

Logika cyfrowa

Wykład 6: układy ze stanem, przerzutniki

Marek Materzok

31 marca 2021

Stan w układach cyfrowych

Stan w urządzeniach elektronicznych



Przykład – pralka:

- Pamięta, że została włączona
- Zmienia tryb na podstawie czasu i wskazań sensorów;
np. napełnianie bębna, grzanie wody, płukanie, pranie, wirowanie

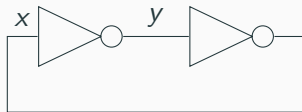
Przykład – alarm

- Aktywacja sensora s włącza alarm
- Ręczne wyłączenie alarmu (reset) r
np. klawiaturką kodową



Układ bramkowy ze stanem

Minimalny przykład:

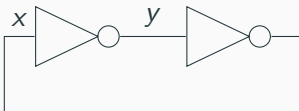


Dwa stany stabilne:

- $x = 0, y = 1$
- $x = 1, y = 0$

Układ bramkowy ze stanem

Minimalny przykład:



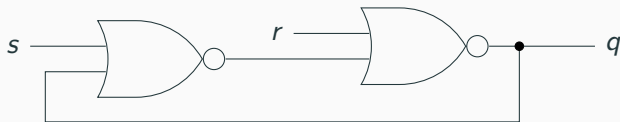
Dwa stany stabilne:

- $x = 0, y = 1$
- $x = 1, y = 0$

Uwaga: w praktyce możliwe stany nieustalone, a nawet oscylacja! (w pewnych sytuacjach)

Przełączalny układ bramkowy ze stanem

Zastępujemy bramki NOT bramkami NOR:



Za pomocą wejść s , r można wpłynąć na stan układu.

Przerzutnik SR

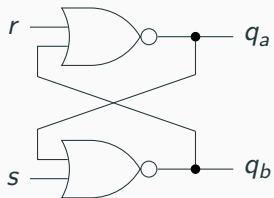


Tabela charakterystyczna:

s	r	q_a	q_b
0	0	q_a	q_b
0	1	0	1
1	0	1	0
1	1	0	0

Przerzutnik SR

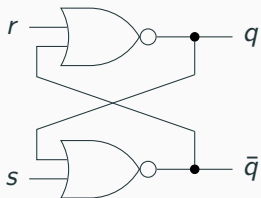
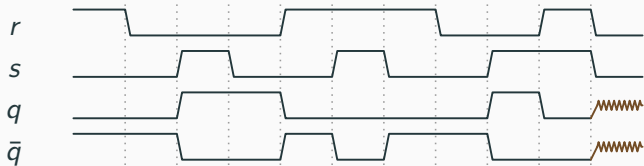


Tabela charakterystyczna:

s	r	q	\bar{q}
0	0	q	\bar{q}
0	1	0	1
1	0	1	0
1	1	0	0



Przerzutnik SR

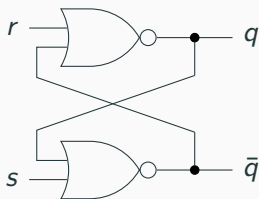
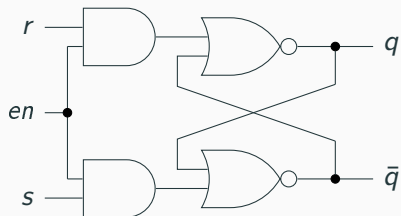


Tabela charakterystyczna:

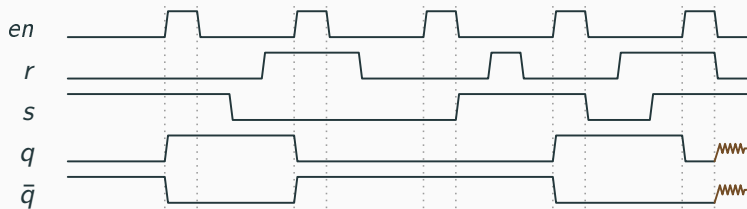
s	r	q	\bar{q}
0	0	q	\bar{q}
0	1	0	1
1	0	1	0
1	1	0	0

```
module sr_latch(output q, nq, input r, s);  
    nor gq(q, r, nq);  
    nor gnq(nq, s, q);  
endmodule
```

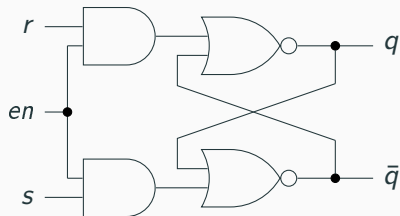
Synchroniczny przerzutnik SR



en	s	r	q	\bar{q}
0	x	x	q	\bar{q}
1	0	0	q	\bar{q}
1	0	1	0	1
1	1	0	1	0
1	1	1	x	x



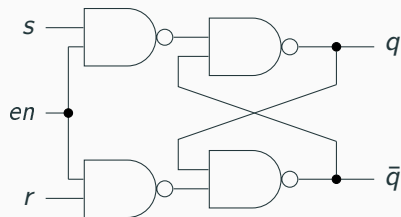
Synchroniczny przerzutnik SR



en	s	r	q	\bar{q}
0	x	x	q	\bar{q}
1	0	0	q	\bar{q}
1	0	1	0	1
1	1	0	1	0
1	1	1	x	x

```
module gated_sr_latch(output q, nq, input en, r, s);  
    logic r1, s1;  
    nor gq(q, r1, nq), gnq(nq, s1, q);  
    and ar(r1, r, en), as(s1, s, en);  
endmodule
```

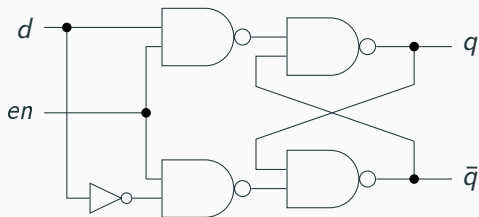
Synchroniczny przerzutnik SR (wersja NAND)



en	s	r	q	\bar{q}
0	x	x	q	\bar{q}
1	0	0	q	\bar{q}
1	0	1	0	1
1	1	0	1	0
1	1	1	x	x

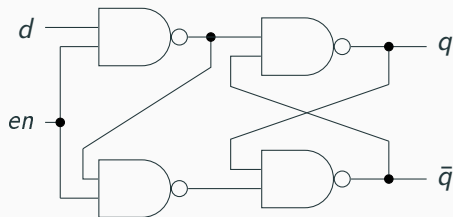
```
module gated_sr_latch(output q, nq, input en, r, s);  
    logic nr, ns;  
    nand gq(q, nr, nq), gnq(nq, ns, q),  
        gr(nr, s, en), gs(ns, r, en);  
endmodule
```

Przerzutnik D (wyzwalany poziomem, zatrzask)

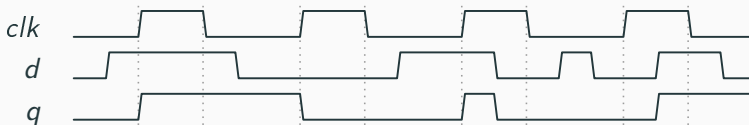


en	d	q
0	x	q
1	0	0
1	1	1

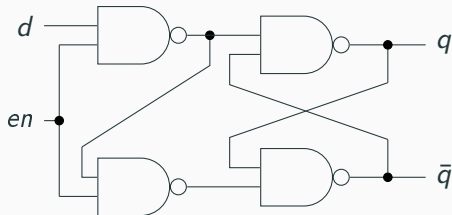
Przerzutnik D (wyzwalany poziomem, zatrzask)



en	d	q
0	x	q
1	0	0
1	1	1



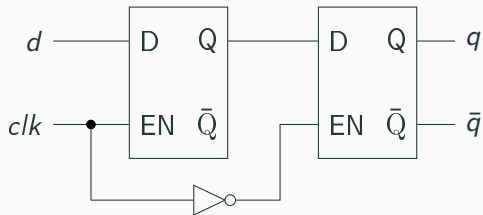
Przerzutnik D (wyzwalany poziomem, zatrzask)



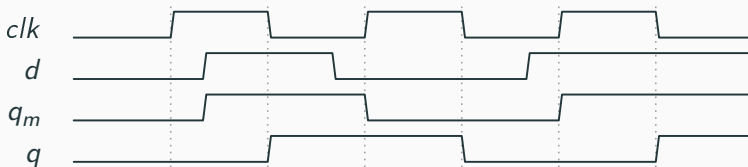
en	d	q
0	x	q
1	0	0
1	1	1

```
module d_latch(output q, nq, input en, d);  
    logic nr, ns;  
    nand gq(q, nr, nq), gnq(nq, ns, q),  
        gr(nr, d, en), gs(ns, nr, en);  
endmodule
```

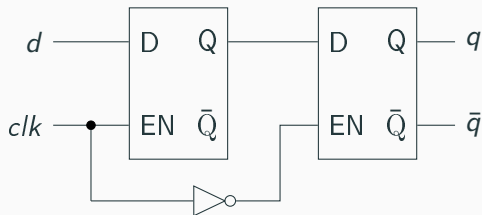

Przerzutnik D typu master-slave (wyzwalany zboczem)



clk	d	q
-	x	q
↓	0	0
↓	1	1



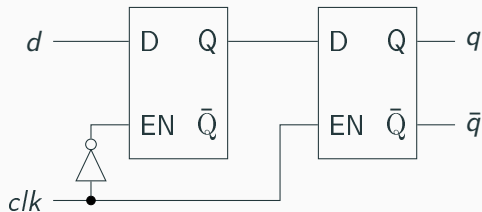
Przerzutnik D typu master-slave (wyzwalany zboczem)



clk	d	q
-	x	q
↓	0	0
↓	1	1

```
module dff_ms(output q, nq, input clk, d);  
    logic q1;  
    d_latch dl1(q1, , clk, d), dl2(q, nq, !clk, q1);  
endmodule
```

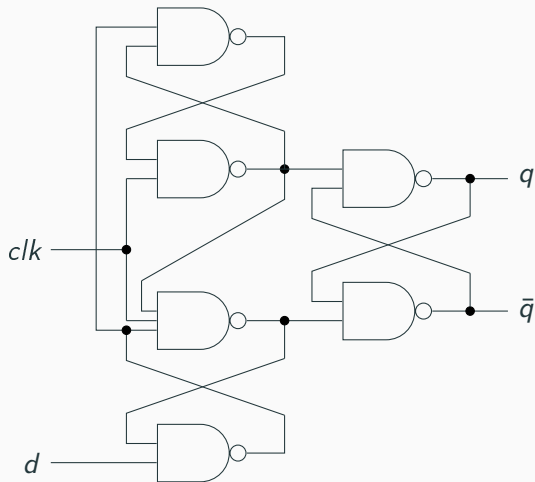
Przerzutnik D typu master-slave (wyzwalany zboczem)



clk	d	q
-	x	q
\uparrow	0	0
\uparrow	1	1

```
module dff_ms(output q, nq, input clk, d);  
    logic q1;  
    d_latch dl1(q1, , !clk, d), dl2(q, nq, clk, q1);  
endmodule
```

Przerzutnik typu D (wyzwalany zboczem narastającym)

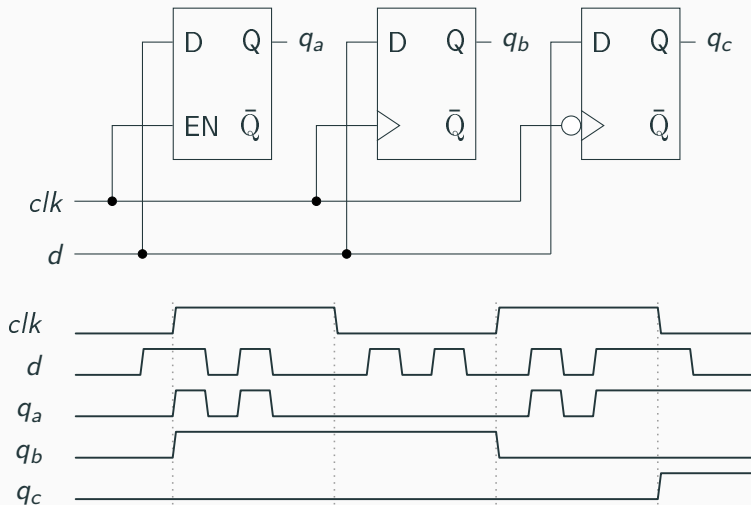


clk	d	q
-	x	q
\uparrow	0	0
\uparrow	1	1

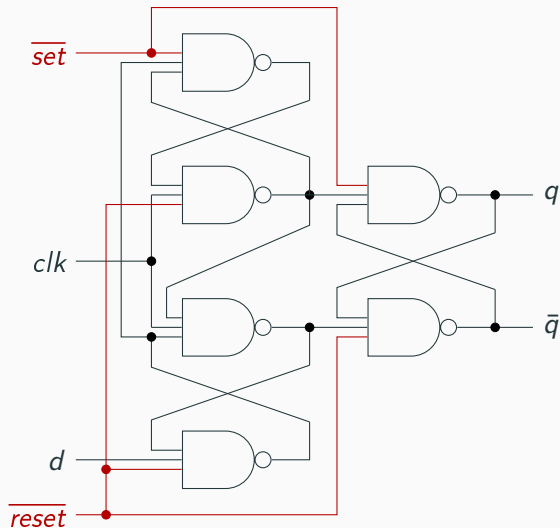
Przerzutnik typu D – implementacja bramkowa

```
module dff(output q, nq, input clk, d);  
    logic r, s, nr, ns;  
    nand gq(q, nr, nq), gnq(nq, ns, q),  
        gr(nr, clk, r), gs(ns, nr, clk, s),  
        gr1(r, nr, s), gs1(s, ns, d);  
endmodule
```

Przerzutniki typu D – porównanie



Przerzutnik D z resetem asynchronicznym

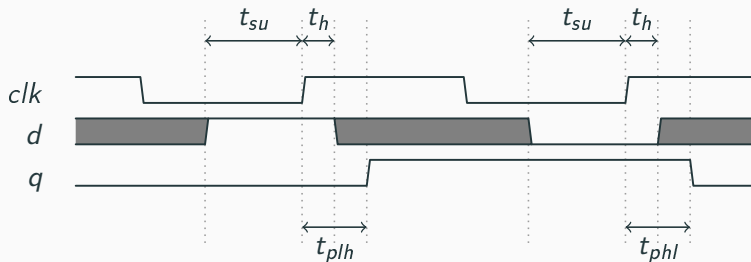


\bar{r}	\bar{s}	c	d	q
0	1	x	x	0
1	0	x	x	1
0	0	x	x	x
1	1	-	x	q
1	1	\uparrow	0	0
1	1	\uparrow	1	1

- *Latch* – przerzutnik asynchroniczny lub wyzwalany poziomem
- *Gated latch* – przerzutnik wyzwalany poziomem
Gated D latch – zatrzask
- *Flip-flop* – przerzutnik wyzwalany zboczem
- *DFF* – skrót od *D flip-flop*
- *Rising edge* – zbocze narastające
- *Falling edge* – zbocze opadające

Właściwości czasowe

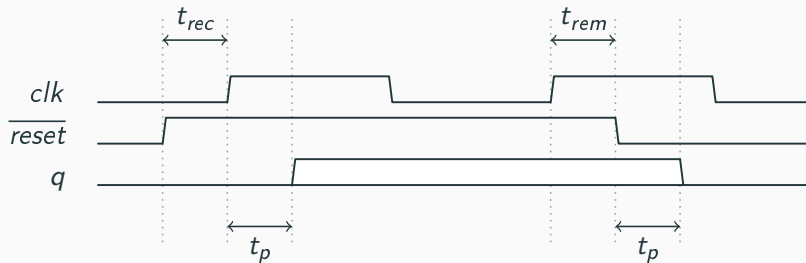
Czas ustalania, podtrzymania, propagacji



- t_{su} – czas ustalania (setup time)
- t_h – czas podtrzymania (hold time)
- t_p – czas propagacji (propagation time)
- t_{plh} – czas propagacji ze stanu niskiego do wysokiego

W trakcie czasu ustalania i podtrzymania sygnał **musi** być stabilny!

Czas powrotu i odwołania



- t_{rec} – czas powrotu (recovery time)
- t_{rem} – czas odwołania (removal time)

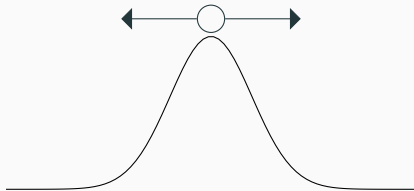
Przykład – 74HC175 (Quad DFF w/reset)

Table 7. Dynamic characteristics ...continued

GND (ground = 0 V); $C_L = 50$ pF unless otherwise specified; for test circuit, see [Figure 10](#)

Symbol	Parameter	Conditions	25 °C			–40 °C to +85 °C		–40 °C to +125 °C		Unit
			Min	Typ	Max	Min	Max	Min	Max	
t_{rec}	recovery time	MR to CP; see Figure 9								
		$V_{CC} = 2.0$ V	5	–33	-	5	-	5	-	ns
		$V_{CC} = 4.5$ V	5	–12	-	5	-	5	-	ns
		$V_{CC} = 6.0$ V	5	–10	-	5	-	5	-	ns
t_{su}	set-up time	Dn to CP; see Figure 7								
		$V_{CC} = 2.0$ V	80	3	-	100	-	120	-	ns
		$V_{CC} = 4.5$ V	16	1	-	20	-	24	-	ns
		$V_{CC} = 6.0$ V	14	1	-	17	-	20	-	ns
t_h	hold time	Dn to CP; see Figure 7								
		$V_{CC} = 2.0$ V	25	2	-	30	-	40	-	ns
		$V_{CC} = 4.5$ V	5	0	-	6	-	8	-	ns
		$V_{CC} = 6.0$ V	4	0	-	5	-	7	-	ns

Metastabilność



Przyczyny:

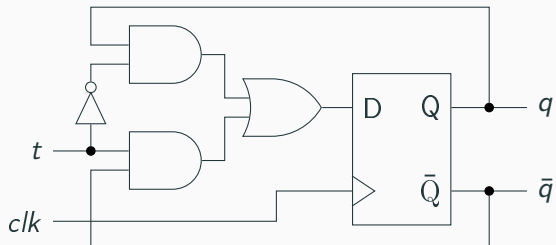
- niestabilny sygnał podczas czasu ustalania lub podtrzymania (przerzutniki D)
- nieprawidłowe wejście (przerzutniki SR)

Środki zaradcze:

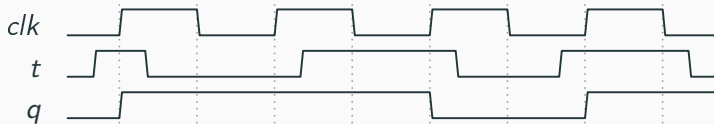
- spowolnienie zegara, skrócenie ścieżki krytycznej
- dodanie synchronizatora (ciągu DFF)

Inne przerzutniki

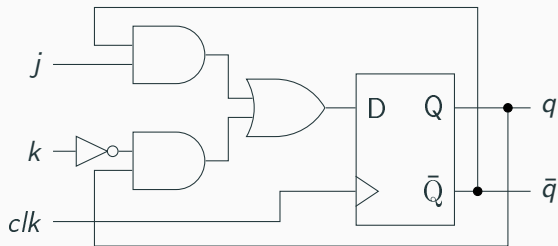
Przerzutnik T



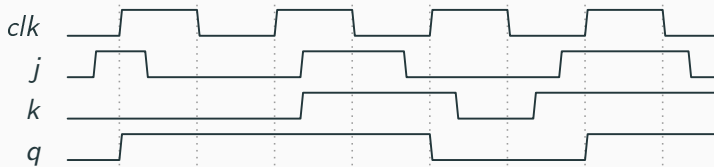
clk	t	q
-	x	q
\uparrow	0	q
\uparrow	1	\bar{q}



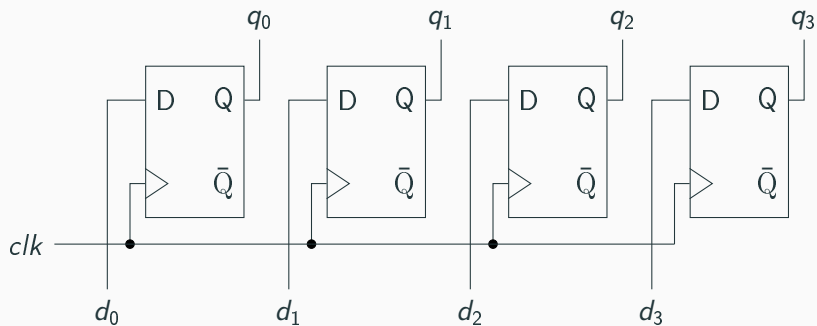
Przerzutnik JK

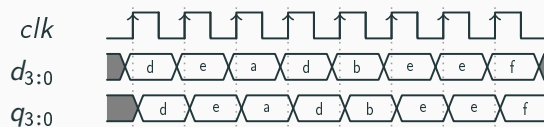
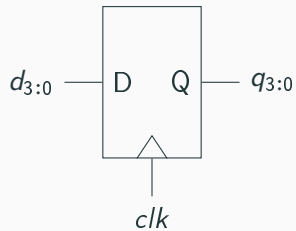


clk	j	k	q
-	x	x	q
\uparrow	0	0	q
\uparrow	0	1	0
\uparrow	1	0	1
\uparrow	1	1	\bar{q}

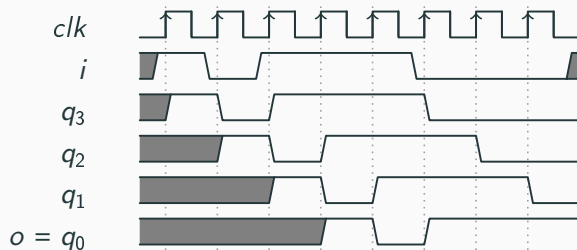
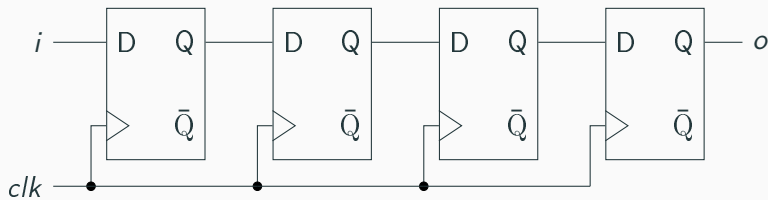


Rejestr

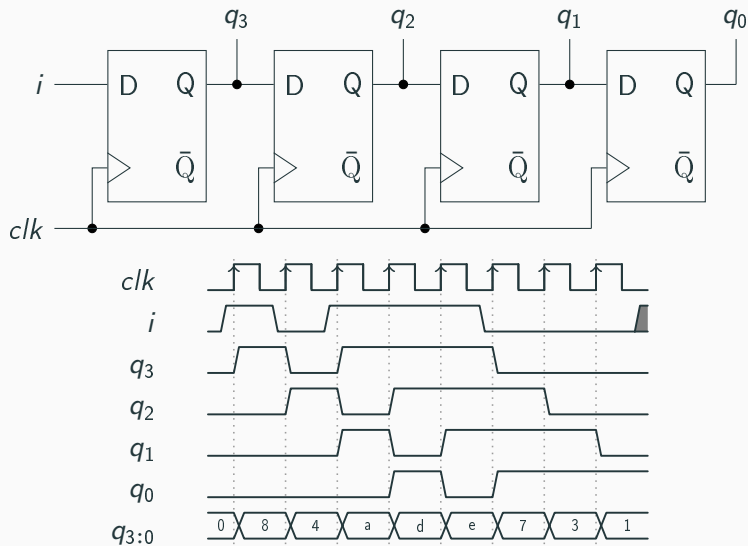




Rejestr przesuwny



Rejestr przesuwny z równoległym odczytem



Rejestr przesuwny z równoległym odczytem i zapisem

