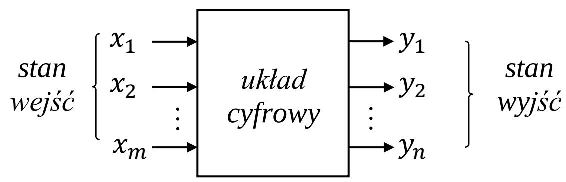
4. Asynchroniczne układy sekwencyjne

4.1. Wprowadzenie

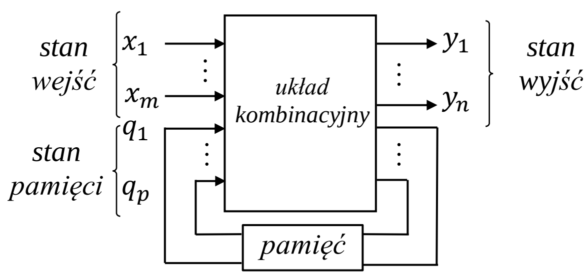
Układy sekwencyjne są układami cyfrowymi, w których zależność między wartościami *sygnałów wejściowych* (tzw. *stan wejść*) i *wyjściowych* (tzw. *stan wyjść*) nie jest jednoznaczna – rys. 4.1.



Rys.4.1. Układ cyfrowy jako „czarna skrzynka”

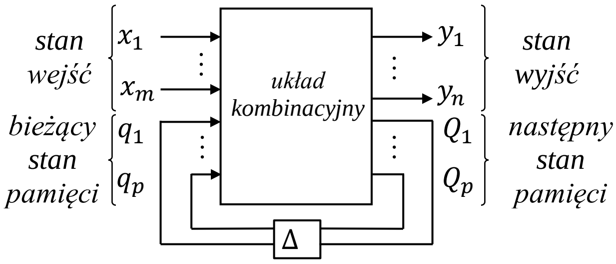
Stan wyjść układu sekwencyjnego zależy nie tylko od aktualnego stanu wejść, ale również od poprzednich stanów wejść, tzn. od kolejności (sekwencji) zmian stanów wejść. Układy sekwencyjne zapamiętują historię oddziaływań sygnałów wejściowych – dlatego też nazywane są *układami z pamięcią*. Pamięć układu, tworzy minimalna liczba wielkości niezbędnych do opisania wszystkich skutków przeszłych oddziaływań. Wielkości te nazywane są *stanem pamięci wewnętrznej* (*stanem pamięci* lub *stanem wewnętrznym*) układu. Ogólny model układu sekwencyjnego z wyodrębnionymi sygnałami reprezentującymi stan pamięci wewnętrznej został przedstawiony na rys. 4.2.

Jeżeli zmiana stanu pamięci może nastąpić bezpośrednio (w dowolnej chwili czasu) pod wpływem zmiany stanu wejść, to układy takie nazywane są *układami asynchronicznymi*. W *układach synchronicznych* zmiana stanu pamięci następuje tylko w ściśle określonych chwilach czasu wyznaczanych przez dodatkowy sygnał wejściowy tzw. *sygnał taktujący* (zegarowy, synchronizujący). Okres zegara musi być dobrany w taki sposób, żeby mogła być wykonana najdłuższa z operacji takiego układu, nawet jeśli jest ona wykonywana bardzo rzadko. Układy asynchroniczne mogą wykrywać moment zakończenia obliczeń, co pozwala im na większą szybkość działania przy mniejszym poborze mocy [13].



Rys.4.2. Model asynchronicznego układu sekwencyjnego

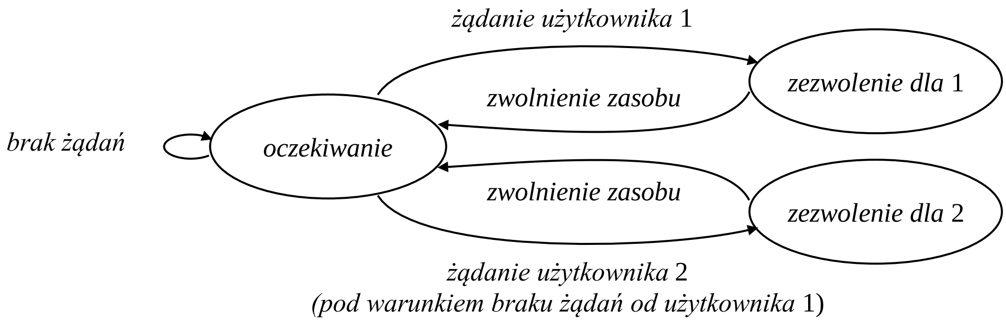
Najwcześniejsze implementacje asynchronicznych układów sekwencyjnych były układami Huffmana opartymi na modelu przedstawionym na rys. 4.3. Huffman w [5] zaproponował model, w którym układ kombinacyjny został rozbudowany o pętle sprzężeń zwrotnych z dodatkowymi elementami opóźniającymi Δ. Takie podejście pozwoliło na realizację pamięci układu sekwencyjnego.



Rys.4.3. Model Huffmana asynchronicznego układu sekwencyjnego

Stany wewnętrzne układu sekwencyjnego mogą mieć charakter stabilny lub niestabilny. Jeśli dla danego stanu wejść sygnały stanu wewnętrznego $q_i = Q_i$ dla każdego $i = 1 \cdots p$, to układ jest w *stabilnym stanie wewnętrznym* lub krótko w *stanie stabilnym*. W wyniku zmiany stanu wejściowego układ kombinacyjny generuje nowe wartości sygnałów stanu Q_i. Ze względu na istnienie opóźnień Δ_i niektóre sygnały $q_i \neq Q_i$, co oznacza, że układ znajduje się w *niestabilnym stanie wewnętrznym* lub krótko w *stanie niestabilnym*. Po upływie czasu Δ_i sygnały stanu wewnętrznego bieżącego i następnego mogą mieć już takie same wartości i w konsekwencji układ będzie w stanie stabilnym. Układ może również znaleźć się w kolejnym stanie niestabilnym i przez serię stanów niestabilnych może przejść do stanu stabilnego. Podstawowym wymaganiem pozwalającym na poprawne funkcjonowanie układu jest dopuszczenie zmian stanu wejściowego tylko w stanach stabilnych. Dodatkowo zakłada się, że nie jest możliwa jednoczesna zmiana kilku sygnałów wejściowych – takie ograniczenia funkcjonowania nazywane są *trybem podstawowym*.

Koncepcja stanu układu będącego konsekwencją oddziaływań sygnałów wejściowych na ten układ została wprowadzona w teorii automatów, w której zdefiniowany został model matematyczny nazywany *automatem skończonym*. Automat skończony jest układem, który może znajdować się w jednym ze skończonej liczby stanów, a każda zmiana wejść automatu powoduje zmianę jego stanu. Działanie automatu skończonego przedstawiane jest za pomocą grafu, którego wierzchołki odpowiadają stanom automatu, a gałęzie oznaczają przejścia pomiędzy stanami wymuszane określonymi sygnałami wejściowymi. Na rys. 4.4 przedstawiony został graf automatu pełniącego funkcję arbitra przydzielającego dwóm użytkownikom dostęp do wspólnego zasobu. Arbiter monitoruje wykorzystanie zasobu i żądania klientów. W przypadku gdy zasób jest wolny i żądanie jest zgłaszanie przez jednego z użytkowników, użytkownik ten dostaje zezwolenie na korzystanie z zasobu. W przedstawionym podejściu użytkownik pierwszy ma wyższy priorytet – jego żądania rozpatrywane są jako pierwsze.



Rys.4.4. Graf stanów układu arbitra

Teoria automatów dzieli automaty skończone na deterministyczne i niedeterministyczne. W automacie deterministycznym zmiana stanu wejść wyznacza jednoznacznie kolejny stan automatu, natomiast w automacie niedeterministycznym może być kilka stanów następnych. Deterministyczny automat skończony definiowany jest formalnie za pomocą „piątki”:

$$(\mathbb{X}, \mathbb{A}, \mathbb{Y}, \delta, \lambda),$$

gdzie

$\mathbb{X}$ to zbiór wszystkich stanów wejściowych (zawiera maksymalnie 2^m stanów, m to liczba sygnałów wejściowych automatu),

$\mathbb{A}$ to zbiór wszystkich stanów automatu (zawiera maksymalnie 2^p stanów, p to liczba elementów tworzących pamięć automatu),

$\mathbb{Y}$ to zbiór wszystkich stanów wyjściowych (zawiera maksymalnie 2^n stanów, n to liczba sygnałów wyjściowych automatu),

δ to *funkcja przejść*, która odpowiada za pamięć układu, a λ to *funkcja wyjść*, która określa stan wyjściowy układu.

Funkcje przejść δ i wyjść λ opisują zależności pomiędzy stanami wejściowymi, wewnętrznymi i wyjściowymi. Funkcja przejść δ wyznacza następny stan wewnętrzny układu na podstawie aktualnego stanu wewnętrznego i stanu wejściowego

$$\delta: \mathbb{A} \times \mathbb{X} \rightarrow \mathbb{A}.$$

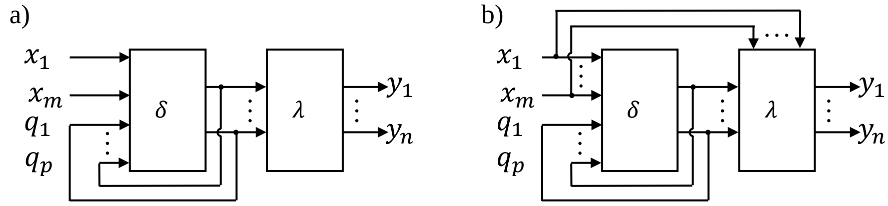
Funkcja wyjść λ wyznacza stan wyjściowy, jej postać określa typ automatu. W *automatach Moore'a* stan wyjściowy jest wyznaczany wyłącznie na podstawie stanu wewnętrznego

$$\lambda: \mathbb{A} \rightarrow \mathbb{Y},$$

w *automatach Mealy'ego* na podstawie stanu wewnętrznego i stanu wejściowego

$$\lambda: \mathbb{A} \times \mathbb{X} \rightarrow \mathbb{Y}.$$

Schematy blokowe układów sekwencyjnych z wyodrębnionymi blokami realizującymi funkcje przejść i wyjść dla automatów Moore'a i Mealy'ego zostały pokazane na rys. 4.5. Układy Moore'a i Mealy'ego są równoważne: każdy automat Moore'a może być przekonwertowany na automat Mealy'ego, a automat Mealy'ego na automat Moore'a. Układy sekwencyjne mogą więc być projektowane w postaci automatów Moore'a lub Mealy'ego, przy czym układy w architekturze Mealy'ego mają zwykle prostszą strukturę. W dalszej części podręcznika rozważane będą wyłącznie układy o architekturze Moore'a.



Rys.4.5. Model układu sekwencyjnego w architekturze a) Moore'a b) Mealy'ego

Zapisując stany wejść i wyjść układu w postaci wektorów

$$\mathcal{X} = (x_1, \cdots, x_m)^T \quad \text{i} \quad \mathcal{Y} = (y_1, \cdots, y_n)^T,$$

a bieżący i następny stan wewnętrzny jako

$$\mathcal{A} = (q_1, \cdots, q_p)^T \quad \text{i} \quad \mathcal{A}' = (Q_1, \cdots, Q_p)^T,$$

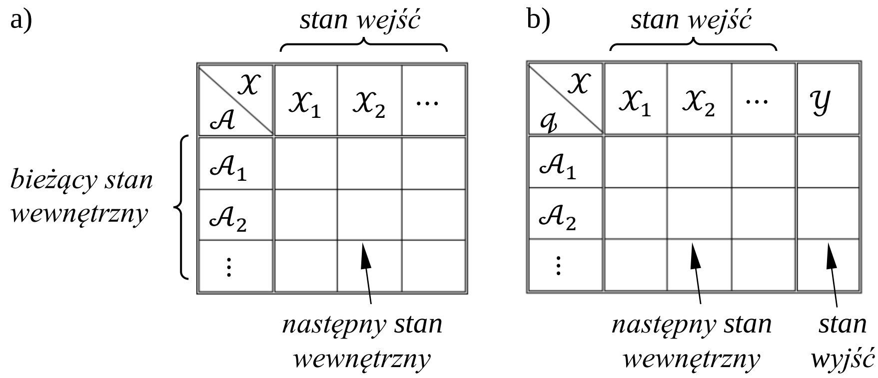
automat Moore'a można opisać równaniami

$$\begin{aligned} \mathcal{A}' &= \delta(\mathcal{A}, \mathcal{X}), \\ \mathcal{Y} &= \lambda(\mathcal{A}). \end{aligned}$$

4.2. Metody opisu układów sekwencyjnych

Działanie projektowanego lub analizowanego układu może być przedstawiane na kilka różnych sposobów. Metody opisu mogą uwzględniać jedynie zależności pomiędzy sygnałami wejściowymi i wyjściowymi, nie uwzględniając informacji o stanach wewnętrznych układu. Taki sposób opisu nazywany jest zewnętrznym i nie pozwala na bezpośrednie zdefiniowanie funkcji przejść i wyjść układu [4]. Oprócz metod zewnętrznych są również metody pełne, które w opisie uwzględniają stany wewnętrzne układu. Stosowane są np. następujące metody:

a) *opis słowny* – metoda zewnętrzna, przedstawia funkcjonowanie układu w sposób opisowy, z opisu powinny wynikać wszystkie stany wejściowe (a właściwie wszystkie możliwe sekwencje tych stanów) i odpowiadające im stany wyjściowe (lub ich sekwencje), opis komplikuje się w miarę wzrostu złożoności układu,
b) *wykres czasowy* – metoda zewnętrzna, przedstawia przebieg w czasie zmian stanów wejściowych i odpowiadających im stanów wyjściowych, opóźnienia wynikające z czasu reakcji elementów układu nie są uwzględniane, oś czasu na ogół nie odzwierciedla czasu trwania poszczególnych faz i skalowana jest taktami (takt odpowiada czasowi pomiędzy kolejnymi zmianami sygnałów wejściowych), do jednoznacznego opisu działania układu niezbędne jest zaznaczenie na wykresie wszystkich możliwych sekwencji stanów wejść i wyjść, przy dużej liczbie sygnałów lub skomplikowanym sposobie ich przetwarzania wykres ten może być mało czytelny,
c) *graf przejść* – metoda pełna, graf, którego wierzchołki odpowiadają stanom wewnętrznym układu, a krawędzie oznaczają przejścia pomiędzy stanami wymuszane określonymi stanami wejściowymi, w układach Moore'a stan wyjść zależy wyłącznie od stanu wewnętrznego, więc jest przyporządkowywany odpowiedniemu wierzchołkowi grafu (w układach Mealye'go stan wyjść zależy od stanu wewnętrznego i od stanu wejść jest więc przyporządkowany podobnie jak stan wejść krawędzi grafu),
d) *tablice przejść i wyjść* oraz *tablice przejść–wyjść* – metody pełne, przedstawiają w formie tabelarycznej informacje zawarte w grafie przejść:
   - *tablica przejść* odpowiada funkcji przejść δ układu, ma tyle wierszy, ile jest stanów wewnętrznych układu i tyle kolumn, ile różnych stanów wejściowych może wystąpić podczas pracy układu, do kratek tablicy wpisywane są stany wewnętrzne, do których przechodzi układ, który był poprzednio w stanie określonym przez wiersz tablicy pod wpływem stanu wejściowego określonego przez kolumnę tablicy (zob. rys. 4.6),
   - *tablica wyjść* odpowiada funkcji wyjść λ układu; dla układu Moore'a stan wyjść jest funkcją tylko stanu wewnętrznego, tablica wyjść ma więc tyle wierszy, ile jest stanów wewnętrznych i jedną kolumnę, w której wpisywane są wartości stanów wyjściowych,
   - *tablica przejść–wyjść* może być skonstruowana tylko dla układów Moore'a, jest tablicą przejść uzupełnioną o dodatkową kolumną zawierającą wartości stanów wyjściowych.



Rys.4.6. Tablica a) przejść b) przejść–wyjść

Wymienione metody opisu zostaną bardziej szczegółowo omówione na dołączonych przykładach.

4.2.1. Układ włącz–wyłącz I

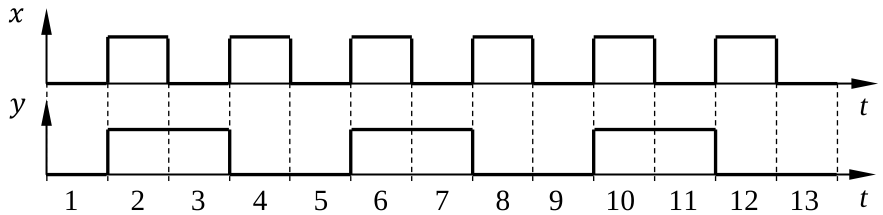
Opis słowny

Układ umożliwia cykliczne włączanie i wyłączanie urządzenia za pomocą przycisku (przycisk jest łącznikiem monostabilnym pozostającym w stanie *włączony*, tylko gdy jest wciskany przez operatora, po zwolnieniu przechodzi w stan *wyłączony*). Jeśli urządzenie nie pracuje, wciśnięcie przycisku powoduje jego włączenie, jeśli urządzenie pracuje, wciśnięcie przycisku powoduje jego wyłączenie. Zwalnianie przycisku nie powoduje zmian stanu pracy urządzenia.

Opis słowny jest zewnętrzną metodą opisu i przedstawia zależności pomiędzy sygnałami wejściowymi i sygnałami wyjściowymi. W przypadku opisywanego układu oznacza to, że przedstawia on zależności pomiędzy stanem przycisku a stanem urządzenia. Zależności te wyraźnie wskazują na sekwencyjny charakter układu: wciśnięcie przycisku powoduje albo włączenie, albo wyłączenie urządzenia – zależność między wartością sygnału wejściowego i wyjściowego nie jest więc jednoznaczna. Taki sposób działania wymaga „pamiętania” aktualnego stanu układu i uzależnienia wartości sygnału wyjściowego od stanów wejściowego i wewnętrznego. Opisywany układ włącza urządzenie, jeśli operator wciśnie przycisk, w sytuacji gdy urządzenie aktualnie nie pracuje. Stany wewnętrzne nie są uwzględniane w zewnętrznych metodach opisu są zawarte w metodach pełnych.

Wykres czasowy

Na rys. 4.7 przedstawiono 13 wybranych taktów pracy układu. Wykres czasowy, podobnie jak opis słowny, jest zewnętrzną metodą opisu i przedstawia zależności pomiędzy sygnałami wejściowymi i sygnałami wyjściowymi. W przypadku rozważanego układu sygnałem wejściowym jest stan przycisku (na wykresie oznaczony jako x), a sygnałem wyjściowym stan urządzenia (na wykresie oznaczony jako y). Zmiany stanu przycisku (przycisk wciśnięty $x = 1$, przycisk zwolniony $x = 0$) powodują zmiany stanu urządzenia (urządzenie pracuje $y = 1$, urządzenie nie pracuje $y = 0$) zgodnie z przedstawionym powyżej opisem słownym. Z wykresu czasowego wynika, że w takcie 1 przycisk jest zwolniony i urządzenie nie pracuje, w takcie 2 wciśnięcie przycisku powoduje, że urządzenie zaczyna pracować, urządzenie pracuje nawet po zwolnieniu przycisku (takt 3) itd.



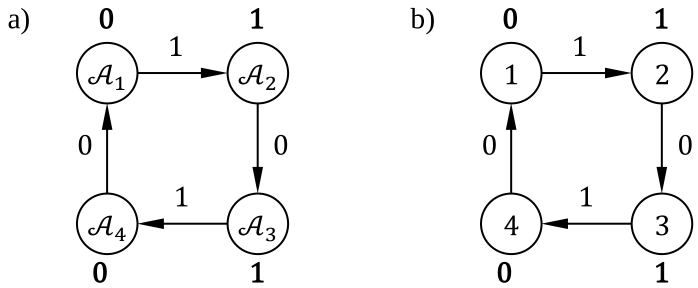
Rys.4.7. Wykres czasowy układu

Graf przejść

Graf przejść układu oparty na modelu Moore'a został przedstawiony na rys. 4.8. Zgodnie z podanym wyżej opisem graf zawiera pełne informacje o układzie: jego stanach wejściowych, wyjściowych i wewnętrznych. Wierzchołki grafu odpowiadają stanom wewnętrznym, które zostały wyodrębnione po wyszukaniu unikalnych kombinacji stanów wejściowych i wyjściowych na rys. 4.7. Takie podejście pozwoliło na zidentyfikowanie 4 stanów:

$\mathcal{A}_1$ – przycisk zwolniony i urządzenie nie pracuje ($x = 0$ i $y = 0$),
$\mathcal{A}_2$ – przycisk wciśnięty i urządzenie pracuje ($x = 1$ i $y = 1$),
$\mathcal{A}_3$ – przycisk zwolniony i urządzenie pracuje ($x = 0$ i $y = 1$),
$\mathcal{A}_4$ – przycisk wciśnięty i urządzenie nie pracuje ($x = 1$ i $y = 0$).

Na grafie do oznaczenia wierzchołków w wersji a) wykorzystane zostały wprowadzone symbole stanów, a w wersji b) przypisane im numery porządkowe. Krawędzie grafu, które łączą poszczególne węzły, odpowiadają stanom wejściowym wymuszającym przejście z jednego stanu wewnętrznego do drugiego lub utrzymującym wybrany stan. Krawędzie mają wyznaczony kierunek i są opisywane wartościami odpowiednich stanów wejściowych. W układzie Moore'a stan wyjść zależy wyłącznie od stanu wewnętrznego i jest przyporządkowywany odpowiedniemu wierzchołkowi grafu. W rozważanym układzie krawędzie grafu są więc opisywane stanem przycisku sterującego pracą urządzenia (0 – przycisk zwolniony, 1 – przycisk wciśnięty), a wierzchołki grafu stanem samego urządzenia (0 – urządzenie nie pracuje, 1 – urządzenie pracuje).



Rys.4.8. Graf przejść układu

Z grafu przejść wynika, że w przypadku, gdy układ znajduje się w stanie $\mathcal{A}_1$, urządzenie nie pracuje (wartość 0 umieszczona nad wierzchołkiem $\mathcal{A}_1$). Przejście ze stanu $\mathcal{A}_1$ do stanu $\mathcal{A}_2$ następuje w wyniku wciśnięcia przycisku (wartość 1 umieszczona nad krawędzią prowadzącą od wierzchołka $\mathcal{A}_1$ do wierzchołka $\mathcal{A}_2$), po przejściu do stanu $\mathcal{A}_2$ urządzenie zaczyna pracować (wartość 1 umieszczona nad wierzchołkiem $\mathcal{A}_2$) itd.

Tablice przejść i wyjść, tablice przejść–wyjść

Rysunek 4.9 przedstawia tablicę przejść–wyjść układu. W wersji a) bieżące i następne stany układu zostały oznaczone jako: $\mathcal{A}_1, \cdots, \mathcal{A}_4$ i $\mathcal{A}'_1, \cdots, \mathcal{A}'_4$. Z pierwszego wiersza tablicy wynika, że w przypadku gdy układ znajduje się w pierwszym stanie wewnętrznym $\mathcal{A}_1$ i przycisk jest zwolniony (0 w nagłówku kolumny), to układ pozostanie w tym stanie ($\mathcal{A}'_1$ w pierwszej komórce tego wiersza oznacza, że następnym stanem układu będzie ten sam stan wewnętrzny). Z drugiej komórki pierwszego wiersza tablicy wynika, że w przypadku gdy układ znajduje się w stanie wewnętrznym $\mathcal{A}_1$ i przycisk jest wciśnięty (1 w nagłówku kolumny), to układ przejdzie do drugiego stanu wewnętrznego ($\mathcal{A}'_2$ w drugiej komórce tego wiersza oznacza, że następnym stanem układu będzie drugi stan wewnętrzny).

a)

$\mathcal{A}$ \ $\mathcal{X}$	**0**	**1**	$\mathcal{Y}$
$\mathcal{A}_1$	$\mathcal{A}'_1$	$\mathcal{A}'_2$	0
$\mathcal{A}_2$	$\mathcal{A}'_3$	$\mathcal{A}'_2$	**1**
$\mathcal{A}_3$	$\mathcal{A}'_3$	$\mathcal{A}'_4$	**1**
$\mathcal{A}_4$	$\mathcal{A}'_1$	$\mathcal{A}'_4$	0

b)

$\mathcal{A}$ \ $\mathcal{X}$	**0**	**1**	$\mathcal{Y}$
1	①	2	0
2	3	②	**1**
3	③	4	**1**
4	1	④	0

Rys.4.9. Tablica przejść–wyjść układu

W tablicy w wersji b), w celu skrócenia zapisu, bieżące i następne stany wewnętrzne zostały opisane przypisanymi im numerami. Dodatkowo stabilne stany wewnętrze, tzn. takie, w których stan wejściowy nie wymusza przejścia do kolejnego stanu wewnętrznego, zostały otoczone kółkiem.

Ostatnia kolumna tablicy przejść–wyjść podaje wartość stanu wyjściowego, tzn. informację o stanach samego urządzenia, które nie pracuje w stanach wewnętrznych $\mathcal{A}_1$ i $\mathcal{A}_4$ i pracuje w stanach $\mathcal{A}_2$ i $\mathcal{A}_3$.

4.2.2. Układ włącz–wyłącz II

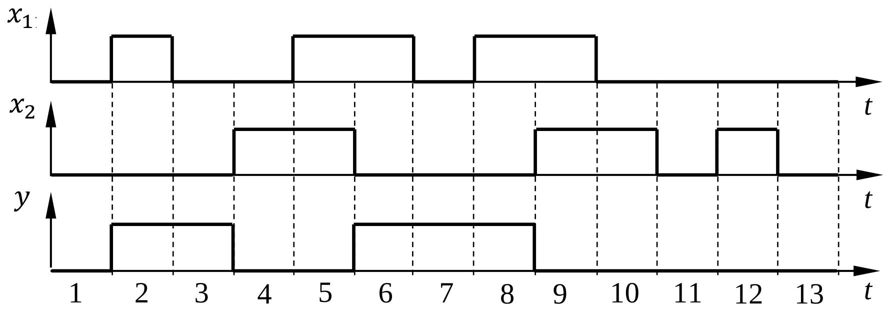
Opis słowny

Układ umożliwia włączanie i wyłączanie urządzenia za pomocą dwóch łączników bistabilnych (łączniki po naciśnięciu zmieniają trwale swój stan: przełączają się ze stanu *wyłączony* we *włączony* i ze stanu *włączony* w *wyłączony*). Włączenie pierwszego łącznika – włączającego – powoduje włączenie urządzenia. Wyłączenie tego łącznika nie powoduje jednak zatrzymania pracy urządzenia. Urządzenie może być wyłączone tylko za pomocą drugiego z łączników – wyłączającego. Łącznik wyłączający ma wyższy priorytet i w sytuacji gdy obydwa łączniki są włączone, urządzenie nie pracuje. Dodatkowo zakłada się, że nie jest możliwa jednoczesna zmiana stanu obydwu łączników – rozważany układ pracuje w *trybie podstawowym*.

Rozważany układ jest układem sekwencyjnym. Znalezienie stanu wejściowego, któremu odpowiadają różne stany wyjściowe wymaga w tym przypadku trochę dłuższej analizy niż w punkcie poprzednim. Na brak jednoznacznej zależności stanu wyjściowego od wejściowego wskazuje stan pracy urządzenia przy wyłączonych łącznikach. Urządzenie w takiej sytuacji może nie pracować, może również pracować. Urządzenie będzie pracowało, jeśli operator najpierw włączy pierwszy łącznik, a później go wyłączy. Takie działanie układu wymusza konieczność „zapamiętywania" historii oddziaływań.

Wykres czasowy

Do przedstawienia działania układu niezbędne są trzy wykresy: wykresy przedstawiające zmiany stanu obydwu łączników (x_1 i x_2) i wykres przestawiający stan pracy urządzenia (y). Z rys. 4.10 wynika, że w takcie 1 obydwa łączniki są wyłączone, a urządzenie nie pracuje, w takcie 2 włączony został pierwszy łącznik, co spowodowało uruchomienie urządzenia, wyłączenie łącznika w takcie 3 nie zatrzymało urządzenia, urządzenie zostało wyłączone dopiero po włączeniu drugiego łącznika w takcie 4 itd.



Rys.4.10. Wykres czasowy układu

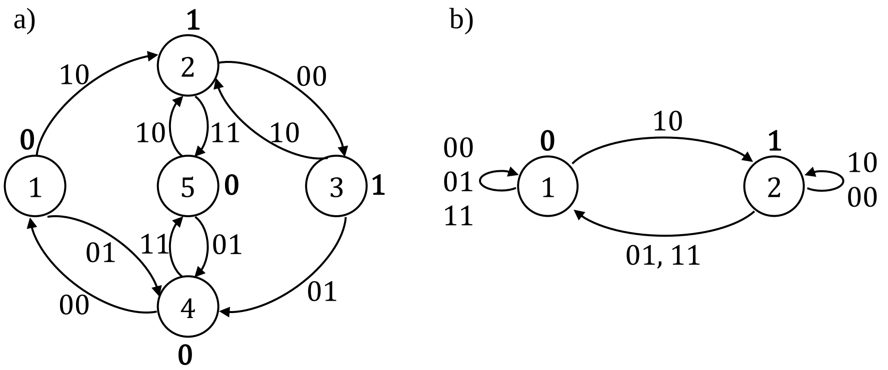
Przedstawianie działania układu na wykresie czasowym przy dwóch sygnałach wejściowych jest już trudniejsze niż w przypadku układu z jednym wejściem: trudniej jest przedstawić wszystkie możliwe sekwencje zmian sygnałów wejściowych i reakcję układu na te zmiany. Łatwo sobie wyobrazić, że przy dużej liczbie sygnałów wejściowych lub wyjściowych czy przy skomplikowanym sposobie przetwarzanie sygnałów wykres czasowy staje się mało czytelny.

Graf przejść

Stany wewnętrzne uwzględnione na rys. 4.11a zostały wyodrębnione po wyszukaniu unikalnych kombinacji stanów wejściowych i wyjściowych na rys. 4.10. Takie podejście pozwoliło na zidentyfikowanie 5 stanów:

$\mathcal{A}_1$ – obydwa łączniki wyłączone, urządzenie nie pracuje,
$\mathcal{A}_2$ – pierwszy łącznik włączony, drugi wyłączony, urządzenie pracuje,
$\mathcal{A}_3$ – obydwa łączniki wyłączone, urządzenie pracuje,
$\mathcal{A}_4$ – pierwszy łącznik wyłączony, drugi włączony, urządzenie nie pracuje,
$\mathcal{A}_5$ – obydwa łączniki włączone, urządzenie nie pracuje.

Wyznaczona w ten sposób liczba stanów wewnętrznych nie jest minimalna, działanie rozważanego układu można opisać, rysując mniejszy graf (rys. 4.11b) z dwoma wierzchołkami reprezentującymi stany: $\mathcal{A}_1$ – urządzenie nie pracuje, $\mathcal{A}_2$ – urządzenie pracuje. Zagadnienia związane z syntezą układów wraz z metodą redukcji stanów wewnętrznych omówione zostaną w punkcie 4.4.



Rys.4.11. Grafy przejść układu

Dla zwiększenia czytelności stany wewnętrzne na obydwu grafach zostały opisane przypisanymi im numerami porządkowymi. Pierwsza cyfra opisu krawędzi grafów określa stan pierwszego łącznika, druga cyfra stan łącznika drugiego. Z grafu z rys. 4.11a można odczytać, że jeśli układ znajduje się np. w stanie $\mathcal{A}_1$, to w przypadku gdy operator włączy pierwszy łącznik (przy wyłączonym drugim) nastąpi przejście do stanu $\mathcal{A}_2$ i w rezultacie urządzenie zostanie włączone (krawędź 10 prowadzi od stanu $\mathcal{A}_1$ do stanu $\mathcal{A}_2$). Jeśli w stanie $\mathcal{A}_1$ operator włączy drugi łącznik (przy wyłączonym pierwszym) to nastąpi przejście do stanu $\mathcal{A}_4$ i w rezultacie urządzenie w dalszym ciągu nie będzie pracować (krawędź 01 prowadzi od stanu $\mathcal{A}_1$ do stanu $\mathcal{A}_4$). Brak krawędzi 11 wychodzącej ze stanu $\mathcal{A}_1$, a więc brak możliwości jednoczesnego włączenia obydwu łączników, jest konsekwencją przyjętego założenia o braku możliwości jednoczesnej zmiany stanu obydwu łączników, które w stanie $\mathcal{A}_1$ są wyłączone. Podobnie z pozostałych wierzchołków grafu prowadzą tylko po dwie krawędzie – operator zawsze może zmienić stan tylko jednego łącznika.

W drugiej wersji liczba stanów układu została zredukowana do dwóch. W stanie $\mathcal{A}_1$ urządzenie jest wyłączone w stanie $\mathcal{A}_2$ urządzenie jest włączone. Włączenie urządzenia, czyli przejście ze stanu $\mathcal{A}_1$ do stanu $\mathcal{A}_2$, jest możliwe tylko w sytuacji, gdy pierwszy łącznik zostanie włączony (przy wyłączonym drugim). Wyłączenie urządzenia, czyli przejście ze stanu $\mathcal{A}_2$ do stanu $\mathcal{A}_1$, może nastąpić w dwóch przypadkach: przy wyłączonym pierwszym łączniku i włączonym drugim lub przy włączonych obydwu łącznikach (wyłączanie ma wyższy priorytet). Na grafie z rys. 4.11b oprócz krawędzi wymuszających zmiany stanów wewnętrznych zaznaczone zostały dodatkowo krawędzie utrzymujące układ w stanach włączonym i wyłączony. Jeśli urządzenie nie pracuje, to wszystkie działania operatora z wyjątkiem sytuacji, gdy pierwszy łącznik jest włączony, a drugi wyłączony utrzymują układ w tym stanie. Podobnie jeśli urządzenie pracuje, operator może włączać i wyłączać pierwszy łącznik o ile nie włączy łącznika drugiego. Wprowadzenie tych dodatkowych krawędzi nie jest konieczne, ale zwiększa precyzję opisu.

Tablica przejść–wyjść

Tablice przejść–wyjść przedstawione na rys. 4.12 odpowiadają grafom przejść rys. 4.11. Symbole „–” w pierwszej tablicy są konsekwencją założenia uniemożliwiającego jednoczesną zmianę stanu obydwu łączników. Jeżeli układ znajduje się w stanie $\mathcal{A}_1$, w którym obydwa łączniki są wyłączone, nie jest możliwe, aby łączniki zostały jednocześnie włączone. W drugiej tablicy niedozwolone przejścia nie są widoczne na skutek redukcji liczby stanów.

a)

$\mathcal{A}$ \ $\mathcal{X}$	**00**	**01**	**11**	**10**	$\mathcal{Y}$
1	1	4	–	2	0
2	3	–	5	2	1
3	3	4	–	2	1
4	1	4	5	–	0
5	–	4	5	2	0

b)

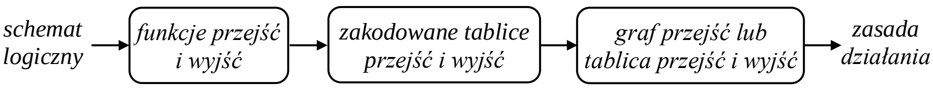
$\mathcal{A}$ \ $\mathcal{X}$	**00**	**01**	**11**	**10**	$\mathcal{Y}$
1	1	1	1	2	0
2	2	1	1	2	1

Rys.4.12. Tablice przejść–wyjść układu

4.3. Analiza układów sekwencyjnych

W ramach wprowadzenia do syntezy układów sekwencyjnych metodą Huffmana, na początek zostanie rozważona analiza kilku przykładowych układów. Głównym celem analizy jest zrozumienie zasad działania układu, w przypadku gdy znana jest jego struktura. Zrozumienie sposobu funkcjonowania układu umożliwia również wykrycie źródeł jego niewłaściwego zachowania takich jak hazard, wyścig krytyczny czy niestabilność.

Zadanie analizy sprowadza się do przejścia od struktury układu w postaci schematu logicznego do opisu w postaci grafu przejść lub tablicy przejść i wyjść (rys. 4.13).

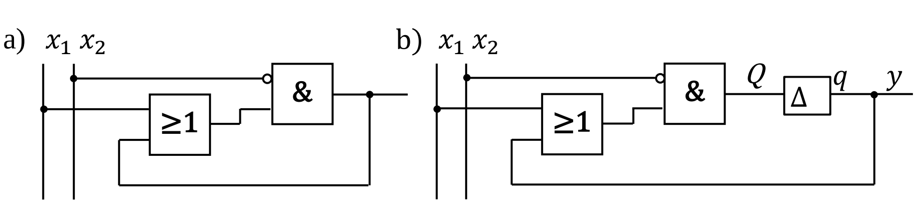


Rys.4.13. Analiza układu sekwencyjnego

Na rys. 4.13 blok *funkcje przejść i wyjść* przedstawia pierwszy etap analizy, w którym interpretowana jest struktura układu podana w postaci schematu logicznego (zob. p. 2.3) i odczytywane są funkcje przejść δ i wyjść λ. W kroku drugim funkcje te zapisywane są w postaci zakodowanych binarnie tablic przejść i wyjść nazywanych *zakodowanymi tablicami przejść* [4], [9], [23] czy krócej *siatkami (tablicami) stanów* [10]. W ostatnim kroku tworzony jest graf przejść lub tablice przejść i wyjść z symbolicznie opisanymi stanami wewnętrznymi, które pozwalają już na zrozumienie zasad działania układu. Poniżej szczegółowo omówiona została analiza kilku prostych układów.

4.3.1. Przykład 1

Schemat logiczny układu rozważanego w tym przykładzie został przedstawiony na rys. 4.14a. Porównując schemat układu z modelami układów sekwencyjnych przedstawionymi na rys. 4.5, można zauważyć, że układ ma wyłącznie blok, którego działanie jest opisywane funkcją przejść δ. Sygnał wyjściowy tego bloku jest jednocześnie sygnałem wyjściowym układu.



Rys.4.14. Schemat logiczny układu

Na rys. 4.14b w celu wprowadzenia stanu wewnętrznego w pętli sprzężenia zwrotnego został umieszczony element opóźniający Δ. Wyjście q tego elementu jest traktowane jako stan bieżący, a wejście Q jako stan następny. Takie podejście pozwala na zapisanie funkcji przejść δ układu w postaci

$$Q = \bar{x}_2(x_1 + q),$$

natomiast funkcję wyjść λ, ze względu na brak dodatkowego przetwarzania sygnału q, można zapisać jako

$$y = q.$$

Otrzymane wyrażenia pozwalają na zapisanie zakodowanej binarnie tablicy przejść–wyjść układu (rys. 4.15). W tablicy przedstawionej na rys. 4.15a zaznaczone zostały grupy zer wynikające z postaci funkcji przejść, która została zapisana w postaci koniunkcyjnej.

a)

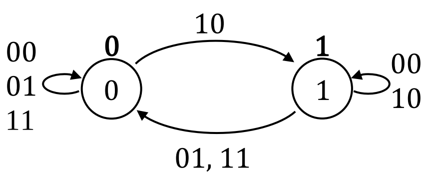
q \ x_1x_2	**00**	**01**	**11**	**10**	y
0	0	0	0	1	0
1	1	0	0	1	**1**

b)

q \ x_1x_2	**00**	**01**	**11**	**10**	y
0	0	0	0	1	0
1	1	0	0	1	**1**

Rys.4.15. Tablice przejść–wyjść układu

W tablicy z rys. 4.15b kółkiem otoczone zostały stany stabilne układu, tzn. takie dla których następny i bieżący stan wewnętrzny układu są takie same. Identyfikacja stanów stabilnych ułatwia narysowanie grafu przejść. Z przedstawionej tablicy wynika, że układ ma dwa stabilne stany wewnętrzne zakodowane jako 0 i 1. Z wiersza dla stanu 0 wynika, że układ pozostaje w tym stanie, do momentu gdy na wejściach pojawią się sygnały: $x_1 = 1$ i $x_2 = 0$. Sygnały te wymuszają przejście układu ze stanu 0 do stanu 1. Z wiersza dla stanu 1 wynika natomiast, że układ pozostaje w tym stanie do momentu wystąpienia sygnałów: $x_1 = 0$, $x_2 = 1$ lub $x_1 =1$, $x_2 = 1$. Dodatkowo z ostatniej kolumny tablicy wynika, że na wyjściu układu w stanie 0 generowany jest sygnał 0, a w stanie 1 sygnał 1. Graf przejść odpowiadający tablicy z rys. 4.15 został przedstawiony na rys. 4.16.

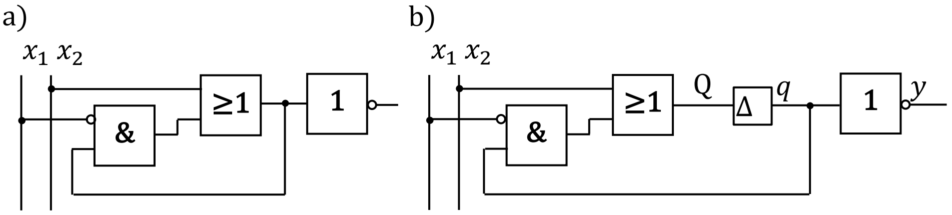


Rys.4.16. Graf przejść układu

Podsumowując analizowany układ, generuje na wyjściu sygnał o wartości 1, jeśli pierwsze wejście zostanie włączone przy wyłączonym drugim. Do wyzerowania wyjścia wystarczy włączyć drugie wejście (niezależnie od tego czy pierwsze wejście jest włączone czy nie). Przedstawiony sposób działania odpowiada zasadom pracy *przerzutnika SR z dominującym wejściem zerującym* (zob. p. 4.5). Pierwsze wejście przerzutnika jest nazywane wpisującym i oznaczane symbolem `S` (ang. set), a drugie wejście to wejście zerujące oznaczane symbolem `R` (ang. reset).

4.3.2. Przykład 2

Drugi z analizowanych układów zawiera zarówno blok, którego działanie jest opisywane funkcją przejść δ jak i blok opisywany funkcją wyjść λ. Sygnał wyjściowy układu nie odpowiada więc w tym przypadku stanowi pamięci wewnętrznej. Na rys. 4.17 przedstawiony został schemat oryginalnego układu i schemat układu uzupełniony o element opóźniający, który pozwala ustalić bieżący i następny stan wewnętrzny układu.



Rys.4.17. Schemat logiczny układu

Funkcje przejść δ i wyjść λ można opisać wyrażeniami:

$$Q = \bar{x}_1 q + x_2,$$

$$y = \bar{q}.$$

Podobnie jak w punkcie poprzednim, otrzymane wyrażenia pozwalają na zapisanie tablicy przejść–wyjść układu (rys. 4.18). W tablicy z rys. 4.18a zaznaczone zostały grupy jedynek wynikające z dysjunkcyjnej postaci funkcji przejść, w tablicy z rys. 4.18b zaznaczone zostały stany stabilne układu.

a)

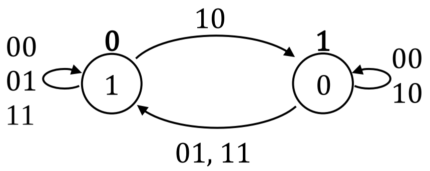
q \ x_1x_2	**00**	**01**	**11**	**10**	y
0	0	1	1	0	**1**
1	1	1	1	0	0

b)

q \ x_1x_2	**00**	**01**	**11**	**10**	y
0	0	1	1	0	**1**
1	1	1	1	0	0

Rys.4.18. Tablice przejść–wyjść układu

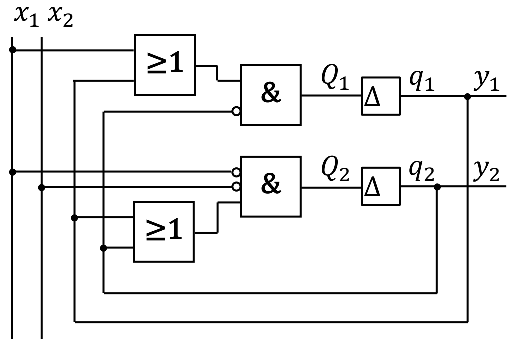
Z narysowanego grafu przejść (rys. 4.19) wynika, że analizowany układ działa w taki sam sposób jak układ z przykładu poprzedniego. Jedyna różnica pomiędzy układami polega na innym kodowaniu stanów wewnętrznych – różnica ta nie jest widoczna, ponieważ reakcje układów są w obydwu przypadkach identyczne.



Rys.4.19. Graf przejść układu

4.3.3. Przykład 3

Układy z przykładów 1 i 2 posiadały tylko jeden element pamięci – do realizacji bloku pamięci układu wykorzystywany był tylko jeden sygnał. Takie rozwiązanie dało w efekcie układy, które mogły znajdować się w dwóch stanach wewnętrznych (stan 0 i stan 1). W przykładzie 3 pamięć układu realizowana jest z pomocą dwóch sygnałów, co daje możliwość pracy w czterech stanach wewnętrznych (stany 00, 01, 10 i 11). Rysunek 4.20 przedstawia schemat układu uzupełniony o elementy opóźniające. Podobnie jak w układzie z przykładu 1, stan wyjściowy układu odpowiada w tym przypadku stanowi pamięci wewnętrznej.



Rys.4.20. Schemat logiczny układu

Ze schematu wynikają następujące funkcje przejść δ i wyjść λ:

$$Q_1 = \bar{q}_2(x_1 + q_1),$$
$$Q_2 = \bar{x}_1\bar{x}_2(q_1 + q_2),$$
$$y_1 = q_1,$$
$$y_2 = q_2.$$

Wyrażenia dla pierwszego i drugiego elementu pamięci zostały zapisane w tablicach pokazanych na rys. 4.21a i 4.21b. Po połączeniu obydwu tablic otrzymano tablicę przejść–wyjść (rys. 4.21c), w której dla zwiększenia czytelności zaznaczone zostały stany stabilne. W tablicy z rys. 4.21d stanom wewnętrznym układu przypisane zostały numery porządkowe. Stanom stabilnym 00, 01 i 10 przypisane zostały kolejne numery 1, 2 i 3. Niestabilny stan wewnętrzny 11 otrzymał numer 4.

a)

q_1q_2 \ x_1x_2	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	0	0	0	0
10	1	1	1	1

Q_1

b)

q_1q_2 \ x_1x_2	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	0	0	0
10	1	0	0	0

Q_2

c)

q_1q_2 \ x_1x_2	00	01	11	10	y_1y_2
00	00	00	10	10	00
01	01	00	00	00	01
11	01	00	00	00	11
10	11	10	10	10	10

Q_1Q_2

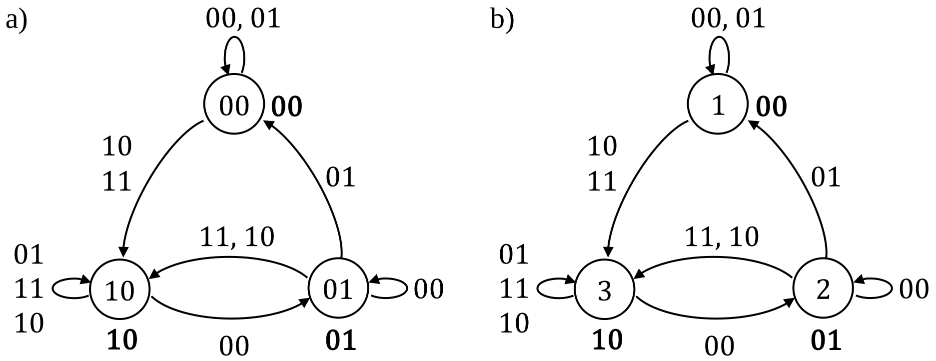
d)

q_1q_2 \ x_1x_2	00	01	11	10	y_1y_2
1	1	1	3	3	00
2	2	1	1	1	01
4	2	1	1	1	11
3	4	3	3	3	10

Q_1Q_2

Rys.4.21. Tablice przejść–wyjść układu

Z analizy pierwszego wiersza tablic z rys. 4.21c czy 4.21d wynika, że jeśli układ znajduje się w pierwszym stanie (stan oznaczony jako 00 lub 1), to pozostaje w nim dla stanów wejść 00 i 01 (w kolumnach 00 i 01 znajduje się ten sam stan). W kolumnach 11 i 10 pierwszego wiersza wpisany jest trzeci stan (stan oznaczony jako 10 lub 3), co oznacza, że stany wejściowe 11 i 10 powodują przejście układu ze stanu pierwszego do trzeciego. Z analizy drugiego wiersza wynika, że jeśli układ znajduje się w drugim stanie (stan oznaczony jako 01 lub 2) to pozostaje w nim dla stanu wejściowego 00. Pojawienie się stanu wejściowego 01 wymusza powrót układu do stanu pierwszego (w kolumnie 01 jest wpisany stan pierwszy). Stany wejściowe 11 i 10 również wymuszają przejście do stanu pierwszego (w kolumnach 11 i 10 jest wpisany stan pierwszy), ale ze względu na niestabilność tego stanu (w pierwszym wierszu w kolumnach 11 i 10 nie ma stanu stabilnego – wpisany jest stan trzeci) następuje przejście do stanu trzeciego. W wierszu reprezentującym trzeci stan układu (10 lub 3) kolumny 11 i 10 zawierają już stan stabilny, więc ostatecznie jeśli układ znajduje się w stanie drugim, to stany wejściowe 11 i 10 powodują przejście układu do stanu trzeciego. W stanie trzecim układ pozostaje dla stanów wejść 01, 11 i 10, natomiast stan wejściowy 00 wymusza przejście do stanu drugiego (przejście jest realizowane w dwóch etapach: najpierw układ przechodzi do niestabilnego stanu czwartego, później do stabilnego stanu drugiego). Przejścia przez niestabilne stany pośrednie nazywane są *przejściami cyklicznymi*. Rozważany układ ma więc trzy przejścia tego typu: dla wejść 11 i 10 w stanie drugim i dla wejścia 00 w stanie trzecim. Omówiony sposób działania układu został przedstawiony na rys. 4.22 (*uwaga*: na grafach przejść zaznacza się wyłącznie stany stabilne).

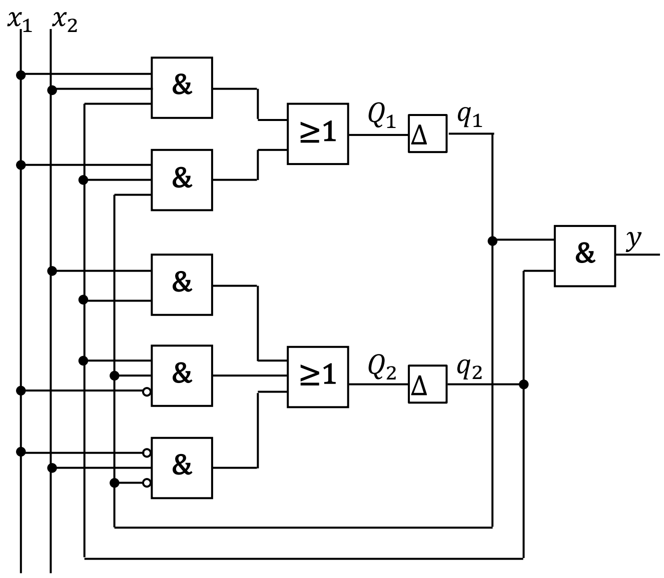


Rys.4.22. Graf przejść układu

Opierając się na grafie przejść, znacznie prościej jest opisać zasadę działania układu. Ze stanu pierwszego układ może przejść wyłącznie do stanu trzeciego. Przejście takie umożliwia podanie sygnału 1 na pierwsze wejście (niezależnie od stanu drugiego wejścia). Ze stanu trzeciego układ może przejść wyłącznie do stanu drugiego. Przejście takie umożliwia wyzerowanie obydwu sygnałów wejściowych. Podanie sygnału 1 na pierwsze wejście, w sytuacji gdy układ znajduje się w stanie drugim powoduje przejście do stanu trzeciego, wyzerowanie pierwszego sygnału i ustawienie drugiego sygnału powoduje przejście układu ze stanu drugiego do pierwszego.

4.3.4. Przykład 4

Na koniec przeanalizowany zostanie układ z dwoma elementami pamięci, w którym stan wyjściowy różni się od stanu wewnętrznego. Schemat układu uzupełniony o elementy opóźniające został przedstawiony na rys. 4.23.



Rys.4.23. Schemat logiczny układu

Ze schematu wynikają następujące funkcje przejść δ i wyjść λ:

$$Q_1 = x_1x_2q_2 + x_1q_1q_2,$$

$$Q_2 = x_2q_2 + \bar{x}_1q_1q_2 + \bar{x}_1x_2\bar{q}_1,$$

$$y = q_1q_2.$$

Funkcje te zostały zapisane w tablicach pokazanych na rys. 4.24. Rysunek 4.25 pokazuje tablice przejść–wyjść powstałe po połączeniu tablic z rys. 4.24 oraz po przypisaniu stanom wewnętrznym numerów porządkowych.

a)

q_1q_2 \ x_1x_2	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	1	1
10	0	0	0	0

Q_1

b)

q_1q_2 \ x_1x_2	00	01	11	10
00	0	1	0	0
01	0	1	1	0
11	1	1	1	0
10	0	0	0	0

Q_2

c)

q_1q_2	y
00	0
01	0
11	1
10	0

Rys.4.24. Tablice odpowiadające funkcjom przejść i wyjść

a)

q_1q_2 \ x_1x_2	00	01	11	10	y
00	00	01	00	00	**0**
01	00	01	11	00	**0**
11	01	01	11	10	**1**
10	00	00	00	00	**0**

Q_1Q_2

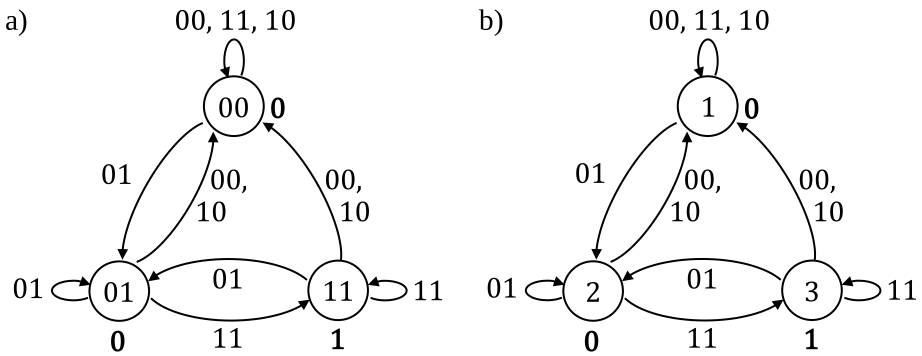
b)

q_1q_2 \ x_1x_2	00	01	11	10	y
1	1	2	1	1	**0**
2	1	2	3	1	**0**
3	2	2	3	4	**1**
4	1	1	1	1	**0**

Q_1Q_2

Rys.4.25. Tablice przejść–wyjść układu

Tablicom z rys. 4.25 odpowiadają grafy przejść pokazane na rys. 4.26.



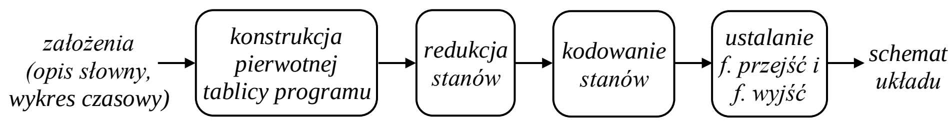
Rys.4.26. Graf przejść układu

Z analizy grafu przejść wynika, że wyjście układu jest ustawiane, jeżeli po stanie wejściowym 01 nastąpi stan 11, we wszystkich pozostałych sytuacjach wyjście jest wyzerowane.

4.4. Synteza układów sekwencyjnych

Synteza to proces odwrotny do analizy, prowadzi od założeń definiujących sposób działania układu do jego projektu. Pierwsze podejścia do syntezy układów sekwencyjnych były formułowane w latach 50. ubiegłego wieku przez Huffmana, Ungera, McCluskeya, Eichelbergera czy Mullera. W latach 80. pojawiły się nowe koncepcje takie jak możliwość wykorzystania metod opartych na sieciach Petriego, czy wykorzystanie techniki samotaktowania z generowanym lokalnie sygnałem zegarowym (*burst – mode machines*).

Wprowadzony w poprzednim punkcie sposób analizy układów sekwencyjnych stanowił wstęp do syntezy układów metodą Huffmana. Zastosowanie metody ogranicza się do układów o maksymalnie trzech wejściach ale pozwala na projektowanie nawet złożonych algorytmów działania. Rysunek 4.27 przedstawia kolejne kroki metody.



Rys.4.27. Synteza metodą Huffmana

W pierwszym kroku, na podstawie założeń opisujących sposób działania układu, sporządzana jest tzw. *pierwotna tablica programu*. Tablica ta jest właściwie tablicą przejść–wyjść, której każdy wiersz zawiera dokładnie jeden stabilny stan wewnętrzny. Pierwotna liczba stanów wewnętrznych układu przyjęta na podstawie pierwotnej tablicy programu na ogół jest większa od liczby stanów niezbędnych do realizacji określonego zadania. W drugim kroku metody przeprowadzana jest redukcja liczby stanów, w wyniku której powstaje tzw. *zredukowana tablica programu* (tablica powstaje w wyniku połączenia wybranych wierszy tablicy pierwotnej). W kolejnym kroku stanom wewnętrznym przypisuje się kody binarne – w rezultacie powstaje *zakodowana tablica programu*. Na podstawie tej tablicy można już zapisać funkcje przejść i wyjść układu, można również wykorzystać tablicę do określenia funkcji wzbudzeń przerzutników, które mogą być wykorzystane do realizacji pamięci układu.

Przebieg metody zostanie krótko omówiony na przedstawionych dalej przykładach. Dalsze szczegóły i przykłady można znaleźć np. w podręcznikach [4], [10], [23].

4.4.1. Układ włącz–wyłącz II

Na początek przeprowadzona zostanie synteza układu pozwalającego na włączanie i wyłączanie urządzenia za pomocą dwóch łączników bistabilnych: łącznika włączającego i łącznika wyłączającego. Układ ten został szczegółowo opisany w punkcie 4.2.2. Do opisu układu wykorzystano dwa zbiory stanów. Pierwszy zbiór składający się z 5 stanów został wyodrębniony na podstawie wykresu czasowego (zob. rys. 4.10) po wyszukaniu unikalnych kombinacji stanów wejściowych i wyjściowych. Sposób konstrukcji drugiego zbioru nie został wyjaśniony – jest on wynikiem redukcji stanów, która zostanie omówiona przy okazji syntezy tego układu.

Konstrukcja pierwotnej tablicy programu

Zgodnie ze schematem przedstawionym na rys. 4.27 na podstawie założeń opisujących sposób działania układu należy na początek przygotować *pierwotną tablicę programu*. Tablica programu jest tablicą przejść–wyjść z wierszami zawierającymi po jednym stabilnym stanie wewnętrznym. Wybór początkowego zbioru stanów na podstawie kombinacji wejściowo–wyjściowych gwarantuje, że każdy wybrany w ten sposób stan jest stabilny dla jednego ustalonego stanu wejściowego (wybór taki może okazać się niewystarczający do skonstruowania grafu przejść i w konsekwencji pierwotnej tablicy programu – zostanie to omówione w ostatnim przykładzie tego punktu). Opis układu w postaci tablicy przejść–wyjść przedstawionej na rys. 4.12a odpowiada więc pierwotnej tablicy programu niezbędnej do przeprowadzenia kolejnych etapów syntezy tego układu.

Redukcja stanów

Redukcja stanów układu prowadzi w efekcie do łączenia wierszy pierwotnej tablicy programu. Algorytm minimalizacji pozwala na redukcję *stanów równoważnych* i *stanów zgodnych*.

Dwa stany wewnętrzne są nazywane *równoważnymi*, gdy są reakcją na te same stany wejściowe (znajdują się w tej samej kolumnie tablicy), mają jednakowe lub *niesprzeczne* stany wyjściowe oraz wszystkie możliwe przejścia z tych stanów prowadzą do takich samych stanów lub stanów równoważnych. Pojęcie *zgodności* dotyczy stanów zdefiniowanych dla różnych stanów wejściowych (znajdujących się w różnych kolumnach tablicy).

Dwa stany wewnętrzne są nazywane *zgodnymi*, w przypadku gdy mają jednakowe lub niesprzeczne stany wyjściowe oraz wszystkie możliwe przejścia z tych stanów prowadzą do takich samych stanów lub stanów zgodnych. *Niesprzeczność* pozwala na łączenie stanów określonych z nieokreślonymi tzn. oznaczonych „–". Formalne definicje równoważności, zgodności i niesprzeczności stanów można znaleźć np. w [4], [23].

Redukcja stanów prowadzi do połączenia odpowiednich wierszy tablicy programu. W połączonym wierszu wpisuje się:

- wartość, którą zawierały odpowiadające sobie komórki tablicy lub wartość z jednej z komórek, w przypadku gdy druga zawierała „–",
- kreskę, jeżeli łączone komórki zawierały „–",
- kółko, jeżeli łączone komórki zawierały tę samą wartość i dodatkowo w jednej z nich wartość ta oznaczała stabilny stan wewnętrzny.

Podczas łączenia wierszy pierwotnej tablicy programu pomocny jest tzw. *wykres redukcyjny*. Na wykresie tym, na obwodzie koła, umieszcza się wszystkie stany wewnętrzne układu. Jeśli dwa stany mogą zostać połączone, to na wykresie łączone są odcinkami. Wykres ułatwia wybór najlepszej metody redukcji stanów i pozwala na zweryfikowanie możliwości redukcji w przypadku większej liczby stanów (kilka stanów może być zredukowanych do jednego, jeżeli każda para stanów jest połączona).

Proces redukcji stanów rozważanego układu został pokazany na rys. 4.28 (rys. 4.28a przedstawia pierwotną tablicę programu pokazaną wcześniej na rys. 4.12a). Analizując zawartość tablicy z rys. 4.28a, można zauważyć, że mogą zostać połączone wiersze:

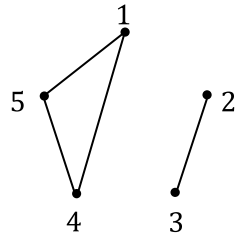
- pierwszy i czwarty (w odpowiadających sobie komórkach znajdują się kolejno: ① i 1, 4 i ④, – i 5, 2 i –, 0 i 0),
- pierwszy i piąty (① i –, 4 i 4, – i ⑤, 2 i 2, 0 i 0),
- drugi i trzeci (3 i ③, – i 4, 5 i –, ② i 2, 1 i 1),
- czwarty i piąty (1 i –, ④ i 4, 5 i ⑤, – i 2, 0 i 0).

Wykres redukcyjny przedstawiony na rys. 4.28b pokazuje znalezione pary stanów zgodnych. Z wykresu wynika, że można połączyć stany: pierwszy, trzeci i piąty (połączone są stany 1 i 3, 1 i 5 oraz 3 i 5 więc każda para stanów jest połączona) oraz stany: drugi i czwarty. Zredukowana tablica programu będąca wynikiem takiego łączenia została pokazana na rys. 4.28c. Tablica ta odpowiada tablicy przedstawionej na rys. 4.12b przygotowanej w punkcie 4.2.2 dla drugiego, minimalnego zbioru stanów.

a)

$\mathcal{A}$ \ $\mathcal{X}$	00	01	11	10	$\mathcal{Y}$
1	1	4	–	2	0
2	3	–	5	2	1
3	3	4	–	2	1
4	1	4	5	–	0
5	–	4	5	2	0

b)



c)

$\mathcal{A}$ \ $\mathcal{X}$	00	01	11	10	$\mathcal{Y}$
1,4,5	1	4	5	2	0
2,3	3	4	5	2	1

Rys.4.28. Redukcja stanów a) pierwotna tablica programu, b) wykres redukcyjny, c) zredukowana tablica programu

Kodowanie stanów

Po wykonaniu redukcji należy otrzymanym stanom wewnętrznym przypisać kody binarne. Przepisanie zredukowanej tablicy programu z użyciem przyjętego sposobu kodowania stanów daje w rezultacie *zakodowaną tablicę programu*. W zredukowanej tablicy programu z rys. 4.28c są tylko dwa stany wewnętrzne oznaczone jako „1,4,5" i „2,3". Stanom tym należy przypisać kody binarne, co oznacza, że jeden ze stanów powinien zostać zakodowany jako 0, a drugi jako 1. Metoda kodowania narzuca pewne wymagania ograniczające sposób przypisywania kodów, ale nie mają one zastosowania w tym przypadku (ograniczenia te zostaną omówione w następnym przykładzie dla układu o większej liczbie stanów wewnętrznych). W rozważanym przypadku stany mogą zostać zakodowane dowolnie. Wydaje się, że bardziej rozsądne jest przypisanie stanowi „1,4,5" kodu 0, a stanowi „2,3" kodu 1. Taki sposób kodowania eliminuje konieczność dodatkowego przekształcania sygnału stanu wewnętrznego w sygnał wyjściowy. Rysunek 4.29 pokazuje tablicę programu po zakodowaniu. W wersji a) stanowi „1,4,5" przypisano kod 0, a stanowi „2,3" kod 1, w wersji b) stany zostały zakodowane odwrotnie.

a)

q \ x_1x_2	00	01	11	10	y
0	0	0	0	1	0
1	1	0	0	1	1

b)

q \ x_1x_2	00	01	11	10	y
1	1	1	1	0	0
0	0	1	1	0	1

Rys.4.29. Zakodowana tablica programu

Ustalanie funkcji przejść i wyjść

Po zakodowaniu zredukowanych tablic programu można określić funkcje przejść i wyjść układu. W przypadku zakodowanej tablicy programu przedstawionej na rys. 4.29a należy określić jedynie funkcję przejść. Funkcja wyjść jest funkcją identycznościową, ponieważ stan wyjściowy odpowiada stanowi wewnętrznemu. Ostatecznie, zapisując funkcję przejść w postaci koniunkcyjnej, układ dla kodowania w wersji a) można opisać wyrażeniami:

$$Q = \bar{x}_2(x_1 + q),$$
$$y = q.$$

W wersji b) stan wyjściowy odpowiada zanegowanemu sygnałowi stanu wewnętrznego, zapisując funkcję przejść w postaci dysjunkcyjnej otrzymuje się wyrażenia:

$$Q = \bar{x}_1 q + x_2,$$
$$y = \bar{q}.$$

Wyznaczone w ten sposób funkcje przejść i wyjść pozwalają już na realizację układu. Warto zauważyć, że układ ten był analizowany w dwóch pierwszych przykładach w punkcie 4.3. Schematy logiczne układu w wersji a) i b) można znaleźć na rys. 4.14 i 4.17.

4.4.2. Układ włącz–wyłącz I

Układ pozwala na cykliczne włączanie i wyłączanie urządzenia za pomocą przycisku (łącznika monostabilnego). Szczegółowy opis działania wraz z grafem przejść zbudowanym dla 4 stanów wyodrębnionych na podstawie wykresu czasowego można znaleźć w punkcie 4.2.1.

Konstrukcja pierwotnej tablicy programu

Podobnie jak w przykładzie poprzednim, opis układu w postaci tablicy przejść przygotowanej w punkcie 4.2.1 (zob. rys. 4.9) odpowiada pierwotnej tablicy programu niezbędnej do przeprowadzenia kolejnych etapów syntezy tego układu.

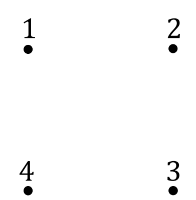
Redukcja stanów

Proces redukcji stanów rozważanego układu został zilustrowany na rys. 4.30 (rys. 4.30a przedstawia pierwotną tablicę programu pokazaną wcześniej na rys. 4.9). Analiza zawartości tablicy programu prowadzi do wniosku, że nie można zmniejszyć liczby stanów wewnętrznych – każde dwa dowolnie wybrane wiersze tablicy zawierają stany sprzeczne. Na wykresie redukcyjnym nie można więc zaznaczyć ani jednej pary stanów zgodnych.

a)

$\mathcal{A}$ \ $\mathcal{X}$	**0**	**1**	$\mathcal{Y}$
1	1	2	0
2	3	2	**1**
3	3	4	**1**
4	1	4	0

b)

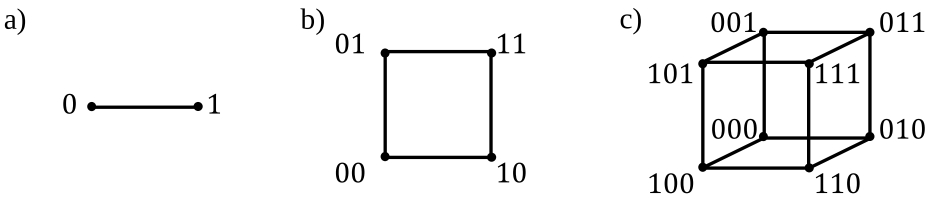


Rys.4.30. Redukcja stanów a) pierwotna tablica programu, b) wykres redukcyjny

Kodowanie stanów

Układ włącz–wyłącz sterowany łącznikami bistabilnymi miał tylko dwa stany wewnętrzne, więc jeden stan otrzymał kod 0, a drugi kod 1. W analizowanym teraz przykładzie układ ma cztery stany wewnętrzne (tablica z rys. 4.30 ma cztery wiersze), dlatego do ich zakodowania należy wykorzystać ciągi zerojedynkowe. W ogólnym przypadku, jeżeli liczba wierszy zredukowanej tablicy programu mieści się w zakresie pomiędzy 2^{p-1} i 2^p, to blok pamięci układu można zrealizować, wykorzystując p elementów pamięci, kodując stany wewnętrzne z pomocą p-elementowych ciągów zerojedynkowych.

Czasami, w celu uniknięcia niekorzystnych zjawisk, takich jak wyścigi, które zostaną omówione dalej, konieczne jest rozbudowanie układu tzn. użycie większej liczby elementów pamięci i w konsekwencji wykorzystanie kodów o długości większej niż p. Problem wyścigów eliminuje zasada kodowania wymuszająca przypisywanie kodów w taki sposób, aby niezbędne przejścia pomiędzy wierszami tablicy programu można było zrealizować przy zmianie stanu tylko jednego elementu pamięci. Stosowanie powyższej zasady sprowadza się do przypisywania kolejnym stanom wewnętrznym sąsiednich kodów Graya (zob. p. 2.5.2). Taki sposób kodowania, ze względu na graficzną interpretację, nazywany jest metodą hipersześcianów. Rysunek 4.31 pokazuje hipersześciany w przestrzeni jedno-, dwu- i trójwymiarowej, tzn. odcinek, kwadrat i sześcian, wykorzystane do kodowania kodami jedno-, dwu- i trzyelementowymi.



Rys.4.31. Kodowanie metodą hipersześcianów

Poprawne zakodowanie zredukowanej tablicy programu sprowadza się do takiego rozmieszczenia odpowiadającego jej grafu przejść, aby krawędzie grafu pokrywały się wyłącznie z krawędziami hipersześcianu.

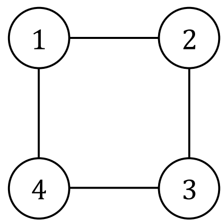
Dla ułatwienia kodowania rysowany jest również *wykres przejść*, który jest uproszczonym grafem przejść utworzonym ze zredukowanej tablicy programu z pominiętymi oznaczeniami kierunków na krawędziach oraz pominiętymi opisami stanów wejściowych i wyjściowych. Poprawność kodowania zapewnia przypisanie sąsiednim wierzchołkom grafu sąsiednich kodów Graya.

Rozważany w tym przykładzie układ ma cztery stany wewnętrzne, a jego wykres przejść (zob. rys. 4.32) ma postać kwadratu. Do zakodowania stanów układu wystarczające są więc dwuelementowe ciągi zerojedynkowe. Kodowanie można przeprowadzić na osiem różnych sposobów, dwa z nich dla stanu pierwszego zakodowanego jako 00, zostały pokazane na rys. 4.32c i 4.32d. Rysunek 4.32e przedstawia zakodowaną tablicę programu, w której przyjęto kodowanie przedstawione na rys. 4.32d. Kodowanie w tym przypadku nie sprawiało żadnych problemów – numery porządkowe stanów zostały zastąpione kodami. W kolejnym przykładzie pojawi się konieczność zakodowania przejścia cyklicznego (tj. przejścia przez niestabilny stan pośredni), które będzie wymagało przekierowania bezpośredniego przejścia pomiędzy stanami przez stan pośredni.

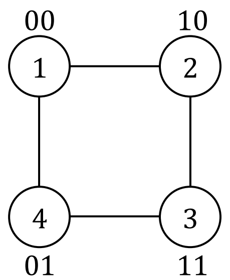
a)

$\mathcal{A}$ \ $\mathcal{X}$	**0**	**1**	$\mathcal{Y}$
1	1	2	0
2	3	2	**1**
3	3	4	**1**
4	1	4	0

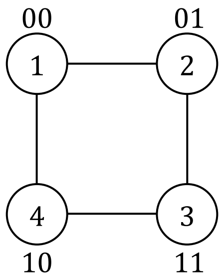
b)



c)



d)



e)

q_1q_2 \ x	**0**	**1**	y
00	00	01	0
01	11	01	**1**
11	11	10	**1**
10	00	10	0

Rys.4.32. Kodowanie stanów a) zredukowana tablica programu, b) wykres przejść, c) i d) przykładowe zbiory kodów, e) zakodowana tablica programu

Ustalanie funkcji przejść i wyjść

W celu określenia funkcji przejść i wyjść układu wygodnie jest zakodowaną tablicę programu podzielić na tablice dla elementów pamięci i na tablicę wyjść.

a)

q_1q_2 \ x	**0**	**1**
00	0	0
01	1	0
11	1	1
10	0	1

Q_1

b)

q_1q_2 \ x	**0**	**1**
00	0	1
01	1	1
11	1	0
10	0	0

Q_2

c)

q_1q_2	y
00	0
01	1
11	1
10	0

Rys.4.33. Zakodowana tablica programu po rozdzieleniu na trzy tablice

Funkcje przejść i wyjść układu wyznaczone na podstawie grup jedynek zaznaczonych na rys. 4.33 można zapisać w postaci:

$$Q_1 = \bar{x}\, q_2 + q_1q_2 + x\, q_1,$$

$$Q_2 = \bar{x}\, q_2 + \bar{q}_1q_2 + x\, \bar{q}_1,$$

$$y = q_2.$$

4.4.3. Układ włącz–wyłącz III

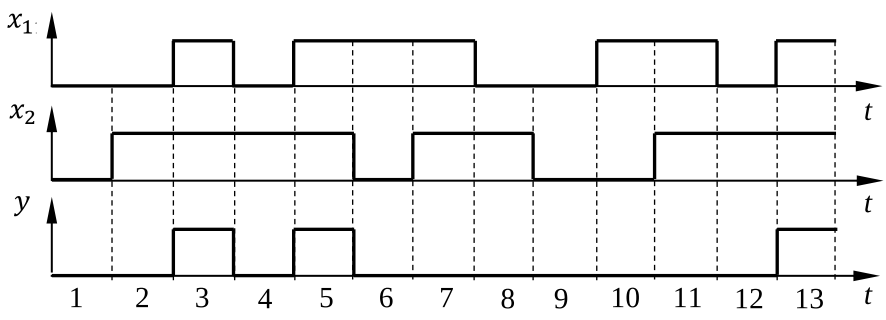
Układ, który zostanie zaprojektowany w tym punkcie nie został opisany w punkcie 4.2. Syntezę należy rozpocząć od sformułowania problemu, tzn. od opisu układu, który należy zaprojektować.

Opis słowny

Układ umożliwia włączanie i wyłączanie urządzenia za pomocą dwóch łączników bistabilnych. Włączenie urządzenia następuje tylko wtedy, gdy zostanie włączony najpierw łącznik drugi, a później pierwszy. Zmiana stanu dowolnego łącznika powoduje wyłączenie urządzenia. Dodatkowo zakłada się, że układ pracuje w *trybie podstawowym*, tzn. nie jest możliwa jednoczesna zmiana stanu obydwu łączników.

Wykres czasowy

Do przedstawienia działania układu niezbędne są trzy wykresy: wykresy przedstawiające zmiany stanu obydwu łączników (x_1 i x_2) oraz wykres przestawiający stan pracy urządzenia (y). Pierwsze trzy takty wykresu czasowego (rys. 4.34) pokazują procedurę włączenia urządzenia: najpierw włączany jest łącznik drugi, a później pierwszy. Pierwszy łącznik został wyłączony w takcie 4, a włączony w takcie 5, co pozwoliło na włączenie urządzenia, ponieważ drugi łącznik był włączony itd.



Rys.4.34. Wykres czasowy układu

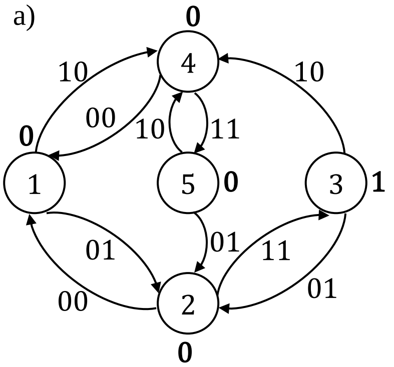
Graf przejść i tablica przejść–wyjść

Analiza wykresu czasowego przedstawionego na rys. 4.34 pozwala na przyporządkowanie 5 unikalnych stanów odcinkom tego wykresu:

$\mathcal{A}_1$ – obydwa łączniki wyłączone, urządzenie nie pracuje,
$\mathcal{A}_2$ – pierwszy łącznik wyłączony, drugi włączony, urządzenie nie pracuje,
$\mathcal{A}_3$ – obydwa łączniki wyłączone, urządzenie pracuje,
$\mathcal{A}_4$ – pierwszy łącznik włączony, drugi wyłączony, urządzenie nie pracuje,
$\mathcal{A}_5$ – obydwa łączniki włączone, urządzenie nie pracuje.

Wyznaczone w ten sposób stany pozwalają na narysowanie grafu przejść układu (rys. 4.35a) i zapisanie tablicy przejść–wyjść (rys. 4.35b), która odpowiada pierwotnej tablicy programu.

a)



b)

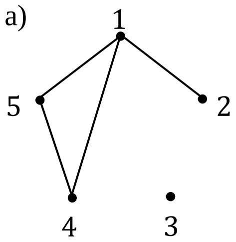
$\mathcal{A}$ \ $\mathcal{X}$	**00**	**01**	**11**	**10**	$\mathcal{Y}$
1	1	2	–	4	0
2	1	2	3	–	0
3	–	2	3	4	**1**
4	1	–	5	4	0
5	–	2	5	4	0

Rys.4.35. Graf przejść i tablica przejść–wyjść układu

Redukcja stanów

Rysunek 4.36 pokazuje przebieg redukcji stanów układu.

a)



b)

Redukcja	
wersja 1	**wersja 2**
1, 4, 5 2 3	1, 2 4, 5 3

c)

$\mathcal{A}$ \ $\mathcal{X}$	**00**	**01**	**11**	**10**	$\mathcal{Y}$
1,4,5	1	2	5	4	0
2	1	2	3	–	0
3	–	2	3	4	**1**

Rys.4.36. Redukcja stanów a) wykres redukcyjny, b) warianty redukcji, c) zredukowana tablica programu

Z wykresu redukcyjnego wynika, że stany 1, 2, 4 i 5 są połączone między sobą, ale nie można ich zredukować do jednego stanu, ponieważ nie są połączone w systemie „każdy z każdym”. Dwa różne warianty redukcji stanów zostały pokazane na rys. 4.36b. Rysunek 4.36c pokazuje zredukowaną tablicę programu uzyskaną dla pierwszej wersji redukcji.

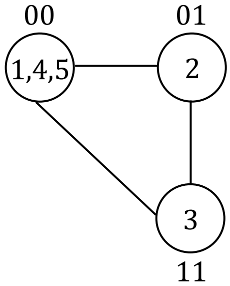
Kodowanie stanów

W analizowanym przykładzie układ może znajdować się w trzech stanach wewnętrznych, które do zakodowania, podobnie jak w przykładzie poprzednim, wymagają wykorzystania dwuelementowych ciągów zerojedynkowych.

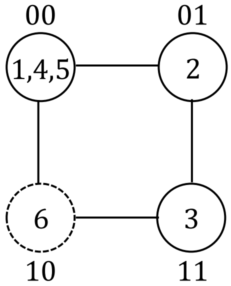
a)

$\mathcal{X}$ / $\mathcal{A}$	**00**	**01**	**11**	**10**	$\mathcal{Y}$
1,4,5	1	2	5	4	0
2	1	2	3	–	0
3	–	2	3	4	**1**

b)



c)



d)

$\mathcal{X}$ / $\mathcal{A}$	**00**	**01**	**11**	**10**	$\mathcal{Y}$
1,4,5	1	2	5	4	0
2	1	2	3	–	0
3	–	2	3	6	**1**
6	–	–	–	4	–

e)

x_1x_2 / q_1q_2	**00**	**01**	**11**	**10**	y
00	00	01	00	00	0
01	00	01	11	–	0
11	–	01	11	10	**1**
10	–	–	–	00	–

Rys.4.37. Kodowanie stanów a) zredukowana tablica programu, b) i c) wykres przejść, d) uzupełniona tablica programu, e) zakodowana tablica programu

Na rys. 4.37a została powtórzona zredukowana tablica programu z rys. 4.36c. Wykres przejść (rys. 4.37b) pokazuje wszystkie przejścia pomiędzy stanami układu wraz z przykładowym kodowaniem. Należy jednak zauważyć, że zaproponowany sposób kodowania stanów nie jest prawidłowy – poprawność kodowania zapewnia przypisanie sąsiednim wierzchołkom grafu sąsiednich kodów Graya, a wierzchołkom „3” i „1,4,5” przypisane zostały kody 00 i 11, które nie są kodami sąsiednimi.

Rysunek 4.37c pokazuje poprawiony wykres przejść z dodatkowym stanem oznaczonym jako „6”, któremu przypisany został kod 10. Przejście pomiędzy stanami „3” i „1,4,5” zostało zrealizowane jako przejście cykliczne z wykorzystaniem niestabilnego stanu pośredniego „6”. Uzupełniony wykres przejść ze zmodyfikowanym kodowaniem stanów jest już poprawny.

Wprowadzenie dodatkowego stanu wymaga również odpowiedniego uzupełnienia tablicy programu (rys. 4.37d). Oprócz wprowadzenia dodatkowego wiersza odpowiadającego nowemu stanowi, poprawione zostało przejście ze stanu „3” do stanu „4”. Tablica nie wymagała innych poprawek, ponieważ układ nie pozwalał na realizację przejścia w drugą stronę. Warto zwrócić uwagę na wartość stanu wyjściowego dla stanu wewnętrznego „6”. Stan ten jest stanem przejściowym pomiędzy stanem „3”, dla którego wyjście układu powinno być ustawione, a stanem „4”, dla którego wyjście powinno być wyzerowane. Poprawne byłoby więc przyjęcie dla tego stanu zarówno wartości wyjściowej 1, jak i wartości 0. Ze względu na to, że obydwa rozwiązania są równie dobre, w tablicy programu wpisana została wartość „–” (gdyby „3” i „4” miały taki sam stan wyjściowy, to również w stanie przejściowym, w celu uniknięcia chwilowej zmiany wyjścia, należałoby przyjąć taką samą wartość stanu wyjściowego).

Ostatecznie zakodowana tablica programu pokazana na rys. 4.37e powstała w wyniku zastąpienia symboli stanów ich kodami.

Ustalanie funkcji przejść i wyjść

Po rozdzieleniu zakodowanej tablicy programu (rys. 4.38) otrzymano funkcje przejść i wyjść układu w postaci:

$$Q_1 = x_1 q_2,$$
$$Q_2 = x_2 q_2 + \bar{x}_1 x_2,$$
$$y = q_1.$$

a)

q_1q_2 \ x_1x_2	00	01	11	10
00	0	0	0	0
01	0	0	1	–
11	–	0	1	1
10	–	–	–	0

Q_1

b)

q_1q_2 \ x_1x_2	00	01	11	10
00	0	1	0	0
01	0	1	1	–
11	–	1	1	0
10	–	–	–	0

Q_2

c)

q_1q_2	y
00	0
01	0
11	1
10	–

Rys.4.38. Zakodowana tablica programu po rozdzieleniu na trzy tablice

Układ włącz–wyłącz w tej wersji odpowiada układowi analizowanemu w punkcie 4.3.4, a występujące niewielkie różnice w stanach przejściowych zakodowanej tablicy przejść–wyjść nie zmieniają działania tego układu.

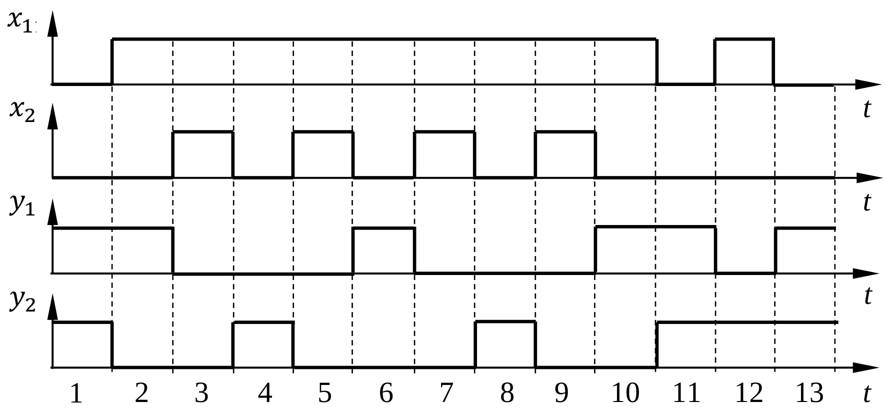
4.4.4. Układ włącz–wyłącz IV

Opis słowny

Układ umożliwia włączanie i wyłączanie dwóch urządzeń w zależności od stanu czujnika generującego dwa sygnały opisujące mierzoną wartość. Należy przyjąć, że czujnik wysyła sygnały 00, jeśli mierzona wartość jest niska, tzn. odczyt jest poniżej dolnej wartości progowej, czujnik wysyła sygnały 11, jeśli mierzona wartość jest wysoka, tzn. odczyt jest powyżej górnej wartości progowej, a sygnały 10 dla odczytów pomiędzy wartościami progowymi (sygnały 01 nie mogą wystąpić). Urządzenia powinny być włączone, jeśli czujnik sygnalizuje wartość niską, powinny być wyłączone dla wartości wysokich, a w przypadku wartości pośrednich urządzenia powinny pracować na przemian.

Wykres czasowy

Do przedstawienia działania układu niezbędne są cztery wykresy: wykresy przedstawiające zmiany stanu czujnika (x_1 i x_2) i wykresy przestawiające stan pracy urządzeń (y_1 i y_2).



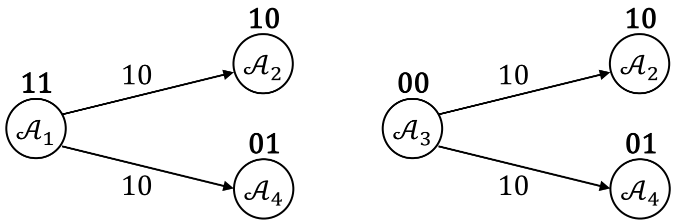
Rys.4.39. Wykres czasowy układu

Graf przejść i tablica przejść–wyjść

Analiza wykresu czasowego przedstawionego na rys. 4.39 pozwala na zidentyfikowanie 4 stanów wewnętrznych:

$\mathcal{A}_1$ – czujnik sygnalizuje wartość niską, urządzenia włączone,
$\mathcal{A}_2$ – czujnik sygnalizuje wartość pośrednią, włączone pierwsze urządzenie,
$\mathcal{A}_3$ – czujnik sygnalizuje wartość wysoką, urządzenia wyłączone,
$\mathcal{A}_4$ – czujnik sygnalizuje wartość pośrednią, włączone drugie urządzenie.

Próba narysowania grafu przejść dla tak wyznaczonych stanów kończy się niepowodzeniem (rys. 4.40). Okazuje się, że po zmianie odczytu czujnika z wartości niskiej do pośredniej układ powinien przejść ze stanu $\mathcal{A}_1$ do stanu $\mathcal{A}_2$ albo do stanu $\mathcal{A}_4$. Podobnie zmiana odczytu czujnika z wartości wysokiej do pośredniej wiąże się z przejściem ze stanu $\mathcal{A}_3$ do stanu $\mathcal{A}_2$ albo do stanu $\mathcal{A}_4$.

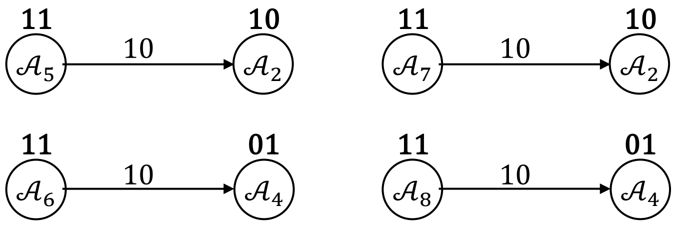


Rys.4.40. Problemy przy rysowaniu grafu przejść

Rozwiązanie powyższego problemu polega na wprowadzeniu dwóch stanów opisujących zachowanie układu przy niskim odczycie czujnika i dwóch stanów dla odczytu wysokiego:

$\mathcal{A}_5$ – czujnik sygnalizuje wartość niską, urządzenia włączone, poprzednio pracowało urządzenie drugie,
$\mathcal{A}_6$ – czujnik sygnalizuje wartość niską, urządzenia włączone, poprzednio pracowało urządzenie pierwsze,
$\mathcal{A}_7$ – czujnik sygnalizuje wartość wysoką, urządzenia wyłączone, poprzednio pracowało urządzenie drugie,
$\mathcal{A}_8$ – czujnik sygnalizuje wartość wysoką, urządzenia wyłączone, poprzednio pracowało urządzenie pierwsze.

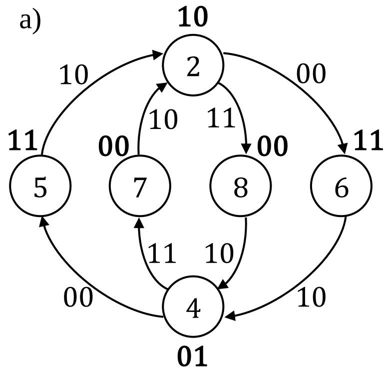
Po zastąpieniu stanu $\mathcal{A}_1$ stanami $\mathcal{A}_5$ i $\mathcal{A}_6$, a stanu $\mathcal{A}_3$ stanami $\mathcal{A}_7$ i $\mathcal{A}_8$ można już rozwiązać problemy przedstawione na rys. 4.40 (rys. 4.41).



Rys.4.41. Rozwiązanie problemów przy rysowaniu grafu przejść

Rysunek 4.42 przedstawia graf przejść i tablicę przejść–wyjść układu z uwzględnieniem nowych stanów wewnętrznych.

a)



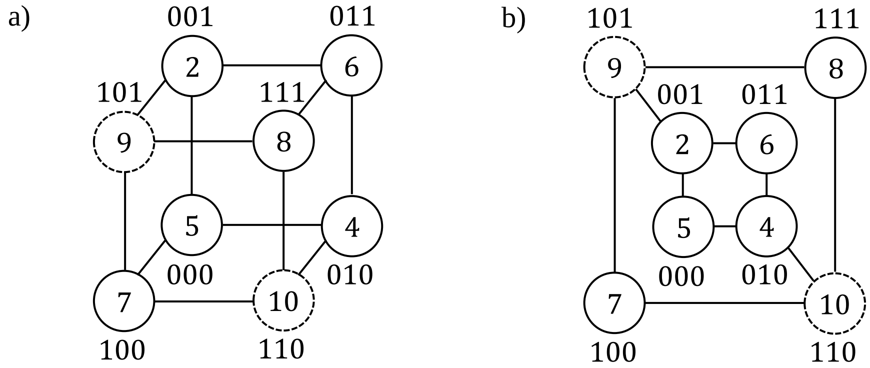
b)

$\mathcal{A}$ \ $\mathcal{X}$	00	01	11	10	$\mathcal{Y}$
2	6	–	8	2	10
4	5	–	7	4	01
5	5	–	–	2	11
6	6	–	–	4	11
7	–	–	7	2	00
8	–	–	8	4	00

Rys.4.42. Graf przejść i tablica przejść–wyjść układu

Redukcja i kodowanie stanów

Biorąc pod uwagę wymaganie niesprzeczności stanów wyjściowych, łatwo zauważyć, że tablica przejść–wyjść układu nie daje możliwości redukcji stanów (nie można połączyć wierszy 5 i 6 oraz wierszy 7 i 8). Projektowany układ ma więc sześć stanów wewnętrznych. Wykorzystywane do kodowania w poprzednich przykładach dwuelementowe ciągi zerojedynkowe pozwalają na zakodowanie maksymalnie czterech stanów, więc w tym przypadku niezbędne jest zastosowanie kodowania z pomocą ciągów trójelementowych. Przykładowe rozmieszczenie grafu przejść na sześcianie, umożliwiające właściwe zakodowanie stanów, zostało pokazane na rys. 4.43.



Rys.4.43. Kodowanie stanów a) metoda sześcianów, b) wykres przejść

Ze względu na to, że stany wewnętrzne 2 i 4 są połączone z czterema pozostałymi stanami układu, a z każdego z wierzchołków sześcianu wychodzą tylko trzy krawędzie, dodane zostały dodatkowe stany wewnętrze 9 i 10 pozwalające na realizację dwóch połączeń w sposób pośredni. Połączenia stanu 2 ze stanami 5 i 6 są połączeniami bezpośrednimi, połączenia ze stanami 7 i 8 są połączeniami realizowanymi z pomocą stanu pośredniego 9. Podobnie realizowane są połączenia stanu 4: połączenia ze stanami 5 i 6 są połączeniami bezpośrednimi, połączenia ze stanami 7 i 8 są połączeniami realizowanymi za pomocą stanu pośredniego 10.

Wprowadzenie dodatkowych stanów wymaga odpowiedniego uzupełnienia tablicy programu. Na rys. 4.44a powtórzona została tablica programu z rys. 4.42b. Rysunek 4.44b pokazuje tablicę programu po wprowadzeniu stanów 9 i 10. Wprowadzenie przejść cyklicznych przez stan 9 wymagało modyfikacji wiersza reprezentującego stan 2: w kolumnie 11 stan 8 został zastąpiony stanem 9, wymuszając przejście do stanu pośredniego, a właściwe przejście do stanu 8 zostało zapisane w wierszu oznaczonym jako 9. Podobnie poprawione zostały wiersze 4, 7 i 8, pozwalając na realizację przejść: $4 \rightarrow 10 \rightarrow 7$, $7 \rightarrow 9 \rightarrow 2$ i $8 \rightarrow 10 \rightarrow 4$.

a)

$\mathcal{X}$ / $\mathcal{A}$	00	01	11	10	$\mathcal{Y}$
2	6	–	8	2	10
4	5	–	7	4	01
5	5	–	–	2	11
6	6	–	–	4	11
7	–	–	7	2	00
8	–	–	8	4	00

b)

$\mathcal{X}$ / $\mathcal{A}$	00	01	11	10	$\mathcal{Y}$
2	6	–	9	2	10
4	5	–	10	4	01
5	5	–	–	2	11
6	6	–	–	4	11
7	–	–	7	9	00
8	–	–	8	10	00
9	–	–	8	2	–0
10	–	–	7	4	0–

Rys.4.44. Tablica programu a) pierwotna, b) uzupełniona

Warto też zauważyć, że wartości stanów wyjściowych w nowych stanach wewnętrznych zostały zapisane jako „–0” i „0–”. Stan 9 wykorzystywany jest do realizacji połączenia stanu 2 ze stanami 7 i 8, co wymaga zmiany stanu wyjściowego z 10 na 00, stąd wartość sygnału na pierwszym wyjściu w stanie 9 może być dowolna, a drugie wyjście powinno być wyzerowane w celu uniknięcia chwilowej zmiany sygnału wyjściowego podczas realizacji przejścia przez stan pośredni. Podobna analiza pozwoliła ustalić stan wyjściowy odpowiadający stanowi 10.

Ostateczną postać tablicy programu po uwzględnieniu proponowanego zbioru kodów przedstawiono na rys. 4.45a. Dla ułatwienia ostatniego etapu syntezy, zakodowana tablica programu jest porządkowana według kodu Graya (rys. 4.45b).

a)

x_1x_2 / $q_1q_2q_3$	00	01	11	10	y_1y_2
001	011	–	101	001	10
010	000	–	110	010	01
000	000	–	–	001	11
011	011	–	–	010	11
100	–	–	100	101	00
111	–	–	111	110	00
101	–	–	111	001	–0
110	–	–	100	010	0–

b)

x_1x_2 / $q_1q_2q_3$	00	01	11	10	y_1y_2
000	000	–	–	001	11
001	011	–	101	001	10
011	011	–	–	010	11
010	000	–	110	010	01
110	–	–	100	010	0–
111	–	–	111	110	00
101	–	–	111	001	–0
100	–	–	100	101	00

Rys.4.45. Zakodowana tablica programu przed i po uporządkowaniu wierszy

Ustalanie funkcji przejść i wyjść

Funkcje przejść i wyjść układu otrzymuje się, rozdzielając zakodowaną tablicę programu na tablice dla elementów pamięci i elementów wyjściowych (rys. 4.46). Pewną trudność przy zapisie wyrażeń logicznych stanowi w rozważanym przypadku to, że funkcje przejść są funkcjami pięciu zmiennych (stany wewnętrzne są kodowane za pomocą ciągów trójelementowych, a stany wejściowe za pomocą ciągów dwuelementowych). Tablice Karnaugha dla pięciu zmiennych mają dodatkową linię symetrii pełniącą funkcję lustra pozwalającego wskazać komórki sąsiednie. Lustro dzieli tablicę na połowę, rozdzielając trójelementowe kody wykorzystane do opisu wierszy czy kolumn. Grupy obejmujące sąsiednie kratki, które można wskazać po uwzględnieniu lustra zostały zaznaczone na rys. 4.46b i 4.46c linią kropkowaną (lustro zostało zaznaczone linią przerywaną).

a)

x_1x_2 / $q_1q_2q_3$	**00**	**01**	**11**	**10**
000	0	–	–	0
001	0	–	1	0
011	0	–	–	0
010	0	–	1	0
110	–	–	1	0
111	–	–	1	1
101	–	–	1	0
100	–	–	1	1

Q_1

b)

x_1x_2 / $q_1q_2q_3$	**00**	**01**	**11**	**10**
000	0	–	–	0
001	1	–	0	0
011	1	–	–	1
010	0	–	1	1
110	–	–	0	1
111	–	–	1	1
101	–	–	1	0
100	–	–	0	0

Q_2

c)

x_1x_2 / $q_1q_2q_3$	**00**	**01**	**11**	**10**
000	0	–	–	1
001	1	–	1	1
011	1	–	–	0
010	0	–	0	0
110	–	–	0	0
111	–	–	1	0
101	–	–	1	1
100	–	–	0	1

Q_3

d)

x_1x_2 / $q_1q_2q_3$	y_1
000	1
001	1
011	1
010	0
110	0
111	0
101	–
100	0

e)

x_1x_2 / $q_1q_2q_3$	y_2
000	1
001	0
011	1
010	1
110	–
111	0
101	0
100	0

Rys.4.46. Zakodowana tablica po rozdzieleniu na pięć tablic

Ostatecznie funkcje przejść i wyjść układu można zapisać jako:

$$
\begin{aligned}
Q_1 &= x_2 + q_1q_2q_3 + q_1\bar{q}_2\bar{q}_3,\\
Q_2 &= \bar{x}_1q_3 + q_2q_3 + x_2q_1q_3 + x_1\bar{q}_1q_2 + x_1\bar{x}_2q_2,\\
Q_3 &= \bar{x}_1q_3 + x_2q_3 + \bar{q}_2q_3 + x_1\bar{x}_2\bar{q}_2,\\
y_1 &= \bar{q}_1\bar{q}_2 + \bar{q}_1q_3,\\
y_2 &= \bar{q}_1q_2 + \bar{q}_1\bar{q}_3.
\end{aligned}
$$

4.4.5. Przyczyny błędów

Wyścigi

Błędne kodowanie stanów prowadzi do powstawania niekorzystnych zjawisk w układach sekwencyjnych takich jak *wyścigi krytyczne*. Wyścig występuje wtedy, gdy zmiana stanu wewnętrznego układu wymaga równoczesnej zmiany więcej niż jednego elementu pamięci. Równoczesna zmiana jest w praktyce niemożliwa, elementy pamięci zaczynają się „ścigać” – dochodzi do wyścigu. W przypadku dwóch elementów może się np. zdarzyć, że:

1) wyścig „wygra” (zmieni swój stan wcześniej) pierwszy element,
2) wyścig „wygra” drugi element,
3) obywa zmienią stan jednocześnie.

Układ będzie mógł przejść ze stanu niestabilnego do stabilnego na kilka sposobów. Jeżeli w każdym z możliwych przypadków osiągnie on ten sam stan stabilny, to wyścig taki nazywany jest *wyścigiem niekrytycznym*, jeżeli przejście może nastąpić do różnych stanów stabilnych, to wyścig jest *wyścigiem krytycznym*. Pozostawianie w projekcie układu wyścigów niekrytycznych nie jest błędem i pozwala w niektórych przypadkach na uzyskanie układów o mniejszej liczbie elementów pamięci [23]. Wyścigi krytyczne muszą być eliminowane, ponieważ prowadzą do błędnego działania układu. Stosowanie zasady kodowania wymuszającej przypisywanie kodów w taki sposób, aby niezbędne przejścia pomiędzy stanami układu były realizowane przy zmianie stanu tylko jednego elementu pamięci likwiduje problem wyścigów.

Na rys. 4.47 i 4.48 pokazane zostały przykłady wyścigów niekrytycznego i krytycznego. Rozważone zostały przejścia ze stanu stabilnego 00 pod wypływem zmiany wejścia z 0 na 1. Pod tablicami podane zostały trzy warianty przebiegu wyścigu (1 – wygrywa pierwszy, 2 – wygrywa drugi, 3 – remis).

a)

q_1q_2 \ x	**0**	**1**
00	00	11
01		11
11		11
10		11

a1: 00 → 10 → 11
a2: 00 → 01 → 11
a3: 00 → 11

b)

q_1q_2 \ x	**0**	**1**
00	00	11
01		01
11		01
10		11

b1: 00 → 10 → 11 → 01
b2: 00 → 01
b3: 00 → 11 → 01

Rys.4.47. Przykłady wyścigu niekrytycznego

a)

q_1q_2 \ x	**0**	**1**
00	00	11
01		01
11		11
10		10

a1: 00 → 10
a2: 00 → 01
a3: 00 → 11

b)

q_1q_2 \ x	**0**	**1**
00	00	11
01		11
11		11
10		10

b1: 00 → 10
b2: 00 → 01 → 11
b3: 00 → 11

Rys.4.48. Przykłady wyścigu krytycznego

Niestabilność

Błędy na etapie projektowania mogą prowadzić do niestabilności układu. Niestabilność oznacza, że układ ze stanu niestabilnego nie może wrócić do żadnego ze swoich stanów stabilnych (wpada w drgania), a jedyną możliwością przerwania drgań jest zmiana stanu wejść układu.

Układ, którego tablica przejść została przedstawiona na rys. 4.49 wpada w drgania po zmianie stanu wejść z 0 na 1, przełączając się cyklicznie pomiędzy niestabilnymi stanami 1 i 0. Drgania układu może przerwać jedynie zmiana stanu wejścia. W zależności od momentu, w którym ta zmiana nastąpi układ przejdzie do stanu stabilnego 0 lub 1.

q \ x	**0**	**1**
0	0	1
1	1	0

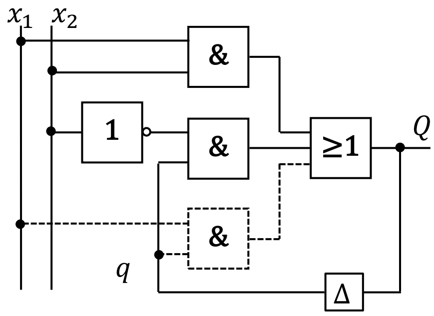
$\rightarrow 1 \rightarrow 0$

Rys.4.49. Przykład niestabilności

Hazard

Zjawisko hazardu zostało omówione dla układów kombinacyjnych, gdzie na skutek opóźnień wprowadzanych przez elementy układu poprawnie logicznie zaprojektowany układ chwilowo generuje na wyjściu błędny sygnał. W przypadku układów sekwencyjnych chwilowe przekłamanie sygnału z bloku pamięci może prowadzić do błędnego działania układu, wymuszając przejście do niewłaściwego stanu stabilnego. Przy projektowaniu układów sekwencyjnych, zapisując funkcje logiczne opisujące układ, należy pamiętać o uwzględnianiu grup antyhazardowych. Wyjątki od tej reguły, tzn. sytuacje, w których hazard nie jest szkodliwy zostały opisane np. w [10]. Na rys. 4.50a został pokazany schemat prostego układu sekwencyjnego pozwalającego na ilustrację problemu hazardu.

a)



$$Q = x_1x_2 + \bar{x}_2q \boxed{+ x_1q}$$

b)

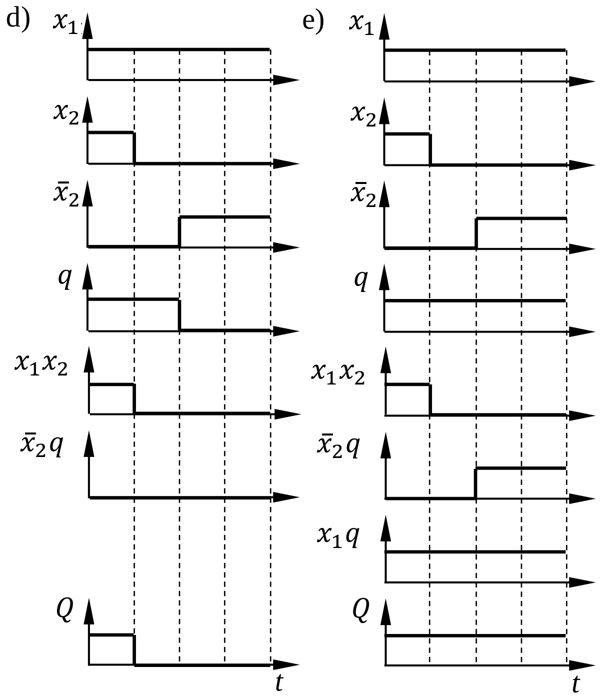
q \ x_1x_2	**00**	**01**	**11**	**10**
0	0	0	1	0
1	1	0	1	1

Q

c)

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

Q

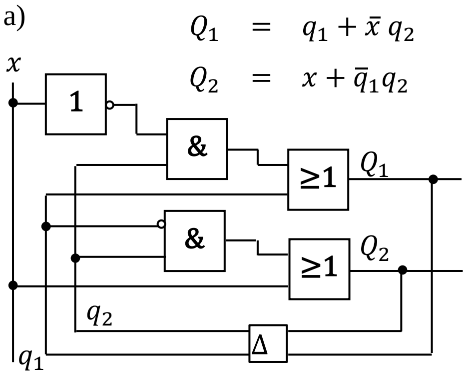


Rys.4.50. Analiza wpływu hazardu na działanie układu a) schemat układu b) i c) tablica przejść, d) i e) wykresy czasowe układu z hazardem i bez hazardu

Wprowadzona grupa antyhazardowa została zaznaczona linią przerywaną na schemacie (rys. 4.50a) oraz otoczona linią kropkowaną w funkcji przejść i tablicy przejść tego układu (rys. 4.50b). W tablicy przejść z rys. 4.50c zaznaczono stany stabilne układu. Okazuje się, że jeśli układ znajduje się w stanie wewnętrznym $q = 1$ dla wejść $x_1 = 1$ i $x_2 = 1$, zmiana drugiego wejścia, w przypadku braku grupy antyhazardowej, prowadzi do zmiany stanu wewnętrznego ($Q = 0$), pomimo że taka zmiana nie powinna nastąpić.

Analizę działania układu przedstawiono w wersji z hazardem na rys. 4.50d i w wersji bez hazardu na rys. 4.50e. Dla uproszczenia założono, że czas przejścia sygnałów przez bramki AND i OR jest pomijalnie mały, a jedyne opóźnienia wprowadza bramka NOT i element opóźniający Δ umieszczony w pętli sprzężenia zwrotnego. Ze względu na opóźnienie wprowadzane przez bramkę NOT po zmianie sygnału x_2 górna bramka AND będzie już miała na wyjściu wartość 0, a środkowa bramka AND będzie miała jeszcze wartość 0. W wyniku tego na wyjściu układu pojawi się fałszywy sygnał 0, który zostanie przekazany na wyjście bramki środkowej, blokując możliwość pojawienia się sygnału 1 na wyjściu tej bramki. Ostatecznie wyjście układu pozostanie w nieprawidłowym stanie 0. Problem ten usuwa wprowadzenie dolnej bramki AND, która przez cały czas utrzymuje sygnał o wartości 1, blokując możliwość pojawienia się na wyjściu układu sygnału o wartości 0.

a)



b)

x / q_1q_2	0	1
00	0	0
01	1	0
11	1	1
10	1	1

Q_1

x / q_1q_2	0	1
00	0	1
01	1	1
11	0	1
10	0	1

Q_2

c)

x / q_1q_2	0	1
00	00	01
01	11	01
11	10	11
10	10	11

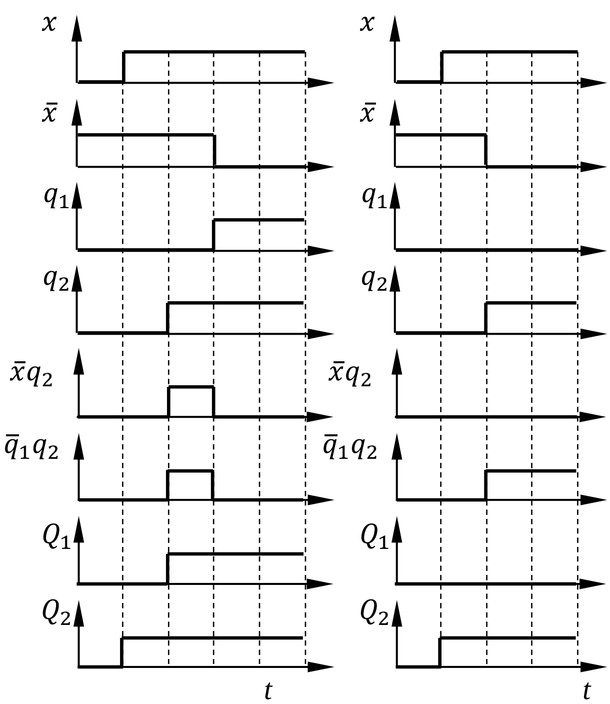
Q_1Q_2

d)

x / q_1q_2	0	1
00	00	01
01	11	01
11	10	11
10	10	11

Q_1Q_2

e) f)



Rys.4.51. Analiza wpływu hazardu na działanie układu a) schemat układu, b) tablice elementów Q_1 i Q_2, c) i d) tablice przejść, e) i f) wykresy czasowe układu z hazardem i bez hazardu

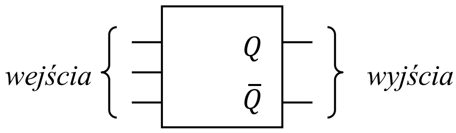
W układach sekwencyjnych o co najmniej dwóch elementach pamięci może występować jeszcze inny rodzaj hazardu, tzw. *hazard zasadniczy*. Zmiana sygnału wejściowego jest w tym przypadku przetwarzana na tyle szybko przez jeden z elementów pamięci, że drugi z elementów reaguje wcześniej na zmianę sygnału pamięci niż na zmianę sygnału wejściowego. W efekcie układ może przejść do niewłaściwego stanu stabilnego. W celu wyeliminowania hazardu tego typu, należy wprowadzić dodatkowe elementy opóźniające w taki sposób, żeby elementy pamięci reagowały szybciej na zmianę stanu wejść niż na zmiany pozostałych elementów pamięci.

Na rys. 4.51a pokazano schemat prostego układu sekwencyjnego pozwalającego na ilustrację problemu hazardu zasadniczego. W tablicy przejść z rys. 4.51c zaznaczone zostały stany stabilne układu. Zgodnie z tablicą, układ znajdujący się w stanie stabilnym $q_1 = 0$, $q_2 = 0$, po ustawieniu wejścia powinien przejść do stanu $q_1 = 0$, $q_2 = 1$, tak jak pokazują to strzałki narysowane liniami ciągłymi (przejście do kolumny 1 w wierszu 00 i przejście do wiersza 01). Jeśli drugi element pamięci zareaguje na zmianę wejścia szybciej, co pokazują strzałki narysowane liniami przerywanymi, to układ będzie się przełączał pomiędzy stanami niestabilnymi $q_1 = 0$, $q_2 = 1$ i $q_1 = 1$, $q_2 = 1$ (w kolumnie 0 tablicy), a na koniec przejdzie do stanu stabilnego $q_1 = 1$, $q_2 = 1$ (w kolumnie 1 tablicy). Zachowanie układu w wersji z hazardem i bez zostało przedstawione na wykresach czasowych (rys. 4.51e i 4.51f). Dla uproszczenia założono, że opóźnienia są wprowadzane wyłącznie przez element opóźniający Δ i bramkę NOT. Na wykresie 4.51e opóźnienie wprowadzane przez bramkę NOT jest dłuższe (wynosi 2 takty), co skutkuje nieprawidłowym działaniem układu.

4.5. Bistabilne przerzutniki asynchroniczne

Przerzutniki są elementarnymi układami sekwencyjnymi zdolnymi do zapamiętania jednego bitu informacji. Zapamiętany bit informacji, czyli 1 lub 0, odpowiada dwóm stanom wewnętrznym przerzutnika 1 i 0. W przerzutnikach bistabilnych obydwa stany wewnętrzne są stanami stabilnymi, a zmiana stanu wewnętrznego, a więc również stanu wyjść, następuje wyłącznie w wyniku zmiany stanu wejść. Przerzutniki monostabilne mają tylko jeden stabilny stan wewnętrzny, po zmianie stanu wejść układ ze stanu stabilnego może przejść do drugiego stanu (tzw. quasi-stabilnego), z którego po pewnym czasie samoistnie powraca do stanu stabilnego.

Przerzutniki asynchroniczne pracują w odróżnieniu od przerzutników synchronicznych bez sygnału taktującego, zmiana stanu pamięci następuje bezpośrednio, w dowolnej chwili czasu, pod wpływem zmiany stanu wejść. Przerzutniki te posiadają tzw. wejścia informacyjne, które w zależności od typu przerzutnika pełnią różne funkcje oraz dwa wzajemnie się uzupełniające (komplementarne) wyjścia: Q i $\bar{Q}$. Na schematach stosowany jest ogólny sposób oznaczania przerzutników przedstawiony na rys. 4.52.



Rys.4.52. Symbol graficzny przerzutnika

Do opisu funkcjonowania przerzutników wykorzystywane są:

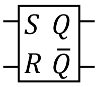
- tablice przejść,
- grafy przejść,
- tablice wzbudzeń,
- tablice charakterystyczne.

W *tablicach wzbudzeń* zapisywane są wartości stanów wejść układu niezbędne do przełączania tego układu pomiędzy wszystkimi możliwi stanami jego pracy. *Tablice charakterystyczne* to zmodyfikowane tabele prawdy, które mają tyle wierszy, ile jest różnych stanów wejściowych układu. Kolumny tablicy opisują wartości sygnałów wejściowych oraz wartości bieżącego i następnego stanu wewnętrznego (stany bieżący i następny oznaczane są symbolami q i Q).

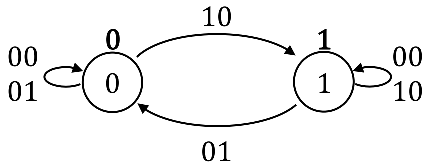
4.5.1. Asynchroniczny przerzutnik SR

Przerzutnik SR jest najprostszym przerzutnikiem, ma dwa wejścia: wejście ustawiające oznaczane symbolem S (ang. set, tzn. ustaw) i wejście zerujące oznaczane symbolem R (ang. reset, tzn. zeruj). Symbol graficzny przerzutnika wraz z różnymi stosowanymi w praktyce metodami opisu jego działania został przedstawiony na rys. 4.53.

a)



b)



c)

q \ SR	**00**	**01**	**11**	**10**
0	0	0	*	1
1	1	0	*	1

Q

d)

S	R	Q
0	0	q
0	**1**	0
1	0	**1**
1	**1**	*

e)

q	Q	S	R
0	0	0	–
0	**1**	**1**	0
1	0	0	**1**
1	**1**	–	0

f)

S	R	q	Q
0	0	0	0
0	0	**1**	**1**
0	**1**	0	0
0	**1**	**1**	0
1	0	0	**1**
1	0	**1**	**1**
1	**1**	0	*
1	**1**	**1**	*

Rys.4.53. Przerzutnik SR a) symbol graficzny, b) graf przejść, c) tablica przejść, d) tablica charakterystyczna, e) tablica wzbudzeń, f) tabela prawdy (oznaczono stan zabroniony)*

Na podstawie tablicy i grafu przejść układu (rys. 4.53b i 4.53c) można opisać działanie przerzutnika w następujący sposób :

- jeżeli na wejście ustawiające S zostanie podany sygnał 1, a na wejście zerujące R sygnał 0, to przerzutnik zostanie „ustawiony”, tzn. na wyjściu Q pojawi się 1, a na $\bar{Q}$ sygnał 0,
- jeżeli na wejście ustawiające S zostanie podany sygnał 0, a na wejście zerujące R sygnał 1, to przerzutnik zostanie „wyzerowany”, tzn. na wyjściu Q pojawi się 0, a na $\bar{Q}$ sygnał 1,
- jeżeli na obydwa wejścia zostanie podany sygnał 0, to stan wyjść przerzutnika nie zmieni się,
- podanie sygnału 1 na obydwa wejścia przerzutnika jest zabronione.

Identyczny opis funkcjonowania układu jest zawarty w pozostałych tabelach. W *tabeli prawdy* dla każdej kombinacji wartości sygnałów wejściowych i poprzedniej wartości sygnału wyjściowego q podana jest nowa wartość sygnału wyjściowego Q, analiza wszystkich przypadków prowadzi do takich samych wniosków, co analiza tablicy i grafu przejść.

Z kolejnych wierszy *tablicy charakterystycznej* można odczytać, że:

- dla wejść $S = 0$ i $R = 0$ stan wyjścia Q nie zmienia się (jest równy q, czyli poprzedniej wartości tego wyjścia),
- na wyjściu Q pojawia się sygnał 0 dla sygnałów wejściowych $S = 0$ i $R = 1$,
- na wyjściu Q pojawia się sygnał 1 dla sygnałów wejściowych $S = 1$ i $R = 0$.

Z *tablicy wzbudzeń* wynika, że:

- jeśli stan wyjścia ma pozostać równy 0 ($q = 0$ i $Q = 0$), to wejście S należy wyzerować, a stan wejścia R może być dowolny (dla $S = 0$ i $R = 0$ przy $q = 0$ stan wyjścia pozostanie bez zmian, tzn. $Q = 0$, natomiast dla $S = 0$ i $R = 1$ wyjście zostanie wyzerowanie, tzn. $Q = 0$),
- jeśli stan wyjścia ma ulec zmianie z $q = 0$ na $Q = 1$, to na wejściu S musi pojawić się wartość 1, a wejściu R wartość 0,
- jeśli stan wyjścia ma ulec zmianie z $q = 1$ na $Q = 0$, to na wejściu S musi pojawić się wartość 0, a wejściu R wartość 1,
- jeśli stan wyjścia ma pozostać równy 1 ($q = 1$ i $Q = 1$), to wejście R należy wyzerować, a stan wejścia S może być dowolny (dla $S = 0$ i $R = 0$ przy $q = 1$ stan wyjścia pozostanie bez zmian, tzn. $Q = 1$, a dla $S = 1$ i $R = 0$ wyjście zostanie ustawione $Q = 1$).

Funkcję przejść przerzutnika można ustalić na podstawie jego tablicy przejść, zapisując wyrażenia logiczne albo w postaci koniunkcyjnej, albo dysjunkcyjnej. Na rys 4.54 zaznaczone zostały grupy zer i jedynek oraz wynikające z nich postacie funkcji przejść układu.

a)

q \ SR	**00**	**01**	**11**	**10**
0	0	0	–	1
1	1	0	–	1

$Q = \bar{R}(q + S)$

b)

q \ SR	**00**	**01**	**11**	**10**
0	0	0	–	1
1	1	0	–	1

$Q = S + q\bar{R}$

Rys.4.54. Tablica przejść przerzutnika z zaznaczonymi grupami zer i jedynek

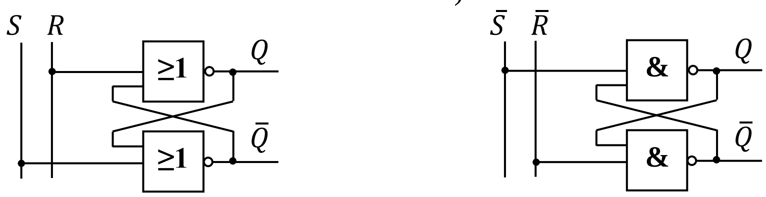
Przekształcając wyrażenia za pomocą prawa podwójnej negacji i praw de Morgana (zob. p. 3.2), otrzymuje się:

$$Q = \bar{R}(q+S) = \overline{\overline{\bar{R}(q+S)}} = \overline{R + \overline{(q+S)}},$$

$$Q = S + q\cdot\bar{R} = \overline{\overline{S + q\cdot\bar{R}}} = \overline{\bar{S}\cdot\overline{(q\cdot\bar{R})}}.$$

Wyrażenia te pozwalają na zbudowanie przerzutnika z wykorzystaniem dwóch bramek NOR lub dwóch bramek NAND (rys. 4.55). Przy założeniu, że stan $S = 1, R = 1$ jest zabroniony, na drugiej bramce obydwu układów otrzymuje się sygnał komplementarny $\bar{Q}$.

a) b)



Rys.4.55. Schemat przerzutnika SR z wykorzystaniem bramek a) NOR, b) NAND

4.5.2. Przerzutniki SR w układach sekwencyjnych

Przerzutniki SR mogą być wykorzystane do realizacji bloku pamięci w układach sekwencyjnych. W celu prawidłowego użycia przerzutników konieczne jest ustalenie *funkcji wzbudzeń* definiujących wartości na wejściach ustawiającym i zerującym. Funkcje wzbudzeń przerzutników mogą być określane z użyciem kilku podejść: z pomocą tablicy wzbudzeń przerzutnika, w wyniku porównania wyrażeń opisujących działanie przerzutnika i projektowanego układu czy poprzez analizę stanów niestabilnych tablicy przejść [10] lub grafu przejść.

Metoda tablic wzbudzeń

Wartości sygnałów, które powinny zostać podane na wejście przerzutnika, ustalane są na podstawie analizy zakodowanej tablicy przejść układu i tablicy wzbudzeń przerzutnika: stany wejść przerzutnika dobierane są w taki sposób, żeby zmienić stany układu, tak jak opisuje to tablica przejść. Sposób postępowania zostanie bardziej szczegółowo omówiony na dołączonych przykładach.

W pierwszym przykładzie zostanie pokazana konstrukcja układu włącz–wyłącz II (zob. p. 4.2.2 i 4.4.1) z wykorzystaniem przerzutnika SR. Na rys. 4.56 przedstawiono: tablicę przejść, tablicę wzbudzeń oraz skonstruowane na ich podstawie tablice opisujące funkcje wzbudzeń przerzutnika. Strzałki na rysunku wskazują reakcję na sygnał $x_1 = 0$, $x_2 = 0$ dla układu w stanie $q = 0$. Z tablicy przejść wynika, że stan układu nie zmienia się ($q = Q = 0$), więc na wejście S przerzutnika należy podać sygnał 0, a sygnał na wejściu R może mieć wartość dowolną. Odczytane w ten sposób wartości należy przepisać do tablic opisujących funkcje wzbudzeń wejść S i R przerzutnika. Odpowiadające sobie wartości zostały oznaczone indeksami [1] i [2]. Taką operację należy powtórzyć dla wszystkich stanów tablicy przejść projektowanego układu.

a)

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

Q

b)

q	Q	S	R
0	0	0^1	$-^2$
0	**1**	**1**	0
1	0	0	**1**
1	**1**	–	0

c)

q \ x_1x_2	**00**	**01**	**11**	**10**
0	0^1	0	0	1
1	–	0	0	–

S

d)

q \ x_1x_2	**00**	**01**	**11**	**10**
0	$-^2$	–	–	0
1	0	1	1	0

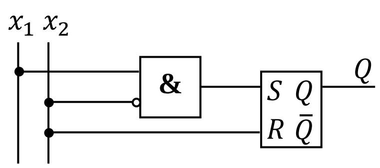
R

Rys.4.56. Metoda tablic wzbudzeń a) tablica przejść układu, b) tablica wzbudzeń przerzutnika, c)i d) tablice dla funkcji wzbudzeń przerzutnika

Z zaznaczonych na rys. 4.56c i 4.56d grup jedynek wynika, że funkcje wzbudzeń przerzutnika można zapisać wyrażeniami:

$$S = x_1\bar{x}_2, \quad R = x_2.$$

Układ włącz–wyłącz II może być więc zbudowany zgodnie ze schematem przedstawionym na rys. 4.57.



Rys. 4.57. Schemat układu włącz–wyłącz II

W przypadku układów z wieloma elementami pamięci każdy z elementów może być zrealizowany z wykorzystaniem przerzutnika, dla którego funkcje wzbudzeń określa się w sposób opisany powyżej. Na rys. 4.58 zostały przedstawione kolejne etapy wyznaczania funkcji wzbudzeń dwóch przerzutników wykorzystanych do realizacji układu włącz–wyłącz I wykorzystującego przycisk (zob. p. 4.2.1 i 4.4.2).

a)

q_1q_2 \ x	**0**	**1**
00	0	0
01	1	0
11	1	1
10	0	1

Q_1

b)

q	Q	S	R
0	0	0^{3}	$-^{4}$
0	**1**	**1**	0
1	0	0	**1**
1	**1**	$-^{1}$	0^{2}

c)

q_1q_2 \ x	**0**	**1**
00	0	1
01	1	1
11	1	0
10	0	0

Q_2

d)

q_1q_2	y
00	0
01	1
11	1
10	0

e)

q_1q_2 \ x	**0**	**1**
00	0	0
01	1	0
11	–	–
10	0	$-^{1}$

S_1

f)

q_1q_2 \ x	**0**	**1**
00	–	–
01	0	–
11	0	0
10	1	0^{2}

R_1

g)

q_1q_2 \ x	**0**	**1**
00	0	1
01	–	–
11	–	0
10	0	0^{3}

S_2

h)

q_1q_2 \ x	**0**	**1**
00	–	0
01	0	0
11	0	1
10	–	$-^{4}$

R_2

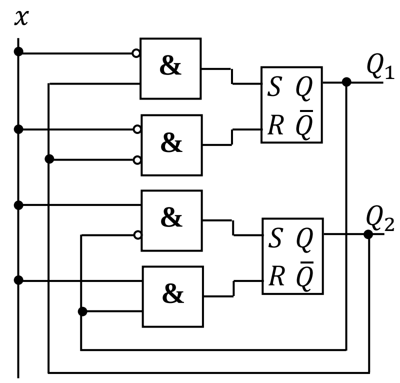
Rys.4.58. Metoda tablic wzbudzeń a) i c) tablica przejść układu po rozdzieleniu dla elementu Q_1 i Q_2, b) tablica wzbudzeń przerzutnika, d) tablica wyjść układu, e)-h) tablice dla funkcji wzbudzeń przerzutników

Strzałki narysowane liniami przerywanymi wskazują reakcję układu na sygnał $x = 1$, gdy znajduje się on w stanie $q_1 = 1$, $q_2 = 0$. Z tablicy przejść wynika, że stan pierwszego elementu pamięci się nie zmienia ($q_1 = Q_1 = 1$), więc wejście R pierwszego przerzutnika należy wyzerować, a sygnał na wejściu S może być dowolny. Wartości odczytane z tablicy wzbudzeń i wartości przepisane do tablic opisujących funkcje wzbudzeń pierwszego przerzutnika zostały oznaczone indeksami ^{1} i ^{2}. W analizowanym przypadku stan drugiego elementu pamięci również nie ulega zmianie ($q_2 = Q_2 = 0$). Z tablicy wzbudzeń przerzutnika wynika, że na wejście S drugiego przerzutnika należy podać sygnał 0, a sygnał na wejściu R może mieć dowolną wartość. Wartości te zostały oznaczone w tablicy wzbudzeń i w tablicach opisujących funkcje wzbudzeń drugiego przerzutnika indeksami ^{3} i ^{4}. Zaznaczone na rys. 4.58 grupy jedynek pozwalają na zapis funkcji wzbudzeń obydwu przerzutników jako:

$$S_1 = q_2\bar{x}, \quad R_1 = \bar{q}_2\bar{x},$$

$$S_2 = \bar{q}_1 x, \quad R_2 = q_1 x.$$

Schemat układu przedstawiono na rys. 4.59.



Rys. 4.59. Schemat układu włącz–wyłącz I

Metoda porównywania wyrażeń

Funkcje wzbudzeń przerzutnika można również otrzymać w wyniku porównania wyrażenia opisującego funkcję przejść projektowanego układu z odpowiednią formą (koniunkcyjną $Q = \bar{R}(q + S)$ lub dysjunkcyjną $Q = S + q\bar{R}$) wyrażenia opisującego działanie przerzutnika.

Rysunek 4.60 przedstawia tablicę i funkcje przejść układu włącz–wyłącz II.

a)

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

Q

b) $Q = \bar{x}_2(q + x_1)$

c) $Q = x_1\bar{x}_2 + q\bar{x}_2$

Rys.4.60. Metoda porównywania wyrażeń a) tablica przejść układu, b) i c) funkcje przejść układu w postaci koniunkcyjnej i dysjunkcyjnej

Z porównania postaci koniunkcyjnych przerzutnika i rozważanego układu:

$$Q = \bar{R}(q + S),$$
$$Q = \bar{x}_2(q + x_1)$$

wynika, że na wejście S przerzutnika powinien być przesłany sygnał z wejścia x_1, a na wejście R sygnał z wejścia x_2:

$$S = x_1,$$
$$R = x_2.$$

Z porównania postaci dysjunkcyjnych:

$$Q = S + q\bar{R},$$
$$Q = x_1\bar{x}_2 + q\bar{x}_2$$

wynika, że na wejścia S i R przerzutnika powinny być przesłane sygnały:

$$S = x_1\bar{x}_2,$$
$$R = x_2.$$

Różnica w otrzymanych postaciach funkcji wzbudzeń jest konsekwencją różnego wykorzystania zabronionego stanu wejściowego. W postaci koniunkcyjnej podanie sygnału o wartości 1 na wejścia ustawiające i zerujące prowadzi do wyzerowania wyjścia (zob. rys. 4.54) – taki przerzutnik jest nazywany przerzutnikiem *SR z dominującym wejściem zerującym*. Łącznik wyłączający w rozważanym układzie ma zgodnie z założeniami wyższy priorytet i w sytuacji gdy obydwa łączniki są włączone, urządzenie nie powinno pracować, co dokładnie odpowiada zasadzie działania przerzutnika z dominującym wejściem zerującym. W postaci dysjunkcyjnej podanie sygnału o wartości 1 na wejścia ustawiające i zerujące prowadzi do ustawienia wyjścia (zob. rys. 4.54) – taki przerzutnik jest nazywany przerzutnikiem *SR z dominującym wejściem ustawiającym*. Wykorzystanie tego przerzutnika do konstrukcji układu wymaga zabezpieczenia przed ustawieniem wyjścia, wyjście jest ustawiane tylko gdy na wejście pierwsze podany zostanie sygnał 1, a na wejście drugie sygnał 0.

W przypadku układów z wieloma elementami pamięci metodę porównywania wyrażeń należy zastosować oddzielnie dla każdego elementu pamięci. Dla układu włącz–wyłącz I z przyciskiem funkcje przejść zostały zapisane jako:

$$Q_1 = \bar{x}\, q_2 + q_1q_2 + x\, q_1,$$
$$Q_2 = \bar{x}\, q_2 + \bar{q}_1q_2 + x\, \bar{q}_1.$$

Ze względu na dysjunkcyjną postać obydwu wyrażeń przerzutniki wykorzystane do ich realizacji należy opisać w postaciach dysjunkcyjnych:

$$Q_1 = S_1 + q_1\bar{R}_1,$$
$$Q_2 = S_2 + q_2\bar{R}_2.$$

W celu uniknięcia pomyłki w funkcjach przejść projektowanego układu warto jest w wyrażeniach na Q_1 i Q_2 wyciągnąć przed nawias odpowiednio q_1 i q_2:

$$Q_1 = \bar{x}\, q_2 + q_1(q_2 + x),$$
$$Q_2 = x\, \bar{q}_1 + q_2(\bar{q}_1 + \bar{x}).$$

Z porównania odpowiednich wyrażeń na Q_1 i Q_2 wynika, że na wejścia ustawiające przerzutników należy wprowadzić sygnały:

$$S_1 = \bar{x}\, q_2,$$
$$S_2 = x\, \bar{q}_1.$$

Wartości sygnałów podawanych na wejścia zerujące wyznacza się, negując sygnały związane z bieżącym stanem wewnętrznym (q_1 oraz q_2) i stosując następnie prawa de Morgana:

$$R_1 = \overline{q_2 + x} = \bar{q}_2\bar{x},$$
$$R_2 = \overline{\bar{q}_1 + \bar{x}} = q_1 x.$$

Warto zauważyć, że otrzymane funkcje wzbudzeń obydwu przerzutników mają taką samą postać jak w przypadku zastosowania metody tablic wzbudzeń.

Metoda analizy stanów niestabilnych tablicy przejść

Stany niestabilne w tablicy przejść wymuszają zmiany odpowiednich elementów pamięci: niestabilne stany 1 wymuszają ustawienie elementu pamięci, więc mogą być wykorzystane do skonturowania wyrażenia dla wejścia ustawiającego przerzutnika, niestabilne stany 0 wymuszają wyzerowanie elementu pamięci mogą być zatem wykorzystane do określenia wyrażenia dla wejścia zerującego. Funkcje wzbudzeń przerzutników mogą być konstruowane zarówno z wykorzystaniem samych stanów niestabilnych, jak i z użyciem pozostałych stanów (stabilnych i nieokreślonych), co pozwala na uzyskanie wyrażeń w postaciach minimalnych.

Rysunek 4.61 przedstawia tablicę przejść układu włącz–wyłącz II z uwzględnieniem niestabilnej jedynki „***1***” i niestabilnych zer „***0***”.

a)

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

Q

b)

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

Q

Rys.4.61. Metoda analizy stanów niestabilnych – układ włącz–wyłącz II

Na rys. 4.61a został wyróżniony niestabilny stan 1, który pozwala na zapis funkcji dla wejścia ustawiającego w postaci kanonicznej lub minimalnej jako:

$$S = \bar{q}\, x_1\, \bar{x}_2,$$
$$S = x_1\, \bar{x}_2.$$

Tablica z rys. 4.61b pozwala na zdefiniowanie funkcji dla wejścia zerującego przerzutnika. *Uwaga*: uwzględniając, że zerowanie wyjścia przerzutnika następuje w wyniku podania sygnału o wartości 1 na wejście zerujące zera w tej tablicy muszą być traktowane jak jedynki i w konsekwencji funkcję wzbudzeń dla wejścia zerującego, należy zapisać w postaci dysjunkcyjnej. W tym przypadku może ona być zapisana jako:

$$R = q\,\bar{x}_1 x_2 + q\,x_1\,x_2,$$
$$R = x_2.$$

Warto zauważyć, że funkcje zapisane w postaciach minimalnych mają taką samą postać, jak w przypadku zastosowania metody tablic wzbudzeń czy metody porównywania wyrażeń.

W przypadku układów z wieloma elementami pamięci metoda analizy stanów niestabilnych musi być wykorzystana oddzielnie dla każdego elementu pamięci. Rozważmy układ włącz–wyłącz I, którego tablica przejść rozdzielona dla pierwszego i drugiego elementu pamięci została pokazana na rys. 4.62. Tak jak w przykładzie poprzednim, uwzględnione zostały niestabilne jedynki „*1*" i niestabilne zera „*0*".

a)

q_1q_2 \ x	**0**	**1**
00	0	0
01	*1*	0
11	1	1
10	0	1

Q_1

b)

q_1q_2 \ x	**0**	**1**
00	0	0
01	1	0
11	1	1
10	*0*	1

Q_1

c)

q_1q_2 \ x	**0**	**1**
00	0	*1*
01	1	1
11	1	0
10	0	0

Q_2

d)

q_1q_2 \ x	**0**	**1**
00	0	1
01	1	1
11	1	*0*
10	0	0

Q_2

Rys.4.62. Metoda analizy stanów niestabilnych – układ włącz–wyłącz I

Tablice z rys. 4.62a i 4.62b są tablicami dla pierwszego elementu pamięci układu. Niestabilna 1 w tablicy rys. 4.62a pozwala na zapis funkcji dla wejścia ustawiającego, a niestabilne 0 w tablicy z rys. 4.62b pozwala na zapis funkcji dla wejścia zerującego pierwszego przerzutnika. W postaci minimalnej funkcje te mogą być zapisane jako:

$$S_1 = \bar{x} q_2,$$
$$R_1 = \bar{x} \bar{q}_2.$$

Niestabilna 1 i niestabilne 0 w tablicach z rys. 4.62c i 4.62d dla drugiego elementu pamięci pozwalają na zapis funkcji wzbudzeń drugiego przerzutnika w postaci:

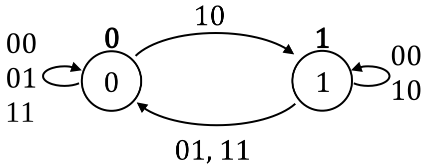
$$S_2 = x \bar{q}_1,$$
$$R_2 = x q_1.$$

Tak jak w poprzednim przykładzie, otrzymane funkcje wzbudzeń przerzutników odpowiadają wyrażeniom otrzymanym z metody tablic wzbudzeń czy metody porównywania wyrażeń.

Metoda analizy grafów przejść

Analiza stanów niestabilnych może być również przeprowadzona na podstawie grafu przejść układu. W przypadku układu z jednym elementem pamięci niestabilne stany 1 wyznaczają krawędź prowadzącą ze stanu 0 do stanu 1, opis tej krawędzi pozwala więc na ustalenie wyrażenia dla wejścia ustawiającego. Krawędź prowadząca ze stanu 1 do stanu 0 jest wyznaczana na podstawie niestabilnych stanów 0, pozwala więc na ustalenie wyrażenia dla wejścia zerującego przerzutnika.

Graf przejść układu włącz–wyłącz II został pokazany na rys. 4.63.



Rys.4.63. Metoda analizy grafu przejść – układ włącz–wyłącz II

Warunek dla wejścia ustawiającego przerzutnika można słownie opisać jako: *włączony łącznik pierwszy przy wyłączonym drugim*, co odpowiada wyrażeniu postaci:

$$S = x_1\bar{x}_2.$$

Warunek dla wejścia zerującego słownie można zapisać jako: *wyłączony pierwszy łącznik przy włączonym drugim lub włączone obydwa łączniki*, co odpowiada wyrażeniu:

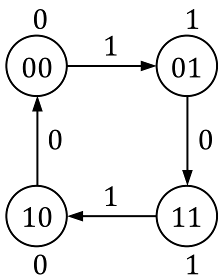
$$R = \bar{x}_1 x_2 + x_1 x_2$$

lub po skróceniu opisu do postaci: *włączony drugi łącznik (stan pierwszego łącznika jest nieistotny, może być włączony lub wyłączony)*, jako:

$$R = x_2.$$

Analiza stanów niestabilnych przeprowadzona z pomocą grafu przejść dała więc taki sam wynik, jak w przypadku metod rozważanych wcześniej.

Analizując stany niestabilne układu z wieloma elementami pamięci za pomocą grafu przejść, należy pamiętać, że zmiana stanu wybranego elementu pamięci zależy od stanu wejść oraz od stanu elementów pamięci tego układu. Rozważmy graf przejść układu włącz–wyłącz I pokazany na rys. 4.64.



Rys.4.64. Metoda analizy grafu przejść – układ włącz–wyłącz I

Krawędź prowadząca ze stanu 01 do stanu 11 pozwala na ustalenie wyrażenia dla wejścia ustawiającego pierwszego przerzutnika. Wyrażenie to można słownie opisać jako: *przycisk zwolniony przy ustawionym drugim elemencie pamięci*. Warunek dla wejścia zerującego wyznacza krawędź prowadząca ze stanu 10 do stanu 00. Warunek ten można zapisać jako: *przycisk zwolniony przy wyzerowanym drugim elemencie pamięci*. Opisy te pozwalają na zapis wyrażeń:

$$S_1 = \bar{x}q_2, \quad R_1 = \bar{x}\bar{q}_2.$$

Warunki dla drugiego przerzutnika są opisane w podobny sposób, dla wejścia ustawiającego (na podstawie krawędzi ze stanu 00 do stanu 01): *przycisk wciśnięty przy wyzerowanym pierwszym elemencie pamięci*, a dla wejścia zerującego (na podstawie krawędzi ze stanu 11 do stanu 10): *przycisk wciśnięty przy ustawionym pierwszym elemencie pamięci*. Prowadzi to do wyrażeń w postaci:

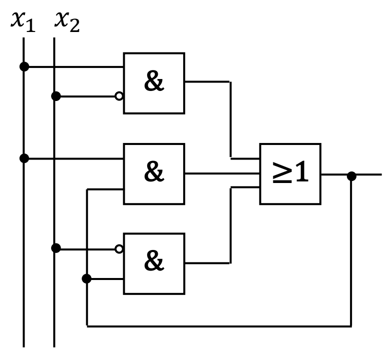
$$S_2 = x\bar{q}_1, \quad R_2 = xq_1.$$

Warto zauważyć, że otrzymane funkcje wzbudzeń przerzutników odpowiadają wyrażeniom otrzymanym wcześniej.

4.6. Zadania

4.1. Przeprowadź analizę układu, którego schemat został przedstawiony na rys. 4.65. W kolejnych krokach analizy należy:

a) zapisać funkcję przejść δ i funkcję wyjść λ,
b) przygotować tablicę przejść–wyjść,
c) narysować graf przejść układu,
d) przygotować opis słowny przedstawiający zwięźle działanie układu.



Rys.4.65. Schemat logiczny układu

4.2. Przeprowadź analizę układu o dwóch stanach wewnętrznych i dwóch wyjściach, którego funkcje przejść i wyjść opisane są wyrażeniami:

$$Q_1 = x_2 q_1 + \bar{x}_1 x_2 q_2,$$
$$Q_2 = x_2 + q_1 + x_1 q_2,$$
$$y_1 = q_1,$$
$$y_2 = q_2.$$

W kolejnych krokach analizy należy:

a) narysować schemat logiczny układu,
b) przygotować tablicę przejść–wyjść,
c) narysować graf przejść układu,
d) przygotować opis słowny przedstawiający zwięźle działanie układu.

4.3. Funkcje przejść układu z zad. 4.2 zostały uproszczone zgodnie z zależnościami:

$$Q_1 = x_2 q_1 + \bar{x}_1 x_2,$$
$$Q_2 = x_2 + x_1 q_2.$$

Czy wprowadzone modyfikacje:

a) zmieniły działanie układu,
b) przyczyniły się do pojawienia się wyścigów i jeśli tak, to jakich.

4.4. Zaprojektuj układ włączający urządzenie w zależności od wartości sygnałów z dwóch czujników temperatury x_1 i x_2. Czujniki generują sygnał o wartości 1, jeśli mierzona temperatura przekroczy zadaną wartość progową. Temperatury progowe są dobrane w taki sposób, że przy wzroście temperatury najpierw reaguje czujnik pierwszy, a później drugi. W przypadku gdy obydwa czujniki sygnalizują stan niski ($x_1 = x_2 = 0$), urządzenie nie powinno pracować, a włączenie urządzenia powinno nastąpić w chwili, gdy czujniki zasygnalizują stan wysoki ($x_1 = x_2 = 1$). Po włączeniu urządzenie powinno pracować do momentu, w którym obydwa czujniki zasygnalizują stan niski.

W kolejnych krokach syntezy należy:

a) narysować wykres czasowy przedstawiający przebieg w czasie zmian sygnałów wejściowych i wyjściowych układu,
b) przeprowadzić syntezę układu metodą Huffmana,
c) narysować schemat logiczny układu.

4.5. Zaprojektuj układ pełniący funkcję arbitra przydzielającego dwóm użytkownikom dostęp do wspólnego zasobu. Arbiter monitoruje wykorzystanie zasobu i żądania klientów. W przypadku gdy zasób jest wolny i żądanie jest zgłaszanie przez jednego z użytkowników, użytkownik ten dostaje zezwolenie na jego wykorzystanie. Układ ma dwa wejścia (żądania dwóch użytkowników) i dwa wyjścia (stan wysoki na wyjściu pierwszym oznacza przydzielenie zasobu pierwszemu użytkownikowi, a stan wysoki na drugim wyjściu drugiemu użytkownikowi). Należy założyć, że arbiter pracuje w trybie podstawowym (nie jest możliwa jednoczesna zmiana dwóch sygnałów wejściowych).

W kolejnych krokach syntezy należy:

a) narysować wykres czasowy przedstawiający przebieg w czasie zmian sygnałów wejściowych i wyjściowych układu,
b) przeprowadzić syntezę układu metodą Huffmana,
c) narysować schemat logiczny układu.

4.6. Zbuduj układy z poprzednich zadań wykorzystując przerzutniki SR.