

Technika cyfrowa — ćwiczenia 3

Krzysztof Mazur <*krzysiek@podlesie.net*>

June 23, 2020

- 1 Wprowadzenie
 - Układy synchroniczne
 - Przerzutniki
 - Przykłady

- 2 Zadania
 - Zadanie 1
 - Zadanie 2
 - Zadanie 3

Układy sekwencyjne

Co to układy sekwencyjne?

Układy sekwencyjne

Co to układy sekwencyjne?

Układy sekwencyjne to takie układy w których wyjście **nie** zależy tylko od wejść. Wszelkie inne układy jak kombinacyjne.

Układy sekwencyjne

Co to układy sekwencyjne?

Układy sekwencyjne to takie układy w których wyjście **nie** zależy tylko od wejść. Wszelkie inne układy jak kombinacyjne.

Od czego w takim razie zależy wyjście?

Układy sekwencyjne

Co to układy sekwencyjne?

Układy sekwencyjne to takie układy w których wyjście **nie** zależy tylko od wejść. Wszelkie inne układy jak kombinacyjne.

Od czego w takim razie zależy wyjście?

W układach sekwencyjnych wyjście może zależeć od:

- stanu — automat Moore'a
- stanu oraz wejść — automat Meally'ego

Układy sekwencyjne

Co to układy sekwencyjne?

Układy sekwencyjne to takie układy w których wyjście **nie** zależy tylko od wejść. Wszelkie inne układy jak kombinacyjne.

Od czego w takim razie zależy wyjście?

W układach sekwencyjnych wyjście może zależeć od:

- stanu — automat Moore'a
- stanu oraz wejść — automat Meally'ego

Co to stan?

Układy sekwencyjne

Co to układy sekwencyjne?

Układy sekwencyjne to takie układy w których wyjście **nie** zależy tylko od wejść. Wszelkie inne układy jak kombinacyjne.

Od czego w takim razie zależy wyjście?

W układach sekwencyjnych wyjście może zależeć od:

- stanu — automat Moore'a
- stanu oraz wejść — automat Meally'ego

Co to stan?

Zobacz Dynamika Układów.

Układy sekwencyjne — stan

Jak reprezentować stan w technice cyfrowej?

Układy sekwencyjne — stan

Jak reprezentować stan w technice cyfrowej?

Poprzez skończoną liczbę (automaty skończone), w Technice cyfrowej kodowaną za pomocą skończonej liczby bitów.

N bitów pozwala na zakodowanie 2^N kombinacji — „stanów”.

Czas ciągły/dyskretny?

Układy sekwencyjne — stan

Jak reprezentować stan w technice cyfrowej?

Poprzez skończoną liczbę (automaty skończone), w Technice cyfrowej kodowaną za pomocą skończonej liczby bitów.

N bitów pozwala na zakodowanie 2^N kombinacji — „stanów”.

Czas ciągły/dyskretny?

W technice cyfrowej rozróżniamy dwa podstawowe przypadki:

- czas ciągły — układy asynchroniczne
- czas dyskretny — układy synchroniczne

Układy synchroniczne

Automat Moore'a

$$\mathbf{q}(i + 1) = f(\mathbf{q}(i), \mathbf{x}(i)) \quad (1)$$

$$\mathbf{y}(i) = g(\mathbf{q}(i)) \quad (2)$$

Automat Meally'ego

$$\mathbf{q}(i + 1) = f(\mathbf{q}(i), \mathbf{x}(i)) \quad (3)$$

$$\mathbf{y}(t) = g(\mathbf{q}(t), \mathbf{x}(t)) \quad (4)$$

$$\mathbf{q}(t) = \mathbf{q}(iT_s + \Delta t), 0 \leq \Delta t < T_s \quad (5)$$

Układy synchroniczne

Moore vs Meally

Automat Moore'a to szczególny przypadek automatu Meally'ego.

Układy synchroniczne

Moore vs Meally

Automat Moore'a to szczególny przypadek automatu Meally'ego.

Automat Meally'ego może przenosić wartość wejść bez opóźnienia.

Układy synchroniczne

Moore vs Meally

Automat Moore'a to szczególny przypadek automatu Meally'ego.

Automat Meally'ego może przenosić wartość wejść bez opóźnienia.

W automacie Meally'ego wyjście zależy od wejść, które potencjalnie mogą się zmieniać pomiędzy cyklami zegara!

Metastabilność

Co to $x(i)$, i kiedy wejścia mogą się zmieniać?

Metastabilność

Co to $x(i)$, i kiedy wejścia mogą się zmieniać?

W układach synchronicznych wejścia **NIE** mogą się zmieniać t_s sekund przed każdą chwilą dyskretną oraz t_h sekund po niej.

Czy możemy to zawsze zapewnić?

Metastabilność

Co to $x(i)$, i kiedy wejścia mogą się zmieniać?

W układach synchronicznych wejścia **NIE** mogą się zmieniać t_s sekund przed każdą chwilą dyskretną oraz t_h sekund po niej.

Czy możemy to zawsze zapewnić?

NIE.

Na sygnały zewnętrzne **NIE** mamy wpływu.

Czy coś można z tym zrobić?

Metastabilność

Co to $x(i)$, i kiedy wejścia mogą się zmieniać?

W układach synchronicznych wejścia **NIE** mogą się zmieniać t_s sekund przed każdą chwilą dyskretną oraz t_h sekund po niej.

Czy możemy to zawsze zapewnić?

NIE.

Na sygnały zewnętrzne **NIE** mamy wpływu.

Czy coś można z tym zrobić?

Stosować **synchronizatory**. Układy, które próbują wejście oraz wystawiają wyjście zgodnie z danym zegarem, używanym przez późniejsze elementy układu.

Kiedy wyjścia są zmieniane.

Wyjścia są zmieniane co cykl zegara, czy nie powoduje to niespełnienia ograniczenia na t_h ?

Kiedy wyjścia są zmieniane.

Wyjścia są zmieniane co cykl zegara, czy nie powoduje to niespełnienia ograniczenia na t_h ?

Może, ale zwykle układy projektowane są tak, że sygnał wystawiany jest z opóźnieniem większym jak t_h .

Natomiast w przypadku różnych układów, lub przesunięć zegara pomiędzy układami, czy wolno zmiennym sygnałem zegarowym (drobne różnice w napięciu progowym różnych przerzutników mogą prowadzić do ogromnych różnic w czasie), nadal to może być problem. Trzeba to sprawdzać. Można też stosować przerzutniki Master-Slave, mniej czułe na te różnice.

Na CPLD/FPGA są dedykowane linie dla zegarów, wchodzące na wszystkie przerzutniki w tym samym czasie. Jeśli używamy tych linii, i wszystkie przerzutniki używają tego samego zegara to nie trzeba się przejmować.

Hazard w sterowaniu pamięcią?

Czy należy eliminować hazard przy sterowaniu pamięcią?

Hazard w sterowaniu pamięcią?

Czy należy eliminować hazard przy sterowaniu pamięcią?

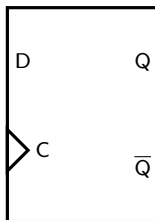
Nie. Stany pomiędzy cyklami zegara **nie** mają znaczenia.

Opis układów synchronicznych

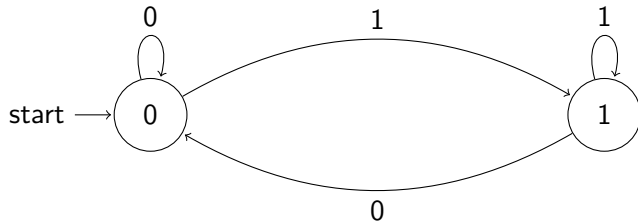
Są dwa podstawowe opisy układów synchronicznych:

- tablica przejść między stanami
- graf przejść między stanami (węzły to stany, strzałki to przejścia między stanami, z opisanymi warunkami)

Przerzutnik D



Przerzutnik D



q(i)	q(i+1) D		y D	
	0	1	0	1
0	0	1	0	0
1	0	1	1	1

Przerzutnik D — implementacja

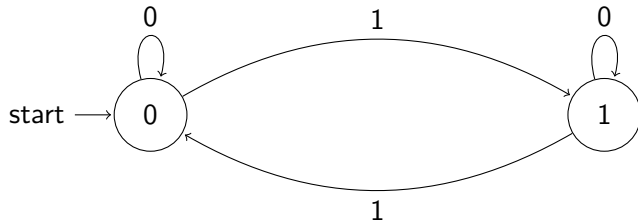
```
module d(c, d, q);  
  input c;  
  input d;  
  output q;  
  
  reg q = 0;  
  always @(posedge c)  
    q <= d;  
  
endmodule
```

Przerzutnik D — jak sterować?

q(i)	q(i+1) <i>D</i>		<i>y</i> <i>D</i>	
	0	1	0	1
0	0	1	0	0
1	0	1	1	1

$$D = q(i + 1)$$

Przerzutnik T



q(i)	q(i+1) T		y T	
	0	1	0	1
0	0	1	0	0
1	1	0	1	1

Przerzutnik T — implementacja

```
module t(c, t, q);  
  input c;  
  input t;  
  output q;  
  
  reg q = 0;  
  always @(posedge c)  
    q <= q ^ t;  
  
endmodule
```

Przerzutnik T — jak sterować?

q(i)	q(i+1)		y	
	T		T	
	0	1	0	1
0	0	1	0	0
1	1	0	1	1

$$T = q(i + 1) \wedge q(i)$$

Przerzutnik T — jak sterować 2?

$q(i)$	$q(i+1)$		T
0	0	0	0
0	1	1	1
1	1	1	0
1	0	0	1

Metoda 1 (zalecana)

Tam gdzie $q(i) = 0$:

$$T = q(i + 1)$$

w pozostałych przypadkach:

$$T = \overline{q(i + 1)}$$

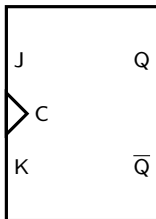
Przerzutnik T — jak sterować 2?

$q(i)$	$q(i+1)$		T
0	0	0	0
0	1	1	1
1	1	1	0
1	0	0	1

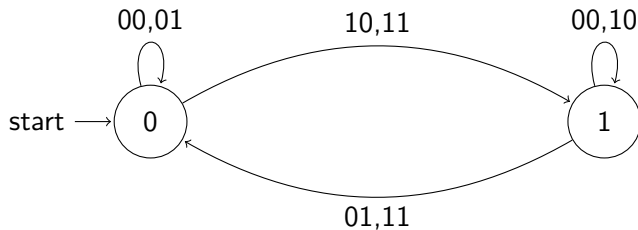
Metoda 2

T ma być 1, tam gdzie mamy pogrubioną jedynekę (**1**) lub pogrubione zero (**0**).

Przerzutnik JK



Przerzutnik JK



q(i)	q(i+1) JK				y JK			
	00	01	11	10	00	01	11	10
0	0	0	1	1	0	0	0	0
1	1	0	0	1	1	1	1	1

Przerzutnik JK — implementacja

```
module jk(c, j, k, q);  
input c, j, k;  
output q;  
reg q = 0;  
always @(posedge c) begin  
    if (j && k)  
        q <= !q;  
    else if (j)  
        q <= 1;  
    else if (k)  
        q <= 0;  
    else  
        q <= q;  
end  
endmodule
```

Przerzutnik JK — jak sterować?

q(i)	q(i+1) JK				y JK			
	00	01	11	10	00	01	11	10
0	0	0	1	1	0	0	0	0
1	1	0	0	1	1	1	1	1

q(i)	q(i+1)		JK	JK
0	0	0	00, 01	0-
0	1	1	10, 11	1-
1	1	1	00, 10	-0
1	0	0	01, 11	-1

Przerzutnik JK — jak sterować 2?

$q(i)$	$q(i+1)$	J	K
0	0	0	-
0	1	1	-
1	1	-	0
1	0	-	1

Tam gdzie $q(i) = 0$: $J = q(i + 1)$, w pozostałych przypadkach:
 $J = -$

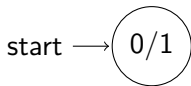
Tam gdzie $q(i) = 1$: $K = \overline{q(i + 1)}$, w pozostałych przypadkach:
 $K = q(i + 1)$

Przykład A — Dzielnik częstotliwości

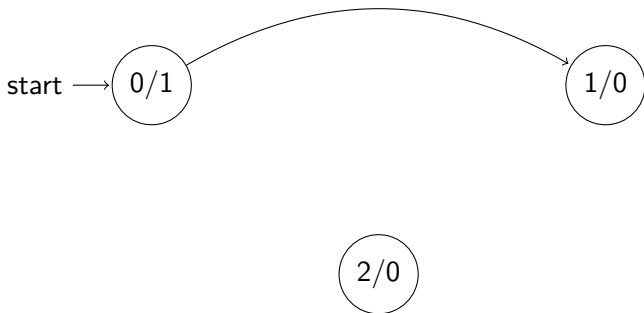
Zaprojektuj synchroniczny dzielnik częstotliwości przez 3.
Wypełnienie sygnału wyjściowego ma wynosić $1/3$.

Czyli chodzi o to by wygenerować sygnał:
1 0 0 1 0 0 1 0 0 1 0 0 ...

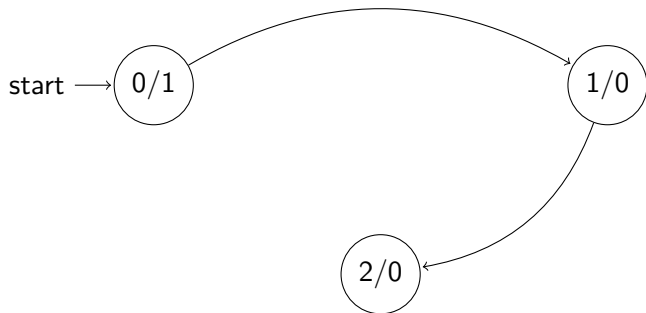
Przykład A



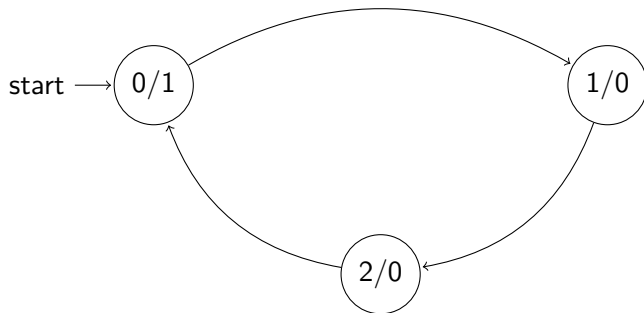
Przykład A



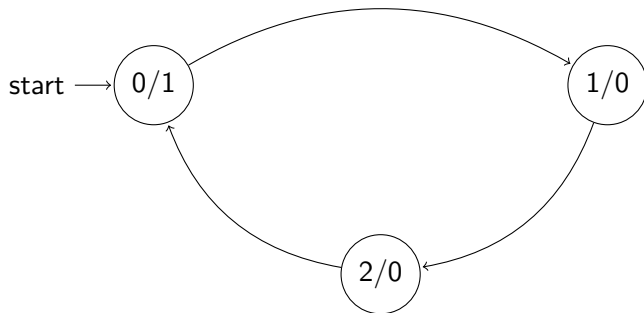
Przykład A



Przykład A



Przykład A



$q(i)$	$q(i + 1)$	y
0	1	1
1	2	0
2	0	0

Przykład A — kodowanie stanów

$q(i)$	$q(i+1)$	y
0	1	1
1	2	0
2	0	0

Numery stanów zamieniamy na kody binarne w dowolny sposób:

Stan	Kod
0	01
1	11
2	10

$q_1(i)q_0(i)$	$q_1(i+1)q_0(i+1)$	y
00	--	-
01	11	1
11	10	0
10	01	0

Przykład A — równanie wyjścia

$q_1(i)q_0(i)$	$q_1(i+1)q_0(i+1)$	y
00	--	-
01	11	1
11	10	0
10	01	0

 q_0

0 1

	0	1
0	-	1
1	0	0

Przykład A — równanie wyjścia

$q_1(i)q_0(i)$	$q_1(i+1)q_0(i+1)$	y
00	--	-
01	11	1
11	10	0
10	01	0

 q_0

0 1

q_1	0	-	1
	1	0	0

Przykład A — równanie wyjścia

$q_1(i)q_0(i)$	$q_1(i+1)q_0(i+1)$	y
00	--	-
01	11	1
11	10	0
10	01	0

 q_0

0 1

q_1	0	-	1
	1	0	0

$$y = \overline{q_1}$$

Przykład A — bit 0, D

$q_1(i)q_0(i)$	$q_1(i+1)q_0(i+1)$	y
00	--	-
01	11	1
11	10	0
10	01	0

 q_0

0 1

		0	1
q_1	0	-	1
	1	1	0

Przykład A — bit 0, D

$q_1(i)q_0(i)$	$q_1(i+1)q_0(i+1)$	y
00	--	-
01	11	1
11	10	0
10	01	0

 q_0

0 1

q_1	0	-	1
	1	1	0

Przykład A — bit 0, D

$q_1(i)q_0(i)$	$q_1(i+1)q_0(i+1)$	y
00	--	-
01	11	1
11	10	0
10	01	0

 q_0

0 1

	0	1
0	-	1
1	1	0

$$D_0 = q_0(i+1) = \overline{q_1} + \overline{q_0}$$

Przykład A — bit 1, D

$q_1(i)q_0(i)$	$q_1(i+1)q_0(i+1)$	y
00	--	-
01	11	1
11	10	0
10	01	0

 q_0

0 1

		0	1
q_1	0	-	1
	1	0	1

Przykład A — bit 1, D

$q_1(i)q_0(i)$	$q_1(i+1)q_0(i+1)$	y
00	--	-
01	11	1
11	10	0
10	01	0

 q_0

0 1

q_1	0	-	1
	1	0	1

Przykład A — bit 1, D

$q_1(i)q_0(i)$	$q_1(i+1)q_0(i+1)$	y
00	--	-
01	11	1
11	10	0
10	01	0

 q_0

0 1

q_1	0	-	1
	1	0	1

$$D_1 = q_1(i+1) = q_0$$

Przykład A — bit 0, T

		q_0	
		q_0	
		0	1
q_1	0	-	1
	1	1	0

Przykład A — bit 0, T

		T_0	
		q_0	
		0	1
q_1	0	-	0
	1	1	1

Przykład A — bit 0, T

T_0

q_0

0 1

q_1	0	-	0
1	1	1	1

Przykład A — bit 0, T

		T_0	
		q_0	
		0	1
q_1	0	-	0
	1	1	1

$$T_0 = q_1$$

Przykład A — bit 1, T

		q_1	
		q_0	
		0	1
q_1	0	-	1
	1	0	1

Przykład A — bit 1, T

		T_1	
		q_0	
		0	1
q_1	0	-	1
	1	1	0

Przykład A — bit 1, T

T_1

q_0

	0	1
0	-	1
1	1	0

Przykład A — bit 1, T

T_1

q_0

	0	1
0	-	1
1	1	0

$$T_1 = \overline{q_1} + \overline{q_0}$$

Przykład A — bit 0, J

		q_0	
		q_0	
		0	1
q_1	0	-	1
	1	1	0

Przykład A — bit 0, J

		J_0	
		q_0	
		0	1
q_1	0	-	-
	1	1	-

Przykład A — bit 0, J

J_0

q_0

0 1

q_1	0	-	-
	1	1	-

Przykład A — bit 0, J

J_0

q_0

0 1

q_1	0	-	-
1		1	-

$$J_0 = 1$$

Przykład A — bit 0, K

		q_0	
		q_0	
		0	1
q_1	0	-	1
	1	1	0

Przykład A — bit 0, K

		K_0	
		q_0	
		0	1
q_1	0	-	0
	1	-	1

Przykład A — bit 0, K

		K_0	
		q_0	
		0	1
q_1	0	-	0
	1	-	1

Przykład A — bit 0, K

		K_0	
		q_0	
		0	1
q_1	0	-	0
	1	-	1

$$K_0 = q_1$$

Przykład A — bit 1, J

		q_1	
		q_0	
		0	1
q_1	0	-	1
	1	0	1

Przykład A — bit 1, J

		J_1	
		q_0	
		0	1
q_1	0	-	1
	1	-	-

Przykład A — bit 1, J

J_1

q_0

0 1

q_1	0	-	1
	1	-	-

Przykład A — bit 1, J

J_1

q_0

0 1

q_1	0	-	1
1		-	-

$$J_1 = 1$$

Przykład A — bit 1, K

		q_1	
		q_0	
		0	1
q_1	0	-	1
	1	0	1

Przykład A — bit 1, K

K_1

q_0

	0	1
0	-	-
1	1	0

q_1

Przykład A — bit 1, K

K_1

q_0

	0	1
q_1 0	-	-
1	1	0

Przykład A — bit 1, K

K_1

q_0

	0	1
q_1 0	-	-
1	1	0

$$K_1 = \overline{q_0}$$

Przykład B — 2-bitowy licznik dwukierunkowy

Jak x_1 jest aktywne to licznik liczy w górę, jak x_0 jest aktywne to licznik liczy w dół. Nigdy nie sa oba aktywne. Licznik ma mieć nasycenie.

Stan	x_1x_0				y
	00	01	11	10	
0	0	0	--	1	0
1	1	0	--	2	1
2	2	1	--	3	2
3	3	2	--	3	3

Przykład B — kodowanie stanów

Stan	x_1x_0				y
	00	01	11	10	
0	0	0	--	1	0
1	1	0	--	2	1
2	2	1	--	3	2
3	3	2	--	3	3

Stany kodujemy jak wyjścia, w NKB.

Stan	x_1x_0				y
	00	01	11	10	
00	00	00	--	01	00
01	01	00	--	10	01
11	11	10	--	11	11
10	10	01	--	11	10

Przykład B — bit 0, J

		x_1x_0			
		00	01	11	10
q_1q_0	00	0	0	-	1
	01	1	0	-	0
	11	1	0	-	1
	10	0	1	-	1

Przykład B — bit 0, J

		x_1x_0			
		00	01	11	10
q_1q_0	00	0	0	-	1
	01	-	-	-	-
	11	-	-	-	-
	10	0	1	-	1

Przykład B — bit 0, J

		x_1x_0			
		00	01	11	10
q_1q_0	00	0	0	-	1
	01	-	-	-	-
	11	-	-	-	-
	10	0	1	-	1

Przykład B — bit 0, J

		x_1x_0			
		00	01	11	10
q_1q_0	00	0	0	-	1
	01	-	-	-	-
	11	-	-	-	-
	10	0	1	-	1

$$J_0 = x_1 + q_1x_0$$

Przykład B — bit 0, K

		x_1x_0			
		00	01	11	10
q_1q_0	00	0	0	-	1
	01	1	0	-	0
	11	1	0	-	1
	10	0	1	-	1

Przykład B — bit 0, K

		x_1x_0			
		00	01	11	10
q_1q_0	00	-	-	-	-
	01	0	1	-	1
	11	0	1	-	0
	10	-	-	-	-

Przykład B — bit 0, K

		x_1x_0			
		00	01	11	10
q_1q_0	00	-	-	-	-
	01	0	1	-	1
	11	0	1	-	0
	10	-	-	-	-

Przykład B — bit 0, K

		x_1x_0			
		00	01	11	10
q_1q_0	00	-	-	-	-
	01	0	1	-	1
	11	0	1	-	0
	10	-	-	-	-

$$K_0 = x_0 + \overline{q_1}x_1$$

Przykład B — bit 1, J

		x_1x_0			
		00	01	11	10
q_1q_0	00	0	0	-	0
	01	0	0	-	1
	11	1	1	-	1
	10	1	0	-	1

Przykład B — bit 1, J

		x_1x_0			
		00	01	11	10
q_1q_0	00	0	0	-	0
	01	0	0	-	1
	11	-	-	-	-
	10	-	-	-	-

Przykład B — bit 1, J

		x_1x_0			
		00	01	11	10
q_1q_0	00	0	0	-	0
	01	0	0	-	1
	11	-	-	-	-
	10	-	-	-	-

Przykład B — bit 1, J

		x_1x_0			
		00	01	11	10
q_1q_0	00	0	0	-	0
	01	0	0	-	1
	11	-	-	-	-
	10	-	-	-	-

$$J_1 = q_0x_1$$

Przykład B — bit 1, K

		x_1x_0			
		00	01	11	10
q_1q_0	00	0	0	-	0
	01	0	0	-	1
	11	1	1	-	1
	10	1	0	-	1

Przykład B — bit 1, K

		x_1x_0			
		00	01	11	10
q_1q_0	00	-	-	-	-
	01	-	-	-	-
	11	0	0	-	0
	10	0	1	-	0

Przykład B — bit 1, K

		x_1x_0			
		00	01	11	10
q_1q_0	00	-	-	-	-
	01	-	-	-	-
	11	0	0	-	0
	10	0	1	-	0

Przykład B — bit 1, K

		x_1x_0			
		00	01	11	10
q_1q_0	00	-	-	-	-
	01	-	-	-	-
	11	0	0	-	0
	10	0	1	-	0

$$K_1 = \overline{q_0}x_0$$

Zadanie 1

Zaprojektuj układ sekwencyjny synchroniczny o strukturze Moore'a o wejściu 2-bitowym x_1x_0 w NKB, wyjściu y oraz sygnale zegarowym c , sprawdzający poprawność transmitowanych ramek na wejściu x_1x_0 . Wartość 3 na wejściu oznacza koniec ramki (i początek kolejnej); wartości 0, 1 i 2 oznaczają dane transmitowane. W momencie kiedy na wejściu pojawi się symbol 3, układ powinien wskazać na wyjściu y poprawność (wartość 1) lub niepoprawność (wartość 0) ostatniej odebranej ramki. Ramka jest poprawna jeśli suma arytmetyczna modulo 3 symboli w niej występujących jest równa 0. W pozostałym przypadku wyjście y może przyjmować stan dowolny. Układ należy zrealizować w postaci minimalnej za pomocą przerzutników JK oraz bramek NAND o dowolnej liczbie wejść.

Zadanie 1 — przykłady

1 symbol, zła ramka:

i	0	1	2
x	3	1	3
y	?	-	0

1 symbol, dobra ramka:

i	0	1	2
x	3	0	3
y	?	-	1

Zadanie 1 — przykłady

3 symbole, zła ramka:

i	0	1	2	3	4
x	3	1	0	1	3
y	?	-	-	-	0

3 symbole, dobra ramka:

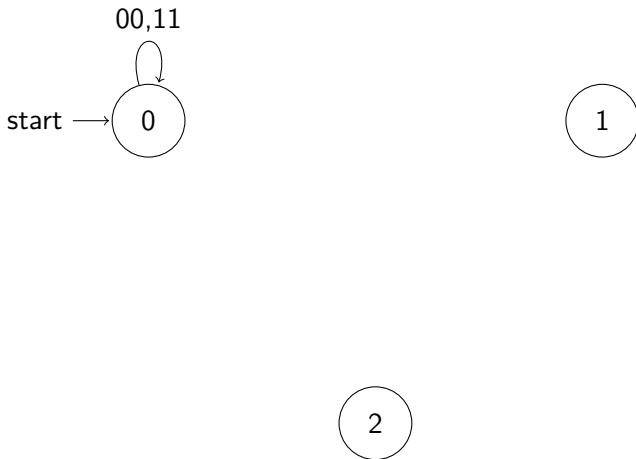
i	0	1	2	3	4
x	3	1	0	2	3
y	?	-	-	-	1

Zadanie 1 — analiza zadania

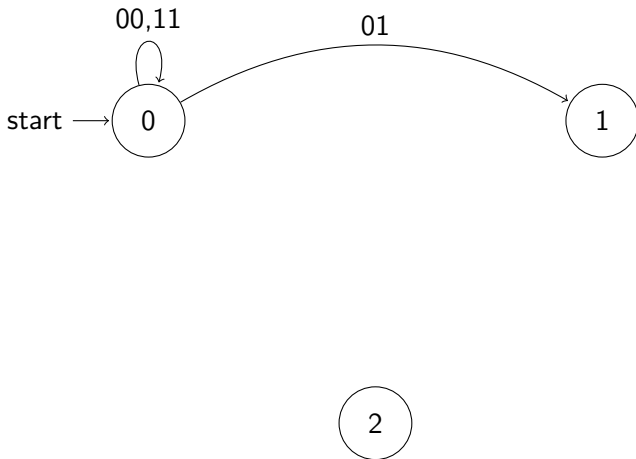
- rozwiązanie ma być w strukturze Moore'a, układ nie zdąży zareagować na koniec ramki — symbol 3, poprawna odpowiedź musi być wystawiana zawsze.
- suma modulo 3 może mieć 3 wartości, od których zależy czy odpowiedź jest poprawna, czy nie, układ naturalnie ma 3 stany.

Stan	Znaczenie	y
0	suma jest podzielna przez 3	1
1	suma podzielona przez 3 daje resztę 1	0
2	suma podzielona przez 3 daje resztę 2	0

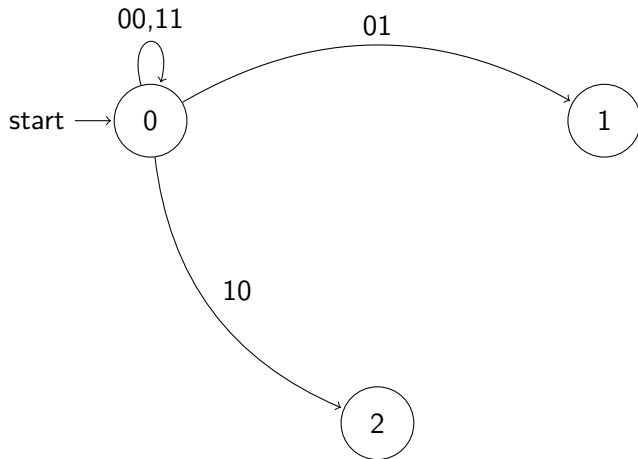
Zadanie 1 — przejścia między stanami



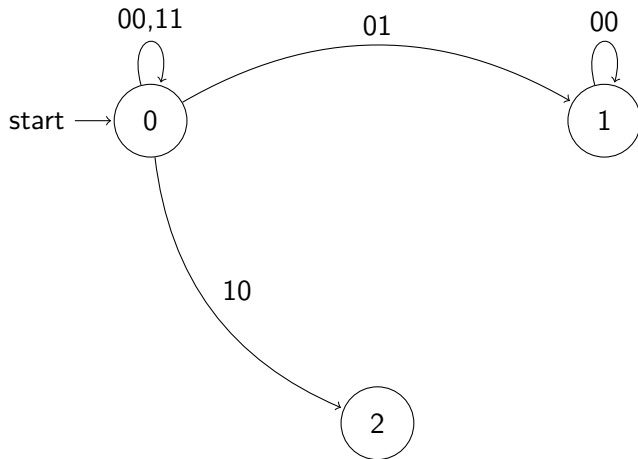
Zadanie 1 — przejścia między stanami



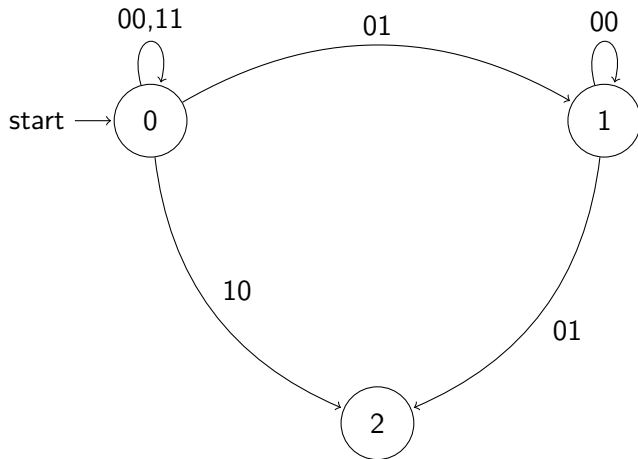
Zadanie 1 — przejścia między stanami



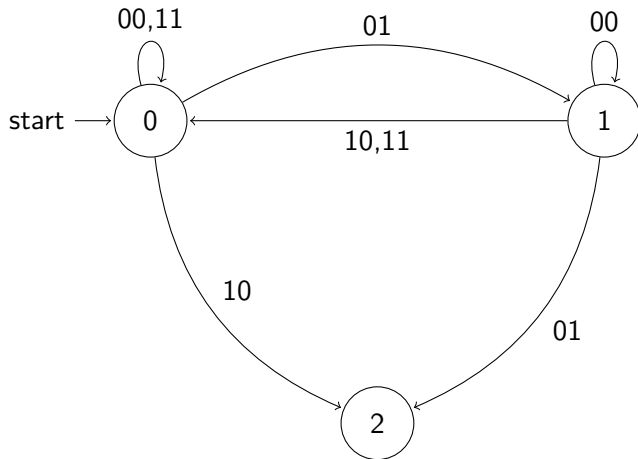
Zadanie 1 — przejścia między stanami



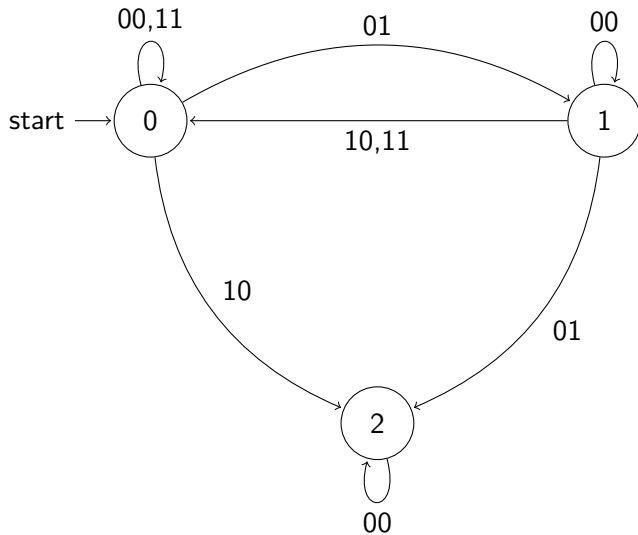
Zadanie 1 — przejścia między stanami



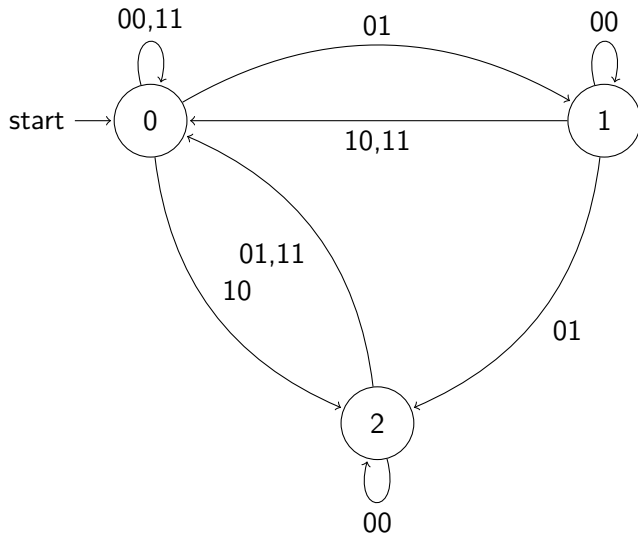
Zadanie 1 — przejścia między stanami



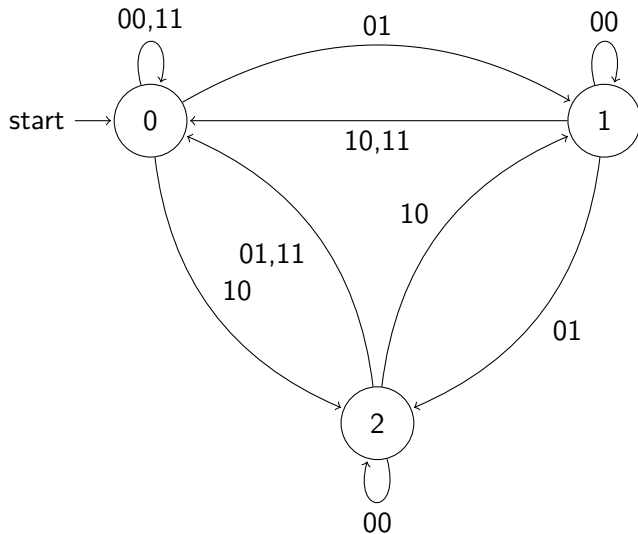
Zadanie 1 — przejścia między stanami



Zadanie 1 — przejścia między stanami



Zadanie 1 — przejścia między stanami



Zadanie 1 — przejścia między stanami

Uzupełnij tablicę.

Stan	x_1x_0				y
	00	01	11	10	
0	?	?	?	?	?
1	?	?	?	?	?
2	?	?	?	?	?

Zadanie 1 — przejścia między stanami

Uzupełnij tablicę.

Stan	x_1x_0				y
	00	01	11	10	
0	0	1	0	2	1
1	1	2	0	0	0
2	2	0	0	1	0

Zadanie 1 — kodowanie stanów

Stan	Kod
0	00
1	01
2	11

Stan	$x_1 x_0$				y
	00	01	11	10	
00	00	01	00	11	1
01	01	11	00	00	0
11	11	00	00	10	0
10	--	--	--	--	-

Zadanie 1 — równanie wyjścia

		q_0	
		0	1
q_1	0	1	0
	1	-	0

Zadanie 1 — równanie wyjścia

		q_0	
		0	1
q_1	0	1	0
	1	-	0

Zadanie 1 — równanie wyjścia

		q_0	
		0	1
q_1	0	1	0
	1	-	0

$$y = \overline{q_0}$$

Zadanie 1 — q_0 — przerzutniki D

$$D_0 = q_0$$

 $x_1 x_0$

00 01 11 10

	00	01	11	10
00	0	1	0	1
01	1	1	0	0
11	1	0	0	0
10	-	-	-	-

Zadanie 1 — q_0 — przerzutniki D

$D_0 = q_0$

$x_1 x_0$

	00	01	11	10
00	0	1	0	1
01	1	1	0	0
11	1	0	0	0
10	-	-	-	-

Zadanie 1 — q_0 — przerzutniki D

$D_0 = q_0$

$x_1 x_0$

	00	01	11	10
00	0	1	0	1
01	1	1	0	0
11	1	0	0	0
10	-	-	-	-

$$D_0 = q_0 \overline{x_1 x_0} + \overline{q_1 x_1 x_0} + \overline{q_0 x_1 x_0}$$

Zadanie 1 — q_1 — przerzutniki D

$$D_1 = q_1$$

		x_1x_0			
		00	01	11	10
q_1q_0	00	0	0	0	1
	01	0	1	0	0
	11	1	0	0	1
	10	-	-	-	-

Zadanie 1 — q_1 — przerzutniki D

$D_1 = q_1$

$x_1 x_0$

	00	01	11	10
00	0	0	0	1
01	0	1	0	0
11	1	0	0	1
10	-	-	-	-

$q_1 q_0$

Zadanie 1 — q_1 — przerzutniki D

$D_1 = q_1$

$x_1 x_0$

	00	01	11	10
00	0	0	0	1
01	0	1	0	0
11	1	0	0	1
10	-	-	-	-

$$D_1 = q_1 \bar{x}_0 + \bar{q}_1 q_0 \bar{x}_1 x_0 + \bar{q}_0 x_1 \bar{x}_0$$

Zadanie 1 — q_0 — przerzutniki T

		q_0			
		$x_1 x_0$			
		00	01	11	10
$q_1 q_0$	00	0	1	0	1
	01	1	1	0	0
	11	1	0	0	0
	10	-	-	-	-

Zadanie 1 — q_0 — przerzutniki T

T_0
 $x_1 x_0$

	00	01	11	10
00	0	1	0	1
01	0	0	1	1
11	0	1	1	1
10	-	-	-	-

$q_1 q_0$

Zadanie 1 — q_0 — przerzutniki T

$x_1 x_0$

	00	01	11	10
00	0	1	0	1
01	0	0	1	1
11	0	1	1	1
10	-	-	-	-

Zadanie 1 — q_0 — przerzutniki T

		$x_1 x_0$			
		00	01	11	10
$q_1 q_0$	00	0	1	0	1
	01	0	0	1	1
	11	0	1	1	1
	10	-	-	-	-

$$T_0 = \overline{q_0} \overline{x_1} x_0 + q_1 x_0 + q_0 x_1 + x_1 \overline{x_0}$$

Zadanie 1 — q_1 — przerzutniki T

		q_1			
		$x_1 x_0$			
		00	01	11	10
$q_1 q_0$	00	0	0	0	1
	01	0	1	0	0
	11	1	0	0	1
	10	-	-	-	-

Zadanie 1 — q_1 — przerzutniki T

T_1
 $x_1 x_0$

	00	01	11	10
00	0	0	0	1
01	0	1	0	0
11	0	1	1	0
10	-	-	-	-

$q_1 q_0$

Zadanie 1 — q_1 — przerzutniki T

		x_1x_0			
		00	01	11	10
q_1q_0	00	0	0	0	1
	01	0	1	0	0
	11	0	1	1	0
	10	-	-	-	-

Zadanie 1 — q_1 — przerzutniki T

		x_1x_0			
		00	01	11	10
q_1q_0	00	0	0	0	1
	01	0	1	0	0
	11	0	1	1	0
	10	-	-	-	-

$$T_1 = q_0\bar{x}_1x_0 + q_1x_0 + \bar{q}_0x_1\bar{x}_0$$

Zadanie 1 — q_0 — przerzutniki JK

		q_0			
		$x_1 x_0$			
		00	01	11	10
$q_1 q_0$	00	0	1	0	1
	01	1	1	0	0
	11	1	0	0	0
	10	-	-	-	-

Zadanie 1 — q_0 — przerzutniki JK

J_0

$x_1 x_0$

	00	01	11	10
00	0	1	0	1
01	-	-	-	-
11	-	-	-	-
10	-	-	-	-

$q_1 q_0$

Zadanie 1 — q_0 — przerzutniki JK

J_0

$x_1 x_0$

	00	01	11	10
00	0	1	0	1
01	-	-	-	-
11	-	-	-	-
10	-	-	-	-

Zadanie 1 — q_0 — przerzutniki JK

J_0

$x_1 x_0$

	00	01	11	10
00	0	1	0	1
01	-	-	-	-
11	-	-	-	-
10	-	-	-	-

$$J_0 = \overline{x_1}x_0 + x_1\overline{x_0}$$

Zadanie 1 — q_0 — przerzutniki JK

		q_0			
		$x_1 x_0$			
		00	01	11	10
$q_1 q_0$	00	0	1	0	1
	01	1	1	0	0
	11	1	0	0	0
	10	-	-	-	-

Zadanie 1 — q_0 — przerzutniki JK

K_0

$x_1 x_0$

	00	01	11	10
00	-	-	-	-
01	0	0	1	1
11	0	1	1	1
10	-	-	-	-

$q_1 q_0$

Zadanie 1 — q_0 — przerzutniki JK

K_0
 $x_1 x_0$

	00	01	11	10
00	-	-	-	-
01	0	0	1	1
11	0	1	1	1
10	-	-	-	-

$q_1 q_0$

Zadanie 1 — q_0 — przerzutniki JK

K_0

$x_1 x_0$

	00	01	11	10
00	-	-	-	-
01	0	0	1	1
11	0	1	1	1
10	-	-	-	-

$$K_0 = q_1 x_0 + x_1$$

Zadanie 1 — q_1 — przerzutniki JK

		q_1			
		$x_1 x_0$			
		00	01	11	10
$q_1 q_0$	00	0	0	0	1
	01	0	1	0	0
	11	1	0	0	1
	10	-	-	-	-

Zadanie 1 — q_1 — przerzutniki JK

J_1
 $x_1 x_0$

	00	01	11	10
00	0	0	0	1
01	0	1	0	0
11	-	-	-	-
10	-	-	-	-

$q_1 q_0$

Zadanie 1 — q_1 — przerzutniki JK

J_1
 $x_1 x_0$

	00	01	11	10
00	0	0	0	1
01	0	1	0	0
11	-	-	-	-
10	-	-	-	-

Zadanie 1 — q_1 — przerzutniki JK

J_1

$x_1 x_0$

	00	01	11	10
00	0	0	0	1
01	0	1	0	0
11	-	-	-	-
10	-	-	-	-

$$J_1 = q_0 \bar{x}_1 x_0 + \bar{q}_0 x_1 \bar{x}_0$$

Zadanie 1 — q_1 — przerzutniki JK

		q_1			
		$x_1 x_0$			
		00	01	11	10
$q_1 q_0$	00	0	0	0	1
	01	0	1	0	0
	11	1	0	0	1
	10	-	-	-	-

Zadanie 1 — q_1 — przerzutniki JK

K_1
 $x_1 x_0$

	00	01	11	10
00	-	-	-	-
01	-	-	-	-
11	0	1	1	0
10	-	-	-	-

$q_1 q_0$

Zadanie 1 — q_1 — przerzutniki JK

		$x_1 x_0$			
		00	01	11	10
$q_1 q_0$	00	-	-	-	-
	01	-	-	-	-
	11	0	1	1	0
	10	-	-	-	-

Zadanie 1 — q_1 — przerzutniki JK

		$x_1 x_0$			
		00	01	11	10
$q_1 q_0$	00	-	-	-	-
	01	-	-	-	-
	11	0	1	1	0
	10	-	-	-	-

$$K_1 = x_0$$

Zadanie 2

Zaprojektuj w postaci minimalnej układ sekwencyjny synchroniczny o wejściu x , wyjściu y , oraz sygnale zegarowym c . Układ ma przekazywać bez opóźnienia wejście x na wyjście y , ale dodatkowo zapewniać, że na każdych czterech kolejnych bitach wyjścia y pojawią się co najwyżej dwie jedynki. W przypadku, gdy ograniczenie to byłoby niespełnione to jedynka na wejściu ma zostać zignorowana i wyjście ma być równe 0. Układ należy zrealizować za pomocą przerzutników JK oraz bramek NOR. Hazard na wyjściu jest dopuszczalny.

Zadanie 2 — analiza zadania

- rozwiązanie ma być w strukturze Meally'ego, układ ma przekazywać sygnał bez opóźnienia
- stan bardzo łatwo zdefiniować — poprzednie 3 bity wyjścia
- ile stanów potrzeba?

Zadanie 2 — analiza zadania

- rozwiązanie ma być w strukturze Meally'ego, układ ma przekazywać sygnał bez opóźnienia
- stan bardzo łatwo zdefiniować — poprzednie 3 bity wyjścia
- ile stanów potrzeba? 7, bo 111 nie jest możliwe.

Zadanie 2 — tablica przejść

Stan	Nowy stan x		y x	
	0	1	0	1
000	000	001	0	1
001	010	011	0	1
011	110	110	0	0
010	100	101	0	1
110	100	100	0	0
111	---	---	-	-
101	010	010	0	0
100	000	001	0	1

Zadanie 2 — y

	x	
	0	1
000	0	1
001	0	1
011	0	0
010	0	1
$q_2q_1q_0$ 110	0	0
111	-	-
101	0	0
100	0	1

Zadanie 2 — y

		x	
		0	1
$q_2q_1q_0$	000	0	1
	001	0	1
	011	0	0
	010	0	1
	110	0	0
	111	-	-
	101	0	0
	100	0	1

Zadanie 2 — y

		x	
		0	1
$q_2q_1q_0$	000	0	1
	001	0	1
	011	0	0
	010	0	1
	110	0	0
	111	-	-
	101	0	0
	100	0	1

$$y = x(\overline{q_1} + \overline{q_0})(\overline{q_2} + \overline{q_1})(\overline{q_2} + \overline{q_0})$$

Zadanie 2 — bit 0, D

	x	
	0	1
000	0	1
001	0	1
011	0	0
010	0	1
110	0	0
111	-	-
101	0	0
100	0	1

Zadanie 2 — bit 0, D

	x	
	0	1
000	0	1
001	0	1
011	0	0
010	0	1
110	0	0
111	-	-
101	0	0
100	0	1

Zadanie 2 — bit 0, D

		x	
		0	1
000		0	1
001		0	1
011		0	0
010		0	1
110	$q_2 q_1 q_0$	0	0
111		-	-
101		0	0
100		0	1

$$D_0 = x(\overline{q_1} + \overline{q_0})(\overline{q_2} + \overline{q_1})(\overline{q_2} + \overline{q_0})$$

Zadanie 2 — bit 0, J

	x	
	0	1
000	0	1
001	0	1
011	0	0
010	0	1
110	0	0
111	-	-
101	0	0
100	0	1

Zadanie 2 — bit 0, J

	x	
	0	1
000	0	1
001	-	-
011	-	-
010	0	1
110	0	0
111	-	-
101	-	-
100	0	1

Zadanie 2 — bit 0, J

	x	
	0	1
000	0	1
001	-	-
011	-	-
010	0	1
$q_2q_1q_0$ 110	0	0
111	-	-
101	-	-
100	0	1

Zadanie 2 — bit 0, J

	x	
	0	1
000	0	1
001	-	-
011	-	-
010	0	1
$q_2q_1q_0$		
110	0	0
111	-	-
101	-	-
100	0	1

$$J_0 = x(\overline{q_2} + \overline{q_1})$$

Zadanie 2 — bit 0, K

	x	
	0	1
000	0	1
001	0	1
011	0	0
010	0	1
110	0	0
111	-	-
101	0	0
100	0	1

Zadanie 2 — bit 0, K

	x	
	0	1
000	-	-
001	1	0
011	1	1
010	-	-
110	-	-
111	-	-
101	1	1
100	-	-

Zadanie 2 — bit 0, K

	x	
	0	1
000	-	-
001	1	0
011	1	1
010	-	-
110	-	-
111	-	-
101	1	1
100	-	-

Zadanie 2 — bit 0, K

	x	
	0	1
000	-	-
001	1	0
011	1	1
010	-	-
110	-	-
111	-	-
101	1	1
100	-	-

$$K_0 = (q_2 + q_1 + \bar{x})$$

Zadanie 2 — bit 1, D

	x	
	0	1
000	0	0
001	1	1
011	1	1
010	0	0
110	0	0
111	-	-
101	1	1
100	0	0

Zadanie 2 — bit 1, D

	x	
	0	1
000	0	0
001	1	1
011	1	1
010	0	0
110	0	0
111	-	-
101	1	1
100	0	0

 $q_2q_1q_0$

Zadanie 2 — bit 1, D

	x	
	0	1
000	0	0
001	1	1
011	1	1
010	0	0
110	0	0
111	-	-
101	1	1
100	0	0

q₂q₁q₀

$$D_1 = q_0$$

Zadanie 2 — bit 1, J

	x	
	0	1
000	0	0
001	1	1
011	1	1
010	0	0
110	0	0
111	-	-
101	1	1
100	0	0

Zadanie 2 — bit 1, J

	x	
	0	1
000	0	0
001	1	1
011	-	-
010	-	-
$q_2q_1q_0$ 110	-	-
111	-	-
101	1	1
100	0	0

Zadanie 2 — bit 1, J

	x	
	0	1
000	0	0
001	1	1
011	-	-
010	-	-
110	-	-
111	-	-
101	1	1
100	0	0

 $q_2q_1q_0$

Zadanie 2 — bit 1, J

	x	
	0	1
000	0	0
001	1	1
011	-	-
010	-	-
110	-	-
111	-	-
101	1	1
100	0	0

$$J_1 = q_0$$

Zadanie 2 — bit 1, K

	x	
	0	1
000	0	0
001	1	1
011	1	1
010	0	0
110	0	0
111	-	-
101	1	1
100	0	0

Zadanie 2 — bit 1, K

	x	
	0	1
000	-	-
001	-	-
011	0	0
010	1	1
110	1	1
111	-	-
101	-	-
100	-	-

Zadanie 2 — bit 1, K

		x	
		0	1
$q_2q_1q_0$	000	-	-
	001	-	-
	011	0	0
	010	1	1
	110	1	1
	111	-	-
	101	-	-
	100	-	-

Zadanie 2 — bit 1, K

		x	
		0	1
$q_2q_1q_0$	000	-	-
	001	-	-
	011	0	0
	010	1	1
	110	1	1
	111	-	-
	101	-	-
	100	-	-

$$K_1 = \overline{q_0}$$

Zadanie 2 — bit 2, D

	x	
	0	1
000	0	0
001	0	0
011	1	1
010	1	1
110	1	1
111	-	-
101	0	0
100	0	0

Zadanie 2 — bit 2, D

	x	
	0	1
000	0	0
001	0	0
011	1	1
010	1	1
110	1	1
111	-	-
101	0	0
100	0	0

Zadanie 2 — bit 2, D

	x	
	0	1
000	0	0
001	0	0
011	1	1
010	1	1
110	1	1
111	-	-
101	0	0
100	0	0

$$D_2 = q_1$$

Zadanie 2 — bit 2, J

	x	
	0	1
000	0	0
001	0	0
011	1	1
010	1	1
110	1	1
111	-	-
101	0	0
100	0	0

 $q_2q_1q_0$

Zadanie 2 — bit 2, J

	x	
	0	1
000	0	0
001	0	0
011	1	1
010	1	1
$q_2q_1q_0$ 110	-	-
111	-	-
101	-	-
100	-	-

Zadanie 2 — bit 2, J

		x	
		0	1
$q_2q_1q_0$	000	0	0
	001	0	0
	011	1	1
	010	1	1
	110	-	-
	111	-	-
	101	-	-
	100	-	-

Zadanie 2 — bit 2, J

	x	
	0	1
000	0	0
001	0	0
011	1	1
010	1	1
110	-	-
111	-	-
101	-	-
100	-	-

$$J_2 = q_1$$

Zadanie 2 — bit 2, K

	x	
	0	1
000	0	0
001	0	0
011	1	1
010	1	1
110	1	1
111	-	-
101	0	0
100	0	0

Zadanie 2 — bit 2, K

	x	
	0	1
000	-	-
001	-	-
011	-	-
010	-	-
$q_2q_1q_0$ 110	0	0
111	-	-
101	1	1
100	1	1

Zadanie 2 — bit 2, K

	x	
	0	1
000	-	-
001	-	-
011	-	-
010	-	-
110	0	0
111	-	-
101	1	1
100	1	1

q₂q₁q₀

Zadanie 2 — bit 2, K

	x	
	0	1
000	-	-
001	-	-
011	-	-
010	-	-
110	0	0
111	-	-
101	1	1
100	1	1

$$J_2 = \overline{q_1}$$

Zadanie 3

Zaprojektuj w strukturze Mealy'ego układ sekwencyjny synchroniczny o wejściach a i b , wyjściu trzybitowym $y_2y_1y_0$ (y_0 jest najmniej znaczącym bitem) w kodzie U2, oraz sygnale zegarowym c . Układ ma mierzyć opóźnienie pomiędzy narastającymi zboczami sygnałów a i b mających tą samą częstotliwość. Możliwy jest pomiar opóźnienia o długości od -2 do 2 cykli sygnału zegarowego c . Wartości dodatnie odpowiadają przypadkowi, kiedy sygnał b jest opóźniony w stosunku do a . W przypadku większych opóźnień wynik ma być nasycony na -2 lub 2 . Wynik na wyjściu powinien być podawany tylko w okresie czasu kiedy sygnały a i b mają stan wysoki. W pozostałych chwilach czasu wyjście układu powinno być wyzerowane. Możliwe jest założenie, że zmierzona wartość jest natychmiast widoczna na wyjściu układu, jeśli daje ono możliwość redukcji liczby stanów.

Zadanie 3 — tablica przejść między stanami

Zadanie 3 — tablica przejść między stanami

Stan	Nowy stan <i>ab</i>				<i>y</i> <i>ab</i>			
	00	01	11	10	00	01	11	10
000	000	111	000	001	000	000	000	000
001	000	001	001	010	000	000	001	000
010	000	010	010	010	000	000	010	000
010	---	---	---	---	---	---	---	---
110	000	110	110	110	000	000	110	000
111	000	110	111	111	000	000	111	000
101	---	---	---	---	---	---	---	---
100	---	---	---	---	---	---	---	---