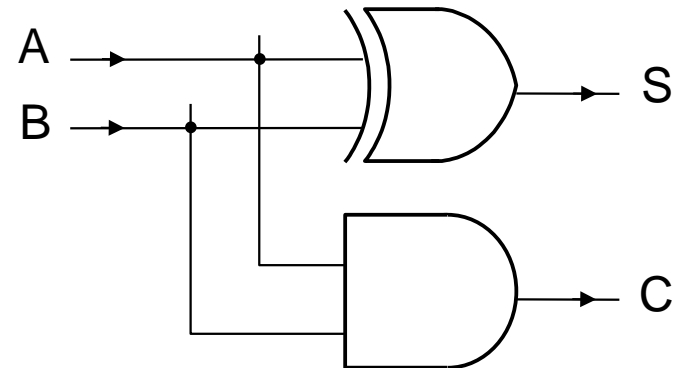
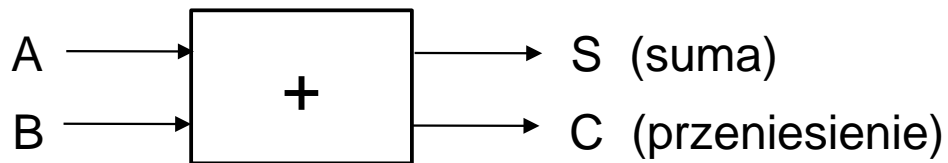


Sumatory

Półsumator

Układ wykonujący dodawanie dwóch jednobitowych liczb binarnych A i B:

$0 + 0 = 0$	$= 00$
$0 + 1 = 1$	$= 01$
$1 + 0 = 1$	$= 01$
$1 + 1 = 10$	$= 10$
A B	CS



Sposób realizacji półsumatora



Sumatory

A	B	S	C
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

C _i	A	B	S	C _o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

A, B – dane wejściowe
C_i – wejście przeniesienia
S – dane wyjściowe (suma)
C_o – wyjście przeniesienia

Sumatory

		A B			
		0 0	0 1	1 1	1 0
C _i	0	0	1	0	1
	1	1	0	1	0

Suma

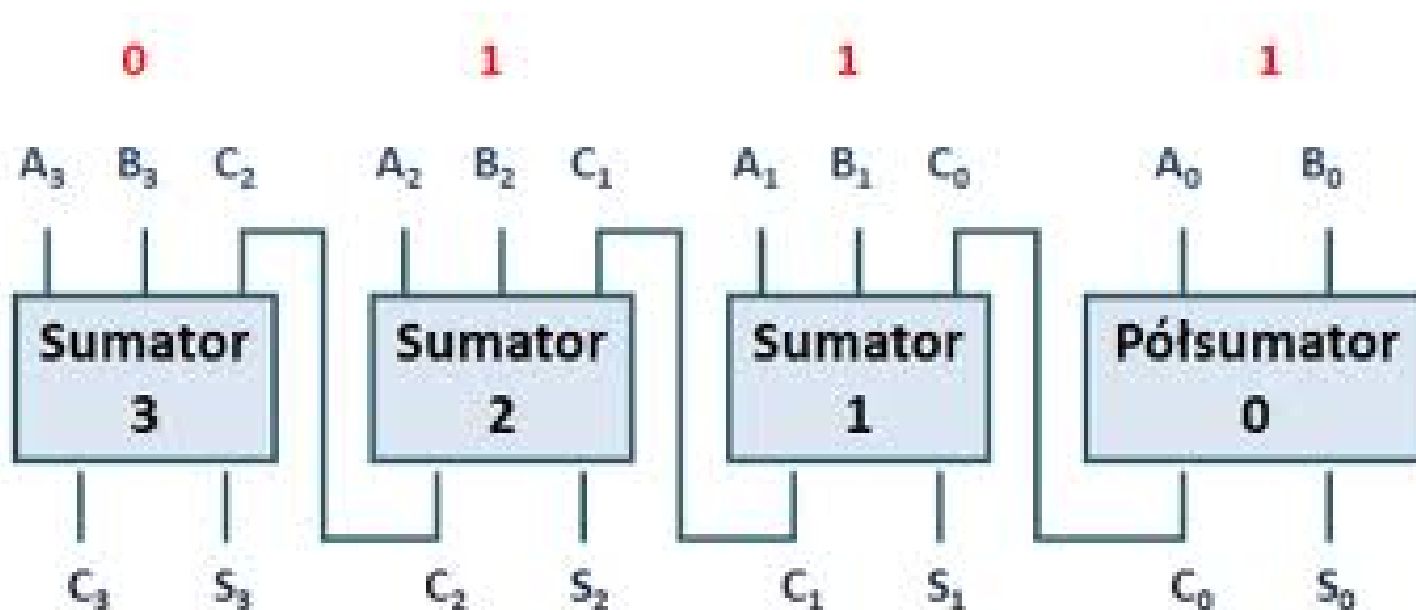
		A B			
		0 0	0 1	1 1	1 0
C _i	0	0	0	1	0
	1	0	1	1	1

Wyjście przeniesienia

$$S = \bar{A} * \bar{B} * C_i + \bar{A} * B * \bar{C}_i + A * B * C + A * \bar{B} * \bar{C}_i$$

$$C_o = CB + CA + AB$$

Sumator 4-bitowy





Sumator 32-bitowy

Można zbudować sumator dla większej ilości bitów złożony z sumatorów 1-bitowych.

Wady takiego rozwiązania: w każdym sumatorze 1-bitowym występuje opóźnienie odpowiedzi względem sygnałów wejściowych. Dla sumatora wielobitowego może być bardzo duże.

Rozwiązanie:

- Określenie wartości przeniesień bez przechodzenia przez wszystkie poprzednie stopnie
- Każdy sumator 1-bitowy działa niezależnie i opóźnienia się nie kumulują

Sumator 32-bitowy

$$C_0 = A_0 B_0 \quad (*)$$

$$C_1 = A_1 B_1 + (A_1 + B_1) C_0 \quad (**)$$

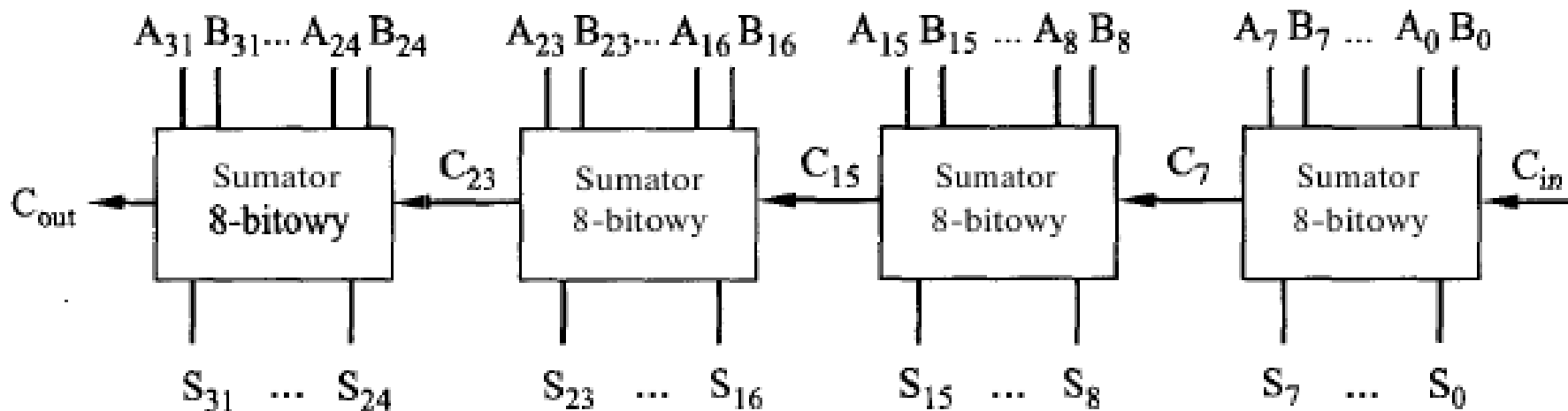
Podstawiając (*) do (**) dostajemy:

$$C_1 = A_1 B_1 + A_1 A_0 B_0 + B_1 A_0 B_0$$

Powtarzając tę procedurę dostajemy kolejne wartości przeniesień . Jednak w przypadku długich liczb to rozwiązanie staje się bardzo skomplikowane.

Sumator 32-bitowy

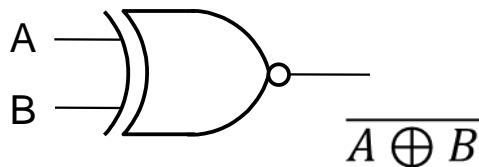
Stosuje się rozwiązania pośrednie. Np. sumator 32-bitowy można zbudować z 4 sumatorów 8-bitowych.



Komparatory

Komparatorem cyfrowym nazywamy układ służący do porównywania dwu lub więcej liczb binarnych. Najważniejsze kryteria porównawcze to $A = B$, $A > B$, $A < B$. Układ sprawdzający wszystkie trzy relacje nazywa się komparatorem uniwersalnym. Najprostsze komparatory umożliwiają jedynie określenie czy dwie porównywane liczby są sobie równe lub która z liczb jest większa.

Kryterium równości dwóch liczb binarnych jest identyczność wszystkich bitów. W przypadku dwóch liczb jednobitowych A i B, informację o tym uzyskać można za pomocą funkcji **negacja EXOR**:



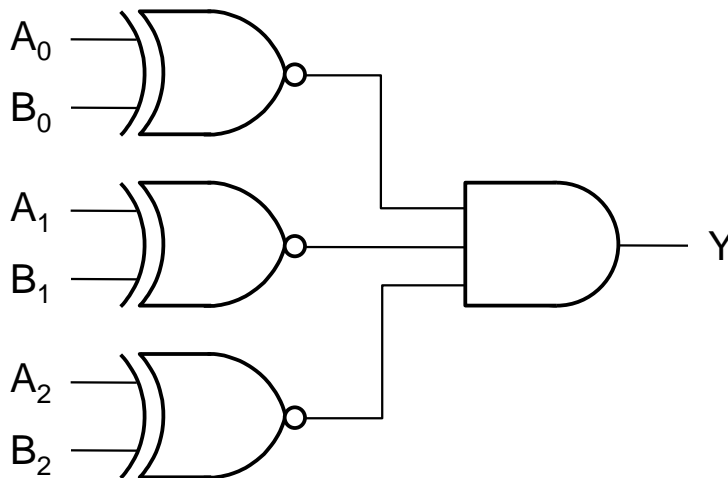
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Wartość 1 na wyjściu sygnalizuje równość $A = B$.

Komparatory

Przykład komparatora równoległego 3 – bitowego

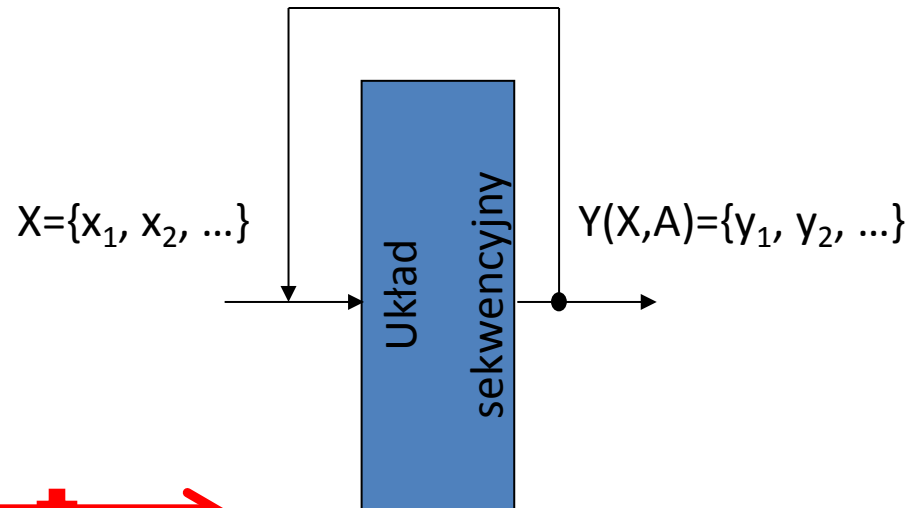
Komparator równoległy to taki układ, na którego wejścia podawane są jednocześnie wszystkie bity porównywanych liczb.



$Y = 1$ tylko wówczas gdy:
 $A_0 = B_0$ i $A_1 = B_1$ i $A_2 = B_2$
czyli $A = B$.

Układ sekwencyjny

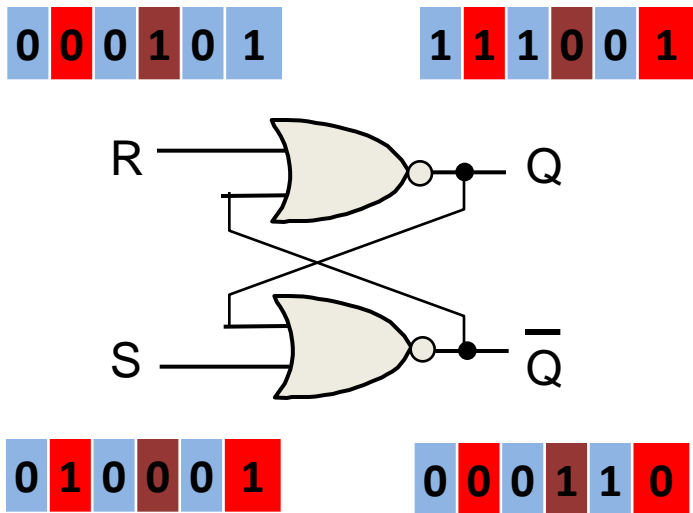
- Stan wyjść zależy stanu wejść i stanu poprzedniego układu
- Przykłady
 - Przerzutnik
 - Rejestr
 - Licznik



$$Y(t_n) = f(X(t_0), X(t_1), \dots, X(t_{n-1}))$$

Przerzutnik RS

schemat logiczny

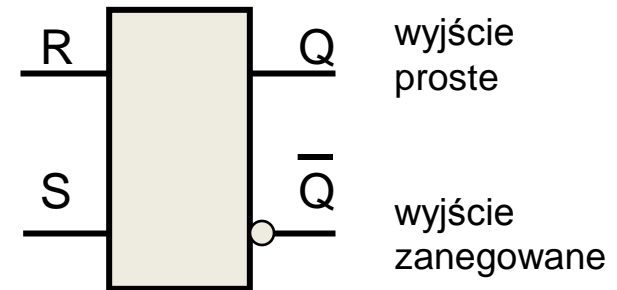


tablica prawdy

R	S	Q_{n+1}
0	0	Q_n
0	1	1
1	0	0
1	1	-

- stan zabroniony

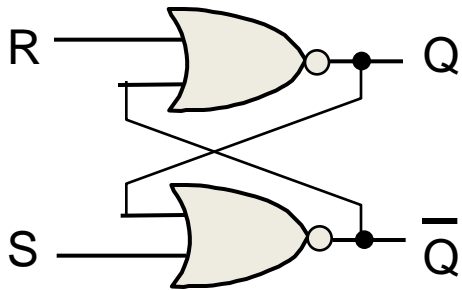
symbol graficzny



S – set
R - reset

Przerzutnik RS

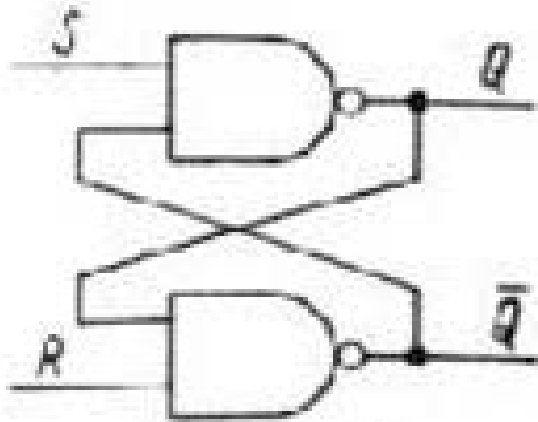
tablica własności (characteristic table)



SR	Q _n	Q _{n+1}
00	0	0
00	1	1
01	0	0
01	1	0
10	0	1
10	1	1
11	0	-
11	1	-

- stan zabroniony

Przerzutnik RS

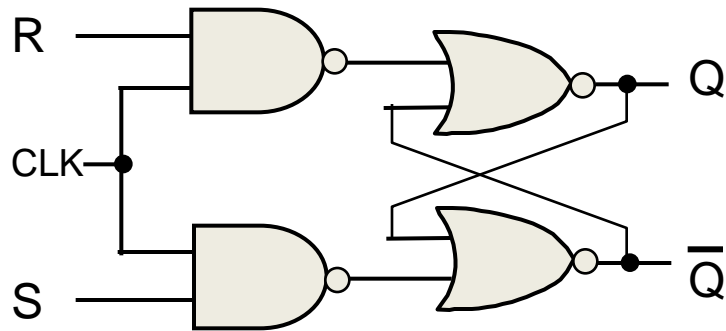


R	S	Q_{n+1}
0	0	-
0	1	0
1	0	1
1	1	Q_n

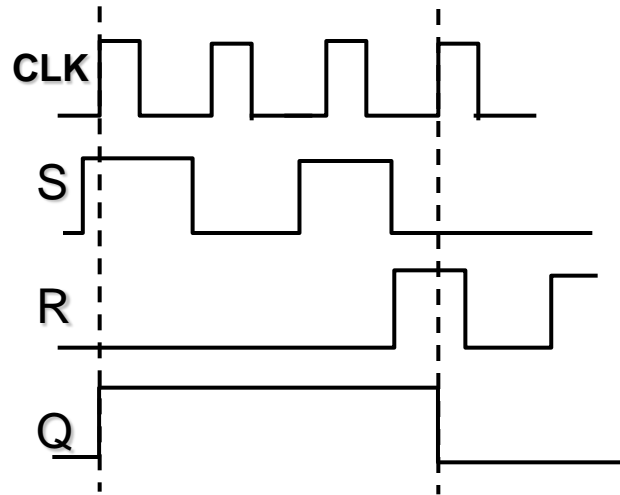
- stan zabroniony

Synchroniczny przerzutnik RS

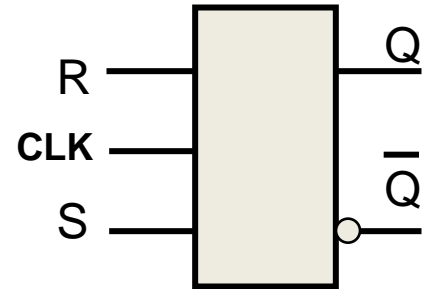
schemat logiczny



wykres czasowy

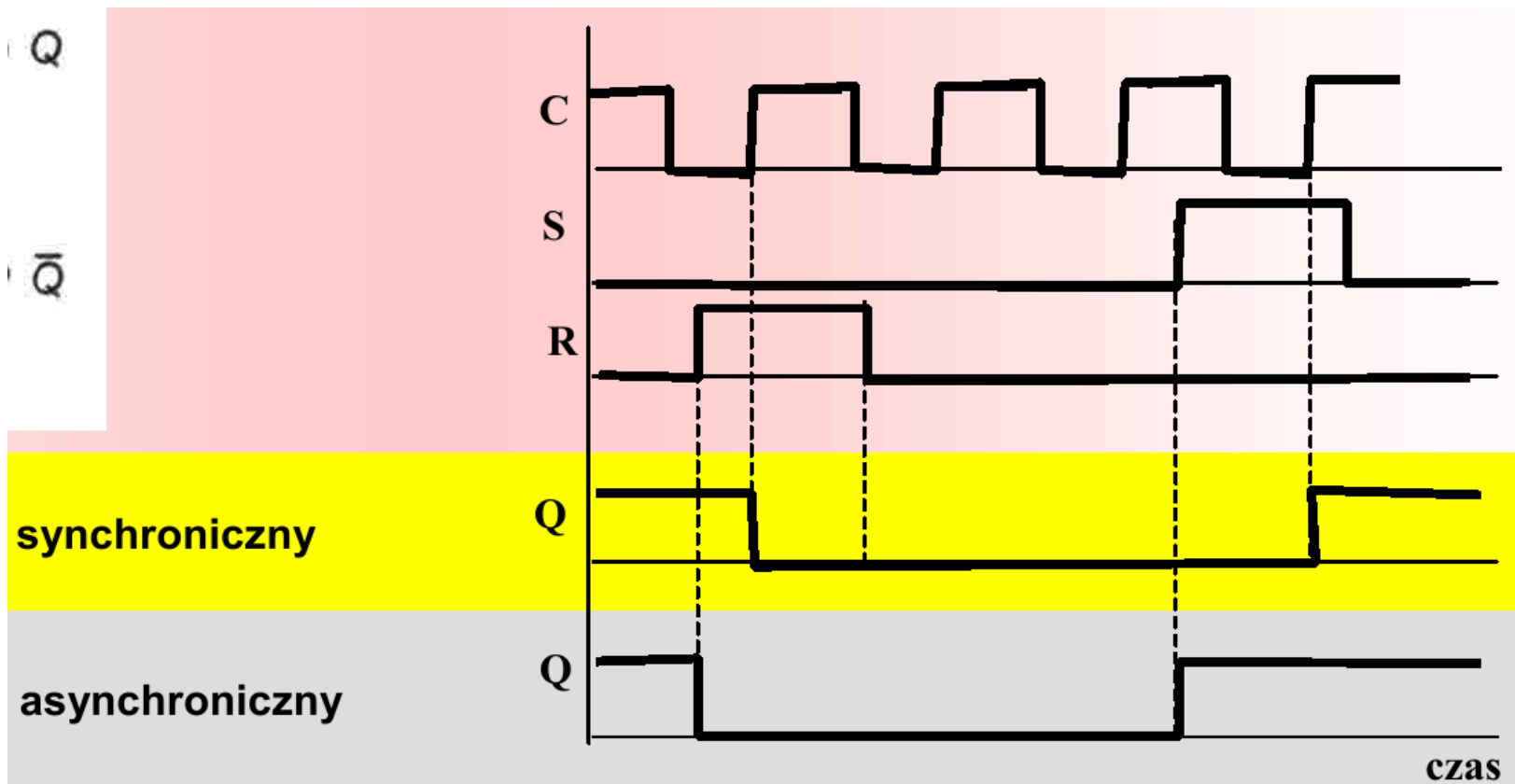


symbol graficzny



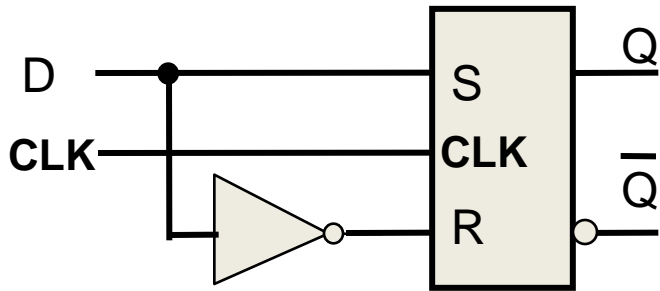
Działanie przerzutnika jest synchronizowane za pomocą impulsów zegarowych

Przerzutniki RS



Przerzutnik D

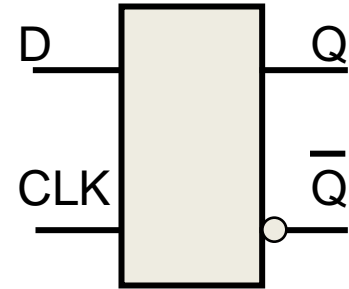
schemat logiczny



tablica prawdy

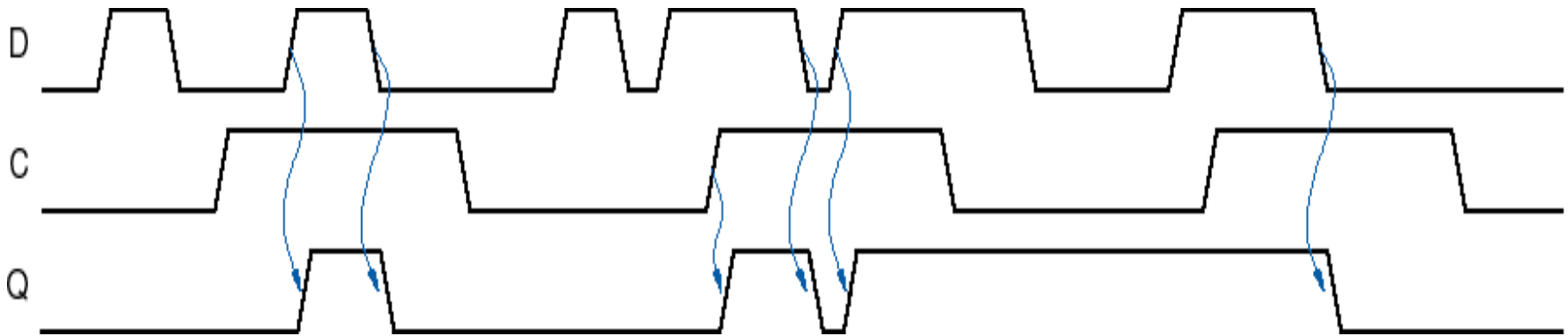
D	CLK	Q_{n+1}
0	0	Q_n
0	1	0
1	0	Q_n
1	1	1

symbol graficzny



Komórka pamiętająca

D-latch operation

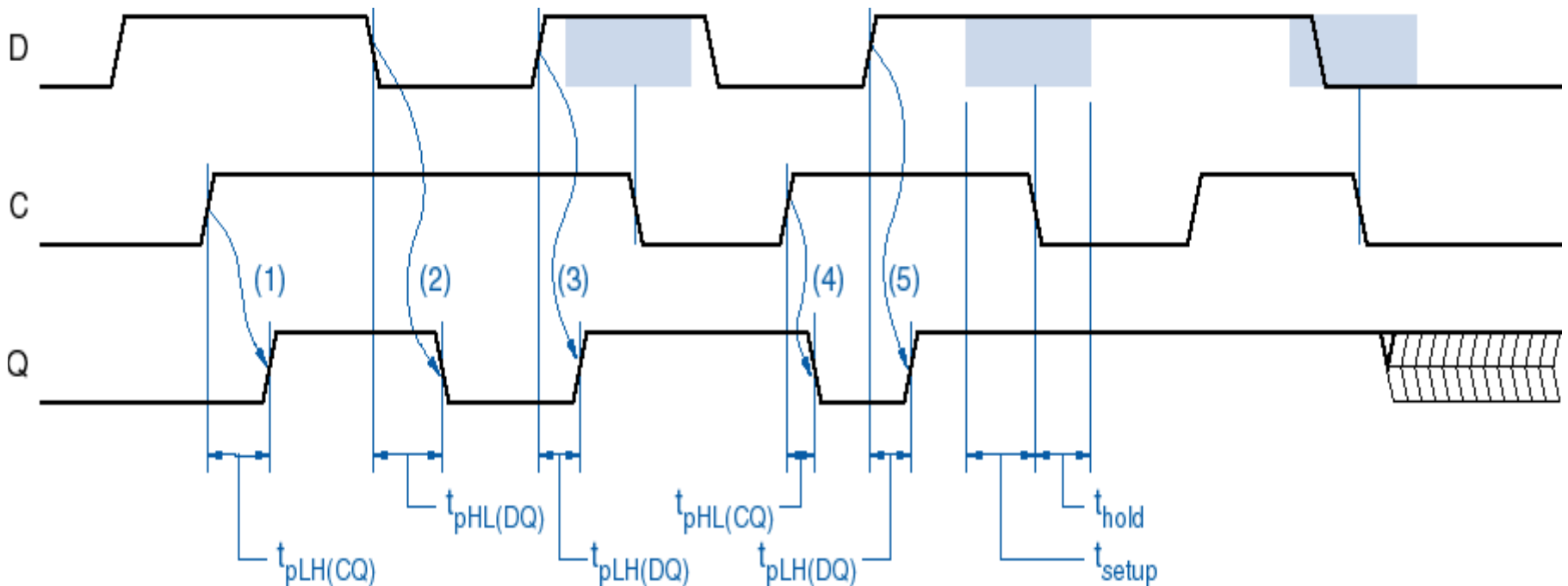


When C is asserted, Q follows the D input, the latch is “open” and the path (D-->Q) is “transparent”.

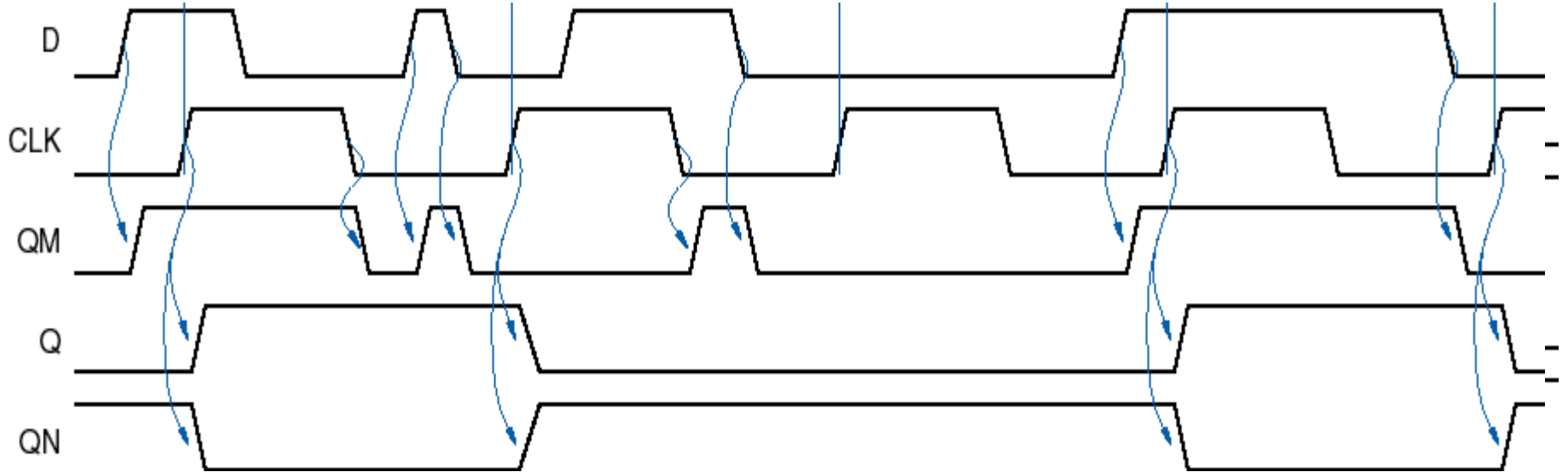
When C is negated, the latch “closes” and Q retains its last value.

D-latch timing parameters

- Propagation delay (from C or D)
- Setup time (D before C edge)
- Hold time (D after C edge)



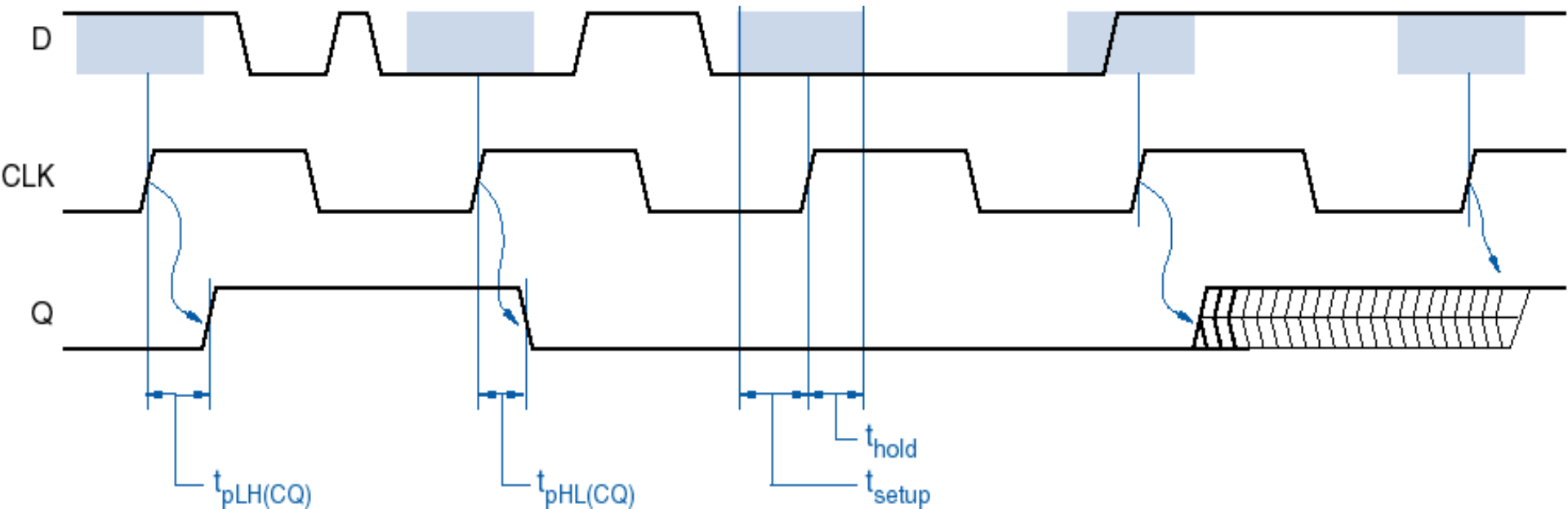
Edge-triggered D flip-flop behavior



D	CLK	Q	QN
0		0	1
1		1	0
x	0	last Q	last QN
x	1	last Q	last QN

D flip-flop timing parameters

- Propagation delay (from CLK)
- Setup time (D before CLK)
- Hold time (D after CLK)

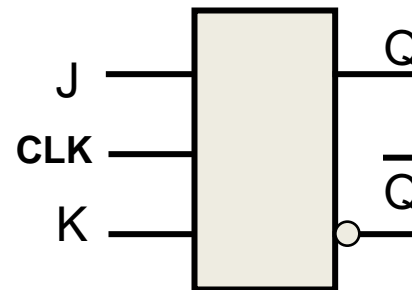


Przerzutnik JK





tablica prawdy

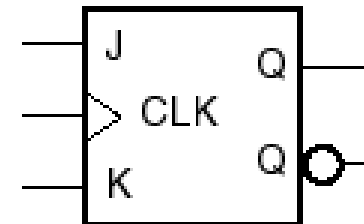
J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

symbol graficzny



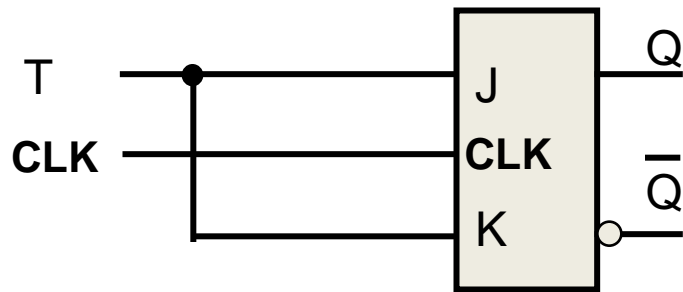
Przerzutnik JK

J	K	CLK	Q	QN
x	x	0	last Q	last QN
x	x	1	last Q	last QN
0	0		last Q	last QN
0	1		0	1
1	0		1	0
1	1		last QN	last Q



Przerzutnik T

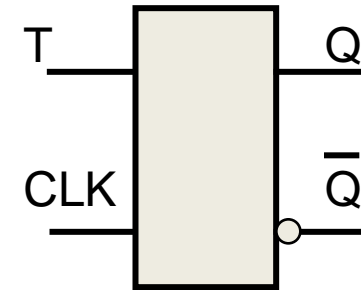
schemat logiczny



tablica prawdy

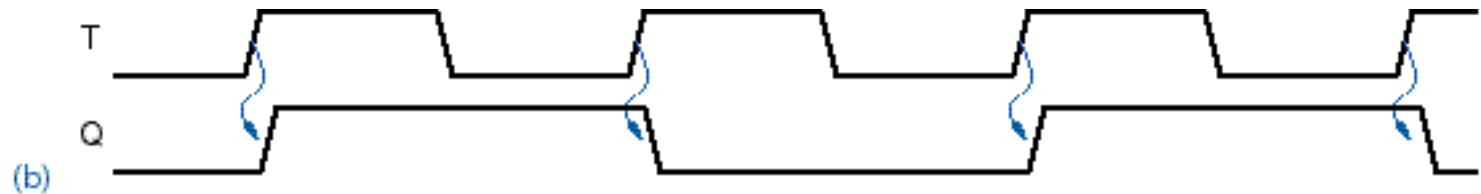
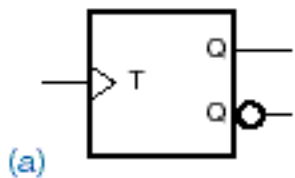
T	Q_{n+1}
0	Q_n
1	\bar{Q}_n

symbol graficzny

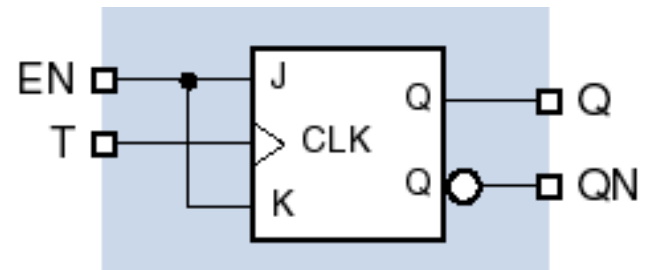
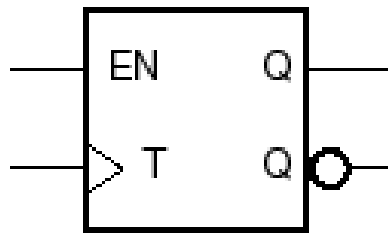
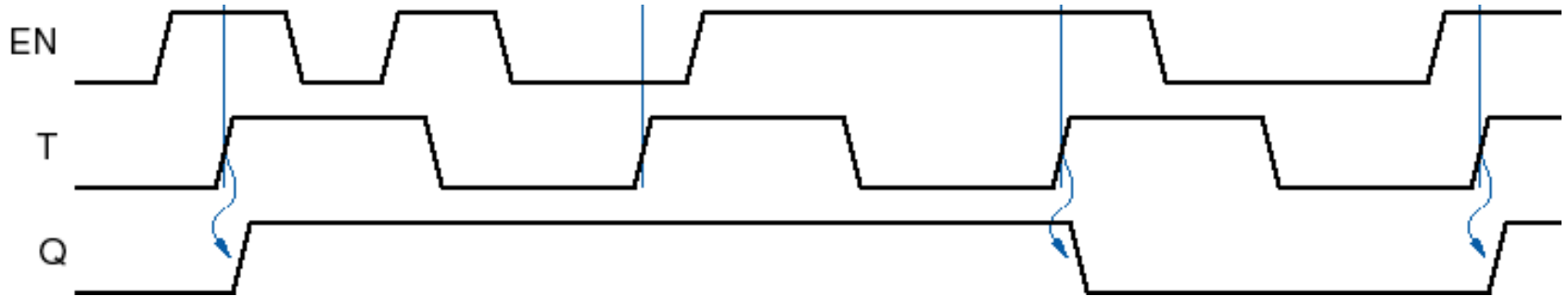


T (toggle) flip-flops

- A T FF changes state on every tick of the clock. (be toggled on every tick)
- Q has precisely half the frequency of the T.
- Important for counters, frequency dividers
- Positive-edge-triggered T FF

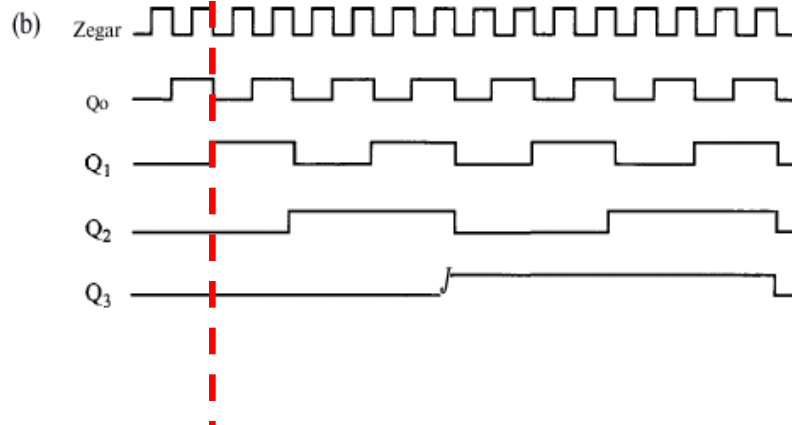
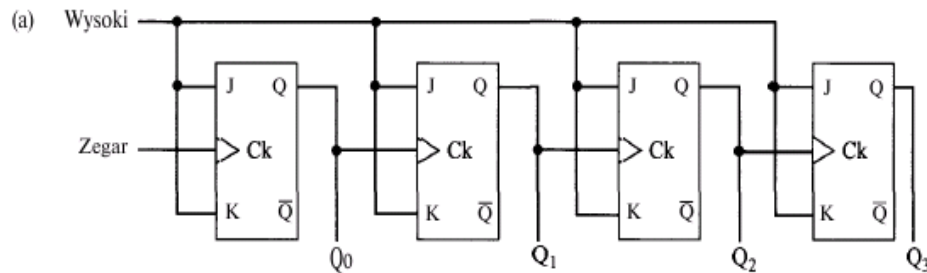


T (toggle) flip-flops with enable



Liczniki

asynchroniczny licznik szeregowy



liczniki wyzwalane
zbroczem opadającym

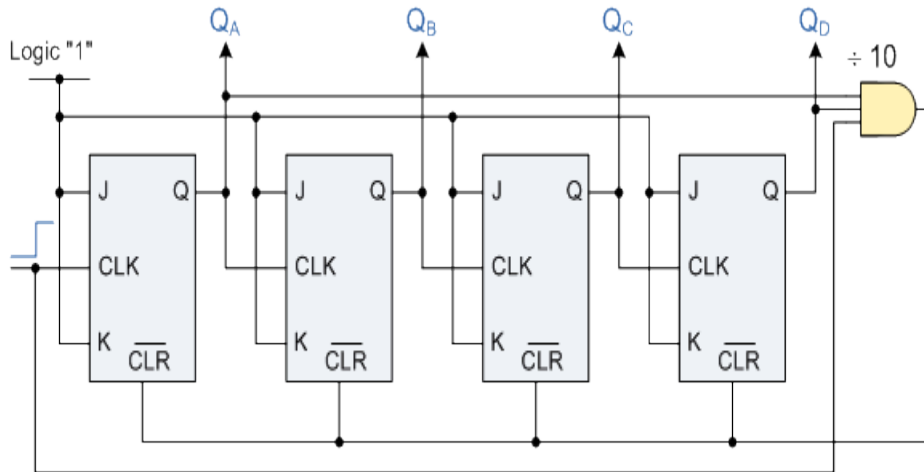
Rejestry których zawartość może być inkrementowana o 1 modulo pojemność rejestru.

n – przerzutników
może liczyć do
 $2^n - 1$

Wada licznika- opóźnienia wprowadzane podczas zmiany wartości proporcjonalne do liczby przerzutników

Liczniki

licznik modulo 10



clk	Qa	Qb	Qc	Qd
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1	1	1	0
8	0	0	0	1
9	1	0	0	1

Rejestry

Rejestry służą do przechowywania informacji cyfrowej zapisanej w kodzie binarnym. Wpisana do rejestru informacja przechowywana jest do chwili wprowadzenia kolejnej, nowej informacji. Informacja ta może być również dostępna do odczytu..

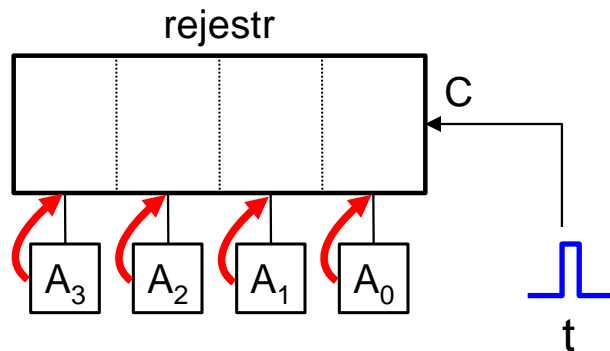
Ze względu na sposób wprowadzania i wyprowadzania informacji rejestry dzielimy na:

- szeregowo – wejście i wyjście szeregowo (rejestry przesuwające)
- równoległe – wejście i wyjście równoległe (rejestry buforowe)
- szeregowo-równoległe – wejście szeregowo, wyjście równoległe
- równoległo-szeregowo – wejście równoległe, wyjście szeregowo.

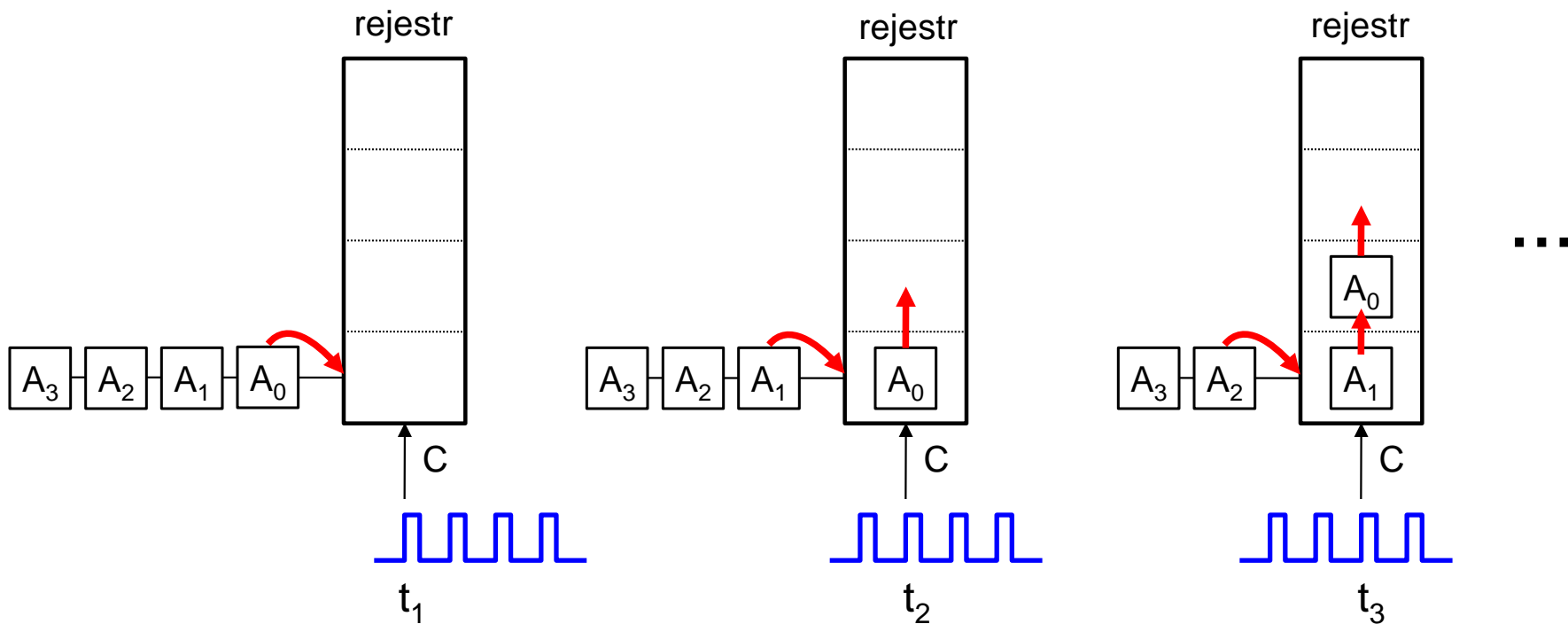
Podstawowym elementem rejestru są przerzutniki.

Liczba bitów informacji jaka może być przechowywana w rejestrze jest nazywana długością rejestru i odpowiada liczbie przerzutników z których jest zbudowany rejestr.

Wprowadzanie równoległe – wszystkie bity słowa informacji wprowadzamy jednocześnie, w jednym takcie zegara:

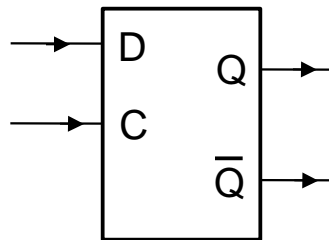


Wprowadzanie szeregowe – słowo wprowadzamy bit po bicie w kolejnych taktach zegara:



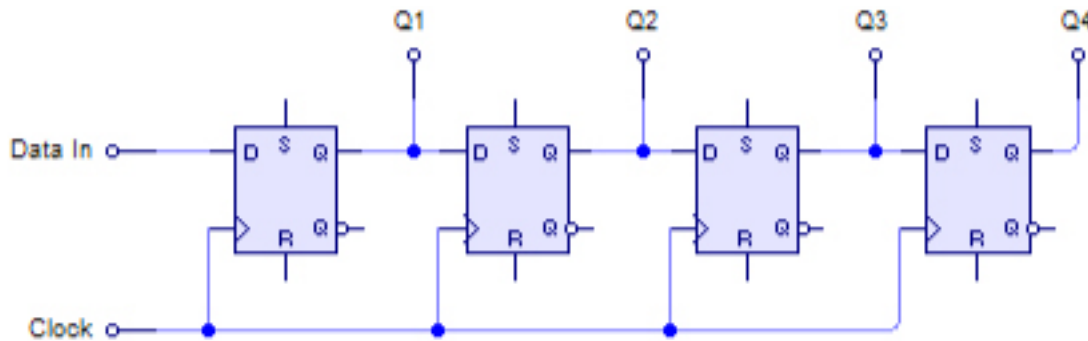
Rejestry

Najprostszym rejestrem jest przerzutnik D.



Zestawienie kilku takich przerzutników, np. 8, bez żadnych połączeń pomiędzy nimi utworzy 8 – bitowy rejestr równoległy (sygnał zegarowy wspólny dla wszystkich przerzutników).

Rejestry



wejście szeregowe,
wyjście równoległe

SIPO

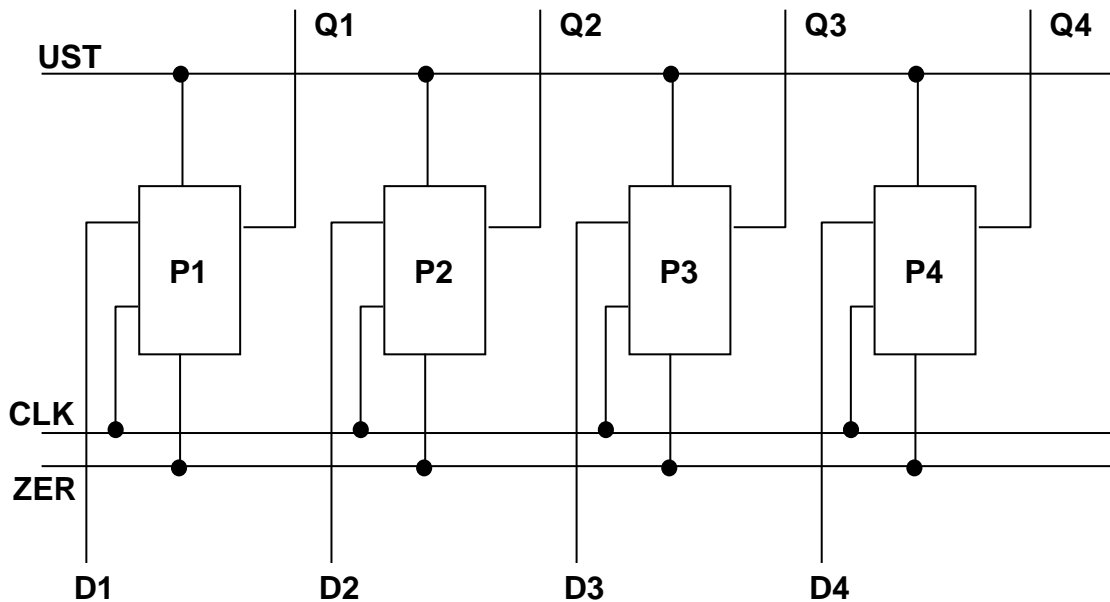
Za każdym impulsem zegarowym dane są przesuwane w o jedną pozycję w prawo

we	stan	wy
0	0 0 0 0	0
1	1 0 0 0	0
0	0 1 0 0	0
0	0 0 1 0	0
0	0 0 0 1	1
0	0 0 0 0	0

Rejestry

Układy do przechowywania 1 lub wielu bitów danych.

rejestry: równoległe i przesuwające



wejście równoległe,
wyjście równoległe

PIPO