

# Ćwiczenie 1

## Transkoder czterobitowej liczby naturalnej na sześciobitową liczbę pierwszą przy użyciu bramek NAND

Antoni Smółka , Krystian Madej, Łukasz Wilański, Tomasz Żmuda

28.03.2024

### 1 Wprowadzenie

Aby zaprojektować transkoder na bramkach NAND, przekształcający czterobitowe liczby naturalne na sześciobitowe reprezentacje liczb pierwszych, należy najpierw zdefiniować mapy Karnaugh dla każdego z sześciu bitów wyjściowych, minimalizując funkcje logiczne do formy z bramkami NAND. Następnie, zbudować układ cyfrowy z wykorzystaniem tych bramek i przetestować jego działanie za pomocą wyświetlaczy siedmiosegmentowych, generatora słów i analizatora stanów logicznych, weryfikując zgodność wyjścia z oczekiwanymi wartościami liczb pierwszych.

## 2 Tabela Prawdy

WEJŚCIE	WARTOŚĆ WEJŚCIA	WYJŚCIE	WARTOŚĆ WYJŚCIA
0000	0	000010	2
0001	1	000011	3
0010	2	000101	5
0011	3	000111	7
0100	4	001011	11
0101	5	001101	13
0110	6	010001	17
0111	7	010011	19
1000	8	010111	23
1001	9	011101	29
1010	10	011111	31
1011	11	100101	37
1100	12	101001	41
1101	13	101011	43
1110	14	101111	47
1111	15	110101	53

Rysunek 1: Tabela Prawdy

### 3 Tablice Karnaugh

**BIT 1**

CD \ AB	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	0

**BIT 2**

CD \ AB	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	0	0	1	0
10	1	1	0	1

**BIT 3**

CD \ AB	00	01	11	10
00	0	0	0	0
01	1	1	0	0
11	1	1	0	1
10	0	1	0	1

**BIT 4**

CD \ AB	00	01	11	10
00	0	0	1	1
01	0	1	0	0
11	0	0	1	1
10	1	1	1	1

**BIT 5**

CD \ AB	00	01	11	10
00	1	1	1	0
01	1	0	1	0
11	0	1	0	1
10	1	0	0	1

**BIT 6**

CD \ AB	00	01	11	10
00	0	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$$y_1 = ab + acd =$$

$$y_2 = \bar{a}bc + a\bar{b}\bar{c} + a\bar{b}\bar{d} + bcd$$

$$y_3 = b\bar{c} + a\bar{c}d + ac\bar{d}$$

$$y_4 = \bar{b}c + \bar{a}b\bar{c}d + a\bar{b} + ac$$

$$y_5 = \bar{a}\bar{c}\bar{b} + \bar{a}cd + ab\bar{c}d + ac\bar{d} + \bar{a}\bar{b}\bar{c} + \bar{b}\bar{c}\bar{d}$$

$$y_6 = a + b + c + d$$

3

Rysunek 2: Tablice Karnaugh dla wszystkich bitów

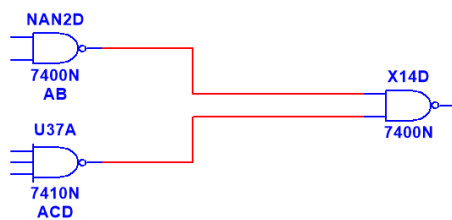
$$\begin{aligned}
 y_1 &= ab + acd = \overline{\overline{ab} \cdot \overline{acd}} \\
 y_2 &= \overline{a}bc + a\overline{b}\overline{c} + a\overline{b}d + bcd = \overline{\overline{abc} \cdot \overline{a\overline{b}\overline{c}} \cdot \overline{a\overline{b}d} \cdot \overline{bcd}} \\
 y_3 &= b\overline{c} + a\overline{c}d + acd = \overline{\overline{b\overline{c}} \cdot \overline{a\overline{c}d} \cdot \overline{acd}} = \overline{\overline{b\overline{c}} \cdot \overline{a\overline{c}d} \cdot \overline{acd}} \\
 y_4 &= \overline{b}c + \overline{a}b\overline{c}d + a\overline{b} + ac = \overline{\overline{\overline{b}c} \cdot \overline{\overline{a}b\overline{c}d} \cdot \overline{a\overline{b}} \cdot \overline{ac}} \\
 y_5 &= \overline{a}\overline{c}d + \overline{a}cd + ab\overline{c}d + acd + a\overline{b}\overline{c} + \overline{b}cd = \overline{\overline{\overline{a}\overline{c}d} \cdot \overline{\overline{a}cd} \cdot \overline{ab\overline{c}d} \cdot \overline{acd} \cdot \overline{a\overline{b}\overline{c}} \cdot \overline{\overline{b}cd}} \\
 y_6 &= a + b + c + d = \overline{\overline{a} \cdot \overline{b} \cdot \overline{c} \cdot \overline{d}}
 \end{aligned}$$

Rysunek 3: Przekształcenia zgodne z algebrą Boole'a na postać iloczynową

## 4 Układy dla poszczególnych bitów

### 4.1 Bit 1

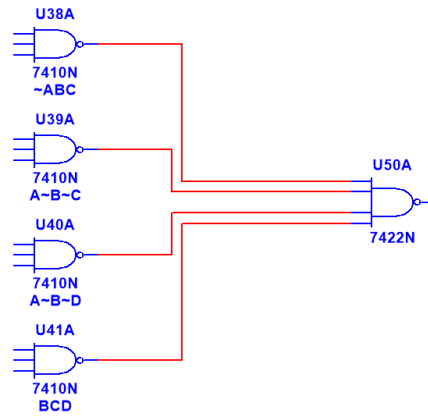
$$Y_1 = \overline{AB} \cdot \overline{ACD} \quad (1)$$



Rysunek 4: Układ dla pierwszego bitu wyjścia

## 4.2 Bit 2

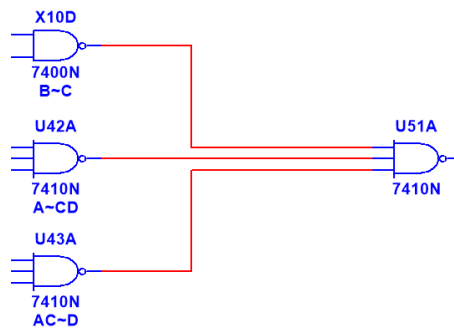
$$Y_2 = \overline{ABC} \cdot \overline{ABC} \cdot \overline{ABD} \cdot BCD \quad (2)$$



Rysunek 5: Układ dla drugiego bitu wyjścia

## 4.3 Bit 3

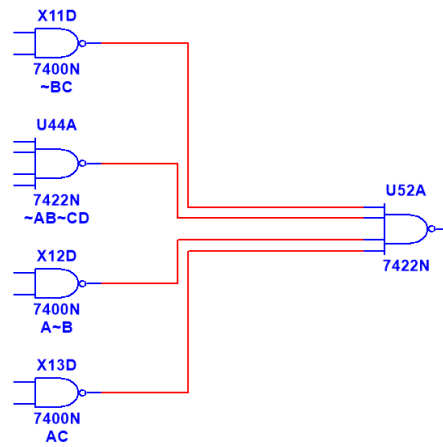
$$Y_3 = \overline{BC} \cdot \overline{ACD} \cdot ACD \quad (3)$$



Rysunek 6: Układ dla trzeciego bitu wyjścia

#### 4.4 Bit 4

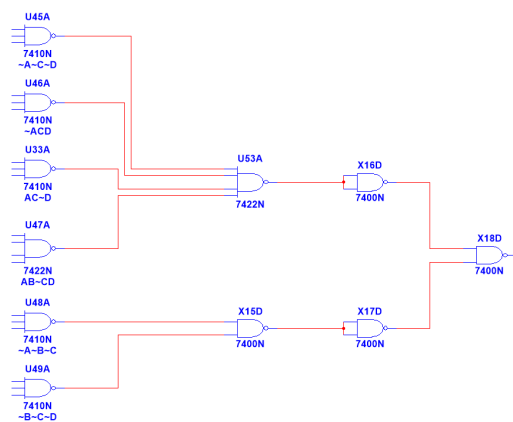
$$Y_4 = \overline{AC} \cdot \overline{BC} \cdot \overline{AB} \cdot ABCD \quad (4)$$



Rysunek 7: Układ dla czwartego bitu wyjścia

#### 4.5 Bit 5

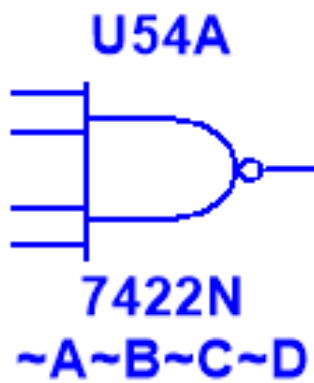
$$Y_5 = \overline{ABD} \cdot \overline{ACD} \cdot \overline{BCD} \cdot \overline{ACD} \cdot ACD \cdot ABCD \quad (5)$$



Rysunek 8: Układ dla piątego bitu wyjścia

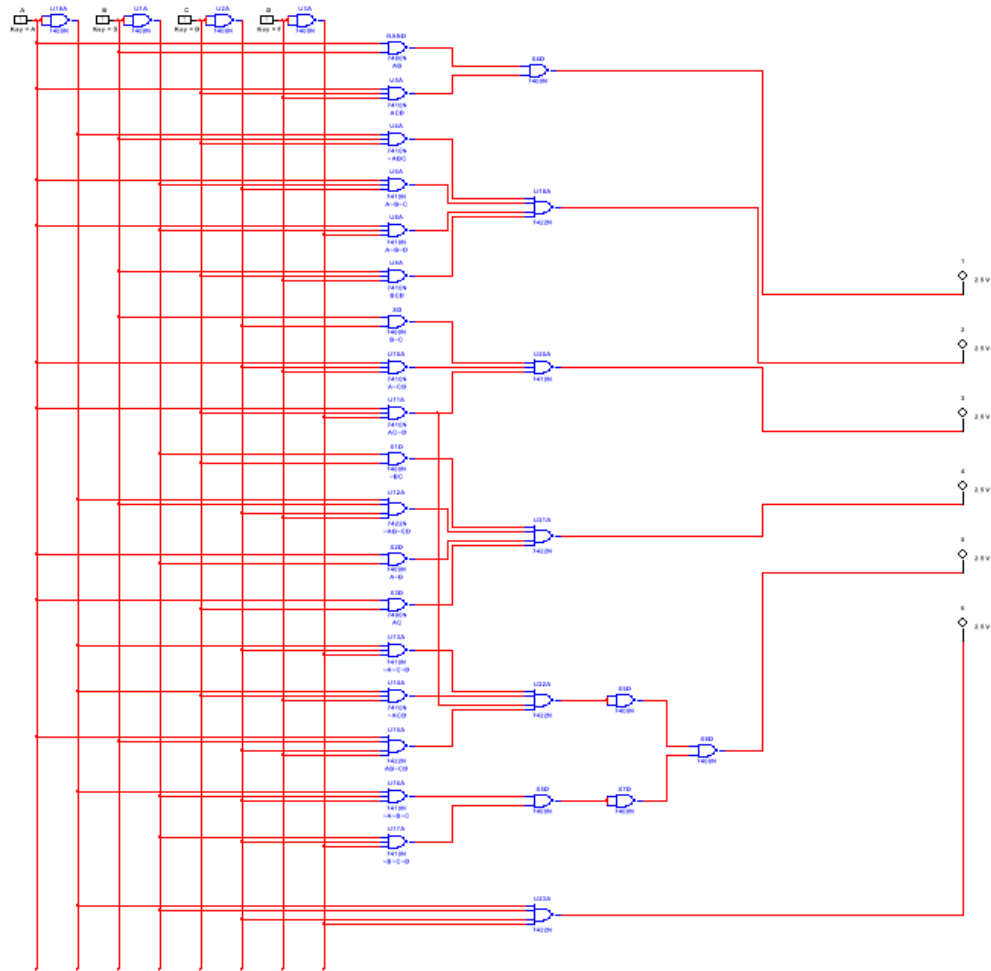
## 4.6 Bit 6

$$Y_6 = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \quad (6)$$



Rysunek 9: Układ dla szóstego bitu wyjścia

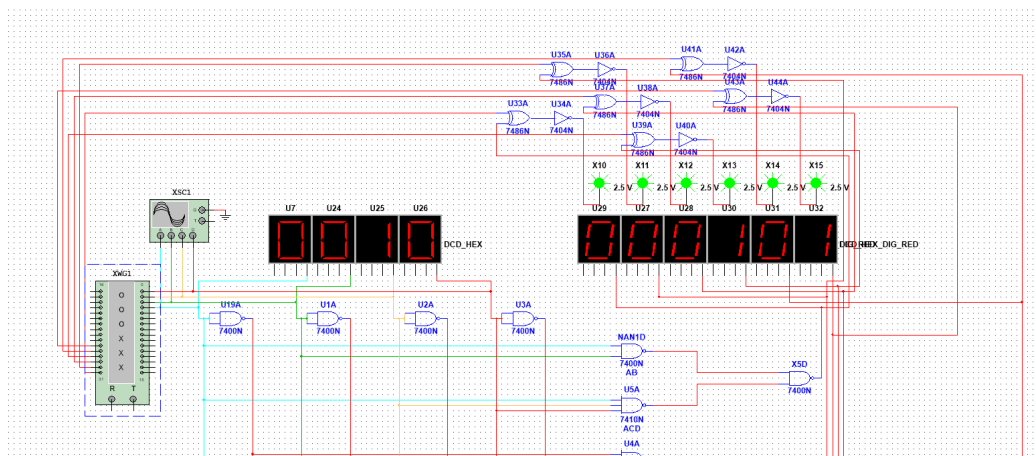
## 5 Projekt Układu



Rysunek 10: Schemat układu zrealizowany na bramkach NAND w programie Multisim



## Układ testujący



Rysunek 11: Zdjęcie układu testującego

Do przetestowania naszego układu wykorzystaliśmy generator słów binarnych, oscyloskop z czterema wejściami oraz wyświetlacze siedmiosegmentowe. Generator słów przechodzi przez wszystkie 4-bitowe słowa binarne i jednocześnie na innych sześciu pinach, oczekiwany wynik w postaci słowa 6-bitowego. 4-bitowe słowo jest wyświetlane i przekazywane do naszego układu, następnie wyjście z naszego układu jest wyświetlane i porównywane z oczekiwanym rezultatem bit po bicie. Jeżeli dany bit na wyjściu się zgadza, zapala się nad nim zielona dioda, jeżeli się nie zgadza - dioda pozostaje zgaszona. Sygnały wysyłane przez generator słów binarnych można podejrzeć na oscyloskopie.