WOJSKOWA AKADEMIA TECHNICZNA
im. Jarosława Dąbrowskiego

Ćwiczenie rachunkowe

Układy kombinacyjne i sekwencyjne

Podstawy automatyki

ZAiUL WML WAT
Warszawa 2017

1 Cel ćwiczenia rachunkowego

Podczas ćwiczenia poruszane będą następujące zagadnienia:

- modelowanie i synteza kombinacyjnych układów przełączających;
- minimalizacja funkcji przełączającej;
- projektowanie układów sterowania sekwencyjnego.

Celem ćwiczenia jest zdobycie umiejętności praktycznej realizacji powyższych zagadnień.

2 Wymagania wstępne

Przed rozpoczęciem ćwiczeń student zobowiązany jest do zapoznania się z treścią niniejszej instrukcji. W szczególności istotne jest posiadanie wiedzy teoretycznej z zakresu poruszanego podczas ćwiczenia rachunkowego. Ponadto student zobowiązany jest prześledzić ze zrozumieniem wszystkie zamieszczone przykłady, aby wiedzieć w jaki sposób rozpocząć rozwiązywanie zadań podczas ćwiczeń. W przypadku posiadania wątpliwości po zapoznaniu się z treścią instrukcji w celu ich wyjaśnienia zaleca się konsultacje się z prowadzącym przed terminem ćwiczeń rachunkowych.

3 Układy kombinacyjne

3.1 Pojęcia podstawowe

System dwójkowy (binarny) jest sposobem zapisu liczb którego podstawą jest liczba 2. Zapis dokonuje się za pomocą symboli „*0*” i „*1*”. Ogólny zapis liczb w systemie binarnym opisuje zależność:

$$L_2 = \sum_{i=n-1}^{0} b_i \cdot 2^i \text{ , gdzie } b_i \in \{0, 1\}$$

Jeśli zmiennej x_i przypiszemy „*1*”, a zmiennej $\overline{x_i}$ przypiszemy „*0*”, to w prosty sposób uzyskuje się reprezentacje binarną.

Układ przełączający - urządzenie sterujące, zbudowane z elementów, które mogą znajdować się w dwóch różnych stanach, określonych jako *stan spoczynkowy* i *stan wzbudzony*. Elementy przełączające, z których zbudowany jest układ przełączający, łączą lub przerywają przepływ energii w obwodzie, np.: przekaźniki i przełączniki elektryczne załączają przepływ energii elektrycznej, rozdzielacze pneumatyczne zmieniają kierunek przepływu sprężonego powietrza, natomiast rozdzielacze hydrauliczne sterują kierunkiem przepływu płynu hydraulicznego.

Funkcja przełączająca $f(x_1, x_2, \ldots, x_n)$ to takie odwzorowanie, które dla kombinacji argumentów x_1, x_2, ..., x_n przyjmujących wartości *„0”* lub *„1”* przyporządkowuje rozwiązanie ze zbioru $\{0, 1\}$. Czynnik jedynki (konstytuanta zera) M_i jest taką funkcją przełączającą, która przyjmuje wartość „0” tylko dla kombinacji wartości zmiennych, wyrażonej w postaci sumy.

Zupełnie normalna postać sumy (postać kanoniczna dysjunkcyjna) funkcji przełączającej - suma tych czynników jedynki (konstytuant jedynki), które są równe *„1”* dla tych samych kombinacji wartości argumentów co zadana funkcja. Postać ogólną wyraża poniższa zależność:

$$F = \sum_{i=1}^{2^n} m_i \cdot f_i$$

gdzie: f_i – iloczyn argumentów;
n – liczba argumentów.

Zupełnie normalna postać iloczynu (postać kanoniczną koniunkcyjna) funkcji przełączającej - suma tych czynników zera (konstytuant zera), które są równe *„0”* dla tych samych kombinacji wartości argumentów co zadana funkcja. Postać ogólną wyraża zależność:

$$F' = \prod_{i=1}^{2^n} \left(M_i + f_i\right)$$

Faktoryzacja (minimalizacja) ma na celu uzyskanie najmniejszej złożoności układu. Realizuje się to poprzez rezygnację z postaci normalnej funkcji i zastąpienie jej postacią po minimalizacji.

3.2 Podstawowe funkcje przełączające

Funkcja przełączająca f(x_1, x_2, …, x_n) n – zmiennych to odwzorowanie:

$$f : D \to \{0, 1\}^n$$

gdzie: $D \subset \{0, 1\}^n$ – uporządkowana piątka *({0, 1}, 0, 1, •, +)* jest algebrą Boole’a

Algebra Boole’a jest systemem umożliwiającym opis układów przełączających. W notacji formalnej algebrę Boole’a zapisujemy jako uporządkowaną piątkę *A=(X, 0, 1, •, +)*, gdzie:

$X=\{x_1, x_2, \ldots, x_n\}$ – zbiór argumentów przyjmujących wartości 0 lub 1;
0 – element neutralny operacji dysjunkcji (sumy);
1 – element neutralny operacji koniunkcji (iloczynu);
• – symbol operacji koniunkcji;
\+ – symbol operacji dysjunkcji.

Poniżej przytoczono kilka najważniejszych funkcji przełączających nazywanych funkcjami Boole’a:

$f(\mathrm{x}_1) = \mathrm{x}_1$	Zmienna x_1
$f(\mathrm{x}_1) = \bar{x}_1$	Negacja zmiennej x_1
$f(\mathrm{x}_1, \mathrm{x}_2) = \mathrm{x}_1 \bullet \mathrm{x}_2$	Koniunkcja, iloczyn zmiennych x_1 i x_2 („i”, „AND”)
$f(\mathrm{x}_1, \mathrm{x}_2) = \mathrm{x}_1 + \mathrm{x}_2$	Dysjunkcja, suma zmiennych x_1 i x_2 („lub”, „OR”)
$f(\mathrm{x}_1, \mathrm{x}_2) = \bar{x}_1\mathrm{x}_2 \bullet \bar{x}_2\mathrm{x}_1$	Suma modulo 2, różni symetryczna („albo”, „XOR”)
$f(\mathrm{x}_1, \mathrm{x}_2) = \bar{x}_1 \bullet \bar{x}_2$	Operacja Pierce’a („NOR”)
$f(\mathrm{x}_1, \mathrm{x}_2) = \bar{x}_1 + \bar{x}_2$	Operacja Sheffer’a („NAND”)

Po dokonaniu założeń:

$$\forall x_i \in X \quad dla \quad i = 1, \dots, n$$

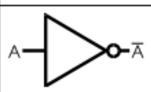
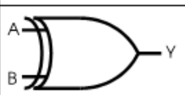
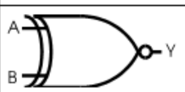
$$\forall x_i \in X \ \exists \ \bar{x}_i \in X \quad dla \quad i = 1, \dots, n$$

gdzie: $\bar{x}_i$ - zanegowany argument x_i; spełnione są następujące aksjomaty algebry Boole'a:

Lp	dysjunkcja(suma)	koniunkcja (iloczyn)	nazwa (opis)
0	$x_1+x_2 \in X$	$x_1 \bullet x_2 \in X$	
1	$\bar{0}=1$	$\bar{1}=0$	
2	$x_1+0 = x_1$	$x_1 \bullet 1 = x_1$	działania na elementach neutralnych
3	$x_1+1 = 1$	$x_1 \bullet 0 = 0$	działania na elementach neutralnych
4	$x_1+ x_1 = x_1$	$x_1 \bullet x_1 = x_1$	prawa idempotentności
5	$x_1+x_2 = x_2+x_1$	$x_1 \bullet x_2 = x_2 \bullet x_1$	prawa przemienności
6	$x_1 \bullet (x_2+x_3) = x_1 \bullet x_2+x_1 \bullet x_3$	$x_1+x_2 \bullet x_3 = (x_1+x_2) \bullet (x_1+x_3)$	prawa rozdzielności
7	$x_1 +\bar{x}_1 = 1$	$x_1 \bullet \bar{x}_1 = 0$	prawa dopełnienia
8	$x_1+(x_2+x_3) = (x_1+x_2)+x_3$	$x_1 \bullet (x_2 \bullet x_3) = (x_1 \bullet x_2) \bullet x_3$	prawo łączności
9	$\bar{\bar{x}}_1 = x_1$		podwójna negacja
10	$\overline{x_1 + x_2} = \bar{x}_1 \bullet \bar{x}_2$	$\overline{x_1 \bullet x_2} = \bar{x}_1 + \bar{x}_2$	prawa de Morgana
11	$x_1+ x_1 \bullet x_2 = x_1$	$x_1 \bullet (x_1+ x_2) = x_1$	prawa pochłaniania
12	$x_1 +\bar{x}_1 \bullet x_2 = x_1+x_2$	$x_1 \bullet (x_1 +\bar{x}_2) = x_1 \bullet x_2$	prawa pochłaniania

Jeśli $D^n = \{0, 1\}^n$ to funkcję przełączającą nazywamy ***zupełną lub w pełni określoną***. W przypadku, gdy istnieją kombinacje argumentów, że funkcja może przyjąć wartość *„0”* lub *„1”* (funkcja nie jest określona, oznacza się symbolem ∅), to nazywamy ją ***niezupełną***.

Poniżej przedstawiono w formie tabeli najważniejsze funkcje logiczne w postaci symboli logiczny, wyrażeń algebraiczny i wartości tych funkcji dla kombinacji argumentów A, B:

FUNKCJA LOGICZNA	SYMBOL LOGICZNY	WYRAŻENIE ALGEBRAICZNE	TRUTH TABLE Inputs A	B	Output Y
AND		$A \cdot B = Y$	0 0 1 1	0 1 0 1	0 0 0 1
OR		$A + B = Y$	0 0 1 1	0 1 0 1	0 1 1 1
NOT *(Inverter)*		$A = \bar{A}$	0 1		1 0
NAND		$\overline{A \cdot B} = Y$	0 0 1 1	0 1 0 1	1 1 1 0
NOR		$\overline{A + B} = Y$	0 0 1 1	0 1 0 1	1 0 0 0
XOR		$A \oplus B = Y$	0 0 1 1	0 1 0 1	0 1 1 0
XNOR		$\overline{A \oplus B} = Y$	0 0 1 1	0 1 0 1	1 0 0 1

3.3 Sposoby opisu układów kombinacyjnych

Istnieją liczne sposoby opisu układów kombinacyjnych. Na potrzeby realizacji ćwiczenia omówione zostaną trzy z nich: **opis słowny, tablica wartości funkcji** i **zależność matematyczna.** Opis słowny jest najczęstszą formą zadawania układów przełączających, który polega na przyporządkowaniu sygnałom wejściowym *X* sygnałów wyjściowych *Y*. Tablica wartości funkcji (zależności) w postaci ciągów zero-jedynkowych jest prostszą formą opisu układu. Określa ona wartości sygnałów *X* i odpowiadające im sygnały *Y* :

stan	x_1	x_2	x_3	y
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Często wykorzystywaną forma opisu układów przełączających jest zależność matematyczna. Zależność ta może mieć postać zupełnie normalną (postać kanoniczną) lub uproszczoną. Wyróżnia się postać dysjunkcyjną (sumy) oraz koniunkcyjną (iloczynu):

	postać kanoniczna	postać uproszczona
dysjunkcja (suma)	$y = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_2x_3 + x_1\bar{x}_2x_3 + x_1x_2x_3$	$y = \sum(0,1,5,7)_{x_1x_2x_3}$
koniunkcja (iloczyn)	$y = (\bar{x}_1 + x_2 + x_3)(x_1 + \bar{x}_2 + x_3)(\bar{x}_1 + \bar{x}_2 + \bar{x}_3)(x_1 + \bar{x}_2 + \bar{x}_3)$	$y = \prod(2,3,4,6)_{x_1x_2x_3}$

Wszystkie powyższe sposoby opisu dotyczą takiego samego układu kombinacyjnego

3.4 Minimalizacja funkcji przełączających

Celem minimalizacji funkcji przełączających jest zmniejszenie liczby elementów, a tym samym kosztów urządzenia. Dodatkowo mniejsza liczba połączeń i elementów zwiększa trwałość i niezawodność urządzenia. Istnieje wiele sposobów minimalizacji funkcji przełączających, na potrzeby ćwiczenia laboratoryjnego omówione zostaną trzy z nich:

1. metoda przekształceń formalnych;
2. minimalizacja z wykorzystaniem tablicy Karnaugha;
3. metoda Quine'a – McCluskeya;

Powyższe metody zostaną omówione dla funkcji przełączającej o postaci kanonicznej sumy:

$$y = \overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4} + \overline{x_1}x_2\overline{x_3}\,\overline{x_4} + \overline{x_1}\,\overline{x_2}x_3\overline{x_4} + \overline{x_1}x_2x_3\overline{x_4} + \overline{x_1}\,\overline{x_2}x_3x_4 + \overline{x_1}x_2x_3x_4$$

3.4.1 Metoda przekształceń formalnych

Metoda przekształceń formalnych stosowana jest w przypadku, gdy funkcja dana jest w postaci wyrażenia algebraicznego. Wykorzystuje się wtedy aksjomaty i prawa algebry Boole'a.

Wyłączając $\overline{x_1}$ uzyskujemy postaci kanonicznej sumy otrzymujemy:

$$y=\overline{x_1}\left(\overline{x_2}\,\overline{x_3}\,\overline{x_4}+x_2\overline{x_3}\,\overline{x_4}+\overline{x_2}x_3\overline{x_4}+x_2x_3\overline{x_4}+\overline{x_2}x_3x_4+x_2x_3x_4\right)$$

W kolejnym kroku wyłączając przed nawias $(\overline{x_2}+x_2)$ otrzymujemy

$$y=\overline{x_1}\left[\left(\overline{x_2}+x_2\right)\left(\overline{x_3}\,\overline{x_4}+x_3\overline{x_4}+x_3x_4\right)\right]$$

Korzystając z prawa dopełnienia $(x+\overline{x}=1)$ pomijamy człon $(\overline{x_2}+x_2)$, a następnie wyłączamy $\overline{x_4}$

$$y=\overline{x_1}\left[\left(\overline{x_3}+x_3\right)\overline{x_4}+x_3x_4\right] \tag{1}$$

Ponownie korzystając z prawa dopełnienia, a następnie prawa działania na elementach neutralnych (x+1=1), otrzymujemy:

$$\overline{x_3}+x_3=x_3+1 \tag{2}$$

Po podstawieniu zależności (2) do równania (1) otrzymujemy funkcję:

$$y=\overline{x_1}\left[\left(x_3+1\right)\overline{x_4}+x_3x_4\right]$$

Przegrupowując wyrazy, otrzymujemy:

$$y=\overline{x_1}\left[\left(\overline{x_4}+x_4\right)x_3+\overline{x_4}\right]$$

W celu wyznaczenia zminimalizowanej funkcji ponownie korzystamy z prawa dopełnienia $\overline{x_4}+x_4=1$, otrzymując końcową funkcję w postaci:

$$y=\overline{x_1}\left(x_3+\overline{x_4}\right)=\overline{x_1}x_3+\overline{x_1}\,\overline{x_4}$$

3.4.2 Metoda tablicy Karnaugha

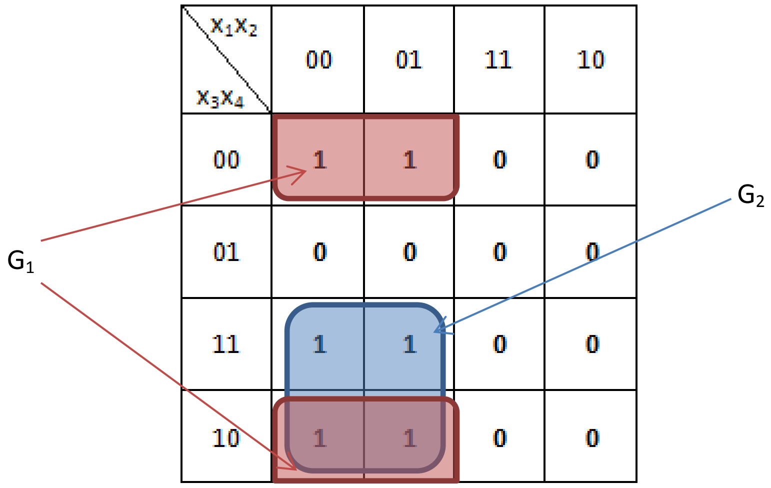
Metoda tablicy Karnaugha należy do grupy najszybszych metod minimalizacji funkcji przełączających małej liczby zmiennych co wynika z dużej komplikacji samego zapisu następującej wraz ze wzrostem ilości zmiennych. Upraszczając funkcję przełączającą przy wykorzystaniu tablicy Karnaugha, należy pamiętać o następujących zasadach:

1. wiersze i kolumny tablicy Karnaugha opisane są w kodzie Greya, tzn. każdy kolejny wiersz i kolumna różnią się od siebie o negację jednej zmiennej;
2. zakreślając jedynki (zera), tworzy się grupy o liczbie elementów 2^N, gdzie n ϵ **N** ;
3. zawsze zakreśla się grupy z największą możliwą ilością jedynek (zer), przy czym należy pamiętać o możliwości sklejenia ze sobą krawędzi równoległych tablicy;
4. grupy mogą posiadać części wspólne;
5. liczba grup jedynek (zer) odpowiada liczbie składników sumy (iloczynu) poszukiwanej funkcji;
6. w przypadku kiedy istnieje możliwość zakreślenia grup na kilka sposobów, arbitralnie wybiera się jeden z nich;
7. dana grupa reprezentuje iloczyn (sumę) tych zmiennych, które nie zmieniają swojej wartości;
8. w przypadku, gdy funkcja przełączająca posiada elementy o wartości nieokreślonej elementy te wpisujemy do tabeli, wprowadzając dla nich specjalne oznaczenie, np. – a następnie wykorzystujemy lub pomijamy, w zależności od potrzeby przy tworzeniu grup (patrz punkt 2).

Stosując zasadę **1** tworzymy tablice Karnaugha, wypełniając ją jedynkami dla elementów funkcji, a pozostałe pola uzupełniamy zerami:

x_1x_2 \ x_3x_4	00	01	11	10
00	**1**	**1**	**0**	**0**
01	**0**	**0**	**0**	**0**
11	**1**	**1**	**0**	**0**
10	**1**	**1**	**0**	**0**

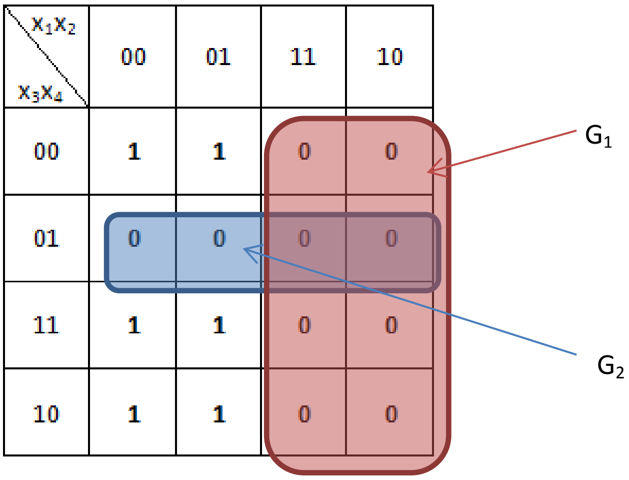
W kolejnym kroku chcąc otrzymać postać dysjunkcyjną zminimalizowanej funkcji tworzymy dwie grupy zawierające po cztery elementy „1" (według zasad **2,3,4**) :



Postępując według wytycznych **5, 6** i **7** odczytujemy zminimalizowaną postać funkcji przełączającej. W czteroelementowej grupie G_1 wartości zmiennych $\overline{x_1}$ i $\overline{x_4}$ nie ulegają zmianie, ponieważ zakreślono grupę jedynek. Funkcja ta przyjmuje postać iloczynu $G_1 = \overline{x_1}\overline{x_4}$. Dla grupy G_2 niezmienne wartości przyjmują parametry $\overline{x_1}$ i x_3, więc grupa ta przyjmuje postać $G_2 = \overline{x_1}x_3$. Po zsumowaniu z grupą G_1 otrzymamy ostatecznie poszukiwaną przez nas funkcję w postaci dysjunkcyjnej:

$$y = \overline{x_1}x_3 + \overline{x_1}\,\overline{x_4}$$

W celu wyznaczenia zminimalizowanej funkcji w postaci koniunkcyjnej należy zacząć od początku wypisywanie tablicy lub też skorzystać z tablicy wypisanej dla postaci dysjunkcyjnej, zakreślając w tym przypadku grupy zer:



Postępując według wytycznych **5, 6** i **7** otrzymujemy następujące grupy $G_1 = \bar{x}_1$ (ponieważ jedynym niezmiennym parametrem w grupie jest x_1 i przyjmuje on wartość jeden) oraz $G_1 = x_3\bar{x}_4$. Po pomnożeniu obu grup otrzymamy zminimalizowaną funkcję w postaci koniunkcyjnej:

$$y = \overline{x_1}\left(x_3 + \overline{x_4}\right)$$

3.4.3 Metoda Quine

Metoda Quine'a – McCluskeya jest zwykle stosowana w przypadku minimalizacji funkcji wielu zmiennych, ponieważ wraz ze wzrostem ich liczby wzrasta też jej efektywność w stosunku do pozostałych metod. W celu minimalizacji funkcji przełączającej tą metodą postępujemy w następujący sposób:

1. minimalizację rozpoczynamy od zapisania w kolejności rosnącej elementów funkcji przełączającej, dla których funkcja ta przyjmuje wartość jeden (zero);
2. w kolejnym kroku uporządkowujemy elementy poprzez zapisanie ich w grupach zawierających identyczne ilości jedynek (zer), przy czym każda kolejna grupa powinna zawierać więcej jedynek (zer) od poprzedniej;
3. w celu znalezienia implikantów prostych porównujemy każdy element w grupie z każdym elementem w grupie sąsiedniej zawierającej jedną jedynkę (zero) więcej;
4. jeżeli elementy różnią się miedzy sobą tylko jednym indeksem, zaznaczamy je oba, a w miejscu tego indeksu wstawiamy kreskę i przepisujemy nowo powstały element do następnej kolumny;
5. procedurę **4** powtarzamy z każdą nową kolumną aż do pozostania elementów niemożliwych do uproszczenia, gdzie elementy nieoznaczone stanowią poszukiwane implikanty proste;
6. tworzymy tabelę, w której w pierwszym wierszu wpisujemy kolejne elementy funkcji przełączającej, a w kolumnie implikanty proste;
7. jeżeli element funkcji spełnia implikant, to na ich przecięciu w tabeli wstawiamy *x*;
8. zminimalizowaną funkcję przełączającą tworzymy z implikantów prostych, które pokrywają wszystkie elementy zadanej funkcji przełączającej;
9. w przypadku gdy w funkcji występują elementy z wartością nieokreśloną wykorzystujemy je przy poszukiwaniu implikantów prostych, a pomijamy przy tworzeniu tabeli (patrz ***6***).

Zapisujemy elementy funkcji przełączającej, dla której przyjmuje ona wartość jeden w postaci binarnej (podpunkt **1**):

0	$\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$	0000
2	$\bar{x}_1x_2\bar{x}_3\bar{x}_4$	0100
4	$\bar{x}_1\bar{x}_2x_3\bar{x}_4$	0010
6	$\bar{x}_1x_2x_3\bar{x}_4$	0110
12	$\bar{x}_1\bar{x}_2x_3x_4$	0011
14	$\bar{x}_1x_2x_3x_4$	0111

Wpisane elementy sortujemy i zapisujemy w grupach (podpunkt **2**.):

0000

0100
0010

0110
0011

0111

Przeszukujemy grupy w celu znalezienia implikantów prostych (podpunkt **3** i **4**). Na przykład elementy „1” i „2” różnią się sobą tylko jednym indeksem – otrzymamy wtedy nowy wyraz w postaci „0-00”, który następnie zapisujemy do nowej kolumny, zaznaczając jednocześnie elementy „1” i „2”:

(1) 0000 x

(2) 0100 x
(3) 0010 x

(4) 0110 x
(5) 0011 x

(6) 0111 x

(1,2) 0-00
(1,3) 00-0

(2,4) 01-0
(3,4) 0-10
(3,5) 001-

(4,6) 011-
(5,6) 0-11

Procedurę **4** powtarzamy z nowo otrzymaną kolumną elementów, otrzymując:

(7) 0-00 x
(8) 00-0 x

(9) 01-0 x
(10) 0-10 x
(11) 001- x

(12) 011- x
(13) 0-11

(8,9) (7,10) 0--0

(11,12) 0-1-

Ponieważ w ostatniej kolumnie znajdują się już tylko implikanty proste, wypisujemy je łącznie z pozostałymi nie oznaczonymi elementami (np. element 13), otrzymując:

$$0 \cdot 11 = \bar{x}_1 x_3 x_4$$
$$0 \cdot 1 \cdot = \bar{x}_1 x_3$$
$$0 \cdot \cdot 0 = \bar{x}_1 \bar{x}_4$$

Po wyznaczeniu implikantów wypełniamy tabelę postępując według kroków **6, 7, 8**.

	$\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$	$\bar{x}_1 x_2 \bar{x}_3 \bar{x}_4$	$\bar{x}_1\bar{x}_2 x_3 \bar{x}_4$	$\bar{x}_1 x_2 x_3 \bar{x}_4$	$\bar{x}_1\bar{x}_2 x_3 x_4$	$\bar{x}_1 x_2 x_3 x_4$
$\bar{x}_1 x_3 x_4$					X	X
$\bar{x}_1 \bar{x}_4$	X	X	X	X		
$\bar{x}_1 x_3$			X	X	X	X

Dla wybranych implikantów zminimalizowana funkcja przyjmuje postać końcową:

$$y = \overline{x_1}\left(x_3 + \overline{x_4}\right) = \overline{x_1}x_3 + \overline{x_1}\,\overline{x_4}$$

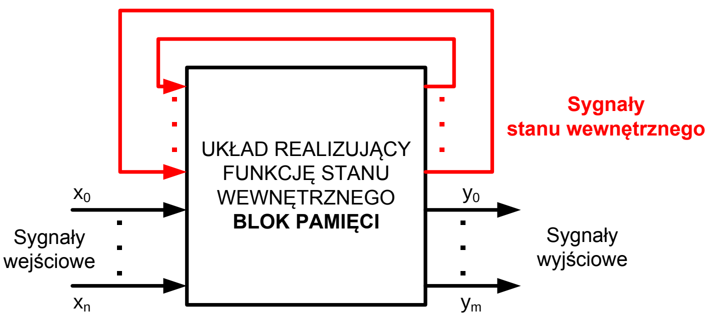
4 Układy sekwencyjne

Układ cyfrowy, w którym aktualny stan wyjść zależy nie tylko od aktualnego stanu wejść, ale również zależy od stanu w którym układ znajdował się wcześniej nazywamy układem sekwencyjnym lub układem z pamięcią (można spotkać określenie z pamięcią stanu).

Przykładem elementarnego układu sekwencyjnego jest układ przełączający, który służy do zaświecenia lampy stołowej z ręcznie uruchamianym przyciskiem o jednym tzw. położeniu stabilnym. Jeśli lampa nie świeci się, to naciśnięcie przycisków powoduje jej zaświecenie. W przypadku gdy lampa jest włączona, to naciśniecie przycisku powoduje wyłączenie lampy. Przyciśniecie przycisku powoduje włączenie lub wyłączenie lampy zależnie od tego czy wcześniej była włączona czy wyłączona.

W rozbudowanych układach sekwencyjnych zależności między wejściami i wyjściami stają się niejednoznaczne, co może prowadzić do tego, że tym samym wektorom wejściowym mogą opowiadać różne wektory wyjść. Wynika to z podstawowej właściwości układu sekwencyjnego mówiącej o tym, że wartość na wyjściu zależy od „historii" układu - „pamięć stanu".

Pamięć realizowana jest przez wprowadzenie sprzężenia zwrotnego. Informacja o stanie, w którym znajduje się układ jest przekazywana na wejście układu. Układ ten nazywa się blokiem pamięci. Blok pamięci odpowiedzialny jest za realizację funkcji stanu układu (Rys.1).



Rys. 1. Blok pamięci układu sekwencyjnego.

W układzie automatycznej regulacji występuje sprzężenie zwrotne do wyznaczenia uchybu regulacji. W układzie sekwencyjnym nie rozróżnia się dodatniego i ujemnego sprzężenia zwrotnego. Sprzężnie zwrotne w układzie sekwencyjnym, to rozszerzenie wektora wejść o dodatkowe elementy, którymi są wyjścia bloku pamięci.

4.1 Klasyfikacja układów sekwencyjnych

Układy sekwencyjne dzielimy na:

- układy sekwencyjne asynchroniczne,
- układy sekwencyjne synchroniczne.

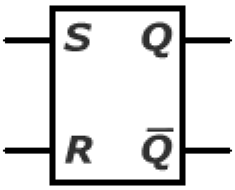
W układach sekwencyjnych asynchronicznych zmiana stanu wewnętrznego następuje bezpośrednio i wyłącznie pod wpływem zmiany stanu wejść. Nowy stan wewnętrzny ustala się po pewnym czasie t określonym przez opóźnienie elementów, z których zbudowany jest układ realizujący funkcję δ.

W układach synchronicznych zmiana stanu wewnętrznego może następować tylko w ściśle określonych chwilach czasu, wyznaczonych przez sygnał doprowadzony do specjalnego wejścia układu. Wejście to, nazywane jest taktującym lub zegarowym i oznaczane jest literą C (ang. clock). Stan wejść oddziałuje na stan wewnętrzny automatu tylko w chwilach czasu, gdy wejście zegarowe jest aktywne. Zmiana stanu wejść, gdy wejście zegarowe jest nieaktywne nie powoduje zmiany stanu wewnętrznego układu.

4.1.1 Przerzutnik asynchroniczny sr

Najprostszymi układami sekwencyjnymi są przerzutniki asynchroniczne (Rys. 2). Przerzutnik tego typu posiada dwa wejścia:

- wejście wpisujące set (s),
- wejście zerujące reset (r).



Rys. 2. Przerzutnik asynchroniczny.

Układ posiada wyjście Q oraz wyjście zanegowane $\overline{Q}$. Przerzutnik realizuje funkcję zgodnie z tabelą na rysunku 3.

a)

s	r	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	-

b)

Q(t)→Q(t+1)		s	r
0	0	0	-
0	1	1	0
1	0	0	1
1	1	-	0

Rys. 3. Tabela wejść przerzutnika sr (a) i tabela przejść przerzutnika sr (b).

4.2 Automat skończony

Modelem układu sekwencyjnego jest automat skończony. Rozróżnia się dwa podstawowe typy automatów:

- automat Mealy'ego,
- automat Moore'a.

4.2.1 Automat Mealy'ego

Automatem skończonym Mealy'ego (Rys. 4) nazywać będziemy układ:

$$M=<X,S,Y,\delta,\lambda>$$

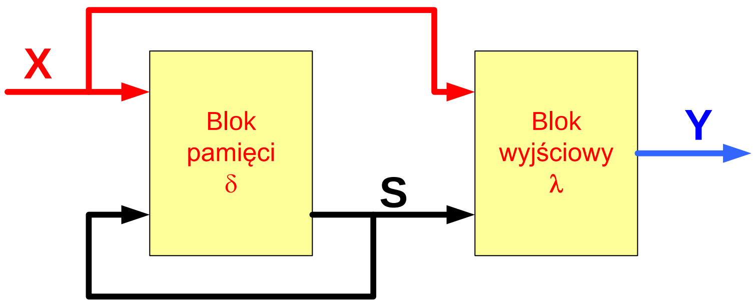
gdzie:
$X=\{x_0,x_1,x_2, \ldots,x_n\}$ – wektor sygnałów wejściowych,
$S=\{s_0,s_1,s_2, \ldots,s_r\}$ – wektor stanów wewnętrznych,
$Y=\{y_0,y_1,y_2, \ldots,y_r\}$ – wektor sygnałów wyjściowych,
δ – funkcja przejść automatu Mealy'ego δ=S x X,
λ– funkcja wyjść automatu Mealy'ego λ=S x X.

W automacie Mealy'ego wartość stanu wewnętrznego zależy od bieżącej wartości stanu, w którym znajduje się automat oraz od sygnałów wejściowych. Wynika z tego, że równanie stanu automatu Mealy'ego realizowane jest przez blok pamięci i przyjmuje postać:

$$S(t+1)=\delta(S(t),X(t)),$$

Wartość na wyjściu automatu zależy od stanu, w którym znajduje się automat oraz od wartości wejściowej. To znaczy, że równanie wyjść automatu Mealy'ego, realizowane przez blok wyjściowy ma postać:

$$Y(t)=\lambda(S(t),X(t)).$$



Rys. 4. Automat Mealy'ego.

4.2.2 Automat Moore'a

Automatem skończonym Moore'a (Rys. 5) nazywać będziemy układ:

$$\mu=<X,S,Y,\delta,\lambda>$$

gdzie:

X={x0,x1,x2, …,xn} – wektor sygnałów wejściowych,

S={s0,s1,s2, …,sr} – wektor stanów wewnętrznych,

Y={y0,y1,y2, …,yr} – wektor sygnałów wyjściowych,

δ – funkcja przejść automatu Moore'a,

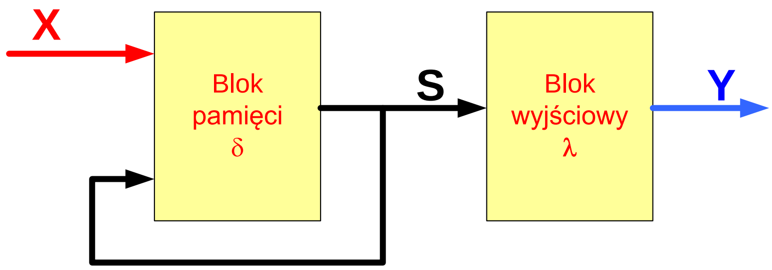
λ– funkcja wyjść automatu Moore'a, W automacie Moore'a wartość stanu wewnętrznego zależy od bieżącej wartości stanu, w którym znajduje się automat oraz od sygnałów wejściowych. Równanie stanu automatu Moore'a, realizowane przez blok pamięci przyjmuje postać taką samą jak w automacie Mealy'ego:

$$S(t+1)=\delta(S(t),X(t)),$$

Wartość na wyjściu automatu zależy tylko od stanu, w którym znajduje się automat, więc równanie wyjść automatu Moore'a, realizowana przez blok wyjściowy:

$$Y(t)=\lambda(S(t)).$$

Jest to podstawowa cecha odróżniająca automat Moore'a od automatu Mealy'ego.



Rys. 5. Automat Moore'a.

4.3 Opis układów sekwencyjnych

Układ sekwencyjny opisywany jest przez:

- opis słowny,
- wykres czasowy,
- graf przejść i wyjść,
- tablicę przejść i wyjść.

Opis słowny jest opisem działania układu, w którym podane są charakterystyczne informacje o wektorze wejściowym, stanach wewnętrznych układu i wektorze wyjściowym.

Wykres czasowy określa wzajemne zależności pomiędzy sygnałami wejściowymi i wyjściowymi. Każdemu sygnałowi przyporządkowane są wartości 0 lub 1. Oś czasu nie jest skalowana najczęściej przedstawia tylko zależności pomiędzy odpowiednimi sygnałami wejściowymi i wyjściowymi.

Tablica przejść opisuje funkcję przejść δ. W odpowiednich polach tabeli wpisuje się wartości następnych stanów. Pole określone jest przez wartość wektora wejściowego oraz stan bieżący.

Tablica wyjść, opisuje funkcję wyjść λ i jest różna zależnie od typu automatu. W automacie Mealy'ego wartość wektora wyjść wpisywana jest w te same pola, co tabela przejść, ponieważ wartość wyjściowa zależy od wektora wejść oraz od stanu układu. W automacie Moore'a generuje się oddzielną tabelę, w której umieszcza się wartości wyjściowe automatu odpowiadające odpowiednim stanom. Należy zauważyć, że zawsze pierwotna tabela stanu i wyjść jest generowana dla automatu Moore'a i dopiero po wprowadzeniu kolejnych przekształceń wyznacza się tabele Mealy'ego albo pozostaje się przy automacie Moore'a.

Graf przejść i wyjść zawiera pełną informację o układzie. W grafie umieszczone są informacje o liczbie stanów wewnętrznych układu cyfrowego S i wektorze wejść i wyjść. Wierzchołki grafu odpowiadają stanom wewnętrznym układu. Gałęzie grafu odpowiadają wektorowi wejść i opisują przejście pomiędzy dwoma stanami. Gałąź jest wyposażona w zwrot, który określa kierunek przechodzenia z bieżącego stanu do następnego. Tak opisywana jest funkcja przejść δ.

Stan wyjść w automacie opisuje się zależnie od typu automatu. W automacie Moore'a wartości wyjściowe zależą bezpośrednio od stanu, w którym znajduje się automat. Wartości wyjściowe bezpośrednio przyporządkowane są wierzchołkom grafu. W automacie Mealy'ego wartości wyjściowe zależą od stanu, w którym znajduje się automat i od wektora wejściowego. Dlatego, w tego typu automatach wartości wektora wyjść umieszczone są obok w gałęziach obok wektora wejściowego.

5 Projektowanie układów sterowania sekwencyjnego

Punktem wyjścia do projektowania układu asynchronicznego jest opis słowny, przebieg czasowy sygnałów wejściowych i wyjściowych, graf lub tabela przejść i wyjść.

Proces projektowania realizowany jest zgodnie z następującymi etapami:

1. Wyznaczenie grafu przejść i wyjść na podstawie opisu słownego lub przebiegów czasowych sygnałów wejściowych i wyjściowych.
2. Sporządzenie pierwotnej tabeli przejść i wyjść.
3. Redukcja pierwotnej tabeli przejść i wyjść.
4. Wyznaczenie funkcji przejść.
5. Wyznaczenie funkcji wyjść.

AB	**Q^n**		**Q^{n+1}**
	0	**1**	
00	1	1	1
01	0	1	Q^n
11	0	0	0
10	1	1	1

Proces projektowania zostanie przedstawiony na przykładzie układu, którego tabela przejść i wyjść podana jest powyżej. Układ będzie pracować synchronicznie zgodnie z taktami zegarowymi podawanymi na wejście dodatkowe „clock". W układzie wartości wyjść równe są stanowi układu Q^n: $y = Q^n$

W tabelo Q^{n+1} oznacza stan następny względem stanu Q^n, taki zapis przedstawia następstwo stanów.

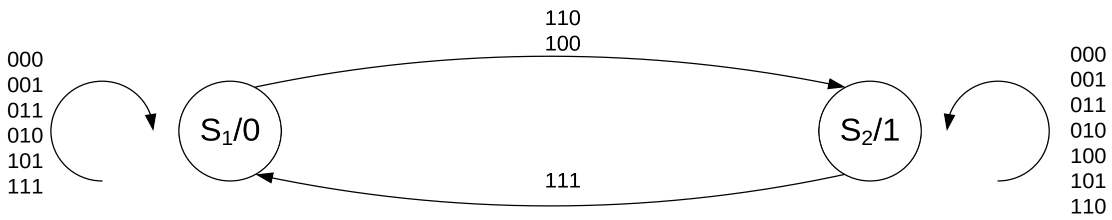
Po wprowadzeniu wejścia zegarowego otrzymamy tabelę:

Q^n	„clock" A B							
	000	001	011	010	110	111	101	100
0	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**
1	**1**	**1**	**1**	**1**	**1**	**0**	**1**	**1**

Zgodnie z tabelą, jeżeli sygnał zegarowy ma wartość „0” to jest utrzymywany aktualny stan niezależnie od stanu wejść A i B. Jeżeli sygnał zegarowy „clock” przyjmuje wartość „1”, to występują trzy przypadki:

1. dla wektora wejściowego (A, B) [1 0] i [0 0] na wyjściu układu będzie wartość „1”,
2. dla wektora wejściowego (A, B) [1 1] na wyjściu układu będzie wartość „0”,
3. dla wektora wejściowego (A, B) [0 1] na wyjściu układu będzie utrzymana wartość stanu Q^n, tak jak dla clock=0.

Układ posiada dwa stany wewnętrzne (stan wewnętrzny Q^n), które odpowiednio przyjmują wartość „0” i „1”.



Rys. 6. Graf przejścia.

Na rysunku 6 przedstawiony jest graf przejścia między poszczególnymi stanami.

Korzystając tabel przejść (rys. 4.b) dla przerzutnika asynchronicznego rs wyznaczone zostaną tabele odpowiedzialne za wzbudzenie odpowiednio wejścia set i reset przerzutnika. W tabeli 1 sprawdzamy stan Qn a następnie dla wektora wejściowego <A, B> określamy jaki będzie następny stan Qn+1, np. dla Qn = 0 i wektora wejściowego <1, 0> (‘clock’ = 1) następny stan jest równy Qn+1=1. Zgodnie z tabela przejść (rys. 4.b) aby przejść ze stanu 0⇒1 należy na wejściu set ustawić wartość 1 a na wejściu reset ustawiona jest wartość 0 (dla Qn=1 i wektora AB=<1, 0> wartość Qn+1=1, co zgodnie z tabelą przejść przerzutnika sr przypisuje wejściu set wartość nieokreśloną „-„„ natomiast wejściu reset wartość 0). W odpowiednich polach tabeli wzbudzeń wejścia set i reset umieszczamy wartość wynikające z tabeli przejść przerzutnika sr.

Tabela dla wejścia ustawiającego S (set) jest następująca:

Q^n	„clock” A B							
	000	001	011	010	110	111	101	100
0	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**
1	**-**	**-**	**-**	**-**	**-**	**0**	**-**	**-**

Aby wyznaczyć najprostszą postać funkcji wzbudzeń najwygodniej zastosować minimalizację funkcji logicznych metodą Karnaugh’a. W tym celu warto przekształcić tabelę dla wejścia ustawiającego S w taki sposób aby stan odpowiadający wejściu „clock”(C) znalazł się razem ze stanem wewnętrznym Qn. Następnie zgodnie z zasadą minimalizacji metodą tablicy Karnaugh’a należy zaznaczyć odpowiednie grupy i wypisać funkcję wzbudzeń dla wejścia „set”.

AB / CQ^n	00	01	11	10
00	**0**	**0**	**0**	**0**
01	-	-	-	-
11	-	-	**0**	-
10	1	**0**	**0**	1

Funkcja wzbudzeń dla wejścia „set" jest następująca:

$$S^n = C\bar{B} = (clock)\bar{B}$$

Analogicznie należy postąpić dla wejścia ustawiającego R (Reset), dla którego tabela jest następująca:

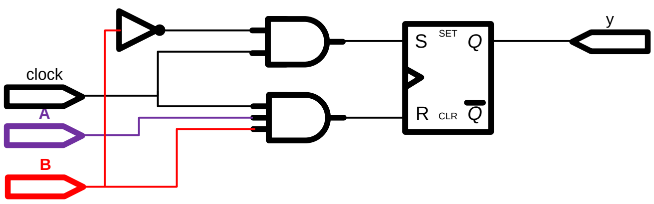
Q^n	„clock" A B							
	000	001	011	010	110	111	101	100
0	-	-	-	-	**0**	-	-	**0**
1	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**

Tablica Karnaugh'a po przekształceniach przyjmuje postać:

AB / CQ^n	00	01	11	10
00	-	-	-	-
01	**0**	**0**	**0**	**0**
11	**0**	**0**	1	**0**
10	**0**	-	-	**0**

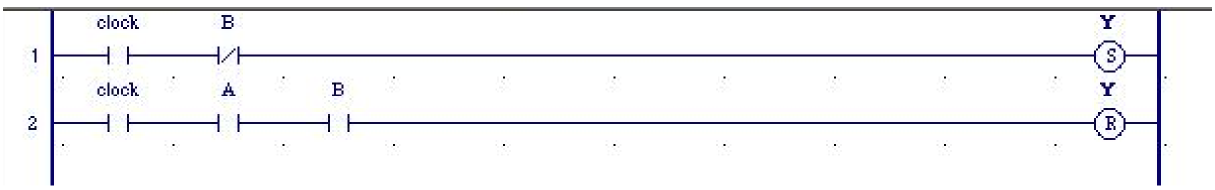
I ostatecznie funkcja wzbudzeń dla wejścia „reset" jest następująca:

$$R^n = CAB = (clock)\ AB$$



Rys. 7. Realizacja układu sterowania.

Podany przykład można zrealizować jako układ elektroniczny (Rys. 7) lub zaprogramować sterownik PLC (Rys 8).



Rys. 8. Program napisany w języku drabinkowym.

6 Literatura

1. Janusz KOWAL „Podstawy automatyki T1", Uczelniane Wydawnictwa Naukowo-Dydaktyczne AGH, Kraków 2004, Sygnatura: 60378
2. Tadeusz Kaczorek „*Teoria sterowania. Tom I Układy liniowe ciągłe i dyskretne*". Państwowe Wydawnictwo Naukowe, Warszawa 1977.
3. Dariusz Horla *„Podstawy automatyki. Ćwiczenia rachunkowe. Część I"*, Wydawnictwo Politechniki Poznańskiej, Poznań 2003.