

## Układy sekwencyjne – przerzutniki, zatrzaski, rejestry

### 1. Cel ćwiczenia

Zapoznanie się z budową i działaniem podstawowych układów sekwencyjnych – przerzutników (*flip-flop*), zatrzasków (*latch*) oraz prostych układów sekwencyjnych rejestrów. Zapoznanie się z metodami projektowania układów rejestrów.

### 2. Elementy sekwencyjne –przerzutniki i zatrzaski

#### 2.1. Konwencja dot. nazewnictwa sygnałów

- wejścia informacyjne (**D, T, J-K**)
- wejścia sterujące (**CE, C, CLR, PRE, R, S, E**)
  - C** – *clock* – wejście zegarowe
  - CLR** – *clear* – asynchroniczne wejście resetujące
  - CE** – *clock enable* – odblokowanie sygnału zegarowego
  - PRE** – *preset* – asynchroniczne wejście ustawiające
  - R** – *reset* – synchroniczne wejście resetujące
  - S** – *set* – synchroniczne wejście ustawiające
  - G** – *gate* – wejście zezwalające
  - GE** – *gate enable* – blokowanie wejścia zezwalającego
- wyjście danych (**Q**)

#### 2.2. Przerzutniki

##### 2.2.1. Typy przerzutników

##### a) typu **D**

C	D	Q'
↑	0	0
↑	1	1
pp	X	Q

Tabela 1. Tabela prawdy. Oznaczenia: 'pp' - pozostałe przypadki, X - dowolna wartość

##### b) typu **T**

C	T	Q'
↑	0	Q
↑	1	/Q
pp	X	Q

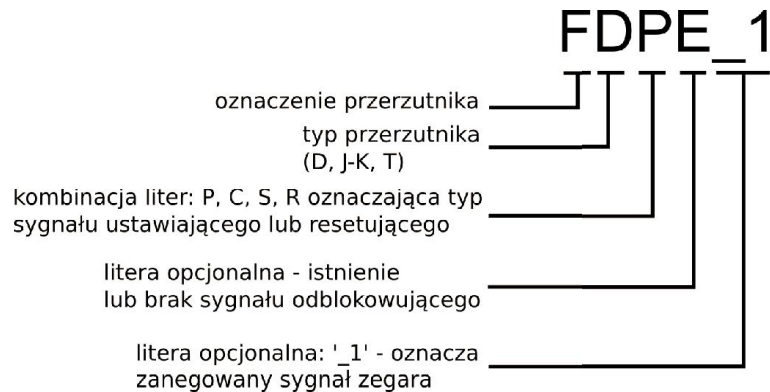
Tabela 2. Tabela prawdy. Oznaczenia: 'pp' - pozostałe przypadki, X - dowolna wartość

##### c) typu **J-K**

C	J	K	Q'
↑	0	0	Q
↑	0	1	0
↑	1	0	1
↑	1	1	/Q
pp	X	X	Q

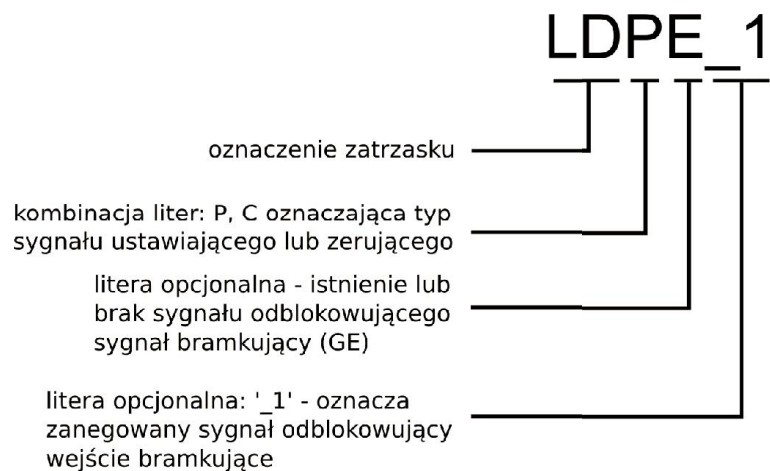
Tabela 3. Tabela prawdy. Oznaczenia: 'pp' - pozostałe przypadki, X - dowolna wartość

### 2.2.2. Konwencja nazewnictwa przerzutników

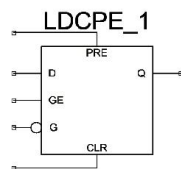


## 2.3. Zatraski

### 2.3.1. Konwencja nazewnictwa zatrasków

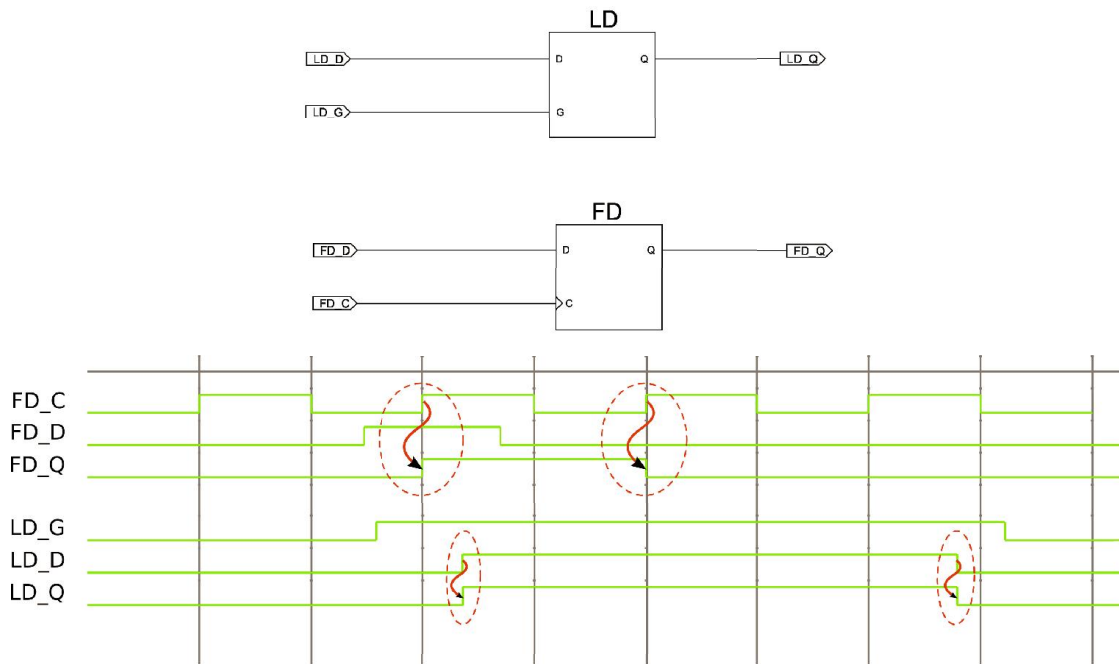


### 2.3.2. Przykład zatrasku



Rysunek 1. Przykładowy symbol zatrasku –zatrask z asynchronicznymi sygnałami CLR, PRE, GE oraz G.

## 2.4. Porównanie przerzutników i zatrząsków



Rysunek 2. Przebiegi ilustrujące porównanie przerzutnika typu D i zatrząsku.

## 3. Elementy sekwencyjne – rejestry

### 3.1. Podział rejestrów

Rejestry są układami sekwencyjnymi umożliwiającymi przechowywanie informacji cyfrowej. Ze względu na sposób wprowadzania i wyprowadzania danych rejestry dzielimy na:

- szeregowe** – umożliwiają szeregowe wprowadzanie i wyprowadzanie danych
- równoległe** – umożliwiają równoległe wprowadzanie i wyprowadzanie danych
- szeregowo-równoległe** – umożliwiają szeregowe wprowadzanie i równoległe wyprowadzanie danych
- równoległo-szeregowe** – umożliwiają równoległe wprowadzanie i szeregowe wyprowadzanie danych

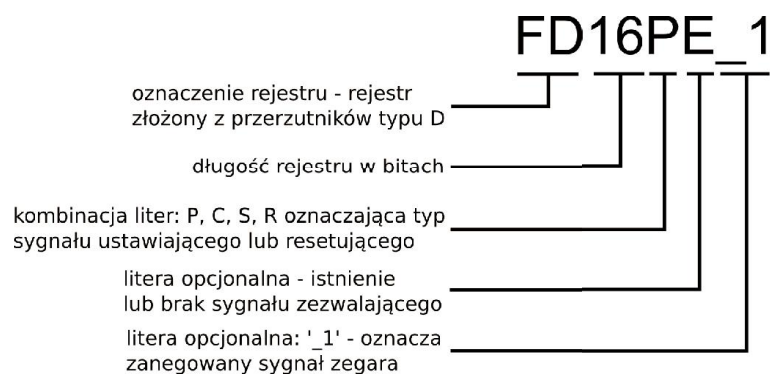
### 3.2. Inna klasyfikacja rejestrów, oparta jest na realizowanych przez nich funkcjach.

Wybrane klasy rejestrów, wg. tej systematyki, to:

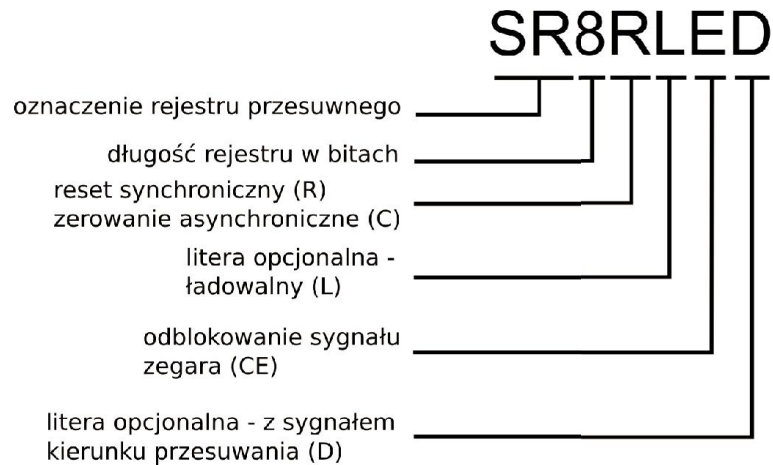
- rejestry przesuwające (przesunięcie arytmetyczne, cykliczne, typu *barrel-shifter*)
- rejestry liczące
- rejestry realizujące przekształcenia sygnału szeregowego na równoległy i równoległego na szeregowy

### 3.3. Przykłady rejestrów wykorzystywanych w Laboratorium

#### 3.3.1. Konwencja nazw rejestrów zbudowanych z przerzutników typu D o różnej długości

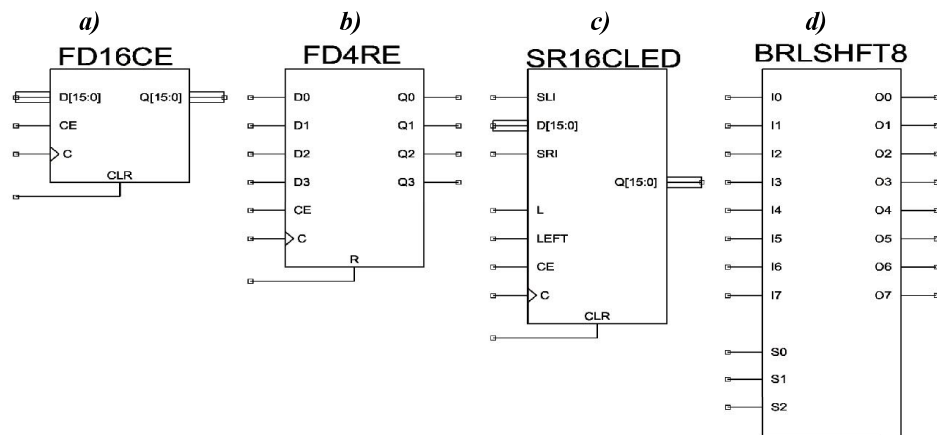


### 3.3.2. Konwencja nazewnictwa rejestrów przesuwnych



### 3.3.3. Przykłady

#### a) oznaczenia



Rysunek 3. Przykłady rejestrów. **a)** rejestr zbudowany w oparciu o przerzutniki D 8-bitowy z synchronicznym zerowaniem i sygnałem odblokowania sygnału zegarowego; **b)** rejestr zbudowany w oparciu o przerzutniki D 16-bitowy z asynchronicznym zerowaniem i sygnałem odblokowania sygnału zegarowego; **c)** 16-bitowy ładowalny rejestr przesuwany z asynchronicznym zerowaniem i sygnałem odblokowania sygnału zegarowego oraz sygnałem wyboru kierunku przesuwania (lewo lub prawo) **d)** 8-bitowy rejestr typu barrel-shifter

#### b) tabele prawdy

wejścia								wyjścia		
CLR	L	CE	LEFT	SLI	SRI	D15-D0	C	Q15'	Q14'-Q1'	Q0'
1	X	X	X	X	X	X	X	0	0	0
0	1	X	X	X	X	D15-D0	↑	D15	D14-D1	D0
0	0	0	X	X	X	X	X	Q15	Q14-Q0	Q0
0	0	1	1	SLI	X	X	↑	Q14	Q13-Q0	SLI
0	0	1	0	X	SRI	X	↑	SRI	Q15-Q2	Q1

Tabela prawdy - rejestr SR16CLED.

wejścia											wyjścia							
S2	S1	S0	I0	I1	I2	I3	I4	I5	I6	I7	O0'	O1'	O2'	O3'	O4'	O5'	O6'	O7'
0	0	0	a	b	c	d	e	f	g	h	a	b	c	d	e	f	g	h
0	0	1	a	b	c	d	e	f	g	h	b	c	d	e	f	g	h	a
0	1	0	a	b	c	d	e	f	g	h	c	d	e	f	g	h	a	b
0	1	1	a	b	c	d	e	f	g	h	d	e	f	g	h	a	b	c
1	0	0	a	b	c	d	e	f	g	h	e	f	g	h	a	b	c	d
1	0	1	a	b	c	d	e	f	g	h	f	g	h	a	b	c	d	e
1	1	0	a	b	c	d	e	f	g	h	g	h	a	b	c	d	e	f
1	1	1	a	b	c	d	e	f	g	h	h	a	b	c	d	e	f	g

Tabela prawdy - rejestr BRLSHFT8.

### 3.4. Projektowanie układów rejestrów – przykłady.

#### 3.4.1. Rejestr przesuwający

$Q_A Q_B Q_C \backslash D$	0	1
000	000	001
001	010	011
010	100	101
011	110	111
100	000	001
101	010	011
110	100	101
111	110	111

$Q'_A Q'_B Q'_C$

Tabela 4. Rejestr przesuwający – tabela przejść.

$Q_A Q_B \backslash Q_C D$	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	0	1	1	0
10	0	1	1	0

$Q'_C$

Tabela prawdy – sygnał  $Q'_C$ .  $Q'_C = D$

$Q_A Q_B \backslash Q_C D$	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	1	1
10	0	0	1	1

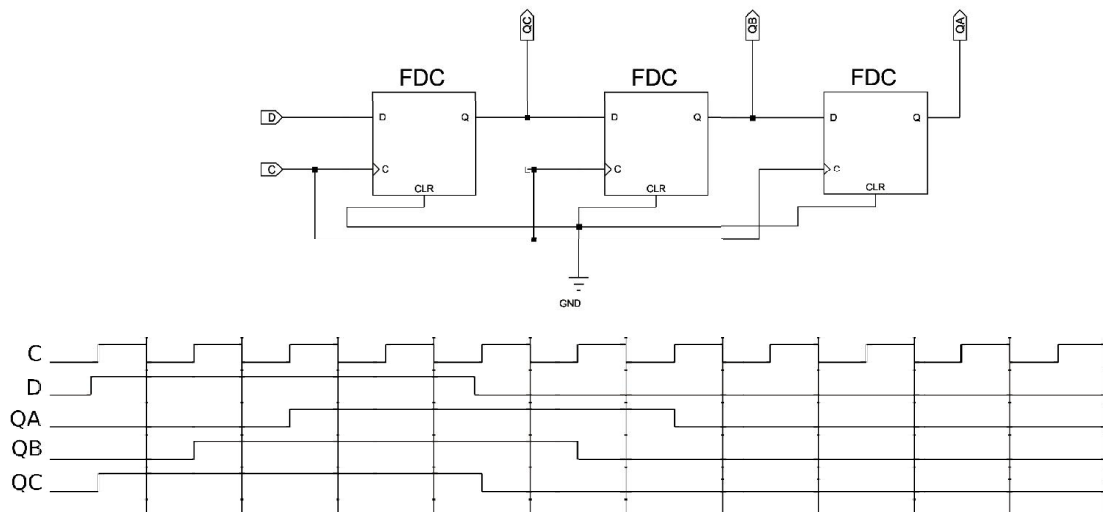
$Q'_B$

Tabela prawdy – sygnał  $Q'_B$ .  $Q'_B = Q_C$

$Q_A Q_B \backslash Q_C D$	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	0	0	0	0

$Q'_A$

Tabela prawdy – sygnał  $Q'_A$ .  $Q'_A = Q_B$



Rysunek 4. Implementacja rejestru przesuwającego 3-bitowego z wykorzystaniem przerzutników typu D oraz przebiegi odpowiednich sygnałów.

#### 4. Rejestry liczące – kody wagowe i niewagowe

Stany *rejestrów liczących* oraz *przesuwających* (będących realizacjami pewnych automatów skończonych) mogą być kodowane w postaci określonych sekwencji bitowych. Sekwencje bitowe, cyklicznie się powtarzające, dzieli się na dwie zasadnicze grupy – *kody wagowe* oraz *kody niewagowe*. Rejestry te konstruuje się w taki sposób, aby sekwencje kodów powtarzały się, tzn. po ostatnim kodzie z danej sekwencji następuje „powrót” rejestru do pierwszego kodu.

**4.1. Kody wagowe** – każda pozycja ma określoną **wagę**; ciąg wag kodu jest zwykle używany jako nazwa kodu, np.:

- NKB
- U1

- c) U2
  - d) BCD
  - e) kod Aikena (kod wagowy; BCD o wagach: 2,4,2,1)
- 4.2. Kody niewagowe** – wagi pozycji nie mają znaczenia, np.:
- a) kod BCD z nadmiarem 3 – tzw. kod +3 (otrzymuje się dodając 3 do cyfry dziesiętnej i zapisując ją w kodzie BCD)
  - b) kod Gray'a
  - c) kod Johnsona
- 4.3. Przykłady**

Kody → Cyfry ↓ wagi →	Kody wagowe			Kody niewagowe		
	Kod Aikena 2 4 2 1	BCD 8 4 2 1	NKB 8 4 2 1	Kod +3 -	Gray'a -	Johnsona -
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0	0 0 0 0 0
1	0 0 0 1	0 0 0 1	0 0 0 1	0 1 0 0	0 0 0 1	0 0 0 0 1
2	0 0 1 0	0 0 1 0	0 0 1 0	0 1 0 1	0 0 1 1	0 0 0 1 1
3	0 0 1 1	0 0 1 1	0 0 1 1	0 1 1 0	0 0 1 0	0 0 1 1 1
4	0 1 0 0	0 1 0 0	0 1 0 0	0 1 1 1	0 1 1 0	0 1 1 1 1
5	1 0 1 1	0 1 0 1	0 1 0 1	1 0 0 0	0 1 1 1	1 1 1 1 1
6	1 1 0 0	0 1 1 0	0 1 1 0	1 0 0 1	0 1 0 1	1 1 1 1 0
7	1 1 0 1	0 1 1 1	0 1 1 1	1 0 1 0	0 1 0 0	1 1 1 0 0
8	1 1 1 0	1 0 0 0	1 0 0 0	1 0 1 1	1 1 0 0	1 1 0 0 0
9	1 1 1 1	1 0 0 1	1 0 0 1	1 1 0 0	1 1 0 1	1 0 0 0 0
10	-	-	1 0 1 0	-	1 1 1 1	-
11	-	-	1 0 1 1	-	1 1 1 0	-
12	-	-	1 1 0 0	-	1 0 1 0	-
13	-	-	1 1 0 1	-	1 0 1 1	-
14	-	-	1 1 1 0	-	1 0 0 1	-
15	-	-	1 1 1 1	-	1 0 0 0	-

*Tabela 5. Przykłady kodów wagowych i niewagowych*

## 5. Zadania

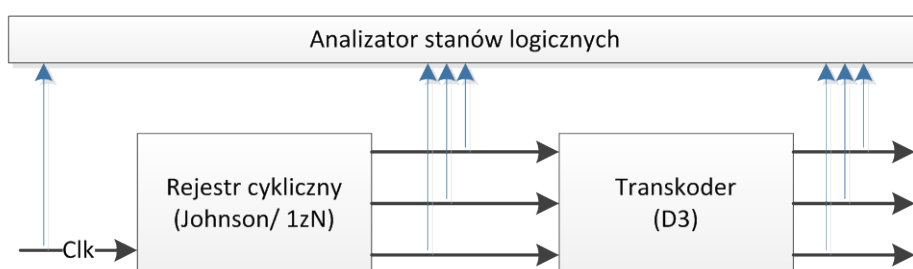
Wykonaj projekt rejestru cyklicznego z transkoderem wg podanych poniżej wymagań. Konstrukcja modułowa urządzenia powinna umożliwiać pomiar sygnałów na wyjściach kolejnych bloków przetwarzania.

A) Wymagania projektowe:

- rejestr cykliczny realizujący na wyjściu 4-bitowy kod Johnsona
- transkoder kodu Johnsona na kod kolumny 'wyj' tablicy dekodowania D3 (kolumna 'we' tablicy D3 zastąpiona kolejnymi symbolami kodu Johnsona)

B) Wymagania projektowe:

- rejestr cykliczny realizujący na wyjściu 6-bitowy kod 1zN
- transkoder kodu 1zN na kod kolumny 'wyj' tablicy dekodowania D3 (kolumna 'we' tablicy D3 zastąpiona kolejnymi symbolami kodu 1zN)



Przygotuj schemat do testowania układów z wykorzystaniem dzielnika częstotliwości `clk_gen_50` i analizatora stanów logicznych.

Dokonaj pomiarów układu wybranego przez prowadzącego przy pomocy analizatora stanów logicznych. Wyniki pomiarów (w postaci elektronicznej) wraz z kartą projektu przedstaw prowadzącemu zajęcia.