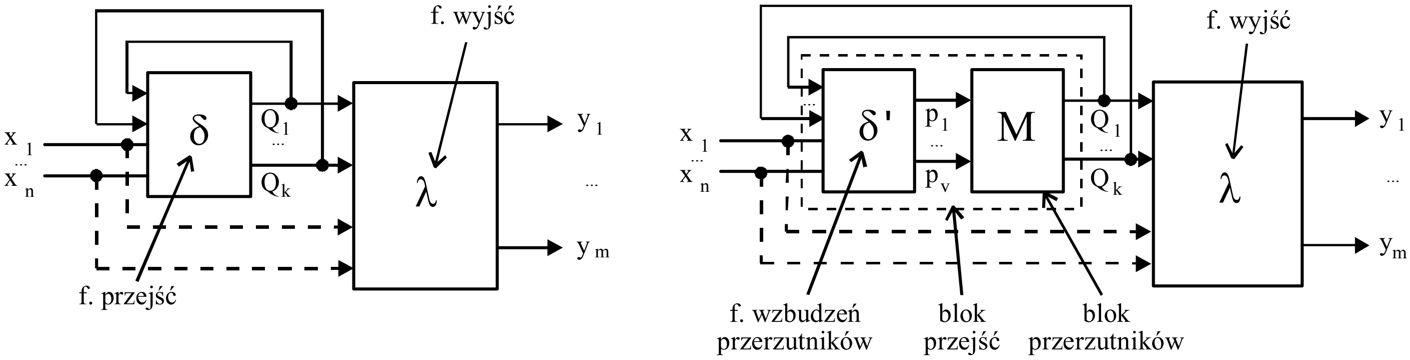
3. Synteza układów sekwencyjnych

Układ sekwencyjny. Jest to układ przełączający (rys. 11), w którym stan wyjść zależy nie tylko od aktualnego stanu wejść lecz także od poprzednich stanów wejść (historii). Układ taki musi więc być wyposażony w pamięć. Związki pomiędzy stanem wejść $X^{[t]}$, aktualnym stanem wewnętrznym $Q^{[t]}$ (w chwili *t*), oraz stanem wyjść $Y^{[t]}$ określają dwie funkcje:

- *funkcja przejść δ* - określa następny stan wewnętrzny jako:
  $Q^{[t+\tau]} = \delta(Q^{[t]}, X^{[t]})$,
- *funkcja wyjść λ* - określa stan wyjść jako:
  $Y^{[t]} = \lambda(Q^{[t]}, X^{[t]})$ lub $Y^{[t]} = \lambda(Q^{[t]})$.



Rys. 11. Schemat układu sekwencyjnego.

Z uwagi na sposób oddziaływania sygnałów wejściowych układy sekwencyjne dzielimy na:

- **asynchroniczne** - reagują natychmiast na każdą zmianę sygnału wejściowego,
- **synchroniczne** - reagują w chwilach czasowych ściśle określonych przez dodatkowy zewnętrzny sygnał taktujący (zegarowy).

Układ sekwencyjny może mieć ponadto strukturę:

- **automatu Moore'a** - stan wyjść zależy wyłącznie od stanu wewnętrznego układu,
- **automatu Mealy'ego** - stan wyjść zależy zarówno od stanu wewnętrznego jak i wejść.

Poniżej podano krótkie porównanie własności układów asynchronicznych i synchronicznych:

układ asynchroniczny	**układ synchroniczny**
- nie mają wejścia synchronizującego	- posiadają dodatkowe wejście synchronizujące (zegarowe, taktujące)
- reagują bezpośrednio na każdą zmianę stanu wejść X	- zmiana stanu wewnętrznego następuje dopiero pod wpływem sygnału zegarowego
- nowy stan ustala się po niezerowym opóźnieniu wynikającym z niezerowych i różnych czasów propagacji w elementach	- nowy stan ustala się po czasie τ od wystąpienia sygnału zegarowego (i musi być mniejszy niż odstęp kolejnych taktów zegarowych)

- stały stan na wejściu (0 lub 1) jest traktowany jako pojedynczy sygnał	- stały stan na wejściu (0 lub 1) trwający przez N taktów sygnału zegarowego traktowany jest jako ciąg N kolejnych sygnałów
- aby zapewnić jednoznaczność analizy, zakłada się, że nie występuje jednoczesna zmiana kilku sygnałów wejściowych oraz wszelkie zmiany wejść są dopuszczalne tylko wtedy, gdy stan wewnętrzny nie ulega zmianie	- w czasie pomiędzy impulsami zegarowymi może się jednocześnie zmieniać kilka sygnałów wejściowych
- mogą wystąpić zarówno stany stabilne ($Q^{[t+\tau]} = Q^{[t]}$) jak i niestabilne ($Q^{[t+\tau]} \neq Q^{[t]}$)	- wszystkie stany wewnętrzne są stabilne (bo są synchronizowane sygnałem zegarowym)
- mogą wystąpić niekorzystne zjawiska (np. tzw. *wyścigi*), które należy wyeliminować (co dodatkowo komplikuje projektowanie)	- nie występują wyścigi (prostsze projektowanie)

Poniżej przedstawiono metodę projektowania głównie asynchronicznych układów sekwencyjnych, która przy pewnych drobnych modyfikacjach może zostać użyta również do projektowania pewnej klasy układów synchronicznych spełniających poniższe założenia:

- blok przerzutników („pamięć”) będzie realizowany z przerzutników synchronicznych,
- częstotliwość sygnału zegarowego jest na tyle duża, że wszystkie zmiany sygnałów wejściowych zostaną zauważone (zakładamy małą częstość zmian sygnałów wejściowych),
- częstotliwość sygnału zegarowego jest na tyle mała, że zdążą zaniknąć wszystkie stany przejściowe (np. *hazardy*) występujące przy realizacji funkcji wzbudzeń przerzutników.

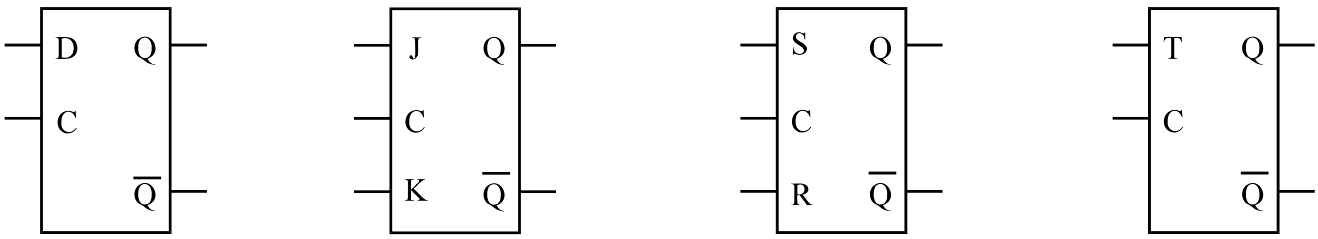
Podano także wskazówki dla projektowania układów synchronicznych.

Synteza układów sekwencyjnych (asynchronicznych lub synchronicznych) polega na:

1. określeniu grafu lub tablicy przejść i wyjść układu (opisującej automat Moore’a),
2. zmniejszeniu (minimalizacji) koniecznej liczby stanów wewnętrznych (redukcja stanów),
3. kodowaniu stanów tablicy przejść i wyjść (np. w naturalnym kodzie dwójkowym),
4. określeniu docelowej struktury bloku przejść (δ) i wyjść (λ) i realizacji tych funkcji.

Podstawowymi składnikami bloku przejść są przerzutniki. W układach synchronicznych zmieniają one swój stan dopiero pod wpływem sygnału zegarowego, w zależności od aktualnych wartości sygnałów na wejściach informacyjnych. Symbole najczęściej używanych przerzutników wraz z tablicami opisującymi ich pracę zostały przedstawione na rys. 12.

pamiętaj: *w układach synchronicznych zmiana stanu wewnętrznego następuje dopiero pod wpływem zewnętrznego sygnału taktującego (zegarowego)* a)



b)

D^n	Q^{n+1}
0	0
1	1

J^n	K^n	Q^{n+1}
0	0	Q^n
0	1	0
1	0	1
1	1	$\overline{Q^n}$

S^n	R^n	Q^{n+1}
0	0	Q^n
0	1	0
1	0	1
1	1	-

T^n	Q^{n+1}
0	Q^n
1	$\overline{Q^n}$

c)

$Q^n \rightarrow Q^{n+1}$	D	J	K	T	S	R
$0 \rightarrow 0$	0	0	-	0	0	-
$0 \rightarrow 1$	1	1	-	1	1	0
$1 \rightarrow 0$	0	-	1	1	0	1
$1 \rightarrow 1$	1	-	0	0	-	0

d)

D: $Q^{n+1} = D^n$

JK: $Q^{n+1} = J^n \overline{Q^n} + \overline{K^n} Q^n$

SR: $Q^{n+1} = S^n + \overline{R^n} Q^n$

T: $Q^{n+1} = T^n \overline{Q^n} + \overline{T^n} Q^n$

Rys. 12. Przerzutniki synchroniczne D, JK, SR, T: a) symbole, b) tablice przejść, c) tablice wzbudzeń, d) funkcje przejść.

Przy projektowaniu układów przełączających najbardziej przydatna jest tablica wzbudzeń, opisująca stan wejść informacyjnych, przy którym występuje określona zmiana stanu wyjść przerzutników (np. z zera na jeden ($0 \rightarrow 1$), itp.).

Określenie tablicy przejść i wyjść.

Pierwszym krokiem w syntezie układu asynchronicznego jest przedstawienie jego właściwości funkcjonalnych (zadanych w formie „zewnętrznego” opisu słownego, wykresu czasowego lub ciągów zerojedynkowych) w formie pierwotnej tablicy przejść i wyjść.

Pierwotna tablica przejść i wyjść powinna zawierać wszystkie wewnętrzne stany stabilne oraz przejścia między nimi. Stan uważa się za stabilny, jeżeli przebiegi przejściowe dobiegły już końca, a jego zmiana może nastąpić dopiero pod wpływem zmiany stanu wejść. W praktyce, będzie to każdy stan, któremu w danej chwili czasu, dla danego stanu wejść odpowiada określony (jednoznaczny) stan wyjść (wynikający ze stanów poprzednich). Przejściem nazywana jest zmiana stanu wewnętrznego, wynikająca ze zmiany stanu wejść.

Pierwotną tablicę przejść i wyjść budujemy tak aby:

- liczba kolumn równała się liczbie różnych kombinacji sygnałów wejściowych,
- liczba wierszy równała się liczbie stanów stabilnych układu,
- każdemu stanowi wewnętrznemu odpowiadał ściśle określony stan wyjść układu.

Przy syntezie układów synchronicznych zwykle **najpierw** tworzy się graf przejść i wyjść, a dopiero **potem** na jego podstawie tablicę przejść i wyjść. Pamiętajmy o tym, że dla układów synchronicznych wszystkie stany wewnętrzne są **stabilne**.

Wypełnianie pierwotnej tablicy przejść i wyjść:

1. wpisujemy wszystkie stany stabilne, tzn. takie, dla których przy niezmiennym stanie wejść następny stan wewnętrzny pokrywa się z aktualnym (stan wewnętrzny się nie zmienia),
2. następnie wpisujemy stany niestabilne układu – występujące krótkotrwale, gdy zmienił się już stan wejść ale stan wewnętrzny jeszcze nie zdążył się zmienić,
3. wykrywamy ew. stany nieokreślone, występujące przy jednoczesnej zmianie kilku sygnałów wejściowych (ale przy założeniach poczynionych powyżej mogą być one utożsamiane ze stanami niestabilnymi, bo docelowo układ będzie realizowany jako synchroniczny!!!),
4. pozostałe kratki uzupełniamy kreskami (stan dowolny – niezdefiniowany przy opisie zadania) lub wpisujemy dowolny stan (zmniejsza to jednak później możliwości minimalizacji),
5. poszczególnym stanom przypisujemy odpowiedni stan wyjść (tablica wyjść).

Redukcja pierwotnej tablicy przejść i wyjść.

Po przygotowaniu pierwotnej tablicy przejść i wyjść poddajemy ją procesowi minimalizacji (redukcji liczby stanów wewnętrznych). Redukcji mogą podlegać stany:

- **równoważne** - odpowiadają im niesprzeczne stany wyjść: (1,1), (1,-), (-,1);
  - przyporządkowane mają jednakowe stany wejść;
  - reakcja na wejścia jest **identyczna**,
- **pseudorównoważne** - odpowiadają im niesprzeczne stany wyjść;
  - przyporządkowane mają jednakowe stany wejść;
  - reakcja na wejścia jest **niesprzeczna**,
- **zgodne** - reakcja na wejścia jest niesprzeczna.

Jeżeli stany wewnętrzne są równoważne (pseudorównoważne) pod warunkiem równoważności (pseudorównoważności) innych stanów wewnętrznych to takie stany nazywamy równoważnymi (pseudorównoważnymi) warunkowo.

Przykład redukcji stanów:

a)

stan	x1x2 (we) 00	01	11	10	(wy) Y1	Y2
1	2	1	3	-	0	1
2	2	4	-	7	1	1
3	2	1	3	-	0	1
4	2	4	3	-	0	-
5	5	-	-	7	1	1
6	-	4	6	7	0	0
7	8	-	6	7	1	0
8	8	1	-	7	0	0

b)

stan	x1x2 00	01	11	10	Y1	Y2
1,3, 4	2	1	3	-	0	1
2,5	2	4	-	7	1	1
6	-	4	6	7	0	0
7	8	-	6	7	1	0
8	8	1	-	7	0	0

c)

stan	x1x2 00	01	11	10	Y1	Y2
a	b	a	a	-	0	1
b	b	a	-	d	1	1
c	c	a	c	d	0	0
d	c	-	c	d	1	0

- stany równoważne → 1, 3 oraz 1, 4 → 1, 3, 4 (a)
- stany pseudorównoważne → 2, 5 (b)
- stany pseudorównoważne warunkowo → 6, 8 (c) (bo 1, 4 są równoważne)
- stany zgodne → a, b oraz c, d

Rys. 13. Redukcja stanów: a) tablica pierwotna, b) tablica po redukcji stanów równoważnych i pseudorównoważnych, c) tablica po redukcji stanów warunkowych

Przed przystąpieniem do redukcji stanów tablicy pierwotnej (rys. 13a), stany „stabilne" warto umieścić w „kółku" lub zacieniować innym kolorem. Zwiększenia to przejrzystość oraz ułatwia analizę pracy układu. Redukcję rozpoczynamy od stanów równoważnych i pseudorównoważnych (rys. 13b). Stany niestabilne (przejścia) zastępujemy stabilnymi, kilka stanów stabilnych odpowiadających jednej kratce zastępujemy jednym (zwykle o najniższym numerze), stany dowolne uzupełniamy stanami określonymi (jeżeli występują). Jeśli po jednokrotnej redukcji stwierdzimy, iż w dalszym ciągu mamy stany dające się zredukować, wówczas powyższą procedurę należy powtórzyć. Sytuacja taka występuje w przypadku istnienia stanów równoważnych lub pseudorównoważnych warunkowo. Ponieważ 1 oraz 4 reprezentują ten sam stan wewnętrzny, 6 i 8 mogą być zredukowane. Na koniec należy uporządkować tablicę, nadając nowe oznaczenia wierszom i kratkom (rys. 13c).

stan	00	01	11	10
a	a/11	a/01	a/01	c/10
c	c/00	a/01	c/00	c/10

Rys. 14. Tablica pierwotna po redukcji stanów zgodnych.

Redukcję stanów zgodnych wykonujemy tylko dla automatu Mealy'ego. W takim przypadku (rys. 14) każdemu stanowi wewnętrznemu może odpowiadać inny stan wyjść, w zależności od aktualnego stanu wejść. Każda kratka tablicy, oprócz następnego stanu wewnętrznego zawiera nowy stan wyjść.

Kodowanie stanów wewnętrznych.

Poprawność działania układów asynchronicznych w dużej mierze zależy od przyjętego sposobu kodowania, ze względu na możliwość wystąpienia tzw. *wyścigów krytycznych*. Zjawisko wyścigów ma miejsce wówczas, gdy przy przejściu ze stanu aktualnego do stanu następnego w zapisach (kodach) tych stanów zmieniają się przynajmniej dwa bity jednocześnie.

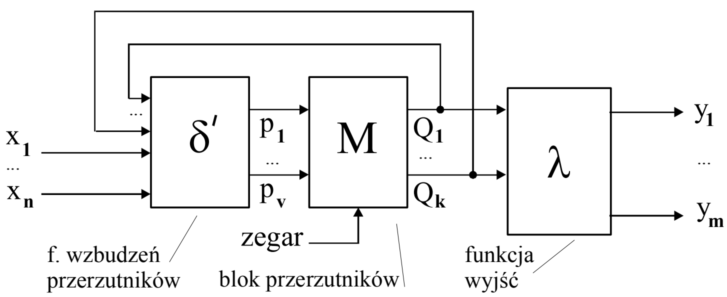
Kodowanie stanów wewnętrznych w układach synchronicznych jest znacznie prostsze, gdyż nie występuje w nich niebezpieczeństwo wyścigów. Sposób kodowania jest w zasadzie dowolny, więc najczęściej są stosowane kody NB (naturalny binarny) i Graya, jakkolwiek jest możliwe użycie innych kodów mających na celu uproszczenie budowy bloku wzbudzeń (δ') lub bloku wyjść (λ), albo dostosowanych do typu użytych przerzutników (D, JK, ...).

Określenie struktury bloku przejść (δ) i wyjść (λ).

Pierwszą czynnością jest wybór typu przerzutników (synchronicznych), które mają być użyte w bloku δ (rys. 11). Najczęściej stosowane są przerzutniki D lub JK, dla których tablice wzbudzeń podano na rys. 12. Na podstawie wybranego typu przerzutnika tworzy się tablice wzbudzeń, a następnie projektuje bloki bloku wzbudzeń δ' i wyjść λ (jako układy kombinacyjne),

Automat Moore'a

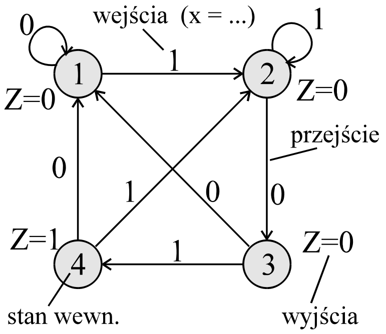
Automatem Moore'a nazywamy układ przełączający, w którym stan wyjść układu zależy w sposób bezpośredni od stanu wewnętrznego (każdemu stanowi wewnętrznemu odpowiada ściśle określony stan wyjść).



$$Q^{[t+\tau]} = \delta(Q^{[t]}, X^{[t]})$$
$$Y^{[t]} = \lambda(Q^{[t]})$$

Rys. 15. Automat Moore'a.

Działanie automatu może być określone tablicą programu lub grafem przejść. W przypadku automatu Moore'a gałęzie grafu opisane są odpowiednim stanem wejść, natomiast stan wyjść przypisany jest wierzchołkom, reprezentującym stan wewnętrzny układu. Graf przejść stanowi zwykle bezpośredni odpowiednik pierwotnej tablicy programu, czasem bywa punktem wyjścia do jej tworzenia. Syntezę układu przeprowadza się na podstawie zredukowanej tablicy programu, w której stany wewnętrzne przedstawione są w postaci liczb dwójkowych reprezentujących stan elementarnych układów pamiętających (przerzutników).



	$x =\ldots$		(wy)
stan	0	1	Z
1	1	2	0
2	3	2	0
3	1	4	0
4	1	2	1

	$x =\ldots$		(wy)
Q1Q2	0	1	Z
00	00	01	0
01	11	01	0
11	00	10	0
10	00	01	1

Rys. 16. Automat Moore'a: a) graf przejść, b) tablica pierwotna, c) tablica zakodowana.

Na rys. 16a pokazany został przykładowy graf przejść automatu Moore'a realizujący wykrywanie sekwencji „101", podawanej na wejście szeregowe (x), sygnalizowanej jako impuls jedynkowy na wyjściu Z. Grafowi temu odpowiada tablica programu jak na rys. 16b. Do zakodowania czterech stanów wewnętrznych wymagane jest użycie dwóch przerzutników ($4=2^2$) o wyjściach $Q1$, $Q2$. Wierszom tablicy (rys. 16c) przypisuje się stan wyjść przerzutników danego stanu wewnętrznego, w poszczególne kratki (zależnie od stanu wejść) wpisywane są „przejścia" do nowych stanów wewnętrznych (nowych stanów przerzutników). Korzystając z tablicy wzbudzeń przerzutnika JK należy następnie utworzyć tablice wzbudzeń Karnaugh (rys. 17). W kratkach wpisujemy stan na odpowiednim wejściu danego przerzutnika, przy którym wystąpi przejście $Q1^n Q2^n \rightarrow Q1^{n+1} Q2^{n+1}$. Np. dla $Q1Q2=01$ i $x=0$ nastąpi przejście do $Q1Q2=11$ ($Q1: 0 \rightarrow 1,\ Q2: 1 \rightarrow 1$), więc: $J1=1,\ K1=-,\ J2=-,\ K2=0$. Funkcja wyjść wynika wprost z tablicy programu: $Z = Q1 \cdot \overline{Q}2$.

$J1^n$:

Q1Q2 \ x	0	1
00	0	0
01	1	0
11	-	-
10	-	-

$K1^n$:

Q1Q2 \ x	0	1
00	-	-
01	-	-
11	1	0
10	1	1

$J2^n$:

Q1Q2 \ x	0	1
00	0	1
01	-	-
11	-	-
10	0	1

$K2^n$:

Q1Q2 \ x	0	1
00	-	-
01	0	0
11	1	1
10	-	-

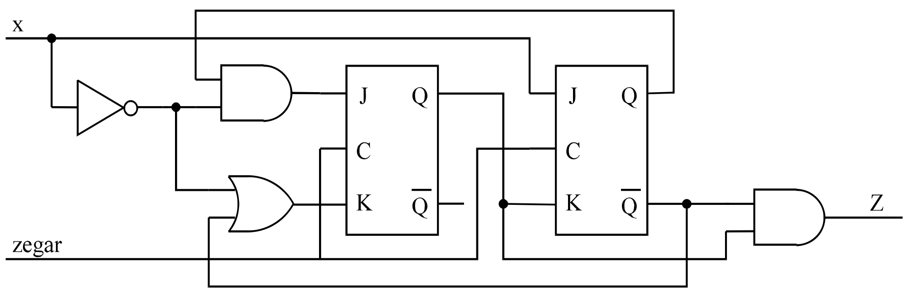
$$J1 = \overline{x} \cdot Q2$$

$$K1 = \overline{x} + \overline{Q2}$$

$$J2 = x$$

$$K2 = Q1$$

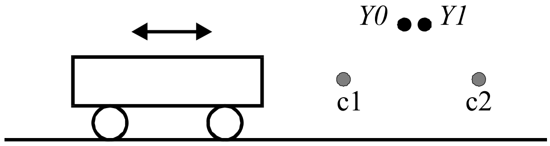
Rys. 17. Tablice wzbudzeń Karnaugh.



Rys. 18. Realizacja automatu Moore'a na przerzutnikach JK.

Przykład 2

Zaprojektować układ wykrywający kierunek ruchu wózka (rys. 19). Czujniki *c1* i *c2* reagują na obecność wózka. Ich rozstaw jest mniejszy niż długość wózka. Lampka *Y0* sygnalizuje przejazd wózka w lewo, a *Y1* - w prawo.



Rys. 19. Przykład 2.

Rozwiązanie:

Zadania tego nie da się rozwiązać jako układu kombinacyjnego, gdyż stanom czujników *c1c2* odpowiada różny stan wyjść *Y0Y1* w zależności od kierunku ruchu wózka. Do wyznaczenia kierunku ruchu potrzebna jest znajomość zarówno aktualnego jak i poprzedniego stanu wejść czujników. Jest to zatem układ sekwencyjny.

W zadaniu mamy trzy podstawowe grupy stanów związane z wyjściami układu: ruch w lewo (*Y0*=1, *Y1*=0), ruch w prawo (*Y0*=0, *Y1*=1) oraz wózek poza zasięgiem czujników (*Y0*=0, *Y1*=0). W grupach tych można natomiast wyróżnić stany wewnętrzne związane z aktywnością czujników rejestrujących ruch wózka w danym kierunku (*c1c2*={01,11,10}). W związku z tym otrzymujemy następującą listę stanów wewnętrznych :

1. wózek poza zasięgiem czujników (*Y0*=0, *Y1*=0, *c1c2*=00)
2. wózek porusza się w lewo (*Y0*=1, *Y1*=0, *c1c2*=01)
3. wózek porusza się w lewo (*Y0*=1, *Y1*=0, *c1c2*=11)
4. wózek porusza się w lewo (*Y0*=1, *Y1*=0, *c1c2*=10)
5. wózek porusza się w prawo (*Y0*=0, *Y1*=1, *c1c2*=10)
6. wózek porusza się w prawo (*Y0*=0, *Y1*=1, *c1c2*=11)
7. wózek porusza się w prawo (*Y0*=0, *Y1*=1, *c1c2*=01)

Na jej podstawie możemy już zbudować pierwotną tablicę przejść i wyjść (jak dla układu asynchronicznego - rys. 20a). Symbolem „*” zostały oznaczone sytuacje, które nie wystąpią przy założeniu niezmienności kierunku ruchu wózka pomiędzy czujnikami. Zostały one zaznaczone dla zwrócenia uwagi na wpływ, jaki mają na działanie programu. Stany wewnętrzne 2, 3, 4 mimo, iż w stanie stabilnym ustawiają wyjścia w ten sam sposób, nie mogą być zredukowane, różnią się bowiem reakcją na zmianę stanu czujników (powodującą zmianę stanu wyjść). Jeśli jednak założymy niezmienność kierunku ruchu wózka, wówczas stany 2, 3, 4 oraz 5, 6, 7 można zredukować. Tablica przejść i wyjść po redukcji przyjmie wtedy postać jak na rys. 20b.

a)

	c1c2				
stan	**00**	**01**	**11**	**10**	**Y0Y1**
1	1	2	-	5	00
2	1*	2	3	-	10
3	-	7*	3	4	10
4	1	-	6*	4	10
5	1*	-	6	5	01
6	-	7	6	4*	01
7	1	7	3*	-	01
8	-	-	-	-	--

b)

	c1c2				
stan	**00**	**01**	**11**	**10**	**Y0Y1**
a	a	b	-	c	00
b	a	b	b	b	10
c	a	c	c	c	01
d	-	-	-	-	--

Rys. 20: a) Pierwotna tablica stanów,
b) tablica po redukcji.

Po zakodowaniu tablicy przejść i wyjść (rys. 21a), tworzymy tablice wzbudzeń (Karnaugh) przerzutników. Dla przerzutników „D" przyjmują one postać jak na rys. 21b.

a) b)

	c1c2				
Q1Q2	**00**	**01**	**11**	**10**	**Y0Y1**
00	00	01	--	11	00
01	00	01	01	01	10
11	00	11	11	11	01
10	--	--	--	--	--

c)

Q1Q2 \ c1c2	00	01	11	10
00	0	0	-	1
01	0	0	0	0
11	0	1	1	1
10	-	-	-	-

$D1 = Q1c2 + Q1c1 + \overline{Q2}c1$

Q1Q2 \ c1c2	00	01	11	10
00	0	1	-	1
01	0	1	1	1
11	0	1	1	1
10	-	-	-	-

$D2 = c1 + c2$

Rys. 21. a) Zakodowana tablica przejść-wyjść, b) tablice wzbudzeń przerzutników.

Funkcje wyjść wyznaczamy na podstawie tablic wyjść (Karnaugh) układu sterowania (rys. 22).

Q1 \ Q2	0	1
0	0	1
1	-	0

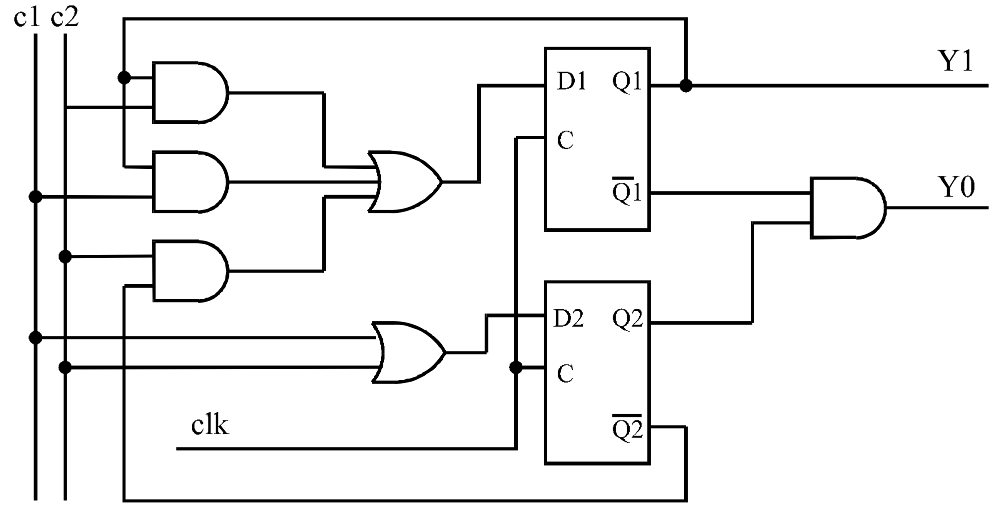
$Y0 = \overline{Q1}Q2$

Q1 \ Q2	0	1
0	0	0
1	-	1

$Y1 = Q1$

Rys. 22. Tablice wyjść (Karnaugh).

Realizacja układu przedstawiona została na rys.23.



Rys. 23. Schemat logiczny układu sterowania dla *Przykładu 2*.