Technika Cyfrowa 2022

Sprawozdanie 1

Gabriel Kaźmierczak,

Jakub Radek,

Miłosz Dobosz,

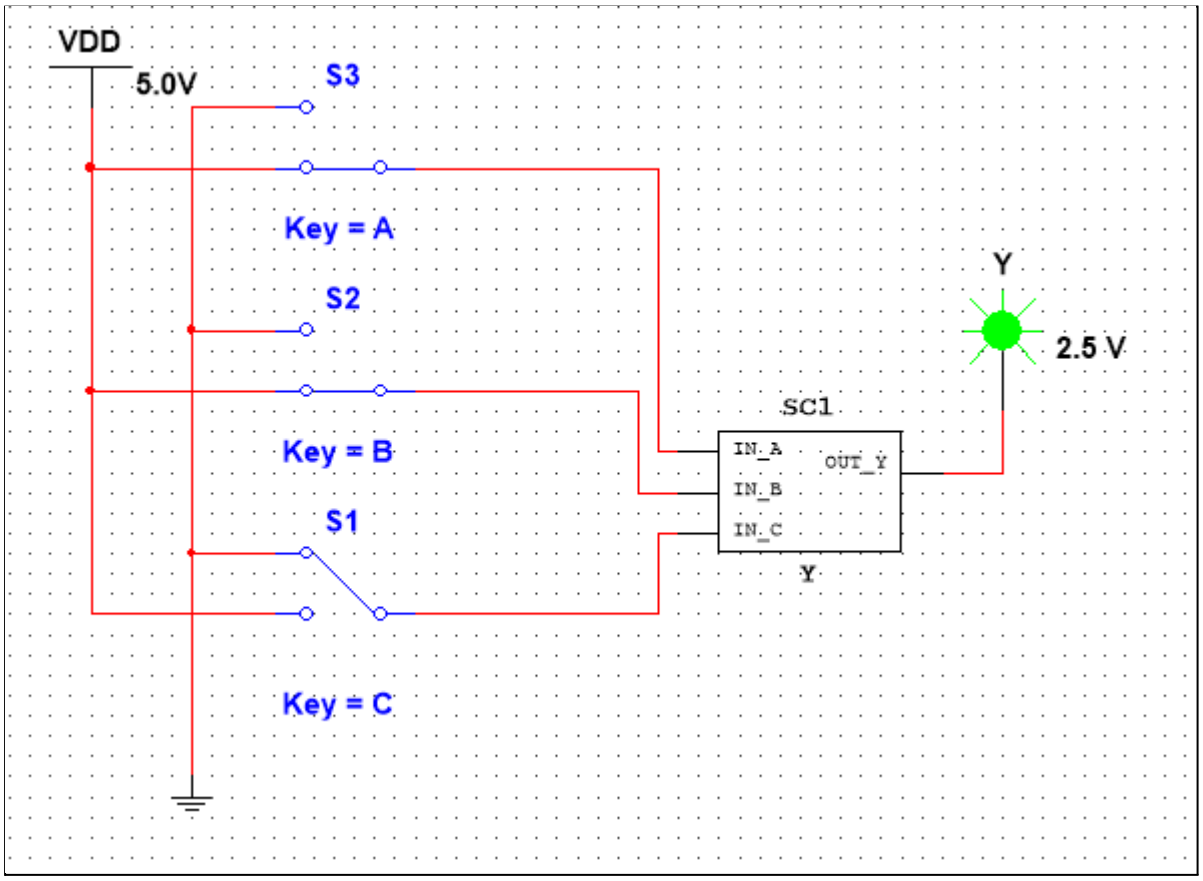
Przemysław Kociuba

Zadanie 1a

Bazując wyłącznie na dwuwejściowych bramkach logicznych NAND, proszę od podstaw zaprojektować, zbudować i przetestować układ realizujący funkcję logiczną:

$$Y = \overline{A}C + B(A + B)$$

Idea układu



Rysunek 1: Idea budowanego układu

Rozwiązanie teoretyczne

Wyrażenie zostało przekształcone do postaci funkcji do zastosowania dla dwuwejściowych bramek NAND

$Y = \overline{A}C + B(A + B)$
$Y = \overline{A}C + BA + BB$ (rozdzielność mnożenia wzgl. dodawania)
$Y = \overline{A}C + BA + B \ (B = B * B)$
$Y = \overline{A}C + B(A + 1)$ (wyciągnięcie B przed nawias)
$Y = \overline{A}C + B \ (A + 1 = 1)$

$Y = \overline{\overline{\overline{AC + B}}}$ (podwójna negacja nie zmienia wartości)

$Y = \overline{\overline{\overline{AC}}\,\overline{B}}$ (prawo de Morgana)

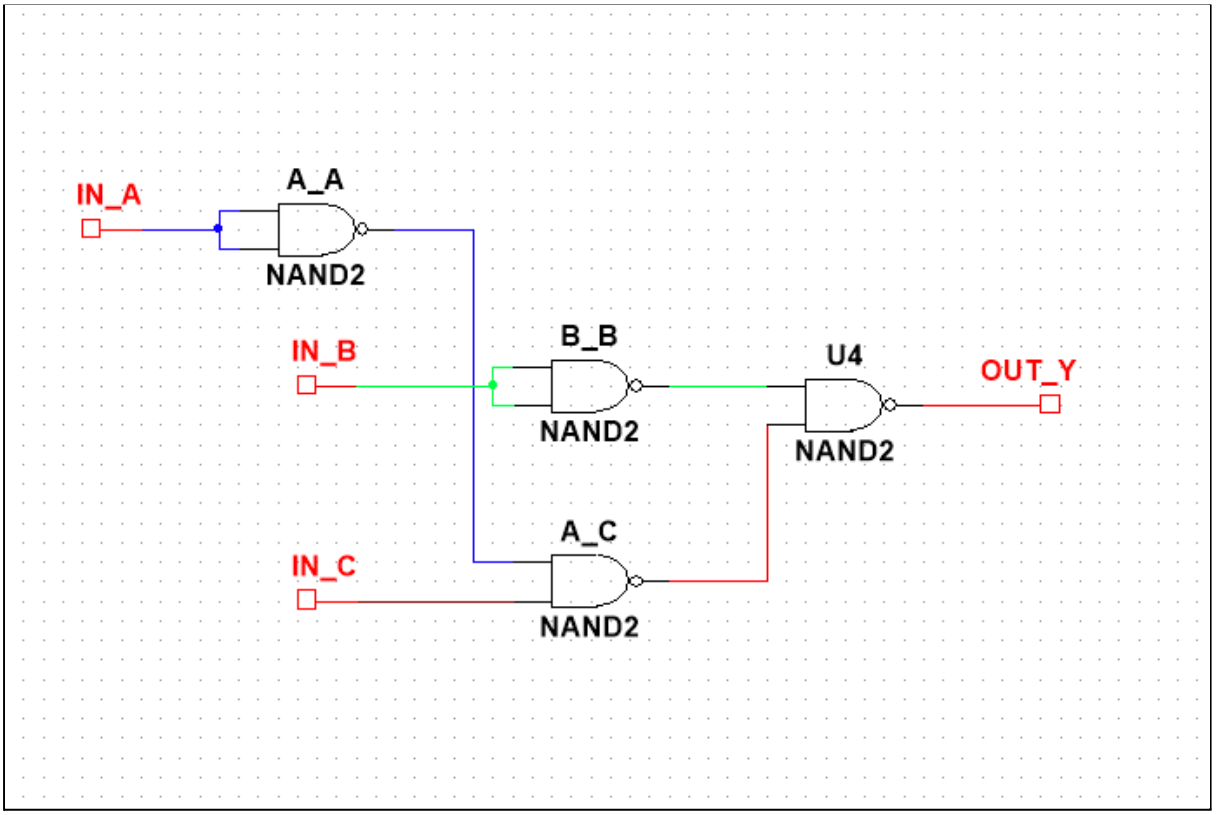
$Y = \overline{\overline{\overline{AA}\,\overline{C}}\,\overline{\overline{B}\,\overline{B}}}$ $(A = A * A)$

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Rysunek 2: Tabela prawdy dla powyższego równania

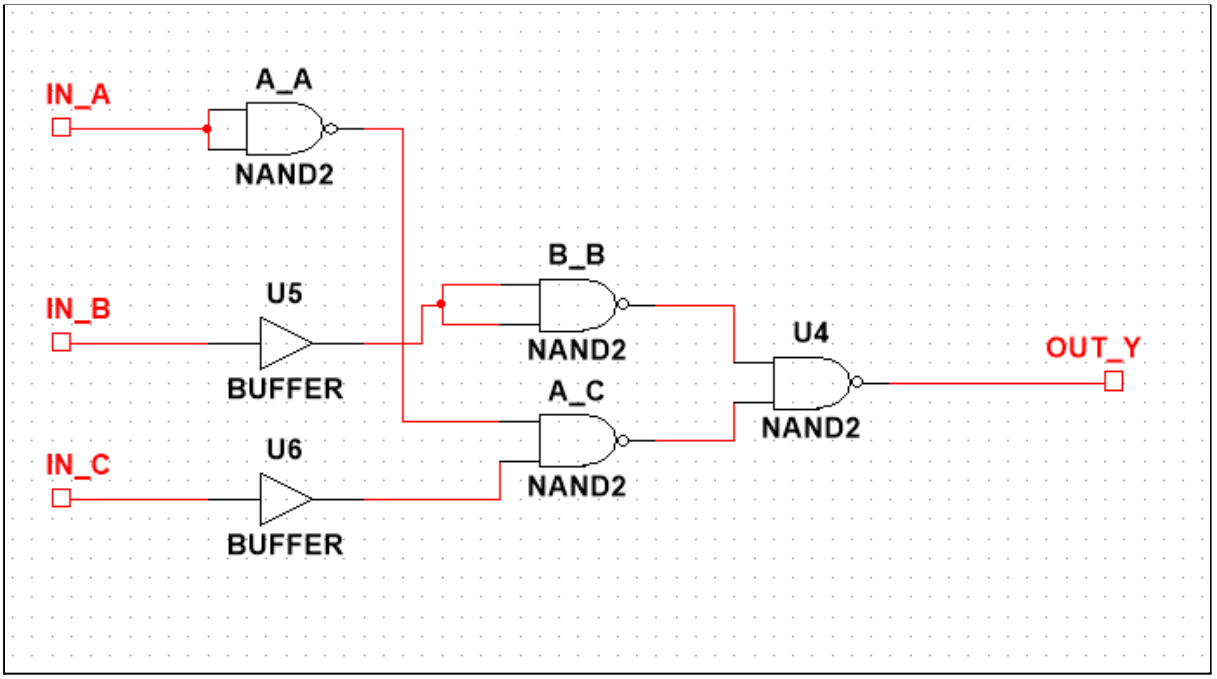
Implementacja w programie Multisim

Do zbudowania układu realizującego przekształconą funkcję wykorzystano cztery bramki NAND



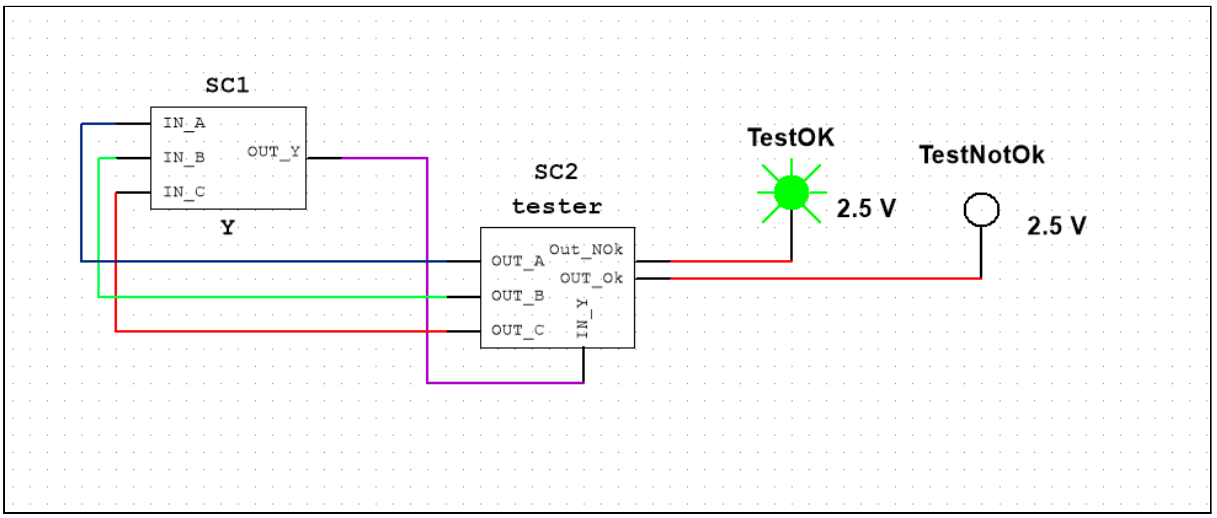
Rysunek 3: Układ dla zadania 1a w programie Multisim

W celu wyeliminowania opóźnień zostały dodane bufory.



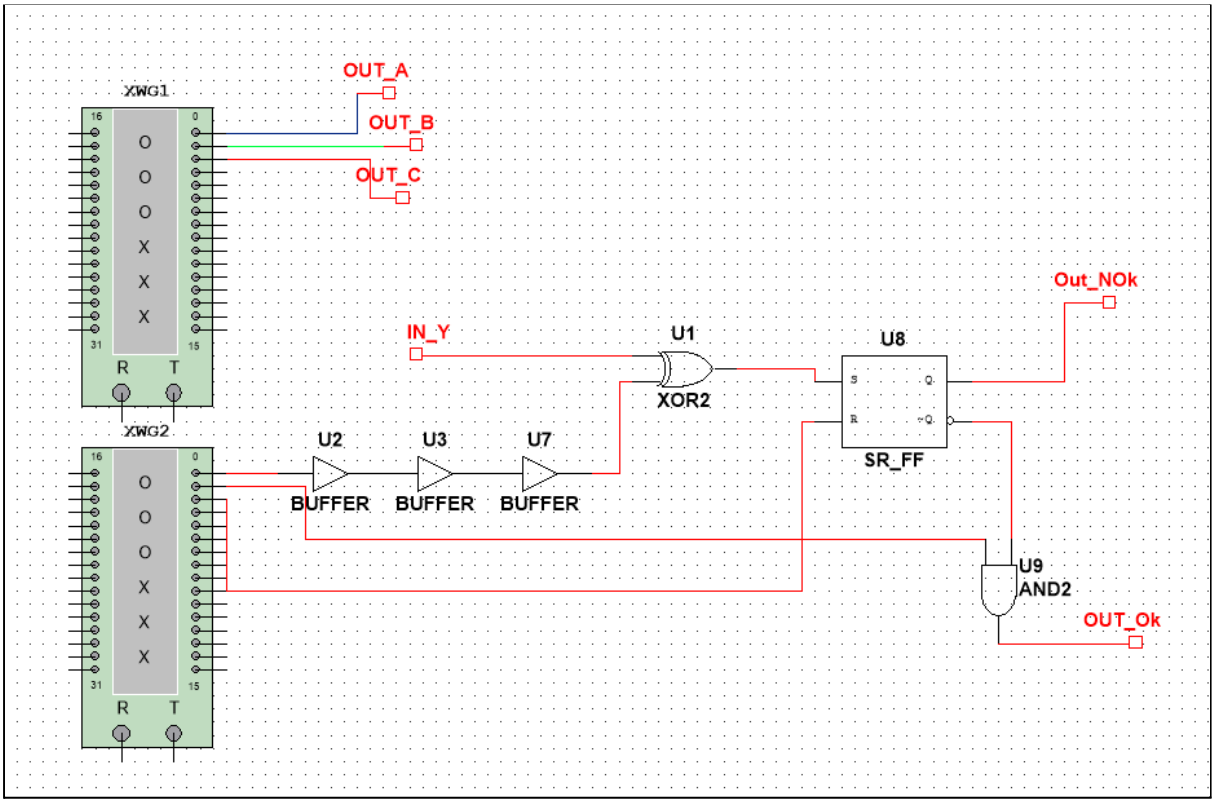
Rysunek 4: Implementacja układu z buforami

Idea układu testującego

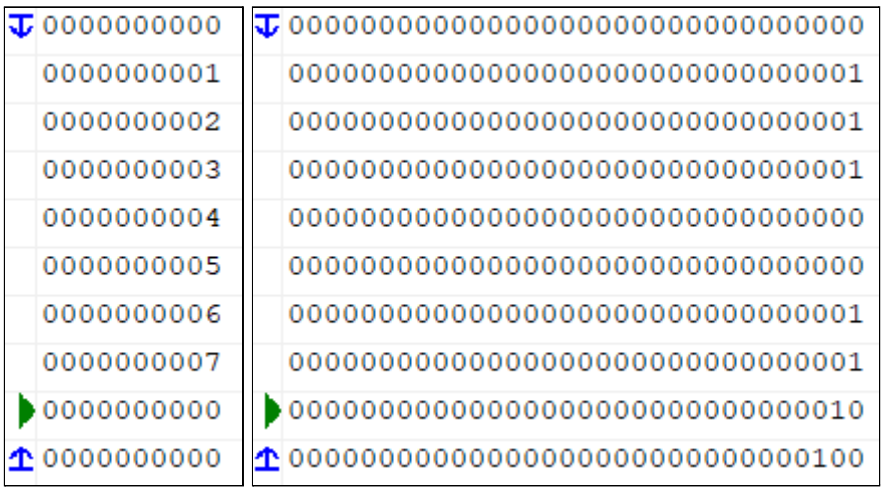


Rysunek 5: Idea układu testującego

Implementacja układu testującego



Rysunek 6: Implementacja układu testującego



Rysunek 7: Słowa generowane przez generatory słów w układzie testującym (XWG1 po lewej i XWG2 po prawej)

Wnioski

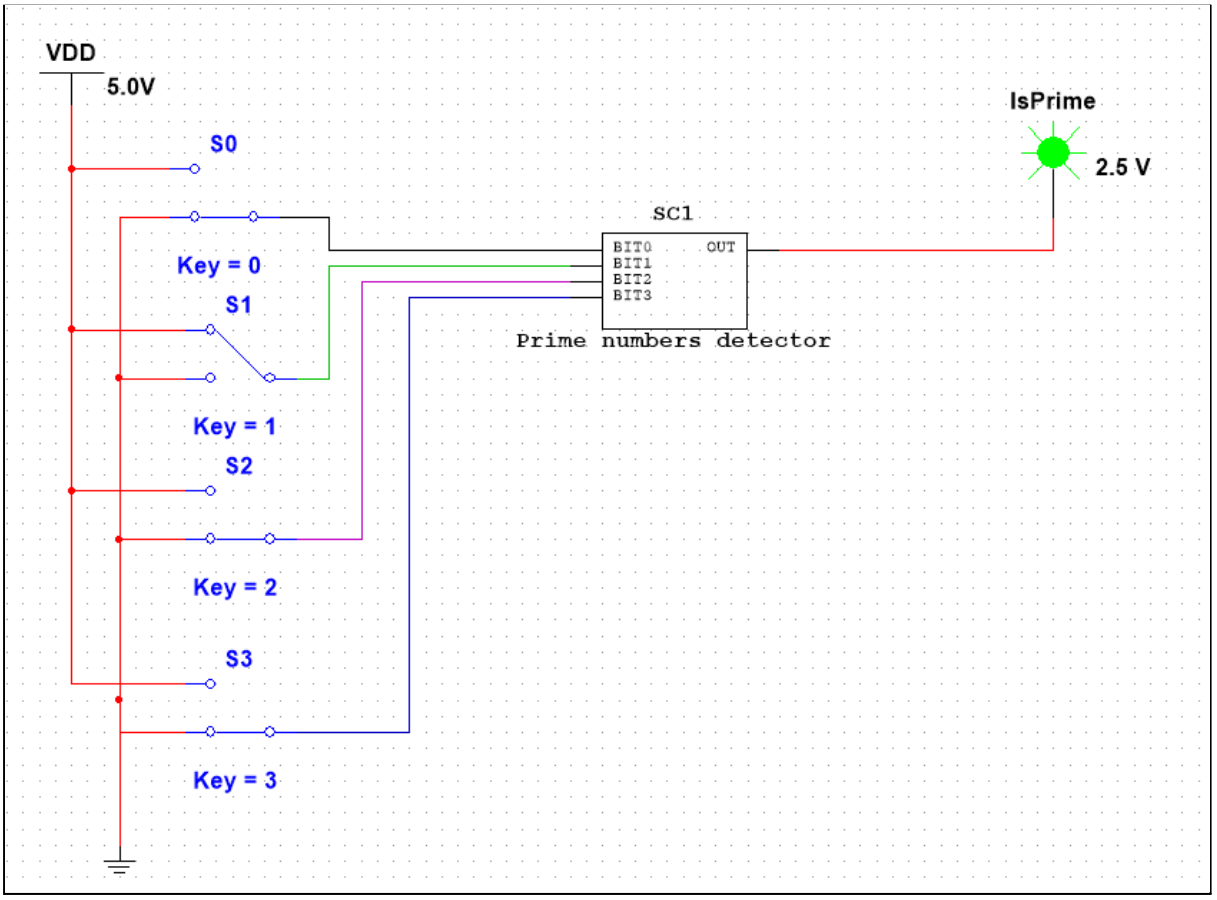
- Korzystając z praw logiki możemy zmniejszyć liczbę użytych elementów elektronicznych, tutaj 5 bramek - NOT, dwie AND oraz dwie OR (które są złożonymi bramkami) na 4 mniej skomplikowane bramki NAND.
- Układ ten może zostać zastosowany jako kontroler świateł nocnych (np. w samochodzie):

- A - Czujnik światła
- B - Tryb ON
- C - Tryb AUTO.

Zadanie 1b

Bazując na dowolnych bramkach logicznych, proszę od podstaw zaprojektować, zbudować i przetestować układ detekcji liczby pierwszej w binarnym słowie czterobitowym.

Idea układu



Rysunek 7: Idea budowanego układu

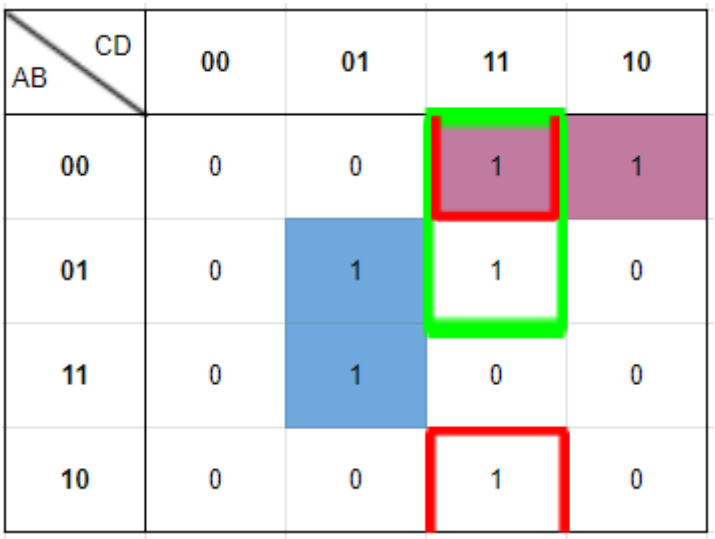
Rozwiązanie teoretyczne

Wprowadzamy oznaczenia A,B,C,D gdzie A oznacza najbardziej znaczący bit, a D najmniej znaczący.
Aby zaprojektować układ wykorzystaliśmy funkcję zapisaną w notacji sumarycznej.

$$f(A,\ B,\ C,\ D)\ =\ \sum(2,\ 3,\ 5,\ 7,\ 11,\ 13)$$

A	B	C	D	PRIME
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

Rysunek 8: Tabela prawdy dla budowanego układu



Rysunek 9: Tabela Karnough dla badanej funkcji

Korzystając z tabeli uzyskujemy postać funkcji logicznej

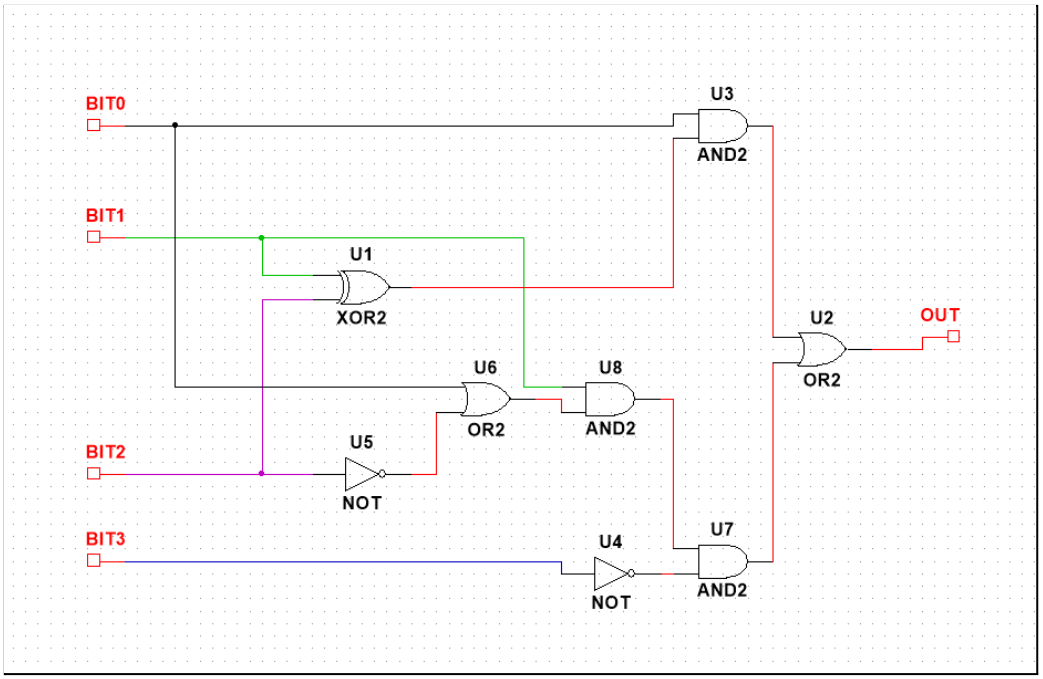
$Y = \overline{A}\,\overline{B}C + \overline{A}BC + \overline{B}CD + B\overline{C}D$

$Y = \overline{A}(\overline{B}C + CD) + D(\overline{B}C + B\overline{C})$ (Wyciągnięcie $\overline{A}$ i D przed nawias)

$Y = \overline{A}(C(\overline{B} + D)) + D(B\ xor\ C)$ (Wyciągnięcie C przed nawias i $\overline{B}C + B\overline{C} = B\ xor\ C$)

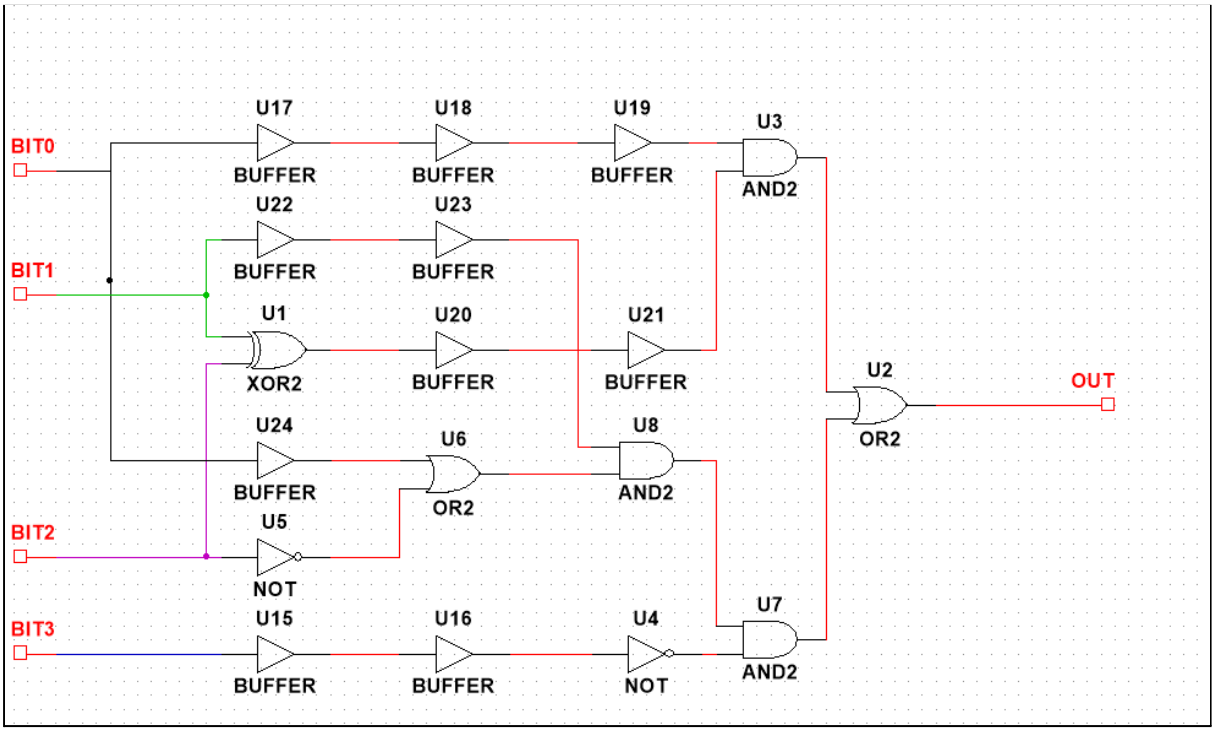
Implementacja w programie Multisim

Do zbudowania układu realizującego powyższą funkcję wykorzystano dwie bramki NOT, trzy bramki AND, bramkę OR oraz bramkę XOR.



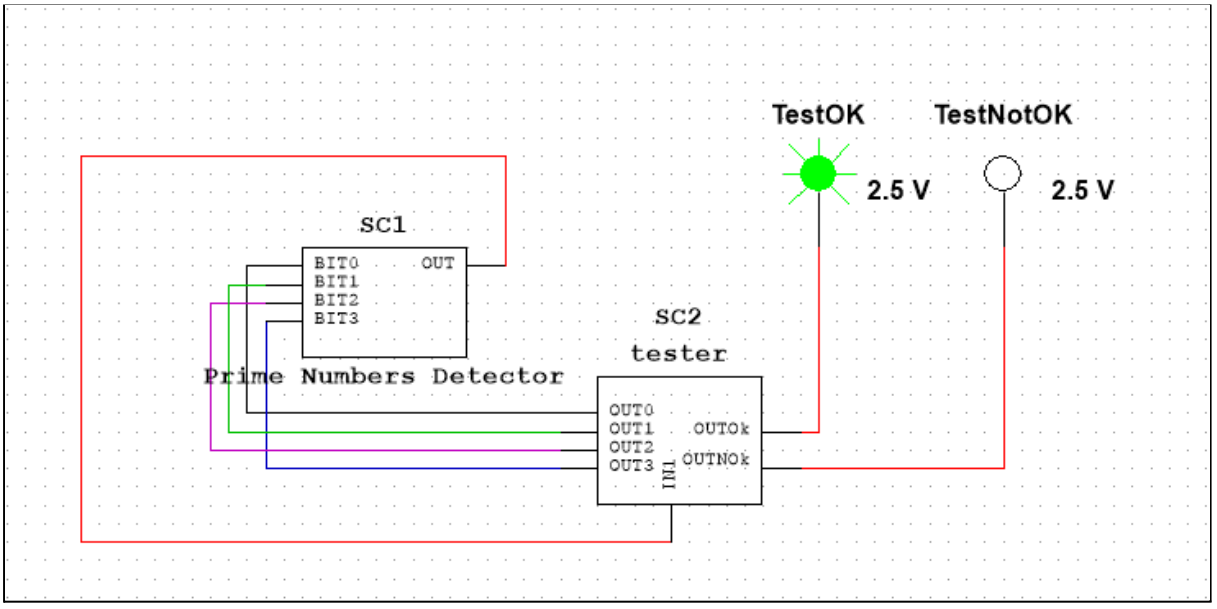
Rysunek 11: Układ dla zadania 1b w programie Multisim

Ze względu na opóźnienia w przesyle sygnału zostały dodane bufory



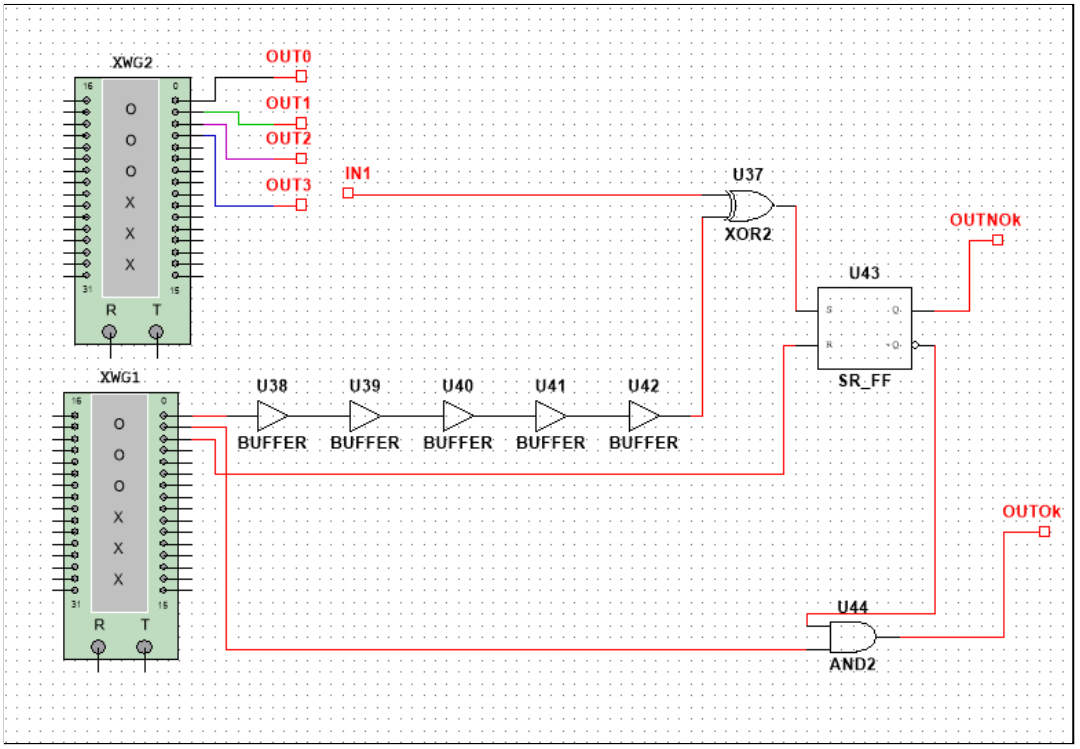
Rysunek 12: Układ dla zadania 1b w programie Multisim z buforami

Idea układu testującego

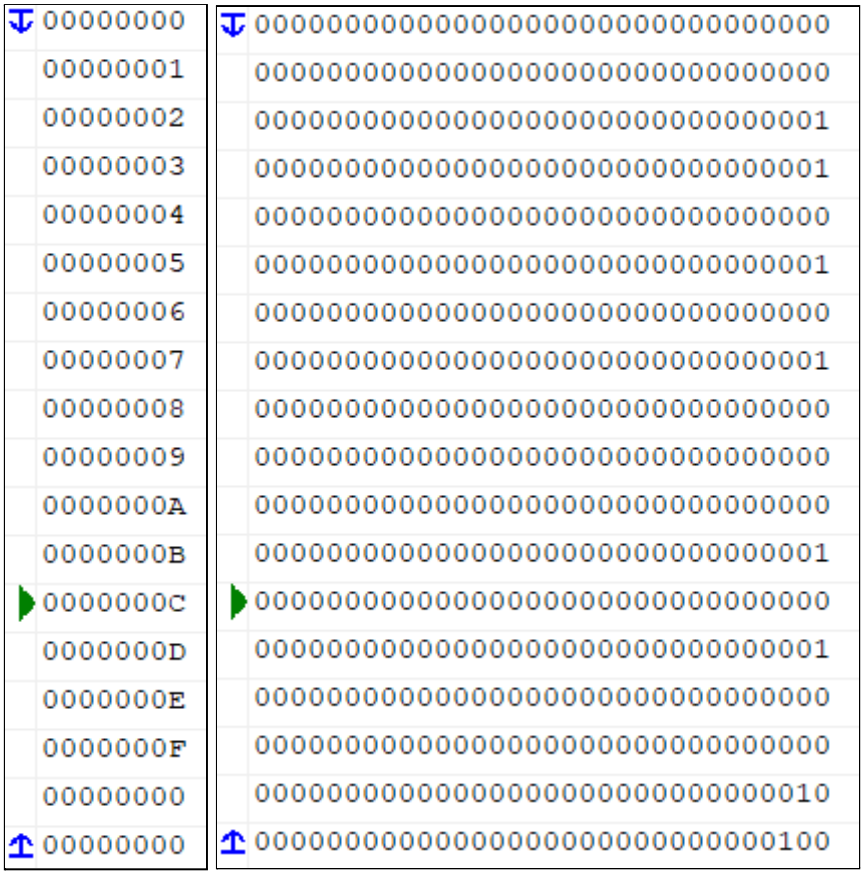


Rysunek 13: Idea układu testującego dla zadania 1b

Implementacja układu testującego



Rysunek 14: Idea układu testującego dla zadania 1b



Rysunek 15: Słowa generowane w układzie testującym przez XWG2 (po lewej) i XWG1 (po prawej)

Wnioski

- Tabelę Karnaugh można wykorzystać do łatwego generowania funkcji logicznych w oparciu o tabelę prawdy.
- W oparciu o układ można utworzyć kontroler poprawności danych przesyłanych sieciowo (np. ustalając że komunikat rozpoczyna się od n liczb pierwszych i kończy się m liczbami pierwszymi, taki układ pozwoli na poprawę detekcji zakłóceń). Tracąc częściowo na wydajności można również szyfrować przesyłane dane np. wplatając co każdy 4 bitowy sygnał 2 sygnały będące liczbami pierwszymi. Wtedy opisywany układ służy jako deszyfrator i odrzuca sygnały zakłócające komunikat.