Transkoder trzybitowej liczby binarnej na układ graficzny emotikony

Bartosz Kaganiec, Julia Demitraszek, Adrian Krawczyk, Damian Chłus

14 kwietnia 2025

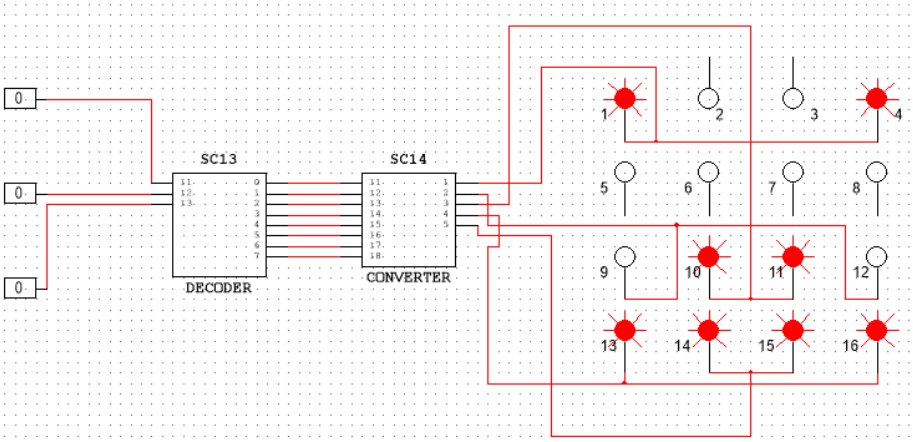
Spis treści

1 Opis zadania

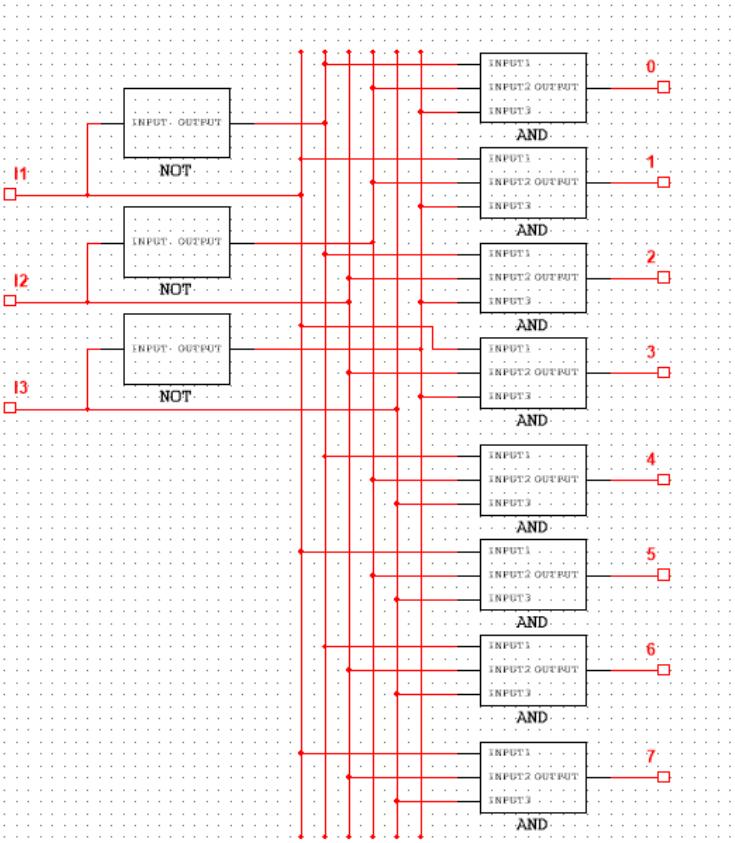
Bazując wyłącznie na bramkach NAND należy zaprojektować układ kombinacyjny realizujący transkoder trzybitowej liczby binarnej na układ graficzny emotikony wyświetlanej na szesnastu punktach.

2 Pierwsze podejście

Na początku nasze sygnały przechodzą przez dekoder, który dla każdej trójki przyporządkowywuje liczbę od 0 do 7 (binarnie wszystkie możliwe układy 3 bitów). Tutaj I1 jest bitem najmniej znaczącym, a I3 bitem najbardziej znaczącym.



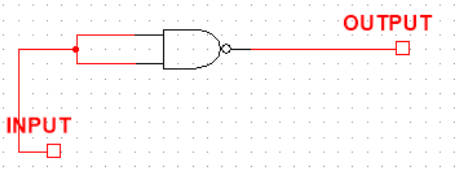
Rysunek 1: Zdjęcie całego układu



Rysunek 2: Schemat dekodera

Wszystkie bramki były wykonane za pomocą bramek NAND:
Wyprowadzenie bramki NOT:

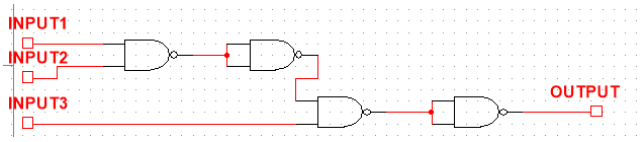
$$\overline{A} = \overline{A \cdot A} \tag{1}$$



Rysunek 3: Bramka NOT

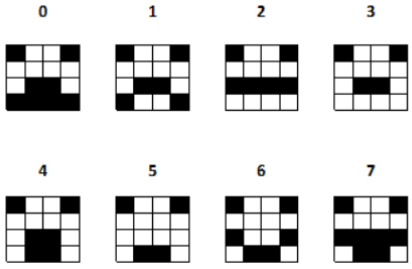
Wyprowadzenie bramki AND:

$$A \cdot B \cdot C = \overline{\overline{(AB)} \cdot C} = \overline{\overline{\overline{(AB)} \cdot C}} \tag{2}$$

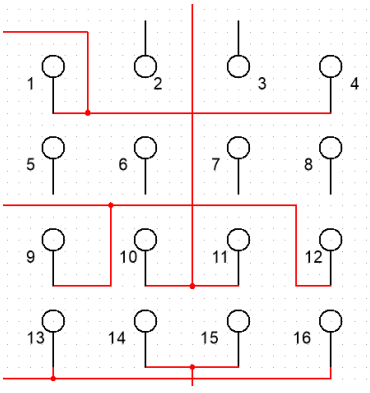


Rysunek 4: Bramka AND

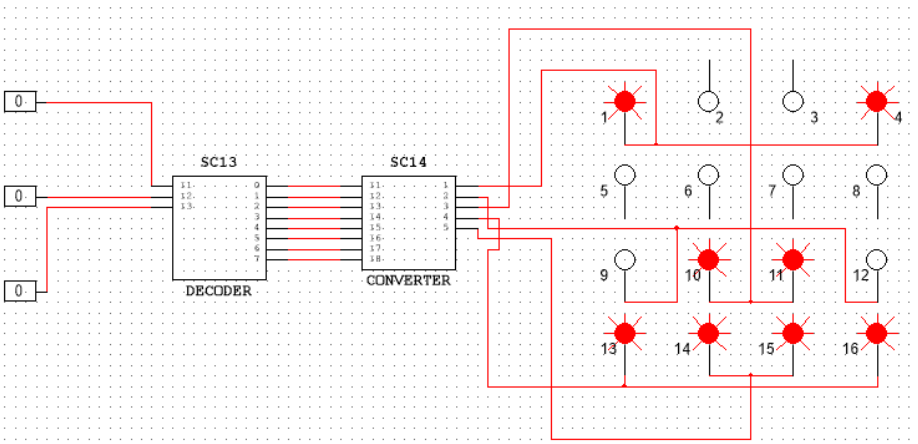
Możliwe emotikony podane w zadaniu:



Rysunek 5: Emotikony



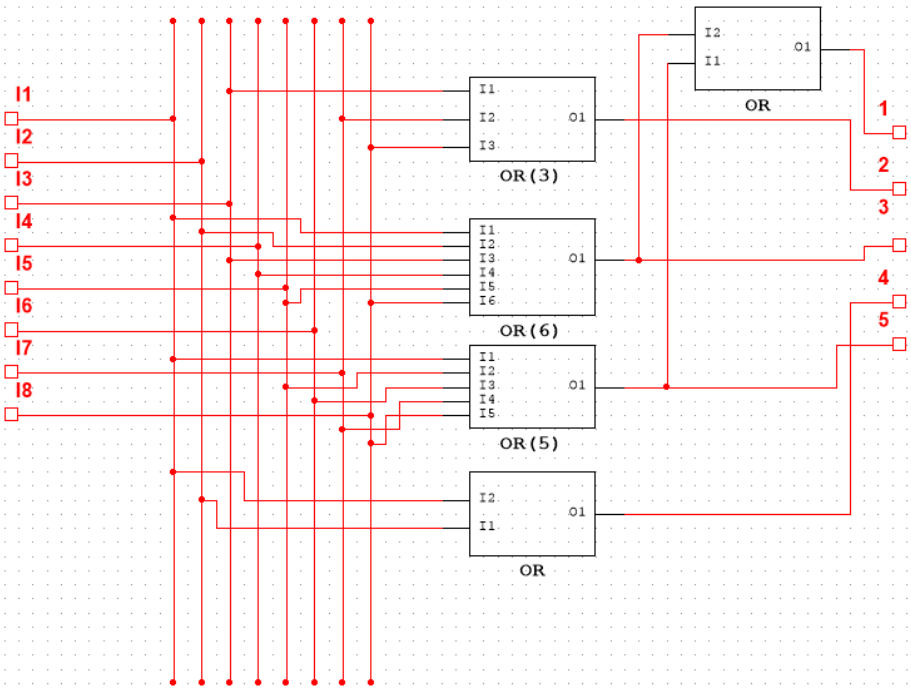
Rysunek 6: Nasza numeracja lampek i układ



Rysunek 7: Zdjęcie całego układu

Następnie przechodzą przez konwerter, który do każdego możliwego sygnału dopasowywuje możliwe wyjścia. Możliwości wyjść:

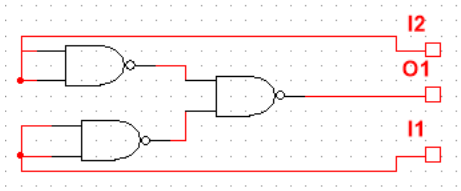
- lampki 1 i 4 są zapalone (wyjście 1)
- lampki 2, 3, 5, 6, 7, 8, są zawsze zgaszone
- dla emotikon 2, 6, 7 zapalone lampki 9 i 12 (wyjście 2)
- dla emotikon 0, 1, 2, 3, 4, 7 są zapalone lampki 10 i 11 (wyjście 3)
- dla emotikon 0, 1 są zapalone lampki 13 i 16 (wyjście 4)
- dla emotikon 0, 4, 5, 6, 7 są zapalone lampki 14 i 15 (wyjście 5)



Rysunek 8: Układ konwertera

Wyprowadzenie bramki OR za pomocą bramek NAND:

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

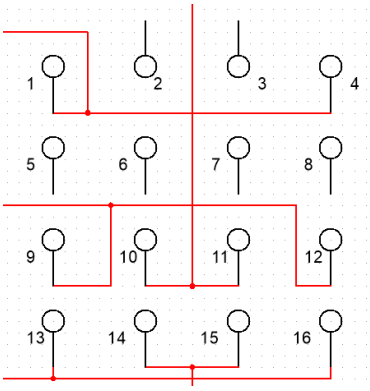


Rysunek 9: Bramka OR

3 Ogólna idea rozwiązania

Na wejściu przyjmujemy liczbę binarną zapisaną na 3 bitach, którą za pomocą odpowiedniego połączenia bramek logicznych NAND będziemy przekształcać na odpowiednie stany próbników. Każda z przyjmowanych liczb będzie reprezentowała zadany w poleceniu 16-punktowy emotikon.

Funkcje logiczne zawierające operacje NAND wyprowadzimy korzystając z funkcji powstałych po odpowiednim zgrupowaniu elementów w tablicach Karnaughta. W celu sprawdzenia poprawności działania układu zbudujemy w programie Multisim stanowisko pozwalające na automatyczne przeprowadzenie testów.



Rysunek 10: Nasza numeracja lampek i układ

Tabela prawdy:

A	B	C	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16
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

4 Tabele Karnaugh i uproszczenie formuł

Kolorami oznaczono grupowane elementy w celu uproszczenia formuł. Kolor żółty, czerwony i zielony oznaczają osobne grupy elementów, natomiast kolor pomarańczowy oznacza, że dany element należy do jakichś obu grup. Pod każdą tabelą umieszczono również wzór formuły z niej wynikający. Skorzystaliśmy z metody z grupowaniem "jedynek", upraszczając funkcje logiczne do alternatywy koniunkcji sygnałów (postać DNF formuły logicznej).

4.1 Wyjścia o stałej wartości

Dla poniższych wyjść, analiza tablic Karnaughta wskazuje na stałą wartość:

- $B_1 = B_4 = 1$ (każda komórka tabeli Karnaughta zawiera 1)

- $B_2 = B_3 = B_5 = B_6 = B_7 = B_8 = 0$ (każda komórka tabeli Karnaughta zawiera 0)

4.2 Pozostałe wyjścia

BC\A	0	1
00	0	0
01	0	0
11	0	1
10	1	1

$$B_9 = B\overline{C} + AB$$

BC\A	0	1
00	1	1
01	1	0
11	1	1
10	1	0

$$B_{10} = \overline{A} + BC + \overline{B} \cdot \overline{C}$$

BC\A	0	1
00	1	0
01	1	0
11	0	0
10	0	0

$$B_{13} = \overline{A} \cdot \overline{B}$$

BC\A	0	1
00	1	1
01	0	1
11	0	1
10	0	1

$$B_{14} = A + \overline{B} \cdot \overline{C}$$

4.3 Symetryczne wyjścia B_9 i B_{12}

Analiza tablic Karnaughta dla wyjść B_9 i B_{12} wykazała identyczny rozkład wartości, co oznacza, że realizują one tę samą funkcję logiczną:

$$B_9 = B_{12} = B\overline{C} + AB$$

	A=0	A=1
BC=00	0	0
BC=01	0	0
BC=11	0	1
BC=10	1	1

$$B_{12} = B\overline{C} + AB$$

4.4 Symetryczne wyjścia B_{10} i B_{11}

Analiza tablic Karnaughta dla wyjść B_{10} i B_{11} wykazała identyczny rozkład wartości, co oznacza, że realizują one tę samą funkcję logiczną:

$$B_{10} = B_{11} = \overline{A} + BC + \overline{B} \cdot \overline{C}$$

	A=0	A=1
BC=00	1	1
BC=01	1	0
BC=11	1	1
BC=10	1	0

$$B_{11} = \overline{A} + BC + \overline{B} \cdot \overline{C}$$

4.5 Symetryczne wyjścia B_{13} i B_{16}

Analiza tablic Karnaughta dla wyjść B_{13} i B_{16} wykazała identyczny rozkład wartości, co oznacza, że realizują one tę samą funkcję logiczną:

$$B_{13} = B_{16} = \overline{A} \cdot \overline{B}$$

	A=0	A=1
BC=00	1	0
BC=01	1	0
BC=11	0	0
BC=10	0	0

$$B_{16} = \overline{A} \cdot \overline{B}$$

4.6 Symetryczne wyjścia B_{14} i B_{15}

Analiza tablic Karnaughta dla wyjść B_{14} i B_{15} wykazała identyczny rozkład wartości, co oznacza, że realizują one tę samą funkcję logiczną:

$$B_{14} = B_{15} = A + \overline{B} \cdot \overline{C}$$

	A=0	A=1
BC=00	1	1
BC=01	0	1
BC=11	0	1
BC=10	0	1

$$B_{15} = A + \overline{B} \cdot \overline{C}$$

5 Wyprowadzenie wzorów wraz ze schematami układu

Symetria towarzysząca niektórym wyjściom pozwala na użycie jednego modułu implementującego tę funkcję dla obu wyjść, co upraszcza projekt i zmniejsza liczbę wymaganych bramek. Korzystając z powyższych tabel Karnaughta, grupujemy sąsiednie elementy (różniące się dokładnie 1 bitem) tworzące prostokąty wielkości 2^n, i stosujemy poniższe zależności, aby otrzymać wyrażenia złożone tylko z operacji NAND.

1. $A = A \cdot A$ (idempotentność)
2. $\overline{A + B} = \overline{A} \cdot \overline{B}$ (pierwsze prawo de Morgana)
3. $\overline{A \cdot B} = \overline{A} + \overline{B}$ (drugie prawo de Morgana)
4. $\overline{\overline{A}} = A$ (prawo podwójnej negacji)
5. $\overline{A \cdot B} = A|B$ (A NAND B) (definicja operacji NAND)

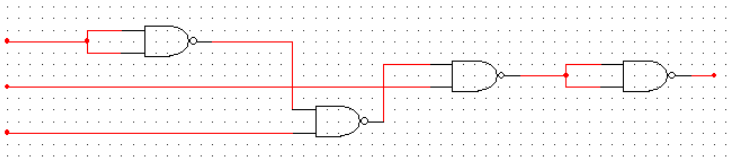
Poniżej przedstawiamy wyprowadzenie wybranych funkcji wyjściowych do postaci używającej tylko operacji NAND:
$B_1 = B_4 = 1$ $B_2 = B_3 = B_5 = B_6 = B_7 = B_8 = 0$
$B_9 = B_{12}$: $B\overline{C} + AB$

BC\A	0	1
00	0	0
01	0	0
11	0	1
10	1	1

$$B_9 = B\overline{C} + AB$$

$$\begin{aligned} B_9 = B_{12} = B\overline{C} + AB &= B(\overline{C} + A) \\ &= B\overline{(C\overline{A})} \\ &= \overline{\overline{B(C\overline{A})}} \end{aligned}$$

$$B_9 = B_{12} = \overline{\overline{B(C\overline{A})}}$$



Rysunek 11: Schemat wyjść B_9 i B_{12}

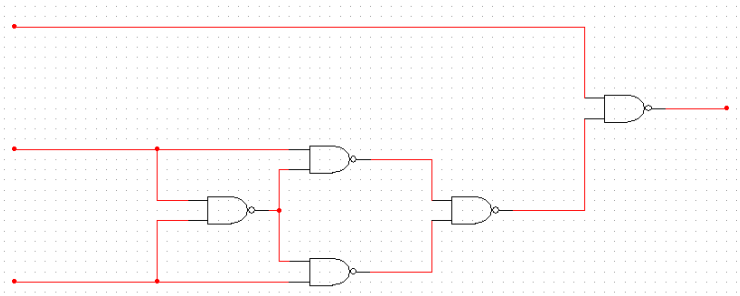
$B_{10} = B_{11}$: $\overline{A} + BC + \overline{B} \cdot \overline{C}$

BC\A	0	1
00	1	1
01	1	0
11	1	1
10	1	0

$$B_{10} = \overline{A} + BC + \overline{B} \cdot \overline{C}$$

$$\begin{aligned} B_{10} = B_{11} = \overline{A} + BC + \overline{B} \cdot \overline{C} &= \overline{A} + \overline{B \oplus C} \\ &= \overline{A \cdot (B \oplus C)} \end{aligned}$$

$$B_{10} = B_{11} = \overline{A \cdot (B \oplus C)}$$



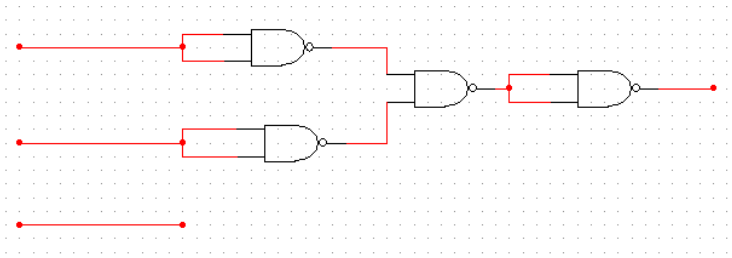
Rysunek 12: Schemat wyjść B_{10} i B_{11}

$B_{13} = B_{16}$: $\overline{A} \cdot \overline{B}$

BC\A	0	1
00	1	0
01	1	0
11	0	0
10	0	0

$$B_{13} = \overline{A} \cdot \overline{B}$$

$$B_{13} = B_{16} = \overline{A} \cdot \overline{B} = \overline{\overline{A} \cdot \overline{B}}$$



Rysunek 13: Schemat wyjść B_{13} i B_{16}

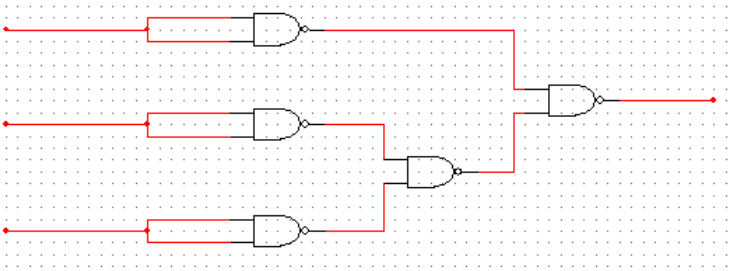
BC\A	0	1
00	1	1
01	0	1
11	0	1
10	0	1

$$B_{14} = A + \overline{B} \cdot \overline{C}$$

$B_{14} = B_{15}$: $A + \overline{B} \cdot \overline{C}$

$$B_{14} = B_{15} = A + \overline{B} \cdot \overline{C} = A + \overline{\overline{\overline{B} \cdot \overline{C}}}$$
$$= \overline{\overline{A} \cdot \overline{(\overline{B} \cdot \overline{C})}}$$

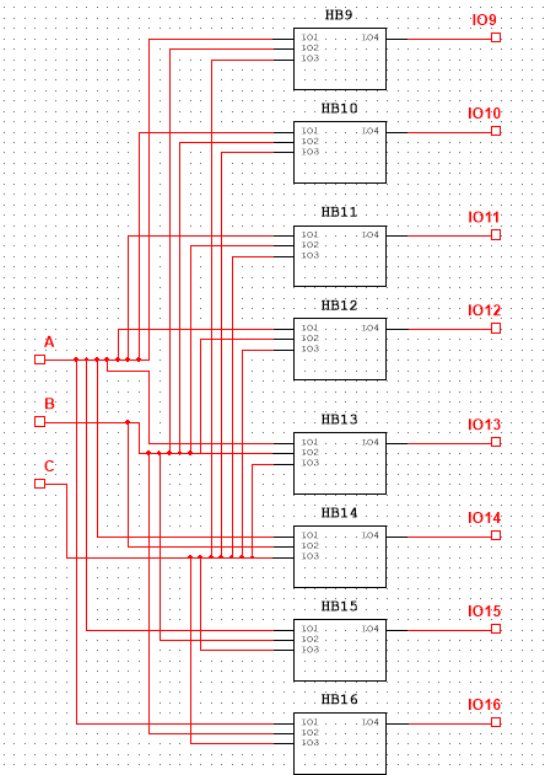
$$B_{14} = B_{15} = \overline{\overline{A} \cdot \overline{(\overline{B} \cdot \overline{C})}}$$



Rysunek 14: Schemat wyjść B_{14} i B_{15}

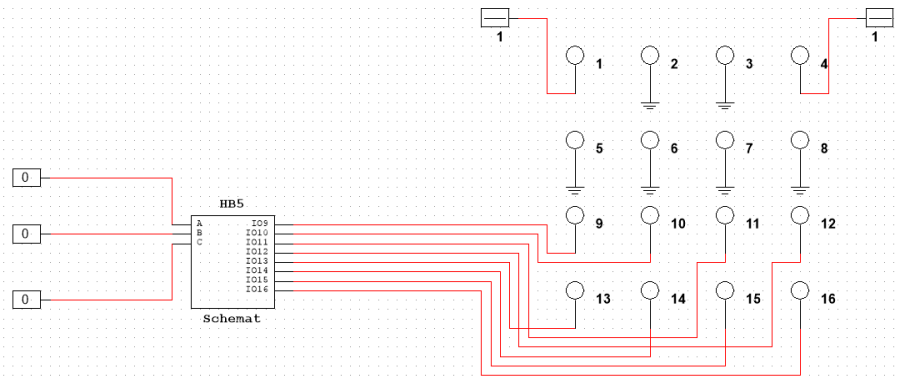
6 Początkowy schemat układu

Układ składa się z trzech wejść cyfrowych (A, B, C), podukładu realizującego przedstawione zadanie, oraz matrycy 16 próbników, które będą się świecić gdy wyjście, do którego są podłączone będzie w stanie wysokim.



Rysunek 15: Pokazane wejścia (A, B, C) podłączone do poszczególnych podukładów

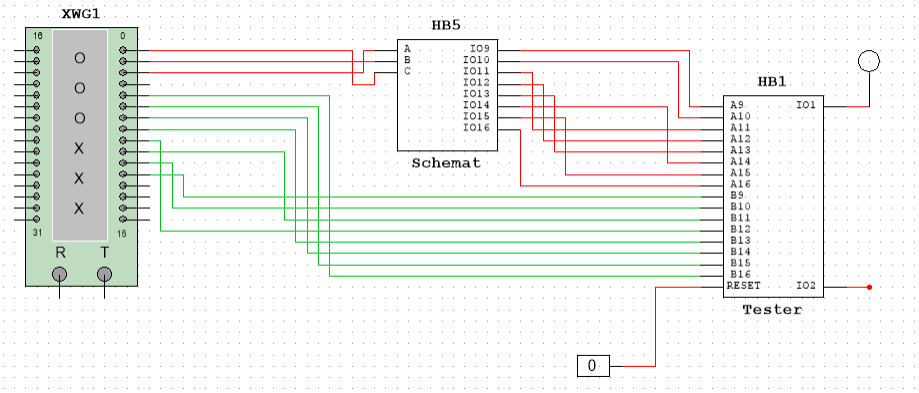
Po opakowaniu całego podukładu w blok hierarchiczny i po podłączeniu do lampek schemat prezentuje się w następujący sposób:



Rysunek 16: Pokazane wejścia (A, B, C) podłączone do schematu, który następnie jest podłączony do lampek

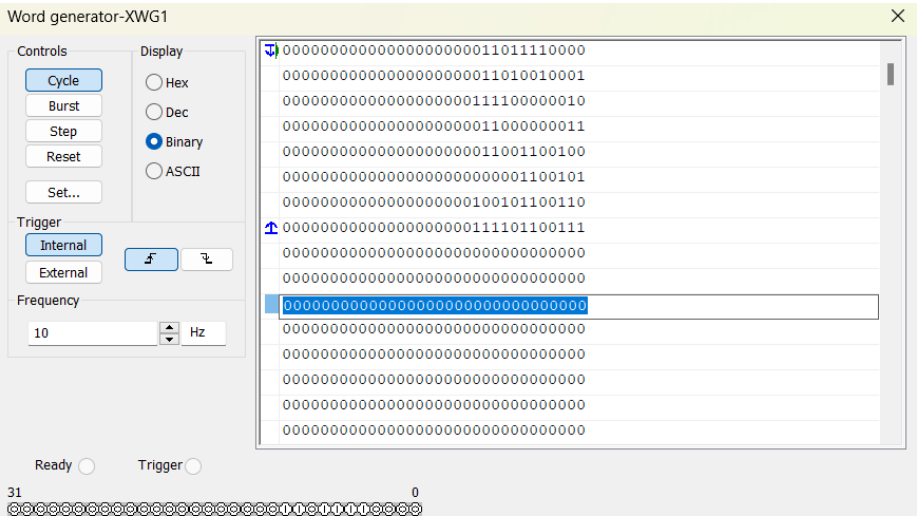
7 Testowanie

W celu przetestowania poprawności działania całego układu zaprojektowaliśmy i zaimplementowaliśmy stanowisko testowe, gdzie generator słów wyznacza poprawne sygnały wyjścia naszego głównego podukładu. Te sygnały są przekazywane do komponentu Testera, którego zadaniem jest weryfikacja poprawności wyjść podukładu.



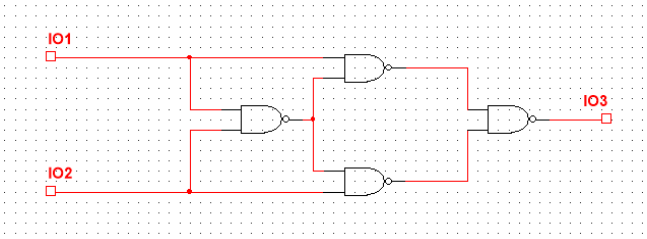
Rysunek 17: Układ testowy

Sekwencje w generatorze słów powstały w taki sposób, że na początku jest 8 wartości oczekiwanych (wartości z bramek B9-B16), następnie jest przerwa (0) i potem wejścia odpowiadające oczekiwanym wyjściom (wartości A, B, C)



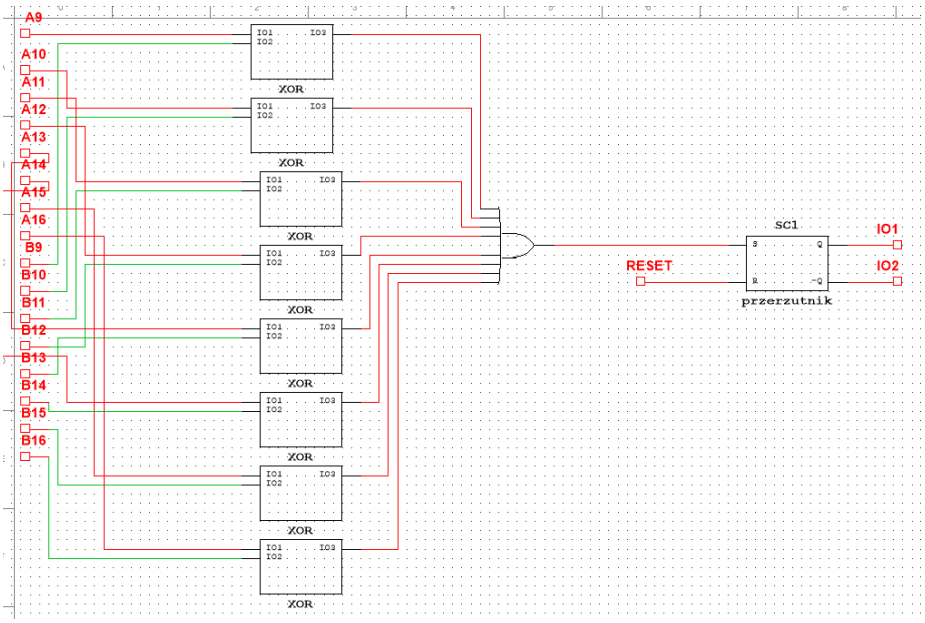
Rysunek 18: Zdjęcie z generatora słów

Zadanie testowania realizowane jest przez 8 bramek XOR, gdzie jednym wejściem każdej bramki jest sygnał wyjściowy podukładu, a drugim jest oczekiwany poprawny sygnał z generatora słów. Po osobnym sprawdzeniu każdego sygnału, wyniki są przekazywane do bramki OR, która przy wykryciu jakiejkolwiek niezgodności, przekazuje sygnał błędu do przerzutnika RS.



Rysunek 19: Zdjęcie podukładu XOR z wykorzystaniem bramek NAND

Przerzutnik RS, dzięki połączeniu wyjścia Q i wejścia SET, zasygnalizuje błąd za pomocą czerwonego próbnika, ma też wejście R, które oznacza zresetowanie przerzutnika, jest ono regulowane sygnałem 0 - brak resetu, 1 - reset.



Rysunek 20: Zdjęcie układu testowego

Przerzutnik został zbudowany na bazie tej tabeli prawdy

S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	1	0
1	1	X	X

Tabele Karnaugh

SR\Q	0	1
00	0	1
01	0	0
11	X	X
10	1	1

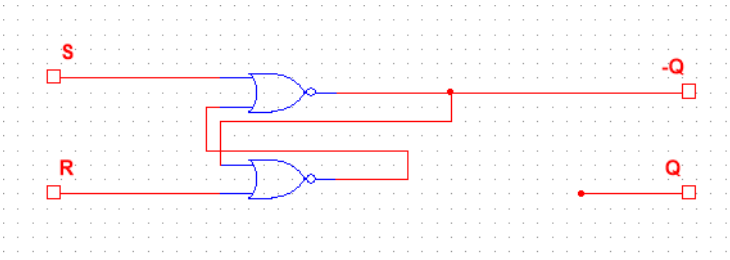
Korzystniejsze ze względu na analizę takiego układu będzie założenie że w miejsce x pojawią się 1 i wobec tego otrzymamy:

SR\Q	0	1
00	0	1
01	0	0
11	1	1
10	1	1

$$Q = S + Q\overline{R}$$

Funkcja przekształcona na funkcję w postaci NOR, przekształcamy równoważnie

$$\overline{Q} = \overline{S + Q\overline{R}} = \overline{S + \overline{\overline{Q} + R}}$$



Rysunek 21: Zdjęcie układu tworzącego wyjście $\overline{Q}$

Tabela Karnaugh dla NOT Q:

SR\$\overline{Q}$	0	1
00	0	1
01	1	1
11	X	X
10	0	0

Tu analogicznie wygodniej jest przyjąć, że w miejsce x wystąpi 1

SR\$\overline{Q}$	0	1
00	0	1
01	1	1
11	1	1
10	0	0

$$\overline{Q} = R + \overline{S} \cdot \overline{Q}$$

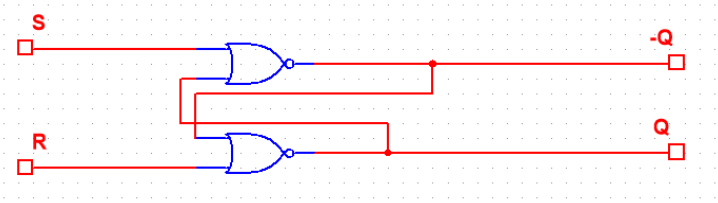
Funkcja przekształcona na funkcję w postaci NOR, przekształcamy równoważnie, z równania:

$$\overline{Q} = R + \overline{S} \cdot \overline{Q}$$

Otrzymujemy:

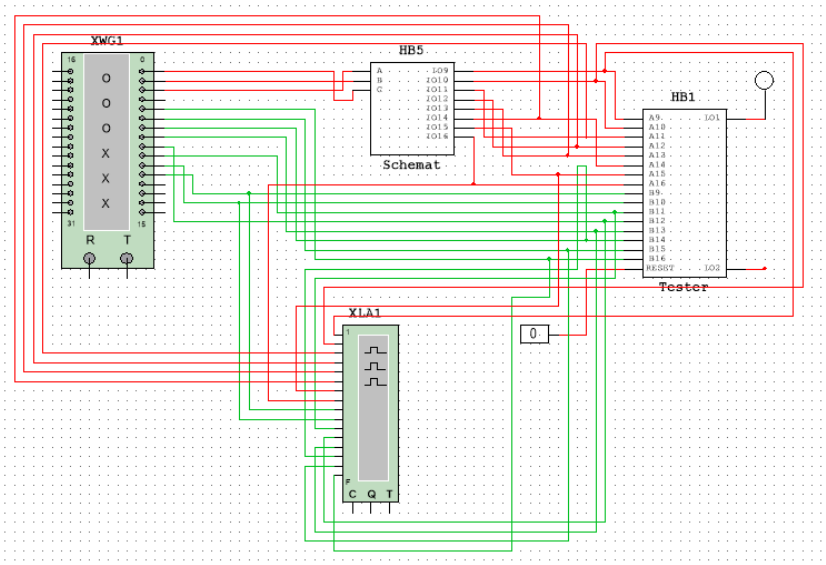
$$Q = \overline{R + \overline{S} \cdot \overline{Q}} = \overline{R + \overline{S + Q}}$$

Po zauważeniu możliwości połączenia obydwu układów w jeden otrzymujemy:

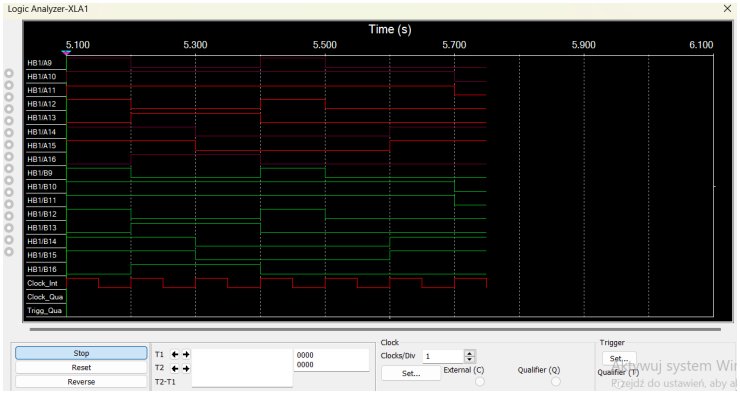


Rysunek 22: Zdjęcie układu tworzącego wyjście Q oraz $\overline{Q}$ a tym samym schemat całego przerzutnika

Schemat z podpiętym analizatorem logicznym:



Rysunek 23: Zdjęcie schematu z analizatorem logicznym



Rysunek 24: Zdjęcie z analizatora logicznego

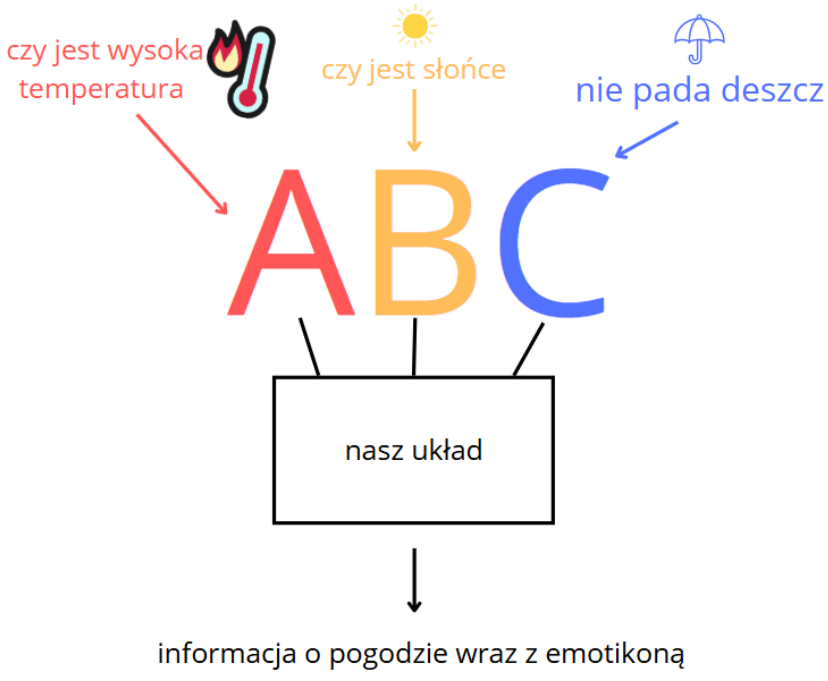
Testom podlegają tylko bramki zmieniające swoje sygnały, ponieważ sygnały od bramek od 1-8 są stale równe 0 lub 1. Jak wydać na powyższym zdjęciu nasz układ działa poprawnie.

8 Zastosowania

Zaprojektowany układ znajduje zastosowanie w systemach, gdzie poszczególne sygnały oznaczają obecność lub brak jakiejś wartości przy jednoczesnym dopasowaniu do sytuacji adekwatnej i czytelnej emotikony.

Dla przykładu nasz układ może reprezentować stany pogody, to znaczy, jeżeli mamy sygnały ABC

- 000 – zimna temperatura, brak słońca, brak deszczu (niezbyt przyjemna pogoda)
- 001 – zimna temperatura, brak słońca, pada deszcz (pogoda jeszcze gorsza)
- 010 – zimna temperatura, jest słońce, brak deszczu (pogoda przyjemna, słoneczna)
- 011 – zimna temperatura, jest słońce, pada deszcz (jest słońce ale pada deszcz)
- 100 – wysoka temperatura, brak słońca, brak deszczu (gorąco, duszno)
- 101 – wysoka temperatura, brak słońca, pada deszcz (parny dzień, burzowo)
- 110 – wysoka temperatura, jest słońce, brak deszczu (upalny dzień)
- 111 – wysoka temperatura, jest słońce, pada deszcz (tropikalne warunki, burza lub wilgotny skwar)



Rysunek 25: Zdjęcie schematu z analizatorem logicznym

Innym przykładem zastosowania może być stan naszej baterii w telefonie tzn poszczególne sygnały mogą oznaczać:

- A – poziom baterii niski
- B – ładowarka podłączona
- C – urządzenie aktywnie używane

Nadane znaczenia:

- 000 - Bateria ok, nie ładuje, brak użycia
- 001 - Bateria ok, nie ładuje, używane
- 010 - Bateria ok, ładuje się, brak użycia
- 011 - Bateria ok, ładuje się, używane
- 100 - Bateria niska, nie ładuje, brak użycia
- 101 - Bateria niska, nie ładuje, używane – alarm!
- 110 - Bateria niska, ładowanie, brak użycia – OK
- 111 - Bateria niska, ładowanie, używane – wydłuż ładowanie

9 Wnioski

Zadanie wykonaliśmy korzystając z metody tabel Karnaughta, wyprowadzając z nich wzory opisujące sygnały wyjściowe. Przekształcenia dokonywaliśmy tak, aby jedynym operatorem logicznym w końcowym wyrażeniu był operator NAND, co pozwoliło zbudować układ bazujący wyłącznie na bramkach tego typu. Jedynym wyjątkiem od tego był układ testujący, ale założyliśmy, że tylko w schemacie właściwym mają być użyte tylko i wyłącznie bramki NAND (w układzie testującym są bramki NOR i NAND).

Utworzenie układu testującego było wysoce uzasadnione ze względu na możliwość pomyłki przy przekształceniach i łączeniu przewodów układu. Pozwoliło to na efektywną weryfikację poprawności dokonanych przekształceń i konstrukcji układu. I teraz sami się przekonaliśmy, że napewno nasz układ działa:)

Przy tym ćwiczeniu nauczyliśmy się bardzo dużo, między innymi jak poznane prawa na przedmiocie logika można wykorzystać w praktyce, gdyż, aby skorzystać tylko i wyłącznie z bramek NAND musieliśmy wykonać pewne przekształcenia, które pozwoliły nasz schemat w taki sposób zapisać(aby tylko używać bramek NAND).

Podczas metody Karnaughta zdecydowaliśmy się skorzystać z metody grupowania jedynek, ponieważ było to dla nas nieco bardziej czytelne, chociaż istnieje

też metoda grupowania zer.

Nauczyliśmy się również, że w przypadku przerzutnika RS sygnał wyjściowy może być wykorzystywany jako wejście dla innych bramek. Takie rozwiązanie stosuje się, gdy chcemy zachować informację o sygnale, który pojawił się wcześniej.

10 Bibliografia

- Wykład "Technika cyfrowa" autorstwa Jacka Długopolskiego
- "NI Multisim: Word Generator digital sequences" film na platformie Youtube autorstwa NTS
- "How Flip Flops Work - The Learning Circuit" film na platformie Youtube autorstwa element14 presents
- SR flip flop strona GeeksForGeeks
- "[Podstawy] #35 - Metoda Karnaugh (cz. 1) | Minimalizacja funkcji logicznych" film na platformie Youtube autorstwa Damiana Orzechowskiego