

Projekt układu kombinacyjnego na bramkach NAND

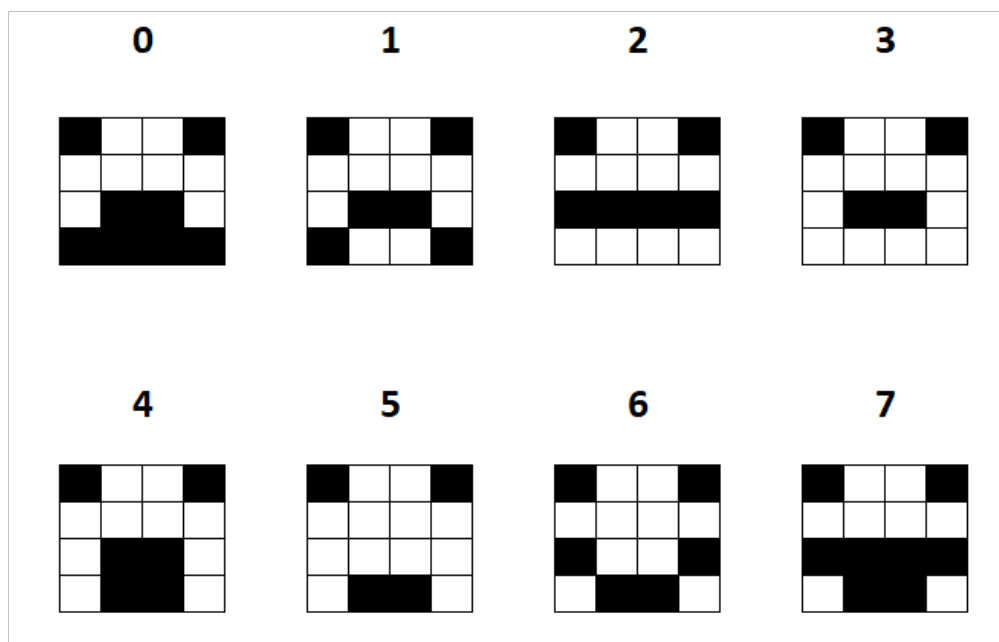
Szymon Borusiewicz, Jakub Zając, Dawid Szłapa, Radosław Szepielak

Marzec 2025

1 Wstęp

Celem niniejszego zadania jest zaprojektowanie układu kombinacyjnego, który realizuje transkoder trzybitowej liczby binarnej na graficzną reprezentację emotikon. Układ ten powinien być oparty wyłącznie na bramkach logicznych **NAND**.

Projektowany układ będzie przekształcał 3-bitowe wejście na 16-punktowy wyświetlacz, odwzorowując wzory przedstawione na poniższym rysunku:



W dalszej części opracowania zostanie przedstawiona szczegółowa analiza działania układu, jego tabela prawdy oraz sposób implementacji przy użyciu bramek **NAND**.

2 Reprezentacja bitowa matrycy LED

Każda emotikona jest wyświetlana na matrycy 4×4 , której piksele oznaczono jako p_0 do p_{15} . Oto układ tych pikseli:

p_0	p_1	p_2	p_3
p_4	p_5	p_6	p_7
p_8	p_9	p_{10}	p_{11}
p_{12}	p_{13}	p_{14}	p_{15}

3 Tabela prawdy dla transkodera

Poniższa tabela przedstawia zależność pomiędzy wejściami A, B, C a stanami pikseli p_0 do p_{15} odpowiedzialnymi za wyświetlanie emotikon na matrycy 4×4 .

C	B	A	p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}
0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	1	1	1	1
0	0	1	1	0	0	1	0	0	0	0	0	1	1	0	1	0	0	1
0	1	0	1	0	0	1	0	0	0	0	1	1	1	1	0	0	0	0
0	1	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0
1	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0	1	1	0
1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0
1	1	0	1	0	0	1	0	0	0	0	1	0	0	1	0	1	1	0
1	1	1	1	0	0	1	0	0	0	0	1	1	1	1	0	1	1	0

Tabela 1: Tabela prawdy dla transkodera 3-bitowego na matrycę LED 4×4

$$p_0 = p_3$$

$$p_1 = p_2 = p_4 = p_5 = p_6 = p_7$$

$$p_8 = p_{11}$$

$$p_9 = p_{10}$$

$$p_{12} = p_{15}$$

$$p_{13} = p_{14}$$

		BA			
		00	01	11	10
C	0	1	1	1	1
	1	1	1	1	1

Rysunek 1: Tabela Karnaugh dla p_0, p_3 funkcji logicznej 3 zmiennych (A, B, C)

$$p_0 = p_3 = 1$$

		BA			
		00	01	11	10
C	0	0	0	0	0
	1	0	0	0	0

Rysunek 2: Tabela Karnaugh dla $p_1, p_2, p_4, p_5, p_6, p_7$ funkcji logicznej 3 zmiennych (A, B, C)

$$p_1 = p_2 = p_4 = p_5 = p_6 = p_7 = 0$$

		BA			
		00	01	11	10
C	0	0	0	0	1
	1	0	0	1	1

Rysunek 3: Tabela Karnaugh dla p_8, p_{11} funkcji logicznej 3 zmiennych (A, B, C)

$$CB + B\bar{A} = \overline{\overline{CB}} \cdot \overline{\overline{B\bar{A}}}$$

		<i>BA</i>			
		00	01	11	10
<i>C</i>	0	1	1	1	1
	1	1	0	1	0

Rysunek 4: Tabela Karnaugh dla p_9, p_{10} funkcji logicznej 3 zmiennych (A, B, C)

$$\overline{C} + \overline{B} \cdot \overline{A} + BA = \overline{\overline{C} \cdot \overline{\overline{B} \cdot \overline{A} \cdot \overline{BA}}} = \overline{\overline{C} \cdot \overline{\overline{BB} \cdot \overline{AA} \cdot \overline{BA}}} = \overline{\overline{\overline{\overline{C} \cdot \overline{BB} \cdot \overline{AA} \cdot \overline{BA}}}}$$

		<i>BA</i>			
		00	01	11	10
<i>C</i>	0	1	1	0	0
	1	0	0	0	0

Rysunek 5: Tabela Karnaugh dla p_{12}, p_{15} funkcji logicznej 3 zmiennych (A, B, C)

$$\overline{C} \cdot \overline{B} = \overline{\overline{\overline{\overline{C} \cdot \overline{B}}}}$$

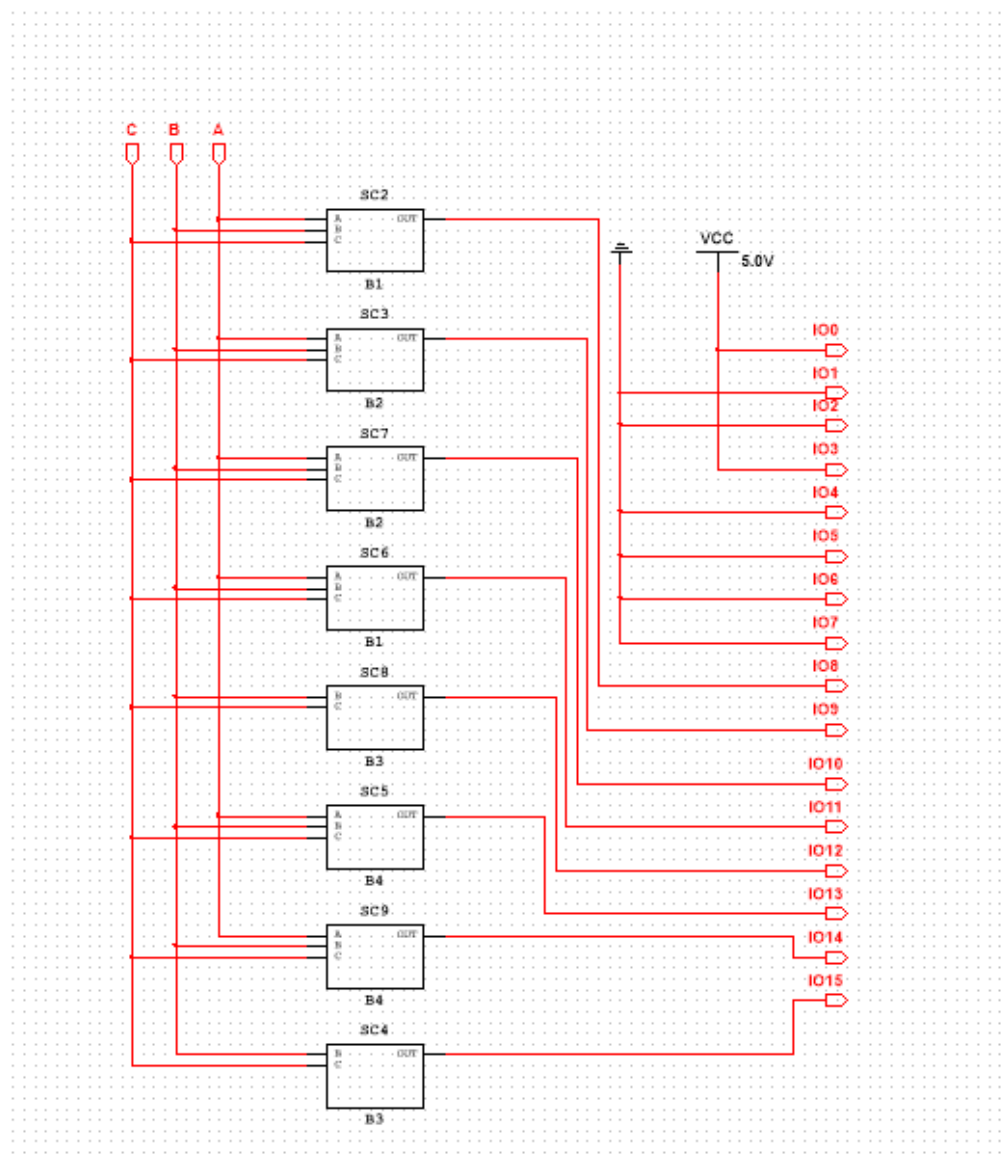
		<i>BA</i>			
		00	01	11	10
<i>C</i>	0	1	0	0	0
	1	1	1	1	1

Rysunek 6: Tabela Karnaugh dla p_{13} , p_{14} funkcji logicznej 3 zmiennych (A , B , C)

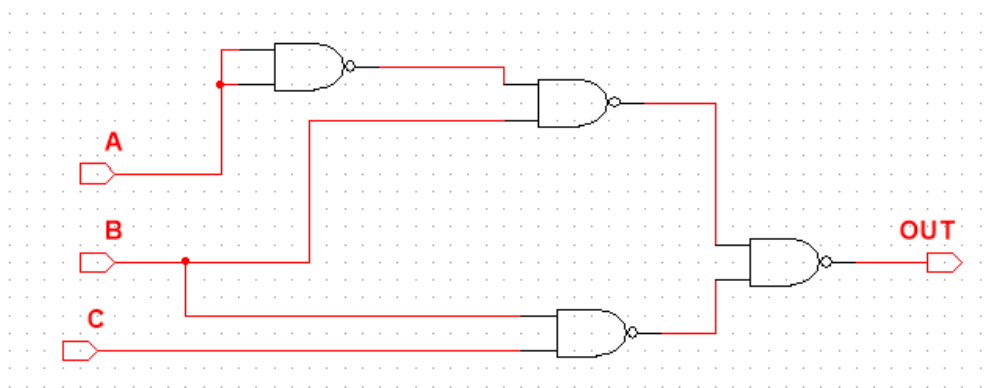
$$C + \overline{B} \cdot \overline{A} = \overline{\overline{C}C} \cdot \overline{\overline{B}B} \cdot \overline{\overline{A}A} = \overline{\overline{C}C} \cdot \overline{\overline{B}B} \cdot \overline{\overline{A}A}$$

4 Implementacja układów

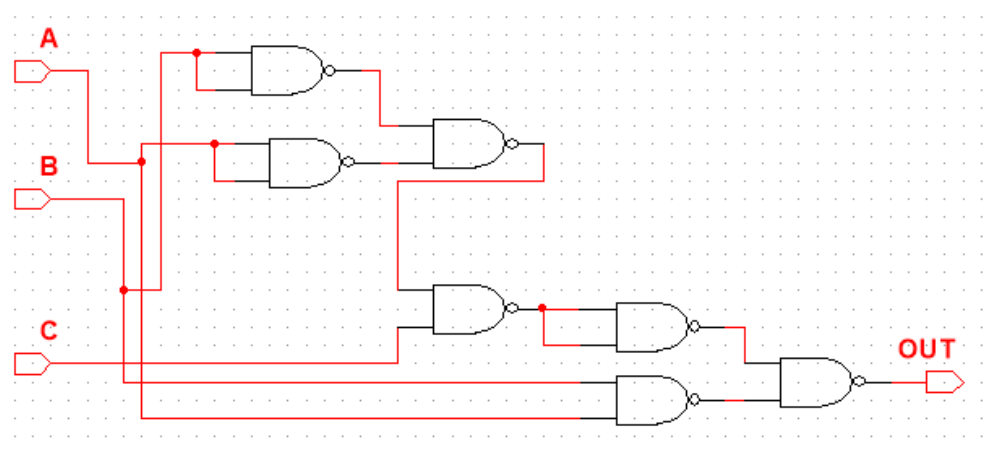
4.1 Układ transkodera



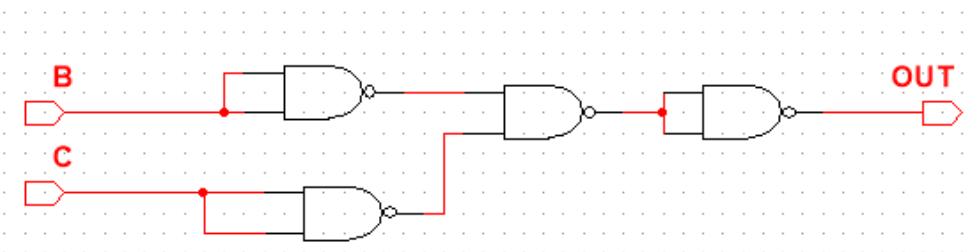
Rysunek 7: Układ transkodera



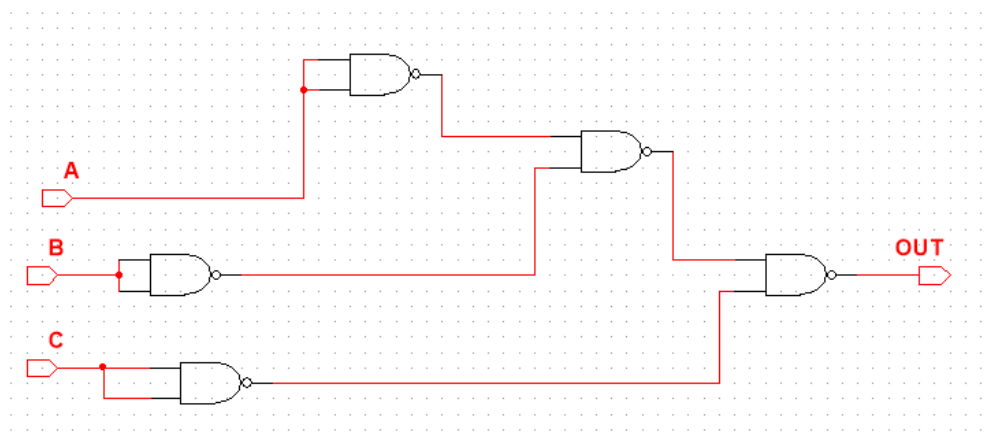
Rysunek 8: Podukład B1



Rysunek 9: Podukład B2



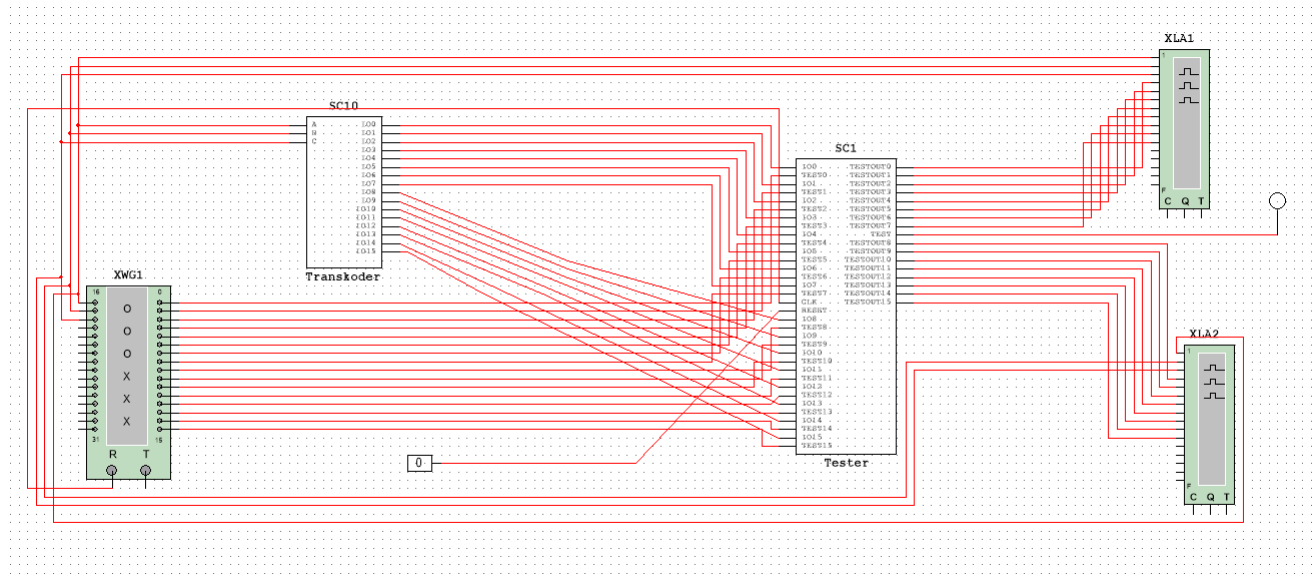
Rysunek 10: Podukład B3



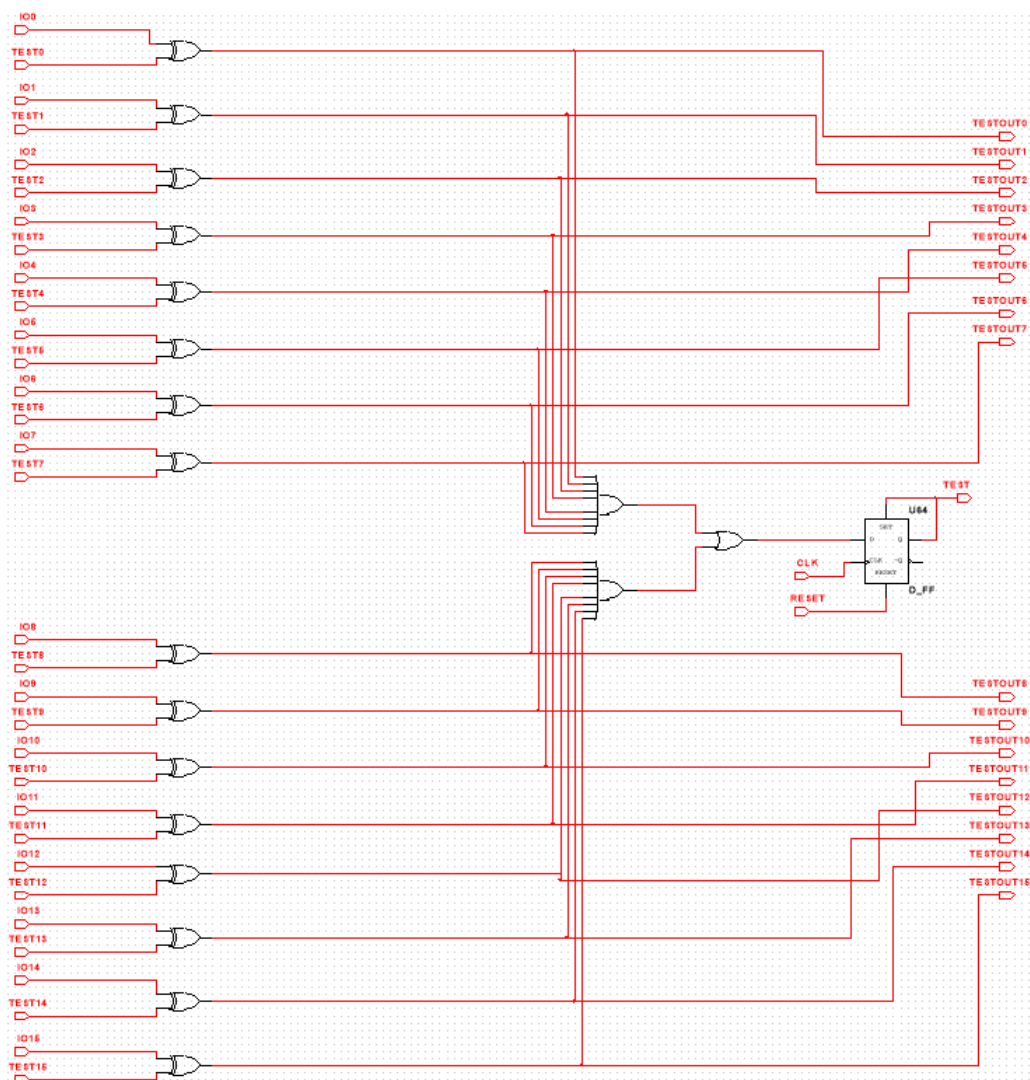
Rysunek 11: Podukład B4

4.2 Układ testujący

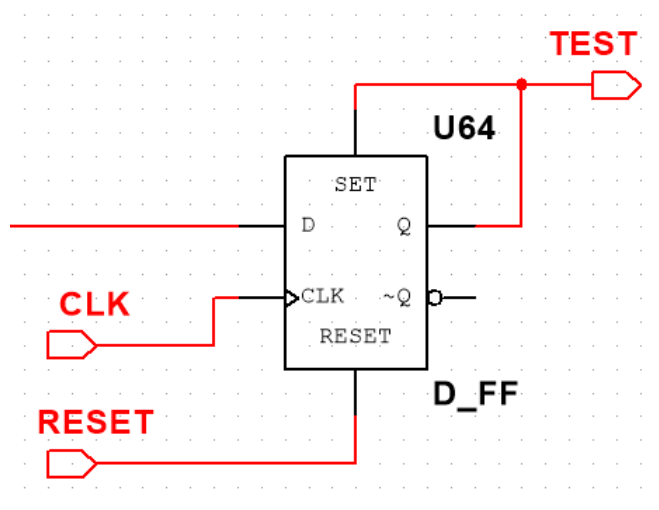
Układ testujący został wykonany przy użyciu przerzutnika typu D i generatora słów. Tester porównuje wyjścia transkodera z wyjściem generatora słów. W przypadku, gdy wyjścia różnią się, zapala się dioda informująca o awarii układu, po czym świeci się do momentu podania stanu wysokiego na wejście RESET. Wyjścia testera TESTOUT podają stan wysoki w przypadku, gdy odpowiednie wejścia IO i TEST różnią się stanem. Użyliśmy dwóch analizatorów stanów logicznych do przedstawienia na każdym z nich bitów wejściowych i bitów wyjścia testera ze względu na ograniczoną liczbę pinów.



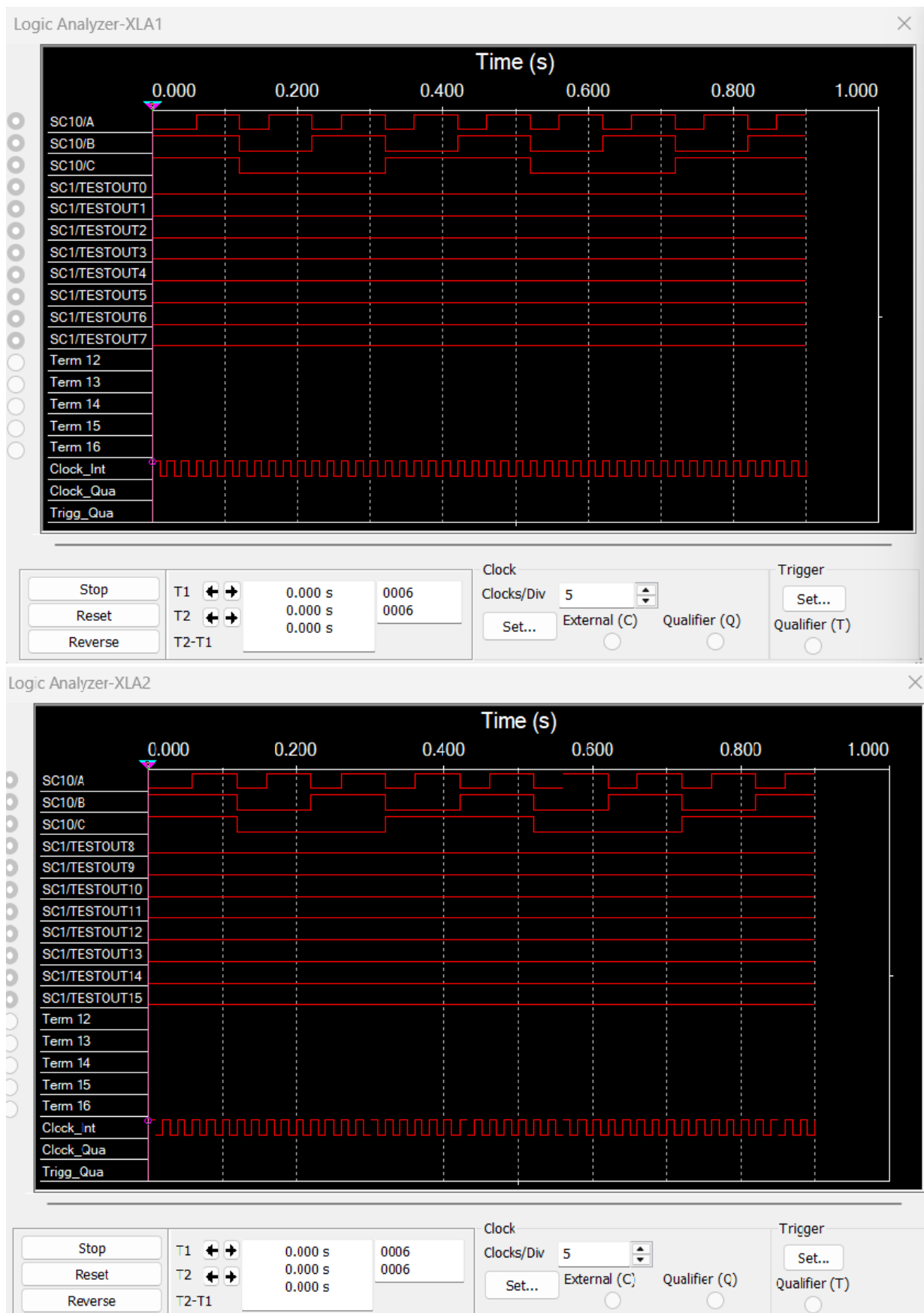
Rysunek 12: Układ testujący



Rysunek 13: Układ tester

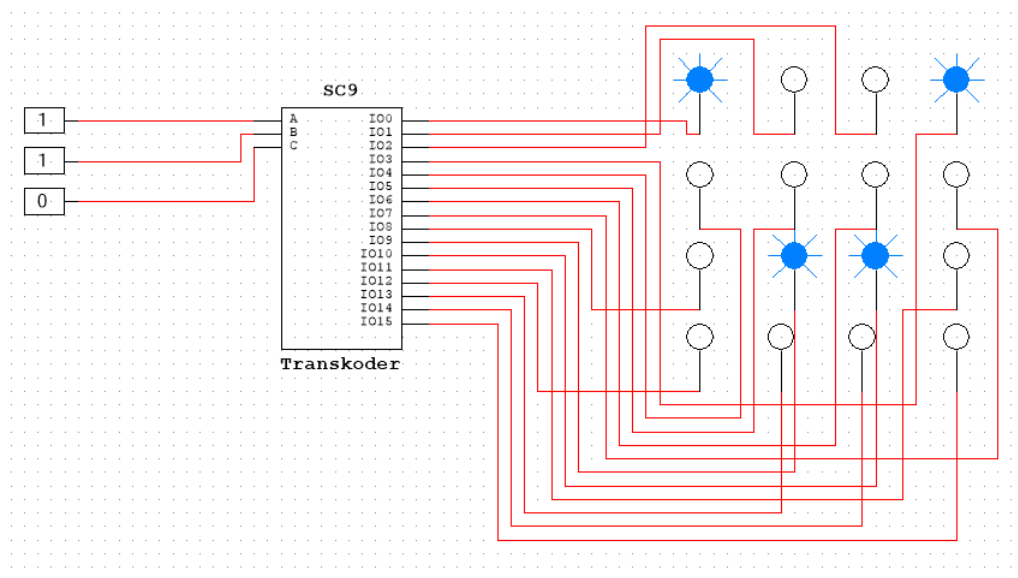


Rysunek 14: Przerzutnik typu D



Rysunek 15: Wskazania analizatora XLA1 oraz XLA2

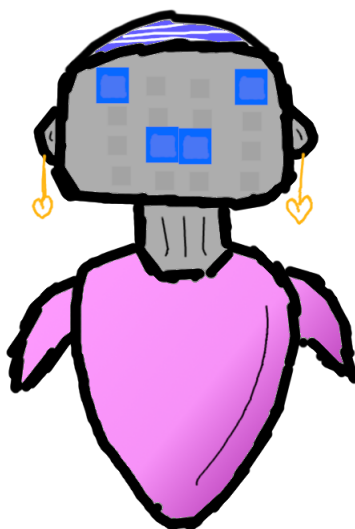
Wskazania analizatorów potwierdzają poprawność działania zaprojektowanego przez nas transkodera



Rysunek 16: Tester

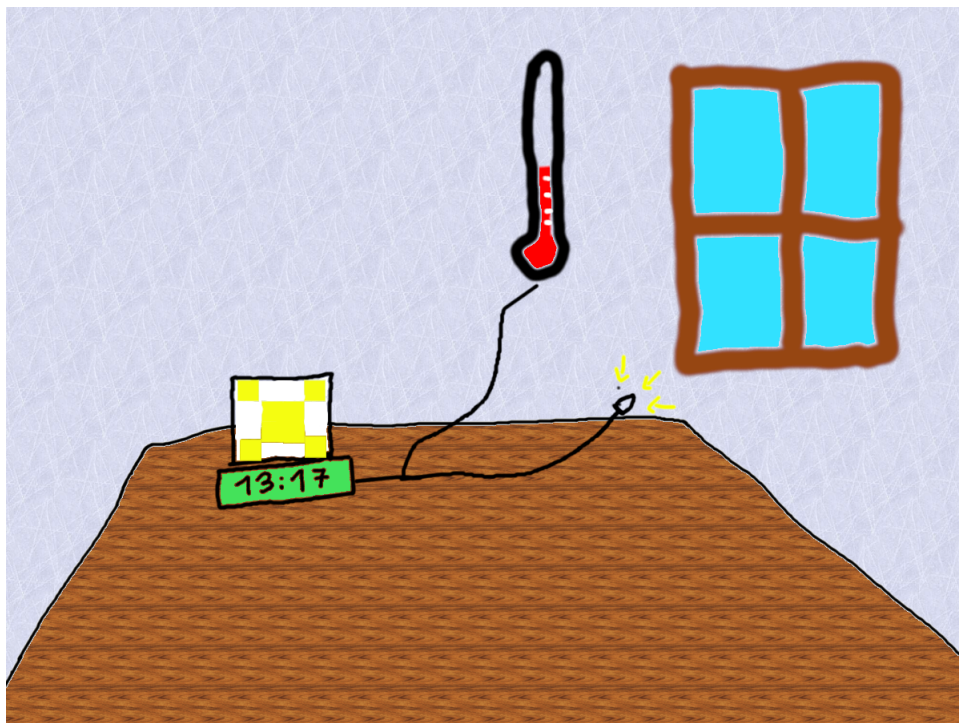
5 Zastosowania

- Proste wyświetlacze emocji:
Taki układ może być użyty w zabawkach lub gadżetach, które wyświetlają emocje (np. uśmiech, smutek) na matrycy LED w zależności od ustawienia przełączników lub sygnałów wejściowych.



Rysunek 17: Przykład wykorzystania wyświetlacza w robocie zabawce

- Pogoda na zewnątrz:
Mając 3 czujniki: wilgotności, fotodiode mierząc natężenie światła oraz termometr, możemy na wyświetlaczu przedstawiać różne emotki odpowiadające aktualnej pogodzie na zewnątrz.



Rysunek 18: Wyświetlanie aktualnej pogody

6 Wnioski

Zastosowanie w ten sposób bramek NAND potwierdza że są one uniwersalnym elementem logicznym. Za ich pomocą przedstawiliśmy wszystkie niezbędne operatory logiczne, co pomogło zrealizować złożoną logikę wyświetlacza emotikon.