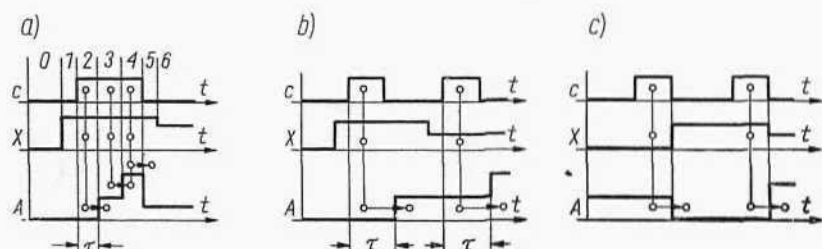


waniem co trzeciego impulsu (może służyć jako dzielnik częstotliwości). Mimo, że stany A , oznaczone przez 0,2,4, niczym się na rysunku nie różnią (podobnie jak 1 i 3), to jednak muszą być nazwane różnie, gdyż właśnie rolą stanu wewnętrznego jest zapamiętanie sytuacji przeszłych, jeśli ma to znaczenie dla stanów przyszłych. Na rys. 4-12 przedstawiono odpowiednie tablice układu statycznego i dynamicznego. Również i ten układ można uważać za synchroniczny autonomiczny, jeśli wstępnie określi się y za pomocą x i A .

4.2. UKŁADY SYNCHRONICZNE

4.2.1. PROBLEMY SYNCHRONIZACJI

Cechą charakterystyczną układów synchronicznych jest obecność impulsów taktujących, które mają powodować zmiany stanu wewnętrznego A tylko w ściśle określonych momentach. Impulsy taktujące są



Rys. 4-13. Różne przypadki oddziaływania impulsów synchronizujących:
a) proste bramkowanie; b) opóźnienie reakcji pamięci; c) zmiana stanu pamięci tylnym zboczem

więc jak gdyby impulsami strobuującymi, pobierającymi próbki sygnałów X i A , i kierującymi je do elementów pamięci. Tę rolę impulsów taktujących najprościej byłoby zrealizować przez bramkowanie (w zespole elementów I) wszystkich sygnałów wzbudzeń sygnałem taktowania c . Jednakże takie proste rozwiązanie nie może być zastosowane, a powody wyjaśnia rys. 4-13a (stany X i A przedstawiono tu w postaci sygnałów wielowartościowych, dla odróżnienia od sygnałów x i Q). Gdy $c = 0$ (takty¹⁾

¹⁾ Są to właściwie mikrotakty — składowe części taktu układu, wyznaczonego

0,1,5,6) — wejścia są odseparowane od elementów pamięci, więc zmiany X nie powodują zmian A , zgodnie z zaleceniami. Pojawienie się $c = 1$ (takt 2) sprawia, że aktualne stany X i A wyznaczają, jak nakazuje funkcja δ , następny stan A . Pojawia się on po czasie τ , będącym czasem opóźnienia wnoszonego przez zaangażowane w zmianę elementy. Na rysunku kółkami i linią łączącą pokazano sposób tworzenia nowego stanu A . Tak więc w takcie 3 pojawi się stan A , wyznaczony w tablicy przejść parą (X, A) z taktu 2. Ale w takcie 3 trwa jeszcze impuls c , nic więc nie stoi na przeszkodzie, by — zgodnie z tablicą przejść — para (X, A) z tego taktu wyprodukowała następny stan A , który pojawi się po czasie τ w takcie 4. W taki sam sposób zostanie określony (taktem 4) stan A w takcie 5. Zniknięcie impulsu c przerywa ten proces. Impuls c trwał tu równo 3τ , więc i stan A zmienił się trzy razy; w praktyce czas τ nie jest stały, czas trwania impulsu c też ulega wahaniom, więc liczba dokonanych przejść byłaby zupełnie przypadkowa. Ponieważ prawidłowa praca układu polega na wykonaniu jednego przejścia w jednym takcie (czyli okresie c) — opisane działanie nie może mieć miejsca.

Sposób na stabilizację przejść wynika bezpośrednio z przedstawionej analizy czasowej, po prostu czas τ powinien być większy od czasu trwania impulsu taktującego T_i , gdyż wtedy nowy stan A pojawi się w przerwie c , gdy elementy pamięci nie reagują na sygnały wzbudzeń. Dokładniej, powinno być

$$T_i \leq \tau \leq T_i + T_p$$

przy czym T_p oznacza czas przerwy w sygnale c . Zmiany stanu A następują tak, jak na rys. 4-13b.

Spełnienie powyższych nierówności może być osiągnięte dwiema drogami:

—przez skrócenie czasu T_i poniżej naturalnych opóźnień τ , właściwych dla danych elementów albo

—przez sztuczne wydłużenie czasu reakcji elementów pamięciowych τ .

Pierwsza droga jest bardzo ryzykowna, gdyż przy zbyt krótkich impulsach c wywołanie jakiegokolwiek zmiany A staje się niemożliwe, a ponadto naturalna wartość τ jest różna nie tylko dla różnych elemen-

tów, lecz także dla różnych stanów A w tym samym układzie. Metoda ta nie jest stosowana.

Druga droga ma dwa warianty, wykorzystywane w praktyce:

—można zastosować w układach pamięciowych linie opóźniające (zwykle o takt, i wtedy $\tau = T_i + T_p$) albo

— można zmieniać stan A tylnym zboczem impulsu c (i wtedy $\tau = T_i$).

Drugie rozwiązanie jest prostsze, daje więcej możliwości realizacji układowych i dlatego praktycznie wyparło już z zastosowań rozwiązanie pierwsze. Przykład działania układu z wyzwaniem tylnym zboczem jest przedstawiony na rys. 4-13c. Fakt, że zmiana stanu następuje tu właśnie przy zmianie c z 1 na 0, należy traktować jako efekt umowy, wynikający z opisanej interpretacji bramkowania; inny sposób bramkowania, lub negator na drodze sygnału c może sprawić, że zmiany A następują przy zmianie c z 0 na 1. Dla celów porządkowych będzie wygodnie jednak przyjąć, że zmiana od tylnego zbocza impulsu c odpowiada sygnałowi c , a zmiana od poprzedniego zbocza — sygnałowi \bar{c} .

Na działanie układu może mieć wpływ moment zmiany stanu X . W układzie całkowicie synchronicznym zmiany te następują w tych samych momentach co zmiany A (rys. 4-13c), gdyż spowodowane są tymi samymi impulsami taktującymi. Bywają jednak układy, w których zmiany X nie są synchronizowane. Jeśli następują one w przerwach c — układ pracuje jak całkowicie synchroniczny. Jeśli zmiany X mogą następować w czasie trwania impulsu — należy to uwzględnić w rozwiązaniu technicznym, gdyż przerzutniki wprowadzają ograniczenia na moment zmian X .

Momenty zmian X i A wpływają na postać sygnałów Y . W układach Moore'a oraz w całkowicie synchronicznych układach Mealy'ego stan Y zmienia się razem z A , zgodnie z tablicą wyjść, nie częściej niż raz w jednym takcie. W przypadkach z niesynchronizowanymi zmianami X budowanie układów Mealy'ego napotyka trudności, gdyż X i A zmieniają się w różnych momentach, a więc stan Y , zależny od X i A , zmieniałby się dwa razy w każdym takcie (jak w przypadku przedstawionym na rys. 4-13b). Gdy zmiany X następują w przerwach przebiegu c , można sygnał c wykorzystać do bramkowania sygnałów wyjściowych, uzyskując właściwy stan Y , gdy $c = 1$ i stan zerowy Y , gdy $c = 0$. Nie zawsze

rozwiązanie takie jest możliwe do przyjęcia. W innych przypadkach trzeba indywidualnie analizować zależność sygnałów y od x , c i A .

4.2.2. MINIMALIZACJA TABLIC PRZEJŚĆ

Tablice przejść i wyjść, uzyskane metodami opisanymi w p. 3.1, nie zawsze mają najmniejszą możliwą liczbę wierszy (czyli stanów A), gdyż nie zawsze ujawnienie równoważnych gałęzi grafu jest łatwe. Liczba stanów wewnętrznych układu wpływa w istotny sposób na stopień złożoności realizacji technicznej, gdyż — w ogólnym przypadku — im więcej jest stanów A , tym więcej musi być elementów pamięciowych i związanych z nimi elementów kombinacyjnych. Wynika stąd ogólna reguła (mająca nieliczne wyjątki), że układ o mniejszej liczbie stanów wewnętrznych A jest prostszy. Istnieje zatem konieczność zminimalizowania liczby stanów A w każdej wstępnie uzyskanej tablicy przejść.

a)

A					Y
3	1	—	4	—	10
4	—	2	4	—	—0

b)

A					Y
(3,4)	1	2	4	—	10

c)

A					Y
3	1,0	5,1	2,1	—	
4	—	6,—	2,1	4,0	

d)

A					Y
(3,4)	1,0	(5,6),1	2,1	4,0	

Rys. 4-14. Przykłady zastępowania dwóch wierszy tablicy jednym, w układach Moore'a (a,b) i Mealy'ego (c,d)

Proces minimalizacji liczby stanów wewnętrznych polega na zastępowaniu dwóch (lub więcej) wierszy tablicy przejść — jednym, bez zmiany sposobu działania układu. Przykłady takiej zamiany przedstawia rys. 4-14. W urywku tablicy układu Moore'a wartości nieokreślone stanów wewnętrznych i sygnału wyjściowego można zastąpić takimi konkretnymi wartościami, że obydwa wiersze staną się identyczne, a więc stany 3 i 4 można uważać za jeden stan, przejściowo nazwany (3,4) — rys. 4-14b.

W urywku tablicy układu Mealy'ego (c) również można zamiast kresek wstawić takie wartości A i Y , by obydwa wiersze były identyczne, z wyjątkiem drugiej kolumny, ze stanami 5 i 6. Jeśli jednak badanie odpowiednich wierszy tablicy wykaże, że 5 i 6 można połączyć w jeden stan (5,6), to można będzie połączyć również 1 i 2 (rys. 4-14d). Jest to przykład warunkowego zastępowania dwóch stanów — jednym.

Dla częściowego sformalizowania procesu minimalizacji celowe będzie wprowadzenie kilku pomocniczych określeń.

Stany A i A' są *niesprzeczne*, gdy są identyczne lub jeden z nich jest nieokreślony (np. 5 i 5, 6 i —, — i —).

Stany Y i Y' są *niesprzeczne*, gdy odpowiednie ich sygnały (y_1, y_2, \dots, y_m) są parami identyczne lub jeden jest nieokreślony (np. 101 i 1-1, 0-1- i -1-0).

a)						
A	X_1	X_2	X_3	X_4		
	1	1,1	2,1	—	3,0	A', Y
	2	1,0	—	4,0	2,1	
	3	3,-	4,1	2,1	1,0	
	4	1,0	1,1	2,-	—	

b)						
A	X_1	X_2	X_3	X_4		
	1,3	1,3;1	2,4;1	2,1	1,3;0	A', Y
	2,1	1,0	1,1	2,4;0	2,1	

c)						
A	X_1	X_2	X_3	X_4		
	(1,3)0	0,1	1,1	1,1	0,0	A', Y
	(2,4)1	0,0	0,1	1,1	1,1	

Ry. 4-15. Przykład minimalizacji tablicy przejść przez połączenie stanów zgodnych

Stany A i A' są *zgodne*, gdy można je zastąpić jednym stanem, bez wpływu na działanie układu.

Przykłady z rys. 4-14 i dokładniejsza analiza wykazują, że stany A i A' można, na podstawie tablicy przejść, uznać za zgodne, gdy:

- 1) pod wpływem każdego X przechodzą w stany niesprzeczne lub zgodne oraz
- 2) dla każdego X odpowiadające im stany Y są niesprzeczne.

Na przykład w tablicy z rys. 4-15a stany 1 i 2 nie są zgodne, gdyż przy X_1 i X_4 produkują sprzeczne sygnały Y . Stany 1 i 3 są zgodne warunkowo — jeśli są zgodne 2 i 4 (oraz 1 i 3, ale ten warunek — jako nie-

istotny — opuszcza się). Ze względu na sprzeczne sygnały wyjściowe niezgodne są także pary 1,4 i 2,3. Stany 2 i 4 są zgodne, natomiast stany 3 i 4 są niezgodne, gdyż pod wpływem X_2 przechodzą w stany niezgodne 4 i 1. Tak więc w tablicy występują dwie pary stanów zgodnych: 2,4 i 1,3. Zastępując każdą z tych par jednym stanem uzyskuje się tablicę z rys. 4-15b, a wprowadzając nowe oznaczenia stanów wewnętrznych — tablicę z rys. 4-15c.

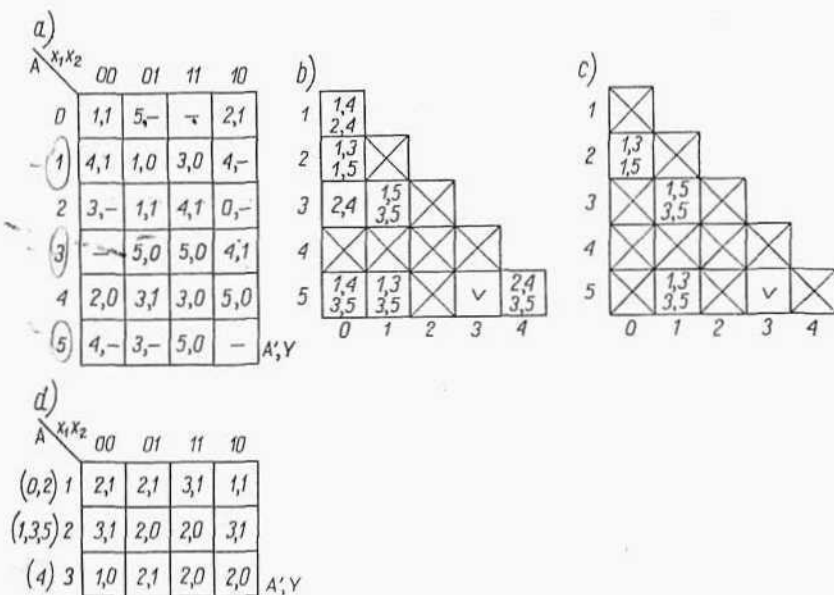
Już ten prosty przykład obrazuje trudności, jakie mogą wynikać przy poszukiwaniu stanów zgodnych, gdy zgodność jednej pary zależy od całego łańcuszka zgodności innych par. Do tego dochodzi jeszcze problem łączenia stanów zgodnych w zbiory większe od dwóch, np. gdy ze zgodnych par (1,3), (3,5) i (1,5) tworzy się zbiór stanów zgodnych {1,3,5}. Ułatwienie tych prac, związanych z poszukiwaniem maksymalnych zbiorów stanów zgodnych, można uzyskać posługując się tzw. *trójkątną tablicą zgodności (metoda Unger-Paulla)*. Zasady tej metody zostaną wyjaśnione w procesie minimalizacji tablicy z rys. 4-16a.

1. Należy wypełnić trójkątną tablicę z rys. 4-16b, wpisując w kratkę o współrzędnych a, b znaczek V — jeśli stany a, b są zgodne, wykreślając tę kratkę — gdy są *jawnie niezgodne* (tzn. mają sprzeczne wyjścia), albo wpisując odpowiednie pary (c, d), gdy zgodność (a, b) zależy od zgodności (c, d).

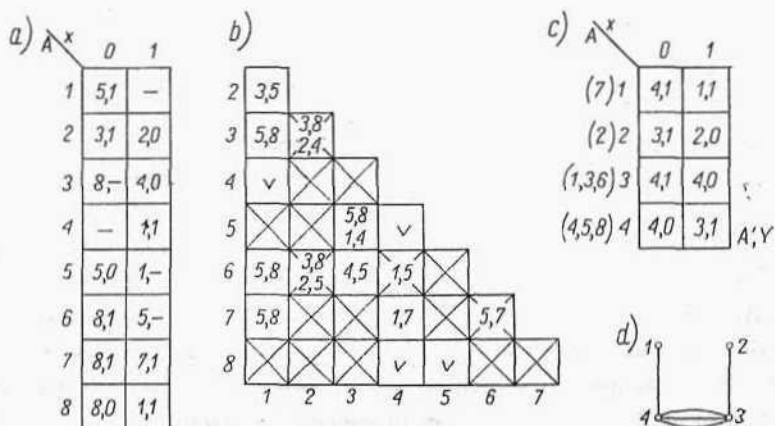
2. Sprawdza się, czy pary wewnątrz tablicy nie odpowiadają kratkom wykreślonym; jeśli tak wykreśla się nowe kratki, aż do uzyskania sytuacji, gdy żadna para nie może być wykreślona. Na przykład kratkę o współrzędnych 0,1 (rys. 4-16b) trzeba wykreślić, gdyż pary w niej umieszczone nie są zgodne — zostały już wykreślone. W ten sposób uzyskuje się tablicę trójkątną jak na rys. 4-16c.

3. Biorąc pod uwagę tylko kratki nie wykreślone (bez względu na ich zawartość) wypisuje się współrzędne kratki, lub kratek, z pierwszej niecałkowicie wykreślonej kolumny o najwyższym numerze (tzn. z prawej strony tablicy). W rozważanym przykładzie będzie to {3, 5}.

W następnej niewykreślonej kolumnie (o mniejszym numerze) sprawdza się zbiór wierszy o niewykreślonych kratkach. Jeśli w zbiorze tym mieszczą się zbiory poprzednio wypisane, rozszerza się je dopisując numer kolumny. Pozostałe kratki opisuje się nowymi parami współrzędnych. Czynności te powtarza się dla wszystkich dalszych kolumn tablicy.



Rys. 4-16. Przykład minimalizacji tablicy przejść z zastosowaniem trójkątnej tablicy zgodności



Rys. 4-17. Przykład minimalizacji tablicy przejść

W przypadku tablicy z rys. 4-16c, po wypisaniu pary $\{3,5\}$ z kolumny 3, należy rozpatrzyć kolumnę 1. Zbiór wierszy z niewykreślonymi kratkami wynosi dla niej $\{3,5\}$, a więc zbiór stanów zgodnych powiększy się do $\{1,3,5\}$. Następna kolumna 0 ma wykreślone kratki w wierszach 1,3 i 5, więc zbiór ten nie będzie dalej rozszerzany, dochodzi natomiast nowa para stanów zgodnych — $\{0,2\}$.

4. Postępowanie kończy się wypisaniem zbioru zbiorów (klasy) stanów zgodnych, zawierającego wszystkie stany układu, a więc także zbiory jednoelementowe. Klasa zawierająca wszystkie stany i wszystkie maksymalne zbiory stanów zgodnych będzie oznaczana przez Φ . W rozważywanym przykładzie jest

$$\Phi = \{1, 3, 5; 0,2; 4\}$$

Ponieważ w uzyskanej klasie Φ żaden stan nie występuje więcej niż raz, można ją uznać za klasę minimalną (Φ_{min}). Zastępując każdy jej zbiór jednym stanem buduje się nową, minimalną tablicę przejść (rys. 4-16d).

Niekiedy klasa Φ zawiera powtarzające się stany, a wówczas proces minimalizacji trzeba rozszerzyć o etap redukcji Φ . Zostanie to wyjaśnione na przykładzie minimalizacji tablicy z rys. 4.17a. Oto poszczególne etapy:

1,2 — tablica z rys. 4-17b

3 — kolumna 5: 5,8;

4: 4,5,8; 4,7;

3: 3,5; 3,6;

1: 1,4,7; 1,3,6; 1,2;

4 — $\Phi = \{4,5,8; 1,4,7; 3,5; 1,3,6; 1,2\}$

Powtarzające się stany 1,2,3 i 4 nasuwają przypuszczenie, że ta postać Φ może być uproszczona. Uproszczenie ma istotne znaczenie, gdy umożliwi usunięcie całego zbioru z klasy, gdyż może to zmniejszyć liczbę elementów pamięciowych, ale usunięcie pojedynczych stanów ze zbiorów też może być korzystne, gdyż niekiedy ujawnia stany nieokreślone.

Redukując klasę Φ należy pamiętać, że każda jej prostsza postać musi spełniać następujące warunki:

- a) zawiera wszystkie stany A (tzn. jest pełna),
 b) dla każdej pary każdego zbioru są spełnione warunki zgodności warunkowej.

Redukcję Φ można rozpocząć od kontroli warunku a), podkreślając te stany, które występują w Φ tylko raz:

$$\Phi = \{4,5,8; 1,4,7; 3,5; 1,3,6; 1,2\}$$

Bez naruszenia warunku a) może więc być usunięty tylko zbiór $\{3,5\}$, trzeba jednak sprawdzić, czy nie naruszy to warunku b). Bardzo jest w tym pomocna tablica trójkątna, z której wynika, że 3,5 mieści się w polu o współrzędnych 1,2, a więc wykreślenie zbioru $\{3,5\}$ z Φ czyni niezgodnym zbiór $\{1,2\}$, którego wykreślić nie można. Można jednak zauważyć, że w zbiorze $\{1,2\}$ tylko stan 2 jest niezbędny i gdyby usunąć zeń stan 1, nie byłoby przeszkód w usunięciu 3,5. Trzeba jednak sprawdzić, czy zamiana zbioru $\{1,2\}$ w $\{2\}$ nie narusza warunku b). Z tablicy trójkątnej wynika, że nie, gdyż 1,2 nie występuje w żadnej kratce, a więc nie warunkuje żadnej zgodności. Klasę Φ można zatem przedstawić w postaci

$$\Phi' = \{4,5,8; 1,4,7; 1,3,6; 2\}$$

Całego zbioru usunąć stąd już nie można, ale stany 1 i 4 występują dwukrotnie i można spróbować je wyrugować. Stanu 4 z $\{4,5,8\}$ wykreślić nie można, gdyż zgodność 4,5 jest potrzebna do zgodności 3,6, a 3 lub 6 nie może już być usunięte. Można natomiast wykreślić 4 z $\{1,4,7\}$, gdyż ani 1,4, ani 4,7 w tablicy trójkątnej nie występują. Stan 1 może być usunięty z dowolnego zbioru, więc otrzymuje się dwie postacie minimalne:

$$\Phi_{min 1} = \{4,5,8; 7; 1,3,6; 2\}$$

$$\Phi_{min 2} = \{4,5,8; 1,7; 3,6; 2\}$$

Nowa, minimalna tablica dla pierwszej postaci jest przedstawiona na rys. 4-17c. Tablica dla drugiej postaci jest identyczna.

Przedstawiona tu procedura redukcji Φ jest nieco uciążliwa, ale na szczęście rzadko musi być stosowana, gdyż przy tworzeniu tablicy przejść układu synchronicznego wg zasad podanych wyżej, otrzymuje się układ minimalny lub zbliżony do minimalnego i Φ ma postać jednoznaczna. Łatwo można to sprawdzić na przykładach z rys. 4-6, 4-7, 4-8, w których minimalizacja nie jest potrzebna.

4.2.3. ZMIANA RODZAJU UKŁADU

Każdy układ sekwencyjny może być zrealizowany w dwóch postaciach: jako układ Mealy'ego lub jako układ Moore'a. Zazwyczaj pierwsza postać ma mniej stanów wewnętrznych, a więc i mniej elementów pamięciowych, ale za to realizacja funkcji wyjściowych Y jest trudniejsza. Nie ma metod, które pozwoliłyby przewidzieć, która realizacja układu będzie prostsza i dlatego często trzeba zbadać stopień złożoności obydwu rozwiązań, do czego potrzebne są przynajmniej tablice przejść i wyjść. Tablice te można uzyskać z opisu słownego metodami opisanymi wyżej, ale najprościej jest utworzyć w ten sposób jedną tablicę (dla której jest to łatwiejsze), a drugą odtworzyć na podstawie pierwszej. Można to zrobić również po zminimalizowaniu pierwotnie otrzymanej tablicy.

a) Tablica Mealy'ego:

A	ab			
	00	01	11	10
0	0,0	0,1	1,0	0,1
1	0,1	1,0	1,1	1,0

b) Tablica z numerami N :

A	ab			
	00	01	11	10
0	00/0	01/1	10/2	01/1
1	01/1	10/2	11/3	10/2

c) Nowa tablica — Moore'a:

A	ab				y
	00	01	11	10	
0	0	1	2	1	0
1	0	1	2	1	1
2	1	2	3	2	0
3	1	2	3	2	1

Rys. 4-18. Zmiana układu Mealy'ego na układ Moore'a dla sumatora szeregowego: a) tablica Mealy'ego; b) tablica z numerami N ; c) nowa tablica — Moore'a

Zasady zamiany tablicy Mealy'ego na tablicę Moore'a są następujące:

1) każdej parze A' , Y wewnątrz tablicy trzeba przyporządkować kolejny numer N , przy czym jednakowym parom odpowiadać powinny jednakowe numery.

2) numery N uważa się za stany wewnętrzne A nowej tablicy układu Moore'a, przypisując każdemu N takie stany następne, jakie miał odpowiadający mu stan A' i taki stan wyjść, jaki był odpowiadający mu stan Y .

Rys. 4-18 wyjaśnia te czynności na przykładzie sumatora szeregowego (z rys. 4-5).

Przy zamianie układu Moore'a na układ Mealy'ego postępowanie jest następujące:

1) symbol stanu wyjść Y odpowiadający stanowi A wpisuje się wewnątrz tablicy przejść, obok odpowiednich symboli A ;

2) uzyskaną w ten sposób tablicę Mealy'ego minimalizuje się łącząc stany zgodne.

Przykład przekształcenia uzyskanej na rys. 4-18c tablicy Moore'a (sumatora szeregowego) w tablicę Mealy'ego jest przedstawiony na rys. 4-19.

a)	$\begin{array}{c cccc c} & \begin{array}{cccc} ab & 00 & 01 & 11 & 10 \end{array} & y \\ \hline \begin{array}{c} A \\ 0 \\ 1 \\ 2 \\ 3 \end{array} & \begin{array}{cccc} 0 & 1 & 2 & 1 \end{array} & \begin{array}{c} 0 \\ 1 \\ 2 \\ 1 \end{array} \end{array}$	b)	$\begin{array}{c cccc c} & \begin{array}{cccc} ab & 00 & 01 & 11 & 10 \end{array} & \\ \hline \begin{array}{c} A \\ 0 \\ 1 \\ 2 \\ 3 \end{array} & \begin{array}{cccc} 0,0 & 1,1 & 2,0 & 1,1 \\ 0,0 & 1,1 & 2,0 & 1,1 \\ 1,1 & 2,0 & 3,1 & 2,0 \\ 1,1 & 2,0 & 3,1 & 2,0 \end{array} & \begin{array}{c} \\ \\ \\ \end{array} \end{array}$	c)	$\begin{array}{c cccc c} & \begin{array}{cccc} ab & 00 & 01 & 11 & 10 \end{array} & \\ \hline \begin{array}{c} A \\ (0,1)0 \\ (2,3)1 \end{array} & \begin{array}{cccc} 0,0 & 0,1 & 1,0 & 0,1 \\ 0,1 & 1,0 & 1,1 & 1,0 \end{array} & \begin{array}{c} \\ \\ \end{array} \end{array}$
----	---	----	--	----	---

Rys. 4-19. Zmiana układu Moore'a na układ Mealy'ego dla sumatora szeregowego: a) tablica Moore'a; b) przeniesienie symboli wyjść; c) zminimalizowana tablica Mealy'ego

4.2.4. KODOWANIE STANÓW WEWNĘTRZNYCH

Po uzyskaniu minimalnej tablicy przejść Moore'a lub Mealy'ego należy tę tablicę zakodować, tzn. opisać za pomocą sygnałów binarnych x , y i Q . Zazwyczaj stany X i Y od samego początku syntezy występują w postaci ciągów sygnałów (x_1, x_2, \dots, x_n) i (y_1, y_2, \dots, y_m) , gdyż wstępne założenia obejmują właśnie relacje między konkretnymi sygnałami x i y . Natomiast stany A , wprowadzone w procesie syntezy, mają dowolną postać umowną (zwykle kolejnych liczb naturalnych) i dlatego przed dalszymi etapami projektowania układu należy je zapisać w standardowej postaci binarnej, czyli każdemu A_i przypisać konkretny ciąg (Q_1, Q_2, \dots, Q_k) .

Pierwszym problemem, który powstaje przy kodowaniu stanów wewnętrznych, jest określenie liczby k sygnałów Q potrzebnych dla opisu K stanów wewnętrznych. Ponieważ można przypuszczać, że im więcej przerzutników (elementów pamięci) z wyjściami Q zawiera układ, tym jest droższy, bardziej skomplikowany i zawodny, należy więc dążyć do możliwie najmniejszej liczby k dla danego K . Najmniejsza możliwa liczba k wynika z zależności

$$2^{k-1} < K \leq 2^k$$