Ćwiczenie 2.

Emil Siatka, Marek Swakoń, Patryk Górski, Piotr Pich

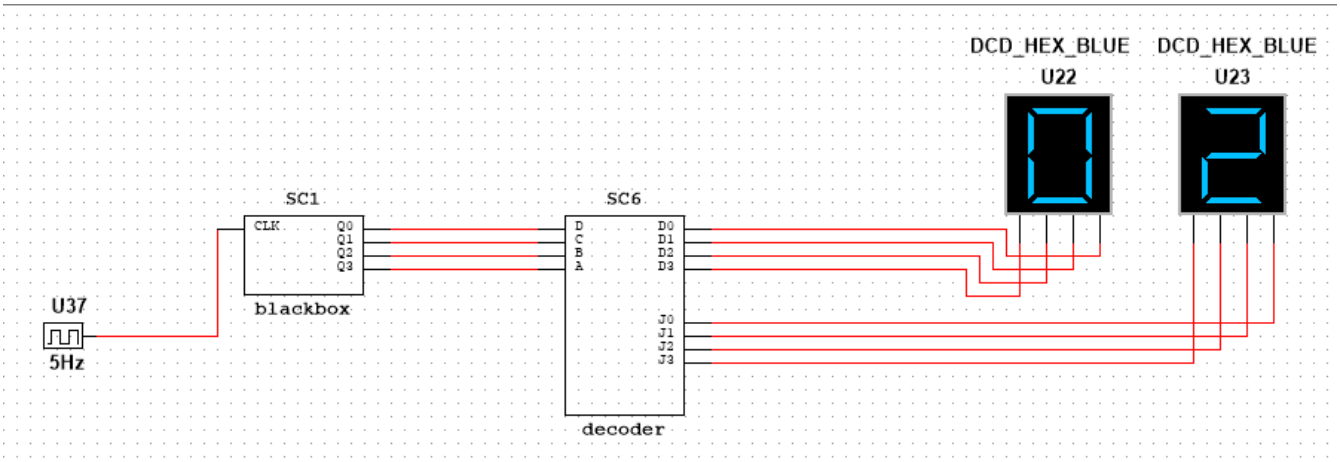
Data oddania: 29.04.2025

Spis treści

1 Wprowadzenie

Celem ćwiczenia jest zaprojektowanie bazującego na jednym typie przerzutników (w naszym przypadku przerzutników typu T) oraz dowolnych bramkach logicznych czterobitowego licznika działającego zgodnie z ciągiem Fibonacciego, gdzie wartość 1 pojawia się tylko raz w cyklu. Aktualna wartość wskazywana przez licznik powinna być widoczna na wyświetlaczach siedmiosegmentowych.

Interfejs naszego układu będzie wyglądał podobnie do przedstawionego na rysunku 1.



Rysunek 1: Interfejs projektowanego układu

Szczególnie interesuje nas podukład `blackbox`, który docelowo implementował będzie interesujący nas mechanizm "zliczania"kolejnych liczb ciągu Fibonacciego.

Przedstawić możemy go jako licznik którego sygnałami wyjściowymi są stany wyjścia czterech przerzutników typu T, które na wejściu T przyjmują wynik czterech czteroargumentowych funkcji zdaniowych T_0, T_1, T_2, T_3, których

argumentami są sygnały wyjściowe w poprzednim cyklu zegara, funkcje T_0, T_1, T_2, T_3 wówczas, przez prostą konwersję pierwszych 8. liczb ciągu Fibonacciego na system dwójkowy, przyjmują poniższe wartości:

Tabela 1: Tabela prawdy dla funkcji T_0, T_1, T_2, T_3

A	B	C	D		T_0	T_1	T_2	T_3
0	0	0	0		1	0	0	0
0	0	0	1		1	0	1	0
0	0	1	0		1	0	1	1
0	0	1	1		1	0	1	1
0	1	0	0		1	0	0	0
0	1	0	1		1	0	1	0
0	1	1	0		1	0	1	1
0	1	1	1		1	0	1	1
1	0	0	0		1	1	0	0
1	0	0	1		1	1	1	0
1	0	1	0		1	0	1	1
1	0	1	1		1	0	1	1
1	1	0	0		0	1	1	0
1	1	0	1		0	1	1	0
1	1	1	0		0	0	1	1
1	1	1	1		0	0	1	1

Podukład `decoder` natomiast, przyjmuje 4 bitową liczbę binarną i zwraca dwie 4-bitowe liczby których wartości są cyframi liczby z zakresu 0-19 w systemie dziesiętnym.

Podukład ten traktować możemy jako układ którego sygnałami wyjściowymi jest 8 czteroarguentowych funkcji zdaniowych $Q_0, Q_1, \ldots, Q_7$ (gdzie funkcje $Q_0, \ldots, Q_3$ odpowiadają liczbie dziesiątek, a $Q_4, \ldots, Q_7$ liczbie jedności) przyjmujących poniższe wartości:

Tabela 2: Tabela prawdy dla funkcji $Q_0, Q_1, \ldots, Q_7$

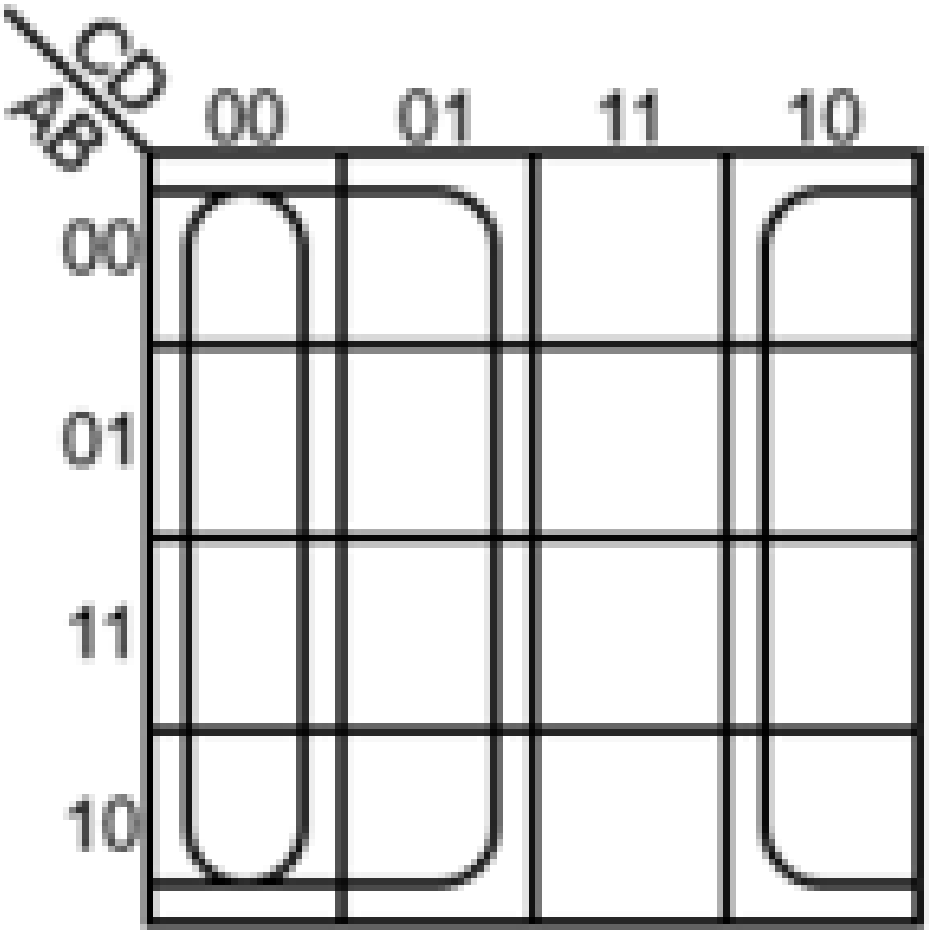
A	B	C	D		Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7
0	0	0	0		0	0	0	0	0	0	0	0
0	0	0	1		0	0	0	0	1	0	0	0
0	0	1	0		0	0	0	0	0	1	0	0
0	0	1	1		0	0	0	0	1	1	0	0
0	1	0	0		0	0	0	0	0	0	1	0
0	1	0	1		0	0	0	0	1	0	1	0
0	1	1	0		0	0	0	0	0	1	1	0
0	1	1	1		0	0	0	0	1	1	1	0
1	0	0	0		0	0	0	0	0	0	0	1
1	0	0	1		0	0	0	0	1	0	0	1
1	0	1	0		1	1	1	1	0	0	0	0
1	0	1	1		1	1	1	1	1	0	0	0
1	1	0	0		1	1	1	1	0	1	0	0
1	1	0	1		1	1	1	1	1	1	0	0
1	1	1	0		1	1	1	1	0	0	1	0
1	1	1	1		1	1	1	1	1	0	1	0

Aby zaprojektować licznik, pozostaje zatem zminimalizować metodą tablic Karnaugh odpowiednio funkcje T_0, T_1, T_2, T_3 (dla podukładu `blackbox`) oraz $Q_0, Q_1, \ldots, Q_7$ (dla podukładu `decoder`) a następnie zbudować go wraz ze stanowiskiem testowym w programie `Multisim`.

2 Minimalizacja funkcji zdaniowych

2.1 Podukład `blackbox`

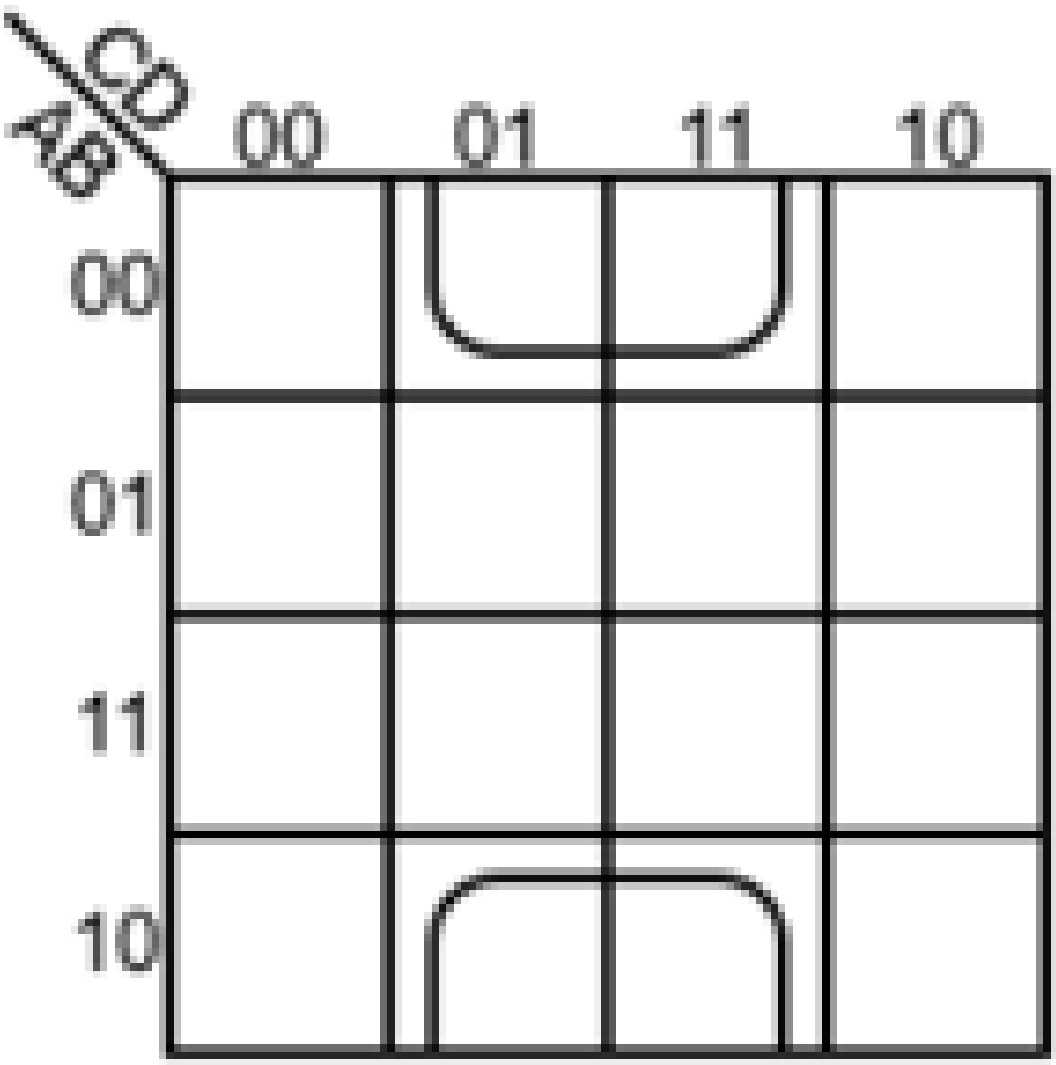
Funkcje T_1, T_2, T_3 oraz T_4 z Tabeli 1. minimalizują poniższe tablice Karnaugh:



Boolean Algebra

$\overline{D} + \overline{C}$

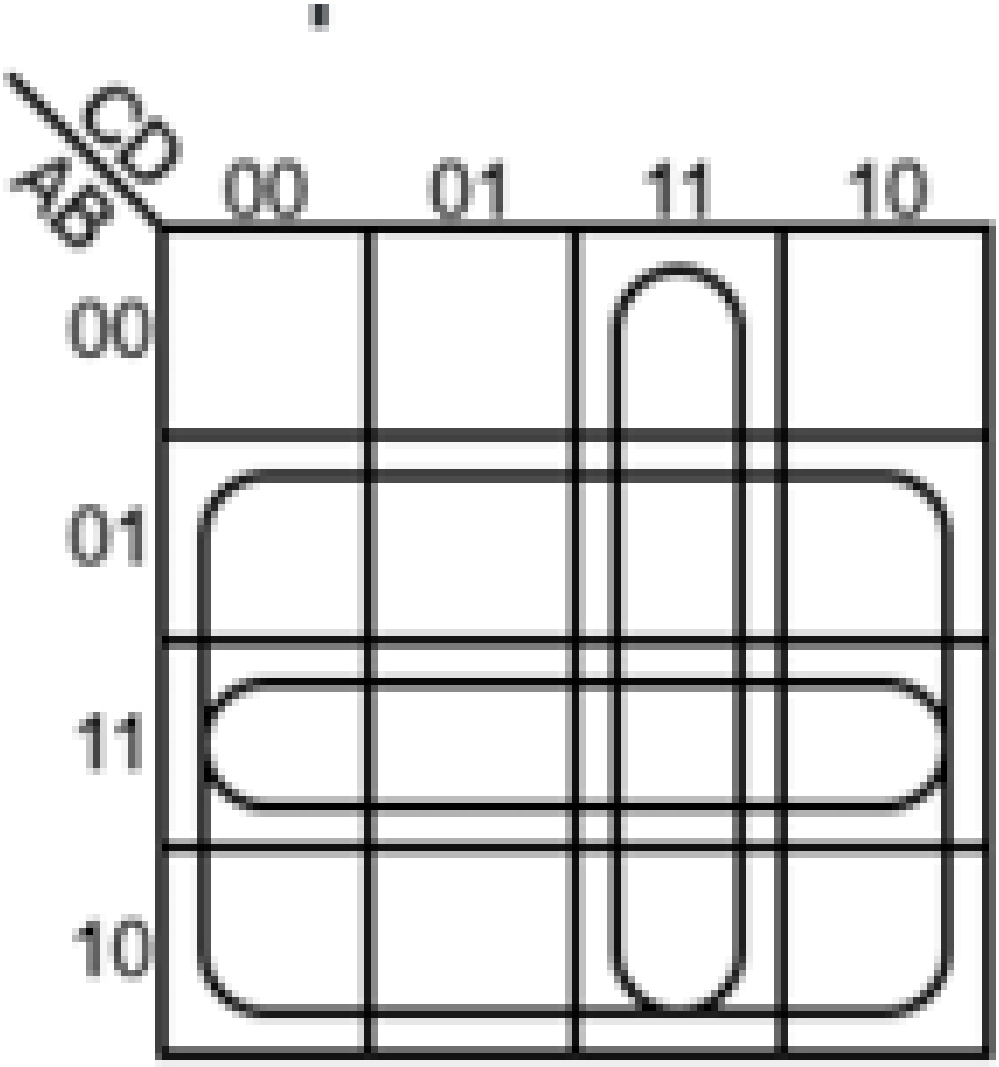
Rysunek 2: Tablica Karnaugh dla funkcji T_0



Boolean Algebra

$\overline{B}D$

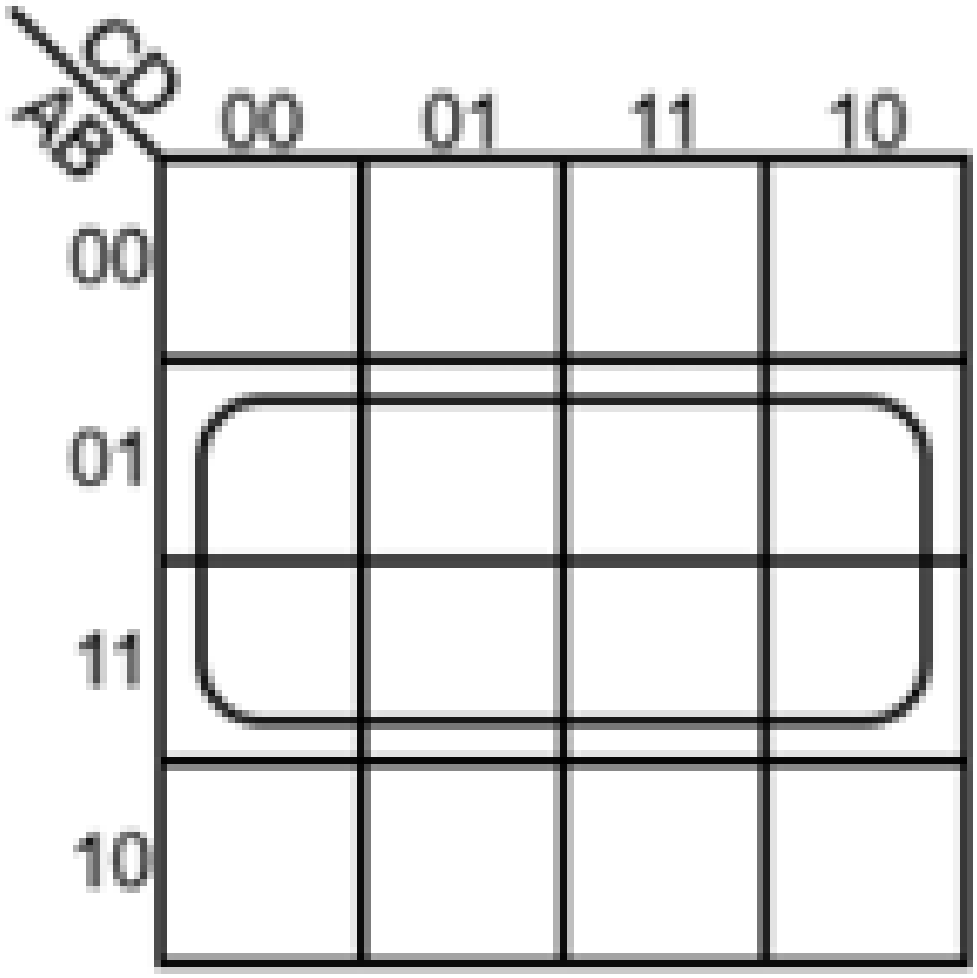
Rysunek 3: Tablica Karnaugh dla funkcji T_1



Boolean Algebra

B+A+CD

Rysunek 4: Tablica Karnaugh dla funkcji T_2



Boolean Algebra

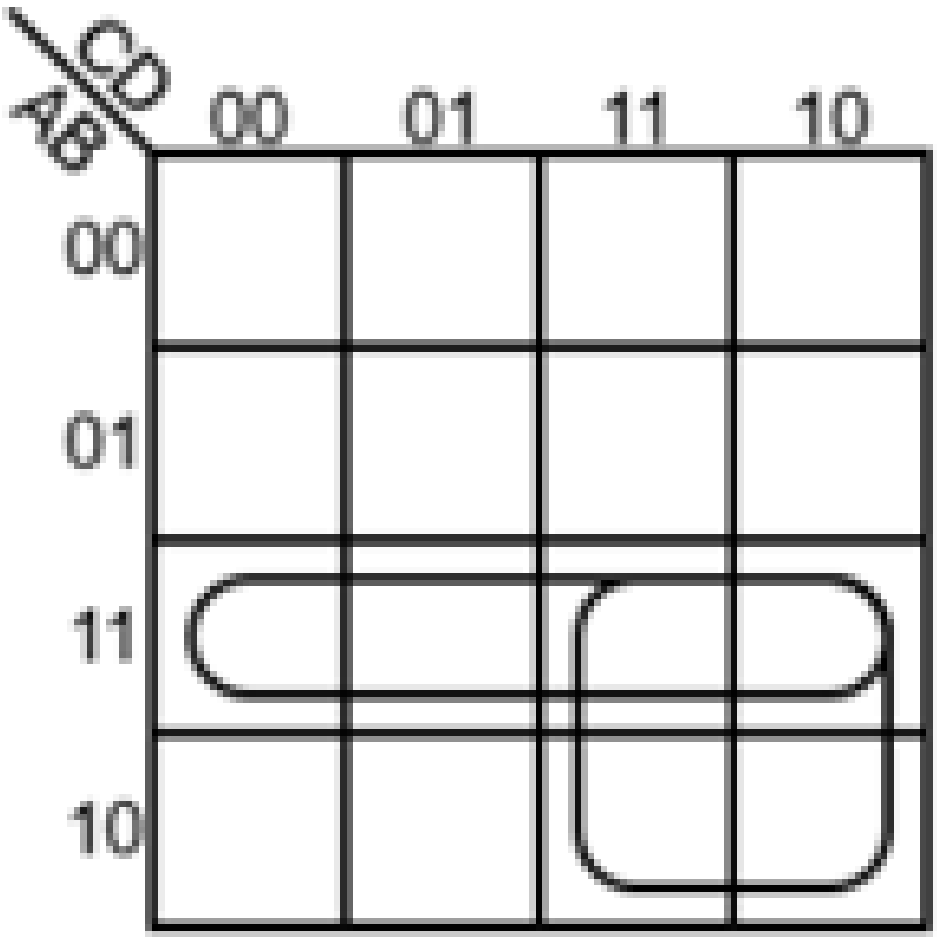
B

Rysunek 5: Tablica Karnaugh dla funkcji T_3

2.2 Podukład `decoder`

Z Tabeli 2. jak również z faktu, że interesuje na wyłącznie konwersja na liczby dziesiętne mniejsze od 20, obserwujemy trywialne uproszczenie:

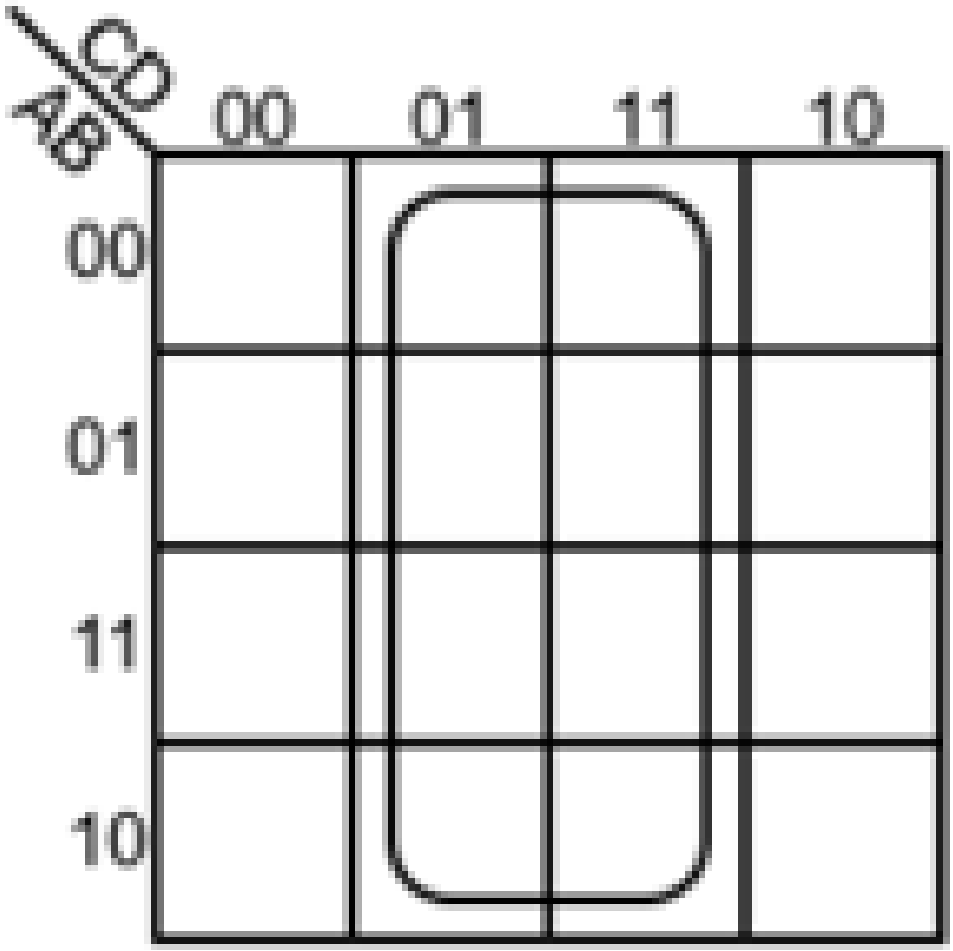
$$Q_0 = Q_1 = Q_2 = Q_3 \tag{1}$$



Boolean Algebra

$$AB + AC$$

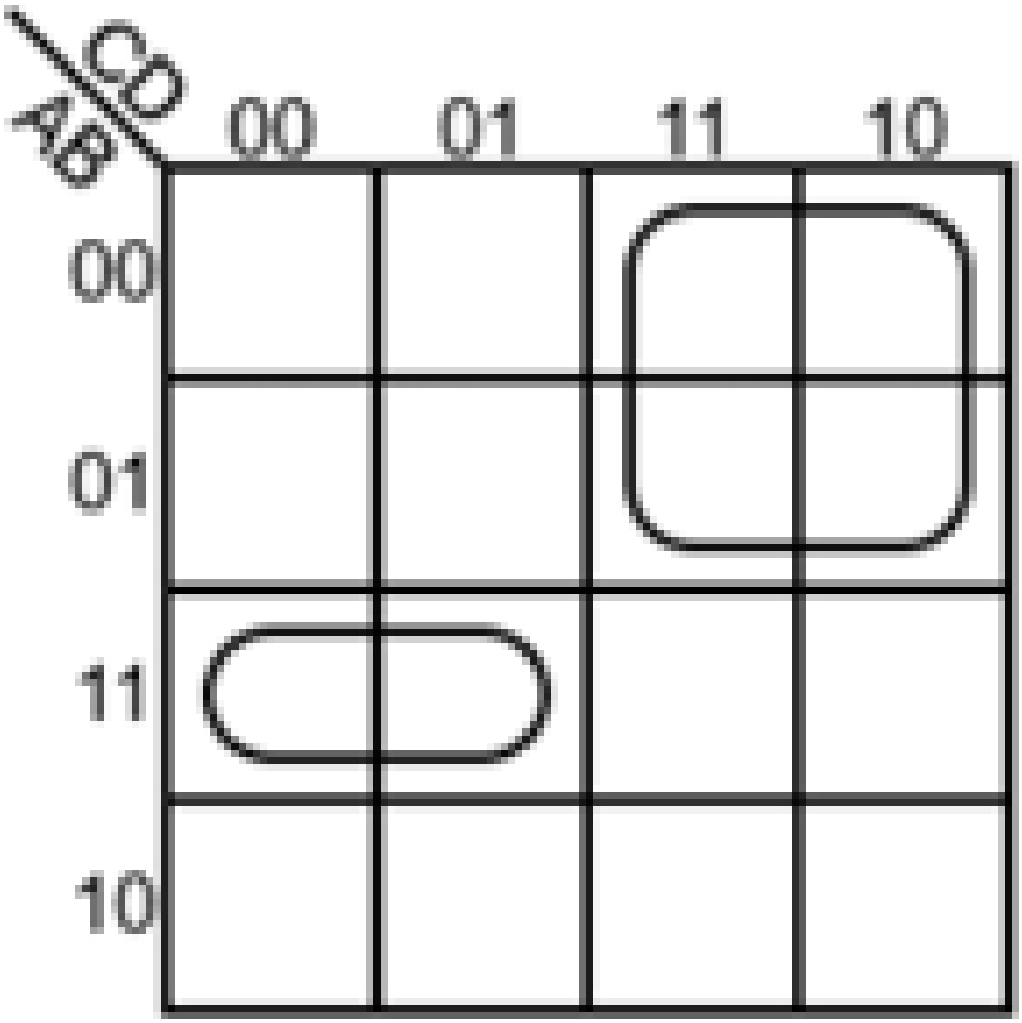
Rysunek 6: Tablica Karnaugh dla funkcji $Q_0, \ldots, Q_3$



Boolean Algebra

D

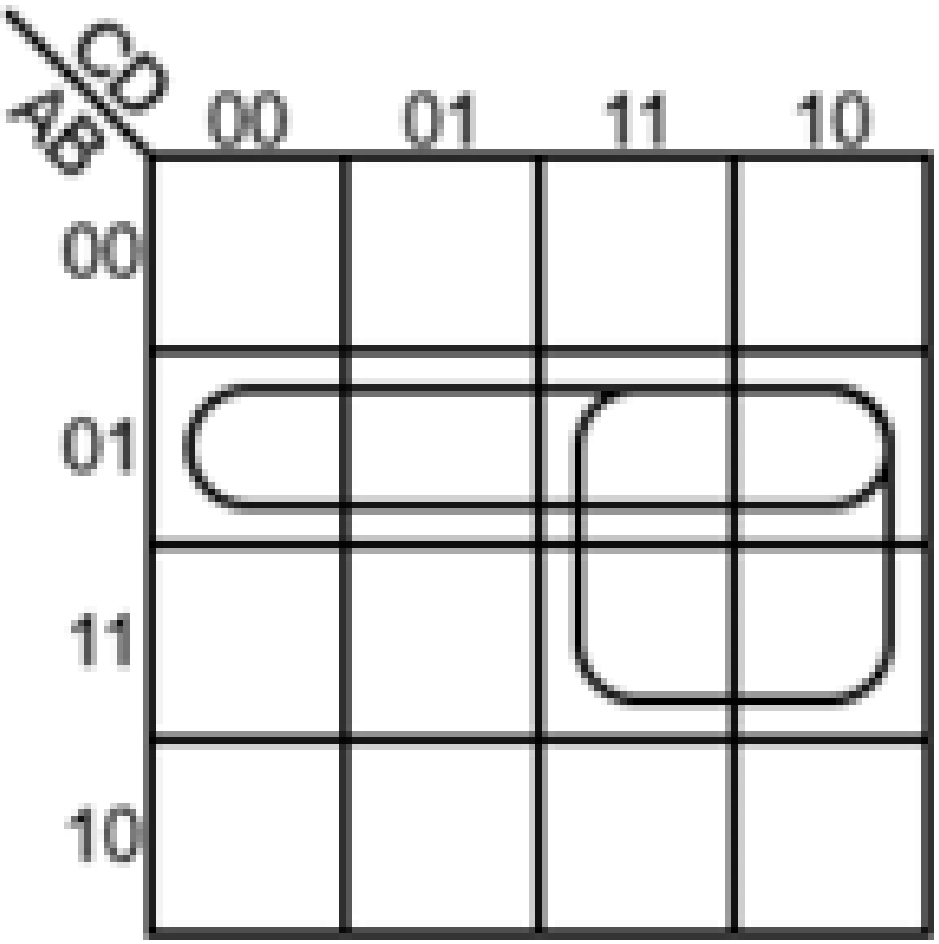
Rysunek 7: Tablica Karnaugh dla funkcji Q_4



Boolean Algebra

$$\overline{A}C + AB\overline{C}$$

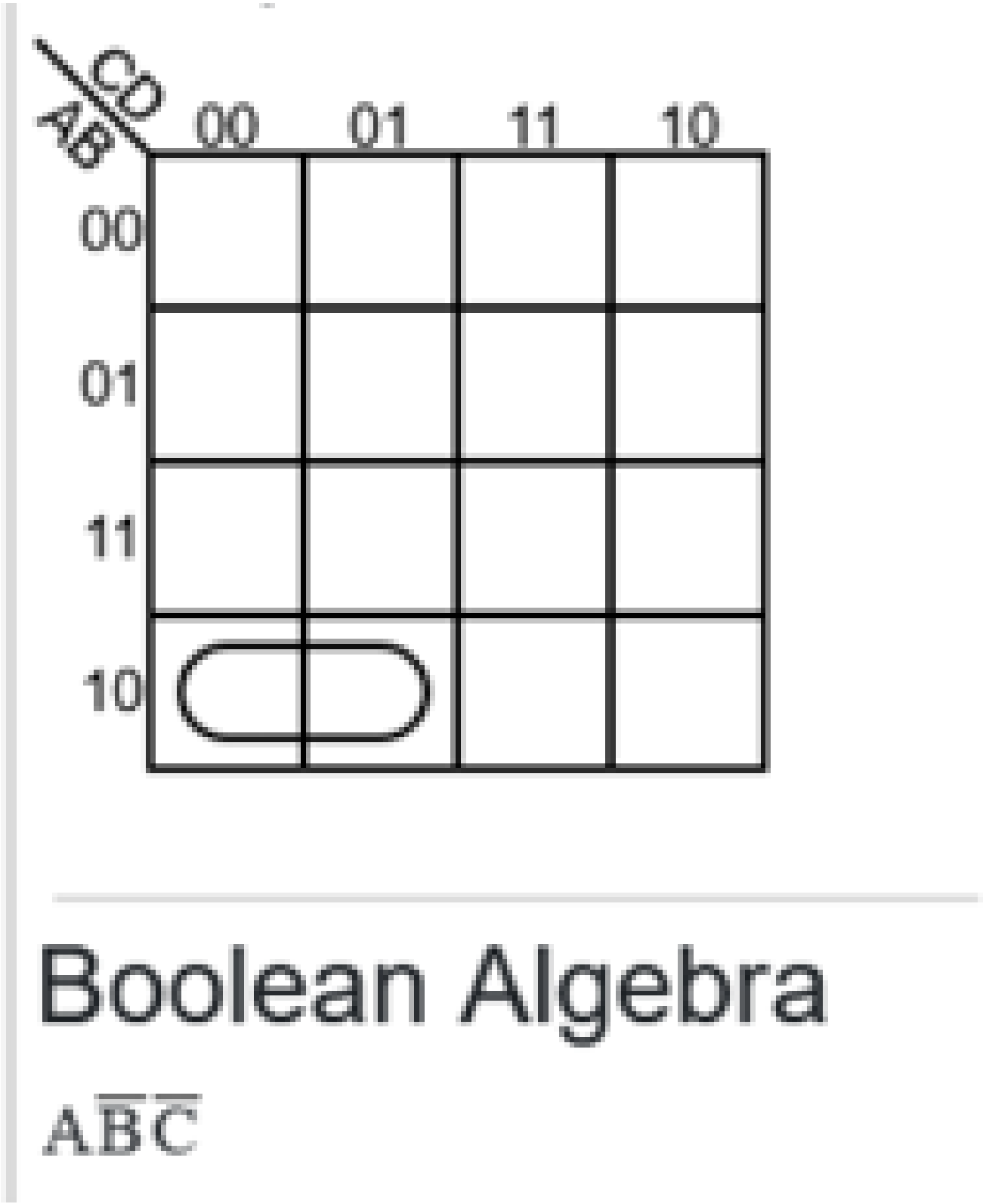
Rysunek 8: Tablica Karnaugh dla funkcji Q_5



Boolean Algebra

$\overline{A}B+BC$

Rysunek 9: Tablica Karnaugh dla funkcji Q_6

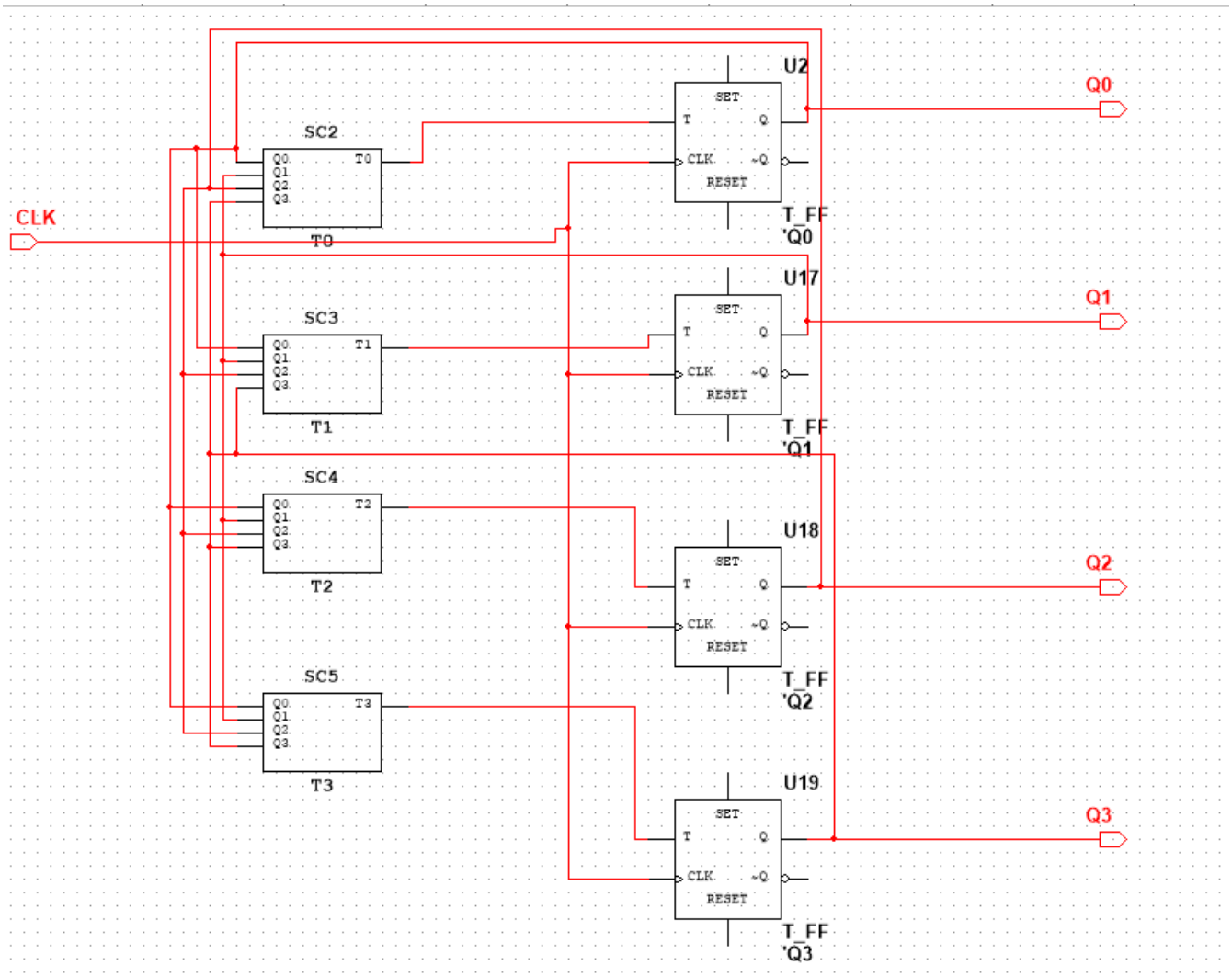


Rysunek 10: Tablica Karnaugh dla funkcji Q_7

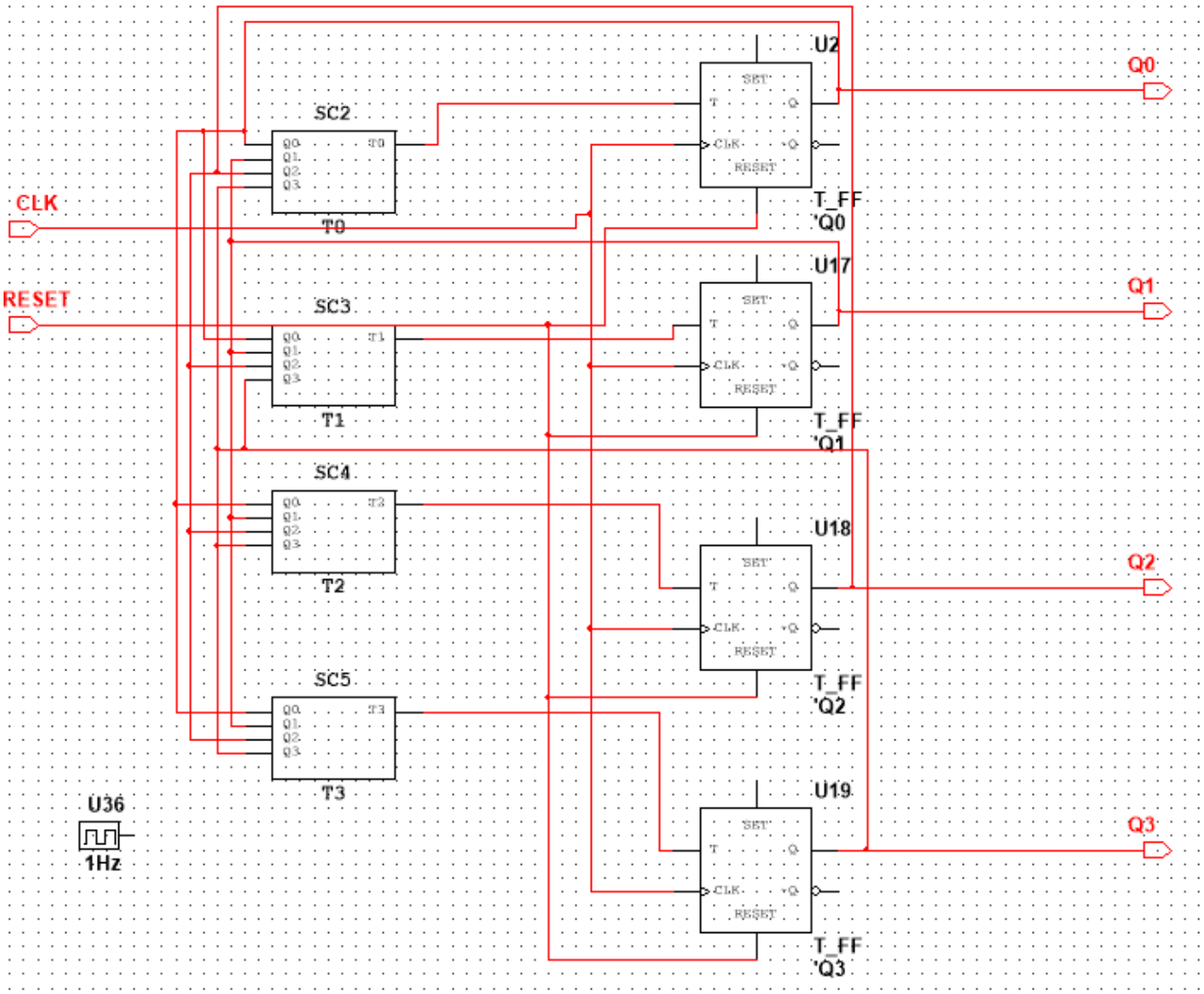
3 Schemat układu

W oparciu o wyprowadzone na rysunkach 2-10 wzory na sygnały wyjściowe, oraz poczynione we wprowadzeniu założenia możemy stworzyć schemat licznika w programie `Multisim`.

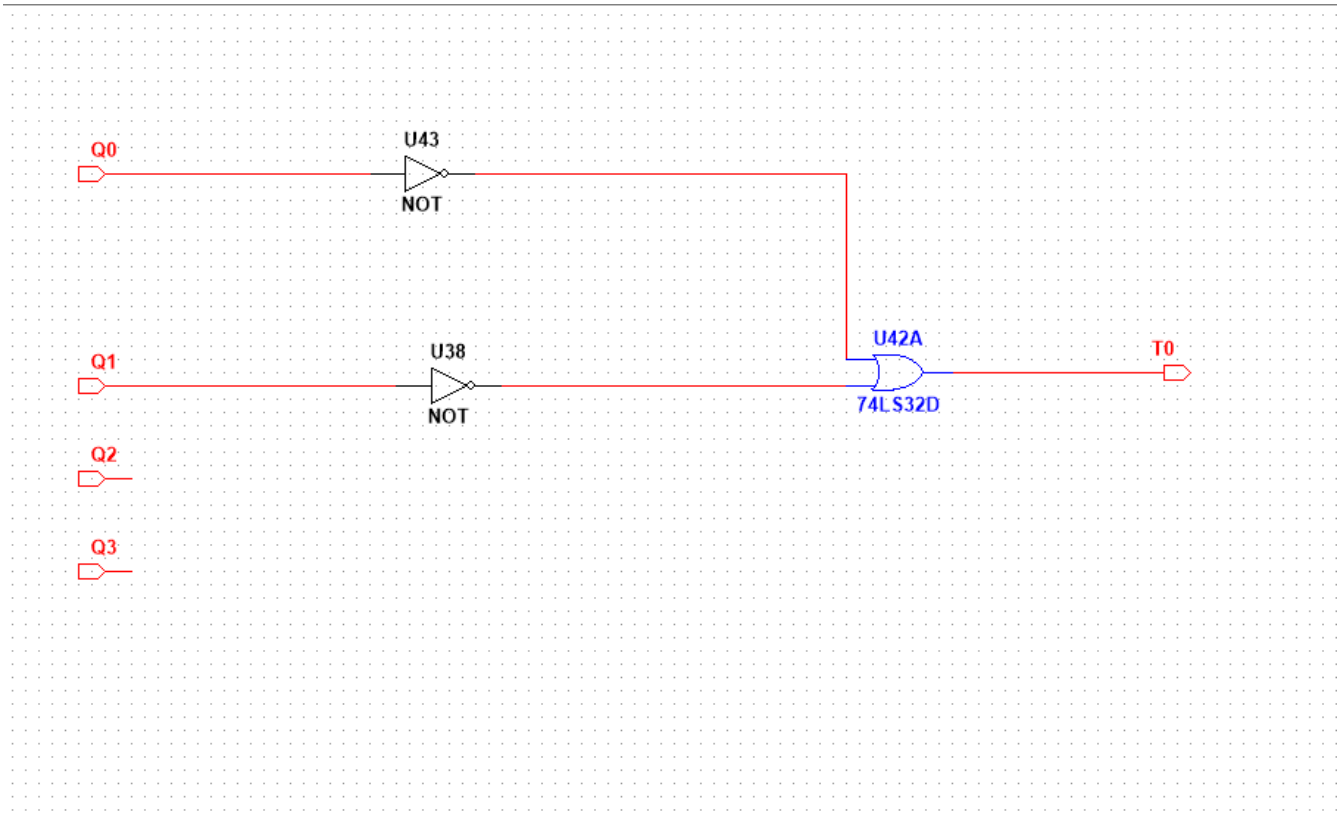
Nasz licznik wzbogacamy dotatkowo o wejście dla sygnału resetującego, które podpinamy bezpośrednio do przerzutników typu T, co będzie przydatne podczas projektowania stanowiska testującego.



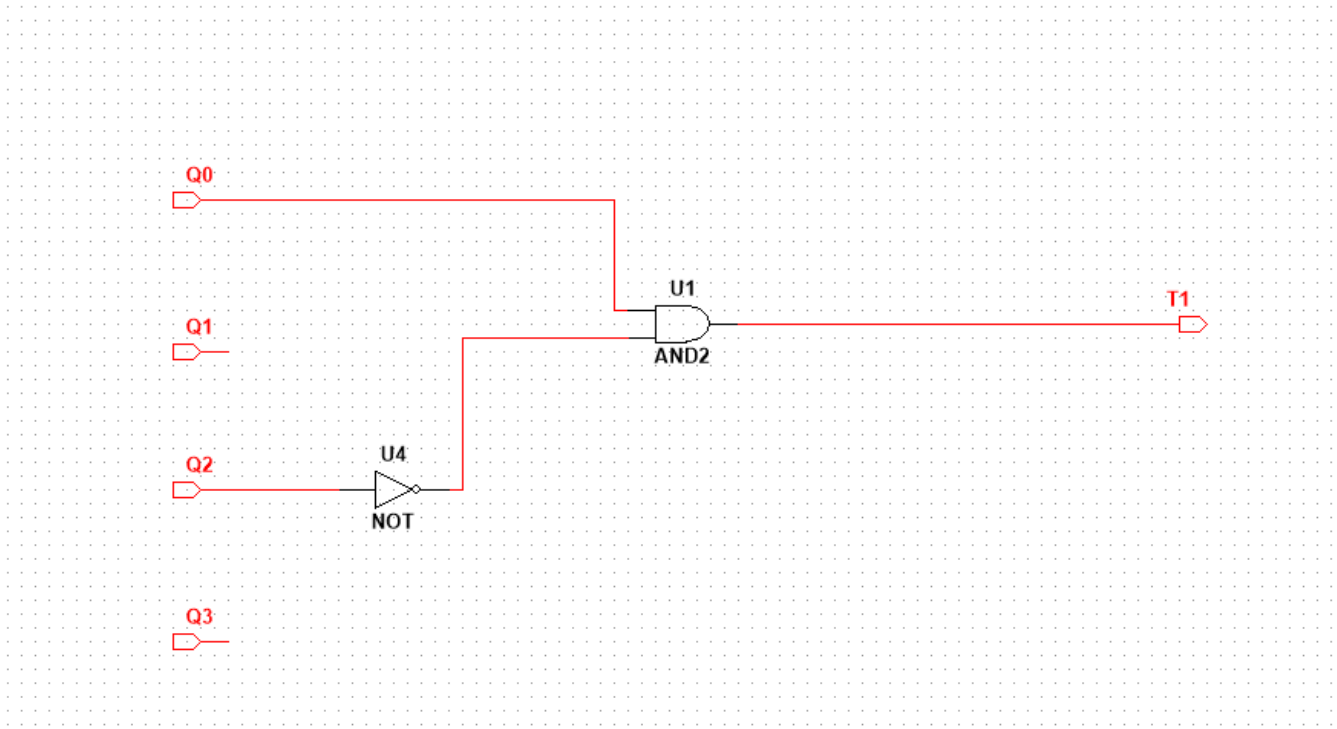
Rysunek 11: Schemat podukładu `blackbox` z podziałem na bramki w programie `Multisim`



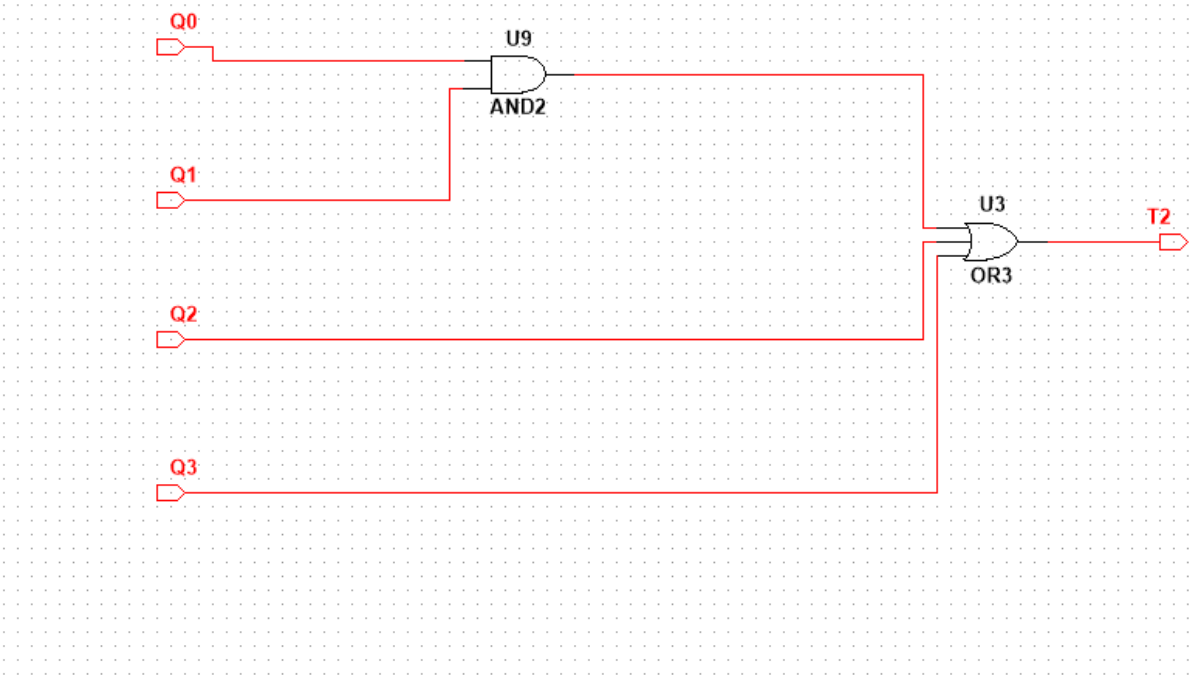
Rysunek 12: Schemat podukładu `blackbox` z dodanym wejściem dla sygnału resetującego



Rysunek 13: Bramka obliczająca wartości funkcji T_0

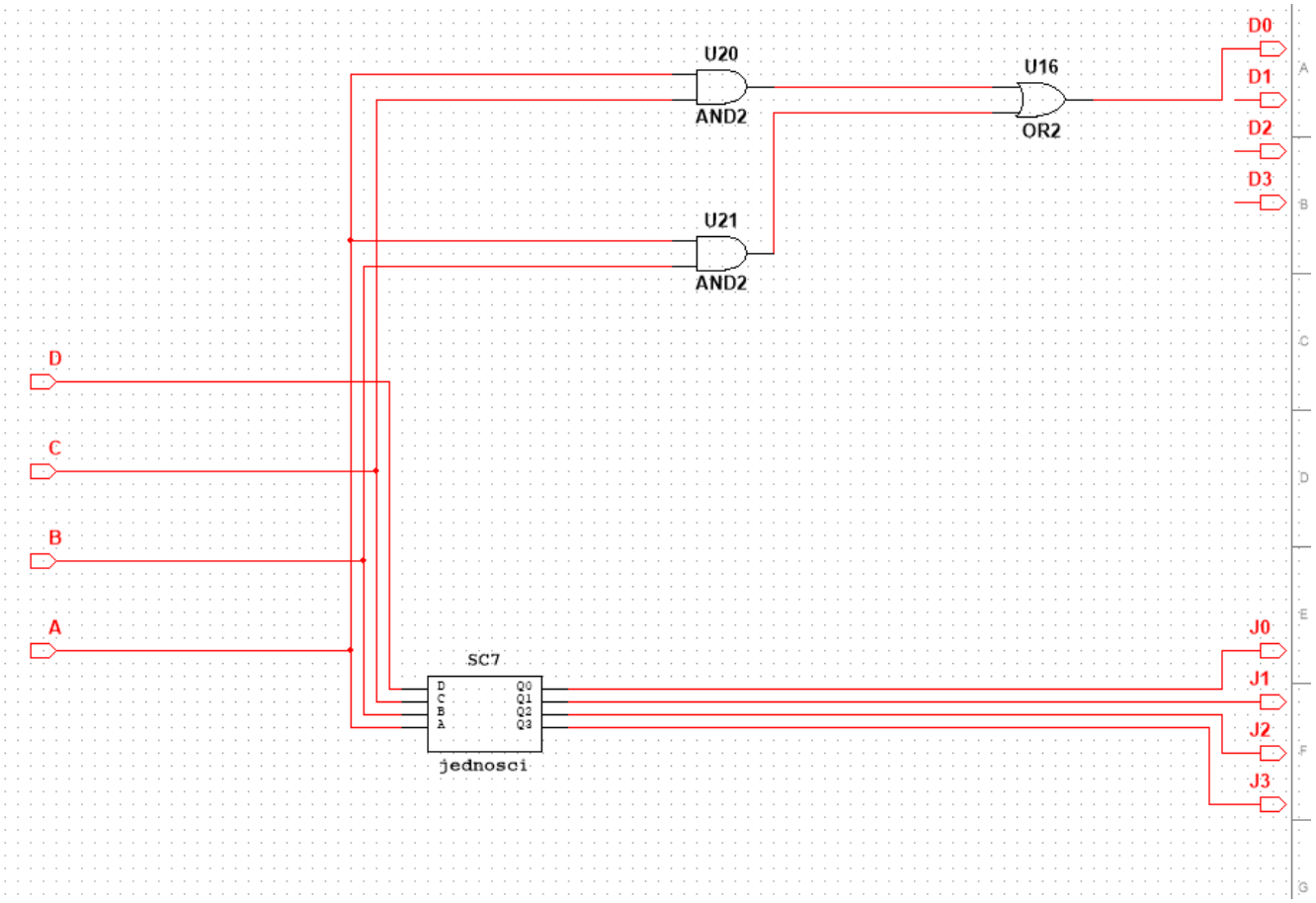


Rysunek 14: Bramka obliczająca wartości funkcji T_1

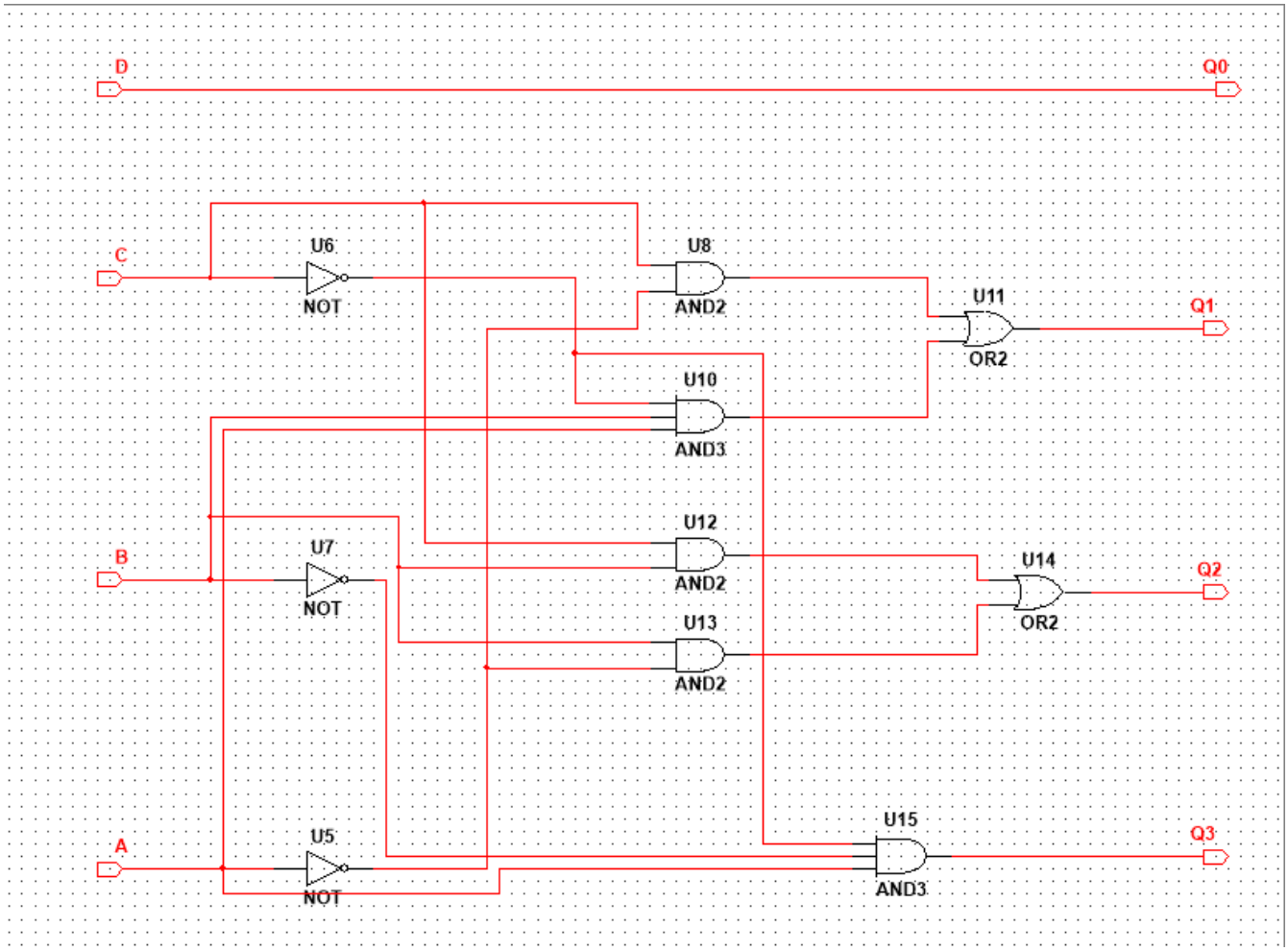


Rysunek 15: Bramka obliczająca wartości funkcji T_2

Rysunek 16: Bramka obliczająca wartości funkcji T_3



Rysunek 17: Schemat podukładu `decoder` (na rysunku $D_0, \ldots D_3$ odpowiada $Q_0, \ldots, Q_3$ natomiast $J_0, \ldots, J_3$ $Q_4, \ldots, Q_7$)



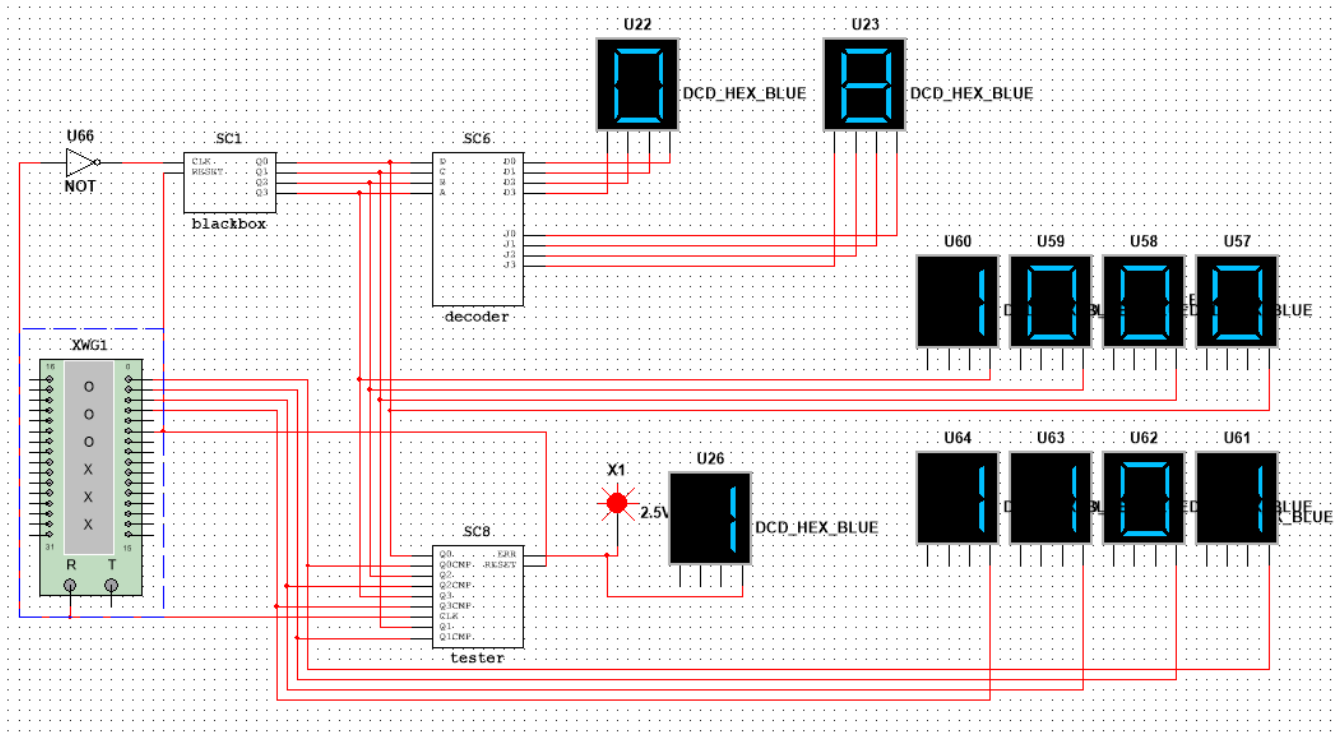
Rysunek 18: Bramka obliczająca wartości funkcji $Q_4, \ldots, Q_7$ (na rysunku $Q_0, \ldots, Q_3$)

4 Stanowisko testujące

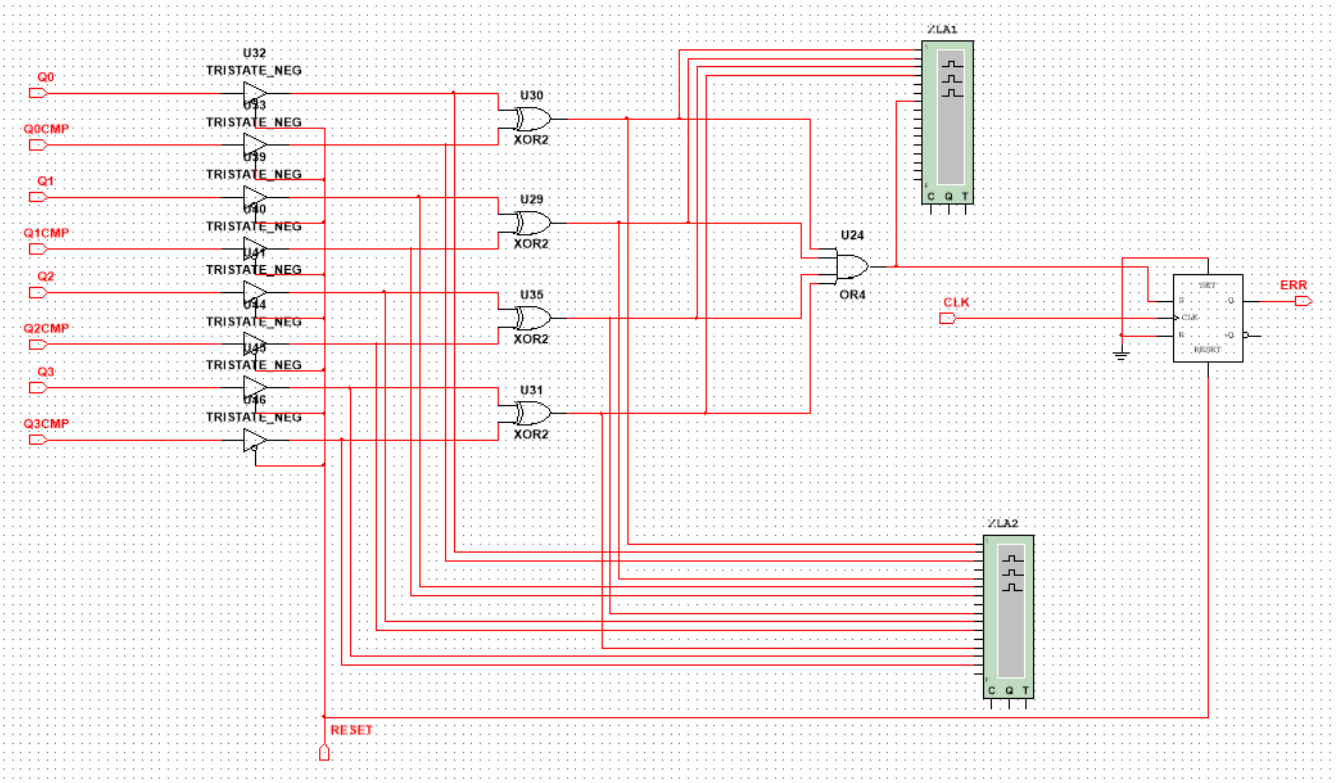
Głównym elementem układu testującego jest podukład `tester` porównujący sygnały wysyłane przez testowany licznik z sygnałami docelowymi nadawanymi równolegle przez generator słów (`XWG1`) i za pomocą przerzutnika SR wysyłający sygnał `ERR` z wartością 1 w przypadku niezgodności, co skutkuje zapaleniem diody i poinformowaniem użytkownika o błędzie.

Stanowisko umożliwia resetowanie stanu testowanego układu w dowolnym momencie w celach diagnostycznych, oraz badanie wewnętrzego stanu podukładu `tester` za pomocą analizatorów stanów logicznych (`XLA1` oraz `XLA2`). Cały układ podpięty jest do wspólnego zegara co gwarantuje poprawność wyników testów.

Wyjścia testowanego układu oraz generatora słów wyświetlane są na wyświetlaczach siedmiosegmentowych co umożliwia stałą kontrolę działania oraz szczegółowe badanie rozbieżności.



Rysunek 19: Podłączenie układu testującego



Rysunek 20: Podukład `tester`

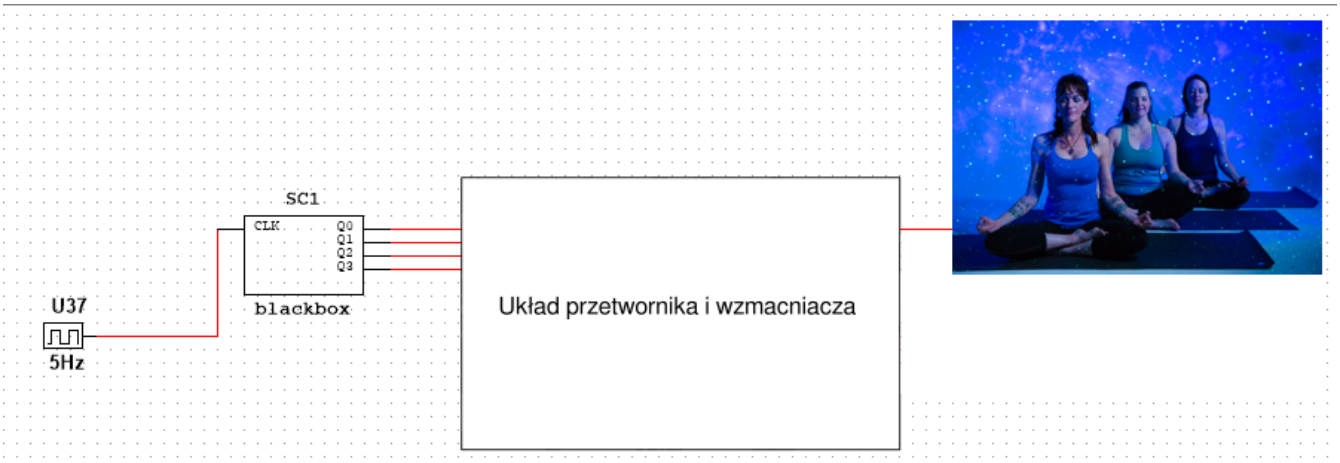
5 Podsumowanie oraz wnioski

Najważniejszym elementem projektu było poprawne zdefiniowanie funkcji sterujących przerzutnikami oraz wyjściem dekodera konwertującego stan licznika na wyświetlacz siedmiosegmentowy. Wybór rozwiązania opartego o przerzutniki typu T (chociaż nie jedyny możliwy, można było zastosować także chociażby prerzutniki typu JK) znacznie ułatwił ten

etap oraz uzasadnienie poprawności proponowanego rozwiązania. Bardziej standardowy podukład `decoder` również mógłby wyglądać zupełnie inaczej, gdyby zdecydowano się na inną metodę wyświetlania (a co za tym idzie konwersji), co jednak przełożyłoby się również na dodatkowe skomplikowanie układu.

Praktyczne zastosowania zaprojektowanego licznika to m.in.:

- Sterowanie animacjami w reklamach cyfrowych, np. dynamiczne zmiany wyświetlanych wzorów na billboardach LED, wykorzystujące niestandardowe sekwencje do przyciągnięcia uwagi.
- Moduły czasowe w systemach IoT do generowania nieregularnych interwałów sygnałowych, np. w celu redukcji kolizji pakietów w sieci.
- Generatory muzyczne
- Biomimetyczne systemy oświetleniowe



Rysunek 21: Przykładowe zastosowanie układu - oświetlenie o "naturalnym"wzroście intensywności

Wykonane zadanie jest dobrym ćwiczeniem z zakresu projektowania niestandardowych liczników opartych na logice kombinacyjnej oraz zastosowania i wyboru odpowiedniego rodzaju przerzutników w projektowanych układach.