

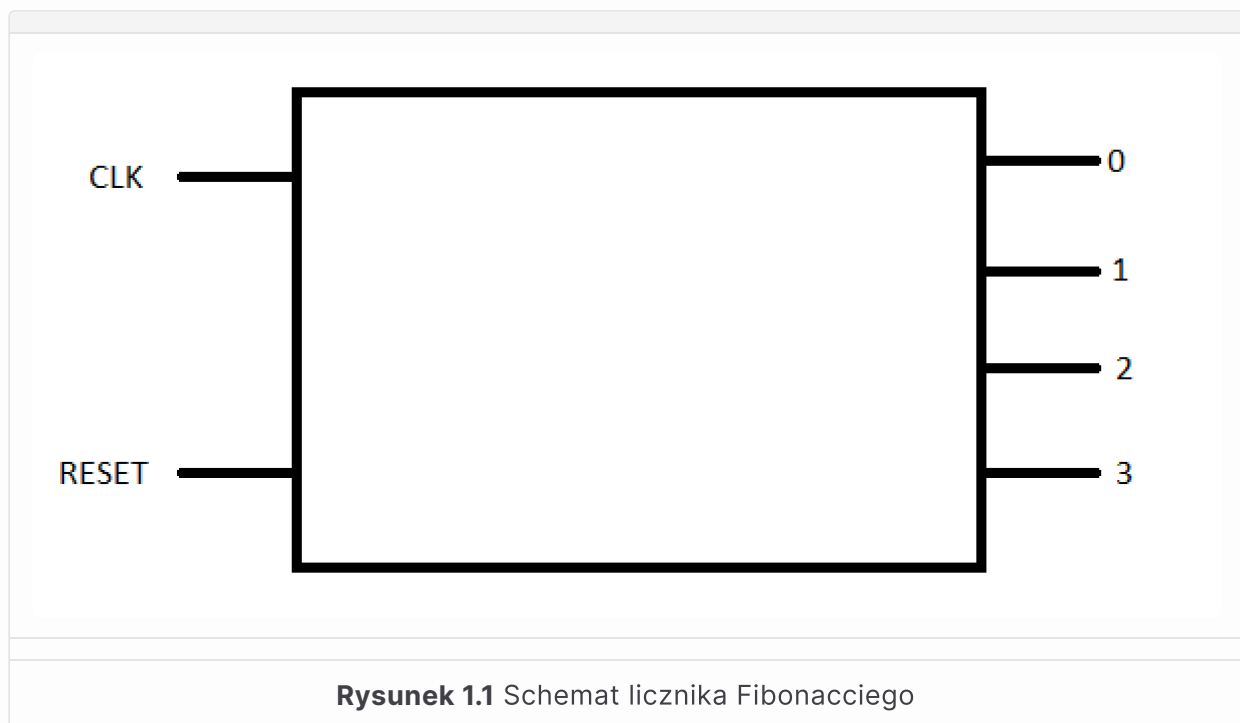
# Technika Cyfrowa - Sprawozdanie z ćwiczenia nr 2: Projekt czterobitowego licznika Fibonacciego

## 1. Wstęp

Celem projektu było zaprojektowanie czterobitowego licznika, który zlicza według ciągu Fibonacciego:

**0, 1, 2, 3, 5, 8, 13, 0, 1, ...**

Licznik działa w pętli i wartość jest wyświetlana na wyświetlaczu siedmiosegmentowym. W projekcie wykorzystano **jeden typ przerzutnika** (T) oraz dowolne bramki logiczne. Rysunek poglądowy układu wygląda następująco:



Układ posiada 2 wejścia: zegarowe i reset oraz 4 wyjścia dla bitów, na których zapisana jest liczba należąca do ciągu

## 2. Schemat układu

---

### Sekwencja stanów licznika Fibonacciego:

0 (0000) → 1 (0001) → 2 (0010) → 3 (0011) → 5 (0101) → 8 (1000) → 13 (1101) → 0 (0000) →  
...

**Tabela 2.1** Tabela przejścia automatu w kolejnym kroku czasowym n

n	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	->	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
0	0	0	0	0	->	0	0	0	1
1	0	0	0	1	->	0	0	1	0
2	0	0	1	0	->	0	0	1	1
3	0	0	1	1	->	0	1	0	1
4	0	1	0	1	->	1	0	0	0
5	1	0	0	0	->	1	1	0	1
6	1	1	0	1	->	0	0	0	0

**Tabela 2.2** Tabela przejść 4 przerzutników typu T w czasie

n	T <sub>3</sub>	T <sub>2</sub>	T <sub>1</sub>	T <sub>0</sub>
0	0	0	0	1
1	0	0	1	1
2	0	0	0	1
3	0	1	1	0
4	1	1	0	1
5	0	1	0	1
6	1	1	0	1

---

### 3. Tabele Karnaugh

**Tabela 3.1** Tabela Karnaugh dla wejścia  $T_3$  w czasie  $n$

$Q_3Q_2 \setminus Q_1Q_0$	00	01	11	10
00	0	0	0	0
01	0	1	0	0
11	0	1	0	0
10	0	0	0	0

Z tabeli 3.1 wynika wzór na  $T_3$  :

$$T_3 = Q_2 \overline{Q_1} Q_0$$

**Tabela 3.2** Tabela Karnaugh dla wejścia  $T_2$  w czasie  $n$

$Q_3Q_2 \setminus Q_1Q_0$	00	01	11	10
00	0	0	1	0
01	0	1	0	0
11	0	1	0	0
10	1	0	0	0

Z tabeli 3.2 wynika wzór na  $T_2$  :

$$T_2 = Q_2 \overline{Q_1} Q_0 + \overline{Q_3} \overline{Q_2} Q_1 Q_0 + Q_3 Q_2 Q_1 Q_0$$

**Tabela 3.3** Tabela Karnaugh dla wejścia  $T_1$  w czasie  $n$

$Q_3Q_2 \setminus Q_1Q_0$	00	01	11	10
00	0	1	1	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

Z tabeli 3.3 wynika wzór na  $T_1$  :

$$T_1 = \overline{Q_3} Q_2 Q_0$$

**Tabela 3.4** Tabela Karnaugh dla wejścia T<sub>0</sub> w czasie n

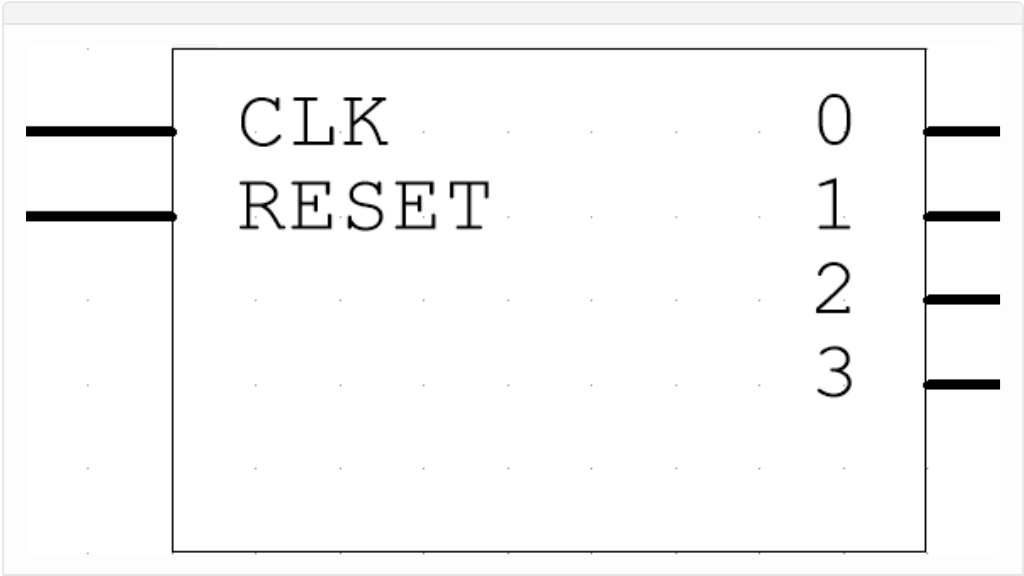
$Q_3Q_2 \backslash Q_1Q_0$	00	01	11	10
00	1	1	0	1
01	0	1	0	0
11	0	1	0	0
10	1	0	0	0

Z tabeli 3.4 wynika wzór na T<sub>0</sub> :

$$T_0 = \overline{Q_3}Q_2Q_1 + Q_3Q_2Q_0 + Q_2\overline{Q_1}Q_0 + \overline{Q_2}Q_1Q_0$$

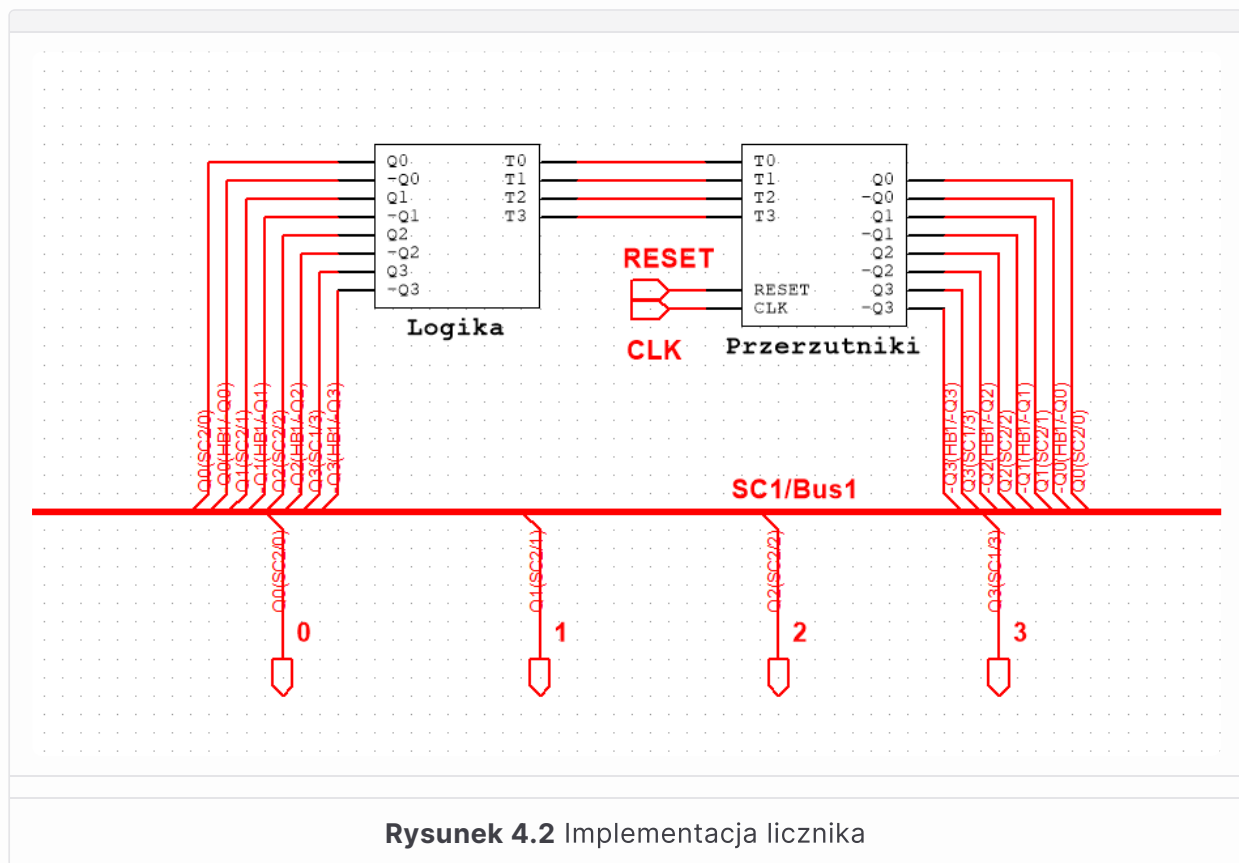
#### 4. Schemat układu

Zaprojektowany licznik wygląda następująco:

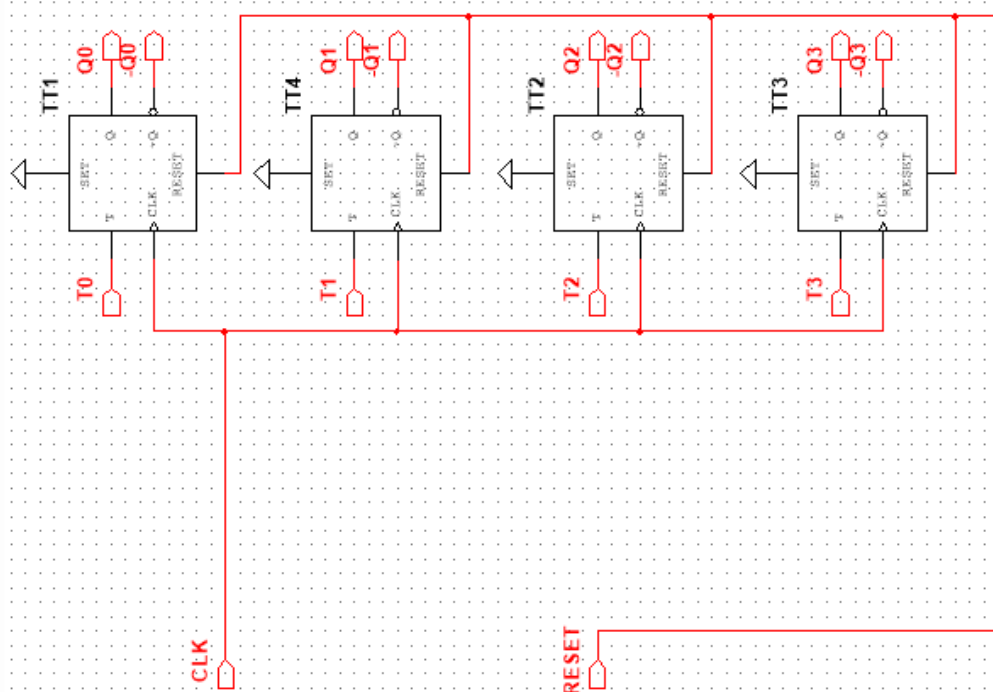


**Rysunek 4.1** Schemat licznika Fibonacciego w programie Multisim

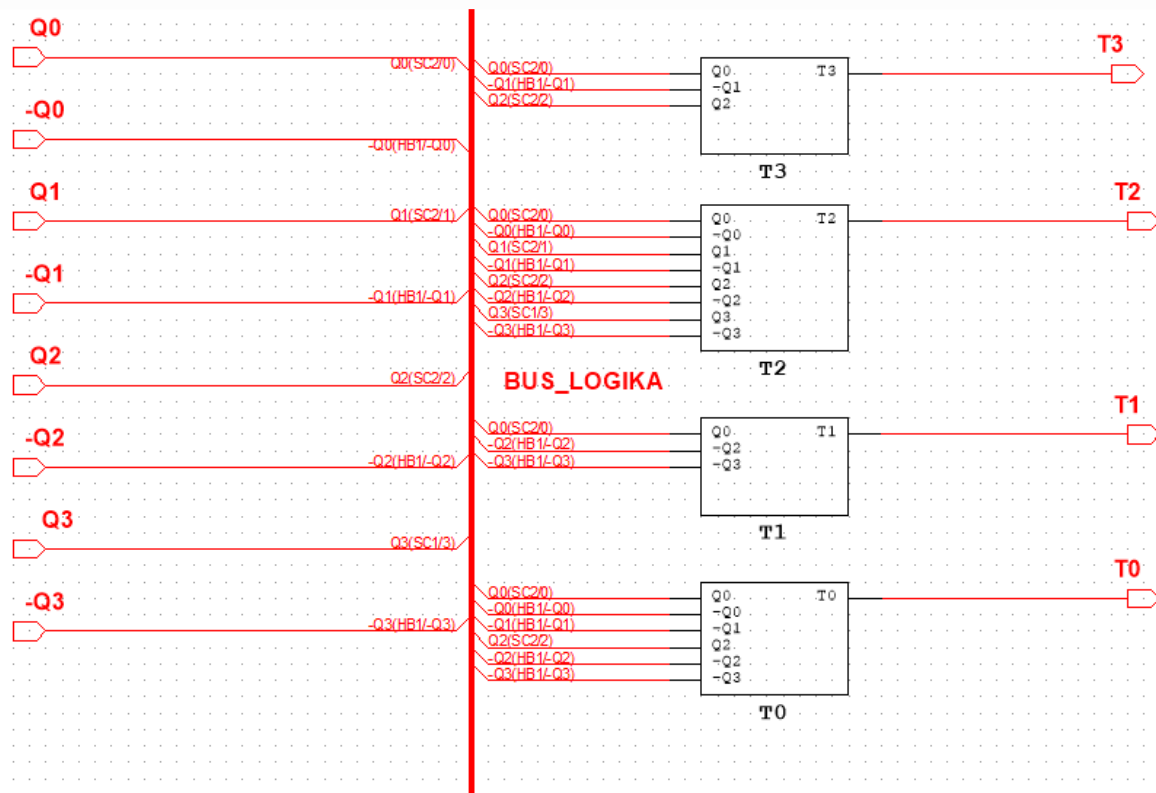
Poniżej przedstawiona jest implementacja:



**Rysunek 4.2** Implementacja licznika

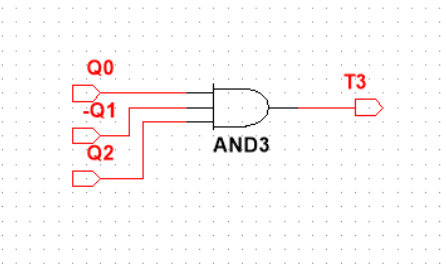
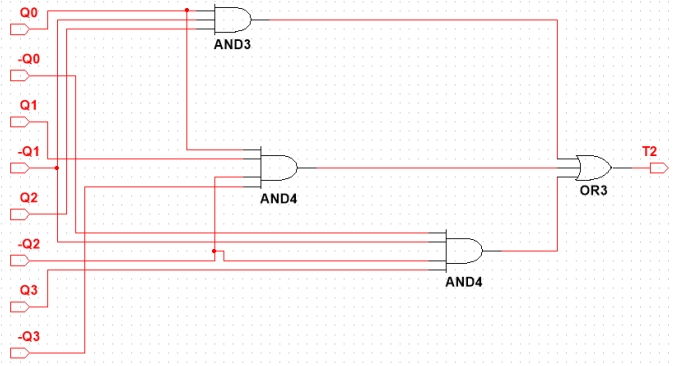
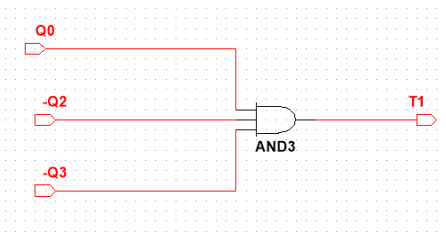
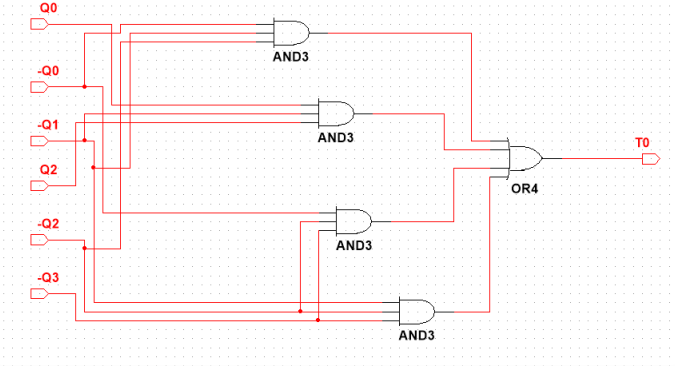


**Rysunek 4.3** Implementacja podukładu "Przerzutniki"

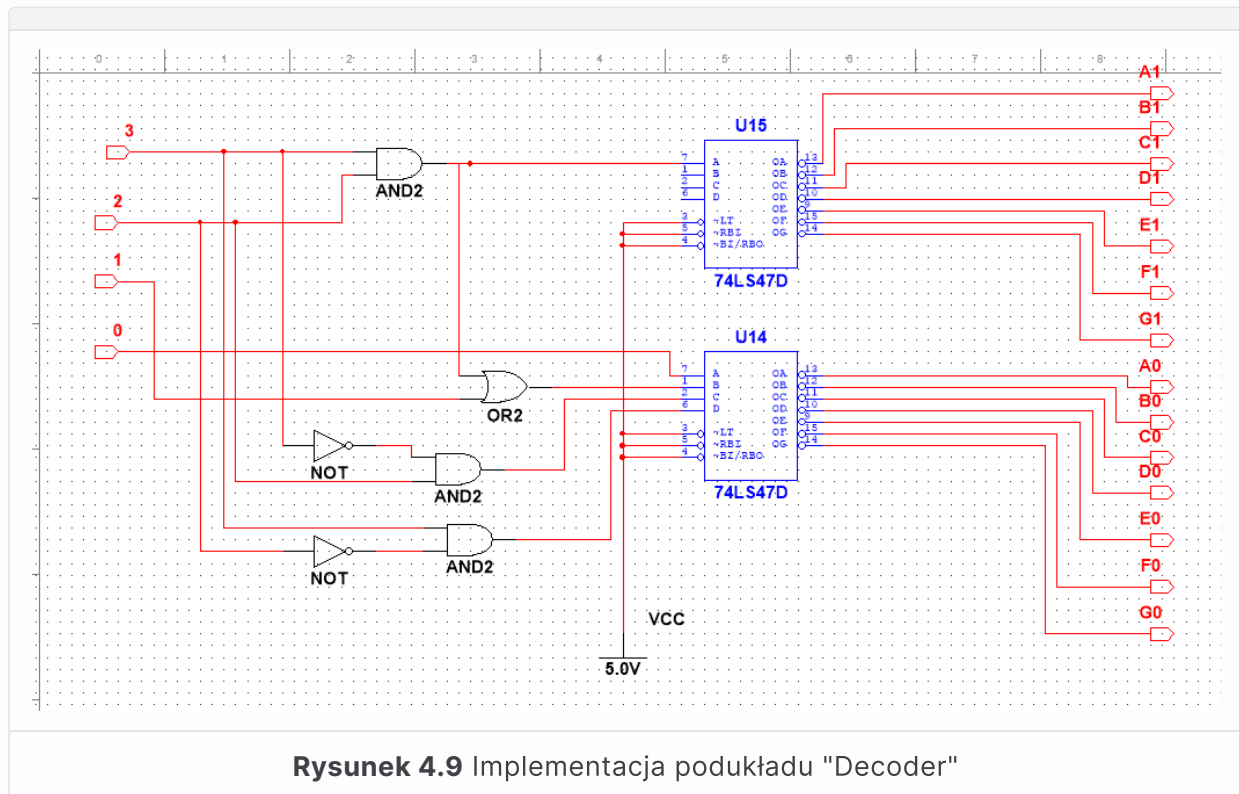


**Rysunek 4.4** Implementacja podukładu "Logika"

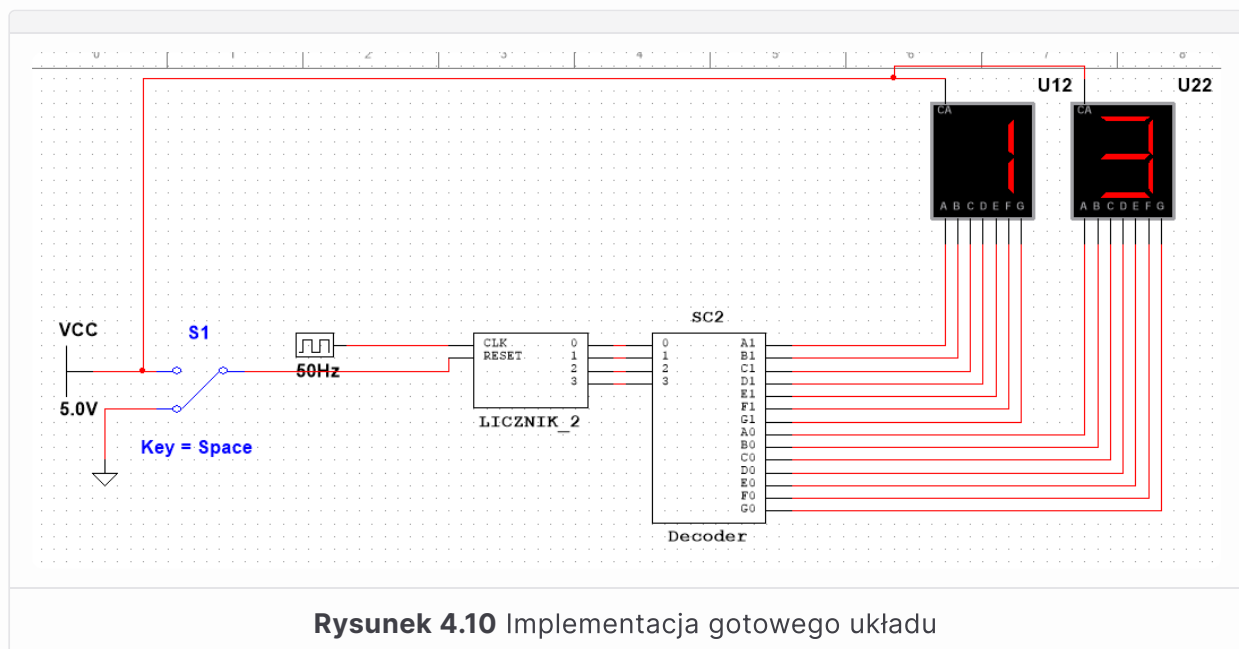
Przedstawione schematy wykorzystują magistrale komunikacyjne. Magistrale te służą do komunikacji między poszczególnymi blokami układu oraz stanowią graficzne uproszczenie układu.

	
<b>Rysunek 4.5</b> Implementacja podukładu "T3"	<b>Rysunek 4.6</b> Implementacja podukładu "T2"
	
<b>Rysunek 4.7</b> Implementacja podukładu "T1"	<b>Rysunek 4.8</b> Implementacja podukładu "T0"

Aby wyświetlić liczby na 2 wyświetlaczach siedmiosegmentowych zaprojektowaliśmy odpowiedni dekodery korzystający z konwerterów BCD-TO-7-SEGMENT-DISPLAY:



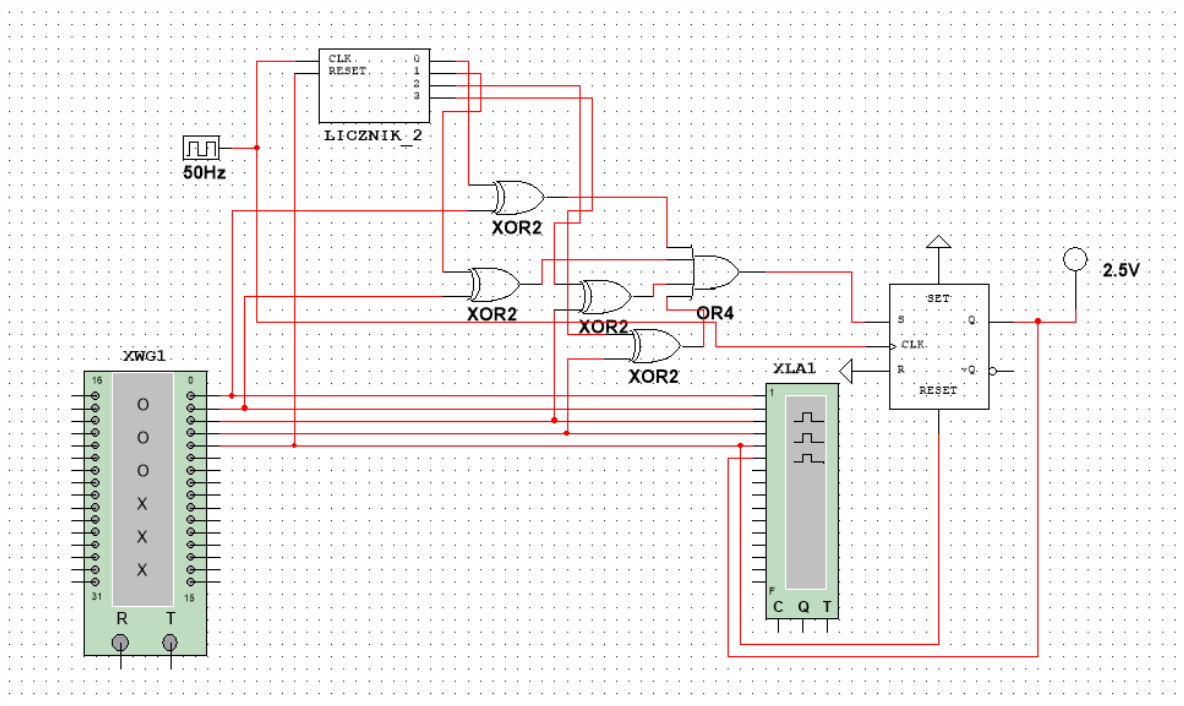
Gotowy układ wraz z wyświetlaczami siedmiosegmentowymi wygląda następująco:



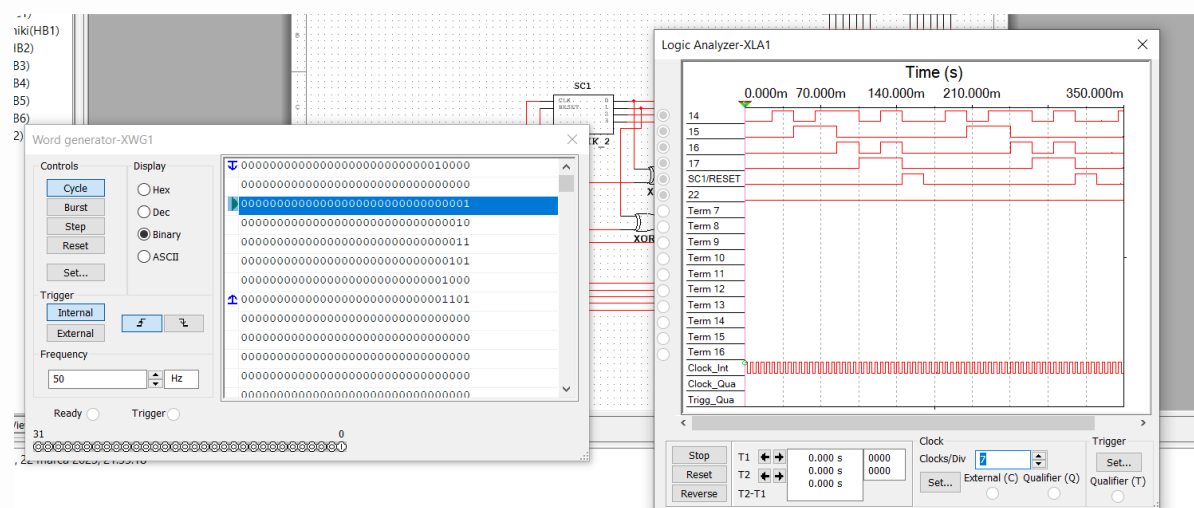
## 5. Układ testowy

Korzystając z wcześniejszych podukładów zrobiliśmy układ testowy w celu sprawdzenia poprawności naszego licznika, korzystając z generatora słów oraz analizatora stanów logicznych. Gdy układ jest wadliwy dioda zapala się na czerwono





Poniżej znajdują się wyniki analizatora logicznego wraz z ustawieniem generatora słów:



Na podstawie analizowanych testów widać, że sekwencja czterech bitów zmienia się zgodnie z oczekiwaniami, bit piąty spełnia funkcję resetowania, a szósty bit pozostaje w stanie niskim, co potwierdza poprawne działanie układu. Generator słów wprowadza kolejne sekwencje testowe, a układ reaguje prawidłowo na wszystkie badane kombinacje wejściowe.

## Wnioski

- Układ poprawnie realizuje zapętłony ciąg Fibonacciego.
- Alternatywnie można byłoby zaprojektować licznik z użyciem pamięci ROM zawierającej kolejne wartości, co upraszczałoby logikę, ale zwiększałoby koszt i złożoność układu.
- Minimalna liczba przerzutników potrzebna do stworzenia 4-bitowego licznika to 4

## Praktyczne zastosowania

- Tego typu licznik można zastosować w systemach losowych lub efektach świetlnych (np. animacje LED w sekwencji Fibonacciego), gdzie nieregularne sekwencje liczb zapewniają bardziej „naturalny” lub mniej przewidywalny efekt.
- Przedstawiony poniżej system wykorzystuje licznik do kontrolowania dostępu do lodówki. Lodówka automatycznie blokuje się po każdym otwarciu na czas zgodny z sekwencją licznika. Na wyświetlaczu widoczny jest aktualny czas blokady, a diody pokazują liczbę poprzednich otwarć. System można zresetować wrzucając monetę do skarbonki.

