Jan Czechowski

Kinga Konieczna

Projekt 1 - Dekoder 7-segmentowy

20 kwietnia 2025

Spis treści

1. Cel laboratorium

Celem laboratorium jest zaprojektowanie i implementacja układu dekodera zrealizowanego za pomocą sieci bramek logicznych, wykorzystujących zminimalizowane równania logiczne. Zadaniem układu jest dekodowanie wartości wejściowych zapisanych w kodzie binarnym (zakres od 0 do 9) oraz wyświetlanie odpowiednich cyfr na **wyświetlaczu 7-segmentowym.**

2. Wyznaczenie wskaźnika

Indeksy:

— Jan Czechowski – 337066
— Kinga Konieczna – 337072

Najmłodsze cyfry indeksów to odpowiednio 6 i 2.
Sumując je:

$$6 + 2 = 8$$

Otrzymany wynik, czyli nasz wskaźnik do zbioru, to **8**.

3. Minimalizacja metodą tablic Karnaugha dla 1 (SOP)

3.1. Opis metody

Proces minimalizacji funkcji metodą Karnaugha obejmuje następujące kroki:

— utworzenie mapy Karnaugha w oparciu o zbiór F oraz liczbę zmiennych wejściowych,
— zaznaczenie w tabeli wartości logicznych „1” odpowiadających elementom zbioru F,
— tworzenie prostokątnych grup z jedynek — każda grupa musi mieć rozmiar będący potęgą liczby 2 (np. 1, 2, 4, 8),
— określenie tzw. implikantów prostych, czyli wyrażeń logicznych odpowiadających każdej z grup,
— usunięcie zbędnych implikantów, które nie są konieczne do pokrycia funkcji,
— zapisanie końcowego wyrażenia jako sumy logicznej (Suma iloczynów) wynikającej z pozostałych implikantów.

Otrzymane wyrażenie stanowi zminimalizowaną postać funkcji logicznej.

3.2. Tablica prawdy funkcji b

Tab. 1: Tablica prawdy dla funkcji b

x_3	x_2	x_1	x_0	b
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	–
1	0	1	1	–
1	1	0	0	–
1	1	0	1	–
1	1	1	0	–
1	1	1	1	–

3.3. Tablica Karnaugha funkcji b

Tab. 2: Tablica Karnaugha dla funkcji b

b — x_3, x_2 \ x_1, x_0	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	-	-	-	-
10	1	1	-	-

Mintermy: 0, 1, 2, 3, 8 i 9. Wyrażenie: $\overline{x_2}$
Mintermy: 0, 4 i 8. Wyrażenie: $\overline{x_0} \cdot \overline{x_1}$
Mintermy: 3 i 7. Wyrażenie: $x_0 \cdot x_1$

3.4. Minimalne równanie boolowskie funkcji b

$$b = \overline{x_2} + \overline{x_1} \cdot \overline{x_0} + x_1 \cdot x_0$$

4. Minimalizacja metodą tablic Karnaugha dla 0 (POS)

4.1. Opis metody

Proces minimalizacji funkcji metodą Karnaugha obejmuje następujące kroki:

— utworzenie mapy Karnaugha w oparciu o zbiór F oraz liczbę zmiennych wejściowych,
— zaznaczenie w tabeli wartości logicznych „1” odpowiadających elementom zbioru F,
— tworzenie prostokątnych grup z jedynek — każda grupa musi mieć rozmiar będący potęgą liczby 2 (np. 1, 2, 4, 8),
— określenie tzw. implikantów prostych, czyli wyrażeń logicznych odpowiadających każdej z grup,
— usunięcie zbędnych implikantów, które nie są konieczne do pokrycia funkcji,
— zapisanie końcowego wyrażenia jako iloczynu logicznego (iloczyn sum) wynikającej z pozostałych implikantów.

Otrzymane wyrażenie stanowi zminimalizowaną postać funkcji logicznej.

4.2. Minimalizacja funkcji c

4.2.1. Tablica prawdy

Tab. 3: Tablica prawdy dla funkcji c

x_3	x_2	x_1	x_0	c
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	–
1	0	1	1	–
1	1	0	0	–
1	1	0	1	–
1	1	1	0	–
1	1	1	1	–

4.2.2. Tablica Karnaugha

Tab. 4: Tablica Karnaugha dla funkcji c

c — x_3, x_2 \ x_1, x_0	00	01	11	10
00	1	1	1	0
01	1	1	1	1
11	-	-	-	-
10	1	1	-	-

Makstermy: 2.

4.2.3. Minimalne równanie

$$c = x_2 + \overline{x_1} + x_0$$

4.3. Minimalizacja funkcji d

4.3.1. Tablica prawdy

Tab. 5: Tablica prawdy dla funkcji d

x_3	x_2	x_1	x_0	d
0	0	0	0	1
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	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	–
1	0	1	1	–
1	1	0	0	–
1	1	0	1	–
1	1	1	0	–
1	1	1	1	–

4.3.2. Tablica Karnaugha

Tab. 6: Tablica Karnaugha dla funkcji d

d: x_3, x_2 \ x_1, x_0	00	01	11	10
00	1	0	1	1
01	0	1	0	1
11	-	-	-	-
10	1	1	-	-

Makstermy: 1.
Makstermy: 4.
Makstermy: 7.

4.3.3. Minimalne równanie

$$d = (x_3 + x_2 + x_1 + \overline{x_0}) \cdot (\overline{x_2} + x_1 + x_0) \cdot (\overline{x_2} + \overline{x_1} + \overline{x_0})$$

5. Minimalizacja metodą ekspansji systematycznej

5.1. Opis metody

Minimalizacja funkcji metodą ekspansji systematycznej obejmuje następujące etapy:

1. Wyznaczenie macierzy odpowiadającej zbiorom F (funkcja) oraz R (rezerwacje),
2. Utworzenie macierzy blokującej dla każdej kostki z F względem zbioru R,
3. Wyznaczenie wszystkich minimalnych pokryć kolumnowych dla macierzy blokującej,
4. Określenie implikantów wynikających z każdego z pokryć kolumnowych,
5. Zbudowanie macierzy występowania implikantów względem oryginalnej macierzy F,
6. Znalezienie minimalnego pokrycia kolumnowego macierzy występowania implikantów.

Końcowym wynikiem procesu jest suma implikantów należących do minimalnego pokrycia, która stanowi zminimalizowaną postać funkcji wejściowej.

5.2. Minimalizacja funkcji e

5.2.1. Zbiory F i R

Tab. 7: Zbiory F i R dla funkcji e

Wartość	Kod binarny	Segment e
0	0000	1
2	0010	1
6	0110	1
8	1000	1
1	0001	0
3	0011	0
4	0100	0
5	0101	0
7	0111	0
9	1001	0

5.2.2. Macierze blokujące

1. Dla $k_0 = 0000$:

$$B(k_0, R) = \begin{pmatrix} 0001 \\ 0100 \end{pmatrix} \Rightarrow L' = \{0, 2\} \Rightarrow [*0*0]$$

2. Dla $k_1 = 0010$:

$$B(k_1, R) = \begin{pmatrix} 0001 \\ 0011 \\ 0110 \end{pmatrix} \Rightarrow L' = \{1\} \Rightarrow [**10]$$

3. Dla $k_2 = 0110$:

$$B(k_2, R) = \begin{pmatrix} 0100 \\ 0111 \\ 0010 \end{pmatrix} \Rightarrow L' = \{2\} \Rightarrow [01*0]$$

4. Dla $k_3 = 1000$:

$$B(k_3, R) = \begin{pmatrix} 1001 \\ 0000 \end{pmatrix} \Rightarrow L' = \{3\} \Rightarrow [1***]$$

5.2.3. Minimalne pokrycie

Tab. 8: Minimalne pokrycie implikantów dla funkcji e

Implikant	0000	0010	0110	1000
$I_0 = [*0*0]$	✓	✓		✓
$I_1 = [**10]$		✓		
$I_2 = [01*0]$			✓	
$I_3 = [1***]$				✓

Wybór minimalnego pokrycia: $I_0 + I_2$

5.2.4. Wynik minimalizacji

$$e = \overline{x_0} \cdot (\overline{x_2} + x_1)$$

5.3. Minimalizacja funkcji f

5.3.1. Zbiory F i R

Tab. 9: Zbiory F i R dla funkcji f

Wartość	Kod binarny	Segment f
0	0000	1
4	0100	1
5	0101	1
6	0110	1
8	1000	1
9	1001	1
1	0001	0
2	0010	0
3	0011	0
7	0111	0

5.3.2. Macierze blokujące

1. Dla $k_0 = 0000$:

$$B(k_0, R) = \begin{pmatrix} 0001 \\ 0010 \\ 0011 \\ 0111 \end{pmatrix} \Rightarrow L' = \{0, 1\} \Rightarrow [**00]$$

2. Dla $k_1 = 0100$:

$$B(k_1, R) = \begin{pmatrix} 0101 \\ 0110 \\ 0111 \\ 0011 \end{pmatrix} \Rightarrow L' = \{0, 2\} \Rightarrow [*1*0], \quad L' = \{1, 2\} \Rightarrow [*10*]$$

3. Dla $k_2 = 0101$:

$$B(k_2, R) = \begin{pmatrix} 0100 \\ 0111 \\ 0110 \\ 0010 \end{pmatrix} \Rightarrow L' = \{1, 2\} \Rightarrow [*10*]$$

4. Dla $k_3 = 0110$:

$$B(k_3, R) = \begin{pmatrix} 0111 \\ 0100 \\ 0101 \\ 0001 \end{pmatrix} \Rightarrow L' = \{0, 2\} \Rightarrow [*1*0]$$

5. Dla $k_4 = 1000$:

$$B(k_4, R) = \begin{pmatrix} 1001 \\ 1010 \\ 1011 \\ 1111 \end{pmatrix} \Rightarrow L' = \{3\} \Rightarrow [1***]$$

6. Dla $k_5 = 1001$:

$$B(k_5, R) = \begin{pmatrix} 1000 \\ 1011 \\ 1010 \\ 1110 \end{pmatrix} \Rightarrow L' = \{3\} \Rightarrow [1***]$$

5.3.3. Macierz pokrycia

Tab. 10: Macierz pokrycia implikantów dla funkcji f

Implikant	0000	0100	0101	0110	1000	1001
$I_0 = [**00]$	✓	✓			✓	
$I_1 = [*1*0]$		✓		✓		
$I_2 = [*10*]$		✓	✓			
$I_3 = [1***]$					✓	✓

5.3.4. Wybór minimalnego pokrycia

Wybrane implikanty: $I_0 + I_1 + I_2 + I_3$

5.3.5. Wynik minimalizacji

$$f = \overline{x_1} \cdot \overline{x_0} + x_2 \cdot \overline{x_0} + x_2 \cdot \overline{x_1} + x_3$$

6. Minimalizacja metodą ekspansji heurystycznej

6.1. Opis metody

Minimalizacja metodą ekspansji heurystycznej składa się z:

— Wyznaczenia zbiorów F i R,
— Dla każdej nierozpatrzonej kostki:
  1. Wyznaczenia macierzy blokującej względem zbioru R,
  2. Wyznaczenia jednego (pierwszego) pokrycia kolumnowego,
  3. Wyznaczenia implikantu wynikającego z pokrycia,
  4. Pokrycia wszystkich kostek objętych przez dany implikant.

Końcowym wynikiem procesu jest suma wyznaczonych implikantów, które pokrywają cały zbiór F.

6.2. Minimalizacja funkcji g

6.2.1. Zbiory F i R

Tab. 11: Zbiory F i R dla funkcji g

Wartość	Kod binarny	Segment g
2	0010	1
3	0011	1
4	0100	1
5	0101	1
6	0110	1
8	1000	1
9	1001	1
0	0000	0
1	0001	0
7	0111	0

6.2.2. Macierze blokujące i implikanty

— Dla $k_0 = 0010$: $L' = \{0, 1\}$ $I_0 = [**10]$, pokrywa także k_4
— Dla $k_1 = 0011$: $L' = \{1, 2\}$ $I_1 = [*01*]$, pokrywa tylko k_1
— Dla $k_2 = 0100$: $L' = \{2, 0\}$ $I_2 = [*1*0]$
— Dla $k_3 = 0101$: $L' = \{1, 2\}$ $I_3 = [*10*]$
— Dla $k_5 = 1000$: $L' = \{3\}$ $I_4 = [1***]$, pokrywa także k_6

6.2.3. Macierz pokrycia

Tab. 12: Macierz pokrycia implikantów dla funkcji g

Implikant	0010	0011	0100	0101	0110	1000 / 1001
$I_0 = [**10]$	✓				✓	
$I_1 = [*01*]$		✓				
$I_2 = [*1*0]$			✓		✓	
$I_3 = [*10*]$				✓		
$I_4 = [1***]$						✓

6.2.4. Wynik minimalizacji

Wybrane implikanty: $I_0 + I_1 + I_2 + I_3 + I_4$

$$g = \overline{x_2} \cdot x_1 + x_2 \cdot \overline{x_0} + x_2 \cdot \overline{x_1} + x_3$$

6.3. Minimalizacja funkcji a

6.3.1. Zbiory F i R

Tab. 13: Zbiory F i R dla funkcji a

Wartość	Kod binarny	Segment a
0	0000	1
2	0010	1
3	0011	1
5	0101	1
6	0110	1
7	0111	1
8	1000	1
9	1001	1
1	0001	0
4	0100	0

6.3.2. Macierze blokujące i implikanty

— $k_0 = 0000 \rightarrow L' = \{0, 2\}$ $I_0 = [*0*0]$ pokrywa także k_1, k_6
— $k_2 = 0011 \rightarrow L' = \{1\}$ $I_1 = [**1*]$ pokrywa k_2, k_4, k_5
— $k_3 = 0101 \rightarrow L' = \{0, 2\}$ $I_2 = [*1*1]$ pokrywa k_5
— $k_7 = 1001 \rightarrow L' = \{3\}$ $I_3 = [1***]$ pokrywa k_6, k_7

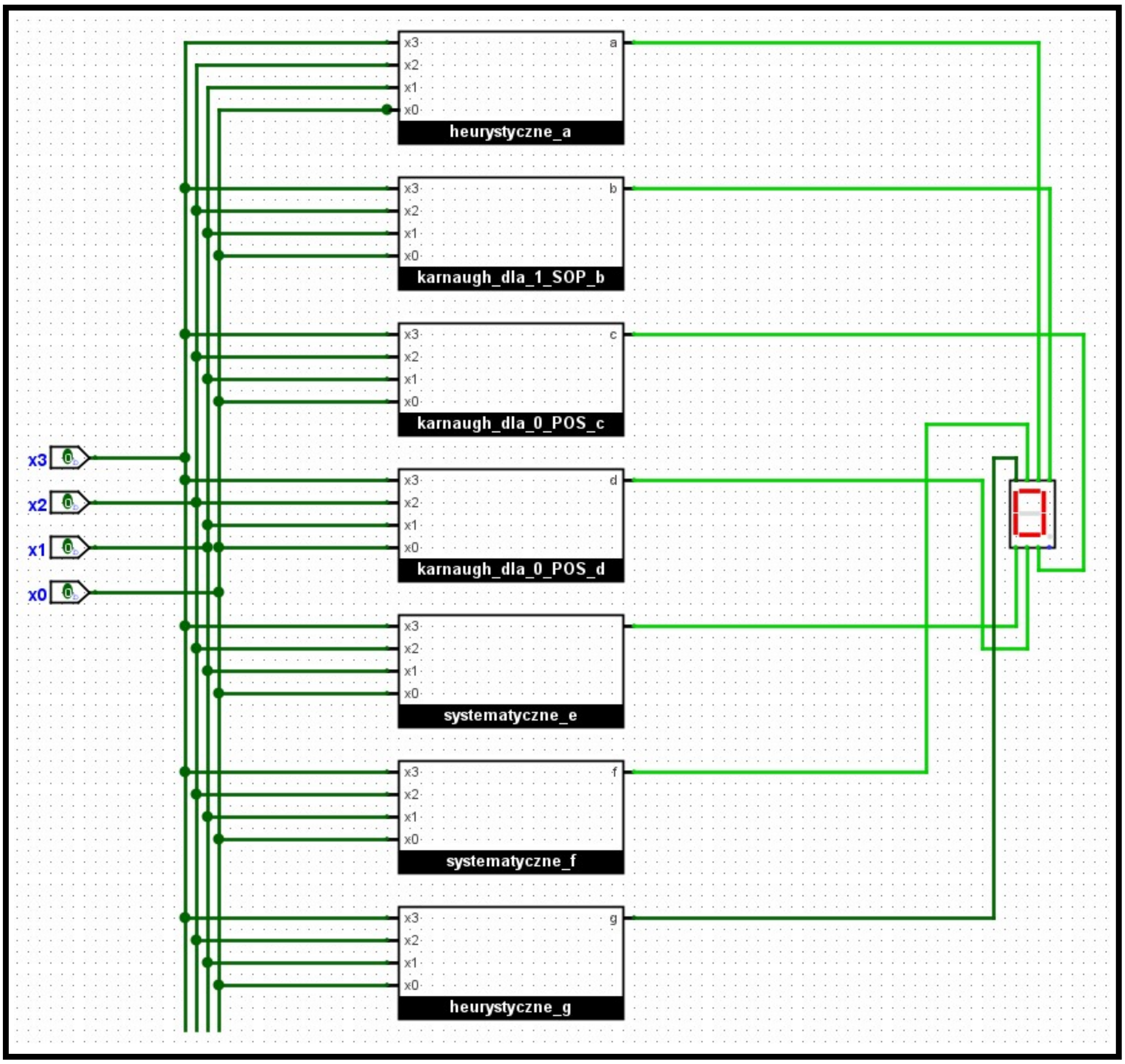
6.3.3. Wynik minimalizacji

$$a = \overline{x_2} \cdot \overline{x_0} + x_1 + x_2 \cdot x_0 + x_3$$

7. Układ dekodera

Na **Rys. 1** poniżej przedstawiono strukturę dekodera 7-segmentowego, w której każdemu z segmentów (oznaczonych literami a–g) przypisano osobny blok logiczny. Każdy z tych bloków został zaprojektowany z wykorzystaniem jednej z czterech różnych metod minimalizacji funkcji logicznych.

W celu zwiększenia przejrzystości całego układu, poszczególne bloki funkcjonalne zostały zaprezentowane w formie tzw. **czarnych skrzynek**, tj. bez szczegółowej reprezentacji wewnętrznej logiki.



Rys. 1: Układ schematu blokowego dekodera zrealizowanego w programie Logisim

8. Symulacja układu

Link do filmiku z nagraną symulacją: `https://youtu.be/vfvw0N3Vah0s`.

9. Wnioski

Zgodność minimalizacji funkcji logicznych została potwierdzona poprzez porównanie tablicy prawdy przed i po uproszczeniu. Natomiast poprawność implementacji funkcji w programie Logisim została zweryfikowana przez zestawienie tablicy prawdy obwodu z tablicą prawdy funkcji po minimalizacji. Zbudowany układ działa poprawnie i realizuje wymagania postawione w zadaniu laboratoryjnym. Przeprowadzone laboratorium pozwoliło na zapoznanie się z metodami minimalizacji funkcji logicznych oraz ich praktyczną realizacją w środowisku symulacyjnym.

Po przeprowadzeniu wszystkich testów oraz realizacji układu logicznego, można wyciągnąć następujące wnioski:

1. **Skuteczność minimalizacji** – Proces minimalizacji funkcji logicznych przy użyciu metod takich jak Karnaugh, ekspansja systematyczna oraz ekspansja heuretyczna okazał się efektywny w redukcji liczby elementów obwodu. Uproszczenie funkcji prowadzi do oszczędności zarówno w projekcie, jak i w czasie pracy układu.
2. **Walidacja projektu** – Sprawdzanie poprawności minimalizacji przez porównanie tablic prawdy jest skutecznym sposobem weryfikacji funkcji logicznych, co zostało potwierdzone testami w Logisimie.
3. **Praktyczna implementacja** – Układ zaprezentowany w laboratorium spełnia wymagania projektowe, a wyniki symulacji w Logisimie odpowiadają założeniom zadania. Minimalizacja nie tylko poprawiła wydajność, ale również zmniejszyła złożoność samego układu logicznego.
4. **Optymalizacja w praktyce** – Optymalizacja obwodów logicznych poprzez minimalizację funkcji pozwala na tworzenie bardziej złożonych układów przy mniejszym zużyciu zasobów, co jest istotnym aspektem w kontekście rzeczywistych zastosowań w elektronice.

Spis tabel

Spis rysunków