ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1969
Seria: AUTOMATYKA z. 14 Nr kol. 267

Jerzy Siwiński, Henryk Małysiak,
Antoni Niederliński, Ferdynand Wagner

3.1. PRACE KATEDRY AUTOMATYKI PROCESÓW PRZEMYSŁOWYCH W ZAKRESIE TEORII AUTOMATÓW

1. Wstęp

Prace naukowe Katedry Automatyki Procesów Przemysłowych prowadzone w zakresie teorii automatów skierowane były głównie na znalezienie możliwie prostych i wygodnych w zastosowaniu sposobów syntezy struktur automatów. Jest to zagadnienie z którym projektant układów automatyki procesów przemysłowych spotyka się stale w swojej codziennej praktyce inżynierskiej.

Wyniki prac katedry prowadzonych w tym zakresie można streścić w następującym zestawieniu:

Synteza automatów sekwencyjnych za pomocą numerycznego zapisu funkcji logicznych - autor prof. dr J. Siwiński.

Synteza automatów sekwencyjnych metodą uproszczoną - autor prof. dr J. Siwiński.

Synteza automatów sekwencyjnych z zastosowaniem przerzutników - autor dr inż. H. Małysiak.

Synteza synchronicznych układów cyfrowych - autor dr inż. F. Wagner.

Projektowanie układów sekwencyjnych o nietrwałej pamięci metodą operatorów czasowych - autor doc. dr Antoni Niederliński.

Każda pozycja tego zestawienia zawiera opracowanie nowych i prostych w zastosowaniu inżynierskich metod projektowania układów logicznych decydujących o strukturze automatu. Dwie pierwsze pozycje były tematem publikacji zagranicznych (Londyn i Moskwa), dwie następne były tematami prac doktorskich, a ostatnia tematem pracy habilitacyjnej. Pracownicy Katedry mają z tego zakresu szereg publikacji krajowych w tym dwie książki prof. dr J. Siwińskiego: "Układy przekaźnikowe w automatyce" WNT Warszawa

1964 i "Układy przełączające w automatyce", WNT Warszawa 1968 oraz jedną pracę zbiorową - skrypt "Zbiór zadań z teorii automatów", Politechnika Śląska, Gliwice 1968.

2. Synteza automatów sekwencyjnych za pomocą numerycznego zapisu funkcji logicznych

Metoda ta może mieć zastosowanie w tych przypadkach, kiedy założone warunki pracy automatu można zapisać w postaci tablic kolejności łączeń. Są to przypadki występujące często w zagadnieniach automatyzacji procesów przemysłowych. Znając proces technologiczny którym chcemy automatycznie sterować, łatwo wyznaczamy elementy wejściowe i wyjściowe szukanego układu logicznego i zestawiamy tablicę kolejności łączeń składającą się z tych elementów. Taka tablica nie zawsze jest rozwiązalna, co sprawdzamy przez obliczenie stanu automatu w poszczególnych taktach jego pracy. Stany te przedstawiamy w numerycznym zapisie dziesiętnym. Jeżeli w ciągu jednego cyklu pracy automatu powstarzają się te same stany w różnych taktach, to tablica zawiera logiczne sprzeczności i jest nierozwiązalna. Wtedy uzupełniamy tablicę elementami pośredniczącymi w taki sposób, aby usunąć te powtórzenia, co jest równoznaczne z usunięciem występujących w niej sprzeczności. Uzupełniona tablica jest rozwiązalna i może być zastosowana do syntezy funkcji logicznych poszczególnych elementów.

Na rys. 2.1a przedstawiono przykład nierozwiązalnej tablicy kolejności łączeń. Aby tablica ta stała się rozwiązalna, należy ją uzupełnić jednym elementem pośredniczącym Q, jak pokazano na rys. 2.1b.

Numeryczny zapis funkcji logicznych otrzymuje się z rozwiązalnej tablicy kolejności łączeń na podstawie następujących wyrażeń:

$$F(X) = \sum_{\alpha=1}^{1} K_{\alpha}^{1} + \sum \phi^{1} \tag{2.1}$$

$$F(X) = \prod_{\beta=1}^{i} K_{\beta}^{0} \cdot \prod \phi^{0} \tag{2.2}$$

gdzie:

i – liczba taktów w których występują warunki działania rozpatrywanego elementu X (stan 1);

j – liczba taktów w których występują warunki niedziałania tegoż elementu (stan 0),

K^1 – składniki jedynki szukanej funkcji logicznej,

K^0 – czynniki zera szukanej funkcji logicznej,

$\emptyset^1$ – te stany obojętne, które traktujemy jako składniki jedynki,

$\emptyset^0$ – te stany obojętne które traktujemy jako czynniki zera.

Przy określaniu taktów w których dla rozpatrywanego elementu występują warunki działania (stan 1) należy zawsze uwzględniać takt poprzedzający cykl załączenia tego elementu, gdyż wtedy powstaje przyczyna jego zadziałania i pominąć ostatni takt cyklu załączenia, gdyż w tym ostatnim także powstaje przyczyna jego wyłączenia. To samo dotyczy taktów w których występują warunki niedziałania (stan 0).

a)

3' 5'

Takty			0	1	2	3	4	5	6	7	8	
Stan elementów	2^0	A	—	+								
	2^1	X_1	—		+			—	+			
	2^2	X_2	—			+	—			+	—	
Stan automatu			0	1	3	7	3	1	3	7	3	

1 cykl

b)

Takty			0	1	2	3	3'	4	5	5'	6	
Stan elementów	2^0	A	—	+								
	2^1	X_1	—	<	+		>		—		+	
	2^2	X_2	—		<	+ >		—				
	2^3	Q	—			<	+	>		—		
Stan automatu			0	1	3	7	15	11	9	1	3	

1 cykl

Rys. 2.1. Tablica kolejności łączeń: a) nierozwiązalna; b) rozwiązalna

Uznanie stanów obojętnych ϕ za składniki jedynki lub czynniki zera wynika bezpośrednio z siatek zależności, zestawionych dla każdego elementu tablicy osobno. W siatkach tych należy wyznaczyć zarówno wejścia jedynkowe jak zerowe. W tym celu zapisujemy dla każdego elementu tablicy kolejności łączeń jednocześnie obydwie postacie wyrażenia logicznego tego elementu, odpowiadającego wzorem (2.1) i (2.2).

Suma otrzymanych z siatek zależności funkcji logicznych odpowiadających elementom tablicy kolejności łączeń określa strukturę automatu.

Przykład 2.1. Wykonać syntezę automatu opisanego tablicą kolejności łączeń z rys. 2.1. stosując metodę zapisu numerycznego funkcji logicznych.

Na podstawie tablicy rozwiązalnej (rys. 2.1b) mamy:

Dla elementu X_1

$$F(X_1)_{1-3} = \sum(1,3,7,15)_{Q\,X_2X_1A} + \sum \phi^1$$

$$F(X_1)_{0,4,5} = \prod(0,11,9)_{Q\,X_2X_1A} \cdot \prod \phi^0$$

Tym wyrażeniem odpowiada siatka zależności z rys. 2.2a.

Stąd

$$F(X_1)_{\Sigma} = a.\bar{q} + x_2; \quad F(X_1)_{\Pi} = a(\bar{q} + x_2)$$

a)

	QX_2 \ X_1A	00 (0)	01 (1)	11 (3)	10 (2)
0	00	0	1	1	
4	01			1	
12	11			1	
8	10		0	0	

X_1

b)

	QX_2 \ X_1A	00 (0)	01 (1)	11 (3)	10 (2)
0	00	0	0	1	
4	01			1	
12	11			0	
8	10		0	0	

X_2

c)

	QX_2 \ X_1A	00 (0)	01 (1)	11 (3)	10 (2)
0	00	0	0	0	
4	01			1	
12	11			1	
8	10		0	1	

Q

Rys. 2.2. Siatki zależności elementów X_1, X_2, Q

Analogicznie dla elementu X_2 otrzymamy na podstawie siatki zależności z rys. 2.2b

$$F(X_2)_{\Sigma} = x_1\bar{q}; \quad F(X_2)_{\Pi} = x_1\bar{q}$$

Podobnie dla elementu Q na podstawie siatki zależności z rys. 2.2c

$$F(Q)_{\Sigma} = x_1\, q + x_2; \quad F(Q)_{\Pi} = x_1(q + x_2)$$

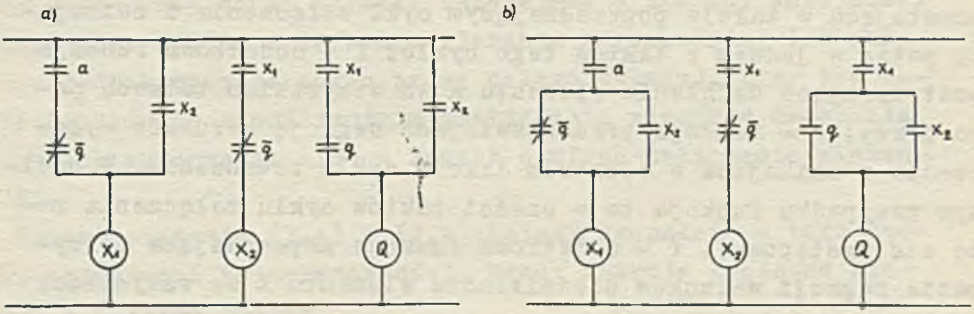
Wyrażenie strukturalne automatu

1) na podstawie warunków działania

$$F = (a\,\bar{q} + x_2) \cdot X_1 + x_1\,\bar{q} \cdot X_2 + (x_1\, q + x_2) \cdot Q$$

2) na podstawie warunków niedziałania

$$F = a(\bar{q} + x_2) \cdot X_1 + x_1\bar{q} \cdot X_2 + x_1(q + x_2) \cdot Q$$



Rys. 2.3. Struktura automatu otrzymana z warunków: a) działania; b) niedziałania

Na rys. 2.3. pokazano szukaną strukturę automatu odpowiadającą otrzymanym wyrażeniem logicznym.

3. Synteza automatów sekwencyjnych metodą uproszczoną

Uproszczona metoda syntezy automatów sekwencyjnych polega na bezpośrednim analizowaniu tablicy kolejności łączeń i otrzymaniu na tej podstawie od razu uproszczonych postaci funkcji logicznych. Metoda ta jest szczególnie przydatna w przypadku dużej liczby elementów występujących w automacie i przy niezbyt dużym skomplikowaniu tablicy kolejności łączeń, co w praktyce przemysłowej często ma miejsce.

Podstawą do określenia warunków pracy poszczególnych elementów jest jeden cykl pracy automatu, tj. taka liczba taktów rozwiązalnej tablicy kolejności łączeń, po których występuje powtórzenie tych samych **stanów** automatu.

Podczas trwania cyklu załączenia dowolnego elementu X powinny być spełnione warunki jego załączenia i równocześnie negacja warunków jego wyłączenia. Powyższe warunki określają bezpośrednio postać funkcji logicznej elementu X.

Chociaż ta metoda wymaga wnikliwej analizy tablic kolejności łączeń, to jednak przy pewnym doświadczeniu projektanta jest bardzo prosta w zastosowaniu.

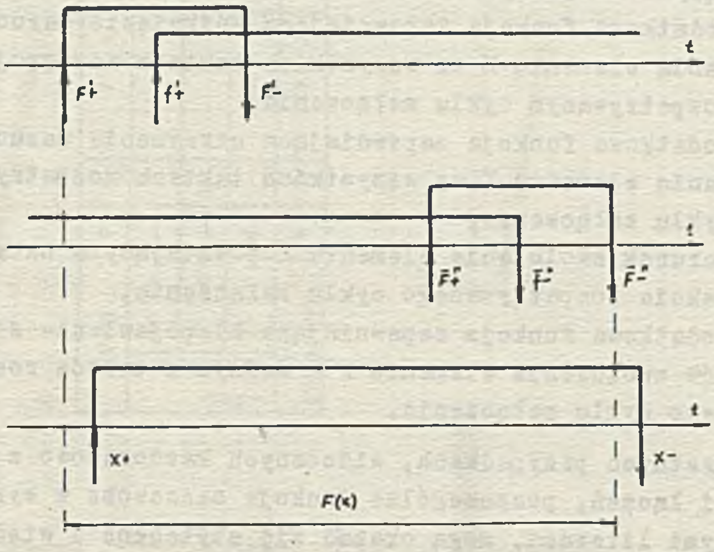
Każdy cykl załączenia dowolnego elementu X można przedstawić w sposób pokazany na rys. 3.1. Na tym rysunku oznaczono: F' – funkcja przedstawiająca przyczynę załączenia elementu X, powstająca w takcie poprzedzającym cykl załączenia i znikająca potem w jednym z taktów tego cyklu; f' – dodatkowa funkcja podtrzymująca działanie elementu X we wszystkich tatkach jego pracy; $\overline{F}''$ – funkcja przedstawiająca negację warunków wyłączenia i znikająca w ostatnim takcie cyklu załączenia. W ogólnym przypadku funkcja ta w części taktów cyklu załączenia może nie występować. $\overline{f}''$ – dodatkowa funkcja zapewniająca utrzymanie negacji warunków niedziałania elementu X we wszystkich taktach jego załączenia.

Podczas cyklu załączenia elementu X powinny istnieć warunki działania $(F' + f')$ i negacja warunków niedziałania $(\overline{F}'' + \overline{f}'')$. Zatem funkcję logiczną elementu X można zapisać w postaci

$$F(X) = (F' + f').(\overline{F}'' + \overline{f}'') \qquad (3.1)$$

Sumę negacji warunków niedziałania $(\overline{F}'' + \overline{f}'')$ można przedstawić jako negaoję iloozynu $\overline{F''.f''}$ i wtedy wyrażenie (3.1) otrzyma postać

$$F(X) = (F' + f') \cdot \overline{F'' f''} \qquad (3.2)$$



Rys. 3.1. Cykl załączenia elementu X automatu wielotaktowego

Przy zestawianiu funkoji logicznej dowolnego elementu X należy także uwzględnić warunki istniejąoe w tablicy kolejnośoi łączeń poza cyklem załączenia elementu X (wcześniej lub później, jednak podczas cyklu pracy całego układu). Przy tym należy liczyć się z możliwością powtórzenia warunków działania w oyklach wyłączenia albo w innych cyklach załączenia elementu X.

Zatem wyrażenia (3.1) i (3.2) należy uzupełnić w taki sposób, aby usunąć te sprzeczności. Wtedy funkcja logiczna elementu X otrzyma postać

$$F(X) = (F' f'_a\, f'_b + f'_c\,) \cdot \overline{F'' f''} \qquad (3.3)$$

gdzie:

F' – warunek zadziałania elementu X powstający w takcie poprzedzającym takt załączania,

f'_a – dodatkowa funkcja zapewniająca zniknięcie warunków działania elementu X we wszystkich taktach poprzedzających rozpatrywany cykl załączenia;

f'_b – dodatkowa funkcja zapewniająca zniknięcie warunków działania elementu X we wszystkich taktach następujących po rozpatrywanym cyklu załączenia;

f'_c – dodatkowa funkcja zapewniająca utrzymanie warunków działania elementu X we wszystkich taktach rozpatrywanego cyklu załączania,

F'' – warunek zwolnienia elementu X powstający w ostatnim takcie rozpatrywanego cyklu załączenia,

f'' – dodatkowa funkcja zapewniająca niepojawienie się warunków zwolnienia elementu X w żadnym z taktów rozpatrywanego cyklu załączenia.

W konkretnych przypadkach, widocznych każdorazowo z tablicy kolejności łączeń, poszczególne funkcje oznaczone w wyrażeniu (3.3) małymi literami, mogą okazać się zbyteczne i wtedy to wyrażenie odpowiednio upraszcza się. W najbardziej prostym przypadku otrzymuje ono postać

$$F(X) = F' \cdot \overline{F''} \qquad (3.4)$$

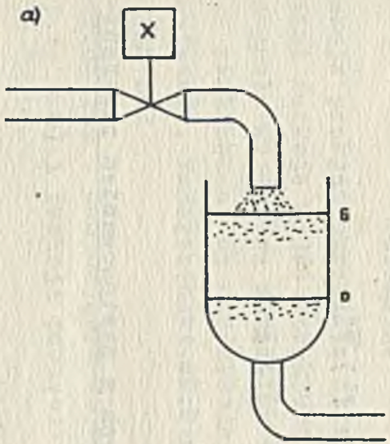
Jeżeli dowolny element X ma w tablicy kolejności łączeń kilka cykli załączenia, to w sposób przedstawiony wyżej wyznacza się funkcje logiczne elementu X dla każdego cyklu osobno

$$F_I(X), \quad F_{II}(X), \ldots$$

i otrzymane wyniki sumuje się

$$F(X) = F_I(X) + F_{II}(X) + \ldots \qquad (3.5)$$

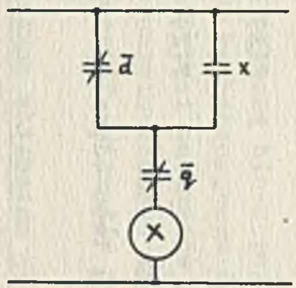
a)



b)

Takty			0	1	2	3	4	5	6	7	
Stan elementów	2^0	G	−		+		−				
	2^1	D	−	+				−		+	
	2^2	X	+			−			+		
Stan automatu			4	6	7	3	2	0	4	6	

c)



Rys. 3.2. Synteza automatu sterującego poziom wody w zbiorniku: a) proces technologiczny; b) tablica kolejności łączeń; c) struktura automatu

Należy podkreślić, że jeżeli tablica kolejności łączeń jest rozwiązalna, to wszystkie dodatkowe elementy, które w różnych przypadkach należy wprowadzać do układu w postaci f'_a, f'_b, f'_c, zawsze znajdują się w tej tablicy.

Funkcje wchodzące do ogólnego wyrażenia (3.3) przedstawione są zwykle za pomocą pojedynczych liter i dlatego przy tej metodzie otrzymuje się w rezultacie od razu uproszczoną postać funkcji logicznych, nie wymagającą już żadnych przekształceń.

Przykład 3.1.

Wykonać metodą uproszczoną syntezę struktury automatu sterującego zaworem X na rurociągu dopływowym zbiornika z wodą w taki sposób, aby wahania poziomu wody nie przekraczały granicy górnej G i dolnej D, rys. 3.2a.

Automat ma dwa elementy wejściowe: czujniki G i D, i jeden element wyjściowy X. Zestawiamy tablicę kolejności łączeń ilustrującą warunki pracy automatu, rys. 3.2b.

Tablica jest rozwiązalna. Na podstawie tej tablicy otrzymamy

$$F(X) = (F' + f').\overline{F''}$$

$$F(X) = (\overline{d} + x) \cdot \overline{g}$$

Wyrażeniu temu odpowiada rysunek 3.2c.

4. Synteza automatów sekwencyjnych z zastosowaniem przerzutników

4.1. Statyczne automaty sekwencyjne

Automatem statycznym nazywa się takie układy sterowania, w których wszystkie informacje przekazywane są w postaci konkretnych wartości sygnałów logicznych, 0 lub 1, a nie ich zmian z 0 na 1 lub z 1 na 0. Algorytmy syntezy bezstykowych automatów sekwencyjnych niewiele różnią się od algorytmów stosowanych przy syntezie układów stykowych. Istotne różnice występują dopiero przy realizacji otrzymanych w procesie syntezy funkcji przejść i wyjść na konkretnych elementach. Niech będzie dany automat sekwencyjny o "n" wejściach $(x_1 \dots x_n)$ i m wyjściach $(Z_1 \dots Z_m)$, którego działanie opisują funkcje przełączające:

funkcje przejść

$$Y_k = \varphi_k(x_1, \dots, x_i, \dots, x_n; \quad y_1, \dots, y_k, \dots, y_l) \qquad (4.1)$$

funkcje wyjść

$$Z_j = f_i(x_1, \dots, x_i, \dots, x_n; \quad y_1, \dots, y_k, \dots, y_l) \qquad (4.2)$$

gdzie:

$i = 1,2 \dots n$ – ilość sygnałów wejściowych automatu;
$j = 1,2 \dots m$ – ilość sygnałów wyjściowych automatu;
$k = 1,2 \dots l$ – ilość sygnałów wyjściowych bloku pamięci automatu.

Funkcje przejść określają strukturę logiczną bloku pamięci, natomiast funkcje wyjść strukturę układu kombinacyjnego formującego sygnały wyjściowe automatu. Bloki pamięci automatów sekwencyjnych realizowane są najczęściej przy użyciu różnego typu przerzutników. Przerzutniki tworzące "pamięć" automatów można podzielić na:

- przerzutniki o dominującym wejściu zerującym (dla $f_{ak} \cdot f_{bk} = 1$, $Y_k = 0$) i funkcji pamięci:

$$Y_k = \bar{f}_{bk}(y_k + f_{ak}) \qquad (4.3)$$

- przerzutniki o dominującym wejściu wpisującym (dla $f'_{ak} \cdot f'_{bk} = 1$, $Y_k = 1$) i funkcji pamięci

$$Y_k = f'_{ak} + \bar{f}'_{bk} y_k \qquad (4.4)$$

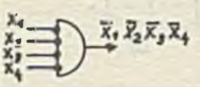
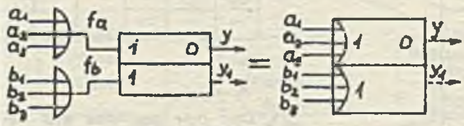
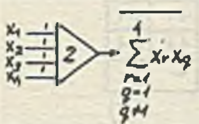
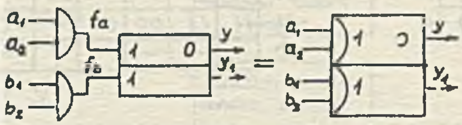
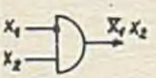
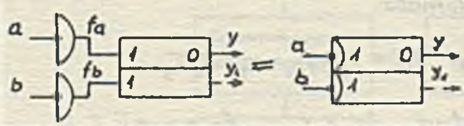
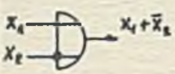
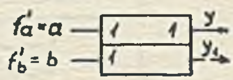
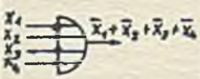
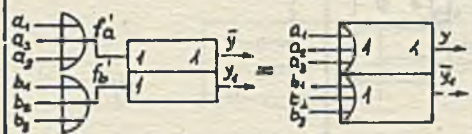
Przykłady typowych przerzutników statycznych, odpowiadające im symbole logiczne i funkcje pamięci przedstawione są w tablicy 4.1. Omawiany sposób syntezy automatów sekwencyjnych sprowadza się do określenia - na podstawie znajomości funkcji przejść Y_k automatu i funkcji pamięci wybranego przerzutnika - takich funkcji wzbudzeń (f_{ak}, f_{bk} lub f'_{ak}, f'_{bk}), by funkcje pamięci realizowane przez przerzutniki pokrywały się z odpowiadającymi im funkcjami przejść Y_k dla wszystkich stanów ich argumentów. W tym celu każdą z funkcji przejść Y_k należy przekształcić do postaci odpowiadającej jej funkcji pamięci zastosowanego przerzutnika.

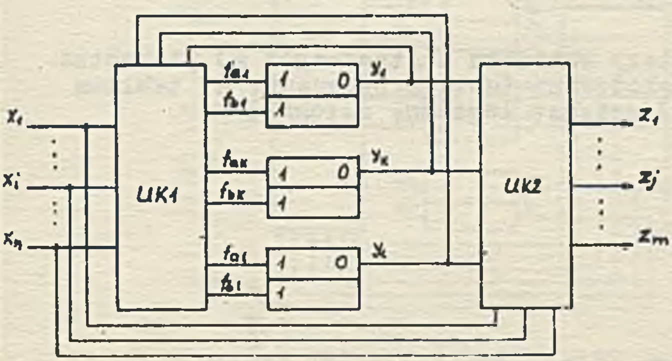
Porównanie tak przedstawionej funkcji przejść z funkcją pamięci przerzutnika, przy pomocy którego ma być ona realizowana, pozwala stosunkowo prosto wyznaczyć jego funkcje wzbudzeń. W ten sposób każdą z funkcji przejść Y_k można zrealizować przy pomocy jednego przerzutnika i pomocniczego układu kombinacyjnego określonego znanymi funkcjami f_{ak}, f_{bk} lub f'_{ak}, f'_{bk}. Schemat blokowy automatu sekwencyjnego z zastosowaniem przerzutników o dominujących wejściach zerujących przedstawiony jest na rys. 4.1. Porównując przedstawioną metodę syntezy statycznych automatów sekwencyjnych z metodą polegającą na określeniu funkcji wzbudzeń przerzutników na podstawie odpowiadających im tablic wzbudzeń można stwierdzić, że jest ona mniej pracochłonna i w niektórych przypadkach daje prostsze rozwiązanie. Przykład syntezy automatu statycznego przedstawiony jest na rys. 4.2.

4.2. Dynamiczne automaty sekwencyjne

Dynamicznymi automatami sekwencyjnymi nazywa się takie układy sterowania, w których informacje przekazywane są zarówno w postaci wartości sygnałów logicznych 0,1, jak i ich zmian z 1 na 0 i z 0 na 1. Bloki pamięci w dynamicznych automatach swkwencyjnych realizowane są za pomocą przerzutników pobudzanych impulsowo. Sygnały impulsowe otrzymuje się za pomocą iloczynów dynamicznych w wyniku różniczkowania opadających względnie narastających zboczy wejściowych sygnałów statycznych. Ze względu na równoczesne występowanie sygnałów statycznych i impulsowych synteza automatów dynamicznych różni się zasadniczo od

Tablica 4.1

Podstawowy element logiczny	Symbol logiczny przerzutnika	Funkcja pamięci przerzutnika
		$Y=\overline{b_1+b_2+b_3}(a_1+a_2+a_3+y)$ dla $f_a f_b \neq 1$, $y_1=\bar{y}$
 Element progowy		$Y=\overline{b_1 b_2}(a_1 a_2+y)$ dla $f_a f_b \neq 1$, $y_1=\bar{y}$
		$Y=b(\bar{a}+y)$ dla $f_a f_b \neq 1$, $y_1=\bar{y}$
		$Y=a+\bar{b}y$ dla $f'_a f'_b \neq 1$, $y_1=\bar{y}$
		$Y=\overline{a_1 a_2 a_3}+b_1 b_2 b_3 y$ dla $f'_a f'_b \neq 1$, $y_1=\bar{y}$



Rys. 4.1. Schemat blokowy statycznego automatu sekwencyjnego z przerzutnikami o dominujących wejściach zerujących

$x_1\, x_2\, x_3$

000	001	011	010	110	111	101	100	z
1	3	–	5	–	–	–	9	0
2	3	–	5	–	–	–	10	1
2	3	4	–	–	–	8	–	1
–	3	4	5	–	7	–	–	0
1	–	4	5	6	–	–	–	0
–	–	–	–	6	7	–	9	0
–	–	–	–	6	7	8	–	0
–	3	–	–	–	7	8	10	1
–	–	–	–	6	–	8	9	0
1	–	–	–	6	–	8	10	1

a)

$x_1\, x_2\, x_3$

$y_1 y_2$	000	001	011	010	110	111	101	100	z
00	1	3	4	5	6	7	8	9	0
01	1	–	–	–	6	–	8	10	1
11	2	3	4	5	–	7	8	10	1
10	–	–	–	–	–	–	–	–	–

b)

$x_1\, x_2\, x_3$

$y_1 y_2$	000	001	011	010	110	111	101	100
00	00	11	00	00	00	00	11	00
01	00	–	–	–	00	–	11	01
11	11	11	00	00	–	00	11	01
10	–	–	–	–	–	–	–	–

$Y_1\, Y_2$

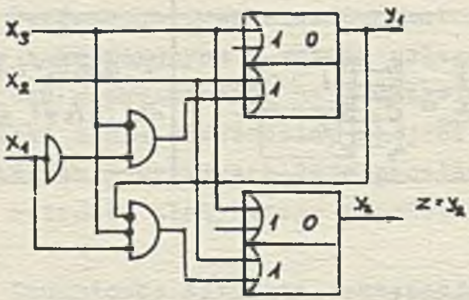
c)

Funkcje przełączające automatu:

$$Y_1 = \overline{x_2 + x_1\bar{x}_3}(y_1 + x_3) \quad \text{stąd} \quad fa_1 = x_3,\ fb_1 = x_2 + x_1\bar{x}_3$$

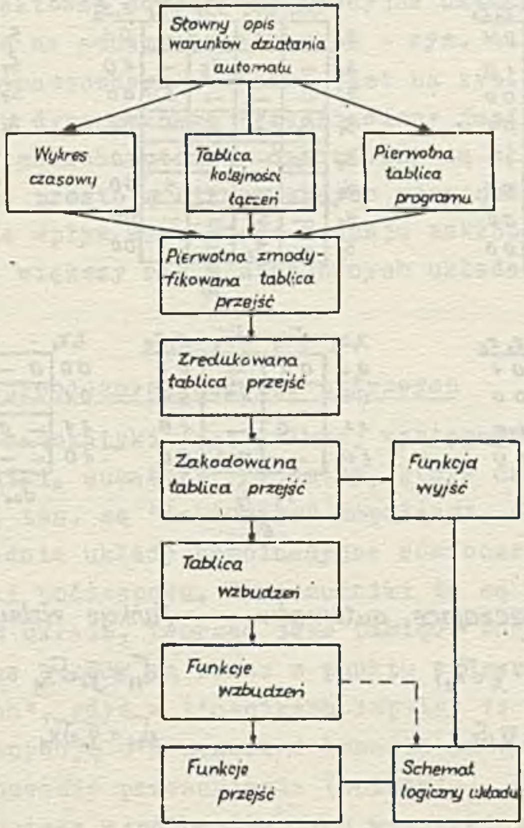
$$Y_2 = \overline{x_2 + \bar{x}_1\bar{x}_3\bar{y}_1}(y_2 + x_3) \quad \text{stąd} \quad fa_2 = x_3,\ fb_2 = x_2 + \bar{x}_1\bar{x}_3\bar{y}_1$$

$$z = y_2$$

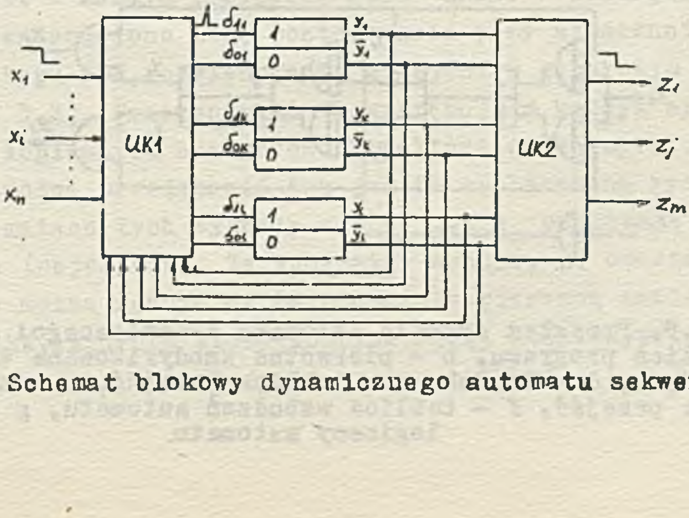


d)

Rys. 4.2. Przykład syntezy automatu statycznego: a) pierwotna tablica programu; b) zakodowana tablica programu; c) tablica stanów; d) schemat logiczny automatu



Rys. 4.3. Przedstawienie sposobu syntezy automatu dynamicznego



s. 4.4. Schemat blokowy dynamicznego automatu sekwencyjnego

x_1x_2: 00	01	11	z_1z_2
1	4	—	01
2	5	—	10
1	3	—	00
—	4	7	01
—	5	8	10
2	6	—	00
—	6	7	00
—	3	8	00

a)

	δx_1	$\delta \bar{x}_1$	δx_2	$\delta \bar{x}_2$	z_1z_2
S_1	—	—	S_4	—	01
S_2	—	—	S_6	—	10
S_3	—	—	—	S_4	00
S_4	S_7	—	—	—	01
S_5	S_8	—	—	—	10
S_6	—	—	—	S_2	00
S_7	—	S_6	—	—	00
S_8	—	S_3	—	—	00

b)

	δx_1	$\delta \bar{x}_1$	δx_2	$\delta \bar{x}_2$	z_1z_2
S_1	S_6	—	S_4	—	0 1
S_6	—	S_6	—	S_7	0 0
S_7	S_3	—	S_2	—	1 0
S_3	—	S_3	—	S_1	0 0

c)

	δx_1	$\delta \bar{x}_2$	z_1z_2
S_1	S_6	—	0 1
S_6	—	S_2	0 0
S_2	S_3	—	1 0
S_3	—	S_1	0 0

d)

y_1y_2	δx_1	$\delta \bar{x}_2$	z_1z_2
00	0 1	—	0 1
01	—	1 1	0 0
11	1 0	—	1 0
10	—	0 0	0 0

$Y_1 \; Y_2$

e)

y_1y_2	δx_1				$\delta \bar{x}_2$			
00	0	—	1	0	—	—	—	—
01	—	—	—	—	1	0	—	0
11	—	0	0	1	—	—	—	—
10	—	—	—	—	0	1	0	—

$\delta_{11}, \delta_{01}, \delta_{12}, \delta_{02}$

f)

Funkcje przełączające automatu:

$$Y_1 = \overline{y_1 \delta \bar{x}_2}\,(y_1 + \bar{y}_1 \delta \bar{x}_2)$$

$$Y_2 = \overline{y_1 \delta x_1}\,(y_2 + \bar{y}_1 \delta x_1)$$

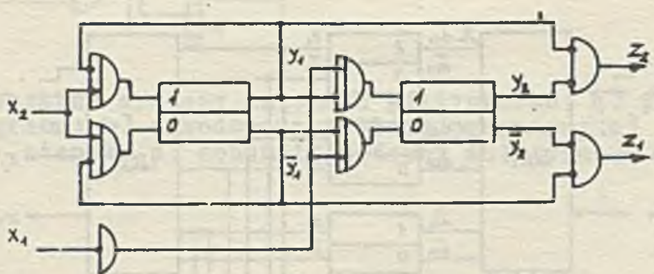
$$Z_1 = y_1 y_2$$

$$Z_2 = \bar{y}_1 \bar{y}_2$$

Funkcje wzbudzeń przerzutników:

$$\delta_{11} = \bar{y}_1 \delta \bar{x}_2 \;, \quad \delta_{01} = y_1 \delta \bar{x}_2$$

$$\delta_{12} = \bar{y}_1 \delta x_1 \;, \quad \delta_{02} = y_1 \delta x_1$$



g)

Rys. 4.5. Przykład syntezy automatu dynamicznego: a - pierwotna tablica programu, b - pierwotna zmodyfikowana tablica przejść, c, d - zredukowane tablice przejść, e - zakodowana tablica przejść, f - tablica wzbudzeń automatu, g - schemat logiczny automatu syntezy automatów statycznych. Poszczególne etapy syntezy pozwalającej projektować dowolne sekwencyjne układy sterowania przedstawione są na schemacie blokowym - rys. 4.3. Schemat blokowy automatu dynamicznego pokazany jest na rys. 4.4. Przykład syntezy automatu dynamicznego przedstawiony jest na rys. 4.5.

W wielu przypadkach automaty dynamiczne są chętnie stosowane ze względu na prostotę otrzymywanych rozwiązań. Należy jednak pamiętać, że wpływ wszelkiego rodzaju zakłóceń na ich pracę jest o wiele większy niż w statycznych układach sekwencyjnych.

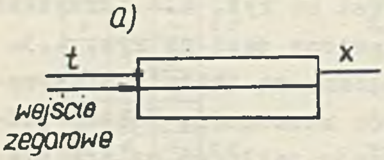
6. Synteza synchronicznych układów cyfrowych

W urządzeniach automatyki przemysłowej występują takie podzespoły jak liczniki, sumatory, rejestry, które często pracują synchronicznie, tzn. są "taktowane" impulsami, oddziałującymi poprzez odpowiednie układy kombinacyjne równocześnie na wszystkie przerzutniki podzespołu. Przerzutniki te są najistotniejszym fragmentem układu, tworząc jego pamięć. Zadania spełnione przez wymienione układy są różne z punktu widzenia roli impulsów "taktujących", gdyż w licznikach impulsy te są zaliczane, w sumatorach decydują o momencie dokonania sumowania, a w rejestrach - o momencie przesunięcia (wpisania). Mimo to można podać metodę syntezy wspólną dla tych wszystkich układów.

Układy te można budować przy zastosowaniu dowolnych przerzutników. W zależności od przyjętego typu przerzutnika metoda syntezy jest w każdym przypadku nieco inna. W niniejszym opracowaniu przedstawiono ideę postępowania przy stosowaniu przerzutników typu J-K, względnie T, z wejściem zegarowym (rys. 5.1 i rys. 5.2). Przerzutniki te występują w każdym systemie elementów logicznych scalonych. Strukturę omawianych układów cyfrowych można przedstawić tak jak to zrobiono na rys. 5.3.

Na schematach tych sygnały a_1, a_2 ... a_m oznaczają sygnały zewnętrzne (wejściowe). Ze schematów wynika, że omawiane układy cyfrowe można rozbić na dwie części: pierwszą zawierająca przerzutniki, i drugą część kombinacyjną, która wypracowuje sygnały sterujące (bramkujące) przerzutnikami.

Synteza takiego układu cyfrowego sprowadza się do znalezienia wyrażeń logicznych opisujących układ kombinacyjny sterujący

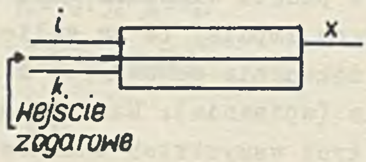


b)

t_n	x_{n+1}
0	x_n
1	$\overline{x}_n$

Rys. 5.1. Przerzutnik T z wejściem zegarowym: a) symbol; b) tabela

przerzutnikami. Sygnałami wejściowymi są dla tego układu sygnały zewnętrzne a_j oraz wyjścia przerzutników x_k. Sygnałami wyjściowymi są sygnały sterujące (bramkujące) t_i względnie j_i oraz k_i.



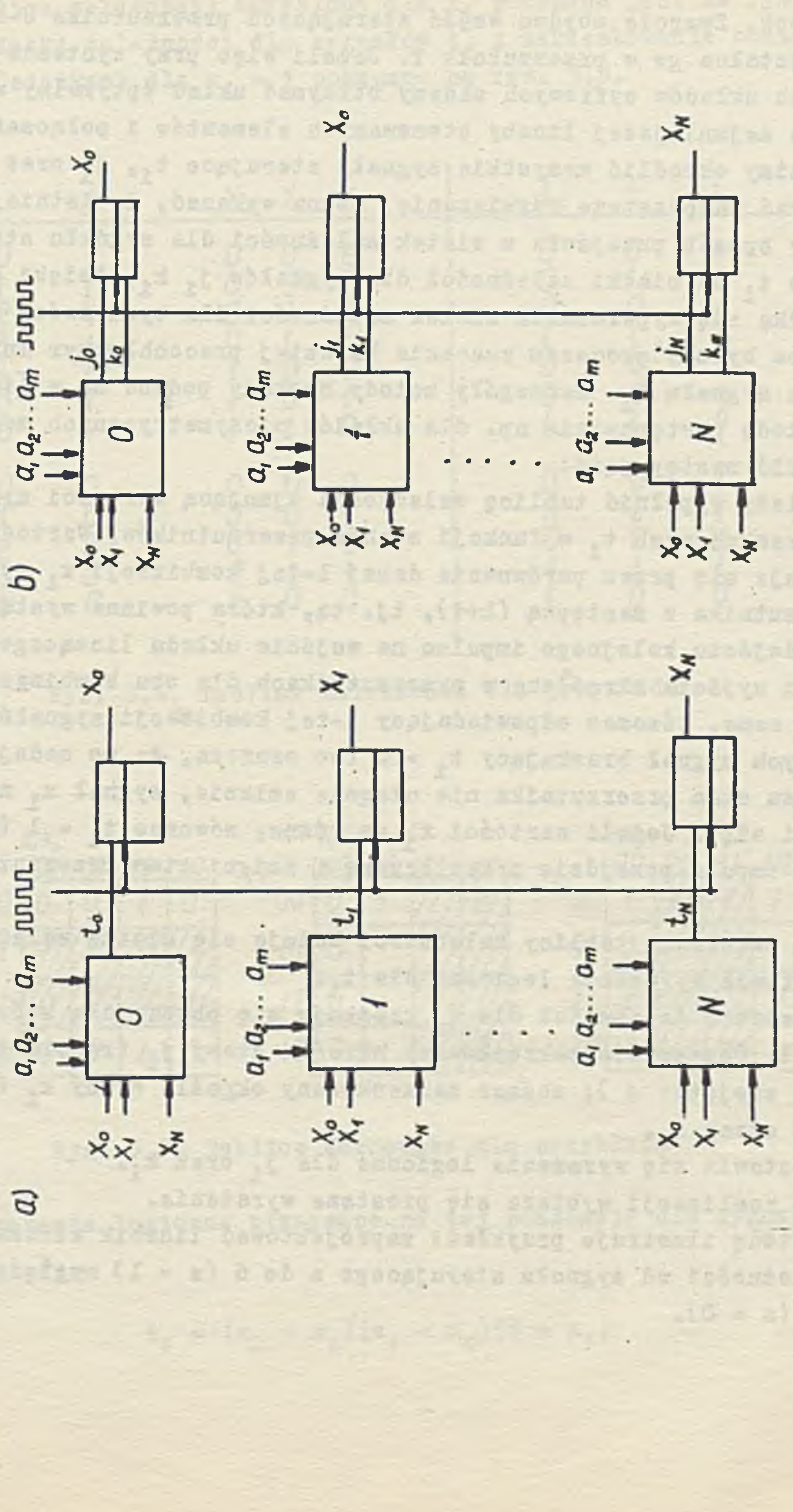
b) tabela przejść

j k	x_{n+1}
0 0	x_{n+1}
0 1	0
1 0	1
1 1	$\overline{x}_n$

Rys. 5.2. Przerzutnik J-K z wejściem zegarowym: a) symbol b) tabela

Punktem wyjścia syntezy układu jest utworzenie tablicy zależności analogicznej do tablic stosowanych przy syntezie wszelkich innych układów kombinacyjnych. Sygnały sterujące określa się w zależności od sygnałów wejściowych na podstawie zadanych warunków pracy układu.

Z tablicy zależności zbudowanej dla układu z przerzutnikiem T określa się wartości sygnału sterującego t_i, a z tablicy dla układu z przerzutnikkem J-K - dwa sygnały sterujące, j_i oraz k_i. Z tablic tych przechodzi się na siatki zależności, na podstawie których określa się wyrażenia logiczne dla sygnałów ste-



Rys. 5.3. Schematy blokowe układów cyfrowych z przerzutnikami z wejściem zegarowym: a) typu T; b) typu J-K

rujących. Zwarcie obydwu wejść sterujących przerzutnika J-K przekształca go w przerzutnik T. Jeżeli więc przy syntezie omawianych układów cyfrowych chcemy otrzymać układ optymalny w sensie najmniejszej liczby stosowanych elementów i połączeń, to musimy określić wszystkie sygnały sterujące t_i, j_i oraz k_i i wybrać najprostsze rozwiązanie. Można wykazać, że istnieje prosty sposób przejścia z siatek zależności dla sygnału sterującego t_i na siatki zależności dla sygnałów j_i k_i, dzięki czemu unika się wypełniania siatek zależności dla tych dwóch sygnałów, co byłoby procesem znacznie bardziej pracochłonnym aniżeli dla sygnału t_i. Szczegóły metody syntezy podane są w [16].

Metodę postępowania np. dla układów niesymetrycznych można określić następująco:

Należy wypełnić tablicę zależności ujmującą wartości sygnałów bramkujących t_i w funkcji stanów przerzutników. Wartości te uzyskuje się przez porównanie danej l-tej kombinacji x_i wyjść przerzutnika z następną (l+1), tj. tą, która powinna wystąpić po nadejściu kolejnego impulsu na wejście układu liczącego. Jeżeli wyjścia określone w przerzutnikach dla obu kombinacji są te same, wówczas odpowiadający l-tej kombinacji sygnałów wejściowych sygnał bramkujący $t_i = 0$ (co oznacza, że po nadejściu impulsu stan przerzutnika nie ulegnie zmianie, sygnał x_i nie zmieni się). Jeżeli wartości x_i są różne, wówczas $t_i = 1$ (kolejny impuls przejdzie przez bramkę i zmieni stan przerzutnika).

Na podstawie tablicy zależności buduje się siatkę zależności i zestawia wyrażenia logiczne dla t_i.

W siatce zależności dla t_i kreskuje się obszar dla którego $x_i = 1$. Obszar nie zakreskowany określa stany j_i (reszta to stany obojętne ϕ); obszar zakreskowany określa stany k_i (reszta to stany ϕ .

Zestawia się wyrażenia logiczne dla j_i oraz k_i.

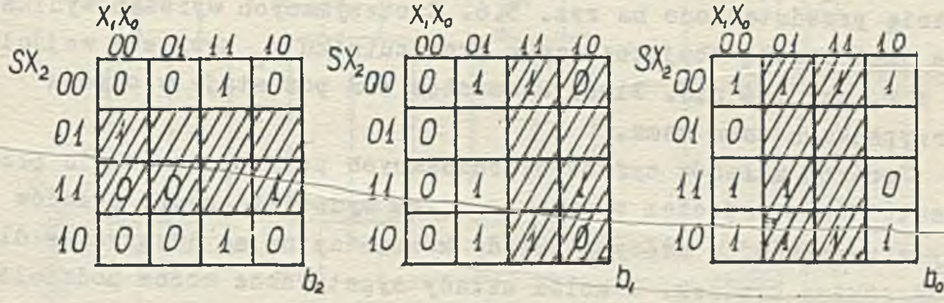
Do realizacji wybiera się prostsze wyrażenia.

Metodę ilustruje przykład: zaprojektować licznik zliczający w zależności od sygnału sterującego s do 6 (s = 1) względnie do 4 (s = 0).

Tablica zależności określona dla t_1 pokazana jest na rys. 5.4. Siatki zależności dla sygnałów t_1 i zakreskowanie obszaru w tych siatkach dla $x_1 = 1$ pokazano na rys. 5.5.

	j	x_2	x_1	x_0	t_2	t_1	t_0
0	1	0	0	0	0	0	1
1	1	0	0	1	0	1	1
2	1	0	1	0	0	0	1
3	1	0	1	1	1	1	1
4	1	1	0	0	0	0	1
5	1	1	0	1	0	1	1
6	1	1	1	0	1	1	0
0	0	0	0	0	0	0	1
1	0	0	0	1	0	1	1
2	0	0	1	0	0	0	1
3	0	0	1	1	1	1	1
4	0	1	0	0	1	0	0

Rys. 5.4. Tablica zależności dla przykładu



Rys. 5.5. Tablice Karnaugha dla przykładu

Wyrażenia logiczne otrzymane na tej podstawie dla sygnałów t_1, j_1, k_1

$$t_2 = (x_2 + x_0)(x_1 + x_0)(\bar{s} + x_1)$$

$$t_1 = (x_1 + x_0)(x_2 + x_0)$$

$$t_0 = (s + x_2)(\bar{x}_2 + x_1)$$

$$j_2 = x_1 x_0$$

$$j_1 = x_0$$

$$j_0 = (s + x_2)(\bar{x}_2 + \bar{x}_1)$$

$$k_2 = \bar{\bar{s}} + x_1$$

$$k_1 = x_2 + x_0$$

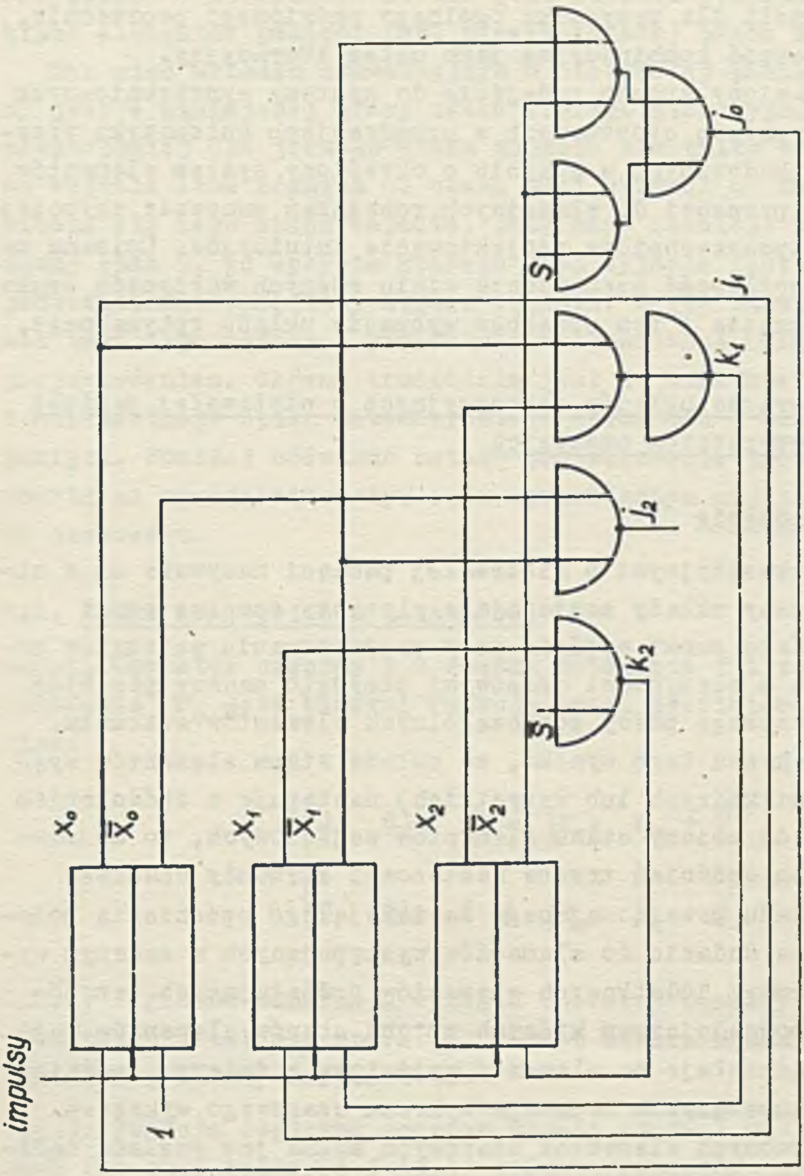
$$k_0 = 1$$

Łatwo sprawdzić, że do realizacji wyrażeń określających sygnały t_2, t_1, t_0 potrzeba 10 elementów NOR, a do otrzymania sygnałów j_2, j_1, j_0, k_2, k_1, k_0 – 8 elementów. To drugie rozwiązanie przedstawiono na rys. 5.6. Z otrzymanych wyrażeń wynika, że jest rzeczą obojętną czy w przerzutniku x_0 zewrzemy wejścia j i k, czy też nie. Ilość elementów NOR pozostaje w obydwu przypadkach taka sama.

Spośród układów cyfrowych budowanych przy zastosowaniu przerzutników z wejściem zegarowym można wydzielić grupę układów symetrycznych, w których układy kombinacyjne są identyczne dla wszystkich komórek. Z kolei układy symetryczne można podzielić na dwa rodzaje:

a) układy w których o zmianie stanu n-tego przerzutnika decydują stany wszystkich "poprzedzających" go pozycji (liczniki, akumulatory);

b) układy w których o zmianie stanu n-tego przerzutnika decyduje jedynie stan "poprzedniej" pozycji (rejestry).



Rys. 5.6. Schemat logiczny układu z przykładu

Można dla obydwu tych rodzajów określić metody syntezy prostsze, aniżeli dla przypadku ogólnego omówionego poprzednio, traktując część kombinacyjną jako układ iteracyjny.

Przedstawiony sposób podejścia do syntezy synchronicznych układów liczących stosowanych w urządzeniach automatyki przemysłowej i budowanych w oparciu o określony system elementów logicznych prowadzi do właściwych rozwiązań znacznie szybciej aniżeli rozpowszechnione projektowanie intuicyjne. Opisana metoda daje możliwość sprawdzenia wielu różnych wariantów szukanego rozwiązania i tym sposobem wybranie układu optymalnego.

6. Projektowanie układów sekwencyjnych o nietrwałej pamięci metodą operatorów czasowych

6.1. Wprowadzenie

Układami sekwencyjnymi o nietrwałej pamięci nazywane są w niniejszej pracy układy zawierające elementy czasowe czyli zwłoczne. Jako punkt wyjścia przy projektowaniu automatów sekwencyjnych z elementami czasowymi przyjęto zadany przebieg wykresu czasowego pracy poszczególnych elementów automatu. Jeżeli z wykresu tego wynika, że zmiana stanu elementów wyjściowych (niektórych lub wszystkich) następuje z opóźnieniem w stosunku do zmiany stanu elementów wejściowych, to do realizacji tych opóźnień trzeba zastosować elementy czasowe. Synteza układu przełączającego zawierającego opóźnienia polega więc na dodaniu do elementów występujących w zadanym wykresie czasowym dodatkowych elementów opóźniających (czasowych), za pośrednictwem których zmiana stanów elementów wejściowych oddziałuje na elementy wyjściowe z żądanym opóźnieniem. Po uzupełnieniu zadanego wykresu czasowego wykresem pracy potrzebnych elementów czasowych można już znaleźć żądaną funkcję czasową realizującą zadany sygnał wyjściowy.

Praca niniejsza dotyczy tej klasy przypadków, kiedy między sygnałami wejściowymi i wyjściowymi automatu wystarcza pośrednictwo elementów czasowych, czyli z pamięci nietrwałej.

Przypadków w których potrzebne jest również pośrednictwo zwykłych elementów pamięci (np. przerzutników) praca nie omawia.

Tak więc układem sekwencyjnym o nietrwałej pamięci nazywany jest w niniejszej pracy układ którego stan wyjścia zależy przynajmniej dla jednego stanu wejścia nie tylko od tego stanu wejścia lecz również od czasu jaki upłynął od momentu pojawienia się tego stanu wejścia, przy czym istnieje taki skończony czas T, po upływie którego stan wyjścia jest zawsze jednoznacznie określony stanem wejścia. W [4] omówiono obszernie tego typu układy i przedstawiono trudności związane z ich projektowaniem. Główną trudnością jest tu brak operatywnego i adekwantnego opisu matematycznego elementów o nietrwałej pamięci. Poniżej omówiono metodę projektowania tych układów opartą na specjalnie w tym celu wprowadzonym pojęciu operatora czasowego.

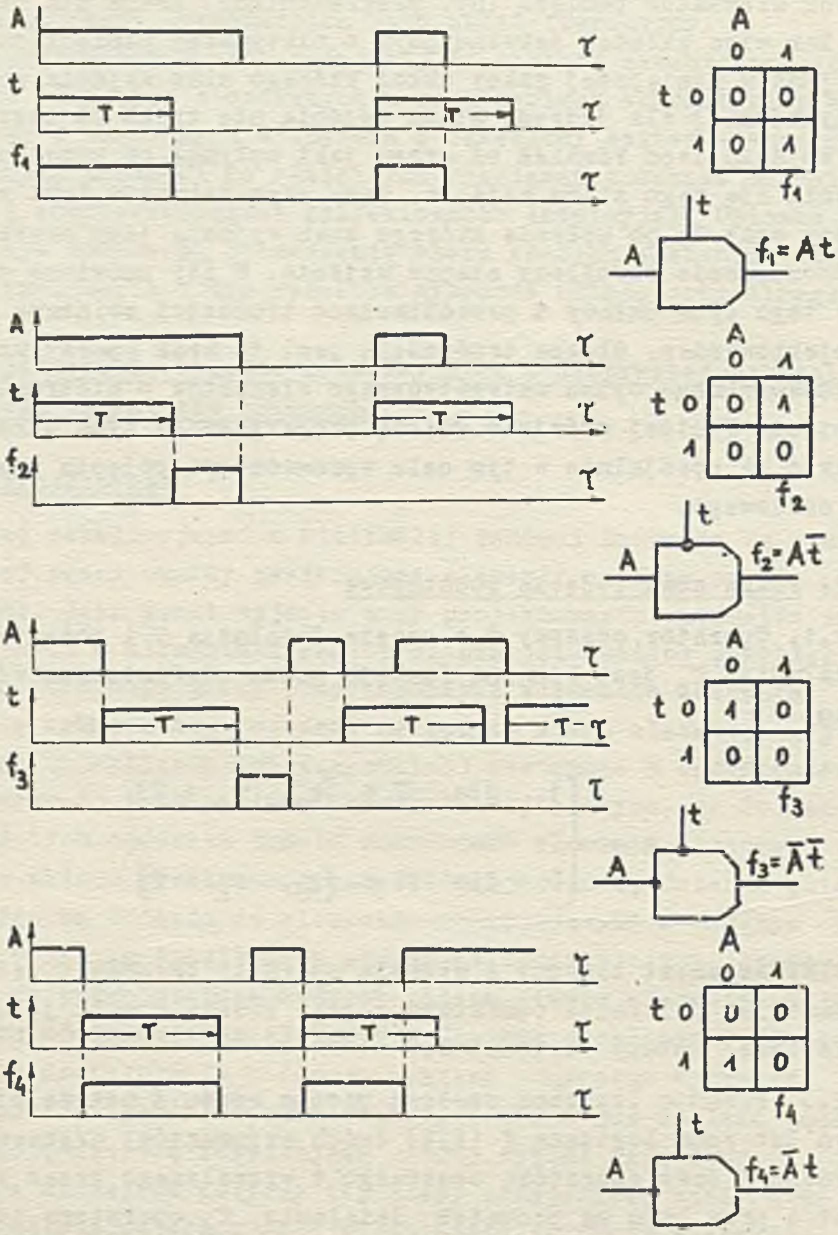
6.2. Pewne nowe pojęcia podstawowe

6.2.1. Operator czasowy t o czasie działania T i początku działania τ_0 jest binarną funkcją czasu zdefiniowaną równaniem:

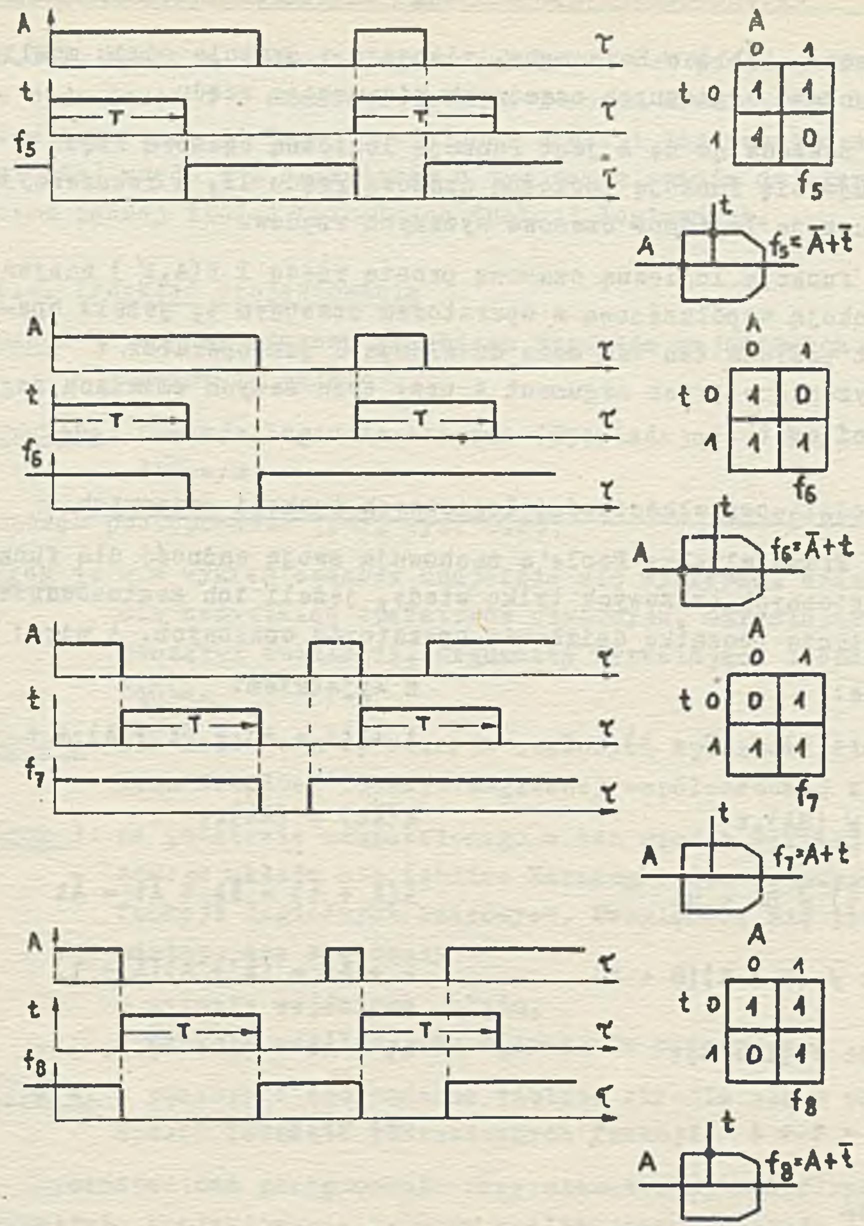
$$t = \begin{cases} 1 & \text{dla} \quad \tau \in [\tau_0, \ \tau_0 + T] \\ 0 & \text{dla} \quad \tau \notin [\tau_0, \ \tau_c + T] \end{cases}$$

6.2.2. Argument binarny A wyzwala operator czasowy t jeżeli początkiem działania operatora t jest zawsze moment jednej określonej zmiany wartości A.

6.2.3. Funkcją logiczną czasową prostą rzędu I nazywa się iloczyn lub sumę logiczną f (A,t) dwóch argumentów: argumentu binarnego A oraz operatora czasówego t wyzwalanego przez argument A przy czym za początek działania τ_0 operatora czasowego t przyjmuje się każdy moment w którym A osiąga wartość A_0 dla której wartość funkcji f(A,t) staje się zależna tylko od operatora czasowego. Rys. 6.1a i 6.1b przedstawiają wykre-



Rys. 6.1a. Funkcje logiczne czasowe proste rzędu I, ich wykresy czasowe, tabela Karnaugha, równania i symbole



Rys. 6.1b. Funkcje logiczne czasowe proste rzędu I, ich wykresy czasowe, tabela Karnaugha, równania i symbole

sy czasowe, tablice Karnaugha, równania i symbole ośmiu możliwych funkcji logicznych czasowych pierwszego rzędu.

6.2.4. Zakładając że A jest funkcją logiczną czasową rzędu I otrzymuje się funkcję logiczną czasową rzędu II, i rekurencyjne - funkcje logiczne czasowe wyższych rzędów.

6.2.5. Funkcję logiczną czasową prostą rzędu I $f(A,t')$ nazywa się funkcją współczasową z operatorem czasowym t, jeżeli operator t posiada ten sam czas działania T jak operator t i jest wyzwalany przez argument A przy tych samych zmianach jego wartości co t'.

6.3. Podstawowe właściwości logicznych funkcji czasowych

6.3.1. Prawa algebry Boole'a zachowują swoją ważność dla funkcji logicznych czasowych tylko wtedy, jeżeli ich zastosowanie nie zmienia początku działania operatorów czasowych. A więc:

ogólnie:	z wyjątkiem:
$B + (A + t) \neq (B + A) + t$	$A + (A + t) = (A + A) + t$
$B(At) \neq (BA).t$	$A(At) = (AA).t$
$B(A + t) \neq BA + Bt$	$\bar{A}(A + t) = \bar{A}A + \bar{A}t = \bar{A}t$
$B + At \neq (B + A)(B + t)$	$\bar{A} + At = (\bar{A} + A)(\bar{A} + t)$
$At + Bt \neq (A + B)t$	$At + At = (A + A).t$
$A + t = t + A$	bez wyjątku
$At = tA$	"
$\overline{At} = \bar{A} + \bar{t}$	"
$\overline{A + t} = \bar{A}\,\bar{t}$	"

6.3.2. Każdą z funkcji logicznych czasowych prostych rzędu I z operatorem t o czasie działania T można zrealizować na drodze superpozycji dozwolonej innej funkcji logicznej czasowej prostej rzędu I z operatorem o tym samym czasie działania T oraz pewnej ilości normalnych funkcji logicznych.

6.4. Przykład projektowania

Dane: wykres czasowy przebiegu sygnałów wejściowych i wyjściowych układu

Szukane: funkcje logiczne czasowe odpowiadające sygnałom wyjściowym

Sposób postępowania (por. rys. 6.2):

Krok 1: w/w wykres czasowy uzupełnia się wykresami czasowymi dla wszystkich operatorów czasowych, określając ich początki działania, argumenty wyzwalające i czasy działania.

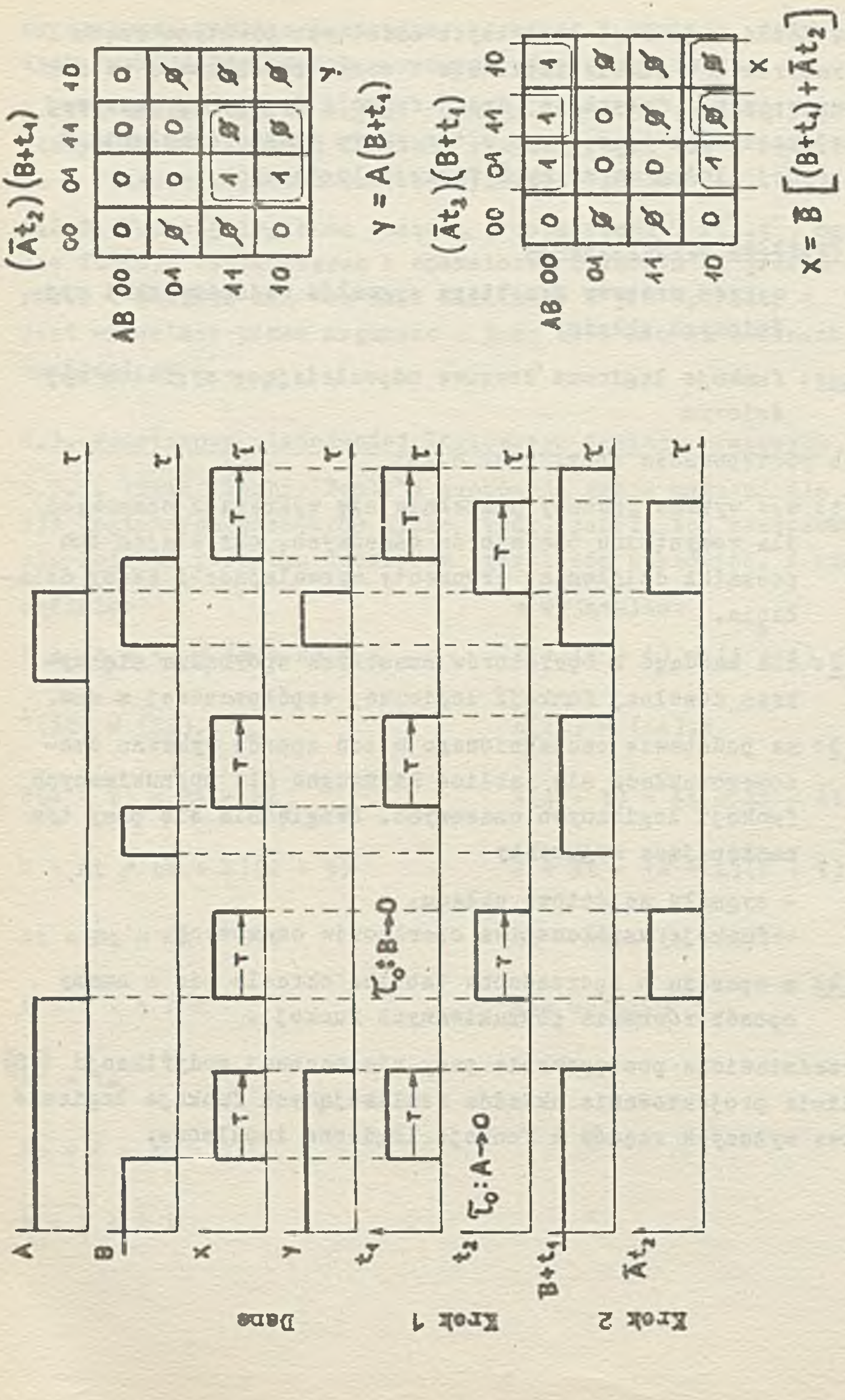
Krok 2: dla każdego z operatorów czasowych sporządza się wykres dowolnej funkcji logicznej współczasowej z nim.

Krok 3: na podstawie uzupełnionego w ten sposób wykresu czasowego układa się tablice Karnaugha dla poszukiwanych funkcji logicznych czasowych. Uwzględnia się przy tym następujące argumenty

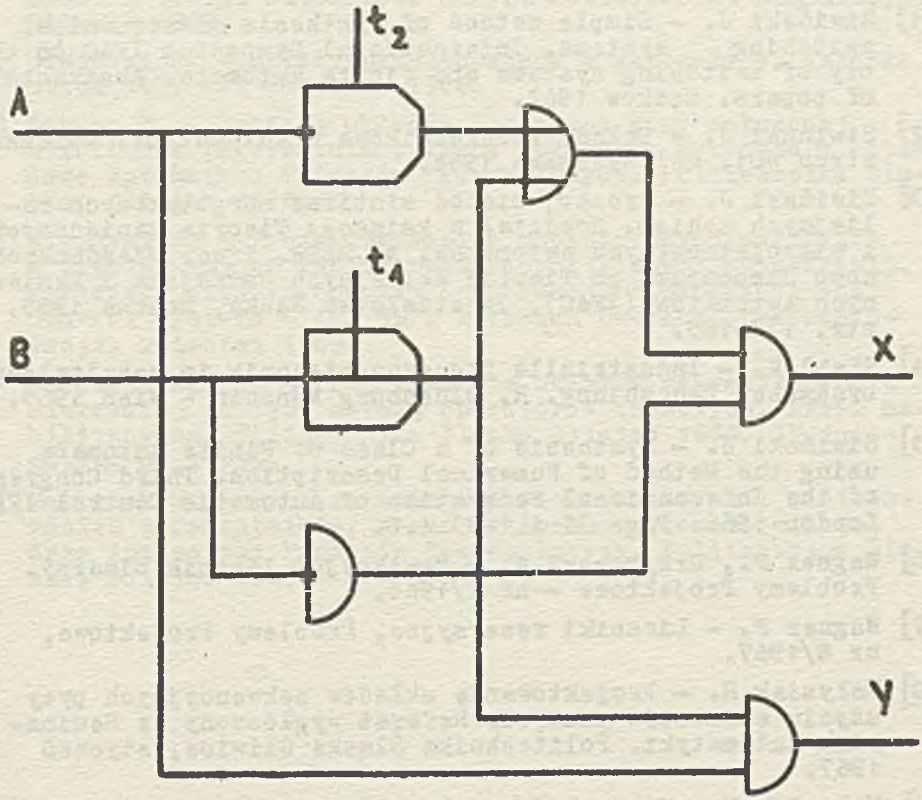
- sygnały wejściowe układu,
- funkcje współczasowe operatorów czasowych.

Krok 4: w oparciu o sporządzone tablice określa się w znany sposób równania poszukiwanych funkcji.

Przedstawione postępowanie przy nieznacznej modyfikacji [18] umożliwia projektowanie układów realizujących funkcje logiczne czasowe wyższych rzędów i funkcje logiczne impulsowe.



Rys. 6.2a. Przebieg rozwiązania dla przykładu w p.4



Rys. 6.2b. Schemat logiczny otrzymanego układu

LITERATURA

[1] Siwiński J. - Simple method of synthesis of sequential switching systems. International Symposium IFAC on theory of switching systems and finite automata. Abstracts of popers. Moskow 1962.

[2] Siwiński J. - Układy przekaźnikowe w automatyce. Książka, stron 401. WNT Warszawa 1964.

[3] Siwiński J. - Prostoj mietod sintieza mnogotaktnych reliejnych schiem. Rozdział w książce: Tieoria kanieoznych i wierojatnostnych awtomatów. AN.SSSR. Trudy Mieżdunarodnowo Simpozjuma po Tieorii Reliejnych Ustrojstw i kanieoznych Awtomatów (IFAC). Izdatielstwo Nauka, Moskwa 1965. str. 177-185.

[4] Stahl K. - Industrielle Steuerungstechnik in schaltalgebraischer Behandlung. R. Oldenburg München - Wien 1965.

[5] Siwiński J. - Synthesis of a Class of Finite Automata using the Method of Numerical Description. Third Congress of the International Federation of Automatic Control IFAC. London 1966. Page 35 A.1-35 A.8.

[6] Wagner F., Grzybowski W. - Rewersyjny licznik binarny. Problemy Projektowe - nr 1/1966.

[7] Wagner F. - Liczniki rewersyjne, Problemy Projektowe, nr 8/1967.

[8] Małysiak H. - Projektowanie układów sekwencyjnych przy użyciu elementów pamięci. Referat wygłoszony na Seminarium Automatyki. Politechnika Śląska Gliwice, styczeń 1967.

[9] Małysiak H., Niederliński A. - Pewne zagadnienia projektowania złożonych układów sekwencyjnych. Materiały Konferencji Naukowo-Technicznej. Komisja Automatyki OW NOT Katowice 1967.

[10] Siwiński J. - Synteza automatów sekwencyjnych za pomocą tablic kolejności łączeń. Materiały Konferencji Naukowo-Technicznej Komisji Automatyki OW NOT Katowice. Katowice 1967.

[11] Małysiak H. - Przegląd pneumatycznych elementów logicznych stosowanych w przemysłowych układach sterowania. Materiały Konferencji Naukowo-Technicznej Komisji Automatyki OW NOT Katowice. Katowice 1967.

[12] Niederliński A. - Projektowanie układów sekwencyjnych o nietrwałej pamięci przy użyciu operatorów czasowych. Materiały Konferencji Naukowo-Technicznej Komisji Automatyki OW NOT Katowice. Katowice 1967.

[13] Wagner F., Grzybowski W. - Zastosowanie układów cyfrowych do sterowania napędów walcowni. Praca Konferencji pt. "Układy logiczne w automatyzacji przemysłu", Katowice, 1967.

[14] Małysiak H., Niederliński A., Wagner F., Ziaja M. - Zbiór zadań z teorii automatów. Skrypt Politechniki Śląskiej, Gliwice 1967.

[15] Siwiński J. - Układy przełączające w automatyce. Książka, stron 547, WNT Warszawa 1968.

[16] Wagner F. - Synteza układów liczących przy założonej strukturze przerzutnika. Praca doktorska wykonana w Katedrze Automatyki Procesów Przemysłowych. Politechnika Śląska, Gliwice 1968.

[17] Siwiński J. - Projektirowanije mnogotaktnych awtomatów z ispolzowanijem tablic posliedowatielnosti wklinczenij. Materiały Międzynarodowej Konferencji "Automation and Control Systems - ARS 69". Morawska Ostrawa, Czechosłowacja, kwiecień 1969.

[18] Niederliński A. - Projektowanie układów sekwencyjnych o nietrwałej pamięci metodą operatorów czasowych. Praca habilitacyjna. Politechnika Śląska Gliwice 1969. ZN Automatyka nr 11/69.

[19] Małysiak H. - Synteza automatów sekwencyjnych z zastosowaniem przerzutników. Praca doktorska wykonana w Katedrze Automatyki Procesów Przemysłowych. Politechnika Śląska, Gliwice 1969.