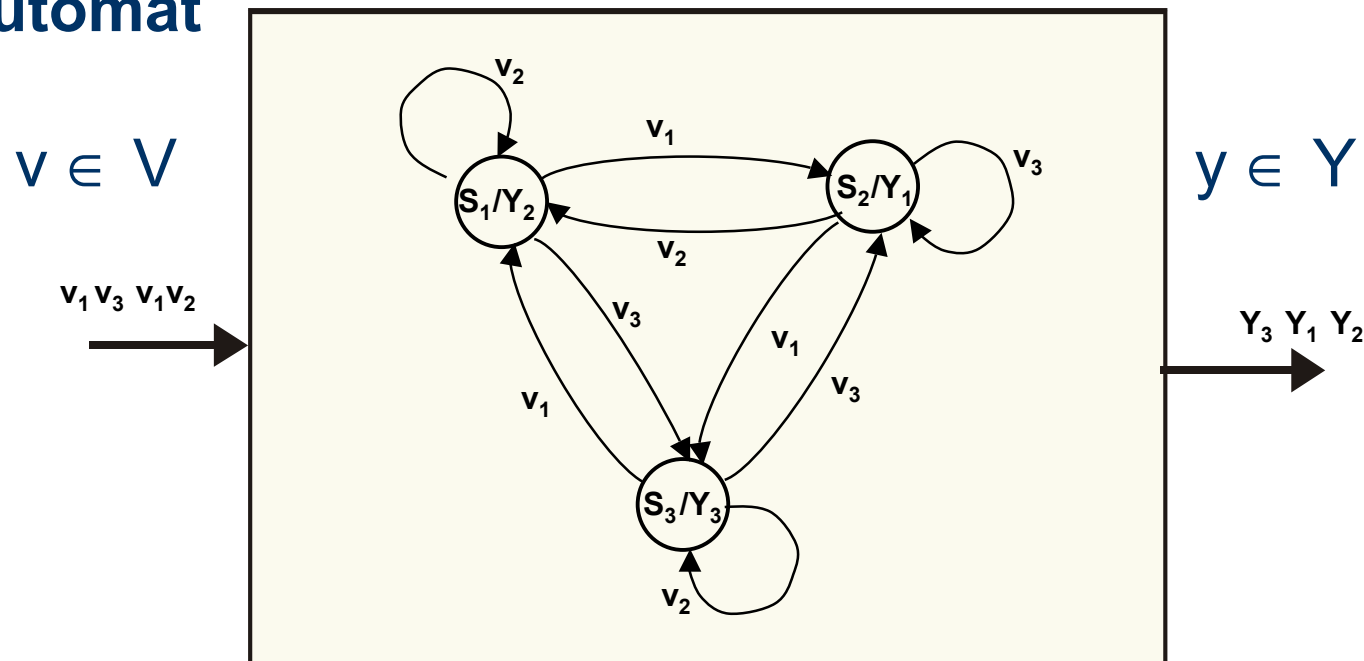


Układy sekwencyjne - pojęcie automatu

Automat



jest definiowany przez określenie:

- a) zbioru liter wejściowych V (X) i wyjściowych Y ,
- b) zbioru stanów wewnętrznych S ,
- c) funkcji przejść (ozn. δ),
- d) funkcji wyjść (ozn. λ).

Funkcja przejść i wyjść

Funkcja przejść: $\delta: \mathbf{S} \times \mathbf{X} \rightarrow \mathbf{S}$

Funkcja wyjść:

$\lambda: \mathbf{S} \times \mathbf{X} \rightarrow \mathbf{Y}$ (tzw. automat Mealy'ego)

$\lambda: \mathbf{S} \rightarrow \mathbf{Y}$ (tzw. automat Moore'a)

Automat może być zupełny lub niezupełny

Specyfikacja automatu

Tablica przejść-wyjść automatu:
Mealy'ego

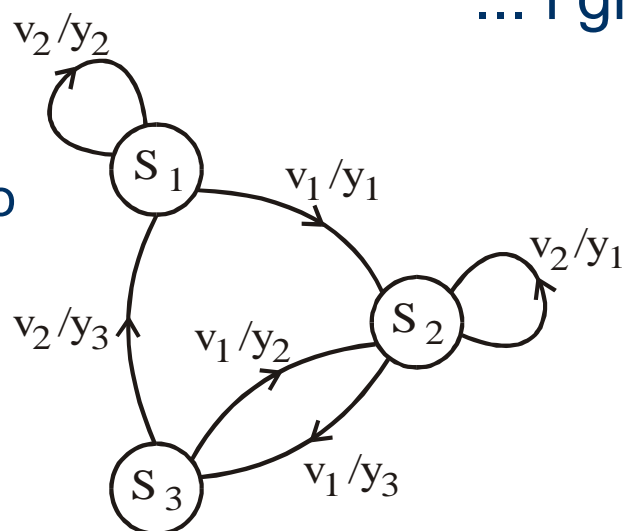
	v_1	v_2	v_1	v_2
S_1	S_2	S_1	y_1	y_2
S_2	S_3	S_2	y_3	y_1
S_3	S_2	S_1	y_2	y_3

Moore'a

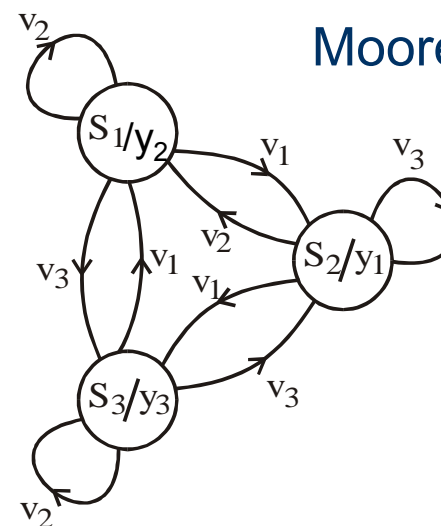
	v_1	v_2	v_3	y
S_1	S_2	S_1	S_3	y_2
S_2	S_3	S_1	S_2	y_1
S_3	S_1	S_3	S_2	y_3

... i graf

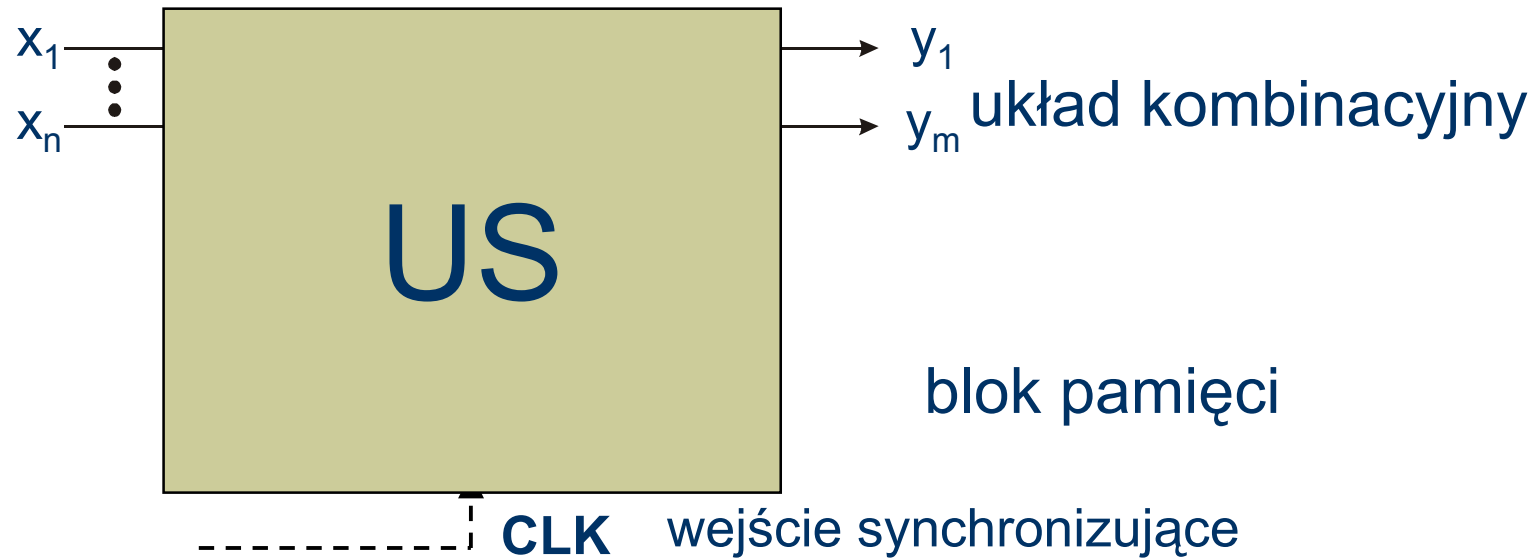
Mealy'ego



Moore'a



Układ sekwencyjny

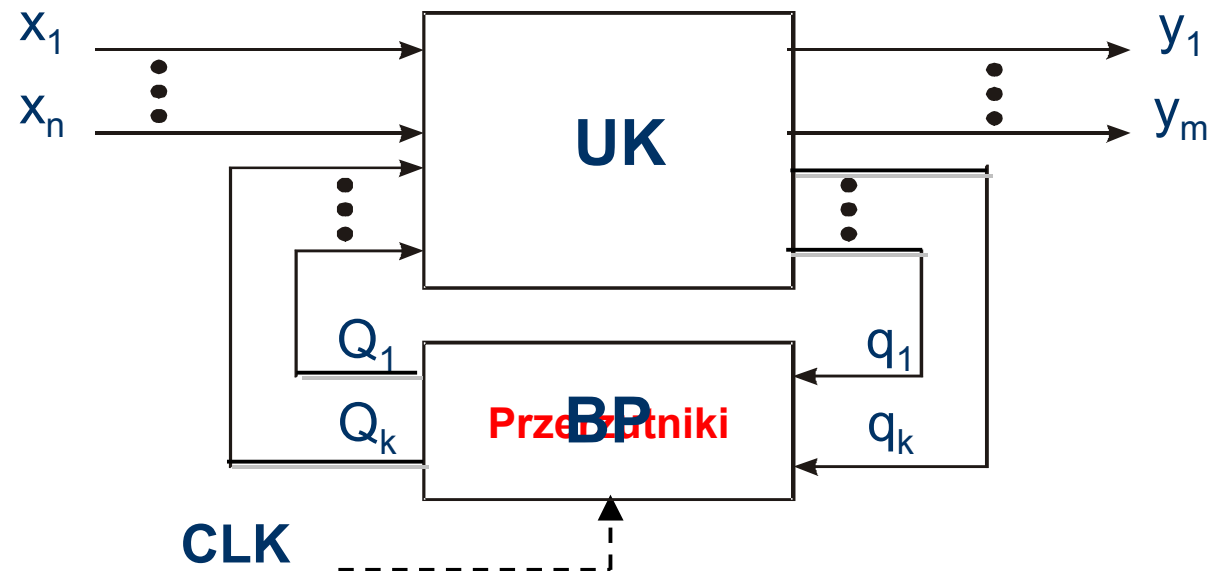


Układy sekwencyjne:

Synchroniczne (BP zbudowany z przerzutników synchronicznych)

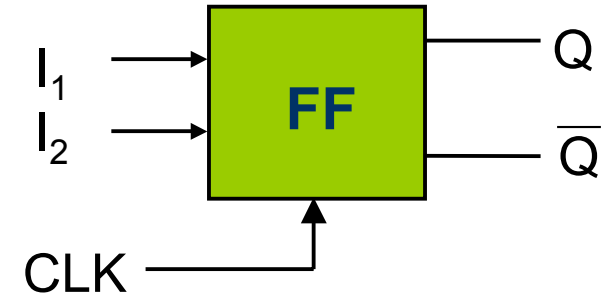
Asynchroniczne (BP realizują opóźnienia lub przerzutniki asynchroniczne)

Synchroniczne układy sekwencyjne

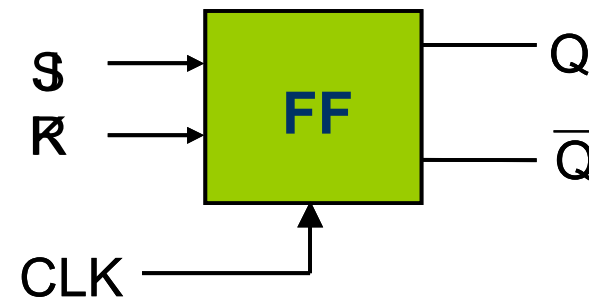
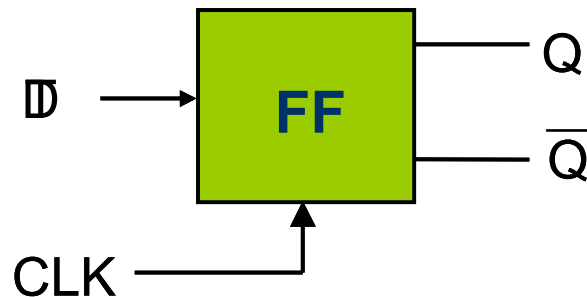


Przerzutniki

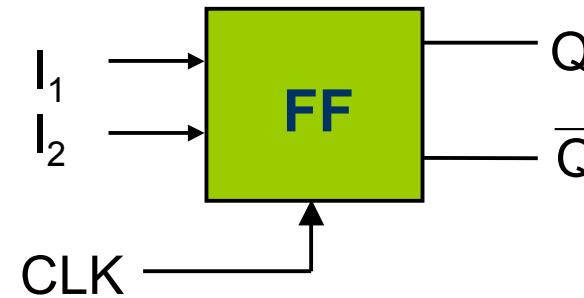
Przerzutnik



W zależności od rodzaju wejść informacyjnych wyróżniamy przerzutniki typu: **D**, **T**, **SR** oraz **JK**.



Przerzutniki



Przerzutnik jest określony:

- ☞ tablicą przejść,
- ☞ równaniem charakterystycznym,
- ☞ tablicą wzbudzeń.

Przerzutniki – tablice przejść

Truth table for D flip-flop:

Q \ D	0	1
0	0	1
1	0	1

Grouped output: Q'

Truth table for T flip-flop:

Q \ T	0	1
0	0	1
1	1	0

Grouped output: Q'

Truth table for SR flip-flop:

Q \ SR	00	01	11	10
0	0	0	–	1
1	1	0	–	1

Truth table for JK flip-flop:

Q \ JK	00	01	11	10
0	0	0	1	1
1	1	0	0	1

Równanie charakterystyczne: $Q' = f(I_1, I_2, Q)$

Przerzutniki – tablice wzbudzeń

Q Q'	D	T	S R	J K
00				
01				
10				
11				

D – delay **T** – trigger

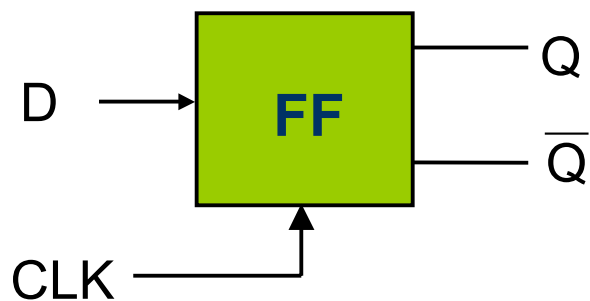
S – Set (wejście włączające)

R – Reset (wejście wyłączające)

J – wejście włączające

K – wejście wyłączające

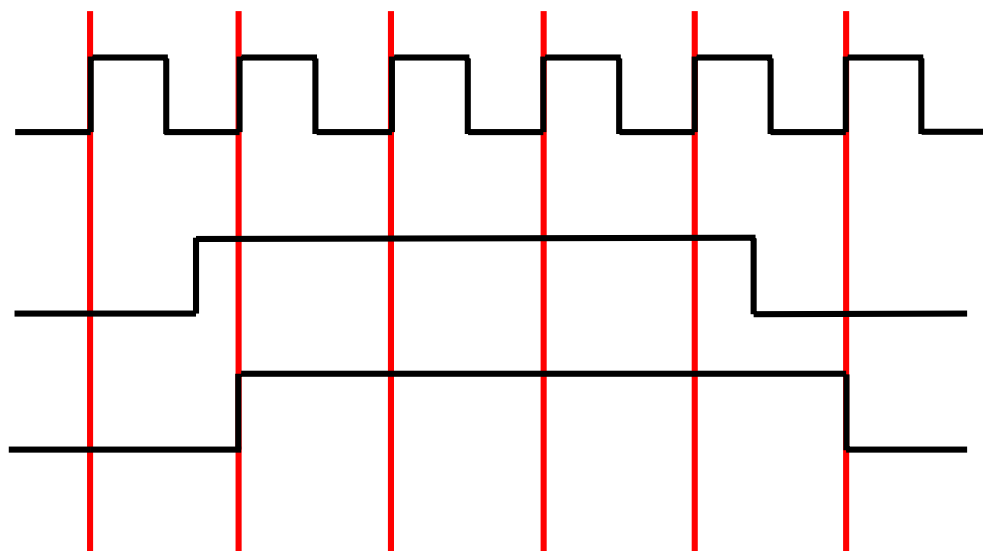
Przebiegi czasowe – przerzutnik typu D



CLK

D

Q

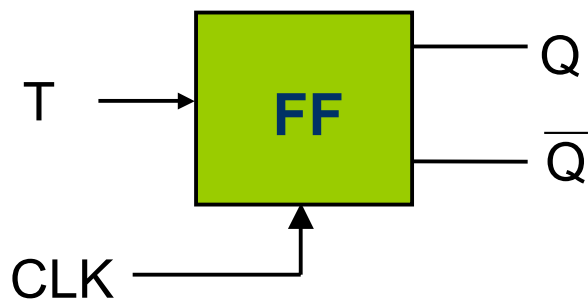


QQ'	D
00	0
01	1
10	0
11	1

I
T
P
W

ZPT

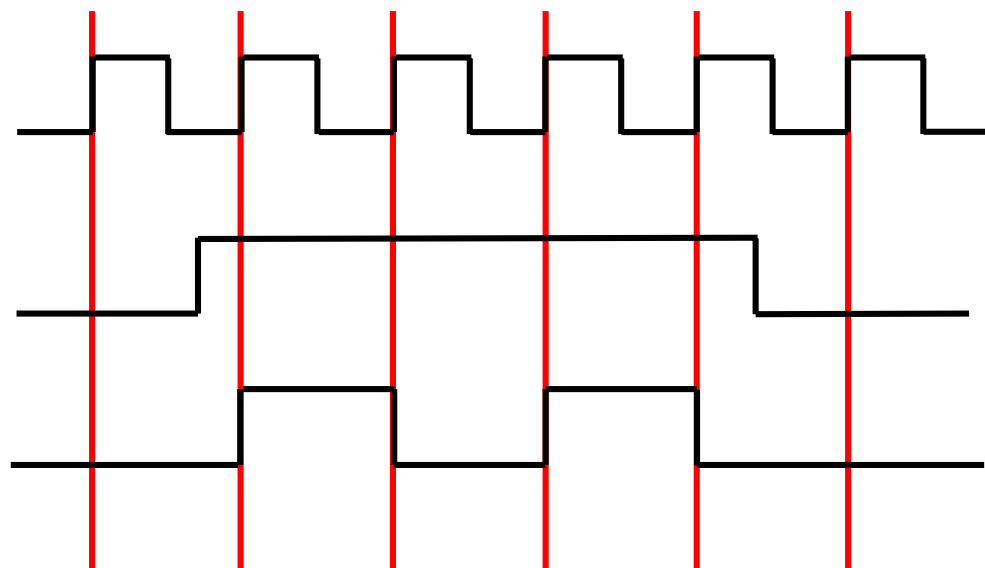
Przebiegi czasowe – przerzutnik typu T



CLK

T

Q

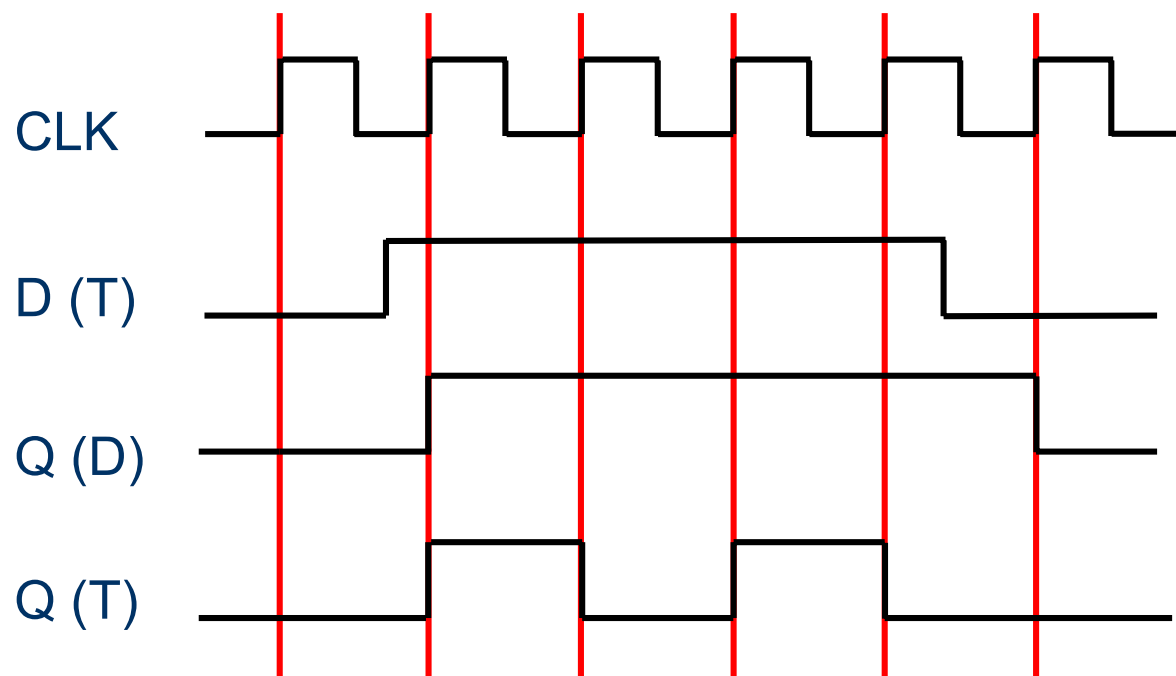


QQ'	T
00	0
01	1
10	1
11	0

I
T
P
W

ZPT

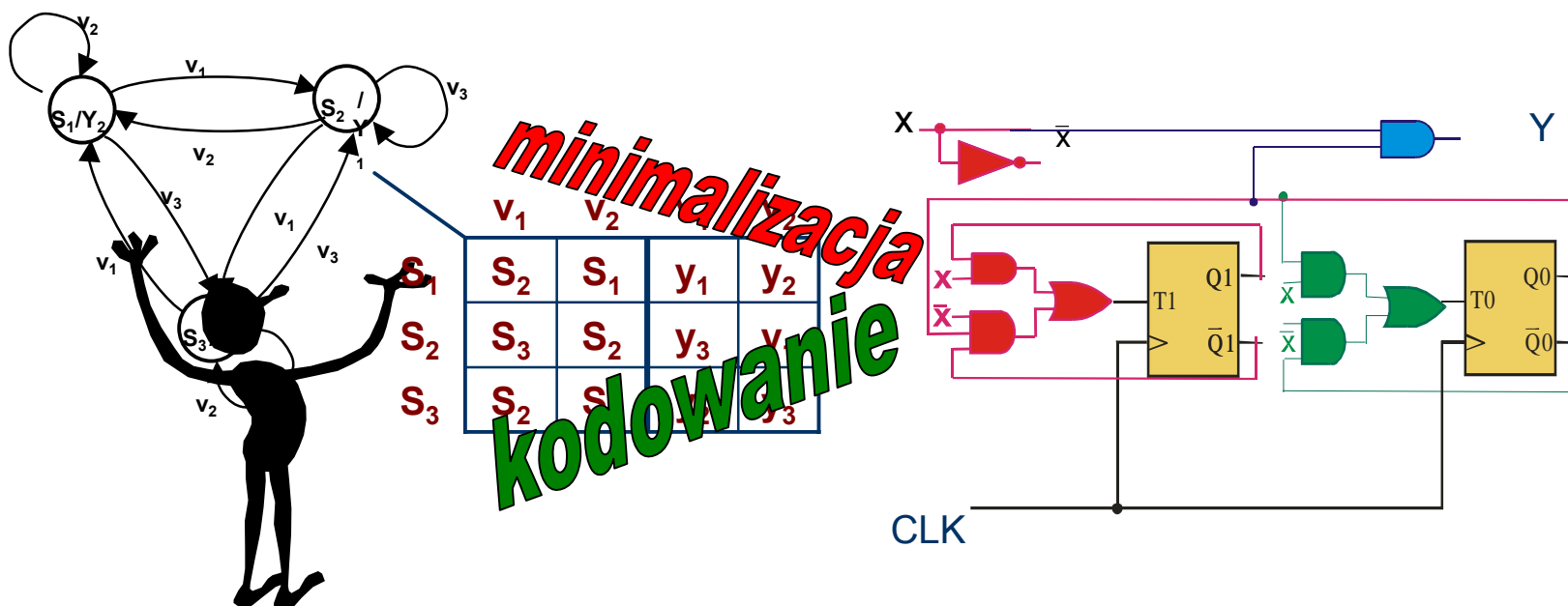
Przebiegi czasowe - porównanie



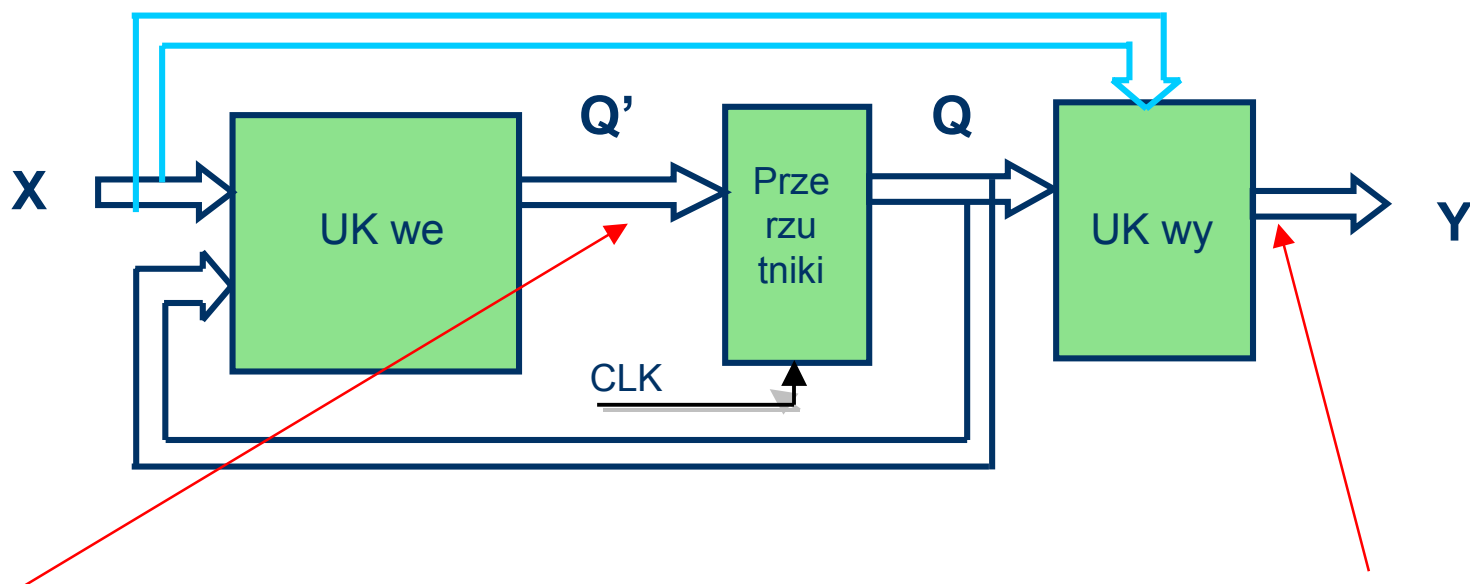
Synteza układów sekwencyjnych

Etapy syntezy:

- synteza abstrakcyjna (utworzenie tablicy przejść-wyjść)
- redukcja (minimalizacja) liczby stanów
- kodowanie stanów, liter wejściowych i wyjściowych
- synteza kombinacyjna (obliczanie funkcji wzbudzeń przerzutników i funkcji wyjściowych)



Synteza kombinacyjna...



Obliczanie funkcji sterujących wejściami
przerzutników (funkcje wzbudzeń)

$$Q' = f(X, Q)$$

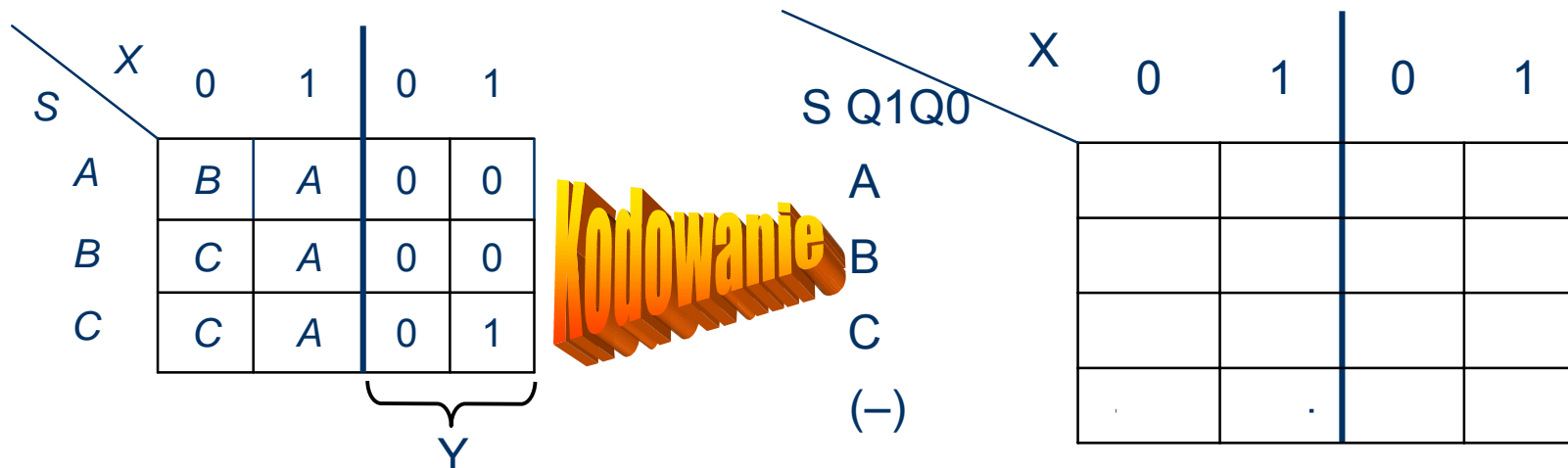
Obliczanie funkcji wyjściowych

$$Y = f(Q) \text{ (Moore)}$$

$$Y = f(X, Q) \text{ (Mealy)}$$

Przykład syntezy (detektor sekwencji)

Wystarczą dwa przerzutniki



Na razie staramy się spełnić wyłącznie warunek jednoznaczności kodowania

Obliczanie funkcji wzbudzeń i wyjściowych

QQ'	D
00	0
01	1
10	0
11	1

		X			
		0	1	0	1
S	Q1Q0				
A	00	01	00	0	0
B	01	11	00	0	0
C	11	11	00	0	1
(-)	10	-	-	-	-

		X	
		0	1
Q1Q0			
00			
01			
11			
10			

D1=Q1'

$$Q1' = D1$$

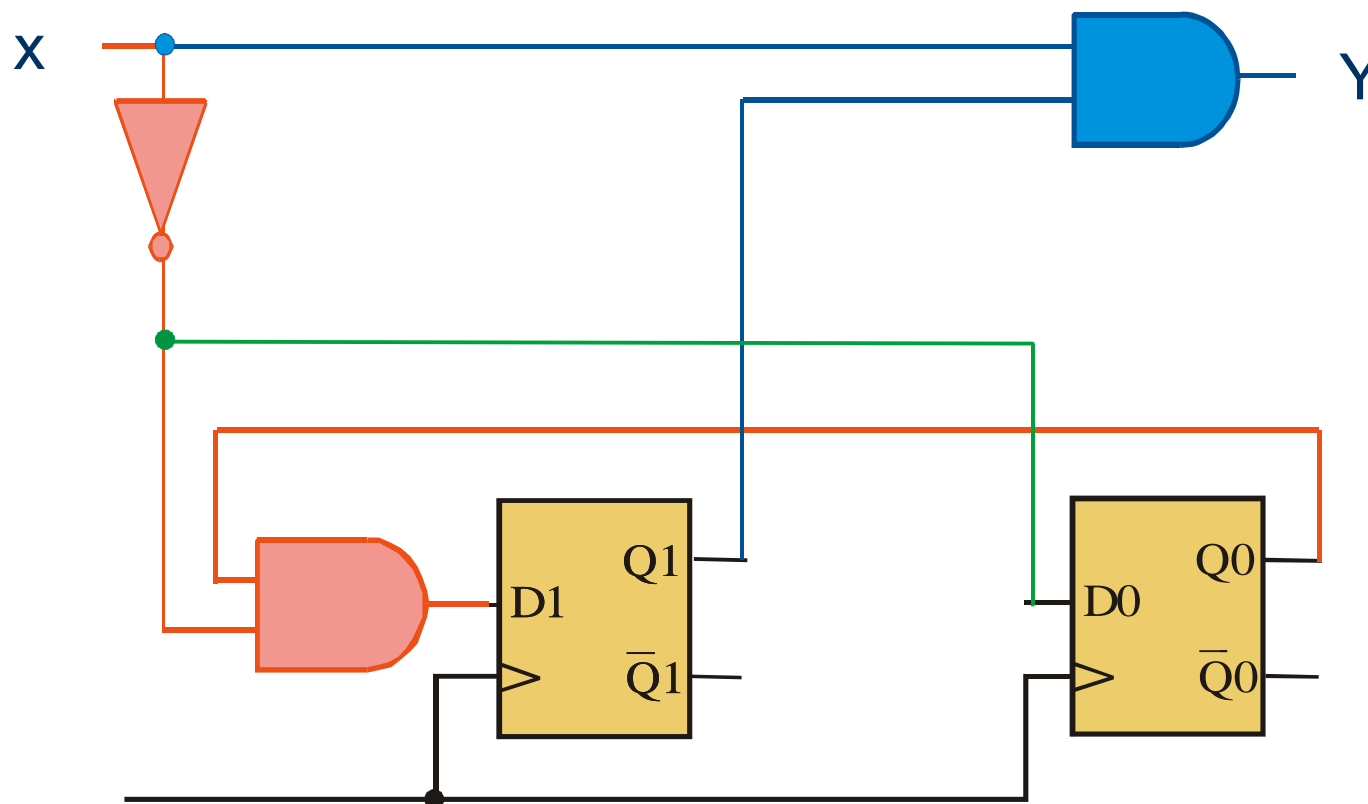
		X	
		0	1
Q1Q0			
00			
01			
11			
10			

D0=Q0'

$$Q0' =$$

Y

Schemat logiczny detektora sekwencji



$$D1 = \bar{x}Q0$$

$$D0 = \bar{x}$$

$$Y = xQ1$$

Przykład... Realizacja T

		X		0	1
Q1Q0	00	0	0	0	0
	01	1	0	0	0
	11	1	0	0	0
	10	-	-	-	-

		X		0	1
Q1Q0	00	0	0	0	0
	01	1	0	0	0
	11	0	1	0	0
	10	-	-	-	-

T1

		X		0	1
Q1Q0	00	1	0	0	0
	01	1	0	0	0
	11	1	0	0	0
	10	-	-	-	-

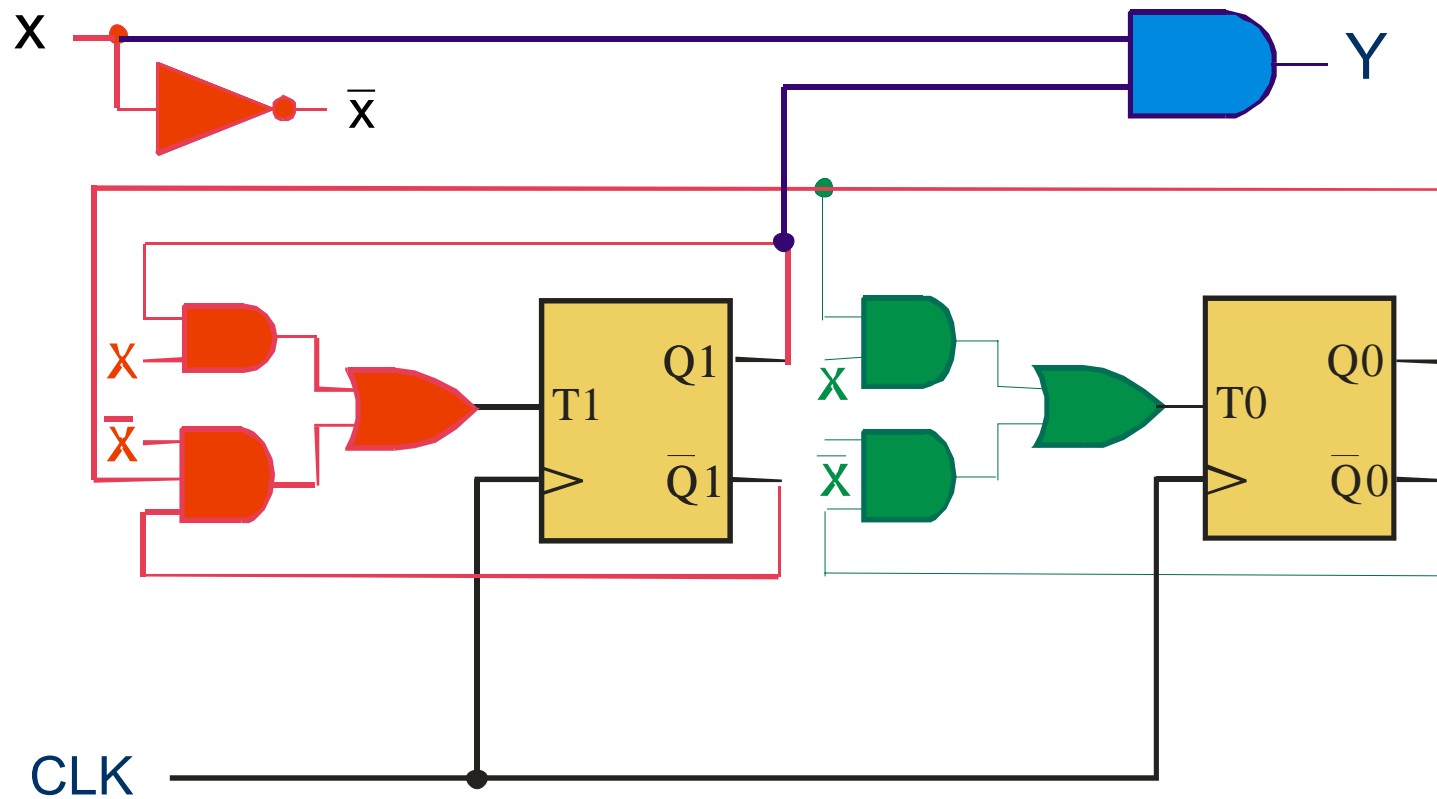
		X		0	1
Q1Q0	00	1	0	0	0
	01	0	1	0	0
	11	0	1	0	0
	10	-	-	-	-

T0 =

QQ'	T
00	0
01	1
10	1
11	0

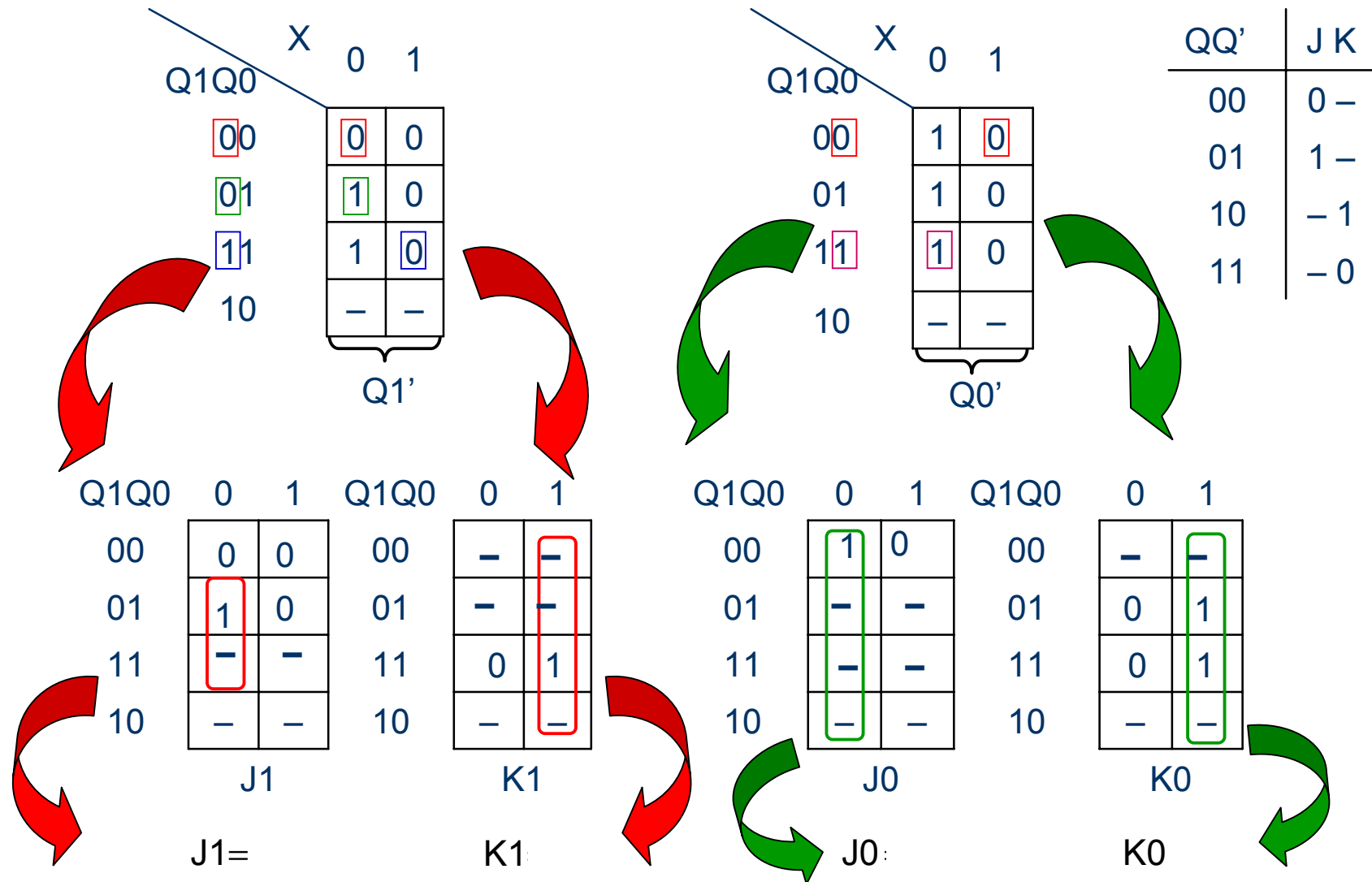
Y – jak poprzednio

Schemat logiczny detektora (T)

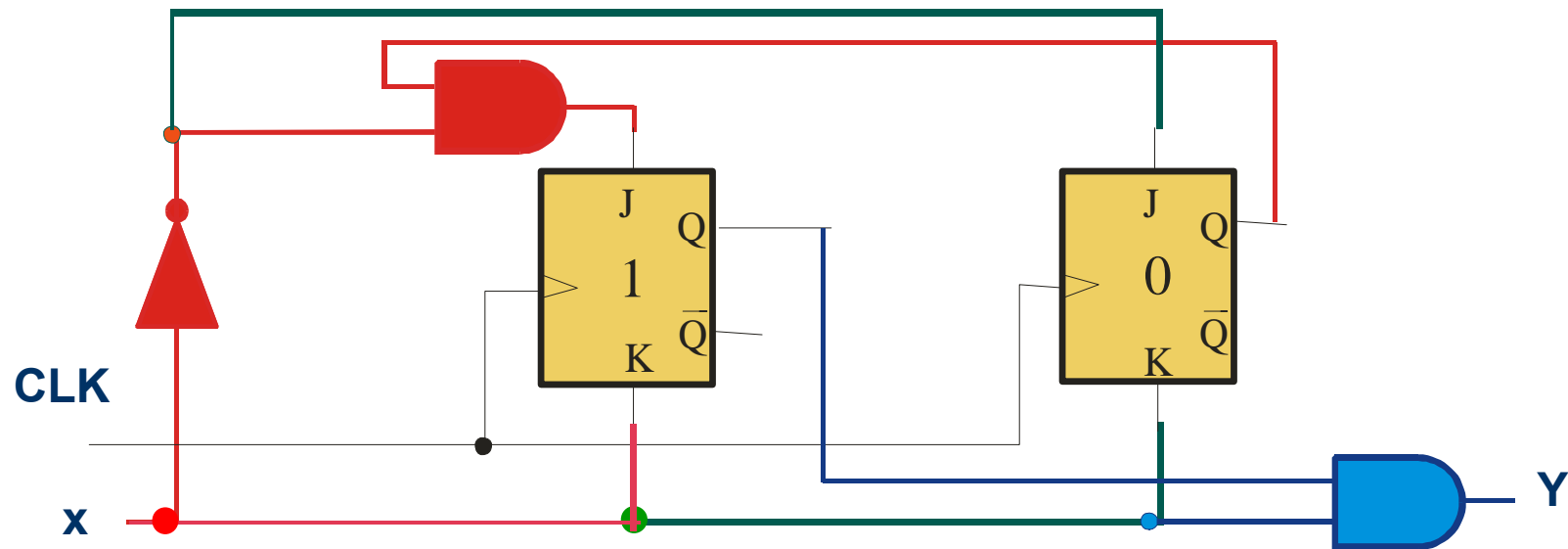


$$T1 = \bar{Q}1Q0\bar{X} + xQ1 \quad T0 = \bar{X}\bar{Q}0 + xQ0 \quad Y - \text{jak poprzednio}$$

Przykład... Realizacja JK



Schemat logiczny detektora (JK)

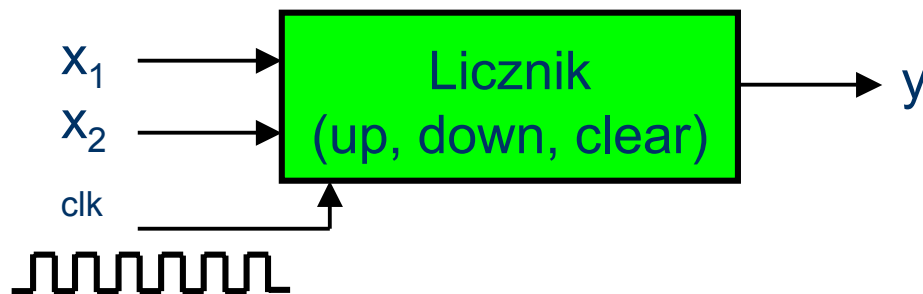


$$J1 = \bar{x}Q0 \quad K1 = x$$

$$J0 = \bar{x} \quad K0 = x$$

Y – jak poprzednio

Zadanie: licznik mod. 5 ze sterowaniem



a – *up* – liczenie do przodu
b – *down* – liczenie do tyłu
c – *clear* – zerowanie

a – *up* – liczenie do przodu
b – *down* – liczenie do tyłu
c – *clear* – zerowanie

S \ X	a	b	c	Y
S0				0
S1				0
S2				0
S3				0
S4				1

KODOWANIE

Licznik ze sterowaniem...

Q ₂ Q ₁ Q ₀	X			
	00	01	11	10
000	001	100	000	000
001	010	000	000	000
010	011	001	000	000
011	100	010	000	000
100	000	011	000	000

Q₂'Q₁'Q₀'

Q ₂ Q ₁ Q ₀	X ₁ X ₂			
	00	01	11	10
000	001	100	000	000
001	010	000	000	000
011	100	010	000	000
010	011	001	000	000
110	---	---	---	---
111	---	---	---	---
101	---	---	---	---
100	000	011	000	000

Q₂'Q₁'Q₀'

Przerzutniki D

Dla przerzutników
typu D: $Q' = D$

x_1x_2 Q2Q1Q0		00	01	11	10
000		001	100	000	000
001		010	000	000	000
011		100	010	000	000
010		011	001	000	000
110		---	---	---	---
111		---	---	---	---
101		---	---	---	---
100		000	011	000	000

$Q2'Q1'Q0'$

x_1x_2 Q2Q1Q0		00	01	11	10
000		0	1	0	0
001		0	0	0	0
011		1	0	0	0
010		0	0	0	0
110		—	—	—	—
111		—	—	—	—
101		—	—	—	—
100		0	0	0	0

$Q2' = D2$

x_1x_2 Q2Q1Q0		00	01	11	10
000		0	0	0	0
001		1	0	0	0
011		0	1	0	0
010		1	0	0	0
110		—	—	—	—
111		—	—	—	—
101		—	—	—	—
100		0	1	0	0

$Q1' = D1$

x_1x_2 Q2Q1Q0		00	01	11	10
000		1	0	0	0
001		0	0	0	0
011		0	0	0	0
010		1	1	0	0
110		—	—	—	—
111		—	—	—	—
101		—	—	—	—
100		0	1	0	0

$Q0' = D0$

Licznik ze sterowaniem - przerzutniki D

x_1x_2	00	01	11	10
$Q_2Q_1Q_0$				
000	0	1	0	0
001	0	0	0	0
011	1	0	0	0
010	0	0	0	0
110	-	-	-	-
111	-	-	-	-
101	-	-	-	-
100	0	0	0	0

D2

x_1x_2	00	01	11	10
$Q_2Q_1Q_0$				
000	0	0	0	0
001	1	0	0	0
011	0	1	0	0
010	1	0	0	0
110	-	-	-	-
111	-	-	-	-
101	-	-	-	-
100	0	1	0	0

D1

x_1x_2	00	01	11	10
$Q_2Q_1Q_0$				
000	1	0	0	0
001	0	0	0	0
011	0	0	0	0
010	1	1	0	0
110	-	-	-	-
111	-	-	-	-
101	-	-	-	-
100	0	1	0	0

D0

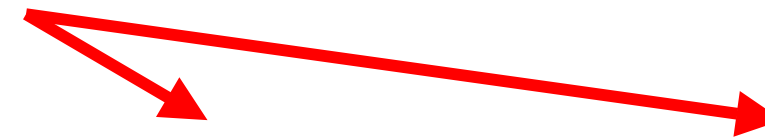
$$D2 = Q1Q0\bar{x}_1\bar{x}_2 + \bar{Q}2Q1Q0\bar{x}_1x_2$$

$$D1 = Q1\bar{Q}0\bar{x}_1\bar{x}_2 + \bar{Q}1Q0\bar{x}_1\bar{x}_2 + Q1Q0\bar{x}_1x_2 + Q2\bar{x}_1x_2$$

$$D0 = Q1\bar{Q}0\bar{x}_1 + \bar{Q}2\bar{Q}0\bar{x}_1\bar{x}_2 + Q2\bar{x}_1x_2$$

Licznik ze sterowaniem - przerzutniki JK

Tablica funkcji Q'			



Tablica funkcji J			

Tablica funkcji K			

Licznik ze sterowaniem - przerzutniki JK

	x_1x_2			
$Q_2Q_1Q_0$	00	01	11	10
000	0	1	0	0
001	0	0	0	0
011	1	0	0	0
010	0	0	0	0
110	-	-	-	-
111	-	-	-	-
101	-	-	-	-
100	0	0	0	0

Q_2'

	x_1x_2			
$Q_2Q_1Q_0$	00	01	11	10
000	0	1	0	0
001	0	0	0	0
011	1	0	0	0
010	0	0	0	0
110	-	-	-	-
111	-	-	-	-
101	-	-	-	-
100	-	-	-	-

J_2

$$J_2 = Q_1Q_0\bar{x}_1\bar{x}_2 + \bar{Q}_1\bar{Q}_0\bar{x}_1x_2$$

	x_1x_2			
$Q_2Q_1Q_0$	00	01	11	10
000	-	-	-	-
001	-	-	-	-
011	-	-	-	-
010	-	-	-	-
110	-	-	-	-
111	-	-	-	-
101	-	-	-	-
100	1	1	1	1

K_2

$$K_2 = 1$$

I
T
P
W

ZPT

Licznik ze sterowaniem - przerzutniki JK

	x_1x_2			
	00	01	11	10
$Q_2Q_1Q_0$				
000	0	0	0	0
001	1	0	0	0
011	0	1	0	0
010	1	0	0	0
110	-	-	-	-
111	-	-	-	-
101	-	-	-	-
100	0	1	0	0

Q_1'

	x_1x_2			
	0	01	11	10
$Q_2Q_1Q_0$				
000	0	0	0	0
001	1	0	0	0
011	-	-	-	-
010	-	-	-	-
110	-	-	-	-
111	-	-	-	-
101	-	-	-	-
100	0	1	0	0

J_1

	x_1x_2			
	00	01	11	10
$Q_2Q_1Q_0$				
000	-	-	-	-
001	-	-	-	-
011	1	0	1	1
010	0	1	1	1
110	-	-	-	-
111	-	-	-	-
101	-	-	-	-
100	-	-	-	-

K_1

$$J_1 = Q_0\bar{x}_1\bar{x}_2 + Q_2\bar{x}_1x_2 \quad K_1 = x_1 + Q_0\bar{x}_2 + \bar{Q}_0x_2$$

I
T
P
W

ZPT

Licznik ze sterowaniem - przerzutniki JK

	x_1x_2			
	00	01	11	10
$Q_2Q_1Q_0$				
000	1	0	0	0
001	0	0	0	0
011	0	0	0	0
010	1	1	0	0
110	-	-	-	-
111	-	-	-	-
101	-	-	-	-
100	0	1	0	0

Q_0'

	x_1x_2			
	00	01	11	10
$Q_2Q_1Q_0$				
000	1	0	0	0
001	-	-	-	-
011	-	-	-	-
010	1	1	0	0
110	-	-	-	-
111	-	-	-	-
101	-	-	-	-
100	0	1	0	0

J_0

	x_1x_2			
	00	01	11	10
$Q_2Q_1Q_0$				
000	-	-	-	-
001	1	1	1	1
011	1	1	1	1
010	-	-	-	-
110	-	-	-	-
111	-	-	-	-
101	-	-	-	-
100	-	-	-	-

K_0

$$J_0 = \overline{Q_2}\overline{x_1}\overline{x_2} + Q_1\overline{x_1} + Q_2\overline{x_1}x_2$$

$$K_0 = 1$$

I
T
P
W

ZPT



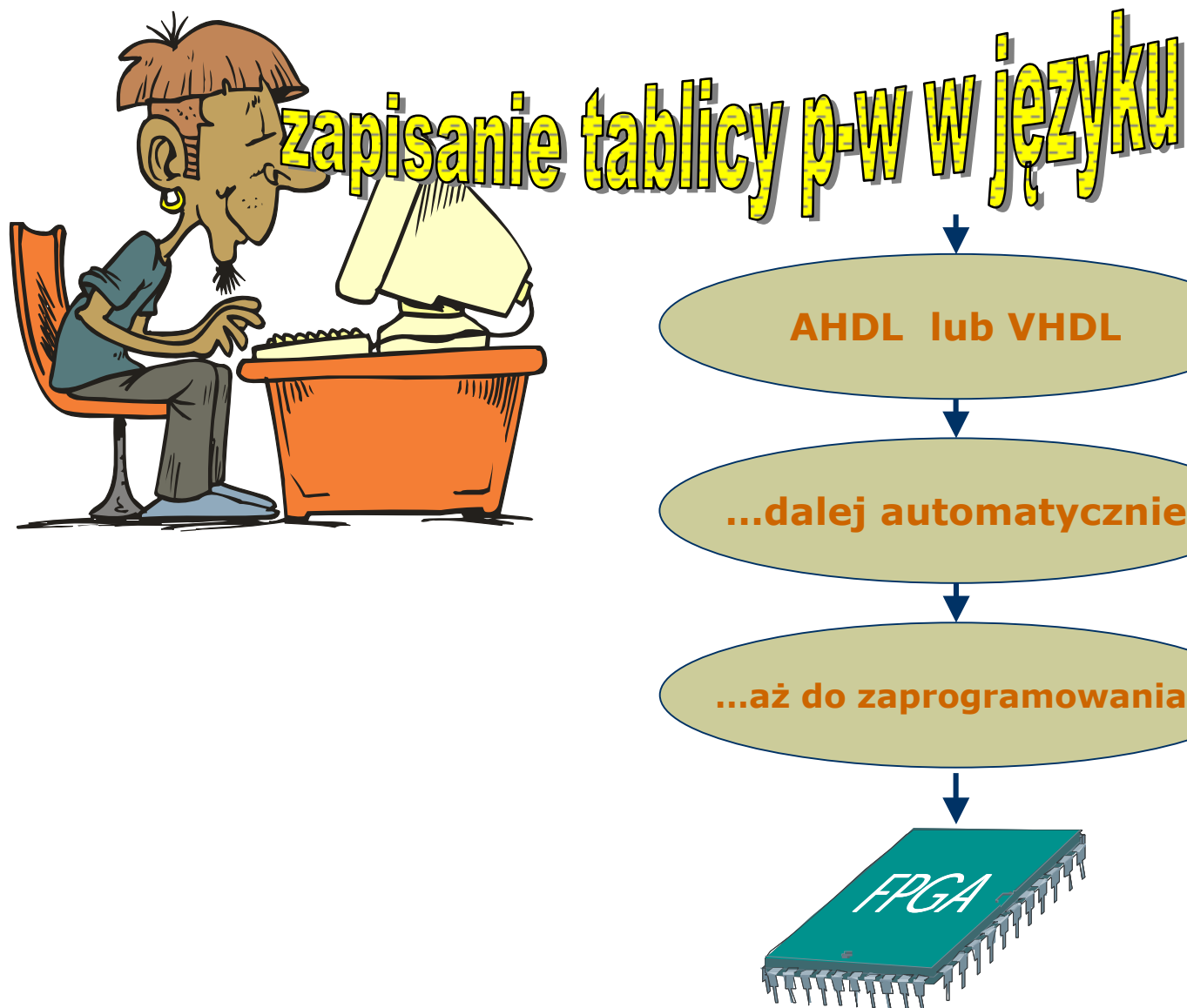
Synteza kombinacyjna układów sekwencyjnych może być (i zazwyczaj jest) procesem żmudnym, trzeba przetwarzać ogromne ilości wypełnione linkami.



Ale nie martwmy się!

Proces ten – w systemach komputerowego projektowania – jest całkowicie zautomatyzowany.

jedyną czynnością użytkownika jest ...



Specyfikacja automatu

Nie wnikając w szczegóły takiego zapisu (będą one omawiane na innych wykładach) trzeba podkreślić, że jest to wierne odwzorowanie tablicy przejść wyjść automatu.

	tablica		
przejdź-wyjść			

Odwzorowanie

AHDL
VHDL

Na przykład licznik ze sterowaniem...

S \ X	00	01	10	Y
S0	S1	S4	S0	0
S1	S2	S0	S0	0
S2	S3	S1	S0	0
S3	S4	S2	S0	0
S4	S0	S3	S0	1

TABLE

%	current	current	next	next	%
%	state	input	state	output	%
s,	v[]	=>	s,	y;	
s0,	B"00,,	=>	s1,	0;	
s0,	B"01,,	=>	s4,	0;	
s0,	B"1X,,	=>	s0,	0;	
s1,	B"00,,	=>	s2,	0;	
s1,	B"01,,	=>	s0,	0;	
s1,	B"1X,,	=>	s0,	0;	
s2,	B"00,,	=>	s3,	0;	
s2,	B"01,,	=>	s1,	0;	
s2,	B"1X,,	=>	s0,	0;	
s3,	B"00,,	=>	s4,	0;	
s3,	B"01,,	=>	s2,	0;	
s3,	B"1X,,	=>	s0,	0;	
s4,	B"00,,	=>	s0,	1;	
s4,	B"01,,	=>	s3,	1;	
s4,	B"1X,,	=>	s0,	1;	

END TABLE;

zapisany w języku AHDL, po wprowadzeniu...

...specyfikacji do edytora tekstowego

The screenshot shows the MAX+plus II Text Editor window with the following content:

```

MAX+plus II - e:\users\pawel\maz2work\chiptrip\chiptrip - [lime_cnt.tdf - Text Editor]
MAX+plus II File Edit Templates Assign Utilities Options Window Help
[Icons] Courier N

TABLE
    %      current      current      next  next %
    %      state        input        state  output %
s,      v[]    =>      s,      y;
s0,     B"00,, =>      s1,      0;
s0,     B"01,, =>      s4,      0;
s0,     B"1X,, =>      s0,      0;
s1,     B"00,, =>      s2,      0;
s1,     B"01,, =>      s0,      0;
s1,     B"1X,, =>      s0,      0;
s2,     B"00,, =>      s3,      0;
s2,     B"01,, =>      s1,      0;
s2,     B"1X,, =>      s0,      0;
s3,     B"00,, =>      s4,      0;
s3,     B"01,, =>      s2,      0;
s3,     B"1X,, =>      s0,      0;
s4,     B"00,, =>      s0,      1;
s4,     B"01,, =>      s3,      1;
s4,     B"1X,, =>      s0,      1;

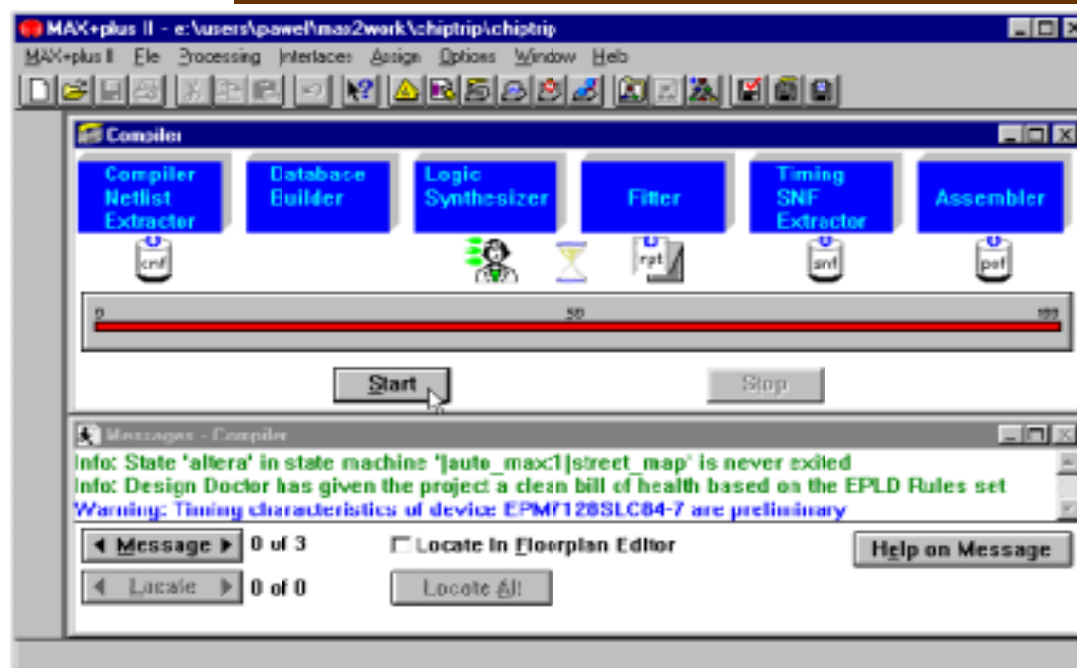
END TABLE;
  
```

The status bar at the bottom shows: Line 1 Col 1 INS

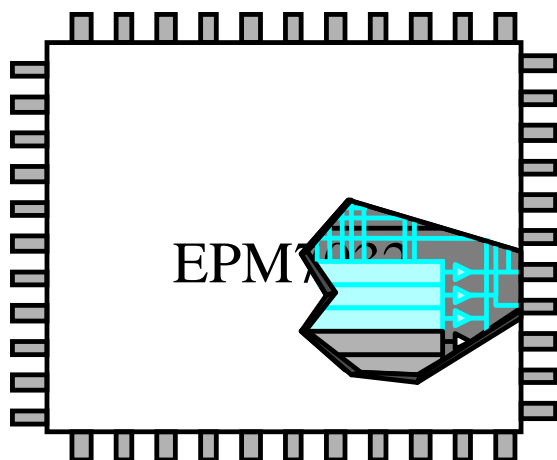
I
T
P
W

ZPT

i uruchomieniu kompilatora...



...zostanie automatycznie zrealizowany bez udziału projektanta.



```
q0 _EQ001 = !q0 & q1 & !X1 & X2
           # q2 & !X1 & X2
           # !q0 & !q2 & !X1 & !X2;

q1 _EQ002 = q2 & !X1 & X2
           # q0 & q1 & !X1 & X2
           # !q0 & q1 & !q2 & !X1 & !X2
           # q0 & !q1 & !q2 & !X1 & !X2;

q2 _EQ003 = !q0 & !q1 & !q2 & !X1 & X2
           # q0 & q1 & !q2 & !X1 & !X2;
```

Funkcje wzbudzeń licznika

(fragment raportu kompilatora):

```

q0_EQ001 = !q0 & q1 & !X1 & X2
# q2 & !X1 & X2
q1_EQ002 = q2 & !X1 & X2
# !q0 & !q2 & !X1 & !X2;
# q0 & q1 & !X1 & X2
# q0 & q1 & !X1 & X2
# !q0 & q1 & !q2 & !X1 & !X2
# q0 & !q1 & !q2 & !X1 & !X2;
q2_EQ003 = !q0 & !q1 & !q2 & !X1 & X2
# q0 & q1 & !q2 & !X1 & !X2;
q2_EQ003 = !q0 & !q1 & !q2 & !X1 & X2
# q0 & q1 & !q2 & !X1 & !X2;

```

odpowiadają one tabelkom Karnaughu podanym na planszy nr 25

Porównanie realizacji „ręcznej” z realizacją komputerową (a)

x_1x_2 Q2Q1Q0	00	01	11	10
000	0	1	0	0
001	0	0	0	0
011	1	0	0	0
010	0	0	0	0
110	—	—	—	—
111	—	—	—	—
101	—	—	—	—
100	0	0	0	0

ręczna

D2

x_1x_2 Q2Q1Q0	00	01	11	10
000	0	1	0	0
001	0	0	0	0
011	1	0	0	0
010	0	0	0	0
110	—	—	—	—
111	—	—	—	—
101	—	—	—	—
100	0	0	0	0

komputerowa

D2

Porównanie realizacji „ręcznej” z realizacją komputerową (b)

x_1x_2 Q2Q1Q0	00	01	11	10
000	0	0	0	0
001	1	0	0	0
011	0	1	0	0
010	1	0	0	0
110	—	—	—	—
111	—	—	—	—
101	—	—	—	—
100	0	1	0	0

ręczna

D1

x_1x_2 Q2Q1Q0	00	01	11	10
000	0	0	0	0
001	1	0	0	0
011	0	1	0	0
010	1	0	0	0
110	—	—	—	—
111	—	—	—	—
101	—	—	—	—
100	0	1	0	0

komputerowa

D1

Porównanie realizacji „ręcznej” z realizacją komputerową (c)

x_1x_2 Q2Q1Q0	00	01	11	10
000	1	0	0	0
001	0	0	0	0
011	0	0	0	0
010	1	1	0	0
110	—	—	—	—
111	—	—	—	—
101	—	—	—	—
100	0	1	0	0

ręczna

D0

x_1x_2 Q2Q1Q0	00	01	11	10
000	1	0	0	0
001	0	0	0	0
011	0	0	0	0
010	1	1	0	0
110	—	—	—	—
111	—	—	—	—
101	—	—	—	—
100	0	1	0	0

komputerowa

D0

Komentarz

Różnice w wyrażeniach boolowskich obliczonych przez kompilator z tymi obliczonymi poprzednio wynikają z faktu, że minimalizacja „ręczna” była wykonana dokładniej.

Czy to jest niedoskonałość kompilatora?

Raczej jego „spryt”, gdyż w realizacji na strukturach typu PAL oba rozwiązania zajmują taką samą liczbę linii iloczynu, czyli kompilator nie jest nadgorliwy.

Wniosek

Skoro kompilator oblicza funkcje wzbudzeń automatycznie, to umiejętność ta nie jest w dzisiejszych czasach sprawą najważniejszą. Z przymrużeniem oka należy więc traktować nauczanie w wielu laboratoriach układów logicznych Naszego Wydziału „sprytnych” sposobów obliczania funkcji wzbudzeń metodą „pogrubionych zer lub pogrubionych jedynek”.

Lepiej skoncentrować się na tych metodach i procedurach syntezy logicznej, które nie są jeszcze wbudowane do systemów komercyjnych.