

Instrukcja ćwiczenia nr 17

Temat: Synteza układów logicznych kombinacyjnych

1. Cel ćwiczenia:

Celem ćwiczenia jest poznanie zasad budowy układów logicznych kombinacyjnych, metodami syntezy oraz budowy z zastosowaniem do cyfrowych układów sterowania.

2. Wprowadzenie:

Układy przetwarzające informacje, których sygnały są dwustanowe, nazywamy układami logicznymi (układami przełączającymi, cyfrowymi). Dwa stany wielkości fizycznej występującej w tych układach przyjęto oznaczać symbolami 0 i 1.

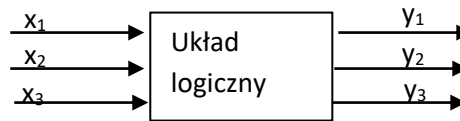
Istnieją dwie grupy układów logicznych: kombinacyjne (jednotaktowe) i sekwencyjne (wielotaktowe). Układem kombinacyjnym jest taki układ, w którym wartości sygnałów wyjściowych zależą tylko od wartości sygnałów wejściowych w danej chwili. Układem sekwencyjnym jest układ, w którym wartości sygnałów wejściowych zależą od wartości sygnałów wejściowych w danej chwili oraz od ich wartości w chwilach poprzednich.

Układ kombinacyjny ogólnie może być przedstawiony na rys. 1 i opisany układem równań:

$$y_1 = f_1(x_1, x_2, x_3)$$

$$y_2 = f_2(x_1, x_2, x_3)$$

$$y_3 = f_3(x_1, x_2, x_3)$$



Rys.1. Schemat układu logicznego.

gdzie f_1 , f_2 i f_3 są funkcjami logicznymi.

Sformułowanie zadania układu logicznego może być w postaci słownej lub tablicy wartości funkcji realizowanej przez układ. Jest to tablica podająca wartości sygnału wyjściowego dla poszczególnych kombinacji wartości sygnałów wejściowych. Proces tworzenia algebraicznego zapisu funkcji, którą ma realizować układ, nazywa się syntezą funkcji. Działanie zmierzające do uzyskania schematu blokowego układu realizującego daną funkcję z uwzględnieniem zastosowanych do jego budowy elementów, nazywa się syntezą strukturalną.

Sposoby przedstawienia informacji w układach cyfrowych

Informacje w układach cyfrowych są przedstawione w przyjętym systemie liczenia zwanym systemem pozycyjnym. System pozycyjny jest systemem, w którym przedstawiamy liczby za pomocą ciągu znaków (cyfr), przy czym z każdą pozycją w ciągu związana jest określona waga.

Wagi poszczególnych pozycji w ciągu są odpowiednimi potęgami podstawy. Każdą liczbę naturalną przedstawiamy w tym systemie przez podane ciągu cyfr stanowiących współczynniki, przez które mnożone są wagi poszczególnych pozycji.

W systemie dziesiętnym, wartość liczby zapisanej w postaci ciągu cyfr $a_n, a_{n-1}, \dots, a_0$ jest określona jako suma

$$L = \sum_{i=0}^n a_i \cdot 10^i$$

Przykład: $425 = 4 \cdot 10^2 + 2 \cdot 10^1 + 5 \cdot 10^0$.

Jeśli podstawę systemu liczenia przyjmiemy liczbę naturalną P , to każdą liczbę L możemy przedstawić w tym systemie jako:

$$L = \sum_{i=0}^n a_i \cdot P^i \text{ gdzie } a_i \in \{0, 1, \dots, P-1\}$$

W układach cyfrowych powszechnie stosuje się dwójkowy system liczenia (podstawa 2). Cyfra binarna nosi nazwę bitu. Wada dwójkowego

systemu liczenia jest fakt, że liczba pozycji wymagana dla przedstawienia dużych liczb jest kilkakrotnie większa w porównaniu w systemie dziesiętnym.

Przykład: $1011010 = 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 64 + 16 + 8 + 2 = 90$.

Operacja arytmetyczne na liczbach w systemie dwójkowym są analogiczne jak w systemie dziesiętnym.

W technice cyfrowej duże znaczenia mają system ósemkowy i szesnastkowy. Niżej podane są oznaczenie i znaki stosowane w systemach pozycyjnych:

B – binarny	$\{0,1\}$
Q – ósemkowy	$\{0,1,2,3,4,5,6,7\}$
D – dziesiętny	$\{0,1,2,3,4,5,6,7,8,9\}$
H – szesnastkowy	$\{0,1,2,3,4,5,6,7,8,9, A, B, C, D, E, F\}$

Przykład:

$$63201_Q = 6 \cdot 8^4 + 3 \cdot 8^3 + 2 \cdot 8^2 + 1 \cdot 8^0 = 6 \cdot 4096 + 3 \cdot 512 + 2 \cdot 64 + 1 = 26241_D$$

$$3E82_H = 3 \cdot 16^3 + 14 \cdot 16^2 + 8 \cdot 16^1 + 2 \cdot 16^0 = 3 \cdot 4096 + 14 \cdot 256 + 8 \cdot 16 + 2 = 16002_D$$

Zamiana liczb:

Liczba ósemkowa na 3 bitową liczbę dwójkową

$$3057_Q = (011 \ 000 \ 101 \ 111) = 011000101111_B$$

Liczba szesnastkowa na 4-bitową liczbę dwójkową

$$AF9_H = (1010 \ 1111 \ 1001) = 101011111001_B$$

Odwrótnie

$$10111010001011_B = (010 \ 111 \ 010 \ 001 \ 011)_B = 27213_Q$$

$$10111010001011_B = (0010 \ 1110 \ 1000 \ 1011)_B = 2E8B_H$$

Podstawowe wiadomości o funkcjach logicznych

Dla funkcji logicznej $y = f(x_1, x_2, \dots, x_n)$, gdzie argumenty przyjmują tylko dwie wartości 0 albo 1, liczba kombinacji wartości argumentów jest skończona. Każdą kombinację wartości argumentów nazwiemy stanem argumentów. Liczba stanów funkcji n argumentów będzie 2^n . Stan argumentów zapisany w postaci ciągu cyfr, z których każda reprezentuje wartość kolejnego argumentu, tworzy liczbę binarną (w systemie dwójkowym). Liczbę tę nazwiemy numerem stanu. Np. dla funkcji czteroargumentowej numer stanu $(x_1, x_2, x_3, x_4) = (1,1,1,1)$ jest 15.

Funkcja logiczna jest w pełni określona, jeśli znane są wartości zmiennej zależnej dla wszystkich stanów argumentów.

Algebra Boole'a

Przedmiotem algebry Boole'a są działania logiczne: negacja, alternatywa i koniunkcja wykonane na zmiennych i liczbach binarnych (0,1). W algebrze tej obowiązują następujące aksjomaty:

1. Prawo przemienności	$x + y = y + x$	$x \cdot y = y \cdot x$
2. Prawo łączności	$x + (y + z) = (x + y) + z$	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$
3. Prawo rozdzielności	$x(y + z) = xy + xz$	$x + yz = (x + y)(x + z)$
4. Prawo Morgan'a	$\overline{x + y} = \bar{x} \cdot \bar{y}$	$\overline{x \cdot y} = \bar{x} + \bar{y}$
5. Prawo podwójnej negacji	$\overline{\bar{x}} = x$	
6. Prawo pochłaniania	$x + xy = x$	$x(x + y) = x$
7. Prawo sklejania	$(x + y)(x + \bar{y}) = x$	$xy + x\bar{y} = x$

Przykłady:

$x + 1 = 1$	$x \cdot 1 = x$
$x + 0 = x$	$x \cdot 0 = 0$
$x + x = x$	$x \cdot x = x$
$x + \bar{x} = 1$	$x \cdot \bar{x} = 0$

Synteza funkcji logicznych

Synteza nazywamy działanie zmierzające do uzyskania algebraicznego zapisu funkcji zdefiniowanego za pomocą tablicy wartości. Zbiór funkcji, który umożliwia syntezę dowolnie złożonych funkcji logicznych, nazywa się systemem funkcjonalnie pełnym. Jeśli wszystkie elementy tego zbioru są niezbędne, to system nazywamy minimalnym. Każdy inny system funkcjonalnie pełny jest nie minimalny.

Każdą funkcję logiczną przedstawić można w tzw. kanonicznej postaci alternatywnej lub równoważnej kanonicznej postaci koniunkcyjnej. Kanoniczna postać alternatywna jest sumą składników jedności K_i utworzonych dla stanów argumentów, przy których zmienna zależna jest jedynką. Kanoniczna postać koniunkcyjna jest iloczynem czynników zera D_i utworzonych dla stanów argumentów, przy których zmienna zależna jest zerem. Tablica 1 podaje czynniki zera i składniki jedności funkcji trzech argumentów i sposób ich numeracji.

Tab. 1. Składniki jedności K_i i czynniki zera D_i funkcji trzyargumentowych

Nr	X_1	X_2	X_3	D_i	K_i
0	0	0	0	$D_0 = x_1 + x_2 + x_3$	$K_0 = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}$
1	0	0	1	$D_1 = x_1 + x_2 + \overline{x_3}$	$K_1 = \overline{x_1} \cdot \overline{x_2} \cdot x_3$
2	0	1	0	$D_2 = x_1 + \overline{x_2} + x_3$	$K_2 = \overline{x_1} \cdot x_2 \cdot \overline{x_3}$
3	0	1	1	$D_3 = x_1 + \overline{x_2} + \overline{x_3}$	$K_3 = \overline{x_1} \cdot x_2 \cdot x_3$
4	1	0	0	$D_4 = \overline{x_1} + x_2 + x_3$	$K_4 = x_1 \cdot \overline{x_2} \cdot \overline{x_3}$
5	1	0	1	$D_5 = \overline{x_1} + x_2 + \overline{x_3}$	$K_5 = x_1 \cdot \overline{x_2} \cdot x_3$
6	1	1	0	$D_6 = \overline{x_1} + \overline{x_2} + x_3$	$K_6 = x_1 \cdot x_2 \cdot \overline{x_3}$
7	1	1	1	$D_7 = \overline{x_1} + \overline{x_2} + \overline{x_3}$	$K_7 = x_1 \cdot x_2 \cdot x_3$

Tab. 2. Tabela wartości przykładowej funkcji logicznej y trzyargumentowej

Nr	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	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Przykładowo wypisano postacie kanoniczne funkcji zdefiniowanej w tab. 2.

$$y = K_3 + K_5 + K_6 + K_7 = \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot x_2 \cdot x_3$$

$$y = D_0 \cdot D_1 \cdot D_2 \cdot D_4 = (x_1 + x_2 + x_3)(x_1 + x_2 + \overline{x_3})(x_1 + \overline{x_2} + x_3)(\overline{x_1} + x_2 + x_3)$$

Podany system numeracji czynników zera i składników jedności umożliwia stosowanie uproszczonego zapisu postaci kanonicznych funkcji. Np. dla funkcji podanej w tab.2 zapisujemy

$$y(x_1, x_2, x_3) = \sum (3,5,6,7) \quad \text{lub} \quad y(x_1, x_2, x_3) = \prod (0,1,2,4)$$

Wykorzystując prawa algebry Boole'a, najczęściej można zmniejszyć liczbę działań logicznych niezbędnych do zapisu danej funkcji. Operacja ta nazywa się minimalizacją funkcji.

Metoda Karnaugh'a

Do ułatwienia procesu przekształceń algebraicznych i minimalizacji funkcji logicznych służą tablice Karnaugh'a. Wnętrze tablicy wypełnia się wartościami funkcji y dla poszczególnych stanów argumentów x_i , wypisanych na obrzeżu. Stany argumentów są tak ułożone, aby składniki jedności (lub czynniki zera) utworzonych dla sąsiednich pól różniły się tylko jednym znakiem negacji, a więc nadają się do sklejania wg wzorów.

$$xy + x\overline{y} = x \quad \text{lub} \quad (x + y)(x + \overline{y}) = x$$

W celu ułatwienia wypełniania tablic na podstawie tablicy wartości funkcji, poszczególnym pola odpowiadające numerom stanu zostały ponumerowane.

Dla rozpatrywanej funkcji tablicę Karnaugh'a podano na rys.1

Tab. 3. Tabela Karnaugh'a dla podanej funkcji trzyargumentowej

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

W wyniku sklejania sąsiednich par oznaczonych kółeczkiem, różniących się tylko jednym znakiem negacji, otrzymamy system funkcjonalnie pełny minimalny:

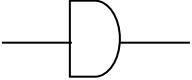
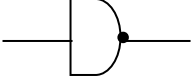
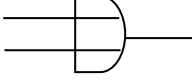
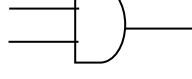
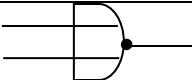
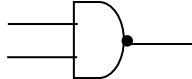
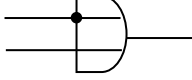
$$y = K_3 + K_5 + K_6 + K_7 = \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot x_2 \cdot x_3$$

$$= (K_3 + K_7) + (K_5 + K_7) + (K_6 + K_7) = x_2 \cdot x_3 + x_1 \cdot x_3 + x_1 \cdot x_2$$

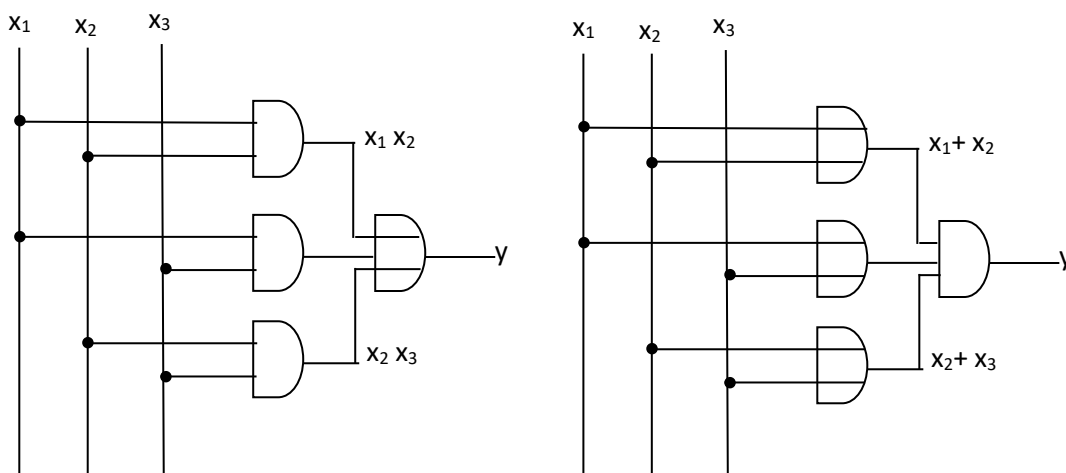
To samo dotyczy sklejania zer.

$$y = (D_0 \cdot D_1) \cdot (D_0 \cdot D_2) \cdot (D_0 \cdot D_4) = (x_1 + x_2)(x_1 + x_3)(x_2 + x_3)$$

Tab. 4. Symbole stosowane w schematach blokowych układów logicznych

Symbol elementu	Funkcja	Nazwa
	$y = x$	Powtórzenie
	$y = \overline{x}$	Negacja (NOT)
	$y = x_1 + x_2$	Alternatywa (OR)
	$y = x_1 \cdot x_2$	Koniunkcja (AND)
	$y = \overline{x_1 + x_2}$	Negacja alternatywy (NOR)
	$y = \overline{x_1 \cdot x_2}$	Negacja koniunkcji (NAND)
	$y = x_1 \rightarrow x_2 = \overline{x_1} + x_2$	Implikacja

Wykorzystując symbole elementów logicznych podanych w tabeli 4, możemy narysować schemat blokowy układu realizującego podaną funkcję logiczną y (rys. 2).

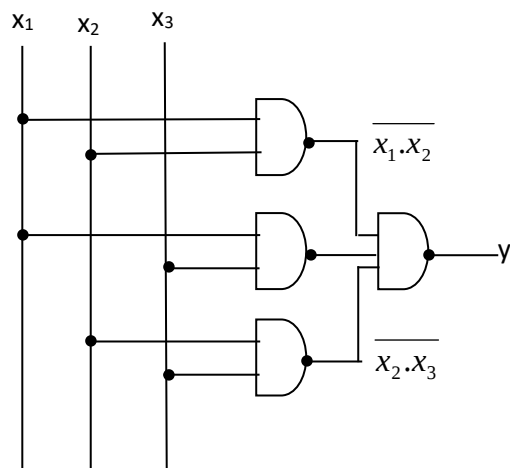


Rys.2. Schemat blokowy budowy układu logicznego: a) zminimalizowanego metodą sklejaną jedynek, a) zminimalizowanego metodą sklejaną zer.

W przypadku stosowania wyłącznie jednego typu bramek logicznych jak NAND lub NOR do budowy układu logicznego, postać zminimalizowaną funkcji y trzeba przekształcić za pomocą prawa Morgan'a. Np.:

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

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

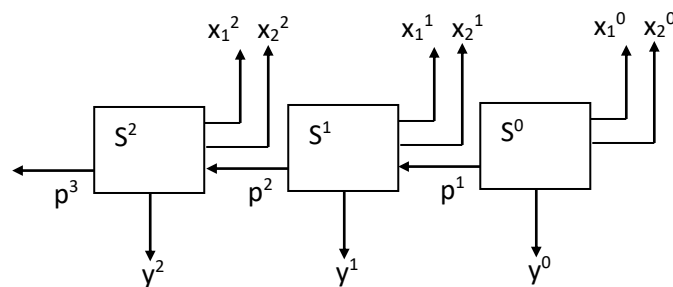


Rys.3. Schemat blokowy układu logicznego zbudowanego z bramek NAND.

Jedną z metod maszynowego dodawania liczb binarnych jest sumowanie równoległe. Sumator wykonujący operację dodawania składa się z połączonych szeregowo, tzw. sumatorów jednopozycyjnych, oznaczonych na rysunku 4 kwadratami.

Sygnałami wejściowymi sumatora jednopozycyjnego są wartości cyfr danej pozycji składników sumy x_1 , x_2 oraz przeniesienie p^i z pozycji niższej. Sygnałami wyjściowymi są sygnał sumy danej pozycji oraz sygnał przeniesienia do pozycji wyższej. Na rys.4 naniesiono dla przykładu stany sygnałów odpowiadające dodawania liczb:

Składnik	$x_1 =$	11
Składnik	$x_2 =$	01
Suma	$y =$	100
Przeniesienie	$p =$	11



Rys.4. Schemat blokowy równoległego sumatora dwójkowego.

Wartość sygnałów wyjściowych sumatora zależą od wartości sygnałów wejściowych. Tabela 5 przedstawia wartości sygnałów wyjściowych sumatora jednopozycyjnego dla wszystkich stanów sygnałów wejściowych.

Tab. 5. Tabela wartości dla sumatora jednopozycyjnego

x_1	x_2	p^{n-1}	y^n	p^n
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Tab. 6. Tabela Karnaugh dla funkcji y^n

$\begin{matrix} x_1, x_2 \\ p^{n-1} \end{matrix}$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

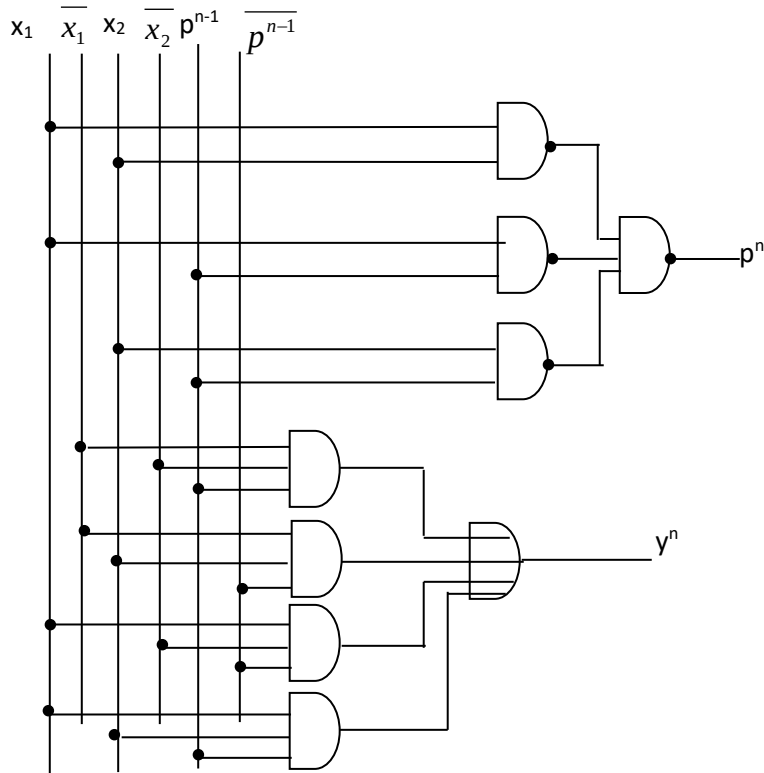
Tab. 7. Tabela Karnaugh dla funkcji p^n

$\begin{matrix} x_1, x_2 \\ p^n \end{matrix}$	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$y^n = \overline{x_1} \cdot \overline{x_2} \cdot p^{n-1} + \overline{x_1} \cdot x_2 \cdot \overline{p^{n-1}} + x_1 \cdot \overline{x_2} \cdot \overline{p^{n-1}} + x_1 \cdot x_2 \cdot p^{n-1}$$

$$p^n = x_1 \cdot x_2 + x_2 \cdot p^{n-1} + x_1 \cdot p^{n-1} = (\overline{x_1 \cdot x_2}) (\overline{x_2 \cdot p^{n-1}}) (\overline{x_1 \cdot p^{n-1}})$$

Sumator jednopozycyjny realizuje jednocześnie dwie funkcje logiczne. Jego schemat przedstawiono na rys.



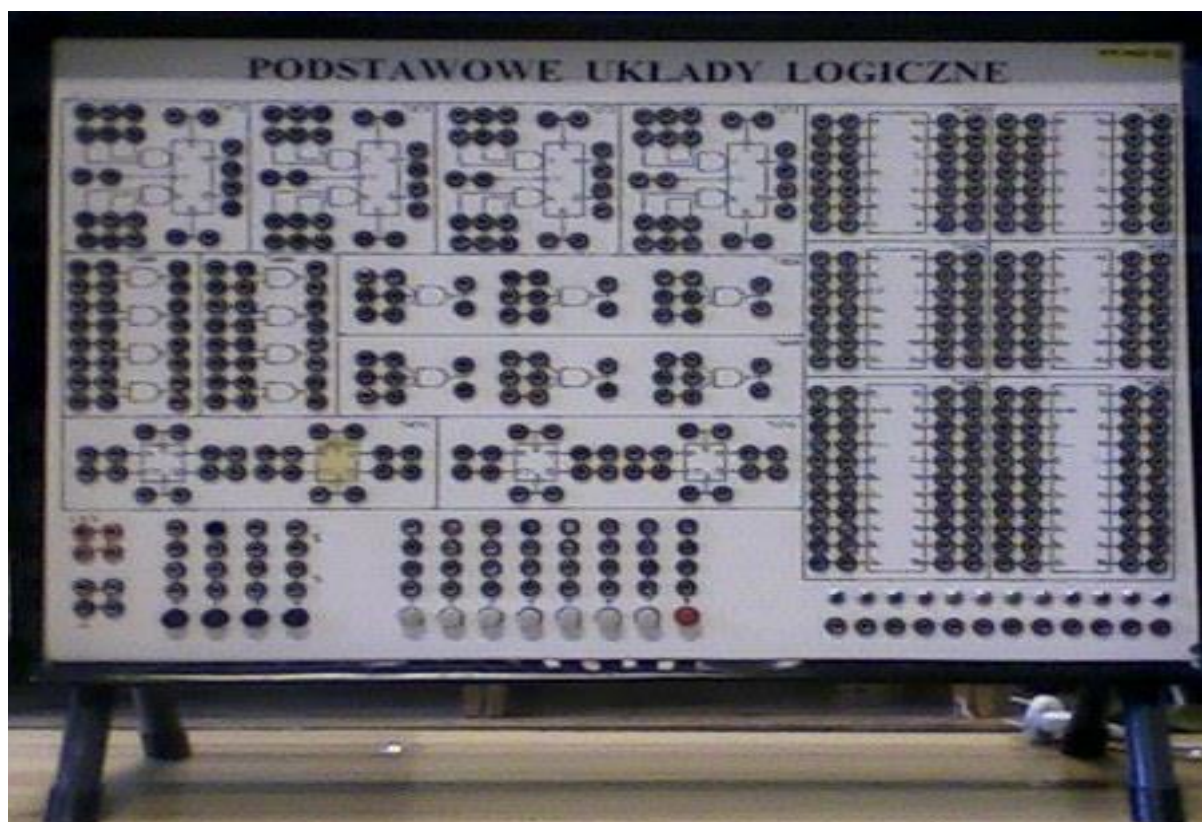
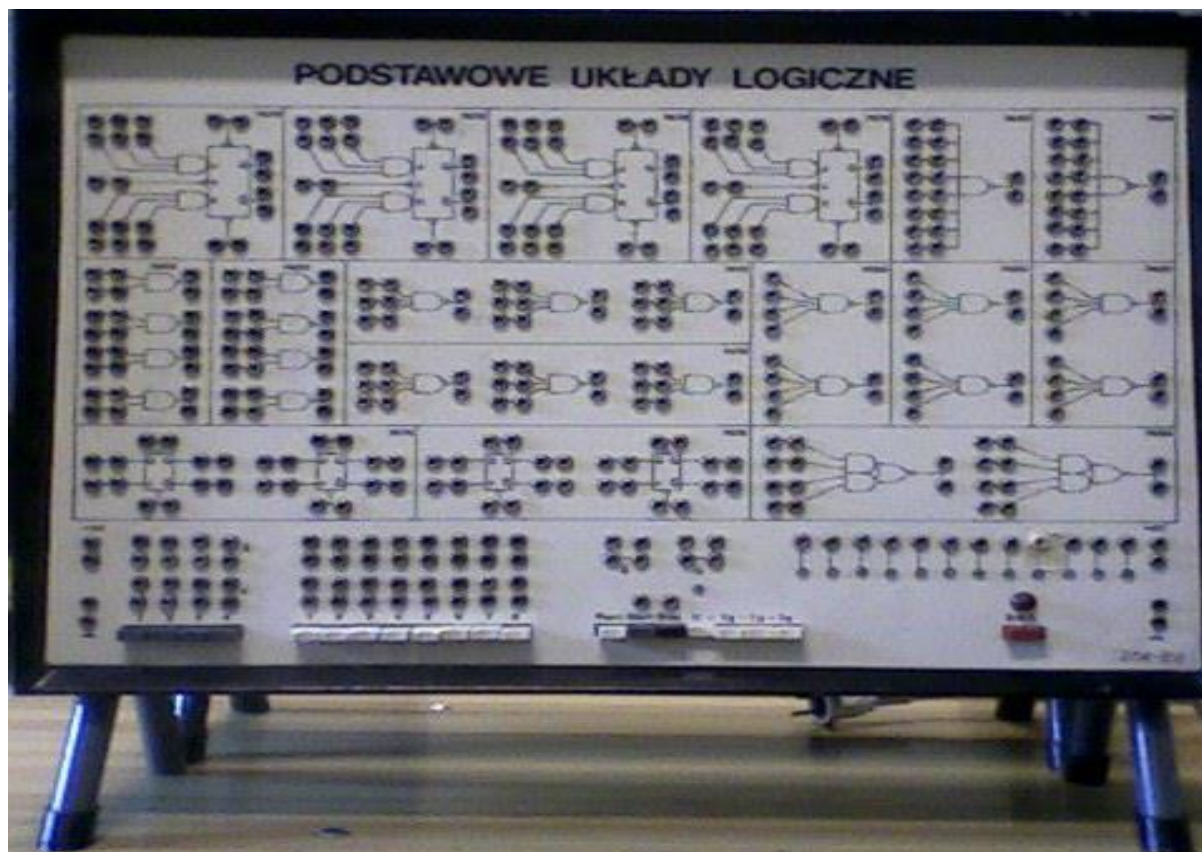
Rys.5. Schemat budowy sumatora jednopozycyjnego.

3. Stanowisko pomiarowe:

Stanowisko laboratoryjne stanowią dwie tablice ze zmontowanymi bramkami logicznymi dostępnymi na rynku. Są to bramki typu:

- negacja (NOT),
- negacja koniunkcji (NAND) z 2, 3, 4, 8-ma wejściami,
- przerzutniki synchroniczne typu D, R-S, J-K,
- Układy scalone

Wyposażone one także są w przyciski generujące sygnały jednobitowe, lampki diodowe sygnalizujące i zegar systemowy taktujący ze stałą nastawialną częstotliwością



Rys.6. Stanowiska badań układów logicznych.

4. Przebieg ćwiczenia

- zbudować tabelę wartości funkcji logicznej układu sterowania podanego przez prowadzącego,
- zbudować tabelę Karnaugh'a i przeprowadzić minimalizację przez sklejenia,
- przekształcić minimalizowaną funkcję z wykorzystaniem bramek NAND za pomocą wzoru Morgan'a,
- narysować schemat i zbudować otrzymany układ na tablicy logicznej,
- przetestować działania układu sterowania zgodnie z podanym zadaniem.

5. Sprawozdanie z ćwiczenia

W sprawozdaniu należy podać:

Schemat połączeń zbudowanego układu z oznaczeniem elementów wg symboliki znormalizowanej wraz z krótkim opisem ćwiczenia.

6. Pytania kontrolne

Logiczne układy sterowania – definicje i rodzaje
Podstawowe działania na funkcjach logicznych – aksjomaty.
Funkcjonalnie układy pełne i minimalne. Postać kanoniczna funkcyjna.
Metoda minimalizacji Karnaugh'a.
Podstawowe bramki logiczne i ich symbole wg normy.

Literatura

1. Findeisen W. : Technika regulacji automatycznej.
2. Traczy W. : Układy cyfrowe automatyki.
3. Wasilewski L. : Laboratorium podstaw techniki cyfrowej.