

Temat 3. Synteza układów sekwencyjnych z bramek logicznych

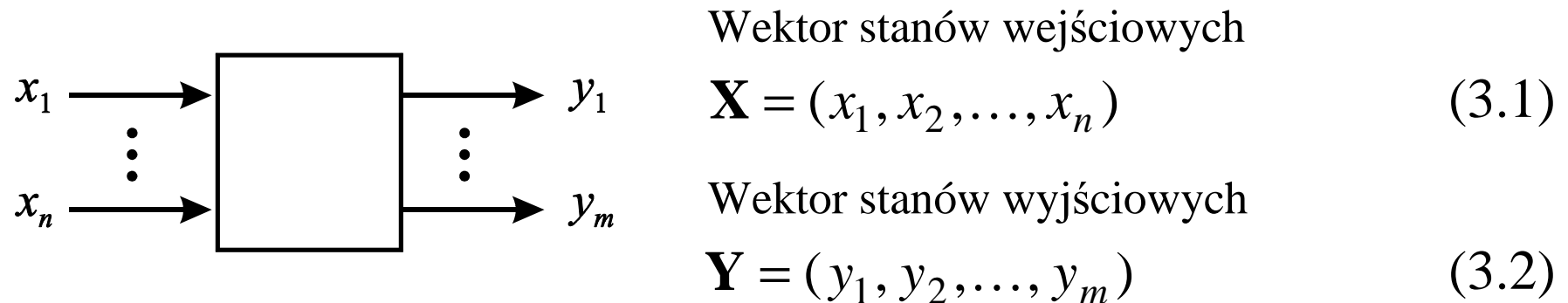
Spis treści do tematu 3

- 3.1. Wprowadzenie
- 3.2. Zjawiska wyścigów i hazardu w asynchronicznych układach sekwencyjnych.
- 3.3. Projektowanie asynchronicznych układów sekwencyjnych metodą tablic przejść/wyjść (Huffmana).
- 3.4. Literatura

3.1. Wprowadzenie

Opis układów cyfrowych (automatów)

Dowolny układ cyfrowy można przedstawić w postaci bloku, który posiada wejścia x_1, x_2, \dots, x_n oraz wyjścia y_1, y_2, \dots, y_m .



W układach kombinacyjnych (ang. *combinational logic circuits*)

$$\mathbf{Y}^t = f(\mathbf{X}^t) \quad (\text{brak pamięci poprzednich stanów}) \quad (3.3)$$

W układach sekwencyjnych (ang. *sequential logic circuits*)

$$\mathbf{Y}^t = f(\mathbf{X}^t, \mathbf{X}^{t-1}, \mathbf{X}^{t-2}, \dots) \quad (3.4)$$

gdzie \mathbf{X}^t i \mathbf{Y}^t oznaczają stany wejść i wyjść w dowolnej chwili czasu t , przez $t-1, t-2, \dots$ oznaczono wcześniejsze cykle pracy układu.

Stan pamięci układu sekwencyjnego opisuje się wektorem stanu wewnętrznego \mathbf{Q} złożonego ze wszystkich binarnych linii sprzężenia zwrotnego

$$\mathbf{Q} = (q_1, q_2, \dots, q_k). \quad (3.5)$$

Pamięć układu sekwencyjnego realizuje się w postaci układu kombinacyjnego objętego sprzężeniem zwrotnym. **Funkcja przejść** δ , realizowana przez ten układ kombinacyjny, określa kolejny stan wewnętrzny \mathbf{Q}^{t+1} na podstawie poprzedzającego stanu wewnętrznego \mathbf{Q}^t i wejściowego \mathbf{X}^t

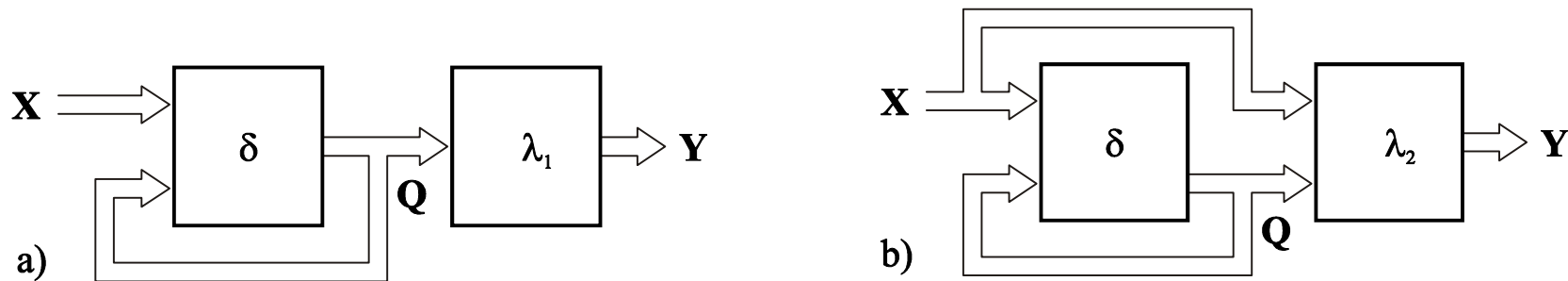
$$\mathbf{Q}^{t+1} = \delta(\mathbf{Q}^t, \mathbf{X}^t). \quad (3.6)$$

Nowy stan wewnętrzny automatu nazywamy **stanem stabilnym** jeżeli $\mathbf{Q}^{t+1} = \mathbf{Q}^t$.

Drugi układ kombinacyjny realizuje **funkcję wyjść** λ . Wyróżnia się automaty o strukturze Moore'a oraz Mealy'ego różniące się argumentami tej funkcji

$$\mathbf{Y}^t = \lambda_1(\mathbf{Q}^t),$$

$$\mathbf{Y}^t = \lambda_2(\mathbf{X}^t, \mathbf{Q}^t). \quad (3.7)$$



Rys. 3.1. Struktura asynchronicznych układów sekwencyjnych:
(a) automat Moore'a; (b) automat Mealy'ego.

Automatem deterministycznym nazywamy strukturę

$$A = (Q, X, Y, \delta, \lambda) \quad (3.8)$$

gdzie:

$Q = \{Q_1, Q_2, \dots, Q_k\}$ – zbiór stanów wewnętrznych,

$X = \{X_1, X_2, \dots, X_k\}$ – zbiór stanów wejściowych,

$Y = \{Y_1, Y_2, \dots, Y_k\}$ – zbiór stanów wyjściowych,

δ – funkcja przejść,

λ – funkcja wyjść.

Automatem zupełnym nazywamy automat, w którym funkcje δ oraz λ są zdefiniowane dla każdej pary wektorów (Q, X) ze zbioru $Q \times X$ (iloczyn kartezjański zbiorów).

Przyjęta symbolika:

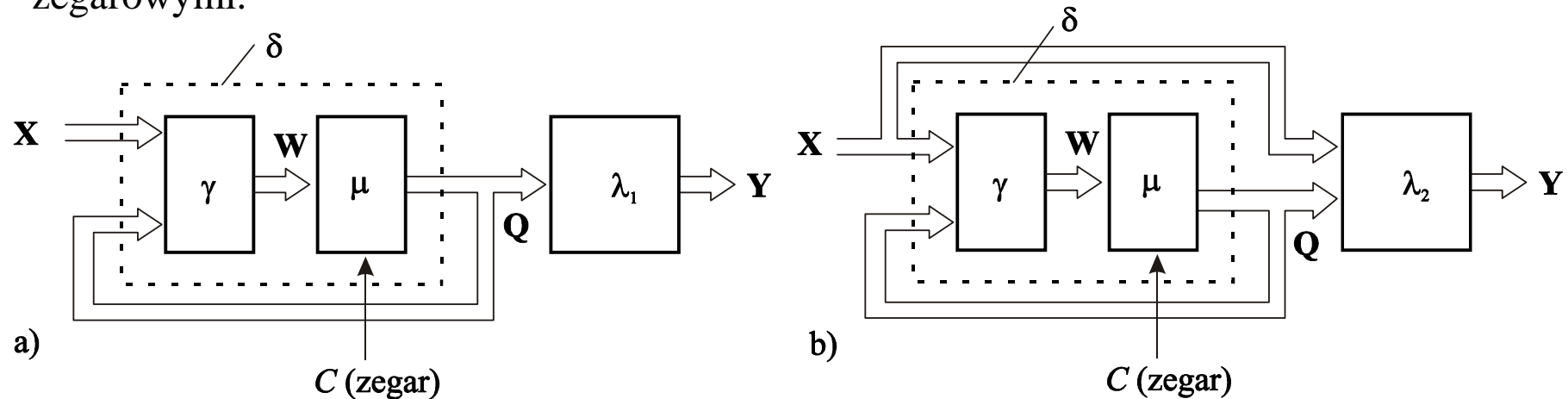
X – zbiór wektorów,

X_i – i -ty wektor w zbiorze X ,

x_i – i -ta binarna składowa wektora.

Asynchroniczne układy sekwencyjne nie posiadają żadnego wejścia zegarowego (synchronizującego). Stan wewnętrzny układu Q zmienia się natychmiast po zmianie sygnałów wejściowych X .

Synchroniczne układy sekwencyjne reagują na zmianę stanu wejściowego X tylko w dyskretnych chwilach czasowych, określonych przez zewnętrzny sygnał zegarowy C . Każdy kolejny stan wewnętrzny jest wytwarzany synchronicznie z impulsami zegarowymi.



Rys. 3.2. Struktura synchronicznych układów sekwencyjnych. Blok wejściowy δ jest złożony z układu kombinacyjnego γ oraz pamięci μ sterowanej sygnałem zegarowym. (a) automat Moore'a; (b) automat Mealy'ego.

Układy asynchroniczne są stosowane głównie w prostych układach automatyki przemysłowej i instalacjach alarmowych. Niejednakowe opóźnienia sygnałów w różnych elementach układu (wyścigi) powodują ryzyko nieprzewidzianych reakcji (hazard). W skomplikowanych układach analiza wyścigów i hazardu jest bardzo trudna. Problemy te nie występują w układach synchronicznych.

Rodzaje hazardu

Hazard statyczny (HS) – zamiast oczekiwanej niezmięionej wartości sygnału wyjściowego następuje chwilowe przejście do stanu błędnego i powrót do stanu prawidłowego. Możliwe są przejścia: $1 \rightarrow 0 \rightarrow 1$ zamiast niezmięionej wartości 1,
 $0 \rightarrow 1 \rightarrow 0$ zamiast niezmięionej wartości 0.

Hazard dynamiczny (HD) – zamiast oczekiwanej jednokrotnej zmiany stanu następuje więcej zmian zanim stan ustali się na prawidłowym docelowym poziomie. Możliwe są przejścia:
 $1 \rightarrow 0 \rightarrow 1 \rightarrow 0$ zamiast przejścia $1 \rightarrow 0$,
 $0 \rightarrow 1 \rightarrow 0 \rightarrow 1$ zamiast przejścia $0 \rightarrow 1$.

Hazard wielokrotny (wHS, wHD) – zwielokrotnione błędne przejścia typu hazard statyczny lub hazard dynamiczny.

Przejście hazardogenne – zmiana stanu wejść układu, przy którym możliwy jest chwilowy błąd wyjścia.

Synteza wyrażeń logicznych metodami Karnaugh'a i Q-MC w przypadku grupowania „1” prowadzi do wyrażeń normalnych o postaci alternatywnej (normalna postać sumy), np.:

$$Q_1 = q_2 \bar{A} + q_2 B + q_1 A \bar{B}. \quad (3.9)$$

W układach bramek logicznych zrealizowanych jako bezpośrednie odpowiedniki takich wyrażeń możliwe są następujące przypadki hazardu:

HS przy stanie 0 – nie występuje,

HS przy stanie 1 – **możliwy** dla przejść pomiędzy różnymi grupami „1” w tablicy Karnaugh'a lub w metodzie Q-MC,

HD – nie występuje.

Synteza wyrażeń logicznych metodami Karnaugh'a i Q-MC w przypadku grupowania „0” prowadzi do wyrażeń normalnych o postaci koniunkcyjnej (normalna postać iloczynu), np.:

$$Q_1 = (q_2 + A)(q_2 + \bar{B})(q_1 + \bar{A} + B). \quad (3.10)$$

W układach bramek logicznych zrealizowanych jako bezpośrednie odpowiedniki takich wyrażeń możliwe są następujące przypadki hazardu:

HS przy stanie 0 – **możliwy** dla przejść pomiędzy różnymi grupami „0” w tablicy Karnaugh'a lub w metodzie Q-MC,

HS przy stanie 1 – nie występuje,

HD – nie występuje.

Rozważmy normalną postać alternatywną przykładowej funkcji przejść (3.9) bez oraz z dodatkowymi wyrażeniami antyhazardowymi.

a) bez wyrażen antyhazardowych

		AB			
		00	01	11	10
q_2q_1	00	0	-	0	0
	01	0	-	0	1
	11	1	-	1	1
	10	1	-	1	0

Q_1

$$Q_1 = q_2 \bar{A} + q_2 B + q_1 A \bar{B},$$

b) z wyrażeniem antyhazardowym

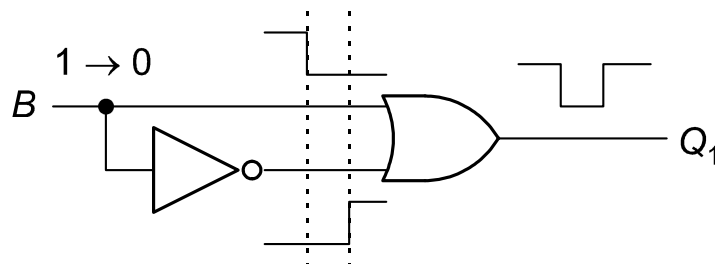
		AB			
		00	01	11	10
q_2q_1	00	0	-	0	0
	01	0	-	0	1
	11	1	-	1	1
	10	1	-	1	0

Q_1

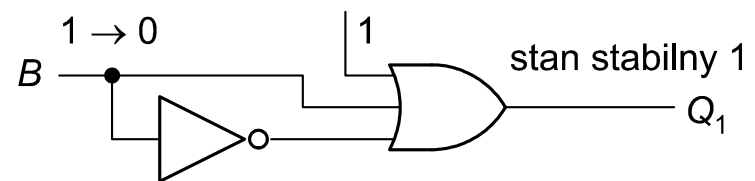
$$Q_1 = q_2 \bar{A} + q_2 B + q_1 A \bar{B} + q_1 q_2,$$

Rozważając wybrane przejście, np. $q_2q_1AB = 1111 \rightarrow 1110$, możemy wstępnie uprościć postać funkcji ustalając wyrazy, które nie biorą udziału w przejściu, tzn. $q_2 = q_1 = A = 1$

$$Q_1 = B + \bar{B},$$



$$Q_1 = B + \bar{B} + 1 = 1,$$



Rozważmy normalną postać koniunkcyjną przykładowej funkcji przejść (3.10) bez oraz z dodatkowymi wyrażeniami antyazardowymi.

a) bez wyrażen antyazardowych

		AB			
		00	01	11	10
q_2q_1	00	0	-	0	0
	01	0	-	0	1
	11	1	-	1	1
	10	1	-	1	0

Q_1

b) z wyrażeniem antyazardowym

		AB			
		00	01	11	10
q_2q_1	00	0	-	0	0
	01	0	-	0	1
	11	1	-	1	1
	10	1	-	1	0

Q_1

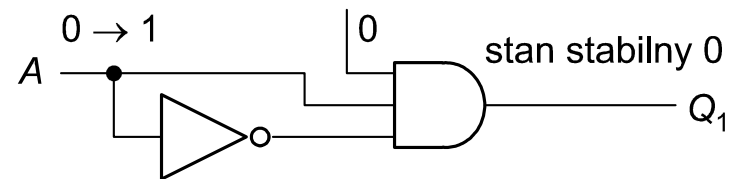
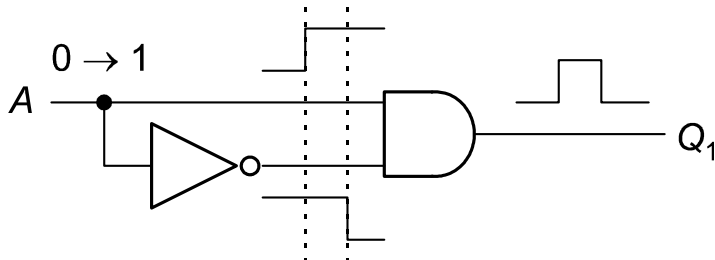
$$Q_1 = (q_2 + A)(q_2 + \bar{B})(q_1 + \bar{A} + B),$$

$$Q_1 = (q_2 + A)(q_2 + \bar{B})(q_1 + \bar{A} + B)(q_2 + q_1).$$

Rozważając wybrane przejście, np. $q_2q_1AB = 0000 \rightarrow 0010$, możemy wstępnie uprościć postać funkcji ustalając wyrazy, które nie biorą udziału w przejściu, tzn. $q_2 = q_1 = B = 0$

$$Q_1 = A \cdot \bar{A},$$

$$Q_1 = A \cdot \bar{A} \cdot 0 = 0,$$

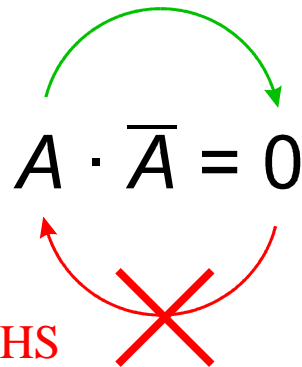


Wyrażenia w normalnej postaci alternatywnej (sumy) lub koniunkcyjnej (iloczynu) można zazwyczaj przekształcać przy użyciu praw algebry Boole'a bez ryzyka wprowadzenia dodatkowego hazardu przy przejściach między stanami sąsiednimi logicznie, tzn. takimi których kody binarne różnią się jednym bitem.

Wyjątki:

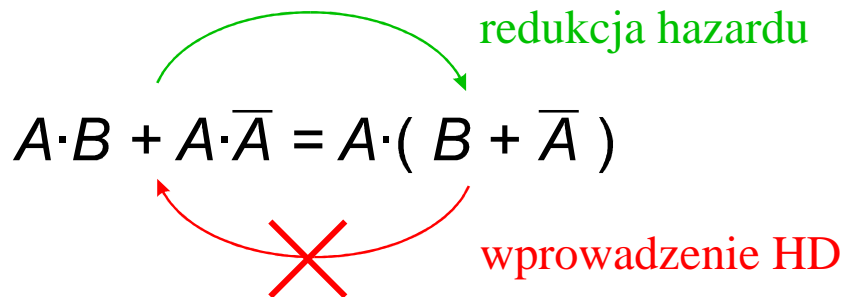
prawo sprzeczności:

redukcja hazardu



wprowadzenie HS

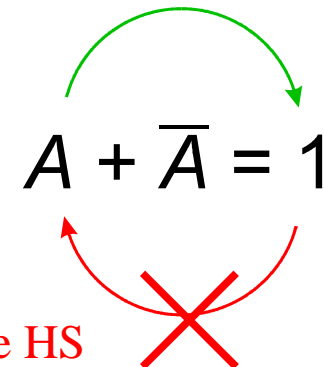
prawo rozdzielności iloczynu wzgl. sumy, gdy prowadzi do wyrażenia alternatywnego, które nie jest normalne, gdyż zawiera iloczyn tożsamościowo równy 0



wprowadzenie HD

prawo wyłączonego środka:

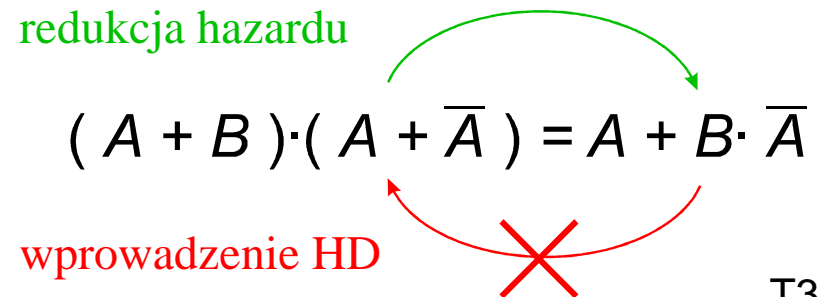
redukcja hazardu



wprowadzenie HS

prawo rozdzielności sumy wzgl. iloczynu logicznego, gdy prowadzi do wyrażenia koniunkcyjnego, które nie jest normalne, gdyż zawiera sumę tożsamościowo równą 1.

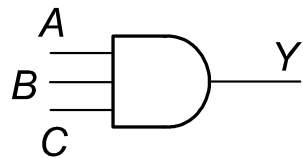
redukcja hazardu



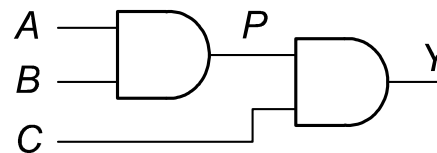
wprowadzenie HD

W przypadku bezpośrednich przejść między stanami, które nie są sąsiednie logicznie, większość przekształceń wykorzystujących prawa algebry Boole'a jest obciążonych ryzykiem wprowadzenia dodatkowego hazardu.

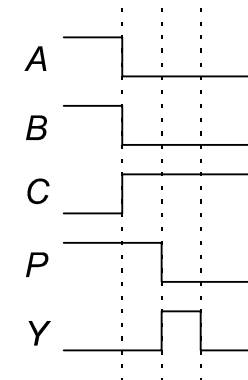
Przykład dla prawa łączności iloczynu: $A \cdot B \cdot C = (A \cdot B) \cdot C$



Układ (prawie) bez hazardu dla wszystkich przejść.



Układ z hazardem statycznym np. dla przejścia $ABC = 110 \rightarrow 001$. Brak hazardu przy przejściach między stanami sąsiednimi logicznie.



Jeżeli dopuszczono przejście jednoczesne na dwóch lub więcej wejściach złożonego bloku kombinacyjnego, to trzeba rozważyć także konsekwencje wszelkich przejść przesuniętych w czasie. Przykładowo, założone przejście $q_2q_1 = 01 \rightarrow 10$, w rzeczywistym układzie może być realizowane na trzy sposoby:

- bezpośrednio: $01 \rightarrow 10$,
- pośrednio: $01 \rightarrow 00 \rightarrow 10$,
- pośrednio: $01 \rightarrow 11 \rightarrow 10$.

3.3. Projektowanie asynchronicznych układów sekwencyjnych metodą tablic przejść/wyjść (Huffmana)

Typowe etapy projektowania:

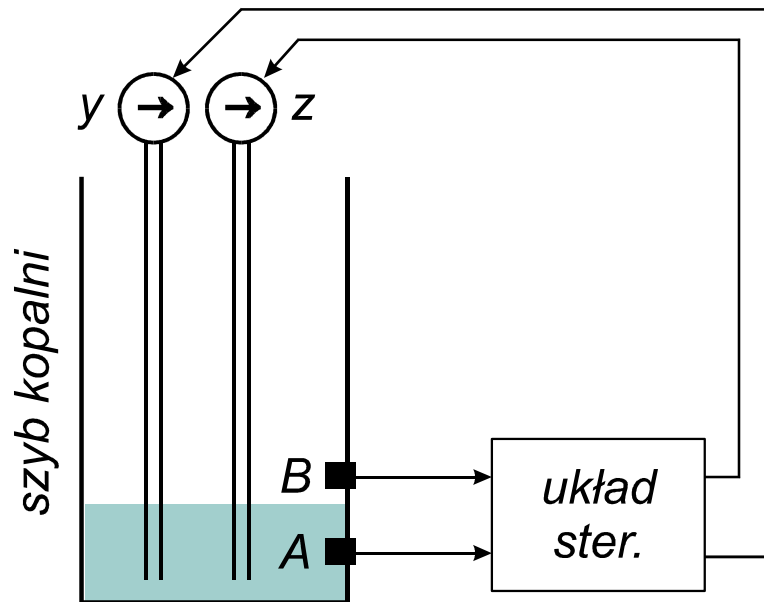
1. Analiza treści zadania i dodatkowe **założenia projektowe**.
2. Sporządzenie **listy stanów wewnętrznych** automatu.
3. Przedstawienie przejść między stanami automatu w formie **grafu przejść**.
4. Zapisanie **pierwotnej tablicy przejść** między stanami automatu Moore'a.
5. Wyszukanie i połączenie **stanów równoważnych**.
6. Wyszukanie i połączenie **stanów zgodnych**. Decyzja o połączeniu takich stanów oznacza przejście od automatu o strukturze Moore'a do automatu Mealy'ego.
7. Zapisanie **zredukowanej tablicy przejść** oraz grafu dla stanów zredukowanych.
8. **Kodowanie stanów wewnętrznych** poprzez binarne stany wewnętrzne q_1, q_2, q_3, \dots .
9. Zapisanie **zakodowanych tablic przejść** dla każdego z binarnych stanów wewnętrznych q_1, q_2, q_3, \dots .
10. **Synteza funkcji przejść** δ kolejno dla poszczególnych binarnych stanów wewnętrznych q_1, q_2, q_3, \dots . Rozważenie i eliminacja ryzyka wynikającego z hazardu w układzie kombinacyjnym realizującym funkcję przejść.
11. **Synteza funkcji wyjść** λ . W zależności od przeznaczenia automatu trzeba podjąć decyzję o dopuszczeniu albo eliminacji **hazardu** w układzie wyjściowym.
12. Sporządzenie **schematu** układu.

Opis słowny układu sekwencyjnego

W kopalni pracują dwie pompy wydobywające wodę z szybu. Poziom wody sygnalizowany jest przez dwa czujniki A i B . Czujnik znajdujący się powyżej poziomu wody wysyła sygnał o wartości 0, natomiast sygnał 1 oznacza zanurzenie w wodzie.

Zaprojektować układ, który steruje pompami y i z według następujących reguł:

- jeżeli poziom wody poniżej czujnika A ($A = B = 0$) – żadna pompa nie pracuje,
- jeżeli poziom wody pomiędzy czujnikami A i B ($A = 1, B = 0$) – pracuje ta pompa, która ostatnio nie pracowała,
- jeżeli poziom wody powyżej czujnika B ($A = B = 1$) – pracują obie pompy.



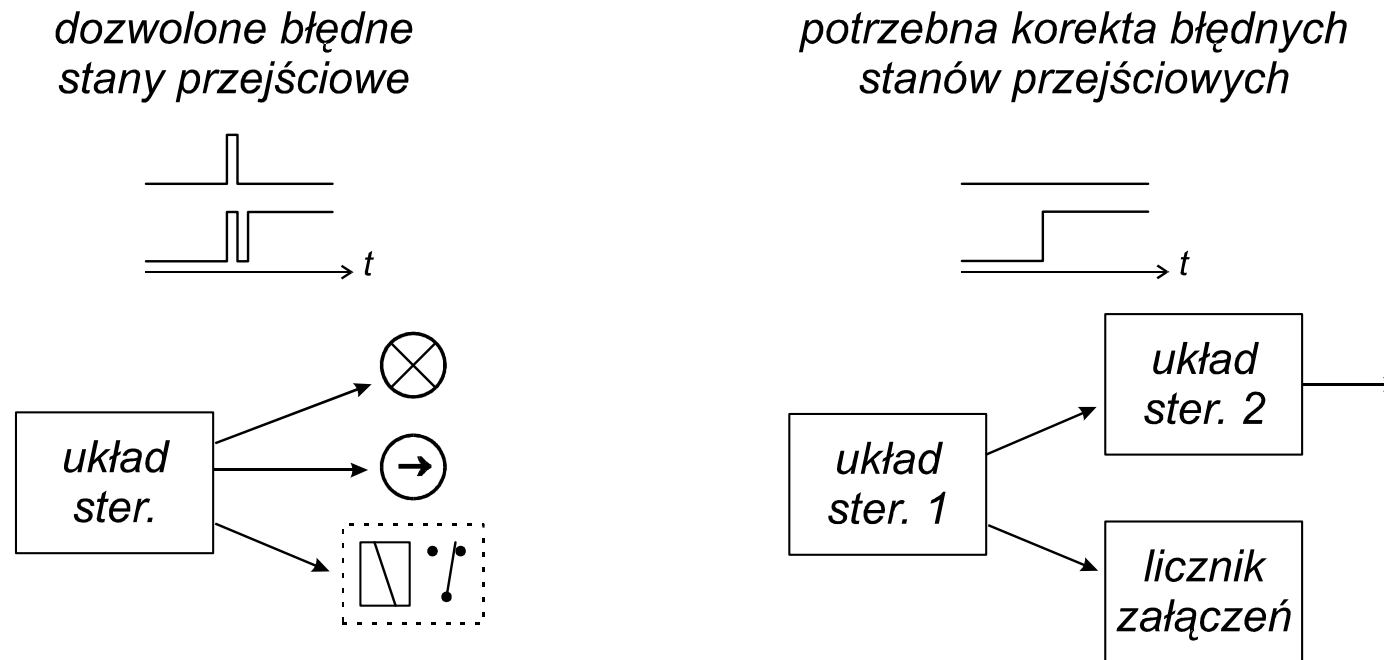
Rys. 3.4. Schemat szybu kopalni z pompami i układem sterowania.

Krok 1. Dodatkowe założenia projektowe

Opis słowny układu nie zawsze zapewnia projektantowi pełne i jednoznaczne dane.

Przyjmujemy założenia:

- nie rozważamy stanów awaryjnych czujników,
- nie określamy, która pompa powinna pracować jako pierwsza po załączeniu układu,
- zmiany stanów czujników wody *A* i *B* zachodzą wolno w porównaniu do szybkości pracy układu sterowania – nie rozważamy zmian jednoczesnych na wejściach układu.
- likwidacja błędnych krótkotrwałych stanów przejściowych na wyjściach układu sterowania nie jest konieczna, gdyż sterowane pompy są układami o dużej bezwładności.



Rys. 3.5. Przykłady odbiorników niewrażliwych oraz wrażliwych na błędne stany przejściowe.

Krok 2. Sporządzenie listy stanów wewnętrznych automatu

Jeżeli stanów wewnętrznych nie można wskazać bezpośrednio na podstawie opisu słownego, to zwykle przyjmujemy początkowo, że każda dozwolona kombinacja stanów wejściowych i wyjściowych odpowiada odrębnemu stanowi wewnętrznemu.

	AB			
yz	00	01	11	10
00	✓			
01				✓
11			✓	
10				✓

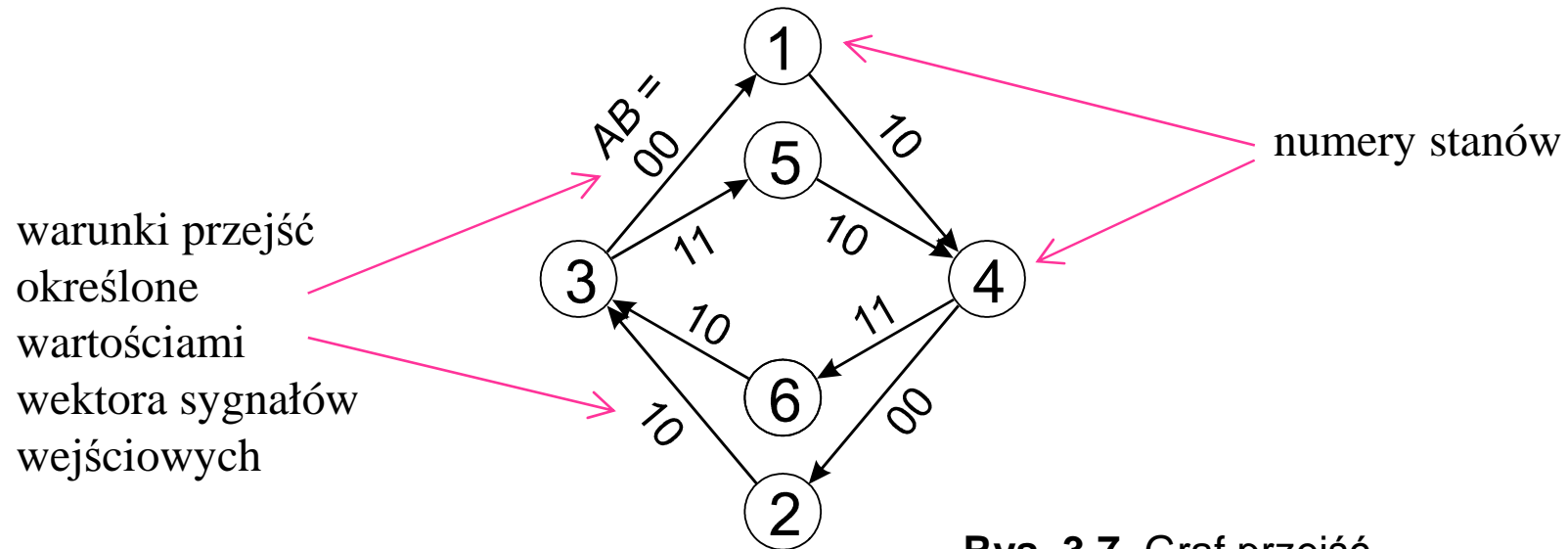
Rys. 3.6. Możliwe kombinacje stanów wejściowych i wyjściowych.

Ponadto trzeba odróżnić stany charakteryzujące się takimi samymi kombinacjami wejść i wyjść ale różne pod względem koniecznej pamięci o wcześniejszych zdarzeniach.

Przyjęta lista pierwotnych stanów wewnętrznych:

- 1) żadna pompa nie pracuje, poprzednio pracowała y,
- 2) żadna pompa nie pracuje, poprzednio pracowała z,
- 3) pracuje pompa y,
- 4) pracuje pompa z,
- 5) obie pompy pracują, poprzednio nie pracowała pompa z,
- 6) obie pompy pracują, poprzednio nie pracowała pompa y.

Krok 3. Przedstawienie przejść między stanami automatu w formie grafu przejść



Rys. 3.7. Graf przejść.

Przyjęta lista pierwotnych stanów wewnętrznych:

- 1) żadna pompa nie pracuje, poprzednio pracowała y,
- 2) żadna pompa nie pracuje, poprzednio pracowała z,
- 3) pracuje pompa y,
- 4) pracuje pompa z,
- 5) obie pompy pracują, poprzednio nie pracowała pompa z,
- 6) obie pompy pracują, poprzednio nie pracowała pompa y.

Krok 4. Zapisanie pierwotnej tablicy przejść i wyjść.

4.a). Zapisanie stanów stabilnych i niemożliwych

		AB				yz
		00	01	11	10	
stan	1	①	-			00
	2	②	-			00
	3		-		③	10
	4		-		④	01
	5		-	⑤		11
	6		-	⑥		11

4.b). Dodanie stanów niestabilnych na przejściach między stanami stabilnymi.

		AB				yz
		00	01	11	10	
stan	1	①	-	-	4	00
	2	②	-	-	3	00
	3	1	-	5	③	10
	4	2	-	6	④	01
	5	-	-	⑤	4	11
	6	-	-	⑥	3	11

stany wejść uporządkowane wg kodu Graya

przykład przejścia przez stan niestabilny

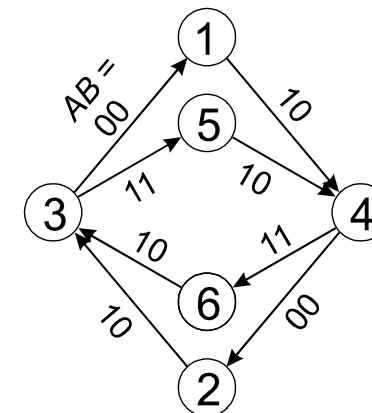
stany wyjść (opcjonalnie)

numery stanów wewnętrznych

Rys. 3.8. Pierwotna tablica przejść i wyjść.

Przyjęta lista pierwotnych stanów wewnętrznych:

- 1) żadna pompa nie pracuje, poprzednio pracowała y,
- 2) żadna pompa nie pracuje, poprzednio pracowała z,
- 3) pracuje pompa y,
- 4) pracuje pompa z,
- 5) obie pompy pracują, poprzednio nie pracowała pompa z,
- 6) obie pompy pracują, poprzednio nie pracowała pompa y.



Krok 5. Wyszukanie i połączenie stanów równoważnych.

stan	AB				yz
	00	01	11	10	
1	①	—	—	4	00
2	②	—	—	3	00
3	1	—	5	③	10
4	2	—	6	④	01
5	—	—	⑤	4	11
6	—	—	⑥	3	11

Stany 1 i 2 mogą być równoważne **pod warunkiem**, że stany 3 i 4 są także równoważne.

Stany 3 i 4 nie mogą zostać uznane za równoważne bezwarunkowo ani warunkowo, bo odpowiadają innym stanom wyjść.

Rys. 3.9. Poszukiwanie stanów równoważnych w tablicy przejść i wyjść.

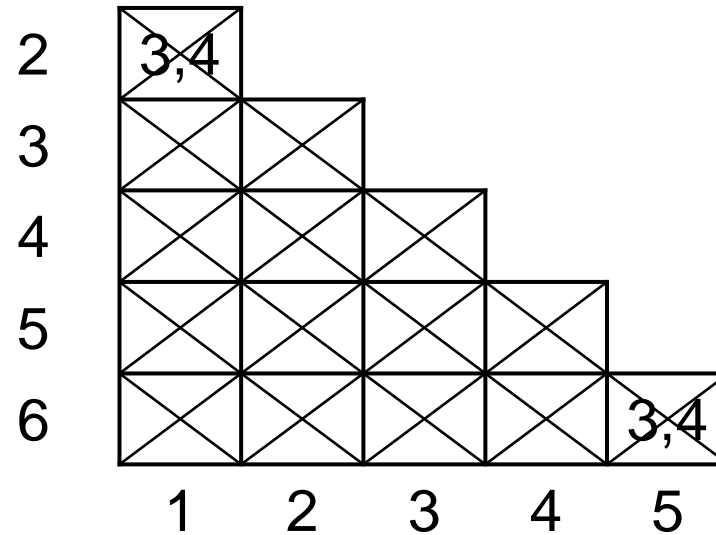
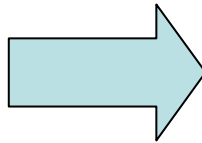
Wniosek: w rozważanej tabeli nie ma stanów równoważnych.

Dwa stany są stanami równoważnymi jeżeli w odpowiadających im wierszach pierwotnej tablicy przejść:

- stany stabilne znajdują się w jednakowych kolumnach,
- stany wyjść w obu wierszach są jednakowe,
- przy dowolnej zmianie stanu wejść układ przechodzi z rozpatrywanych stanów do stanów jednakowych lub równoważnych,
- w miejsce (—) możemy wpisać cokolwiek jeżeli umożliwi to uznanie stanów za równoważne.

Aby nie pominąć żadnej pary stanów, sprawdzanie równoważności stanów można prowadzić na tablicy trójkątnej.

		AB				
		00	01	11	10	yz
stan	1	①	—	—	4	00
	2	②	—	—	3	00
	3	1	—	5	③	10
	4	2	—	6	④	01
	5	—	—	⑤	4	11
	6	—	—	⑥	3	11



Rys. 3.10. Tablica trójkątna automatu.

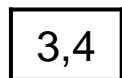
Etapy sprawdzania:



- początkowo wszystkie pola tablicy trójkątnej są puste,



- skreślamy pary stanów o niezgodnych wyjściach,



- wpisujemy warunki równoważności, np. stany 1 i 2 są równoważne pod warunkiem, że stany 3 i 4 okażą się równoważne,

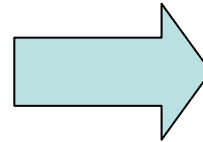


- skreślamy jeśli warunek okaże się fałszywy,

Jeśli pozostały komórki bez skreślenia, to istnieją stany równoważne bezwarunkowo lub warunkowo.

Przykład równoważności warunkowej cyklicznej

	x		
	0	1	y
1	1	2	0
2	3	2	1
3	3	4	0
4	1	4	1



2	X		
3	2,4 1,3	X	
4	X		1,3 2,4
	1	2	3

Rys. 3.11. Przykładowa tablica przejść i wyjść.

Rys. 3.12. Przykładowa tablica trójkątna.

W rozważanym przykładzie warunkiem równoważności pary stanów 1 oraz 3 jest równoważność stanów 2 oraz 4, jednakże warunkiem równoważności stanów 2 oraz 4 jest równoważność stanów 1 oraz 3.

Jeżeli mamy do czynienia z zapętlonym warunkiem równoważności stanów, to równoważność zachodzi.

Krok 6. Wyszukanie i połączenie stanów zgodnych.

stan	AB				yz
	00	01	11	10	
1	①	–	–	4	00
2	②	–	–	3	00
3	1	–	5	③	10
4	2	–	6	④	01
5	–	–	⑤	4	11
6	–	–	⑥	3	11

→ stany 1 i 5 są zgodne
→ stany 2 i 6 są zgodne

Uwaga: decyzja o połączeniu stanów zgodnych odpowiadających różnym stanom wyjść odpowiada przejściu od automatu o strukturze Moore'a do automatu Mealy'ego.

Rys. 3.13. Poszukiwanie stanów zgodnych w tablicy przejść.

Dwa stany są stanami zgodnymi jeżeli w odpowiadających im wierszach pierwotnej tablicy przejść:

- stany z jednakowej kolumny mają takie same numery – zgodność ma być bezwarunkowa,
- w miejsce (–) możemy wpisać cokolwiek jeżeli umożliwi to uznanie stanów za zgodne.

	AB			
	00	01	11	10
1	①	2	–	–
2	1	②	–	3
...

1,2 → 1'
 3 → 2'

	AB			
	00	01	11	10
1'	①'	①'	–	2'
...

Rys. 3.14. Przykład łączenia stanu stabilnego i niestabilnego w jednej kolumnie. Po połączeniu zapisuje się stan stabilny.

Krok 7. Zapisanie zredukowanej tablicy przejść oraz grafu dla stanów zredukowanych.

stan	AB				yz
	00	01	11	10	
1	①	-	-	4	00
2	②	-	-	3	00
3	1	-	5	③	10
4	2	-	6	④	01
5	-	-	⑤	4	11
6	-	-	⑥	3	11

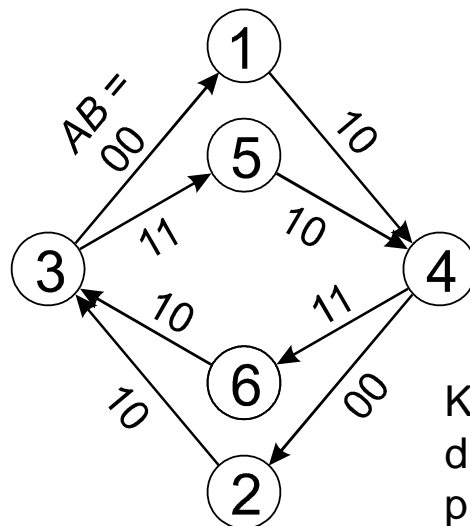
Kopia pierwotnej tablica przejść.

stan	AB			
	00	01	11	10
1,5	①	-	⑤	4
3	1	-	5	③
2,6	②	-	⑥	3
a) 4	2	-	6	④

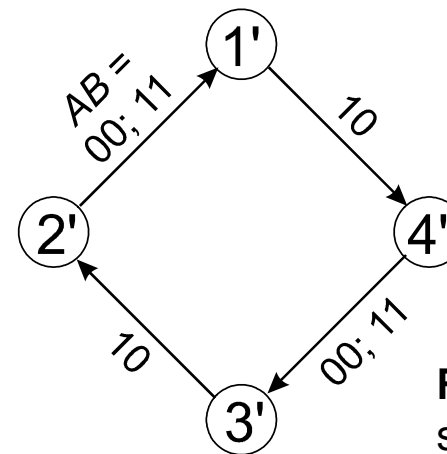
1,5 → 1'
3 → 2'
2,6 → 3'
4 → 4'

stan	AB			
	00	01	11	10
1'	①'	-	①'	4'
2'	1'	-	1'	②'
3'	③'	-	③'	2
b) 4'	3'	-	3'	④'

Rys. 3.15. Zredukowana tablica przejść: (a) notacja ze stanami pierwotnymi, (b) notacja ze stanami zredukowanymi.



Kopia grafu przejść dla stanów pierwotnych



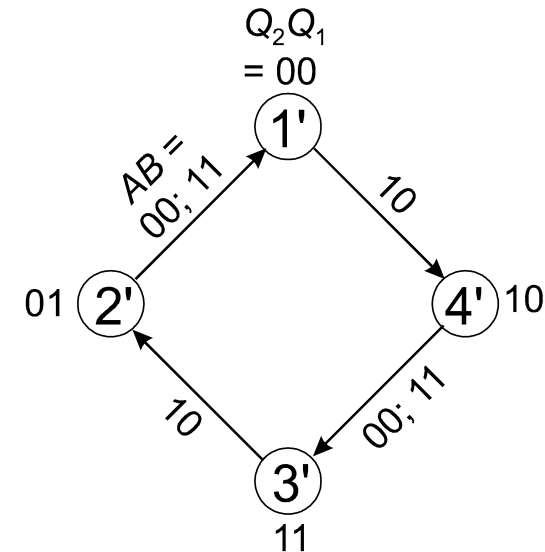
Rys. 3.16. Graf dla stanów zredukowanych.

Krok 8. Kodowanie stanów wewnętrznych poprzez binarne stany wewnętrzne q_1, q_2, q_3 .

Kodowanie stanów to przyporządkowanie symbolicznym oznaczeniom stanów (liczby dziesiętne, litery) wartości binarnych stanów linii sprzężenia zwrotnego q_1, q_2, q_3, \dots . Dla liczby stanów zredukowanych $\leq 2^n$ wystarczy n sygnałów binarnych, o ile nie zachodzi konieczność zwiększenia liczby sygnałów dla uniknięcia **wyścigu krytycznego**.

Tabela 3.1. Optymalne kodowanie stanów automatu.

stany przed zakodowaniem	stany po zakodowaniu	
	Q_2	Q_1
1'	0	0
2'	0	1
3'	1	1
4'	1	0



Rys. 3.17. Graf przejść z uwzględnieniem kodowania stanów.

W rozważanym przykładzie wszystkie przejścia mogą odbywać się między **stanami sąsiednimi logicznie**, tzn. takimi które różnią się tylko jednym bitem.

Przypadki bardziej złożonych przejść zostaną omówione dalej w innym przykładzie.

Krok 9. Zapisanie zakodowanych tablic przejść dla każdego z binarnych stanów wewnętrznych q_1, q_2, q_3, \dots .

		AB			
		00	01	11	10
stan					
1'		①'	–	①'	4'
2'		1'	–	1'	②'
3'		③'	–	③'	2
4'		3'	–	3'	④'

Kopia zredukowanej tablicy przejść.

stany przed zakodowaniem	stany po zakodowaniu	
	Q_2	Q_1
1'	0	0
2'	0	1
3'	1	1
4'	1	0

Kopia tablicy kodowania stanów.



		AB			
		00	01	11	10
q_2q_1					
00		00	–	00	10
01		00	–	00	01
11		11	–	11	01
10		11	–	11	10

Q_2Q_1

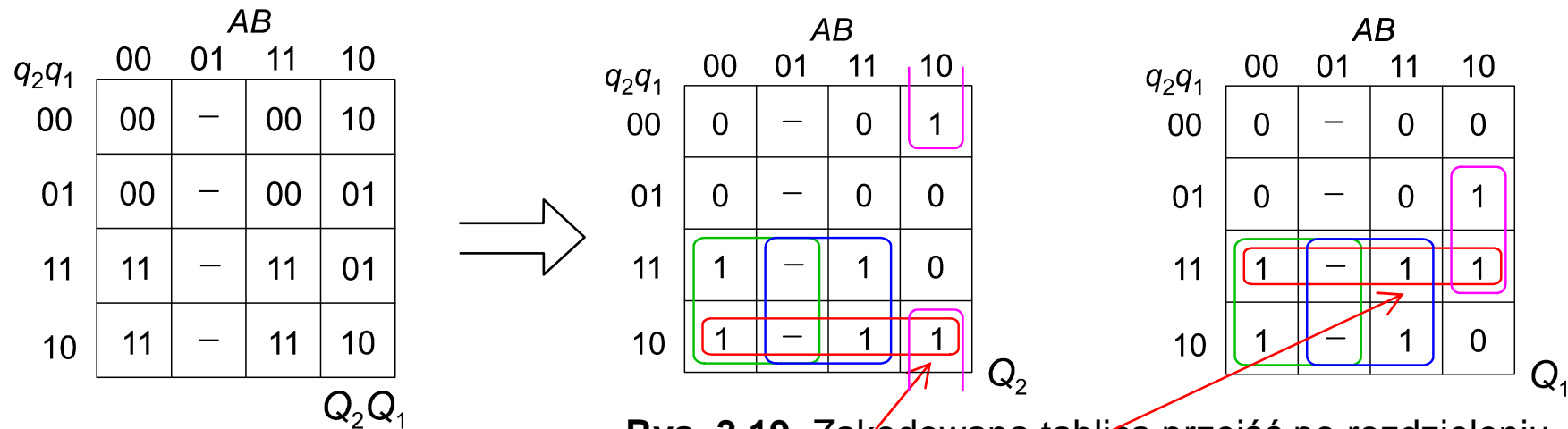
Rys. 3.18. Zakodowana tablica przejść.

Użyte oznaczenia:

$$q_k \equiv q_k^t \quad (3.11)$$

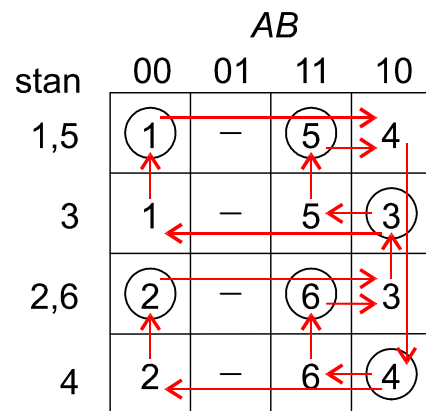
$$Q_{k'} \equiv q_k^{t+1}$$

Krok 10. Synteza funkcji przejść δ kolejno dla poszczególnych binarnych stanów wewnętrznych q_1, q_2, q_3, \dots



Kopia zakodowanej tablicy przejść.

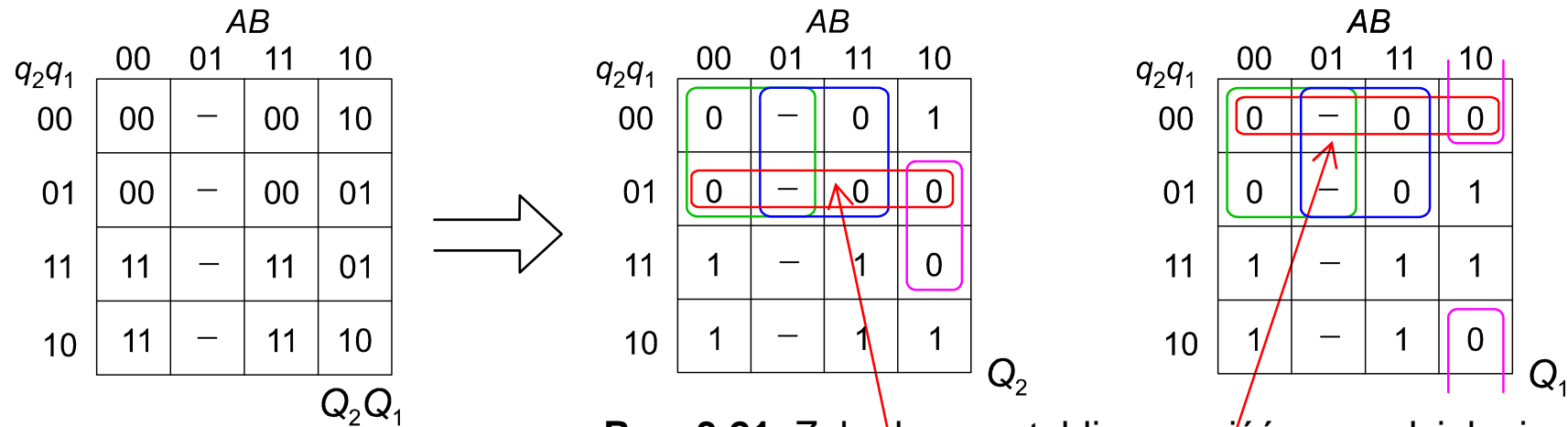
Rys. 3.19. Zakodowana tablica przejść po rozdzieleniu na tablice Karnauga dla poszczególnych wewnętrznych sygnałów binarnych.



Rys. 3.20. Graf przejść naniesiony na tablicę przejść. Notacja z numerami stanów pierwotnych.

Pętle czerwone wprowadzono w celu eliminacji **ryzyka** związanego z różnicami w czasach propagacji sygnałów po różnych drogach w wejściowym bloku kombinacyjnym. Bez dodatkowych pętli występują **hazardowe** przekłamanie stanów wewnętrznych, które mogą się utrwalić w kolejnym cyklu. W układzie wolnym od ryzyka wszystkie możliwe przejścia między „1” powinny odbywać się w jednej pętli.

c.d. syntezy funkcji przejść δ - przypadek łączenia zer na tablicach Karnaugh.



Kopia zakodowanej tablicy przejść.

Rys. 3.21. Zakodowana tablica przejść po rozdeleniu na tablice Karnaugh dla poszczególnych wewnętrznych sygnałów binarnych.

W przypadku łączenia zer na tablicach Karnaugh wszystkie możliwe przejścia między „0” powinny odbywać się w jednej pętli. W przeciwny razie układ będzie zagrożony **ryzykiem** niejednoznacznego działania.

Do dalszych przekształceń przyjmiemy przypadek grupowania jedynek w tablicach przejść. W rozważanym przykładzie grupowanie zer prowadzi do wyrażeń o takim samym stopniu złożoności.

Krok 11. Synteza funkcji wyjść λ .

Wariant 1 – nadmiarowe krótkie impulsy na wyjściu są niedozwolone

11.a). Zapisanie stanów stabilnych i niemożliwych

		AB			
		00	01	11	10
q_2q_1	00	(00)	–	(11)	
	01		–		(10)
	11	(00)	–	(11)	
	10		–		(01)

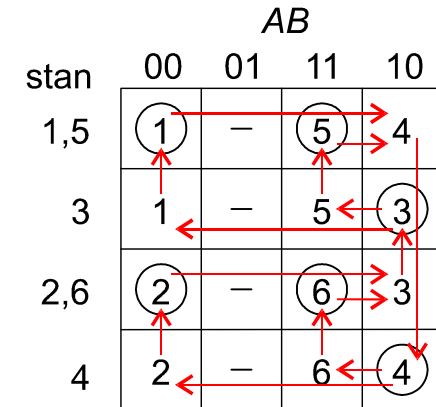
yz

Rys. 3.22. Tablica wyjść.

11.b). Dodanie stanów niestabilnych na przejściach między stanami stabilnymi.

		AB			
		00	01	11	10
q_2q_1	00	(00)	–	(11)	01
	01	-0	–	1-	(10)
	11	(00)	–	(11)	10
	10	0-	–	-1	(01)

yz



Kopia grafu przejść na tablicy przejść.

Sygnaly wyjściowe yz dla stanów niestabilnych określamy następująco:

- Jeżeli sygnaly odpowiadające początkowemu i końcowemu stanowi stabilnemu są jednakowe, to taki sam sygnal przypisujemy pośredniemu stanowi niestabilnemu.
- Jeżeli sygnaly odpowiadające początkowemu i końcowemu stanowi stabilnemu są różne, to pośredniemu stanowi niestabilnemu przypisujemy sygnal dowolny (–).
- Niektóre komórki mogą uczestniczyć w przejściach od różnych stabilnych stanów początkowych. Jeżeli różne przejścia prowadzą do innych wniosków, to wybieramy stan zgodny z docelowym stanem stabilnym (zamiast „–”).

		AB			
		00	01	11	10
q ₂ q ₁	00	0	–	0	1
	01	0	–	0	0
	11	1	–	1	0
	10	1	–	1	1

Q₂

		AB			
		00	01	11	10
q ₂ q ₁	00	0	–	0	0
	01	0	–	0	1
	11	1	–	1	1
	10	1	–	1	0

Q₁

a). Tablice przejść,

brak wyrażeń wspólnych
dla tablic przejść oraz wyjść

Wariant 1

		AB			
		00	01	11	10
q ₂ q ₁	00	0	–	1	0
	01	–	–	1	1
	11	0	–	1	1
	10	0	–	–	0

y

		AB			
		00	01	11	10
q ₂ q ₁	00	0	–	1	1
	01	0	–	–	0
	11	0	–	1	0
	10	–	–	1	1

z

b). Tablice wyjść

Rys. 3.23. Poszukiwanie wspólnych wyrażeń
w funkcjach przejść i funkcjach wyjść.

$$Q_1 = q_2 \bar{A} + q_2 B + q_1 A \bar{B} + \underline{q_1 q_2},$$

$$Q_2 = q_2 \bar{A} + q_2 B + \bar{q}_1 A \bar{B} + \underline{\bar{q}_1 q_2},$$

wyraży anty hazardowe
odpowiadające
czerwonym pętlom

$$y = B + q_1 \bar{B},$$

$$z = B + \bar{q}_1 \bar{B}.$$

Niedozwolone zwężenie liliowych
pętli w celu otrzymania wspólnych
wyrażeń, bo może to spowodować
błędne stany przejściowe na
wyjściach xy dla niektórych
dozwolonych przejść, np.
 $q_2 q_1 AB = 1010 \rightarrow 1011.$

Krok 11. Synteza funkcji wyjść λ .

Wariant 2 – dozwolone nadmiarowe krótkie impulsy na wyjściu

11.a). Zapisanie stanów stabilnych i niemożliwych

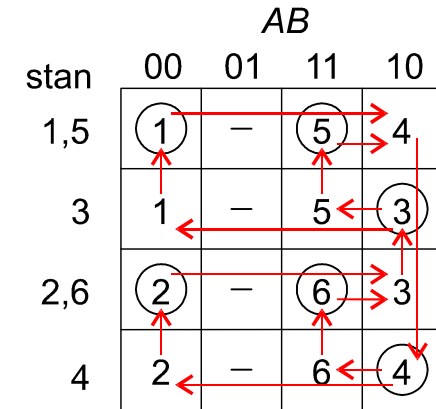
		AB			
		00	01	11	10
q_2q_1	00	(00)	–	(11)	
	01		–		(10)
	11	(00)	–	(11)	
	10		–		(01)

yz

11.b). Dodanie stanów niestabilnych na przejściach między stanami stabilnymi.

		AB			
		00	01	11	10
q_2q_1	00	(00)	–	(11)	--
	01	--	–	--	(10)
	11	(00)	–	(11)	--
	10	--	–	--	(01)

yz



Rys. 3.24. Tablica wyjść.

Kopia grafu przejść na tablicy przejść.

We wszystkie kratki odpowiadające stanom niestabilnym wpisujemy sygnał dowolny „–” dla obu wyjść.

		AB			
		00	01	11	10
q ₂ q ₁	00	0	-	0	1
	01	0	-	0	0
	11	1	-	1	0
	10	1	-	1	1

		AB			
		00	01	11	10
q ₂ q ₁	00	0	-	0	0
	01	0	-	0	1
	11	1	-	1	1
	10	1	-	1	0

a). Tablice przejść,

$$Q_1 = q_2 \bar{A} + q_2 B + q_1 A \bar{B} + \underline{q_1 q_2},$$

$$Q_2 = q_2 \bar{A} + q_2 B + \bar{q}_1 A \bar{B} + \underline{\bar{q}_1 q_2},$$

wyraży antyazardowe
odpowiadające
czerwonym pętlom

wspólne wyrażenia

Wariant 2

		AB			
		00	01	11	10
q ₂ q ₁	00	0	-	1	-
	01	-	-	-	1
	11	0	-	1	-
	10	-	-	-	0

		AB			
		00	01	11	10
q ₂ q ₁	00	0	-	1	-
	01	-	-	-	0
	11	0	-	1	-
	10	-	-	-	1

b). Tablice wyjść

$$y = B + q_1 A \bar{B},$$

$$z = B + \bar{q}_1 A \bar{B}.$$

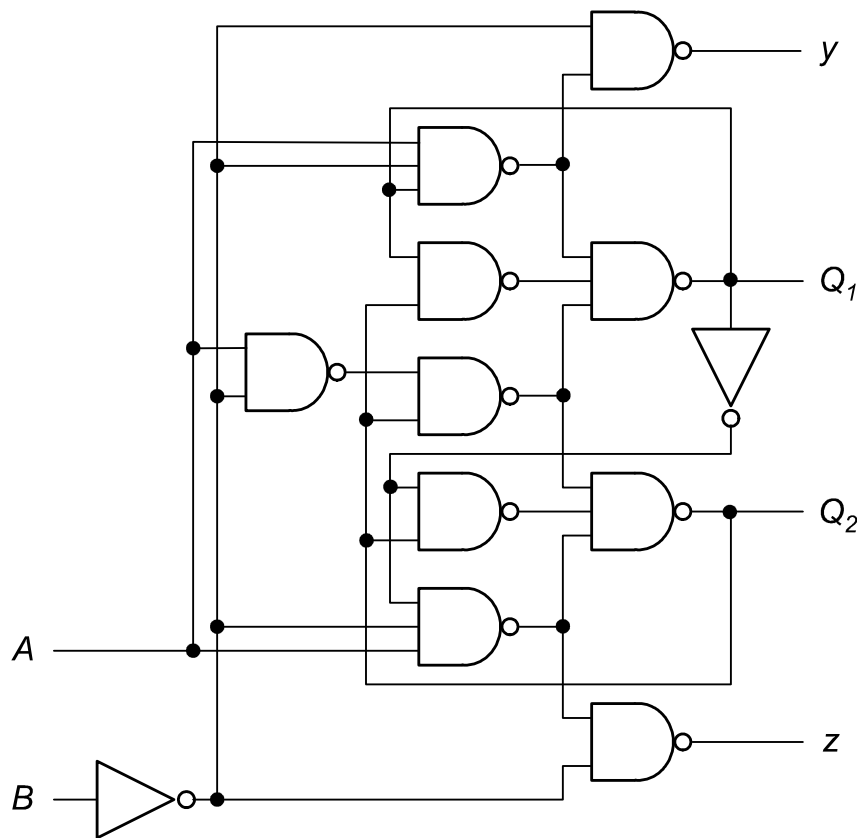
możliwe rozszerzenie liliowych pętli na tablicach wyjść ale to jest nieopłacalne, bo spowoduje utratę wspólnych wyrażeń.

Rys. 3.25. Poszukiwanie wspólnych wyrażeń w funkcjach przejść i funkcjach wyjść.

Krok 12. Sporządzenie schematu układu.

Założenie 1: układ zostanie zrealizowany przy wykorzystaniu bramek NAND i NOT.

Założenie 2: dopuszczamy hazard w wyjściowym ukl. Kombinacyjnym (wariant 2).



Rys. 3.26. Schemat układu sterowania pompami w szybie kopalni.

$$\begin{aligned}
 Q_1 &= q_2 \bar{A} + q_2 B + q_1 \bar{A} \bar{B} + q_1 q_2 = \\
 &= q_2 (\bar{A} + B) + q_1 \bar{A} \bar{B} + q_1 q_2 = \\
 &= \overline{\overline{q_2 \bar{A} B} \overline{q_1 \bar{A} \bar{B}} \overline{q_1 q_2}}, \quad (3.12)
 \end{aligned}$$

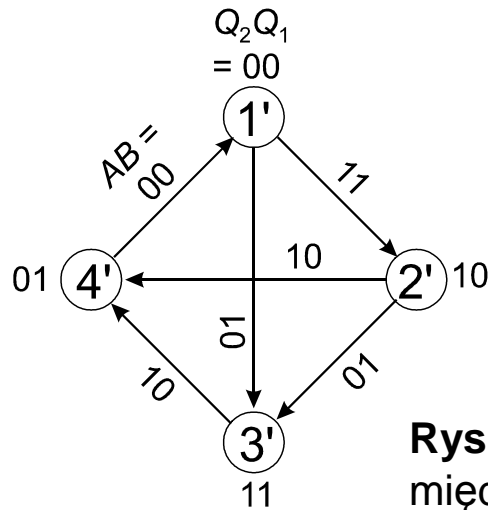
$$\begin{aligned}
 Q_2 &= q_2 \bar{A} + q_2 B + \bar{q}_1 \bar{A} \bar{B} + \bar{q}_1 q_2 = \\
 &= q_2 (\bar{A} + B) + \bar{q}_1 \bar{A} \bar{B} + \bar{q}_1 q_2 = \\
 &= \overline{\overline{q_2 \bar{A} B} \overline{\bar{q}_1 \bar{A} \bar{B}} \overline{\bar{q}_1 q_2}}, \quad (3.13)
 \end{aligned}$$

$$y = \overline{\overline{B} \overline{q_1 \bar{A} B}}, \quad (3.14)$$

$$z = \overline{\overline{B} \overline{q_1 \bar{A} B}}. \quad (3.15)$$

podkreślono powtarzające się wyrażenia

Przykład asynchronicznego układu sekwencyjnego z koniecznym hazardem



Żadne trzy kody binarne nie są sąsiednie logicznie, zatem dla przedstawionego grafu nie można całkowicie wyeliminować przejść między niesąsiednimi logicznie stanami wewnętrznymi.

Rys. 3.27. Przykład grafu z przejściami między stanami niesąsiednimi logicznie.

		AB			
		00	01	11	10
stan	q_2q_1	00	01	11	10
	1'	1' → 3'	2' → 3'	2' → 3'	-
	2'	1' → 3'	3' ← 2'	2' → 4'	4' ← 2'
	3'	1' → 3'	3' ← 2'	-	4' ← 2'
4'	1' → 3'	3' ← 2'	-	4' ← 2'	

a). Błędny projekt! Zagrożenie wyścigiem krytycznym, tzn. może istnieć droga prowadząca do błędnego stanu stabilnego.

		AB			
		00	01	11	10
stan	q_2q_1	00	01	11	10
	1'	1' → 3'	2' → 3'	2' → 3'	4' ← 2'
	2'	1' → 3'	3' ← 2'	2' → 4'	4' ← 2'
	3'	1' → 3'	3' ← 2'	-	4' ← 2'
4'	1' → 3'	3' ← 2'	-	4' ← 2'	

b). Dobrze. Dopuszczono tylko wyścig niekrytyczny, tzn. wszystkie możliwe drogi prowadzą do poprawnego stanu stabilnego.

Rys. 3.28. Tablica przejść dla układu o grafie przedstawionym na rys. 3.21.

3.4. Literatura

- [1] P. Misiurewicz, *Układy automatyki cyfrowej*, Wydawnictwa Szkolne i Pedagogiczne, Warszawa, 1984.
- [2] H. Kamionka-Mikuła, H. Małysiak, B. Pochopień, *Synteza i analiza układów cyfrowych*, Wydawnictwo Pracowni Komputerowej Jacka Skamierskiego, Gliwice 2006.
- [3] C. Zieliński, *Podstawy projektowania układów cyfrowych*, PWN, Warszawa 2003.
- [4] W. Traczyk, *Układy cyfrowe. Podstawy teoretyczne i metody syntezy*, WNT, Warszawa 1986.
- [5] A. Skorupski, *Podstawy techniki cyfrowej*, WKiŁ, Warszawa 2004.
- [6] J. Kalisz, *Podstawy elektroniki cyfrowej*, WKiŁ, Warszawa 2002.
- [7] A. Barczak, J. Florek, T. Sydoruk, *Elektroniczne techniki cyfrowe*, VIZJA PRESS&IT Sp. z o.o., Warszawa 2006.