

przerzutników typu T — należy przy wyborze podziałów kierować się przede wszystkim kryterium minimalnej liczby zmian.

Z tablicy na rys. 4-29c można otrzymać:

$$\begin{aligned} D_1 &= Q_2 & D_2 &= \bar{Q}_1 & D_3 &= Q_2 Q_3 + Q_2 x \\ T_1 &= Q_1 \bar{Q}_2 + \bar{Q}_1 Q_2 & T_2 &= Q_1 Q_2 + \bar{Q}_1 \bar{Q}_2 = \bar{T}_1 & T_3 &= Q_1 \bar{Q}_2 Q_3 + Q_2 \bar{Q}_3 x \\ S_1 &= J_1 = Q_2 & S_2 &= J_2 = \bar{Q}_1 & S_3 &= J_3 = Q_2 x \\ R_1 &= K_1 = \bar{Q}_2 & R_2 &= K_2 = Q_1 & R_3 &= K_3 = \bar{Q}_2 \end{aligned}$$

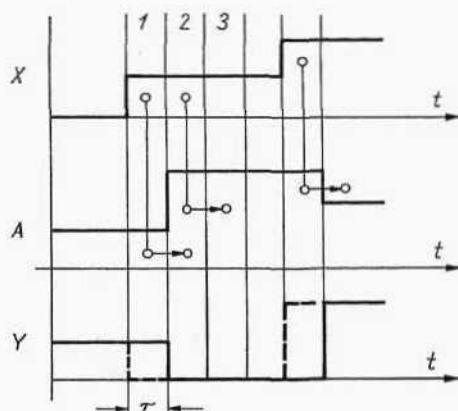
Schemat jednej z możliwych wersji układu przedstawiono na rys. 4-30. Sygnał  $Z$  sprowadza układ do stanu początkowego 0 (000). Funkcja  $y$  wynika z rys. 4-23a,d:

$$y = Q_1 \bar{Q}_2 Q_3 x$$

### 4.3. UKŁADY ASYNCHRONICZNE STATYCZNE

#### 4.3.1. ZALEŻNOŚCI CZASOWE

W układach asynchronicznych statycznych wszelkie zmiany stanu wewnętrznego (stanu pamięci) i stanu wyjść są wywołane bezpośrednio zmianami stanu wejść, przy czym — zgodnie z zależnościami (4-7) — o tym, jaki będzie „nowy” stan  $A^{t+1}$ , decydują „stare” stany  $A^t$  i  $X^t$ . Na rys. 4-31 kropkami i linią łączącą pokazano sposób tworzenia nowego



Rys. 4-31. Zależności czasowe między sygnałami w układzie asynchronicznym

stanu  $A$ . Czas  $\tau$  trwania mikrotaktu uwzględnia opóźnienie między zmianą  $X$  a wywołaną tym zmianą stanu pamięci. Mikrotakt 1 oznacza stan przejściowy, charakteryzujący się tym, że  $A^{t+1} \neq A^t$ . Działające w tym czasie sygnały wzbudzeń powodują zmianę stanu  $A$ ; w następnych mikrotaktach  $A^{t+1} = A^t$ , a więc układ osiągnął stan stabilny. Trwa on aż do następnej, odpowiedniej zmiany  $X$ . Występowanie stanów stabilnych jest charakterystyczną cechą układów asynchronicznych i stanowi zabezpieczenie przed lawinową zmianą stanów, będącą istotnym problemem w układach synchronicznych (rys. 4-13a).

W układach Moore'a zmiany stanów wyjść  $Y$  następują wraz ze zmianami  $A$  i odpowiednie momenty są jednoznacznie określone (linia ciągła na wykresie  $Y$  — rys. 4-31). W układach Mealy'ego wartości sygnałów wyjściowych są dokładnie określone w stanach stabilnych, natomiast w przejściowych stanach niestabilnych mogą być różne. Prawidłowa praca dalszych układów, korzystających z sygnałów  $Y$ , wymaga zazwyczaj, aby sygnały wyjściowe zachowały ciągłość w stanach niestabilnych, tzn. aby miały wówczas wartość taką jak w poprzednim albo w następnym stanie stabilnym. Konkretnie ustalenie tych wartości może być dokonane w ostatnich etapach syntezy, tak by odpowiednie funkcje były możliwie najprostsze.

Na wykresach czasowych, stanowiących najczęściej stosowany sposób wstępnego opisywania układów asynchronicznych, nanoszone są tylko stany stabilne układu. Na ich podstawie tworzy się pierwotną tablicę przejść i wyjść układu, w której każdy stan stabilny znajduje się w oddzielnym wierszu. Stany niestabilne można nanieść w tablicy analizując kolejność występowania stanów stabilnych.

#### 4.3.2. MINIMALIZACJA TABLIC

Tablica przejść utworzona na podstawie wykresu czasowego ma prawie zawsze liczbę stanów większą od minimalnej i dlatego minimalizacja tablicy jest niezbędnym etapem procesu syntezy. Podobnie jak w przypadku układów synchronicznych, etap ten realizuje się przez poszukiwanie stanów zgodnych. Definicje stanów niesprzecznych i zgodnych były podane wyżej, ale dla układów asynchronicznych obecność stanów stabilnych zmienia nieco rolę stanów  $Y$  i dlatego sposób określania zgodności jest trochę inny.

Stany  $A$  i  $A'$  można, na podstawie tablicy przejść, uznać za zgodne, gdy:

- 1) pod wpływem każdego  $X$  przechodzą w stany niesprzeczne lub zgodne i
- 2) jeśli są stabilne przy tym samym  $X$ , to odpowiadające im stany  $Y$  są niesprzeczne.

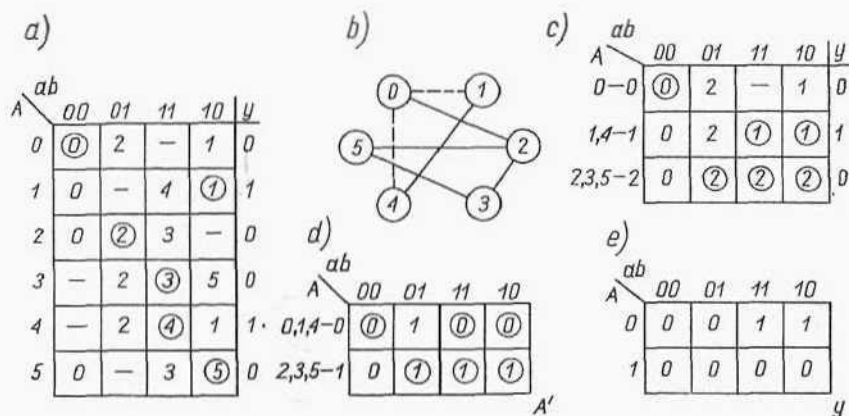
Przy poszukiwaniu maksymalnych zbiorów stanów zgodnych można posługiwać się opisaną wyżej metodą z tablicą trójkątną, ale łatwiej uzyskuje się wyniki korzystając z pewnych specjalnych cech tablic układów asynchronicznych. Cechy te polegają na wyraźnym uporządkowaniu stanów wewnątrz tablicy; jeśli w tablicy pierwotnej w kolumnie  $X_j$  występuje stan  $A_i$ , to wiadomo, że stany  $A_i$  występują tylko w kolumnie  $X_j$ , a jeden ze stanów  $A_i$  jest stabilny (w wierszu  $A_i$ ).

W takiej sytuacji główną trudność tworzenia zbiorów stanów zgodnych — zgodność warunkową (gdy zgodność  $a, b$  zależy od zgodności  $c, d$  itd.) — można przenieść na same tylko stany stabilne, których jest mniej niż wszystkich stanów w tablicy. Ponieważ najprostszym kryterium niezgodności stanów jest sprzeczność wyjść  $Y$  dla stanów stabilnych przy tym samym  $X$ , więc od takich właśnie stanów trzeba rozpocząć analizę.

Stany  $A$  i  $A'$ , zgodne i stabilne przy tym samym  $X$ , są nazywane *stanami pseudorównoważnymi*. Wyszukuje się je względnie łatwo, analizując kolejno kolumny tablicy przejść. Wszystkie zbiory stanów pseudorównoważnych tworzą klasę  $\Psi$ . Znajomość klasy  $\Psi$  bardzo ułatwia badanie zgodności wszystkich innych stanów, gdyż jeśli stany  $A_1$  i  $A_2$  przechodzą, przy jakimś  $X$ , do stanów  $A_3$  i  $A_4$ , a para  $A_3, A_4$  nie należy do  $\Psi$ , to  $A_1$  i  $A_2$  z pewnością nie są zgodne. Jeśli para  $A_3, A_4$  należy do  $\Psi$ , a inne warunki są spełnione, to  $A_1$  i  $A_2$  są zgodne. Jeśli założyć się, że każdy zbiór z klasy  $\Psi$  zostanie zastąpiony jednym stanem, to znika problem zgodności warunkowej i ocenić zgodność innych stanów można bardzo łatwo. Dodatkową pomocą w tym może być *wykres skracania* (graf), którego wierzchołkami są zbiory stanów pseudorównoważnych i poszczególne inne stany, a gałęzie oznaczają zgodność odpowiednich stanów. Maksymalny zbiór stanów zgodnych będzie wskazany przez największy zbiór nawzajem połączonych wierzchołków grafu. Wybierając najmniejszą liczbę takich zbiorów, obejmującą wszystkie stany bez

ich powtarzania, uzyskuje się bezpośrednio minimalną klasę stanów zgodnych  $\Phi_{min}$ . Każdy zbiór tej klasy można zastąpić jednym stanem, uzyskując w ten sposób tablicę minimalną.

Tablica pierwotna jest zawsze tablicą Moore'a, co umożliwia prowadzenie minimalizacji dwiema drogami: dla uzyskania minimalnej tablicy Moore'a lub minimalnej tablicy Mealy'ego. Ponieważ nie można przewidzieć, która droga będzie lepsza, należy rozważyć obydwie. W tym celu na wykresie skracania stany (zbiory) zgodne o wyjściach niesprzecznych łączy się linią ciągłą, a stany zgodne o wyjściach sprzecznych — linią przerywaną. Dalsze wykorzystywanie figur o liniach ciągłych prowadzi do układu Moore'a, natomiast wykorzystywanie wszystkich linii prowadzi do układu Mealy'ego.



Rys. 4-32. Minimalizacja tablicy układu bramkowania generatora: a) tablica pierwotna, b) wykres skracania; c) minimalna tablica Moore'a; d) minimalna tablica przejść Mealy'ego; e) tablica wyjść układu Mealy'ego

Kolejność postępowania przy minimalizacji można przedstawić następująco:

- 1) wyznaczyć stany pseudorównoważne i klasę  $\Psi$ ,
- 2) narysować wykres skracania,
- 3) wyznaczyć  $\Phi_{min}$  dla układów Moore'a i Mealy'ego,
- 4) na podstawie  $\Phi_{min}$  narysować minimalne tablice przejść i wyjść układów Moore'a i Mealy'ego.

Pierwszym przykładem postępowania niech będzie utworzona w p. 4.1.3 (rys. 4-9b) tablica pierwotna układu bramkowania generatora (rys. 4-32a). Parami pseudorównoważnymi mogłyby być (3,4) i (1,5), ale mają sprzeczne wyjścia, więc zbiór  $\mathcal{P}$  jest pusty. Wykres skracania (rys. 4-32b) ma w tej sytuacji tyle wierzchołków, ile jest stanów. Z połączeń wyraźnie wynika, że jednym ze zbiorów klasy  $\Phi_{min}$  będzie {2,3,5}. W przypadku układu Moore'a pozostałymi zbiorami będą {0} i {1,4} więc

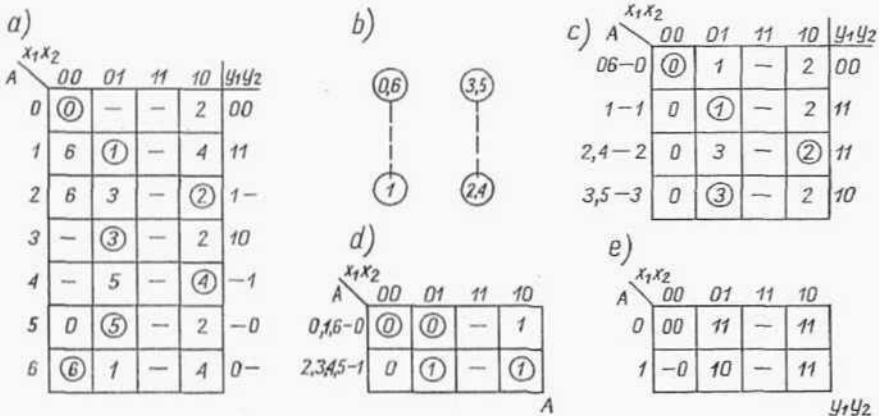
$$\Phi_{min} = \{0; 1,4; 2,3,5\}$$

Z połączenia 0-2 zrezygnowano, redukując w ten sposób liczbę stanów w  $\Phi_{min}$  do minimum.

W przypadku układu Mealy'ego otrzymuje się

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

Tablicę wyjść układu Mealy'ego buduje się na podstawie tablicy pierwotnej przypisując odpowiednie sygnały wyjściowe stanom stabilnym tablicy przejść. Wartości  $Y$  dla stanów niestabilnych określa się,



Rys. 4-33. Przykład minimalizacji tablicy

analizując istniejące przejścia, np. w pierwszym wierszu niestabilnemu stanowi wewnętrznemu 1 musi odpowiadać  $y = 0$ , gdyż za pośrednictwem tego stanu układ przechodzi ze stanu 0 (tablica pierwotna  $y = 0$ ) do stanu 2 ( $y = 0$ ), a więc musi być zachowana trwała wartość  $y = 0$ .

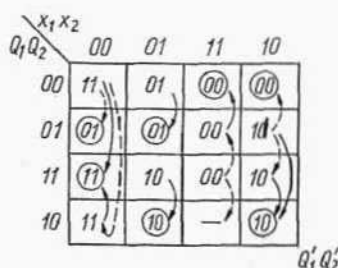
Przykładem nieco bardziej złożonym może być tablica pierwotna z rys. 4-33a. Analiza kolumn wykazuje warunkową pseudorównoważność stanów: (0,6), jeśli (2,4), a (2,4), jeśli (3,5). Ponieważ stany 3 i 5 są pseudorównoważne bez dodatkowych warunków, więc

$$\psi' = \{0,6; 2,4; 3,5\}$$

Uważając każdą z tych par za jeden stan można narysować wykres skracania (rys. 4-33b), a na jego podstawie tablice układów Moore'a (c) i Mealy'ego (d, e). Stanowi niestabilnemu 1 (rys. 4-33d) muszą odpowiadać sygnały wyjściowe 11, gdyż pośredniczy on w przejściu ze stanu o wyjściach 11 do stanu o wyjściach 11. Stan niestabilny 0 pośredniczy przy zmianach wyjść z 10 na 00 i z 11 na 00, a więc mogą mu odpowiadać sygnały 10 lub 00, czyli (-0).

### 4.3.3. KODOWANIE STANÓW WEWNĘTRZNYCH

Metody kodowania stosowane przy syntezie układów synchronicznych nadają się również do kodowania układów asynchronicznych, ale dochodzą tu pewne dodatkowe wymagania, zmieniające sposób postępowania. Wyjaśnia to rys. 4-34, na którym przedstawiono zakodowaną



Rys. 4-34. Tablica przejść z różnymi postaciami wyjść

tablicę przejść. Jeśli opisany tablicą układ jest w stanie  $A = (00)$  i działa  $X = (00)$ , to powinien on przejść do  $A = (11)$ . Przy takim przejściu zmieniają swój stan obydwa elementy pamięciowe ( $Q_1$  i  $Q_2$ ). Ponieważ nie istnieją dwa elementy fizyczne o identycznych właściwościach, więc zawsze jeden z elementów pamięciowych zareaguje wcześniej niż drugi i układ znajdzie się w stanie pośrednim  $A = (10)$  lub  $A = (01)$  (pokazano to liniami przerywanymi na rys. 4-34). Ze stanu  $A = (10)$  naturalne

wymuszenia kierują układ do  $A' = (11)$ , więc ta droga zakończy się właściwie, ale stan  $A = (01)$ , przy  $X = (00)$ , jest stabilny i układ przy opóźnionej reakcji  $Q_1$  zostanie w stanie  $A = (01)$ . Opisane zjawisko, polegające na uzyskiwaniu błędnego stanu stabilnego na skutek różnic w czasach działania elementów (lub innych opóźnień na drodze sygnału) nazywa się *wyścigiem krytycznym*.

W rozpatrywanej tablicy przy  $X = (01)$  zmienia wartość najwyżej jeden sygnał  $Q$ , więc wyścigów nie ma. Przy  $X = (11)$  i  $A = (11)$  występuje wyścig elementów pamięci, ale wystarczy tu w miejsce stanu nieokreślonego (kreski) wpisać (00), by wszystkie możliwe przejścia prowadziły do właściwego celu. Wyścig taki nosi nazwę *wyścigu niekrytycznego*. W ostatniej kolumnie tablicy, przy  $A = (01)$ , również występuje wyścig krytyczny, prowadzący przy opóźnionej reakcji  $Q_1$  do stanu  $A = (00)$ , zamiast do  $A = (10)$ . W tym przypadku niewłaściwe przejście można względnie łatwo zlikwidować wykorzystując fakt, że nie tylko ze stanu  $A = (01)$ , lecz także z  $A = (11)$  układ przechodzi do  $A' = (10)$ . Można zatem w kratce o współrzędnych  $A = (01)$ ,  $X = (10)$  zamiast  $A' = (10)$  napisać  $A' = (11)$ , co sprawi, że ze stanu  $A = (01)$  pod wpływem  $X = (10)$  układ przejdzie kolejno drogę:  $(01) \rightarrow (11) \rightarrow (10)$ , bez wyścigu dochodząc do właściwego stanu. Tego typu zabieg nosi nazwę *przejścia cyklicznego* i związany jest z pewnym wydłużeniem czasu reakcji układu.

Jeśli wprowadzany przejściem cyklicznym stan pośredni powoduje impulsowe, krótkotrwałe zmiany sygnałów wyjściowych, to cykl taki może być niedopuszczalny, ze względu na układy odbierające sygnały wyjściowe.

Rozważania powyższe umożliwiają sformułowanie dwóch bardziej ogólnych wniosków:

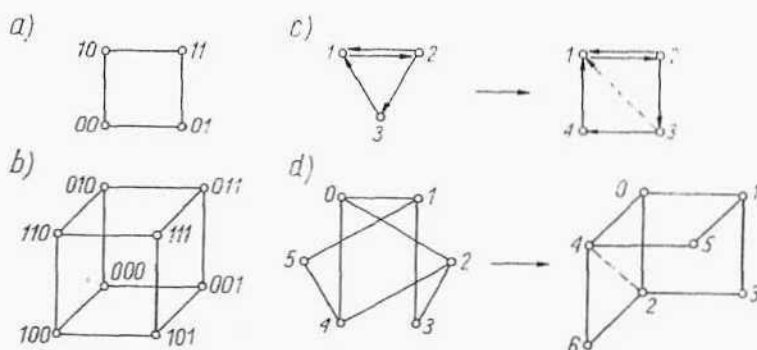
1) gdy w kolumnie tablicy występuje tylko jeden stan stabilny, to ta kolumna nie może być miejscem wyścigu krytycznego, należy tylko usunąć ewentualne stany nieokreślone;

2) wyścig krytyczny w kolumnie  $X$  można zamienić na niekrytyczny, jeśli istnieją warunki wprowadzenia przejścia cyklicznego, tzn. gdy w tej kolumnie  $X$  jest jeszcze inny stan niestabilny, o takich samych sygnałach  $Q$  jak stan krytyczny, a sygnały wyjściowe nie wykazują krótkotrwałych zmian;

3) jeśli przejścia cyklicznego zastosować nie można, należy zmienić kod stanów wewnętrznych.

Zmiana kodu, a właściwie — wybór kodu stanów wewnętrznych układu asynchronicznego powinien więc być dokonany tak, aby nie mogły powstać wyścigi krytyczne. Ten warunek jest ważniejszy niż kryteria złożoności, gdyż wyścig może uniemożliwić poprawną pracę urządzenia.

Najprostszą metodą uniknięcia wyścigów krytycznych jest takie kodowanie, aby każde dwa stany  $A, A'$  takie że  $A$  przechodzi w  $A'$ , miały sąsiednie wyrażenia kodowe, tzn. aby odpowiadające im sygnały  $Q$  różni-



Rys. 4-35. Przykłady typowych kostek dwu- i trzypięciowymiarowych (a,b) i przekształcania wykresu przejść do postaci kostki (c,d)

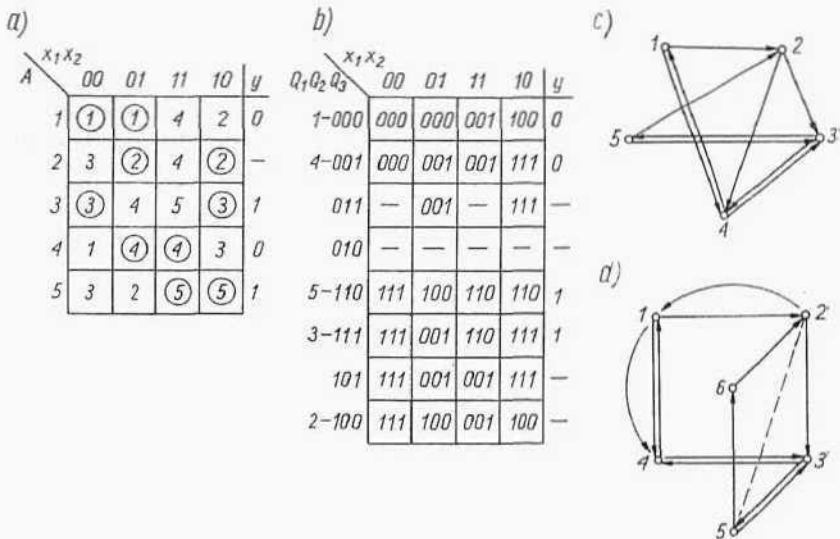
ły się w jednym tylko miejscu. Pomocą w takim doborze kodu mogą być uproszczone wykresy przejść. Na rys. 4-35ab przedstawiono figury (kostki) o czterech i ośmiu stanach, zakodowane tak, że każde dwa połączone ze sobą stany mają sąsiednie wartości kodu. Jeśli wykres przejść projektowanego układu da się przedstawić w postaci takiej kostki lub jej części, to — wybierając kod np. tak jak na kostkach wzorcowych — można całkowicie uniknąć wyścigów krytycznych.

Jeśli wykres przejść ma trzy wzajemnie połączone stany (rys. 4-35c), sytuację bez wyścigu można uzyskać, wprowadzając pomocniczy stan czwarty, a zamiast przejścia 1-3 wymuszając przejście 1-4-3. Dodatkowy stan 4 należy wprowadzać między tymi stanami, które są powiązane najmniejszą liczbą przejść.



Inny przykład sprowadzania wykresu przejść do postaci kostki jest pokazany na rys. 4-35d. Pomocniczy stan 6 umożliwia wyrugowanie przekątnej na jednej ze ścian kostki. Stany pomocnicze wprowadza się we wszystkich przypadkach, gdy wykresu przejść nie można bezpośrednio przedstawić w postaci kostki.

Opisana metoda nie zawsze daje dobre wyniki, jej stosowanie przy liczbie stanów większej od 8 jest trudne, a ponadto nie uwzględnia możliwości uproszczenia układu. Z tych powodów bardziej godna polecenia jest metoda oparta na rachunku podziałów, jak w przypadku układów



Rys. 4-36. Przykład kodowania: a) minimalna tablica przejść i wyjść; b) tablica zakodowana; c) uproszczony wykres przejść; d) wykres w postaci kostki

synchronicznych. Ponieważ stosowana tam procedura wyboru kodu prowadziła do minimalizacji układu, ale nie uwzględniała problemu wyścigów — należy ją uzupełnić dodatkowymi kryteriami.

Można zauważyć, że jeśli każde dwa stany  $A_1$  i  $A_2$ , takie że  $A_1$  przechodzi w  $A_2$ , mają kody o części wspólnej (tzn. o identycznych wartościach jednego lub więcej sygnału  $Q$ ), która nie występuje w kodach innych stanów nie przechodzących w  $A_2$ , to wyścigi krytyczne nie

powstają. Wynika to z faktu, że skoro istnieje niezmienna część wyrażenia kodowego przy zmianie  $A_1$  w  $A_2$ , to zamiast  $A_2$  (na skutek opóźnień w reakcji) może wystąpić tylko taki stan  $A_3$ , który ma identyczną z  $A_1$  i  $A_2$  stałą część kodu. Takie stany  $A_3$  jednak (jeśli występują), przechodzą zgodnie z założeniem w  $A_2$ , a więc nie powodują wyścigów krytycznych.

Wybór kodu spełniającego opisane wymagania jest trudny, gdyż sprowadza się do analizy wszystkich par stanów  $(A, A')$ , takich, że  $A$  przechodzi w  $A'$ . Problem można jednak znacznie uprościć, wykorzystując podziały wewnętrzne, gdyż zgodnie z ich definicją, każdy blok z  $\pi(X_i)$  obejmuje stany przechodzące do jakiegoś jednego stanu  $A_j$ . Jeśli przyjęte do kodowania podziały  $\tau$  separują od siebie bloki podziałów wewnętrznych, to kody stanów przechodzących w  $A_j$  będą miały część wspólną (bo stany te występują w jednym bloku jakiegoś  $\tau$ ), inną niż stany pozostałe (występujące w innym bloku  $\tau$ ). Spełnione są więc opisane wyżej wymagania i kod nie powoduje wyścigów krytycznych. Na przykład z tablicy na rys. 4-36a otrzymuje się:

$$\pi(X_0) = \{\overline{1,4}; \overline{2,3,5}\} = \tau_1$$

$$\pi(X_1) = \{\overline{1}; \overline{2,5}; \overline{3,4}\}$$

$$\pi(X_2) = \{\overline{1,2}; \overline{3,4}; \overline{5}\} \quad \pi(X_1) + \pi(X_2) = \{\overline{1,25}; \overline{3,4}\} = \tau_3$$

$$\pi(X_3) = \{\overline{1,2,4}; \overline{3,5}\} = \tau_2$$

$$\tau_1 \tau_2 \tau_3 = 0$$

Łatwo można sprawdzić, że wszystkie bloki podziałów wewnętrznych są w  $\tau$  wzajemnie odseparowane, np.  $\overline{1}$  od  $\overline{2,5}$  — w  $\tau_1$ ,  $\overline{1,2-5}$  — w  $\tau_2$  itd. Zastosowanie kodu

	$\tau_1$	$\tau_2$	$\tau_3$
1	0	0	0
2	1	0	0
3	1	1	1
4	0	0	1
5	1	1	0

nie tylko zabezpiecza przed wyścigiem krytycznym, lecz także upraszcza w istotny sposób realizację, co wynika z rozważań z p. 4.2.4. Zakodowaną tablicę przedstawiono na rys. 4-36b; wiele jej stanów nieokreślonych nale-

zało dodatkowo określić na podstawie analizy przejść, np. przejście z (100) w (111) sprawia, że i (101) musi przejść w (111). Czynności te są potrzebne tylko wtedy, gdy w tablicy są stany nieokreślone, przy czym rugowanie kresek zazwyczaj nie wiąże się z komplikacją układu. Na rys. 4-36c, d pokazano odpowiedni wykres przejść i jego zmodyfikowana postać. Modyfikacja ta wymaga wprowadzenia przejścia cyklicznego (z 2 do 4 przez 3) i dodatkowego stanu 6 (z 5 do 2 przez 6). Zabiegi te nie były potrzebne przy zastosowaniu metody podziałów.

a)					
$x_1 x_2$					
A	00	01	11	10	y
1	(1)	2	(1)	(1)	0
2	—	(2)	3	1	0
3	4	(3)	(3)	(3)	1
4	(4)	—	—	1	1
A'					

b)					
$x_1 x_2$					
$q_1 q_2$	00	01	11	10	y
1-00	00	01	00	00	0
2-01	—	01	11	00	0
3-11	10	11	11	11	1
4-10	10	—	—	00	1
$q_1' q_2'$					

Rys. 4-37. Przykład kodowania

W opisanym przykładzie wybrane do kodowania podziały  $\tau$  separowały wszystkie bloki podziałów wewnętrznych, więc nie było potrzeby wyszukiwania par  $(A, A')$ . Niekiedy jednak istnieje konieczność dodatkowego badania par. Na przykład z tablicy na rys. 4-37a można wypisać następujące podziały wewnętrzne:

$$\pi(X_0) = \{\overline{1}; \overline{3,4}; (2)\} \leq \{\overline{1,2}; \overline{3,4}\} = \tau_1$$

$$\pi(X_1) = \{\overline{1,2}; \overline{3}; (4)\} \leq \tau_1$$

$$\pi(X_2) = \{\overline{1,2,4}; \overline{3}\}$$

$$\pi(X_3) = \{\overline{1}; \overline{2,3}; (4)\} \leq \{\overline{1,4}; \overline{2,3}\} = \tau_2$$

$$\tau_1 \tau_2 = 0$$

Stany nieokreślone zostały wzięte w nawias, a dla przekształcania podziałów nawias oznacza, że zawarte w nim elementy mogą być dołączone do dowolnego bloku, albo tworzyć odrębny blok.

Podziały  $\tau_1$  i  $\tau_2$  spełniają wszystkie warunki, z wyjątkiem jednego — nie separują bloków podziału  $\pi(X_2)$ . W takim przypadku należy zrezygno-

wać z separowania całych bloków i przeanalizować pary. Zamiast dodatkowego warunku 1,2,4-3, powstają wówczas dwa warunki 1,2-3 i 1,4-3, gdyż 2 i 4 przechodzą w 1. Łatwo można sprawdzić, że warunek oddzielenia 1,2-3 jest spełniony przez  $\tau_1$ , a 1,4-3 — przez  $\tau_2$ , a zatem kodowanie zgodnie z tymi podziałami nie powoduje wyścigów krytycznych i prowadzi do najprostszej realizacji.

Na podstawie powyższych rozważań i przykładów można zestawić następującą kolejność czynności przy kodowaniu:

- 1) wypisać podziały wewnętrzne;
- 2) na ich podstawie utworzyć podziały prawidłowe;
- 3) wybrać takie podziały prawidłowe, aby ich iloczynem był podział zerowy;
- 4) sprawdzić czy wybrane podziały separują wszystkie bloki podziałów wewnętrznych; jeśli tak — wykonać p. 9, jeśli nie — wykonać p. 5;
- 5) wypisać nie spełnione warunki separacji bloków i zamienić je na warunki separacji par;
- 6) sprawdzić, czy wybrane podziały prawidłowe separują te pary; jeśli tak — wykonać p. 9, jeśli nie — wykonać p. 7;
- 7) sprawdzić, czy wprowadzenie przejść cyklicznych likwiduje warunki separacji; jeśli tak — wprowadzić je i wykonać p. 9, jeśli nie — wykonać p. 8;
- 8) wprowadzić dodatkowy podział prawidłowy, realizujący warunki separacji i wykonać p. 9;
- 9) na podstawie wybranych podziałów prawidłowych utworzyć kod i zakodować tablice przejść i wyjść.

Piąty punkt tego algorytmu wiąże się ze zwiększeniem liczby elementów pamięciowych i jest stosowany tylko w ostateczności, gdy zawiodą inne sposoby uniknięcia wyścigu krytycznego.

Jeśli istnieje kilka wariantów rozwiązania, o wyborze mogą decydować pary podziałów lub podziały zewnętrzne, podobnie jak w układach synchronicznych.

#### 4.3.4. REALIZACJA PAMIĘCI I FUNKCJE WZBUDZEŃ

W układach asynchronicznych statycznych są stosowane dwie metody realizacji pamięci.