Opracował: dr inż. Krzysztof Smyksy,
Pracownia Mechanizacji, Automatyzacji i Projektowania Odlewni

Zminimalizować funkcję logiczną - „y”, metodą tablic Karnaugh`a:

$$y = \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x_3} + x_2 \cdot x_3$$

- Tabela stanów:
  składnik czerwony: wartość 1 dla stanu wejść: 011,
  składnik niebieski: wartość 1 dla stanu wejść: 100 oraz 110,
  składnik fioletowy: wartość 1 dla stanu wejść: 011 oraz 111,

  *w **sumie logicznej** wystarczy, żeby jeden składnik był równy 1- a całe wyrażenie jest równe 1,

L.p.	x_1	x_2	x_3	y		
0	0	0	0	**0**		
1	0	0	1	**0**		
2	0	1	0	**0**		
3	0	1	1	**1**		
4	1	0	0	**1**		
5	1	0	1	**0**		
6	1	1	0	**1**		
7	1	1	1	**1**		

- Tabela Karnaugha:

		x_2, x_3			
		00	01	11	10
x_1	0	**0**	**0**	**1**	**0**
	1	**1**	**0**	**1**	**1**

*w kolorowych polach wartość funkcji y dla odpowiednich stanów

<u>Tworząc schemat układu logicznego wyłącznie na elementach NAND</u> – łączymy 1 (wolno łączyć sąsiednie 1 – także występujące na brzegu tablicy i w jej narożach; obejmujemy tylko grupy zawierające: 2,4 lub 8….. komórek; wszystkie 1 muszą należeć do grupy, jeżeli nie możemy danej 1 włączyć do grupy – tworzymy dla niej odrębne wyrażenie logiczne)

W przykładzie występują dwie grupy dla 1: niebieska i fioletowa, *w wyrażeniu opisującym grupę eliminujemy ten sygnał wejściowy, którego wartość ulega zmianie ; jeżeli dany sygnał (który nie zmienia wartości dla danej grupy) ma wartość 1- we wzorze występuje bez negacji, jeżeli ma wartość 0 – sygnał jest zanegowany

dla niebieskiej : $\boldsymbol{x_1 \cdot \overline{x_3}}$

dla fioletowej : $\boldsymbol{x_2 \cdot x_3}$

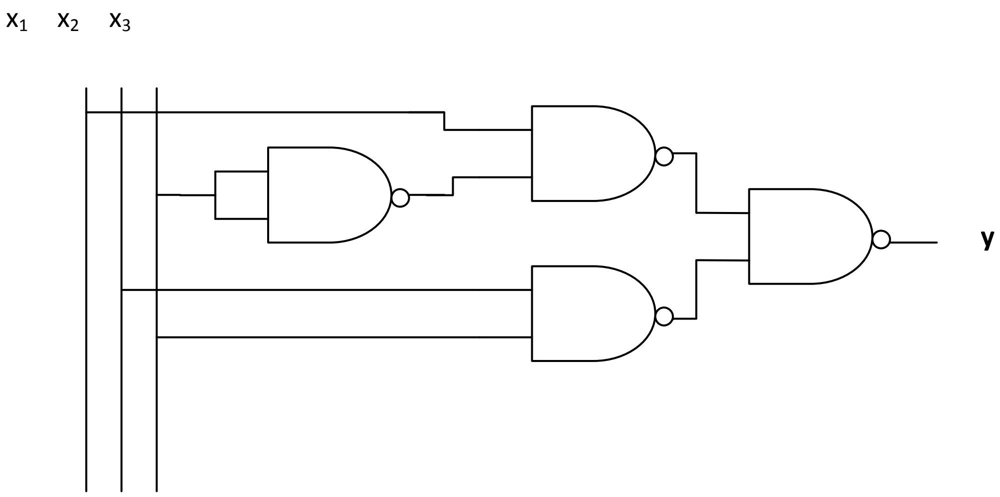
wynikowa postać funkcji „dla 1"- suma iloczynów :

$$\boldsymbol{y = x_1 \cdot \overline{x_3} + x_2 \cdot x_3}$$

Elementy NAND realizują działania: iloczynu i przeczenia, musimy powyższą funkcję zanegować podwójnie (dwukrotna negacja nie zmienia wartości wyrażenia logicznego – prawo podwójnego przeczenia) i zastosować jedno z praw de Morgana:

$$\boldsymbol{y = \overline{\overline{x_1 \cdot \overline{x_3} + x_2 \cdot x_3}} = \overline{\overline{x_1 \cdot \overline{x_3}} \cdot \overline{x_2 \cdot x_3}}}$$

W końcowym wzorze występują jedynie operacje iloczynu i negacji, można więc zbudować schemat funkcjonalny na dwuwejściowych elementach NAND:



Tworząc schemat układu logicznego wyłącznie na elementach NOR – łączymy 0 (wolno łączyć sąsiednie 0 – także występujące na brzegu tablicy i w jej narożach; obejmujemy tylko grupy zawierające: 2,4 lub 8….. komórek; wszystkie 0 muszą należeć do jednej z grup, jeżeli nie możemy danego 0 włączyć do grupy – tworzymy dla niego odrębne wyrażenie logiczne)

W przykładzie występują dwie grupy dla 0: żółta i pomarańczowa,

*w wyrażeniu opisującym grupę eliminujemy ten sygnał wejściowy, którego wartość ulega zmianie ; jeżeli dany sygnał (który nie zmienia wartości dla danej grupy) ma wartość 0- we wzorze występuje bez negacji, jeżeli ma wartość 1 – sygnał jest zanegowany

dla żółtej : $\boldsymbol{x_1 + x_3}$

dla pomarańczowej : $x_2 + \overline{x_3}$

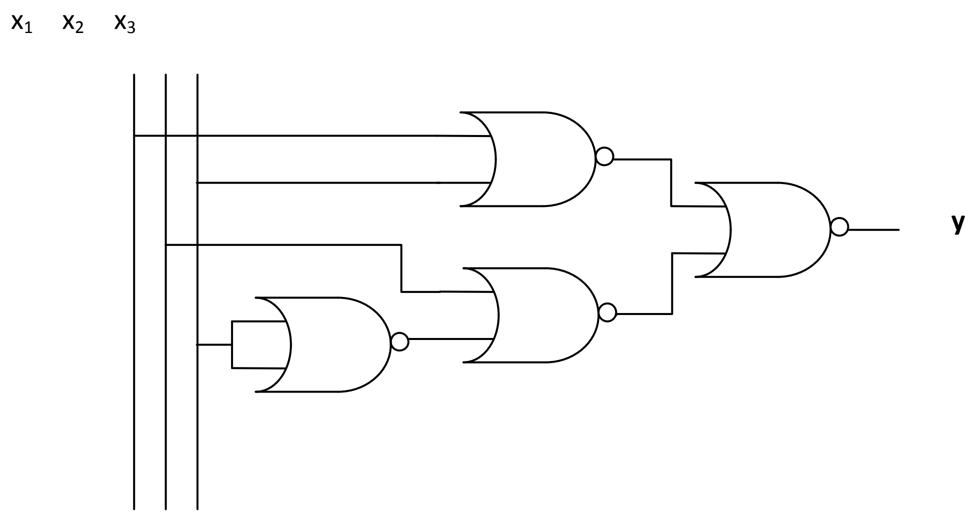
Końcowa postać funkcji „dla 0”- iloczyn sum:

$$\boldsymbol{y} = (\boldsymbol{x_1} + \boldsymbol{x_3}) \cdot (x_2 + \overline{x_3})$$

Elementy NOR realizują działania sumy i przeczenia, musimy powyższą funkcję zanegować podwójnie (dwukrotna negacja nie zmienia wartości wyrażenia logicznego – prawo podwójnego przeczenia) i zastosować jedno z praw de Morgana:

$$\boldsymbol{y} = \overline{\overline{(\boldsymbol{x_1} + \boldsymbol{x_3}) \cdot (x_2 + \overline{x_3})}} = \overline{\overline{(\boldsymbol{x_1} + \boldsymbol{x_3})} + \overline{(x_2 + \overline{x_3})}}$$

W końcowym wzorze występują jedynie operacje sumy i negacji, można więc zbudować schemat funkcjonalny na dwuwejściowych elementach NOR:



*kształt symboli NOR nieco różni się od zalecanego- wykorzystano symbole ze standardowej biblioteki kształtów WORD