Przykładowy projekt automatu – komparator szeregowy

- Przyjmuje szeregowo bity dwóch słów w NKB od najmniej znaczącego
- Bity pojawiają się wraz z zegarem
- Na wyjściu daje "większy", "mniejszy" lub "równy" w kodzie 1 z 3 dla dotychczasowych bitów
- Automat Moore'a o 2 wejściach i 3 wyjściach

Stany (automat Moore'a)

S_1	$a=b$	OUT	001
S_2	$a>b$	OUT	100
S_3	$a<b$	OUT	010

Graf automatu

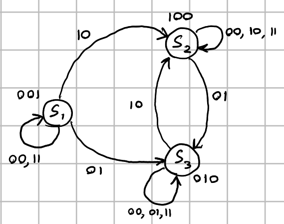


Tabela przejść i wyjść

S \ ab	00	01	11	10	WMR
S_1	S_1	S_3	S_1	S_2	001
S_2	S_2	S_3	S_2	S_2	100
S_3	S_3	S_3	S_3	S_2	010

Zakodowana tabela przejść i wyjść

Trzeba wykorzystać tyle bitów na reprezentację stanu, żeby każdy miał unikalny kod

$S_1 \to 00$ Przedstawiam stan na 2 bitach Q_1Q_0

$S_2 \to 01$

$S_3 \to 10$

11 nie jest używane ale będzie wykorzystany przy realizacji

Q_1Q_0 \ ab	00	01	11	10	WMR
00	00	10	00	01	001
01	01	10	01	01	100
10	10	10	10	01	010

Uszeregowana tabela – zgodnie z kodem Graya

Q_1Q_0 \ ab	00	01	11	10	WMR
00	00	10	00	01	001
01	01	10	01	01	100
11	–	–	–	–	–
10	10	10	10	01	010

Automat można zrealizować na różnych typach przerzutników

Wzbudzenia przerzutników D

najprostsza realizacja ale nie najtańsza

Q_1Q_0 \ ab	00	01	11	10	WMR
00	00	10	00	01	001
01	01	10	01	01	100
11	–	–	–	–	–
10	10	10	10	01	010

D_1:

Q_1Q_0 \ ab	00	01	11	10
00	0	1	0	0
01	0	1	0	0
11	–	–	–	–
10	1	1	1	0

D_0:

Q_1Q_0 \ ab	00	01	11	10
00	0	0	0	1
01	1	0	1	1
11	–	–	–	–
10	0	0	0	1

$D_1 = \bar{a}b + Q_1\bar{a} + Q_1 b$ $D_0 = Q_0\bar{b} + Q_0 a + a\bar{b}$

Dla D_1 przepisuję bit Q_1, a dla D_0 przepisuję bit Q_0,
ogólnie do D_i idzie Q_i.
Każdą funkcję minimalizuję na mapie Karnaugha

Tabela wyjść

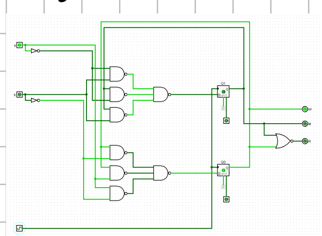
Q_1Q_0	W	M	R
00	0	0	1
01	1	0	0
11	–	–	–
10	0	1	0

$W = Q_0$

$M = Q_1$

$R = \bar{Q_1}\bar{Q_0}$

Realizacja



Inny typ przerzutnika może dać tańszą realizację

Wzbudzenia przerzutników T

Q_1Q_0 \ ab	00	01	11	10	WMR
00	00	**10**	00	**01**	001
01	01	**10**	01	01	100
11	–	–	–	–	–
10	10	10	10	**01**	010

zaznaczam bity które się zmieniają

Q^t	→	Q^{t+1}	T
0	→	0	0
0	→	**1**	1
1	→	**0**	1
1	→	1	0

Przepis dla T

1: dla pogrubionych

0: dla niepogrubionych

T_1:

Q_1Q_0 \ ab	00	01	11	10
00	0	1	0	0
01	0	1	0	0
11	–	–	–	–
10	0	0	0	1

T_0:

Q_1Q_0 \ ab	00	01	11	10
00	0	0	0	1
01	0	1	0	0
11	–	–	–	–
10	0	0	0	1

$$T_1 = \overline{Q}_1 \bar{a} b + Q_1 a \bar{b}$$

$$T_0 = Q_0 \bar{a} b + Q_0 a \bar{b}$$

Wzbudzenia przerzutników JK

Q_1Q_0 \ ab	00	01	11	10	WMR
00	00	**10**	00	**01**	001
01	01	**10**	01	01	100
11	–	–	–	–	–
10	10	10	10	**01**	010

Q^t	→	Q^{t+1}	J	K
0	→	0	0	–
0	→	**1**	1	–
1	→	**0**	–	1
1	→	1	–	0

Przepis dla JK

J:

1 dla pogrubionych jedynek

0 dla niepogrubionych zer

– dla pozostałych

K:

1 dla pogrubionych zer

0 dla niepogrubionych jedynek

– dla pozostałych

J_1:

Q_1Q_0 \ ab	00	01	11	10
00	0	1	0	0
01	0	1	0	0
11	–	–	–	–
10	–	–	–	–

K_1:

Q_1Q_0 \ ab	00	01	11	10
00	–	–	–	–
01	–	–	–	–
11	–	–	–	–
10	0	0	0	1

J_0:

Q_1Q_0 \ ab	00	01	11	10
00	0	0	0	1
01	–	–	–	–
11	–	–	–	–
10	0	0	0	1

K_0:

Q_1Q_0 \ ab	00	01	11	10
00	–	–	–	–
01	0	1	0	0
11	–	–	–	–
10	–	–	–	–

JK często daje prostszą realizację dzięki temu jak dużo '–' się powtarza

Wzbudzenia przerzutników RS

Q_1Q_0 \ ab	00	01	11	10	WMR
00	00	**10**	00	**01**	001
01	01	**10**	01	01	100
11	–	–	–	–	–
10	10	10	10	**01**	010

Q^t	→	Q^{t+1}	S	R
0	→	0	0	–
0	→	**1**	1	0
1	→	**0**	0	1
1	→	1	–	0

Przepis dla RS

S:

1 dla pogrubionych jedynek

0 dla zer

– dla niepogrubionych jedynek

R:

1 dla pogrubionych zer

0 dla jedynek

– dla niepogrubionych zer

S_1:

Q_1Q_0 \ ab	00	01	11	10
00	0	1	0	0
01	0	1	0	0
11	–	–	–	–
10	–	–	–	0

R_1:

Q_1Q_0 \ ab	00	01	11	10
00	–	0	–	–
01	–	0	–	–
11	–	–	–	–
10	0	0	0	1

S_0:

Q_1Q_0 \ ab	00	01	11	10
00	0	0	0	1
01	–	0	–	–
11	–	–	–	–
10	0	0	0	1

R_0:

Q_1Q_0 \ ab	00	01	11	10
00	–	–	–	0
01	0	1	0	0
11	–	–	–	–
10	–	–	–	0

RS najlepiej w ogóle nie używać

Zamiana typu przerzutnika

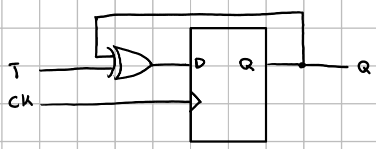
Realizacja przerzutnika T za pomocą przerzutnika D

1) Tworzę tabelę przejść szukanego przerzutnika

Q \ T	0	1
0	0	1
1	1	0

D

Realizacja



2) Znajduję funkcję wzbudzeń

$$D = T \oplus Q$$

Nie daje najtańszej realizacji ale przydaje się w sytuacjach awaryjnych

Automat zupełny (w pełni określony)

wszystkie przejścia i wyjścia są zdefiniowane

Automat niezupełny (nie w pełni określony)

przynajmniej jedno przejście lub wyjście nie jest zdefiniowane

Stany równoważne

s \ x_1x_0	00	01	11	10	y
1	4	2	1	3	0
2	4	1	2	3	0

Stany automatu zupełnego dla których
- wyjścia są identyczne
- stany następne (X - następniki) są takie same lub równoważne

Stany zgodne

s \ x_1x_0	00	01	11	10	y
1	4	2	1	–	0-
2	–	1	2	3	01

Stany automatu niezupełnego dla których
- wyjścia są niesprzeczne
- stany następne (X - następniki) są niesprzeczne lub zgodne

Minimalizacja liczby stanów

Dążymy do znalezienia równoważnego automatu o jak najmniejszej liczbie stanów

Mniej stanów → mniej przerzutników → mniej funkcji wzbudzeń → mniej bramek → mniejszy koszt

Minimalizacja przykładowego automatu

s \ x_1x_0	00	01	11	10	y
1	4	3	1	2	0
2	5	1	4	3	0
3	6	1	3	2	0
4	6	3	3	1	1
5	2	3	6	1	0
6	4	3	1	3	1

Trójkątna tabela skracania

Krok 1.

Dla każdej pary stanów (kratki) wpisuję stany zgodne warunkowo - których zgodność zależy od zgodności innych stanów

Wykreślam kratkę jeśli stany na pewno nie są zgodne - mają różne wyjścia

	1	2	3	4	5
2	4,5 1,3 1,4 2,3				
3	4,6	5,6 3,4			
4	X	X	X		
5	2,4 1,6 1,2	1,3 4,6	2,6 1,3 3,6 1,2	X	
6	X	X	X	1,3	X

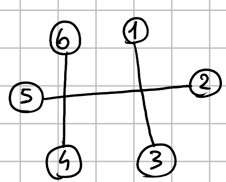
Krok 2.

Wykreślam te kratki, które zależą od pary stanów, które już wiem, że nie są zgodne

	1	2	3	4	5
2	~~4,5 1,3 1,4 2,3~~				
3	4,6	~~5,6 3,4~~			
4	X	X	X		
5	~~2,4 1,6 1,2~~	1,3 4,6	~~2,6 1,3 3,6 1,2~~	X	
6	X	X	X	1,3	X

Krok 3.

Dla pozostałych par stanów (niewykreślonych komórek) tworzę graf stanów zgodnych



Krok 4.

Znaleźć minimalny zbiór stanów zgodnych

$$\Phi_{min} = \{ \underbrace{\{1,3\}}_{A}, \underbrace{\{4,6\}}_{B}, \underbrace{\{2,5\}}_{C} \}$$

Krok 5.

Połączyć stany zgodne w tabeli przejść

s \ x_1x_0	00	01	11	10	y
1	4	3	1	2	0
2	5	1	4	3	0
3	6	1	3	2	0
4	6	3	3	1	1
5	2	3	6	1	0
6	4	3	1	3	1

→

x_1x_0	00	01	11	10	y
A	B	A	A	C	0
B	B	A	A	A	1
C	C	A	B	A	0

Tak zminimalizowany układ można już zrealizować na wybranych przerzutnikach

Przekształcenie automatu Moore'a w automat Mealy'ego

s \ x_1x_0	00	01	11	10	y
A	B	A	A	C	0
B	B	A	A	A	1
C	C	A	B	A	0

s \ x_1x_0	00	01	11	10
A	B/1	A/0	A/0	C/0
B	B/1	A/0	A/0	A/0
C	C/0	A/0	B/1	A/0

Dla każdej komórki tabeli przyporządkowuję
- następny stan w automacie Moore'a
- wyjście związane z następnym stanem w automacie Moore'a

Przekształcenie automatu Mealy'ego na automat Moore'a

s \ x	0	1
A	A/01	B/00
B	C/11	A/10
C	C/00	A/10

A/01	1
A/10	2
B/00	3
C/00	4
C/11	5

s \ x	0	1	y
1	1	3	01
2	1	3	10
3	5	2	00
4	4	2	00
5	4	2	11

Tworzę tabelę z unikalnymi parami przejść i wyjść z tabeli Mealy'ego i przypisuję im nowe nazwy
→ to będzie zbiór stanów automatu Moore'a

Dla każdego wiersza w tabeli Moore'a
- przypisuję wyjście z pomocniczej tabeli
- przypisuję przejścia do "nowych" stanów odpowiadających tym z tabeli Mealy'ego

Jeden typ automatu może dać tańszą realizację niż drugi – warto sprawdzić oba

Automat Mealy'ego często daje lepszą liczbę stanów i gorszą funkcję wyjść

Algorytm projektowania synchronicznych automatów sekwencyjnych

1. Na podstawie opisu słownego sporządzić graf automatu
2. Na podstawie grafu automatu sporządzić tabelę przejść i wyjść
3. Zminimalizować liczbę stanów
4. Zakodować tabelę przejść
5. Wyznaczyć funkcję wzbudzeń wybranych przerzutników
6. Wyznaczyć funkcję wyjść
7. Sporządzić testy automatu

Minimalizacja niezupełnego automatu Mealy'ego

Tabela przejść i wyjść $\longrightarrow$ Tabela przejść, Tabela wyjść

$s \backslash x_1x_0$	00	01	11	10
1	–	3/1	4/1	2/1
2	4/0	–	–	–
3	6/0	6/1	–	–
4	–	6/0	1/0	5/1
5	–	–	2/1	–
6	3/0	–	2/0	3/1

Tabela przejść

$s \backslash x_1x_0$	00	01	11	10
1	–	3	4	2
2	4	–	–	–
3	6	6	–	–
4	–	6	1	5
5	–	–	2	–
6	3	–	2	3

s'

Tabela wyjść

$s \backslash x_1x_0$	00	01	11	10
1	–	1	1	1
2	0	–	–	–
3	0	1	–	–
4	–	0	0	1
5	–	–	1	–
6	0	–	0	1

y

Trójkątna tabela skracania

	1	2	3	4	5
2					
3	3,6	4,6			
4	X		X		
5	2,4			X	
6	X	X ~~3,4~~	3,6	1,2 3,5	X

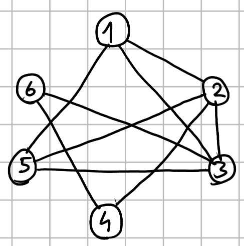
$s \backslash x_1x_0$	00	01	11	10
1	–	3	4	2
2	4	–	–	–
3	6	6	–	–
4	–	6	1	5
5	–	–	2	–
6	3	–	2	3

s'

$s \backslash x_1x_0$	00	01	11	10
1	–	1	1	1
2	0	–	–	–
3	0	1	–	–
4	–	0	0	1
5	–	–	1	–
6	0	–	0	1

y

Graf stanów zgodnych



Wybieram z grafu zbiór Φ grupujący stany tabeli pierwotnej

np. $\Phi = \{\{1,2,3,5\}, \{2,4\}, \{3,6\}, \{4,6\}\}$

Żeby zbiór stanów zgodnych był minimalny $\Phi = \Phi_{min}$ musi spełniać warunki

1. Pokrycia
   Wszystkie stany z pierwotnej tabeli muszą być reprezentowane w Φ

2. Zamknięcia
   Dla każdego stanu wejść wszystkie stany z grupy w zbiorze Φ muszą przechodzić do stanów zawartych w grupie zbioru Φ

Zbiór grup stanów zgodnych

$S \backslash x_1x_0$	00	01	11	10
1	-	3	4	2
2	4	-	-	-
3	6	6	-	-
4	-	6	1	5
5	-	-	2	-
6	3	-	2	3

S'

dla $\Phi = \{\{1,2,3,5\}, \{2,4\}, \{3,6\}, \{4,6\}\}$

warunek pokrycia spełniony bo zawera 1,2,3,4,5,6

warunek zamknięcia spełniony

$x_1x_0 = 00$ ze stanów 1,2,3,5 do 4,6 ✓

$x_1x_0 = 01$ ze stanów 1,2,3,5 do 3,6 ✓

$x_1x_0 = 11$ ze stanów 1,2,3,5 do 2,4 ✓

$x_1x_0 = 11$ ze stanów 4,6 do 1,2 ✓

$x_1x_0 = 10$ ze stanów 4,6 do 3,5 ✓

dla pozostałych grup nie ma konfliktu

ale potrzeba aż 4 stanów

$S \backslash x_1x_0$	00	01	11	10
1	-	3	4	2
2	4	-	-	-
3	6	6	-	-
4	-	6	1	5
5	-	-	2	-
6	3	-	2	3

S'

Dla $\Phi = \{\{1,2,3,5\}, \{4,6\}\}$

warunek pokrycia spełniony bo zawera 1,2,3,4,5,6

warunek zamknięcia niespełniony

$x_1x_0 = 01$ ze stanów 1,2,3,5 do 3,6 ✗

$S \backslash x_1x_0$	00	01	11	10
1	-	3	4	2
2	4	-	-	-
3	6	6	-	-
4	-	6	1	5
5	-	-	2	-
6	3	-	2	3

S'

Po rozbiciu $\{1,2,3,5\}$ na $\{1,5\}$ i $\{2,3\}$

$\Phi = \{\{1,5\}, \{2,3\}, \{2,4\}, \{3,6\}, \{4,6\}\}$

dla $\Phi = \{\{1,5\}, \{2,4\}, \{3,6\}\}$

warunek pokrycia spełniony

warunek zamknięcia spełniony

$x_1x_0 = 11$ z 1,5 do 2,4 ✓

dla pozostałych nie ma konfliktu

potrzeba tylko 3 stanów

$$\Phi_{min} = \{\underset{A}{\{1,5\}}, \underset{B}{\{2,4\}}, \underset{C}{\{3,6\}}\}$$

Tabela automatu minimalnego

x_1x_0	00	01	11	10
A	-	C/1	B/1	B/1
B	B/0	C/0	A/0	A/1
C	C/0	C/1	B/0	C/1

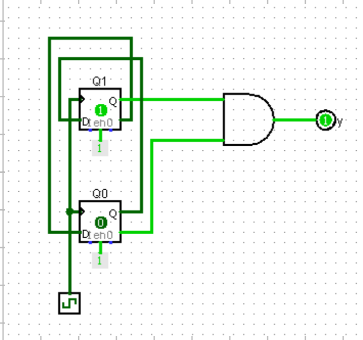
Funkcja wyjść

$Q_1Q_0 \backslash x_1x_0$	00	01	11	10
00	-	1	1	1
01	0	0	0	1
11	0	1	0	1
10	-	-	-	-

$$y = \overline{Q}_0 + x_1\overline{x}_0 + Q_1\overline{x}_1x_0$$

Automat bezwyjściowy

przyjmuje tylko sygnał zegara jako wejście
automat typu Moore'a



Przykład - automat sygnalizujący co czwarty impuls zegara

s	s'	y
1	2	0
2	3	0
3	4	0
4	1	1

Q_1Q_0	$Q_1'Q_0'$	y
00	01	0
01	11	0
11	10	0
10	00	1

$D_1 = Q_0$

$D_0 = \overline{Q_1}$

$y = Q_1\overline{Q_0}$