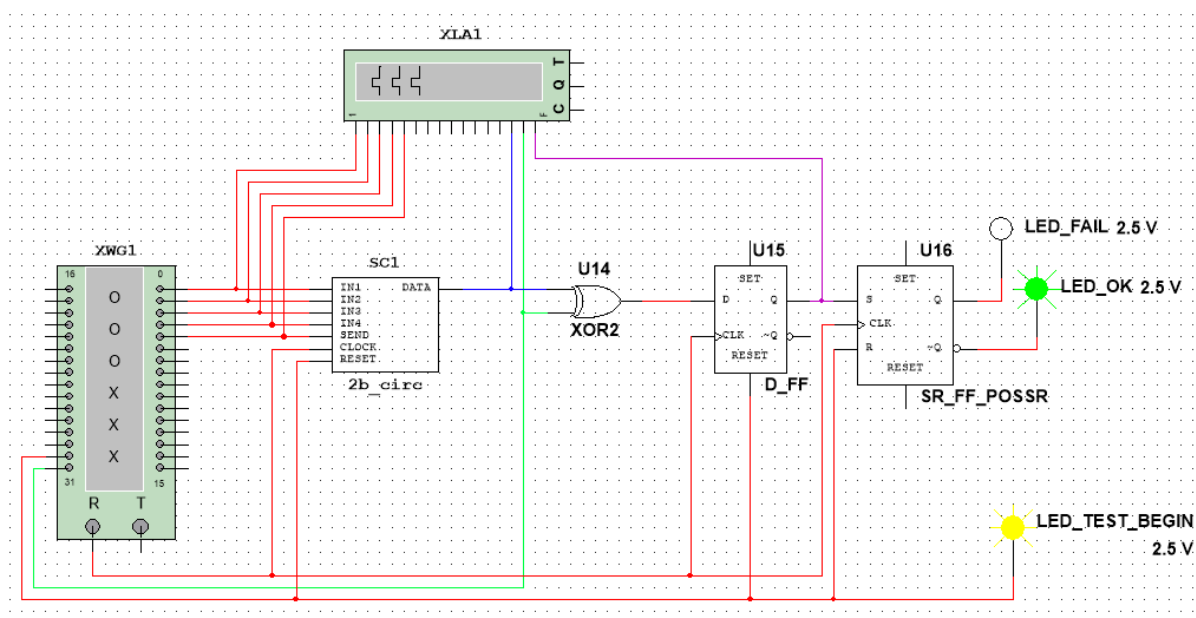
2.2 Implementacja w programie Multisim

Do budowy rejestru wykorzystano 5 przerzutników typu D, 4 bramek AND, 4 tróstanowych buforów i rezystora.



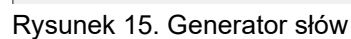
Rysunek 13. Układ PISO

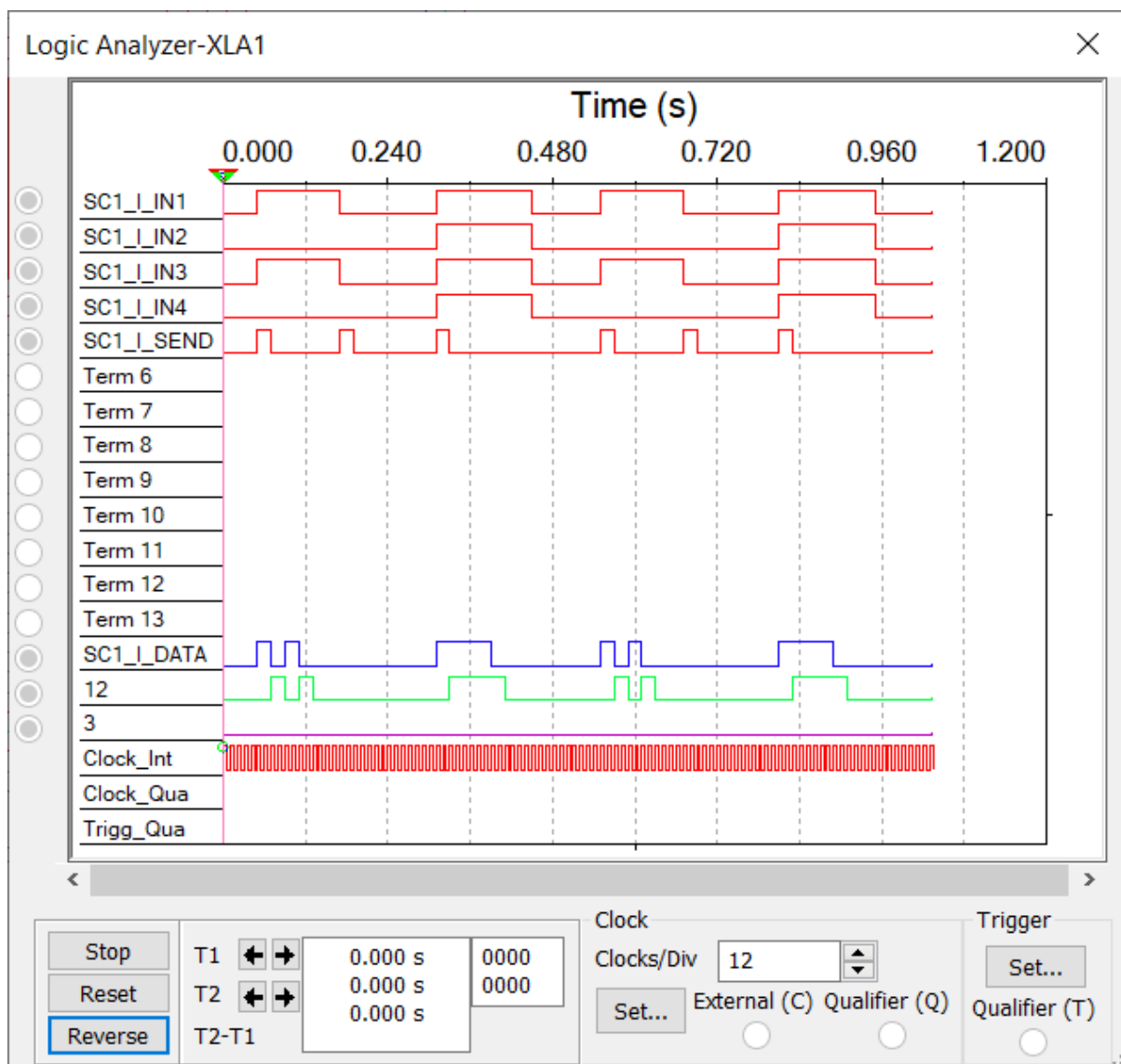
2.3 Układ testujący



Rysunek 14. Układ testujący

Rysunek 15. Generator słów





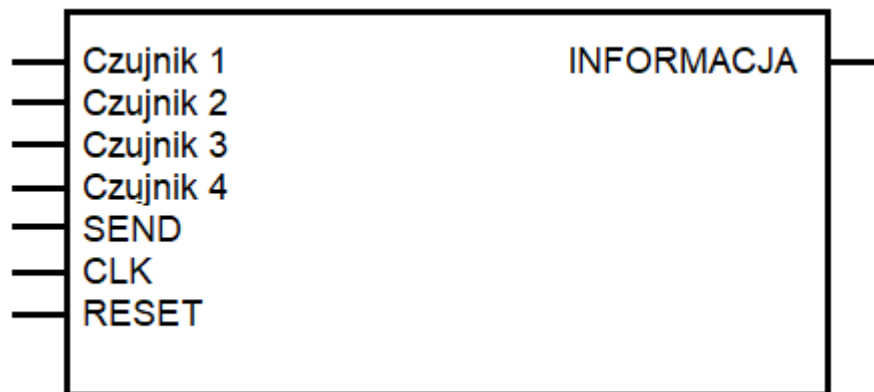
Rysunek 16. Analizator logiczny

Jak widzimy wyniki (wykres niebieski) są zgodne z oczekiwaniami (wykres zielony). Różnica symetryczna (wykres fioletowy) jest zawsze równa 0 co dowodzi poprawności schematu

2.4 Wnioski

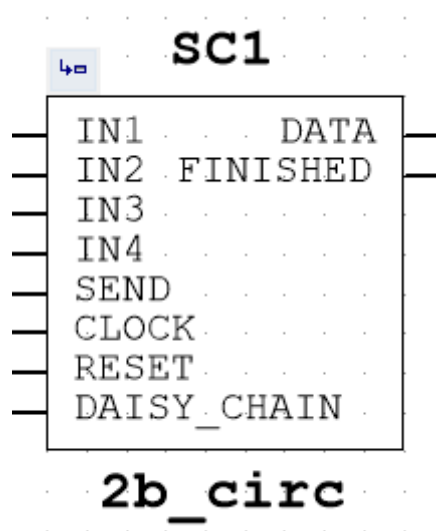
1. Rejestr PISO można zrealizować wykorzystując przerzutniki typu D, bramki AND, trójstanowe buffory i rezystor.
2. Rejestr PISO może być wykorzystany do przesyłania danych z czujników do mikrokontrolera. Przykładowo jeśli mamy dane czujniki generujące dane równolegle w tym czujniki: temperatury, wilgotności, ciśnienia, zmierzchu to mikrokontroler może odczytać dane z rejestru PISO przy użyciu wejścia szeregowego. Powoduje to zmniejszenie liczby wejść mikrokontrolera co pozwala zmniejszyć koszty i zwiększyć efektywność systemu.

3. Rejestr PISO może zostać wykorzystany w systemach alarmowych. Mając 4 czujniki antywłamaniowe na oknach, chcemy przesyłać informację o włamaniu do centrali w innym budynku wykorzystując tylko jeden kabel

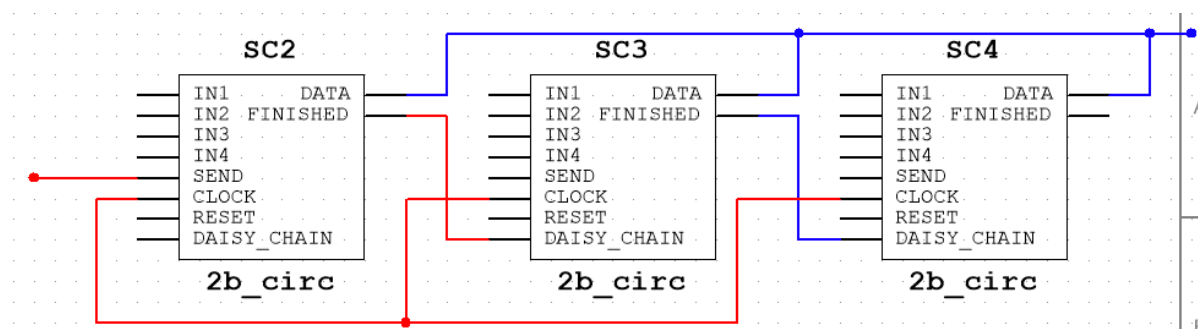


Rysunek 17. Koncepcja obwodu alarmowego

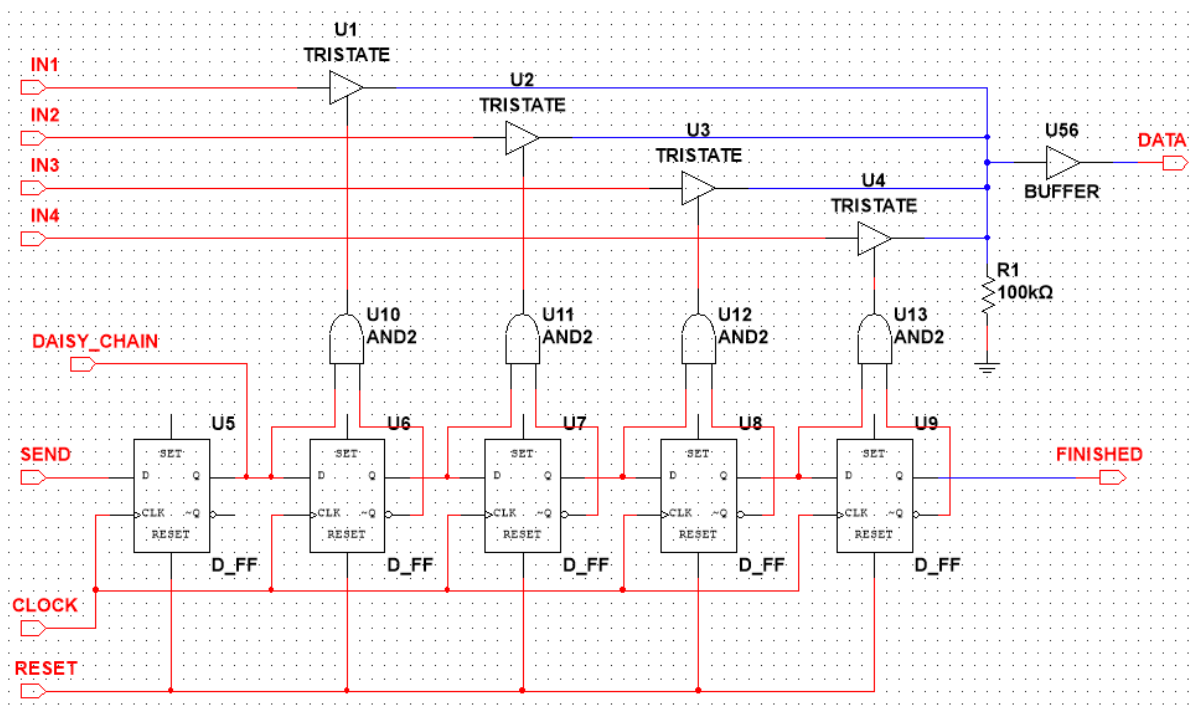
4. Jeśli dodamy dodatkowe wejście DAISY_CHAIN oraz wyjście FINISHED jesteśmy w stanie połączyć kilka układów ze sobą i utworzyć z nich jeden większy rejestr. Przykładowo łącząc ze sobą, 4 układy otrzymamy rejestr 16 bitowy. Łączenie odbywałoby się przez przekazywanie sygnału FINISHED do wejścia następnego DAISY_CHAIN następnego rejestru. Co istotne, rozwiązanie to nie wymaga modyfikowania układu, a jedynie wyprowadzenia dwóch dodatkowych pinów podłączonych do odpowiednich punktów układu.



Rysunek 18. Układ z możliwością łączenia



Rysunek 19. Łączenie ze sobą zmodyfikowanych układów



Rysunek 20. Schemat zmodyfikowanego układu

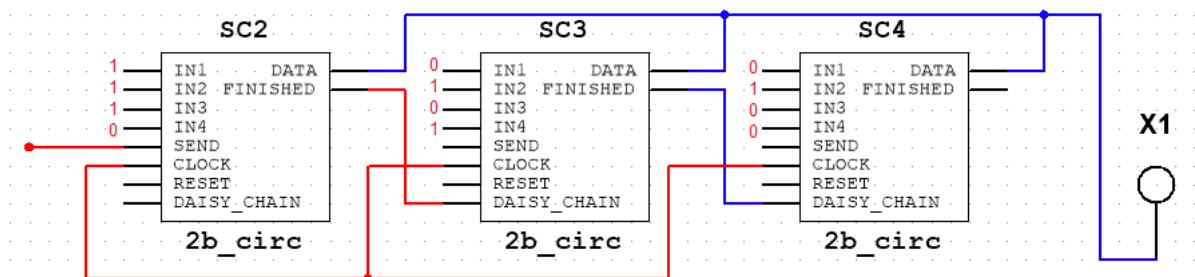
5. Diasy-chainowany rejestr PISO o odpowiedniej długości można wykorzystać do nawaniania kodu Morse'a. Na wejściu takiego układu można podać zera i jedyńki, za pomocą których kodujemy kropki i kreski nadawanej litery.

Przykładowo nadawanie litery "B" wygląda następująco. Najpierw kod Morse'a trzeba zamienić na sygnały dla wejść, które będą kodować długość trwania kropek i kresk. W tym celu należy zamienić każdą kreskę w trzy jedyńki i każdą kropkę w jedną jedyńkę.

Litera "B" jest reprezentowana w kodzie Morse'a przez "— • • •", więc zamiana wygląda w ten sposób.

—	•	•	•
1110	10	10	10

Odstęp między znakami należy kodować jako "000", a między słowami jako "0000000".



Rysunek 21. Rysunek obrazujący nadawanie litery "B"

Ostatecznie na wejściach mają pojawić się następujące wartości:

ln_0	ln_1	ln_2	ln_3	ln_4	ln_5	ln_6	ln_7	ln_8	ln_9
1	1	1	0	1	0	1	0	1	0