

i oznacza po prostu minimalną liczbę bitów, potrzebnych do przedstawienia w postaci dwójkowej liczby K . Niekiedy minimalizacja liczby elementów pamięci nie prowadzi do najprostszego układu, ale jest to problem dalszy; na początku należy próbować zastosować minimalną wartość k .

Gdy k jest już określone, pozostaje ustalenie relacji między konkretnymi stanami A i ciągami wartości (Q_1, Q_2, \dots, Q_k) . Na poprawne działanie układu synchronicznego nie ma to wprawdzie żadnego wpływu, ale — jak się okazuje — ma bardzo duży wpływ na stopień złożoności tego układu. Wynika to z faktu, że tablica przejść i wyjść jest po zakodowaniu doprowadzana do postaci tablicy Karnaugh'a, z której znanymi metodami są otrzymywane pewne funkcje kombinacyjne. Na postać tych funkcji istotny wpływ ma to, czy sąsiadujące ze sobą w tablicy stany A_i i A_j są zapisane jako np. 011 i 111, czy jako 000 i 111, gdyż w pierwszym przypadku istnieją większe możliwości sklepień, a więc uproszczenia funkcji i układu. Podobnie nie jest obojętne czy stan 000 przechodzi pod wpływem jakiegoś X w stan 001 czy w 111, gdyż w pierwszym przypadku zmiana angażuje mniej sprzętu i można spodziewać się, że układ będzie prostszy. Te i inne podobne przyczyny sprawiają, że kodowanie stanów wewnętrznych powinno być przeprowadzane tak, by powstały w dalszych etapach układ był możliwie najprostszy. Jest to jednak problem bardzo trudny, gdyż już przy $K = 5$, możliwych, odmiennych sposobów kodowania jest 140 i liczba ta szybko rośnie ze wzrostem K , a więc porównanie wszystkich wersji jest praktycznie nierealizowalne. Nawet za pomocą maszyny cyfrowej trudno jest przekroczyć barierę $K = 9$ (10 milionów wariantów kodowania), pozostaje więc poszukiwanie metody umożliwiającej ocenę złożoności układu bez potrzeby jego syntezy już na podstawie tablic przejść i wyjść (przed ich zakodowaniem). Niestety, metody umożliwiającej odszukanie najlepszego wariantu kodowania dowolnego układu dotychczas nie ma, a nawet, gdy powstanie, będzie zbyt złożona dla syntezy niemaszynowej, gdyż musi uwzględnić wiele różnych czynników mających wpływ na złożoność układu. Również najogólniejsza z istniejących metod (Hartmanisa i Stearnsa), chociaż nie obejmuje wszystkich układów i czynników, w swej klasycznej postaci jest mało użyteczna, gdyż wymaga licznych, pracochłonnych czynności, nawet w przypadku prostych tablic o $K = 5$. W zastosowaniach praktycznych szybciej uzyskuje się dobry wynik badając tablice pod kątem wypełnienia kilku różnych wa-

runków mających wpływ na złożoność układu. Postępowanie takie nie gwarantuje wprowadzić, że rezultat będzie najlepszy, ale uwzględnia wiele czynników, daje pewną swobodę wyboru rozwiązania i zwykle prowadzi do rozwiązań najlepszych lub zbliżonych do najlepszych.

Gdy układ sekwencyjny ma dwa stany (np. 1,2), problem kodowania nie istnieje, gdyż są tylko dwie możliwości:

	A	B
1 —	0	1
2 —	1	0

Ponieważ wersja A różni się od B tylko negacją, zamiana Q na \bar{Q} i \bar{Q} na Q w A — daje B i odwrotnie. Przerzutniki mają zazwyczaj zarówno wyjście Q jak i \bar{Q} więc złożoność obydwu wersji jest identyczna, a nawet gdyby realizacja jednej wersji była mniej wygodna — przejście na drugą wersję jest proste. Wynika stąd wniosek, który będzie wykorzystywany dalej: kombinacje kodowe różniące się o negację są równoważne i wystarczy uwzględnić jedną z nich.

Tak więc — w przypadku dwóch stanów wewnętrznych istnieje w zasadzie tylko jeden sposób kodowania.

Gdy układ ma trzy stany (np. 1,2,3) — można wypisać wiele wariantów kodowania, np.

	A	B	C	D	E	F
1 —	00	00	00	11	00	01
2 —	01	01	11	10	11	11
3 —	11	10	10	00	01	00

Jednakże dokładniejsza analiza wykazuje, że wersja D powstała z A przez negowanie, E — to wersja C z miejscami zamienionymi, a F — to wersja B z miejscami zamienionymi i negacją drugiej wartości. Oczywiście, zamiana miejsc w kodzie nie może wpłynąć na złożoność układu, bo wszystkie funkcje logiczne są przemienne, a więc wersje z wartościami zamienionymi Q można uznać za równoważne i uwzględniać tylko jedną z nich. Wynika stąd, że ze wszystkich podanych wyżej możliwości kodowania trzech stanów, tylko wersje A , B i C różnią się od siebie w sposób istotny dla realizacji układu. Okazuje się, że również inne, dowolne wersje,

są równoważne jednej z wersji A , B , C . Można się o tym przekonać za pomocą wyprowadzonego przez Hartmanisa *rachunku podziałów*, bardzo wygodnego narzędzia przy poszukiwaniu najlepszych metod kodowania.

Podział jest to pełny (tzn. zawierający wszystkie elementy rozpatrywanego zbioru) zbiór podzbiorów rozłącznych. Podzbiory są nazywane *blokami* i wyodrębnione za pomocą kreski nad ich elementami, a podziały są oznaczane małymi literami greckimi.

Na przykład $\pi = \{\overline{1,2}; \overline{3,5}; \overline{4}\}$ jest trójblokowym podziałem zbioru $\{1,2,3,4,5\}$, natomiast $\{1; 3,5; 4\}$ i $\{1,2; 3,5; 2,4\}$ nie są podziałami tego zbioru.

Podział π_1 jest *nie większy* od określonego na tych samych elementach podziału π_2 ($\pi_1 \leq \pi_2$), jeśli każdy blok z π_1 jest zawarty w jakimś bloku z π_2 , np.

$$\{\overline{1,2}; \overline{3}; \overline{4}\} \leq \{\overline{1,2}; \overline{3,4}\}$$

gdyż $\{1,2\} \subseteq \{1,2\}$, $\{3\} \subseteq \{3,4\}$ i $\{4\} \subseteq \{3,4\}$.

Zgodnie z tą relacją, podziałem *najmniejszym* jest podział o blokach jednoelementowych, oznaczany przez 0. Na przykład:

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

Podziałem największym jest podział o jednym bloku, oznaczany przez 1, np.

$$\{\overline{1, 2, 3, 4, 5}\} = 1$$

Dla dowolnego π jest $\pi \geq 0$ i $\pi \leq 1$.

Iloczyn podziałów π_1 i π_2 (tego samego zbioru) to podział, którego blokami są przekroje bloków π_1 z blokami π_2 . Na przykład:

$$\{\overline{1,2,3}; \overline{4,5,6}\} \cdot \{\overline{1,2,6}; \overline{3,4,5}\} = \{\overline{1,2}; \overline{3}; \overline{6}; \overline{4,5}\},$$

gdyż $\{1,2,3\} \cap \{1,2,6\} = \{1,2\}$, $\{1,2,3\} \cap \{3,4,5\} = \{3\}$ itd.

Podobnie: $\{\overline{1,2}; \overline{3}; \overline{4,5}\} \cdot \{\overline{1,3,4}; \overline{2,5}\} = 0$.

Suma podziałów π_1 i π_2 (tego samego zbioru) to najmniejszy podział π taki, że jeśli a jest elementem jakiegoś bloku z π_1 lub π_2 , to cały ten blok jest zawarty w jednym bloku π . Na przykład:

$$\{\overline{1,2}; \overline{3,4}; \overline{5,6}\} + \{\overline{1,3}; \overline{2,4}; \overline{5,6}\} = \{\overline{1,2,3,4}; \overline{5,6}\},$$

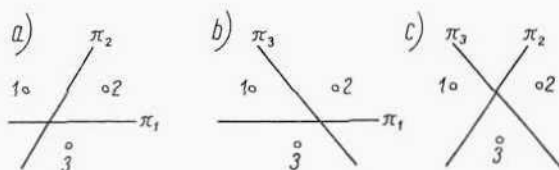
gdyż, skoro $\{1,2\}$ z (π_1) jest zawarte w bloku m (z π), to i $\{1,3\} \cup \{2,4\}$ (z π_2) zawarte są w m , a także $\{3,4\}$ (z π_1) jest zawarte w m , czyli

$$m = \{1,2\} \cup \{1,3\} \cup \{2,4\} \cup \{3,4\} = \{1,2,3,4\}$$

Podobnie: $\{\overline{1},4; \overline{3}; \overline{2},5\} + \{\overline{1},3; \overline{2},4; \overline{5}\} = 1$

Dla dowolnych π_1 i π_2 jest $\pi_1 \cdot \pi_2 \leq \pi_1$ oraz $\pi_1 + \pi_2 \geq \pi_1$.

Można zauważyć, że w wariacie A kodowania trzech stanów, pierwszy bit swą wartością 0 albo 1 wprowadza podział $\pi_1 = \{1,2; \overline{3}\}$, a drugi bit — $\pi_2 = \{\overline{1}; \overline{2},3\}$. Na rys. 4-20a pokazano to w sposób graficzny.



Rys. 4-20. Podziały zbioru o trzech elementach; wersje podziałów przydatnych do kodowania

Każda para stanów jest tu przedzielona linią podziału, co oznacza, że nie będzie dwóch stanów zakodowanych w taki sam sposób. Warunek ten można też sprawdzić za pomocą iloczynu podziałów; gdy daje on w wyniku 0 — kodowanie zgodne z podziałami przypisze każdemu stanowi inny kod.

Opisanie wartości Q za pomocą podziału π jest możliwe wtedy, gdy podział π ma dwa bloki, przy czym nie jest istotne któremu z nich przypisze się wartość 0, a któremu 1. W przypadku trzech stanów istnieją tylko trzy podziały dwublokowe:

$$\pi_1 = \{\overline{1},2; \overline{3}\}, \quad \pi_2 = \{\overline{1}; \overline{2},3\} \quad \text{oraz} \quad \pi_3 = \{\overline{1},3; \overline{2}\}$$

Ponieważ $\pi_1 \cdot \pi_2 = 0$, $\pi_1 \cdot \pi_3 = 0$ oraz $\pi_2 \cdot \pi_3 = 0$, więc każde dwa z tych trzech podziałów mogą być wykorzystane do kodowania. W ten właśnie sposób powstały wersje A , B i C (rys. 4-20). Wszystkie pozostałe wersje opisuje się również tymi samymi podziałami, a różnica polega jedynie na zamianie miejsc (π_2, π_1 zamiast π_1, π_2) lub wartości (0 zamiast 1 i przeciwnie), co nie zmienia stopnia przydatności kodu.

W przypadku układów o czterech stanach wewnętrznych, podziałów dwublokowych może być więcej, ale nie wszystkie one są przydatne do opisywania wartości Q , przy zachowaniu warunku $k = 2$. Na przykład dla podziału $\pi_1 = \{1; 2,3,4\}$ nie można dobrać takiego podziału π_2 , by było $\pi_1 \cdot \pi_2 = 0$, więc do kodowania nie wystarczyłyby dwa sygnały Q . Jeśli chce się zachować $k = 2$, blok każdego podziału nie może mieć więcej niż 2 elementy, tak by mogły one być rozdzielone przez drugi podział, przyjęty dla opisu Q . Istnieją tylko trzy takie podziały:

$$\pi_1 = \{1,2; 3,4\}, \quad \pi_2 = \{1,3; 2,4\} \quad \text{oraz} \quad \pi_3 = \{1,4; 2,3\}$$

Ponieważ $\pi_1 \cdot \pi_2 = 0$, $\pi_1 \cdot \pi_3 = 0$ i $\pi_2 \cdot \pi_3 = 0$, więc każde dwa z tych podziałów mogą być zastosowane do opisu kodu i istnieją trzy nierównoważne warianty kodowania:

	A	B	C
	$\pi_1 \pi_2$	$\pi_1 \pi_3$	$\pi_2 \pi_3$
1 —	00	00	00
2 —	01	01	11
3 —	10	11	01
4 —	11	10	10

Gdy liczba stanów K jest większa od 4, przydatność podziałów do określenia nierównoważnych kodów jest jeszcze większa, ale i trudności z ich wyznaczeniem rosną, gdyż wzrasta gwałtownie liczba możliwych wariantów. Ich wypisywanie nie jest celowe, gdyż i tak porównywanie bezpośrednie jest niemożliwe, trzeba natomiast pamiętać jakiego typu podziały mogą być użyteczne. Podziały, na podstawie których można wybrać wartość Q , będą nazywane *prawidłowymi* i muszą spełniać dwa warunki:

- liczba bloków wynosi 2,
- liczba elementów w bloku nie przekracza 2^{k-1} .

Drugi warunek wynika z konieczności uzyskania iloczynu k podziałów równego 0. Za k przyjmuje się wartość minimalną. Na przykład dla $K = 5$, będzie $k = 3$ i podział $\{1; 2,3,4,5\}$ jest prawidłowy, natomiast dla $K = 6$ również $k = 3$, ale podział $\{1; 2,3,4,5,6\}$ nie jest prawidłowy. Podziały prawidłowe będą oznaczane literą τ .

Problem kodowania K stanów wewnętrznych sprowadza się więc do wyboru takich k podziałów prawidłowych, żeby ich iloczynem było 0 i aby złożoność realizacji układu była możliwie mała. Ten drugi warunek może być badany różnymi sposobami, które zostaną opisane w takiej kolejności, w jakiej należy je stosować w procesie kodowania.

1. Układ sekwencyjny składa się z trzech podstawowych części (rys. 4-1) i złożoność każdej z nich ma istotny wpływ na złożoność całości układu. Przez wstępne przyjęcie minimalnej wartości k , został już zminimalizowany układ pamięciowy; inne problemy jego złożoności wiążą się z konkretnymi realizacjami technicznymi elementów pamięciowych i będą opisane dalej. Względnie łatwo można oszacować złożoność układu kombinacyjnego wyjściowego, realizującego funkcję λ . Opisujące ten układ równanie (4-5) można, po zastąpieniu stanów sygnałami, zapisać w postaci rodziny równań:

$$y_i = \lambda_i(Q_1, Q_2, \dots, Q_k, x_1, x_2, \dots, x_n) \quad (i = 1, 2, \dots, m) \quad (4-12)$$

Zależności sygnałów y_i od x_j (w układzie Mealy'ego) nie da się zmniejszyć, gdyż wynika ona z procesu minimalizacji stanów k , można jednak tak dobrać kod stanów wewnętrznych, czyli konkretne wartości Q , by realizacja funkcji logicznych λ_i była najprostsza. Trudność polega na tym, że trzeba to zrobić bez wypisywania wielu różnych kodów i funkcji. Aby problem uprościć przyjmuje się, że realizacja funkcji jest tym prostsza, im mniej zmiennych ma ta funkcja, a więc drogą do uproszczenia realizacji λ_i jest wyrugowanie możliwie największej liczby zmiennych Q . W najlepszym przypadku — gdy jakiś sygnał y_a zależy od jednej tylko zmiennej Q_b , wartości y_a i Q_b w poszczególnych wierszach tablicy są identyczne albo różnią się o negację, czyli ogólnie, podziały zbioru stanów A , wprowadzone wartościami y_a i Q_b , muszą być równe. Na przykład w tablicy z rys. 4-19a wartość y wprowadza podział stanów A

$$\pi(y) = \{\overline{0,2}; \overline{1,3}\}$$

Jeśli taki sam podział przyjmie się dla jednego z sygnałów Q (np. Q_1), to będzie

$$y = Q_1 \quad \text{albo} \quad y = \overline{Q_1}$$

Podział wprowadzony przez wartość y_i będzie nazywany *podziałem sewnętrznym* i oznaczany przez $\pi_j(y_i)$, przy czym indeks j świadczy o tym,

że podział dotyczy stanu wejść X_j . Indeks ten występuje tylko w przypadku układów Mealy'ego i może mieć postać liczby dwójkowej lub dziesiętnej. Na przykład dla tablicy z rys. 4-17c będzie

$$\pi_0(y) = \{\overline{1,2,3}; \overline{4}\}$$

$$\pi_1(y) = \{\overline{1,4}; \overline{2,3}\}$$

Najlepiej jest, gdy podziały przyjęte do kodowania (τ_r) są takie same jak podziały zewnętrzne, tzn. gdy

$$\pi_j(y_i) = \tau_r$$

Jeśli $\pi_j(y_i)$ nie jest podziałem prawidłowym, ale jest spełniona zależność

$$\pi_j(y_i) \geq \tau_r \cdot \tau_s$$

kodowanie stanów A zgodnie z podziałami τ_r oraz τ_s spowoduje, że y_i będzie zależeć tylko od zmiennych Q_r oraz Q_s i ewent. (x_1, x_2, \dots, x_n) . Wynika to z faktu, że każdy z czterech bloków podziału $\tau_r \tau_s$ może być jednoznacznie opisany iloczynem logicznym typu $Q_r Q_s (Q_r \overline{Q_s}$ itp.).

Jeśli dla tego samego y_i ale różnych x_j podziały $\pi_j(y_i)$ są różne, funkcja y_i będzie iloczynem logicznym albo sumą członów, o postaci wynikającej z $\pi_j(y_i)$.

Na przykład w przypadku tablicy z rys. 4-17c podziały zewnętrzne wskazują na celowość przyjęcia kodu z podziałami:

$$\tau_1 = \pi_1(y) = \{\overline{1,4}; \overline{2,3}\}$$

oraz

$$\tau_2 = \{\overline{1,2}; \overline{3,4}\} \quad \text{albo} \quad \{\overline{1,3}; \overline{2,4}\}$$

gdyż wtedy

$$\pi_0(y) \geq \tau_1 \tau_2 = 0$$

Będzie więc

$$y = f_1(Q_1, x) + f_2(Q_1, Q_2, x)$$

albo

$$y = f_3(Q_1, x) \cdot f_4(Q_1, Q_2, x)$$

przy czym f_1 i f_2 są iloczynami, a f_3 i f_4 — sumami zmiennych z negacją lub bez.

Korzyści wynikające z uwzględniania podziałów zewnętrznych są szczególnie widoczne w układach Moore'a. W wielu takich układach (np. w licznikach) kody stanów wewnętrznych przyjmuje się wyłącznie na podstawie podziałów zewnętrznych, oczywiście pod warunkiem, że iloczyn tych podziałów jest zerem. W innych przypadkach podziały $\pi_j(y_i)$ mają charakter pomocniczy.

2. Uproszczenie drugiego bloku kombinacyjnego w układzie sekwencyjnym (UK1 na rys. 4-1) jest trudniejsze. Zależność (4-3) można przepisać w postaci

$$q_i = \gamma_i(Q_1, Q_2, \dots, Q_k, x_1, x_2, \dots, x_n) \quad (i = 1, 2, \dots) \quad (4-13)$$

a problemem uproszczenia realizacji sprowadzić — tak jak poprzednio — do problemu rugowania z tych równań zmiennych Q . Postać funkcji γ_i zależy w dużym stopniu od rodzaju zastosowanych elementów pamięciowych, ale ogólnie można przyjąć, że złożoność tych funkcji jest proporcjonalna do złożoności funkcji logicznych, opisujących zakodowaną tablicę przejść (tzn. tablicę z sygnałami Q zamiast stanów A). Założenie to znajdzie potwierdzenie w dalszych etapach syntezy, a wynika z niego wniosek, że dla uproszczenia funkcji γ_i , trzeba tablicę przejść analizować pod kątem takiej zamiany wierszy miejscami, by ich późniejsze zakodowanie kodem Gray'a dało najprostsze funkcje opisujące tablicę. Jedną z czynności, która może w tym pomóc, jest takie rozlokowanie wierszy, by jednakowe symbole stanów znalazły się obok siebie, co umożliwi później utworzenie odpowiednich grup Karnaugh'a. Zamiast prób zmiany wierszy w tablicy, wygodniej będzie zastosować rachunek podziałów. Jeśli istnieją takie stany A_a, A_b i A_c , że dla jakiegoś X_j

$$\delta(A_a, X_j) = A_c \quad \text{oraz} \quad \delta(A_b, X_j) = A_c$$

to tworząc ze stanów A_a oraz A_b blok, uzyskuje się *podział wewnętrzny* $\pi(X_j)$. Jeśli do kodowania przyjmie się taki podział τ_r , że

$$\tau_r \geq \pi(X_j)$$

to stany A_a oraz A_b znajdą się w wierszach o identycznej wartości Q_r , co uprości funkcję opisującą stan A_c . Na przykład dla tablicy z rys. 4-19a będzie:

$$\pi(X_0) = \pi(X_1) = \pi(X_2) = \pi(X_3) = \{\overline{0,1}; \overline{2,3}\}$$

i przyjęcie tego podziału do kodowania w istotny sposób uprości realizację, gdyż we wszystkich kolumnach jednakowe symbole znajdują się obok siebie. Oczywiście nie zawsze wnioski są tak jednoznaczne i optymistyczne. Na przykład dla tablicy z rys. 4-17c otrzymuje się

$$\pi(X_0) = \{\overline{1,3,4}; \overline{2}\} \quad \pi(X_1) = \{\overline{1}; \overline{2}; \overline{3}; \overline{4}\} = 0$$

Ponieważ nie ma takiego podziału prawidłowego τ , by było $\tau \geq \pi(X_0)$, a dla dowolnego τ jest $\tau \geq \pi(X_1) = 0$, więc te podziały wewnętrzne nie dają żadnej użytecznej informacji (dla $k = 2$). Jeśli większość podziałów wewnętrznych ma zbyt duże bloki — może to stanowić wskazówkę, że zwiększenie k uprości wyraźnie układ γ . Podział wewnętrzny równy 1, podobnie jak 0, nie jest użyteczny.

Rozmieszczenie stanów w kolumnach tablicy może być wykorzystane w jeszcze inny sposób. Gdy dla jakiegoś X_j jest $\pi(X_j) \neq 0$, znaczy to, że w kolumnie X_j występują nie wszystkie symbole stanów wewnętrznych, można więc utworzyć *podział wtórny* $\pi_w(X_j)$, składający się z bloku stanów występujących w kolumnie X_j oraz stanów pozostałych ujętych w nawias. Jeśli istnieje podział prawidłowy nie mniejszy od $\pi_w(X_j)$, to przyjęcie go do kodowania sprawi, że w całej kolumnie X_j odpowiednia zmienna Q będzie albo zerem albo jedynką, co uprości funkcję. Na przykład z tablicy na rys. 4-19a otrzymuje się:

$$\begin{aligned} \pi_w(X_0) &= \{\overline{0,1}; (\overline{2,3})\} & \pi_w(X_1) &= \pi_w(X_2) = \{\overline{1,2}; (\overline{0,3})\} \\ \pi_w(X_3) &= \{\overline{2,3}; (\overline{0,1})\} \end{aligned}$$

więc istnieją podziały τ takie, że

$$\begin{aligned} \tau_1 &= \{\overline{0,1}; \overline{2,3}\} \geq \pi_w(X_0) & \text{ i } & \tau_1 \geq \pi_w(X_3) \\ \tau_2 &= \{\overline{0,3}; \overline{1,2}\} \geq \pi_w(X_1) & \text{ i } & \tau_2 \geq \pi_w(X_2) \end{aligned}$$

Ponieważ poprzednio uzyskano

$$\pi(y) = \{\overline{0,2}; \overline{1,3}\} = \tau_3 \quad \text{ oraz } \quad \pi(X) = \{\overline{0,1}; \overline{2,3}\} = \tau_1$$

powstaje problem, które z tych podziałów wybrać. Nie ulega wątpliwości, że τ_1 , jako powtarzający się najczęściej, musi zostać wybrany, ale pozostałe podziały dają dwa warianty:

- τ_1 i τ_2 — prostszy UK1, bardziej złożony UK2,
- τ_1 i τ_3 — prostszy UK2, bardziej złożony UK1.

Ponieważ sygnał wyjściowy jest tylko jeden, więc układ UK2 zawsze będzie względnie prosty i bardziej uzasadnione jest przyjęcie wariantu pierwszego.

3. W dotychczasowych rozważaniach wersje kodowania upraszczające realizację były otrzymywane na podstawie podziałów, powstałych z odrębnej analizy każdej kolumny tablic przejść lub wyjść. Nie ulega jednak wątpliwości, że również poziome związki w tablicy przejść w istotny sposób wpływają na efektywność kodowania, jako że grupy Karnaugh tworzy się nie tylko przez łączenie kratek pionowych, lecz także poziomych. Bardzo przydatną, syntetyczną ocenę związków w tablicy przejść można uzyskać, badając przejścia ze stanu do stanu pod wpływem różnych X . Na przykład w tablicy z rys. 4-21 para stanów $\{1,2\}$ przechodzi pod wpływem X_0 do stanów $\{2,4\}$, a para $\{3,4\}$ — do $\{1,3\}$, co można zapisać w formie przejść między podziałami:

$$\{\overline{1,2}; \overline{3,4}\} \xrightarrow{X_0} \{\overline{1,3}; \overline{2,4}\}$$

Ponieważ te same podziały występują po prawej stronie również dla pozostałych X , można napisać, że

$$\{\overline{1,2}; \overline{3,4}\} \rightarrow \{\overline{1,3}; \overline{2,4}\} \quad \text{lub} \quad \tau_1 \rightarrow \tau_2$$

przy czym zapis ten będzie oznaczał, że każda para stanów z każdego bloku w τ_1 przechodzi pod wpływem dowolnego X w parę stanów (niekoniecznie różnych) z jakiegoś bloku w τ_2 , przy czym liczba bloków w τ_2 nie może już być zwiększona. Tego typu podziały są też nazywane *parą podziałów* i oznaczane przez (τ_1, τ_2) . W parach takich warunek dotyczący liczby bloków w τ_2 nie jest obowiązujący, jest więc parą $(\tau_1, 1)$. Ogólnie — gdy (τ_1, τ_2) jest parą, to parami są także (π_1, τ_2) i (τ_1, π_2) , jeśli $\pi_1 \leq \tau_1$ oraz $\pi_2 \geq \tau_2$.

Jeśli $\pi_1 \xrightarrow{X_0} \pi_2$ oraz $\pi_1 \xrightarrow{X_1} \pi_3$, to $\pi_1 \xrightarrow{X_0, X_1} \pi_2 + \pi_3$, a gdy (jak w tablicy z rys. 4-19a)

$$\{0,3\} \xrightarrow{X_0} \{0,1\} \quad \text{i} \quad \{1,2\} \xrightarrow{X_0} \{0,1\}$$

to

$$\{\overline{0,3}; \overline{1,2}\} \xrightarrow{X_0} \{\overline{0,1}; \overline{2}; \overline{3}\}$$

Dla tablicy z rys. 4-17c otrzymuje się więc

$$\{\overline{1,2}; \overline{3,4}\} \xrightarrow{x_0} \{\overline{3,4}; \overline{1}; \overline{2}\}, \quad \{\overline{1,2}; \overline{3,4}\} \xrightarrow{x_1} \{\overline{1,2}; \overline{3,4}\}$$

czyli

$$\{\overline{1,2}; \overline{3,4}\} \rightarrow \{\overline{1,2}; \overline{3,4}\}$$

a zatem

$$\{\overline{1,2}; \overline{3,4}\} \rightleftharpoons \{\overline{1,2}; \overline{3,4}\}$$

W takim przypadku, gdy (π, π) jest parą podziałów, mówi się, że π ma *własność zastępczą*.

Przydatność tego rodzaju analiz do kodowania polega na tym, że gdy $\tau_r \rightarrow \tau_s$ i obydwie te podziały są użyte do kodowania Q_r i Q_s , to

a)		$x_0 \quad x_1 \quad x_3 \quad x_2$					b)		$q_1 q_2 \quad x_0 \quad x_1 \quad x_3 \quad x_2$				
A	1	2	4	3	1		$Q_1 Q_2$	1-00	01	11	10	00	
	2	4	2	1	3			2-01	11	01	00	10	
	3	1	3	4	2			4-11	10	00	01	11	
	4	3	1	2	4			3-10	00	10	11	01	
													$Q_1' Q_2'$

Rys. 4-21. Przykład tablicy przejść: a) przed kodowaniem; b) po zakodowaniu

w równaniach typu (4-13), dotyczących elementu Q_s , jedyną zmienną Q będzie Q_r . Wynika to z faktu, że każdy blok w τ_s i odpowiadająca mu wartość Q_s , są jednoznacznie określone przez odpowiedni blok w τ_r i jego wartość Q_r , będącą współrzędną bloku z τ_s . Wobec tego, jeśli $\tau_1 \rightleftharpoons \tau_2$, to wzbudzenia elementu Q_1 zależą tylko od Q_2 i X , a wzbudzenie elementu Q_2 zależą tylko od Q_1 i X , bez względu na to, jak dużo jest elementów Q .

Na przykład dla tablicy z rys. 4-21a będzie

$$\tau_1 = \{\overline{1,2}; \overline{3,4}\} \rightleftharpoons \{\overline{1,3}; \overline{2,4}\} = \tau_2$$

$$\tau_3 = \{\overline{1,4}; \overline{2,3}\} \rightleftharpoons \{\overline{1,4}; \overline{2,3}\} = \tau_3$$

Przyjęcie do kodowania τ_1 i τ_3 albo τ_2 i τ_3 uprościłoby tylko wzbudzenia Q_3 , natomiast kodowanie wg podziałów τ_1 i τ_2 uprości zarówno

wzbudzenia Q_1 jak i Q_2 . Odpowiednio zakodowana tablica jest przedstawiona na rys. 4-21b. Rozkład zer i jedynek wskazuje na duże możliwości minimalizacji funkcji opisującej tę tablicę.

Jak wynika z powyższych rozważań, najbardziej korzystne dla uproszczenia realizacji układu są przypadki, w których udaje się znaleźć pary podziałów typu

$$\tau_1 \rightleftharpoons \tau_2$$

albo dłuższe łańcuchy zamknięte, np.

$$\tau_1 \rightarrow \tau_2 \rightarrow \tau_3 \rightarrow \tau_1$$

Nieco jest gorzej, gdy obowiązują tylko przejścia typu

$$\tau_1 \rightarrow \tau_2 \quad \text{lub} \quad \tau_2 \rightleftharpoons \tau_2$$

gdyż wtedy uproszczona będzie tylko realizacja wzbudzeń Q_2 . Jeszcze mniejsze efekty uzyskuje się, gdy lewy podział pary nie jest prawidłowy

$$\pi \rightarrow \tau_1 \quad \text{oraz} \quad \pi \geq \tau_2 \tau_3$$

Kodowanie wg τ_1 , τ_2 i τ_3 sprawia wówczas, że wzbudzenia Q_1 zależą tylko od Q_2 , Q_3 i X . Zależności te można rozszerzyć na iloczyn trzech i więcej podziałów, ale korzystać z ich stosowania staje się problematyczna. Oszacowanie zysków wynikających z kodowania wg par podziałów jest, jak widać, proste, bardzo trudne jest natomiast znalezienie tych par. Stosuje się tu różne metody, wszystkie złożone i pracochłonne. Niżej przedstawiony sposób postępowania wydaje się najbardziej praktyczny.

Można wykazać, że warunkiem koniecznym istnienia pary podziałów (τ_j, τ_i) jest, aby dla każdego X było

$$\tau_i \geq \pi(X) \quad \text{lub} \quad \tau_i \geq \pi_w(X)$$

Dzieje się tak dlatego, że przy niezerowych podziałach wewnętrznych istnieją co najmniej dwa takie stany A_a i A_b , które przechodzą w stan A_c , jeśli więc istnieje przejście $\tau_j \rightarrow \tau_i$, to stany A_a i A_b muszą należeć do jednego bloku w τ_j , lub też w τ_i występuje blok, zawierający wszystkie stany następne przy danym X .

Dla układu o dwóch stanach wejść (X_0, X_1) warunki konieczne istnienia pary podziałów (τ_j, τ_i) można więc zapisać wyrażeniem:

$[\tau_j \geq \pi(X_0) \text{ lub } \tau_i \geq \pi_w(X_0)]$ i $[\tau_j \geq \pi(X_1) \text{ lub } \tau_i \geq \pi_w(X_1)]$
z którego wynikają cztery możliwości:

1. $\tau_j \geq \pi(X_0)$ oraz $\tau_j \geq \pi(X_1)$, czyli $\tau_j \geq \pi(X_0) + \pi(X_1)$,
2. $\tau_j \geq \pi(X_0)$ oraz $\tau_i \geq \pi_w(X_1)$,
3. $\tau_i \geq \pi_w(X_0)$ oraz $\tau_j \geq \pi(X_1)$,
4. $\tau_i \geq \pi_w(X_0)$ oraz $\tau_i \geq \pi_w(X_1)$, czyli $\tau_i \geq \pi_w(X_0) + \pi_w(X_1)$.

Przy większej liczbie stanów X możliwości będzie odpowiednio więcej, ale zazwyczaj tylko niektóre nierówności mogą być spełnione i liczba kombinacji sugerujących podziały tworzące pary nie jest duża.

Jeśli dla wszystkich stanów X zachodzi

$$\tau_i \geq \pi_w(X)$$

to istnieje para $(1, \tau_i)$. Oznacza to, że zakodowana zgodnie z τ_i zmienna Q_i nie zależy od żadnej zmiennej Q , lecz tylko od stanu X .

Jeśli dla jakiegoś X podział wewnętrzny jest zerowy, to każdy podział τ jest od niego większy i nie istnieje podział τ nie mniejszy od podziału wtórnego. Kolumna tablicy odpowiadająca takiemu stanowi X nie wnosi więc żadnych informacji przydatnych do wyznaczania par podziałów. Na przykład w tablicy z rys. 4-17c tylko pierwsza kolumna jest interesująca. Otrzymuje się z niej

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

Nie istnieje podział τ_j taki, że $\tau_j \geq \pi(X_0)$, ale istnieje $\tau_i = \{\overline{1,2}; \overline{3,4}\}$ taki, że $\tau_i \geq \pi_w(X_0)$, można więc przypuszczać, że zachodzi tu przejście

$$\tau_j \rightarrow \{\overline{1,2}; \overline{3,4}\}$$

Z drugiej kolumny tablicy wynika, że do bloku $(1,2)$ można przejść tylko ze stanów $1,2$, a do bloku $(3,4)$ — tylko ze 3 i 4 , a zatem:

$$\tau_1 = \{\overline{1,2}; \overline{3,4}\} \neq \{\overline{1,2}; \overline{3,4}\} = \tau_1$$

Przyjmując drugi podział w postaci

$$\tau_2 = \{\overline{1,4}; \overline{2,3}\}$$

otrzymuje się kod optymalny.

Jeśli wszystkie podziały wewnętrzne tablicy są zerowe, trzeba szukać par podziałów w inny sposób. Najlepiej byłoby dla wszystkich

możliwych podziałów prawidłowych τ_i znaleźć takie π_i , by obowiązywało przejście

$$\pi_i \rightarrow \tau_i$$

Potem należałoby wybrać prawidłowe π_i , a pozostałe zbadać, czy nie są iloczynami niektórych τ_i . Droga ta jest uciążliwa i dlatego, jeśli można poprzestać na rozwiązaniach zbliżonych do optymalnych, wygodniej jest zastosować działanie uproszczone. Polega ono na poszukiwaniu takich τ'_i , by było

$$\tau_i \rightarrow \tau'_i$$

czyli na wybieraniu ze wszystkich możliwych podziałów prawidłowych tylko tych, które przechodzą w podziały prawidłowe. Oczywiście, najchętniej są wybierane takie podziały, że

$$\tau_i \neq \tau'_i \quad \text{albo} \quad \tau_i = \tau'_i$$

W praktyce tablice przejść z wyłącznie zerowymi podziałami wewnętrznymi występują zwykle w układach licznikowych, w których kod jest najczęściej określany samymi podziałami zewnętrznymi. Konieczność poszukiwania wszystkich podziałów τ_i występuje więc bardzo rzadko.

4. Zdarza się niekiedy, że żadna z podanych wyżej metod nie daje jednoznacznego rozwiązania problemu kodowania. W wyborze optymalnego wariantu może być wówczas pomocna metoda minimalizacji liczby zmian stanu pamięci. Opiera się ona na przypuszczeniu, że zakodowana tablica przejść, w której liczba zmian zera w jedynekę i jedynki w zero jest minimalna, będzie miała prostszą realizację niż wszystkie inne. Przypuszczenie to okazuje się słuszne, ale pod warunkiem, że uwzględniono już wszystkie wskazania, wynikające z podziałów wewnętrznych i par podziałów.

Minimalizacja liczby zmian polega na takim doborze kodu, by kod stanu A_a różnił się na możliwie najmniejszej liczbie pozycji od kodów stanów A_b, A_c, \dots , do których A_a przechodzi pod wpływem różnych stanów X . Minimalizacja liczby zmian w całej tablicy musi uwzględniać sumę zmian poszczególnych stanów. Przy ocenie różnych wariantów kodu pod tym kątem bardzo przydaje się uproszczony graf (wykres) przejść, w którym liczba połączeń między punktami a i b równa jest liczbie przejść z A_a do A_b i z A_b do A_a w tablicy przejść. Dla tablicy

z rys. 4-17c odpowiedni wykres jest przedstawiony na rys. 4-17d. Wyżej pokazano, że bardzo korzystny jest w tym przypadku podział $\{1,2; \overline{3},4\}$, ale — dla minimalizacji UK1 — pozostaje otwarty problem drugiego podziału. Może to być $\{\overline{1},3; \overline{2},4\}$ albo $\{1,4; \overline{2},3\}$. Z rys. 4-17d wynika, że w drugim przypadku mniej linii łączących ulegnie przerwaniu (przy podziale), więc i liczba zmian będzie mniejsza. Podział ten jest taki sam jak $\pi_i(y)$, co potwierdza jeszcze dokonany uprzednio wybór.

Gdy liczba stanów K jest większa niż 4, wygodnie jest wykres przejść rysować w postaci kostek, starając się, by możliwie najwięcej linii łączących stany znajdowało się na krawędziach kostek. Podziały stanów sprowadzają się wówczas do wyboru płaszczyzny przekroju kostki.

Przy rozwiązywaniu zadań praktycznych należy wybierać podział w następującej kolejności:

- 1) tworzące pary (najlepiej — obustronne),
- 2) powstające z podziałów wewnętrznych lub zewnętrznych,
- 3) dające minimalną liczbę zmian stanu elementów pamięci.

Opisane metody wyboru kodu zostaną teraz zilustrowane przykładami.

Z tablicy na rys. 4-16d można otrzymać:

$$\pi_0(y) = \{\overline{1},2; \overline{3}\} = \tau_1 \qquad \pi_1(y) = \{\overline{1},3; \overline{2}\} = \tau_2$$

$$\pi_2(y) = \tau_1 \qquad \pi_3(y) = \{\overline{1}; \overline{2},3\} = \tau_3$$

$$\pi(X_1) = \{\overline{1},2,3\} \qquad \pi_w(X_1) = \{\overline{2}; (\overline{1},3)\}$$

$$\pi(X_3) = \{\overline{1}; \overline{2},3\} = \tau_3 \qquad \pi_w(X_3) = \{\overline{2},3; (\overline{1})\} \leq \tau_3$$

Parę mógłby tworzyć podział τ_3 z podziałem niemniejszym od $\pi_w(X_1)$, ale jak wynika z tablicy

$$\tau_3 \rightarrow 1$$

Wobec braku użytecznych par podziałów najlepiej będzie wybrać podziały powtarzające się najczęściej, tzn. τ_1 i τ_3 , tworząc na przykład kod

	τ_1	τ_3
1	0	0
2	0	1
3	1	1

Z tablicy na rys. 4-22a (bez uwzględniania sygnałów wyjściowych) otrzymuje się:

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

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

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

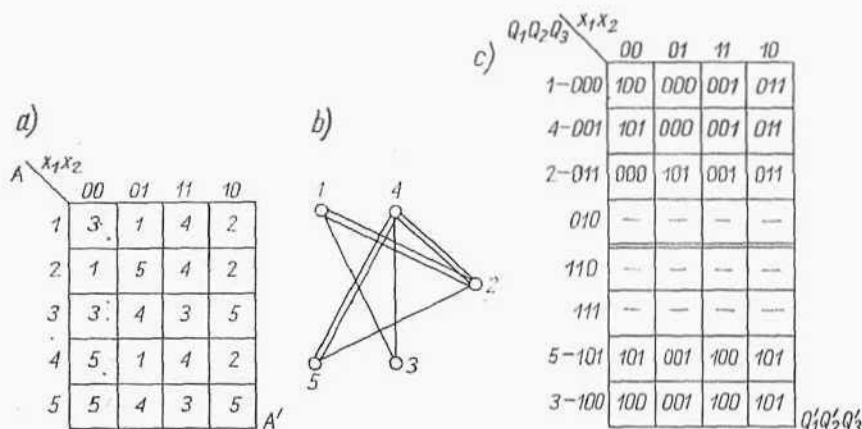
$$\pi_w(X_1) = \{\overline{1,4,5}; (\overline{2,3})\}$$

$$\pi(X_2) = \{\overline{1,2,4}; \overline{3,5}\} = \tau_1$$

$$\pi_w(X_2) = \{\overline{2,5}; (\overline{1,3,4})\}$$

$$\pi(X_3) = \tau_1$$

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



Rys. 4-22. Przykład kodowania: a) tablica przejść przed kodowaniem; b) uproszczony wykres przejść; c) zakodowana tablica przejść

Jeśli przy poszukiwaniu pary (τ_j, τ_i) za τ_j weźmie się podział nie mniejszy od $\pi(X_0)$, to τ_i musi być nie mniejszy od podziałów wtórnych dla X_1, X_2 i X_3 , co jest niemożliwe. Podobna analiza innych podziałów wewnętrznych wskazuje na jedną tylko możliwość: gdy $\tau_j = \tau_1$ oraz $\tau_i \geq \pi_w(X_0)$.

Badanie tablicy potwierdza, że

$$\tau_1 = \{\overline{1,2,4}; \overline{3,5}\} \rightarrow \{\overline{1,3,5}; \overline{2,4}\} = \tau_2$$

Ponieważ

$$\tau_1 \tau_2 = \{\overline{1}; \overline{2,4}; \overline{3,5}\}$$

więc trzeci podział musi oddzielić 2 od 4 oraz 3 od 5. Do jego wyznaczenia

warto wykorzystać określone wyżej podziały. Na przykład dla $\tau_3 = \{1,2,5; \overline{3,4}\}$ jest $\tau_1 \cdot \tau_2 \cdot \tau_3 = 0$ oraz $\tau_3 \geq \pi_w(X_2)$ i $\tau_3 \geq \pi_w(X_3)$, co z pewnością uprości realizację. Zalecany kod będzie więc miał następującą postać

	τ_1	τ_2	τ_3
1	0	0	0
2	0	1	0
3	1	0	1
4	0	1	1
5	1	0	0

Podział τ_2 nie jest jedynym podziałem prawidłowym nie mniejszym od $\pi_w(X_0)$; warunek ten spełniają jeszcze $\tau_4 = \{1,2,3,5; \overline{4}\}$ i $\tau_5 = \{1,3,4,5; \overline{2}\}$. Przyjmując do kodowania τ_1 i τ_5 uzyskuje się

$$\tau_1 \tau_5 = \{\overline{1,4}; 2; \overline{3,5}\}$$

skąd wynika, że w tym przypadku trzeci podział powinien mieć postać

$$\{\overline{1,3}; \overline{4,5}, ()\} \text{ albo } \{\overline{1,5}; \overline{3,4}, ()\}$$

Nawias oznacza tu pozostałe elementy, których przynależność do któregoś bloku jest dowolna. Odpowiedni podział można wybrać wśród podziałów wtórnych, można też zastosować minimalizację liczby zmian w tablicy. Z uproszczonego wykresu przejść na rys. 4-22b (pominięto tu połączenia nie mające wpływu na wybór) wynika, że pierwsza postać wprowadzi mniej zmian stanu i że stan 2 lepiej będzie dołączyć do 4,5. Otrzymuje się w ten sposób podział o minimalnej liczbie zmian: $\tau_6 = \{\overline{1,3}; \overline{2,4,5}\}$. Zakodowana tablica przejść dla podziałów τ_1, τ_5, τ_6 jest przedstawiona na rys. 4-22c.

Następnym przykładem niech będzie selektor tetrad, którego graf został utworzony na rys. 4-7c, a odpowiednia tablica — na rys. 4-23a. Podział zewnętrzny jest nieprawidłowy, natomiast pozostałe mają postać:

$$\pi(X_0) = \{\overline{0}; \overline{1}; \overline{2}; \overline{3}; \overline{4,5}\} \quad \pi_w(X_0) = \{\overline{0,1,2,4,5}; (\overline{3})\}$$

$$\pi(X_1) = \{\overline{0}; \overline{1}; \overline{2,3}; \overline{4,5}\} \quad \pi_w(X_1) = \{\overline{0,1,3,5}; (\overline{2,4})\}$$

Wynika stąd, że w parze (τ_j, τ_i) musi być

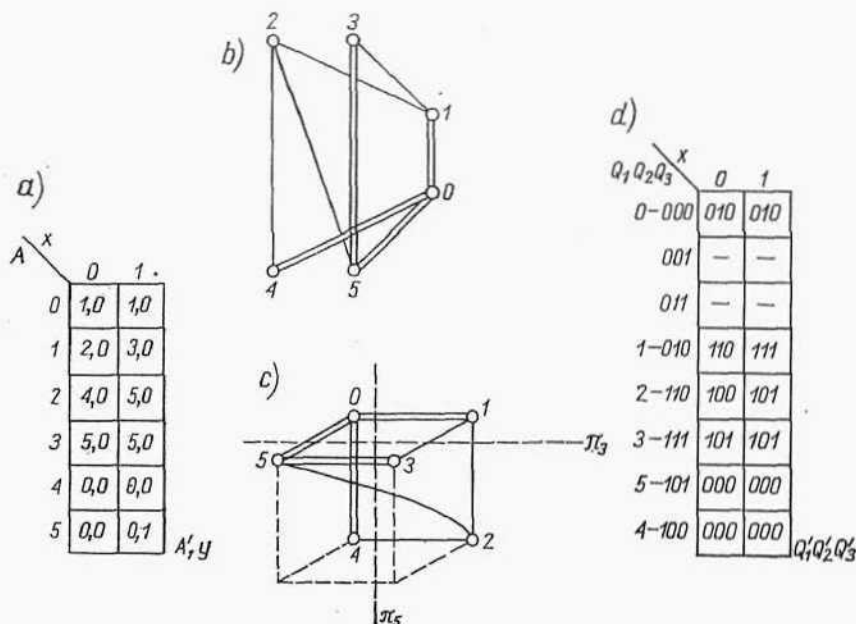
$$\tau_j \geq \pi(X_0) + \pi(X_1) = \pi(X_1)$$

albo

$$\tau_j \geq \pi(X_0) \quad \text{oraz} \quad \tau_i \geq \pi_w(X_1)$$

Dla drugiego warunku przyjmując $\tau_i = \{\overline{0,1,3,5}; \overline{2,4}\}$ uzyskuje się z tablicy $\tau_j = \{\overline{0,3,4,5}; \overline{1,2}\}$, więc

$$\tau_1 = \{\overline{0,3,4,5}; \overline{1,2}\} \rightarrow \{\overline{0,1,3,5}; \overline{2,4}\} = \tau_2$$



Rys. 4-23. Kodowanie selektora tetrad: a) tablica przejść i wyjść; b) wykres przejść; c) wykres przejść w postaci kostki; d) zakodowana tablica przejść

Z pierwszego warunku można wyznaczyć aż 5 różnych τ_j , których przejścia należy zbadać:

$$\tau_3 = \{\overline{0,1}; \overline{2,3,4,5}\} \rightleftharpoons \{\overline{0,4,5}; \overline{1,2,3}\} = \tau_5$$

$$\tau_4 = \{\overline{0,2,3}; \overline{1,4,5}\} \rightleftharpoons \tau_4$$

$$\tau_5 = \{\overline{0,4,5}; \overline{1,2,3}\} \rightleftharpoons \tau_3$$

$$\tau_6 = \{\overline{0,1,2,3}; \overline{4,5}\} \rightarrow \{\overline{0}; \overline{1,2,3,4,5}\} \quad (\text{nieprawidłowy})$$

$$\tau_7 = \{\overline{0,1,4,5}; \overline{2,3}\} \rightarrow \tau_6$$

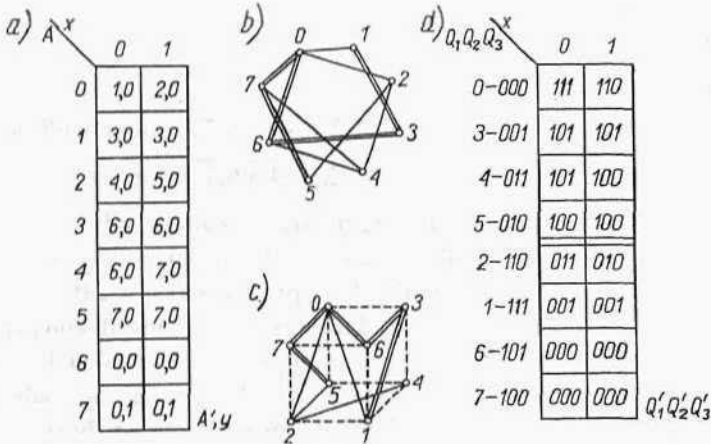
Najbardziej cenne właściwości mają podziały τ_3 , τ_4 i τ_5 , ale

$$\tau_3 \tau_4 \tau_5 = \{\overline{0}; \overline{1}; \overline{2,3}; \overline{4,5}\} \neq 0$$

Z tego samego powodu nie mogą być przyjęte nieco gorsze rozwiązania: τ_1 , τ_2 , τ_4 oraz τ_4 , τ_6 , τ_7 . Wobec tego najlepiej będzie wybrać τ_3 i τ_5 , a trzeci podział wybrać tak, aby rozdzielał 2 od 3 i 4 od 5. Mogą więc wystąpić dwa warianty trzeciego podziału:

$$\{\overline{2,4}; \overline{3,5}, ()\} \quad \text{albo} \quad \{\overline{2,5}; \overline{3,4}, ()\}$$

Z uproszczonego wykresu przejść (rys. 4-23b) wynika, że pierwszy wariant spowoduje mniej zmian stanu elementów pamięci. Trudniej



Rys. 4-24. Kodowanie drugiej wersji selektora tetrad: a) tablica przejść i wyjść, b) wykres przejść; c) wykres przejść w postaci kostki; d) zakodowana tablica przejść

jest natomiast o decyzję dotyczącą pozostałych elementów zbioru stanów, gdyż żadne połączenie nie jest specjalnie wyróżnione; można tylko stwierdzić, że 0 i 1 powinny być w jednym bloku. Wobec tego istnieją dwa podziały równoważne:

$$\tau_8 = \{\overline{0,1,2,4}; \overline{3,5}\} \quad \text{i} \quad \tau_9 = \{\overline{0,1,3,5}; \overline{2,4}\}$$

i dwa równoważne sposoby kodowania; τ_3 τ_5 , i τ_8 oraz τ_3 , τ_5 i τ_9 . Do identycznych wniosków dochodzi się na podstawie innej postaci wykresu

przejsć (rys. 4-23c). Jak widać, płaszczyznę dzielącą niepodzielone stany można poprowadzić dwoma sposobami, wg τ_8 albo τ_9 . Zakodowana tablica (wariant z τ_8) jest przedstawiona na rys. 4-23d.

Inna wersja selektora tetrad była opisywana grafem z rys. 4-8c, któremu odpowiada tablica z rys. 4-24a. Podziały zewnętrzne są tu również bezużyteczne, natomiast pozostałe to:

$$\pi(X_0) = \{\overline{0}; \overline{1}; \overline{2}; \overline{3,4}; \overline{5}; \overline{6,7}\} \quad \pi_w(X_0) = \{\overline{0,1,3,4,6,7}; (\overline{2}, \overline{5})\}$$

$$\pi(X_1) = \{\overline{0}; \overline{1}; \overline{2}; \overline{3}; \overline{4,5}; \overline{6,7}\} \quad \pi_w(X_0) = \{\overline{0,2,3,5,6,7}; (\overline{1}, \overline{4})\}$$

Nie istnieje podział τ_i nie mniejszy od podziałów wtórnych, więc ewentualnej pary trzeba szukać, rozpatrując

$$\tau_j \geq \pi(X_0) + \pi(X_1) = \{\overline{0}; \overline{1}; \overline{2}; \overline{3,4,5}; \overline{6,7}\}$$

Otrzymuje się:

$$\tau_1 = \{\overline{0,3,4,5}; \overline{1,2,6,7}\} \not\Rightarrow \tau_1$$

$$\tau_2 = \{\overline{1,3,4,5}; \overline{0,2,6,7}\} \rightarrow \{\overline{0,1,2,4,5}; \overline{3,6,7}\} \text{ (nieprawidłowy)}$$

$$\tau_3 = \{\overline{2,3,4,5}; \overline{0,1,6,7}\} \rightarrow \{\overline{0,1,2,3}; \overline{4,5,6,7}\} = \tau_4 \rightarrow 1$$

Próba wykorzystania τ_1 , τ_2 i τ_4 nie daje rezultatu, gdyż $\tau_1 \tau_2 \tau_4 \neq 0$, można więc dobrać dwa podziały do τ_1 albo jeden podział do τ_3 i τ_4 . W pierwszym przypadku można wykres przejść z rys. 4-24b przerysować w postaci kostki z rys. 4-24c, w której podział τ_1 jest dokonany płaszczyzną równoległą do płaszczyzny rysunku, a maksymalna liczba linii łączących jest umieszczona na krawędziach. Dokonując podziału kostki dwiema płaszczyznami równoległymi do pozostałych jej boków, otrzymuje się

$$\tau_5 = \{\overline{0,3,6,7}; \overline{1,2,4,5}\} \quad \text{oraz} \quad \tau_6 = \{\overline{0,2,5,7}; \overline{1,3,4,6}\}$$

Po zakodowaniu wg podziałów τ_1 , τ_5 , τ_6 uzyskuje się tablicę z rys. 4-24d.

Przykładem o mniej typowym rozwiązaniu będzie tablica z rys. 4-25, opisana podziałami:

$$\begin{array}{lll} \pi(y) = \{\overline{0}; \overline{1,2,3}\} & \pi(X_0) = \{\overline{0,2,3}; \overline{1}\} & \pi_w(X_0) = \{\overline{0,1}; (\overline{2}, \overline{3})\} \\ & \pi(X_1) = \{\overline{0,1,3}; \overline{2}\} & \pi_w(X_1) = \{\overline{0,2}; (\overline{1}, \overline{3})\} \\ & \pi(X_2) = \{\overline{0,1,2}; \overline{3}\} & \pi_w(X_2) = \{\overline{0,3}; (\overline{1}, \overline{2})\} \\ & \pi(X_3) = 1 & \pi_w(X_3) = \{\overline{0}; (\overline{1}, \overline{2}, \overline{3})\} \end{array}$$

A	$x_1 x_2$						y
	00	01	11	10			
0	1	2	0	3	0		
1	0	2	0	3	1		
2	1	0	0	3	1		
3	1	2	0	0	1		

Rys. 4-25. Przykład tablicy przejść i wyjść

Żaden warunek występowania par podziałów nie może tu być spełniony, pozostają więc 3 możliwości:

- wybrać dwa podziały prawidłowe (dla czterech stanów jest to łatwe) posługując się kryterium minimalnej liczby zmian,
- wybrać podział $\tau_1 = \{\overline{0,3}; \overline{1,2}\}$, gdyż $\tau_1 \geq \pi_w(X_2)$ i $\tau_1 \geq \pi_w(X_3)$, a drugi podział dobrać jak wyżej,
- zrezygnować z wartości minimalnej k i wybrać trzy pierwsze podziały wewnętrzne.

Trzecie rozwiązanie narzuca kod nieminimalny:

0-000

1-100

2-010

3-001

ale jest to wersja najlepsza, jeśli element pamięciowy nie jest zbyt kosztowny lub złożony.

4.2.5. ELEMENTY PAMIĘCI I FUNKCJE WZBUDZEŃ

Pamięć układów sekwencyjnych synchronicznych jest budowana z elementów będących prostymi (elementarnymi) układami Moore'a. Elementy te winny spełniać następujące dwa warunki:

- muszą mieć *pełny system przejść*, tzn. możliwość bezpośredniego przechodzenia z dowolnego stanu (wewnętrznego) do dowolnego stanu;
- muszą mieć *pełny system wyjść*, tzn. każdy stan wewnętrzny powinien odpowiadać innemu stanowi wyjść (stan wyjść jednoznacznie określa stan wewnętrzny).