

# Temat 4. Podstawowe bloki funkcjonalne

---

*Spis treści do tematu 4*

4.1. Cyfrowe bloki komutacyjne

4.2. Przerzutniki

4.3. Liczniki

4.4. Rejestry

4.5. Układy arytmetyczne

4.6. Literatura

# 4.1. Cyfrowe bloki komutacyjne

Cyfrowe bloki komutacyjne są układami kombinacyjnymi, służącymi do przełączania (komutacji) dróg sygnałów cyfrowych pomiędzy pozostałymi blokami funkcjonalnymi.

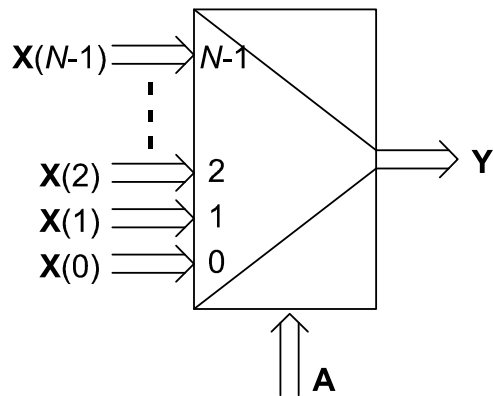
## 4.1.1. Multiplexer

Multiplexer służy do wybierania jednego z wielu słów wejściowych i przesyłania go na wyjście. Na wejścia danych multiplexera podany jest zespół  $N$  słów  $n$ -bitowych

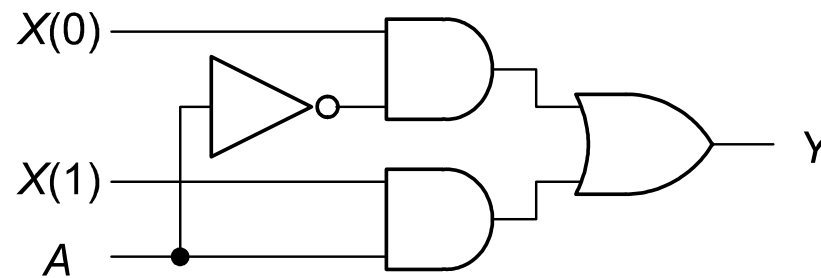
$$\mathbf{X}(0), \mathbf{X}(1), \dots, \mathbf{X}(N-1). \tag{4.1}$$

Do określenia wejścia danych, którego stan jest przesyłany na wyjście  $\mathbf{Y}$ , służy liczba kodowana w NKB (naturalnym kodzie binarnym) i wprowadzana na wejście adresowe  $A$

$$\mathbf{Y} = \mathbf{X}(A) . \tag{4.2}$$



Rys. 4.1. Symbol multiplexera.

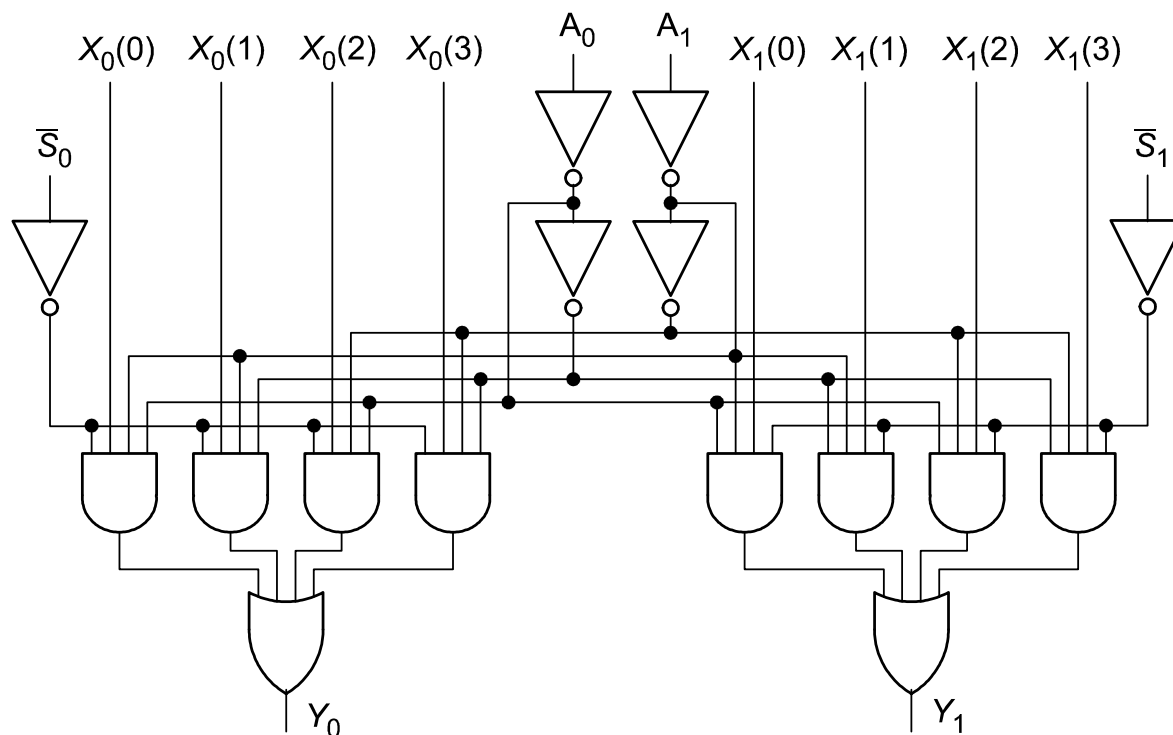


Rys. 4.2. Przykład 1-bitowego multiplexera 1 z 2. T4-2

**Tabela 4.1.** Przykłady multiplekserów wykonywanych jako układy scalone z rodziny 74xx.

symbol. kat.	funkcja	wejście/a strobowania	wyjście/a
74150	jeden mult. 1 z 16	zanegowane	zanegowane
74151	jeden mult 1 z 8	zanegowane	proste i zanegowane
74153	dwa mult. 1 z 4	dwa niezależne zanegowane	proste
74157	cztery mult. 1 z 2	zanegowane wspólne	proste

Multipleksery w układach 74153 i 74157 mają wspólne wejście adresowe.



$$\begin{aligned} Y_0 &= S_0 \cdot X_0(A), \\ Y_1 &= S_1 \cdot X_1(A), \end{aligned} \quad (4.3)$$

gdzie  $S_0$ ,  $S_1$  – sygnały strobowania. Na wejście układu 74153 należy doprowadzić zanegowane sygnały strobowania.

**Rys. 4.3.** Schemat podwójnego multipleksera 1 z 4 w układzie 74153.

## Typowe zastosowania multiplekserów:

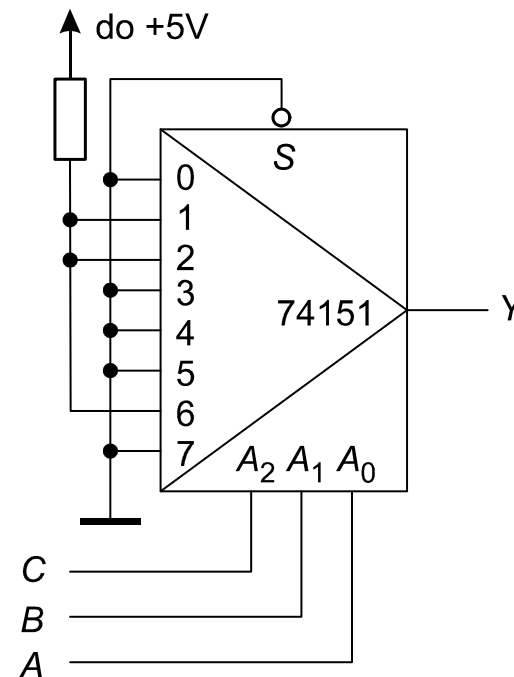
- wybór jednego z wielu źródeł danych (np. sekwencyjne przeglądanie styków klawiatury),
- zamiana postaci informacji z równoległej na szeregową (np. w multipleksowych systemach przesyłania danych,
- realizacja prostych układów kombinacyjnych.

## Przykład wykorzystania multipleksera jako uniwersalnego układu kombinacyjnego

Zadanie: wykonać układ kombinacyjny o 3 wejściach  $A$ ,  $B$  i  $C$  i jednym wyjściu  $Y$ , który pracuje zgodnie z podaną tablicą prawdy.

$C$	$B$	$A$	$Y$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

**Tabela. 4.2.** Tablica prawdy do zadania.



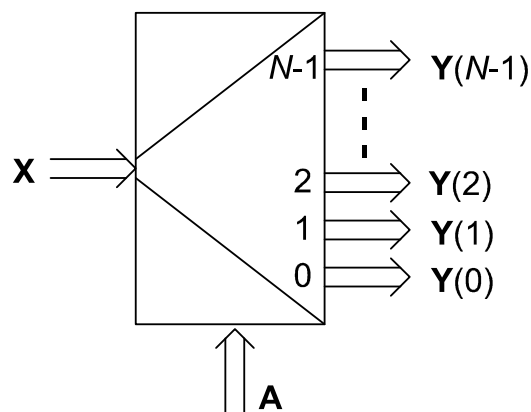
**Rys. 4.4.** Realizacja układu komb. zgodnego z tabelą 4.2. przy użyciu układu scalonego 74151.

## 4.1.2. Demultiplekser

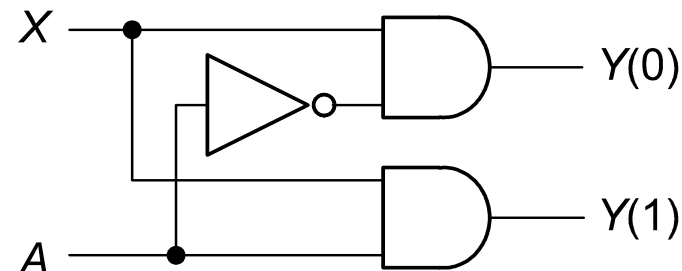
Demultiplekser służy do przesyłania słowa wejściowego  $\mathbf{X}$  na jedno z  $N$  wyjść, przy czym numer tego wyjścia jest określony przez słowo adresowe  $A$ . Na pozostałych wyjściach jest utrzymywany stan 0 (dotyczy wyjść bez negacji)

$$\mathbf{Y}(i) = \begin{cases} \mathbf{X} & \text{gdy } i = A, \\ 0 & \text{gdy } i \neq A. \end{cases} \quad (4.4)$$

**Uwaga:** w demultiplekserach dostępnych jako gotowe układy scalone wyjścia  $\mathbf{Y}(i)$  są zazwyczaj zanegowane.



Rys. 4.5. Symbol multipleksera.

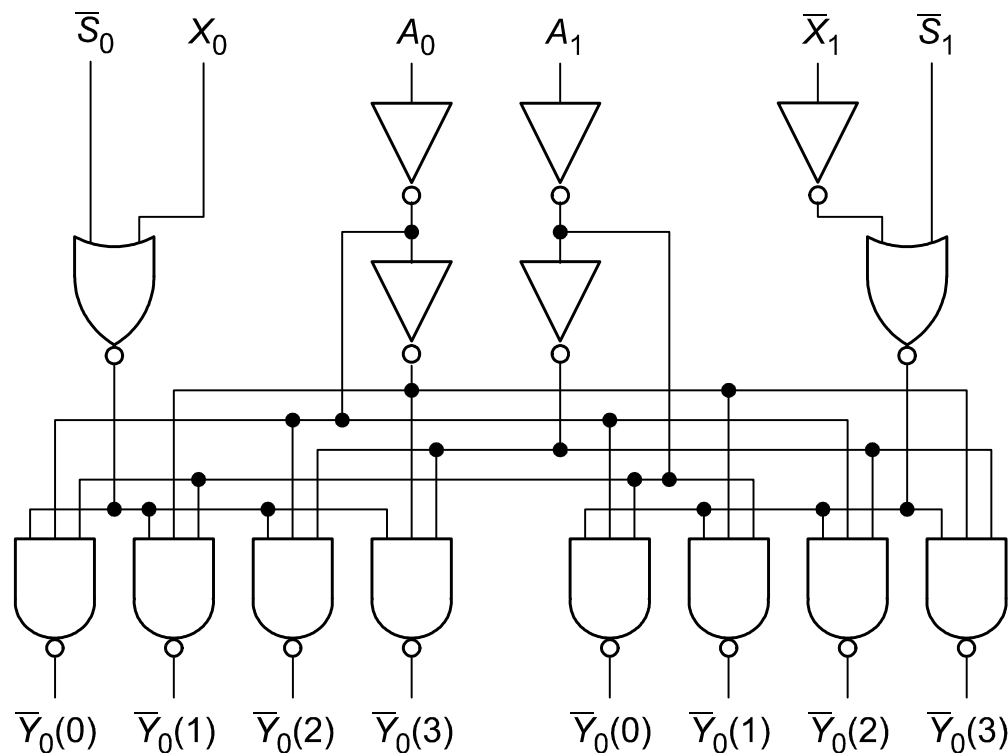


Rys. 4.6. Przykład demultipleksera z 1 na 2 linie.

**Tabela 4.3.** Przykłady demultiplekserów wykonywanych jako układy scalone z rodziny 74xx.

symbol. kat.	funkcja	wejście/a strobowania	wyjścia
74138	jeden dem. 1 na 8	zanegowane	zanegowane
74154	jeden dem. 1 na 16	zanegowane	zanegowane
74155	dwa dem. 1 na 4	dwa niezależne zanegowane	zanegowane
74156	dwa dem. 1 na 4	dwa niezależne zanegowane	Zanegowane, otw. kolekt.

Demultipleksery w układach 74155 i 74156 mają wspólne wejście adresowe.



Uwaga: w jednym z dwóch demultiplekserów wejście danych jest zanegowane.

$S_0, S_1$  – sygnały strobowania.  
Gdy  $\bar{S} = 1$ , to wszystkie wyjścia danego demultipleksera są w stanie 1.

**Rys. 4.7.** Schemat podwójnego demultipleksera z 1 na 4 linie w układzie 74155.

## Typowe zastosowania demultiplekserów:

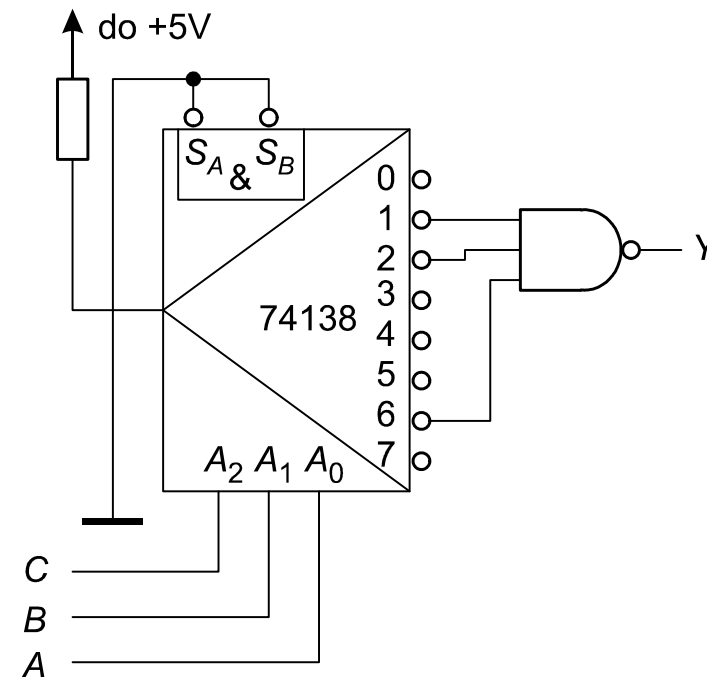
- wybór jednego z wielu odbiorników sygnału z danego źródła,
- zamiana postaci informacji z szeregowej na równoległą na (np. w multipleksowych systemach przesyłania danych),
- realizacja prostych układów kombinacyjnych.

## Przykład wykorzystania układu demultipleksera i bramki NAND jako uniwersalnego układu kombinacyjnego

Zadanie: wykonać układ kombinacyjny o 3 wejściach  $A$ ,  $B$  i  $C$  i jednym wyjściu  $Y$ , który pracuje zgodnie z podaną tablicą prawdy.

$C$	$B$	$A$	$Y$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Tablica prawdy do zadania.  
(kopia tabeli 4.2)



**Rys. 4.8.** Realizacja układu komb. zgodnego z tabelą 4.2. przy użyciu układu scalonego 74151.

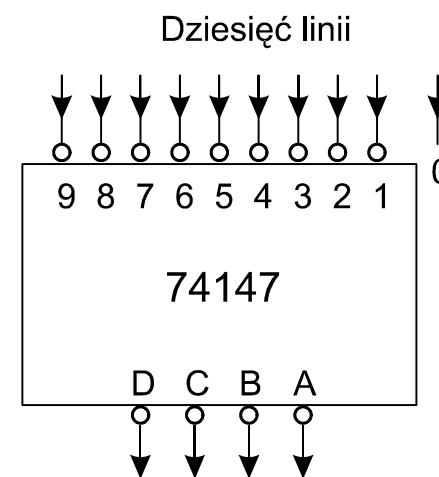
## 4.1.3. Kodery

Koderem nazywany układ, który przetwarza informację wyrażoną w kodzie „1 z  $N$ ” na równoważną informację wyrażoną w innym kodzie. Jeżeli w **koderze priorytetowym** wyróżnione jest więcej niż jedno wejście, to stan wyjść wskazuje na wejście o najwyższym priorytecie.

Układ scalony 74147 jest koderem 1 z 10 na jedną cyfrę kodu BCD (*ang. binary-coded decimal*, czyli zapis dziesiętny kodowany dwójkowo). Wejścia i wyjścia układu są zanegowane. Priorytet wejścia rośnie wraz z numerem wejścia.

9	8	7	6	5	4	3	2	1	$D$	$C$	$B$	$A$
1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	0	–	1	1	0	1
1	1	1	1	1	1	0	–	–	1	1	0	0
1	1	1	1	1	0	–	–	–	1	0	1	1
1	1	1	1	0	–	–	–	–	1	0	1	0
1	1	1	0	–	–	–	–	–	1	0	0	1
1	1	0	–	–	–	–	–	–	1	0	0	0
1	0	–	–	–	–	–	–	–	0	1	1	1
0	–	–	–	–	–	–	–	–	0	1	1	0

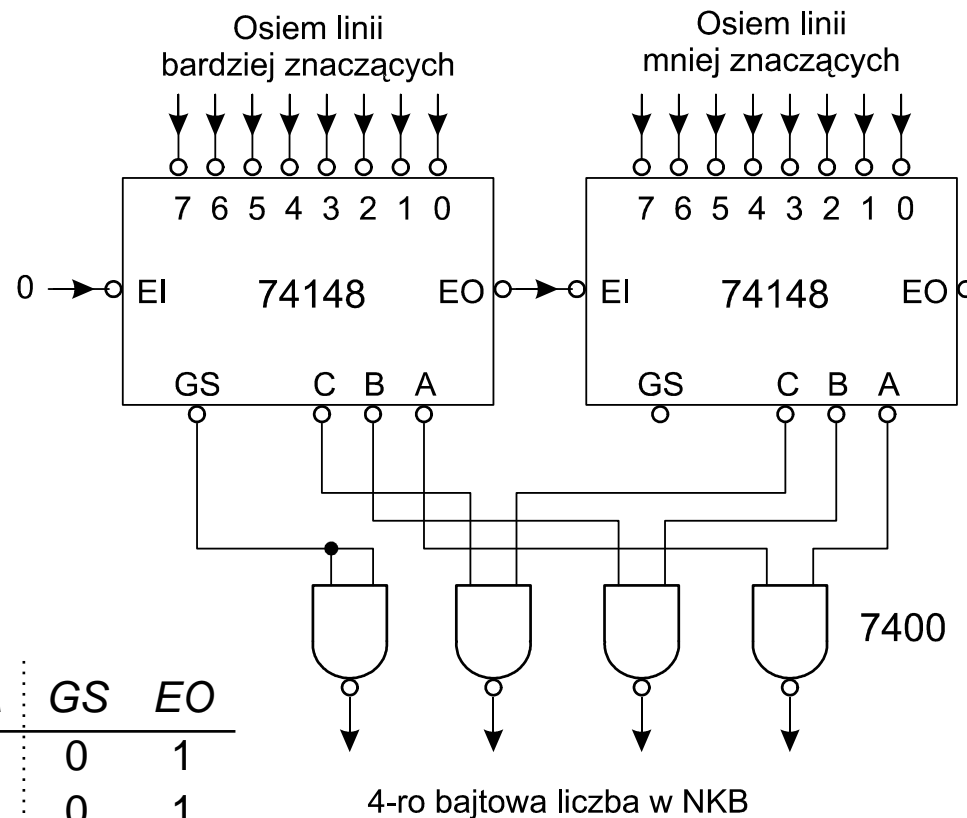
**Tabela. 4.4.** Tablica stanów kodera 74147.



**Rys. 4.9.** Symbol kodera 74147.



Układ scalony 74148 jest koderem 1 z 8 o zanegowanych wejściach i zanegowanych wyjściach, w którym priorytet rośnie wraz z numerem wejścia. Układ posiada dodatkowe wejście EI (*enable input*) oraz wyjścia EO (*enable output*) i GS (*group strobe*), które umożliwiają łączenie koderów.



**Rys. 4.10.** Schemat koderu priorytetowego 1 z 16 zbudowanego z dwóch koderów 1 z 8.

<i>EI</i>	7	6	5	4	3	2	1	0	<i>C</i>	<i>B</i>	<i>A</i>	<i>GS</i>	<i>EO</i>
0	1	1	1	1	1	1	1	0	1	1	1	0	1
0	1	1	1	1	1	1	0	–	1	1	0	0	1
0	1	1	1	1	1	0	–	–	1	0	1	0	1
0	1	1	1	1	0	–	–	–	1	0	0	0	1
0	1	1	1	0	–	–	–	–	0	1	1	0	1
0	1	1	0	–	–	–	–	–	0	1	0	0	1
0	1	0	–	–	–	–	–	–	0	0	1	0	1
0	0	–	–	–	–	–	–	–	0	0	0	0	1
1	–	–	–	–	–	–	–	–	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0

**Tabela. 4.5.** Tablica stanów pojedynczego koderu 74148.

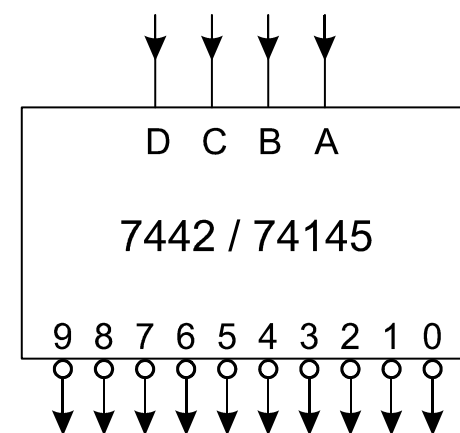
## 4.1.4. Dekodery

Dekoderem nazywany układ, który przetwarza informację wyrażoną w dowolnym kodzie na równoważną informację wyrażoną w kodzie „1 z  $N$ ”.

Układ scalony 7442 jest dekodrem jednej cyfry dziesiętnej w kodzie BCD na kod 1 z 10. Wyjścia układu są zanegowane. Układ 74145 jest funkcjonalnym odpowiednikiem układu 7442, z wyjątkiem wyjść które są typu otwarty kolektor.

<i>D</i>	<i>C</i>	<i>B</i>	<i>A</i>	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	1	1	1	0	1
0	0	1	0	1	1	1	1	1	1	1	0	1	1
0	0	1	1	1	1	1	1	1	1	0	1	1	1
0	1	0	0	1	1	1	1	1	0	1	1	1	1
0	1	0	1	1	1	1	1	0	1	1	1	1	1
0	1	1	0	1	1	1	0	1	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1	1	1	1
1	0	0	0	1	0	1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	1	1	1	1	1	1	1
st. zabronione				1	1	1	1	1	1	1	1	1	1

**Tabela. 4.6.** Tablica stanów dekodera 7442 /74145.



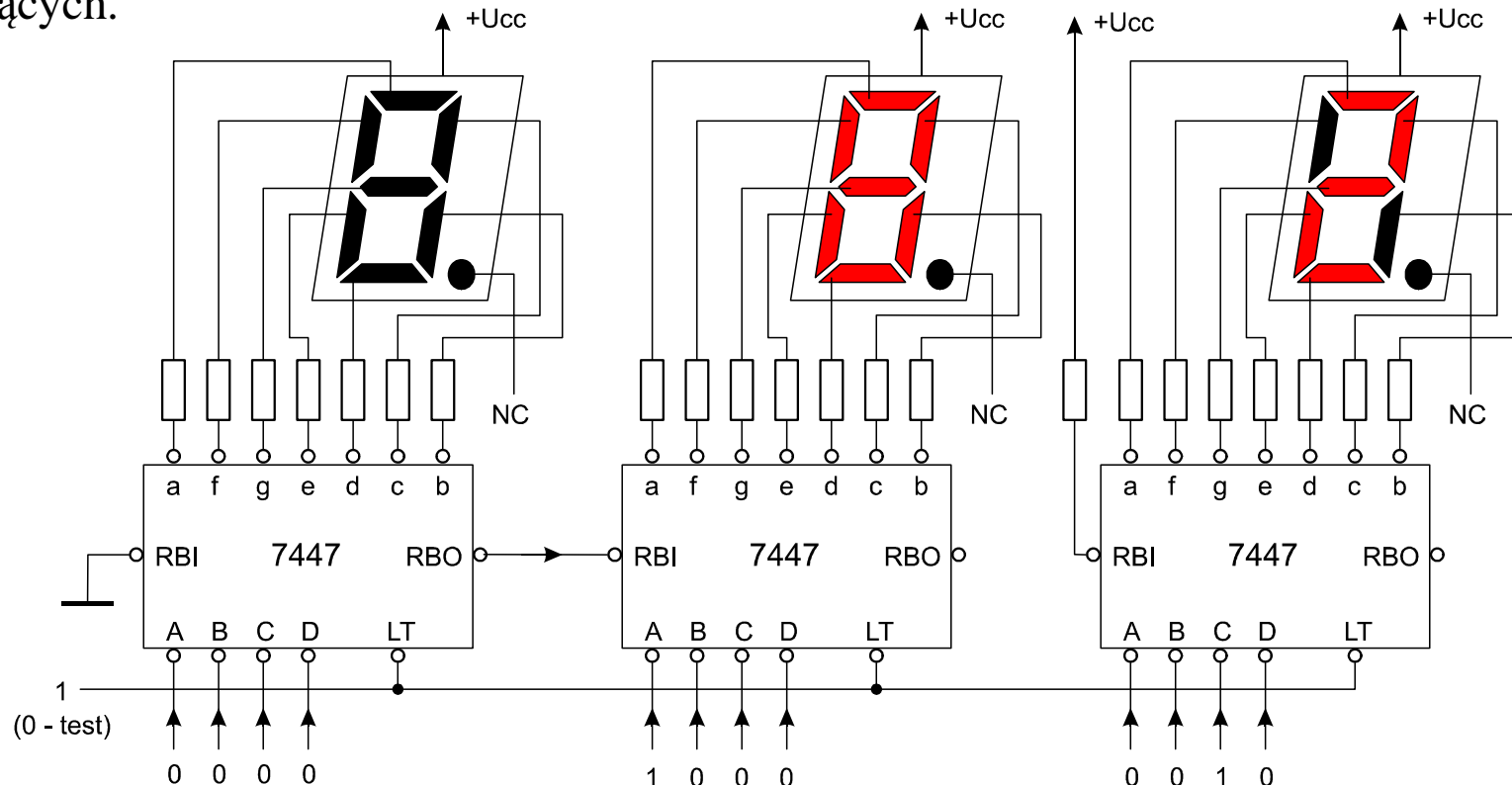
**Rys. 4.11.** Symbol dekodera.

**Uwaga:** demultiplekser z wejściem ustawionym trwale na stan 1 pracuje jak dekodery.

## 4.1.5. Transkodery

Transkoderem (translatorem) nazywamy układ przetwarzający informację wyrażoną w dowolnym kodzie na równoważną informację w innym kodzie. Kod wejściowy i wyjściowy transkodera są różne od kodu „1 z N”.

Układ scalony 7447 jest transkoderem jednej cyfry dziesiętnej w kodzie BCD na kod numerycznego wyświetlacza 7-io segmentowego. Dodatkowe wyjście RBO (*ripple blanking output*) oraz wejście RBI (*ripple blanking input*) służą do wygaszania zer nieznaczących.

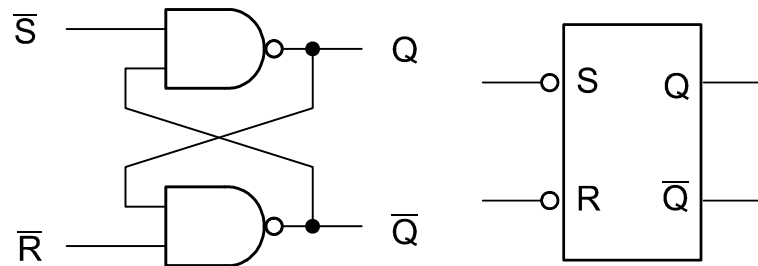


Rys. 4.12. Schemat układu sterowania wyświetlaczem numerycznym z wygaszaniem zer T4-11

## 4.2. Przerzutniki

### 4.2.1. Przerzutnik prosty typu RS

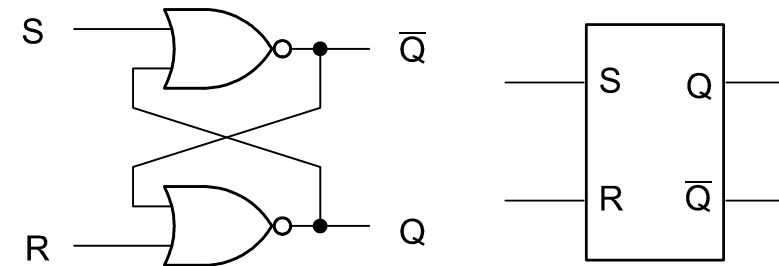
Przerzutnik prosty typu RS jest elementarnym asynchronicznym układem pamięci. Podstawowa wersja tego przerzutnika zawiera dwa wejścia:  $R$  - zerowanie (*ang. reset*) oraz  $S$  - ustawianie (*ang. set*).



**Rys. 4.13.** Realizacja przerzutnika RS ustawianego poziomem 0 i jego symbol.

$\bar{S}$	$\bar{R}$	$Q$	$\bar{Q}$	
0	0	1*	1*	zabronione
0	1	1	0	
1	0	0	1	
1	1	$Q_0$	$\bar{Q}_0$	bez zmian

**Tabela 4.7.** Przejścia przerzutnika z rys. 4.13.



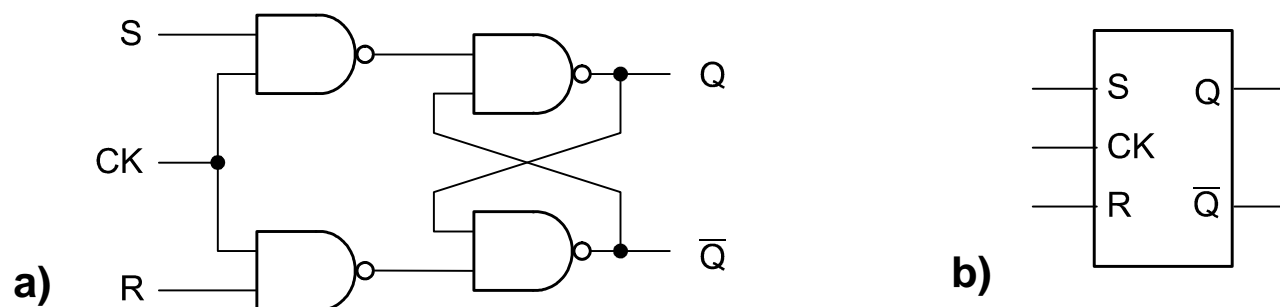
**Rys. 4.14.** Realizacja przerzutnika RS ustawianego poziomem 1 i jego symbol.

$S$	$R$	$Q$	$\bar{Q}$	
0	0	$Q_0$	$\bar{Q}_0$	bez zmian
0	1	0	1	
1	0	1	0	
1	1	0*	0*	zabronione

**Tabela 4.8.** Przejścia przerzutnika z rys. 4.14.

## 4.2.2. Przerzutnik synchroniczny RS

W synchronicznym przerzutniku RS wejścia  $R$  i  $S$  (*ang. reset i set*) pełnią rolę wejść przygotowujących, tzn. stany wyjść są ustawiane zgodnie ze stanem tych wejść wyłącznie podczas odpowiedniego poziomu na wejściu taktującym  $CK$  (*ang. clock*).



**Rys. 4.15.** Synchroniczny przerzutnik RS zapisywany podczas poziomu 1 na wejściu taktującym  $CK$ . (a) realizacja przerzutnika z bramek NAND; (b) symbol graficzny.

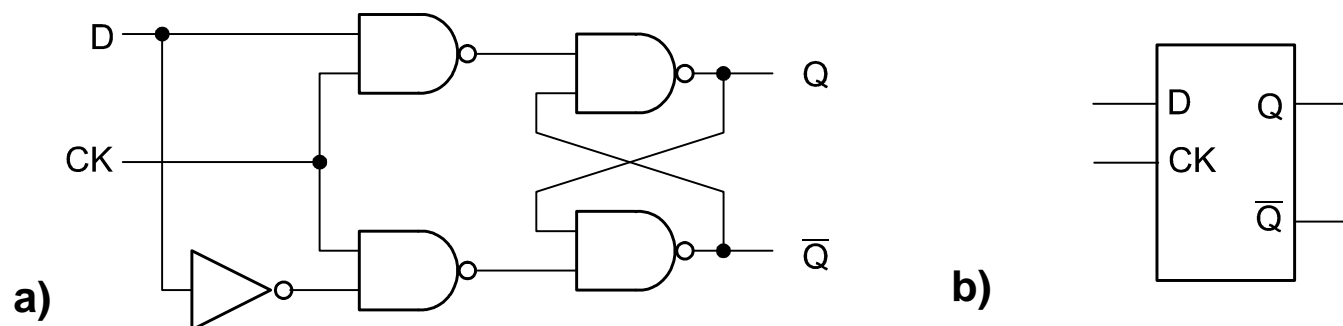
$CK$	$S$	$R$	$Q$	$\bar{Q}$	
0	–	–	$Q_0$	$\bar{Q}_0$	bez zmian
1	0	0	$Q_0$	$\bar{Q}_0$	bez zmian
1	0	1	0	1	
1	1	0	1	0	
1	1	1	1*	1*	zabronione

Na wszystkich wejściach stan 1 jest stanem ustawiającym przerzutnik z rys. 4.15.

**Tabela 4.9.** Przejścia przerzutnika z rys. 4.15. Symbol „–” oznacza stan dowolny. T4-13

## 4.2.3. Przerzutnik D wyzwalany poziomem

Przerzutnik typu D wyzwalany poziomem jest modyfikacją synchronicznego przerzutnika RS, w którym dzięki zastosowaniu dodatkowego inwertera między wejściami  $R$  i  $S$  wyeliminowano niedozwoloną kombinację stanów wejść.



**Rys. 4.16.** Przerzutnik D wyzwalany poziomem 1 na wejściu taktującym  $CK$ .  
(a) realizacja przerzutnika z bramek NAND; (b) symbol graficzny.

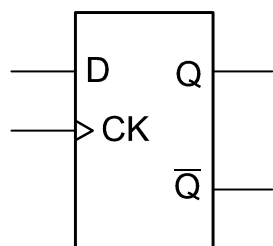
$D$	$CK$	$Q$	$\bar{Q}$	
0	0	$Q_0$	$\bar{Q}_0$	bez zmian
1	0	$Q_0$	$\bar{Q}_0$	bez zmian
1	1	1	0	
0	1	0	1	

Przerzutniki D wyzwalane poziomem są często nazywane „zatrzaskami” (*ang. latch*). Nazwa ta nie dotyczy przerzutników D wyzwalanych zboczem.

**Tabela 4.10.** Przejścia przerzutnika D z rys. 4.16.

## 4.2.4. Przerzutniki D i T wyzwalane zboczem

Przerzutnik typu D wyzwalany zboczem przepisuje stan wejścia informacyjnego  $D$  na wyjście  $Q$  tylko podczas odpowiedniego zbocza impulsu zegarowego  $CK$  (typowo zbocze rosnące  $0 \rightarrow 1$ ).

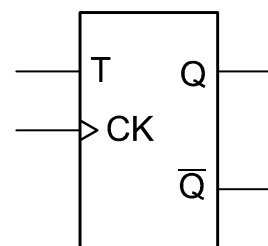


**Rys. 4.17.** Symbol graficzny przerzutnika D wyzwalanego zboczem  $0 \rightarrow 1$ .

$D$	$CK$	$Q$	$\bar{Q}$	
–	0	$Q_0$	$\bar{Q}_0$	bez zmian
0	↑	0	1	
1	↑	1	0	

**Tabela 4.11.** Przejścia przerzutnika D z rys. 4.17.

Przerzutnik typu T wyzwalany zboczem zmienia stan wyjścia  $Q$  na przeciwny podczas odpowiedniego zbocza impulsu zegarowego  $CK$  gdy wejście informacyjne  $T=1$  albo nie wykonuje żadnej operacji gdy  $T=0$ .



**Rys. 4.18.** Symbol graficzny przerzutnika T wyzwalanego zboczem  $0 \rightarrow 1$ .

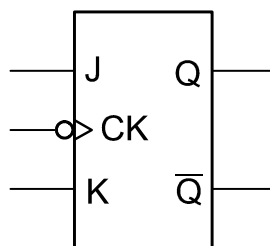
$T$	$CK$	$Q$	$\bar{Q}$	
–	0	$Q_0$	$\bar{Q}_0$	bez zmian
0	↑	$Q_0$	$\bar{Q}_0$	bez zmian
1	↑	$\bar{Q}_0$	$Q_0$	odwrotnie

**Tabela 4.12.** Przejścia przerzutnika T z rys. 4.18.

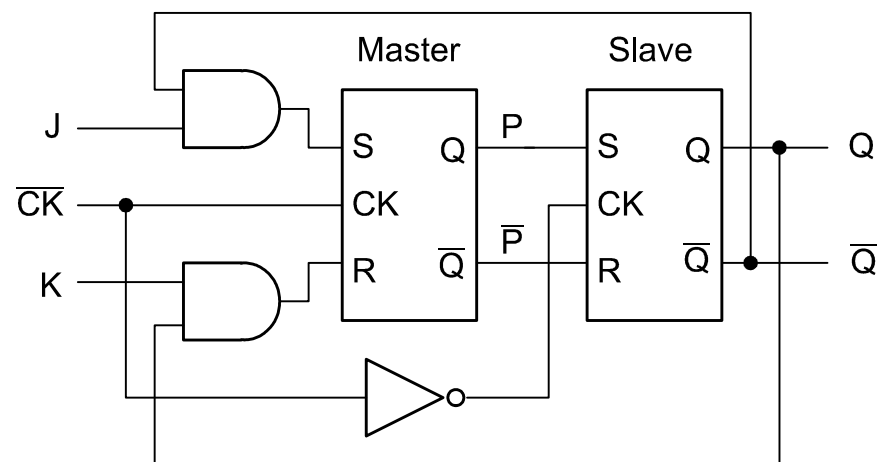
Uwaga: przerzutniki D i T wyzwalane zboczem mogą mieć także wejścia bezpośrednie (asynchroniczne)  $R$  i  $S$  działające jak wejścia przerzutnika prostego RS.

## 4.2.5. Przerzutnik dwutaktowy JK (Master-Slave)

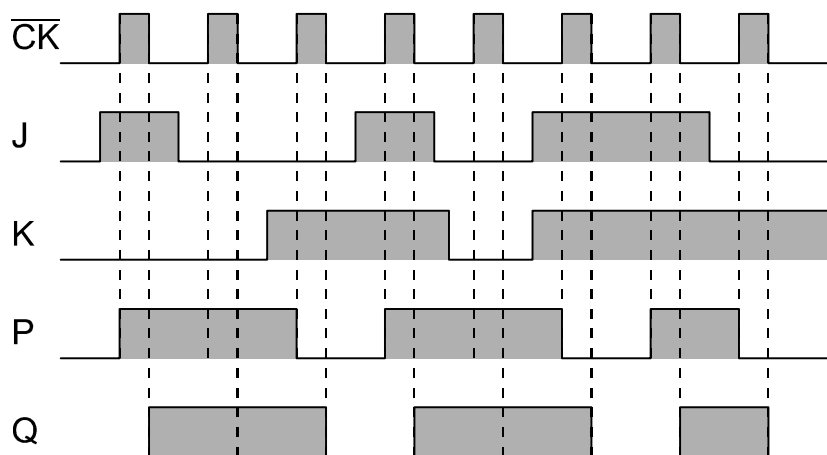
Przerzutnik dwutaktowy JK jest złożony z dwóch przerzutników synchronicznych RS, których zapis odbywa się kolejno podczas innych poziomów na wejściu zegarowym CK.



**Rys. 4.19.** Symbol graficzny przerzutnika JK wyzwalanego impulsem dodatnim  $0 \rightarrow 1 \rightarrow 0$ .



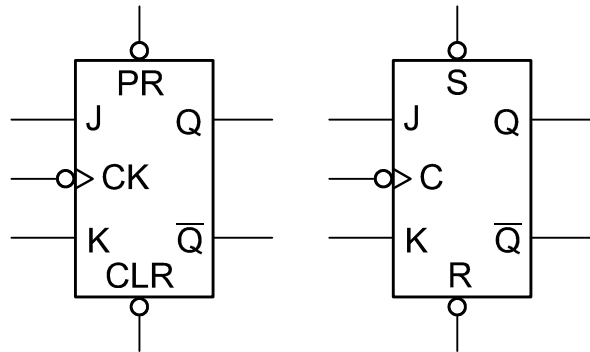
**Rys. 4.20.** Zasada budowy dwutaktowego przerzutnika JK.



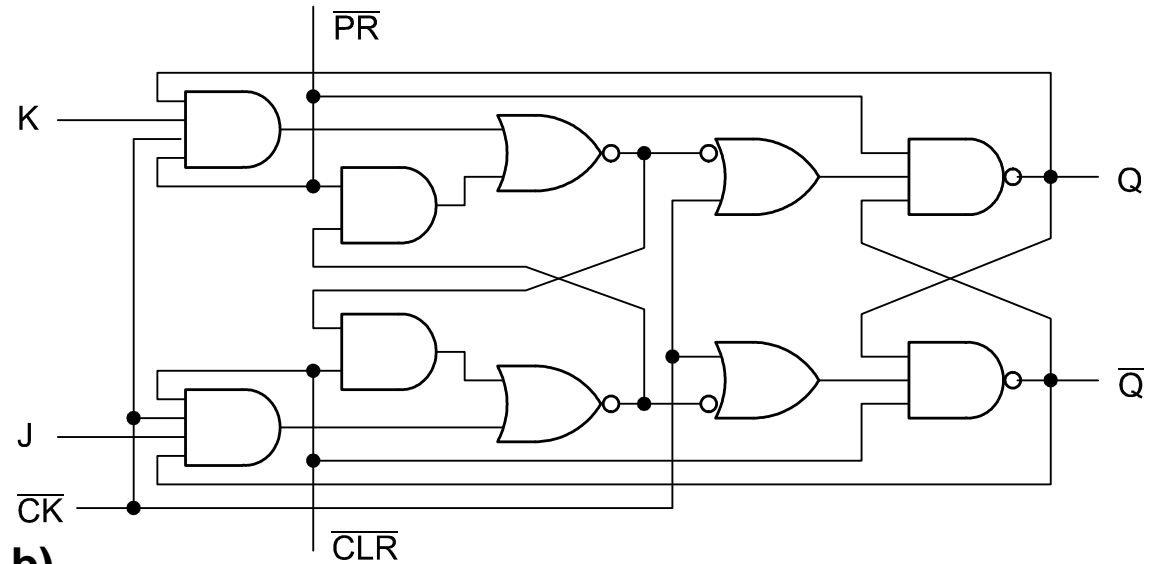
**Rys. 4.21.** Wykresy czasowe dla dwutaktowego przerzutnika JK. Zmiany stanów wejść J i K nie oddziałują natychmiast na wyjścia. Przerzutnik „Slave” powtarza stany przerzutnika „Master” z opóźnieniem o czas trwania stanu 1 na wejściu zegarowym.



Przykłady zróżnicowanej symboliki z literatury



a)



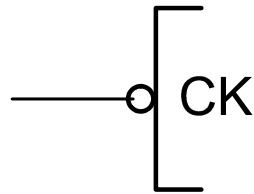
b)

**Rys. 4.22.** Przerzutnik JK-MS wyzwalany dodatnim impulsem  $0 \rightarrow 1 \rightarrow 0$  rozszerzony o wejścia bezpośrednio  $CLR$  (clear) i  $PR$  (preset) dostępny w układach scalonych 7476.

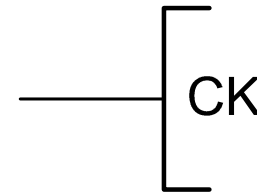
$\overline{CLR}$	$\overline{PR}$	$J$	$K$	$\overline{CK}$	$Q$	$\overline{Q}$	
0	0	–	–	–	1*	1*	zabronione
0	1	–	–	–	0	1	
1	0	–	–	–	1	0	
1	1	0	0	$\uparrow \downarrow$	$Q_0$	$\overline{Q}_0$	bez zmian
1	1	1	0	$\uparrow \downarrow$	1	0	
1	1	0	1	$\uparrow \downarrow$	0	1	
1	1	1	1	$\uparrow \downarrow$	$\overline{Q}_0$	$Q_0$	odwrotnie

**Tabela 4.13.** Przejścia przerzutnika JK MS z układu scalonego 7476.

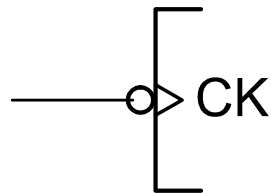
Zależnie od sposobu wyzwalania przerzutnika stosowane są następujące symbole wejścia zegarowego:



wyzwalanie poziomem 0

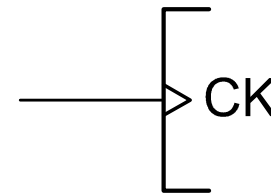


wyzwalanie poziomem 1



w przerzutnikach wyzwalanych zboczem aktywne jest zbocze opadające  $1 \rightarrow 0$ ;

w przerzutnikach dwutaktowych zapis do „Master” na zboczu rosnącym  $0 \rightarrow 1$ , przepisanie stanu do „Slave” na zboczu opadającym  $1 \rightarrow 0$ . Pełny cykl zapisu obejmuje impuls dodatni  $0 \rightarrow 1 \rightarrow 0$ .



w przerzutnikach wyzwalanych zboczem aktywne jest zbocze rosnące  $0 \rightarrow 1$ ;

w przerzutnikach dwutaktowych zapis do „Master” na zboczu opadającym  $1 \rightarrow 0$ , przepisanie stanu do „Slave” na zboczu rosnącym  $0 \rightarrow 1$ . Pełny cykl zapisu obejmuje impuls ujemny  $1 \rightarrow 0 \rightarrow 1$ .

## Przegląd układów scalonych w rodziny 74xx zawierających przerzutniki

### Przerzutniki proste RS z bramek NAND

- 74279 – 4 przerzutniki RS z wejściami zanegowanymi, wyprowadzone tylko wyjścia  $Q$ .

### Przerzutniki D wyzwalane poziomem 1 „zatrzaski”

- 7475 – 4 zatrzaski z wyjściami  $Q$  oraz  $\text{NOT}(Q)$ ,
- 7477 – 4 zatrzaski z wyjściami tylko  $Q$ ,

### Przerzutniki D wyzwalane zboczem rosnącym $0 \rightarrow 1$

- 7473 – 2 niezależne przerzutniki z dodatkowymi wejściami CLR,
- 7474 – 2 niezależne przerzutniki z dodatkowymi wejściami PR i CLR,
- 74174 – 6 przerzutników ze wspólnymi wejściami  $CK$  i  $CLR$ . Wyjścia tylko  $Q$ .

### Przerzutniki JK Master-Slave wyzwalane impulsem dodatnim $0 \rightarrow 1 \rightarrow 0$

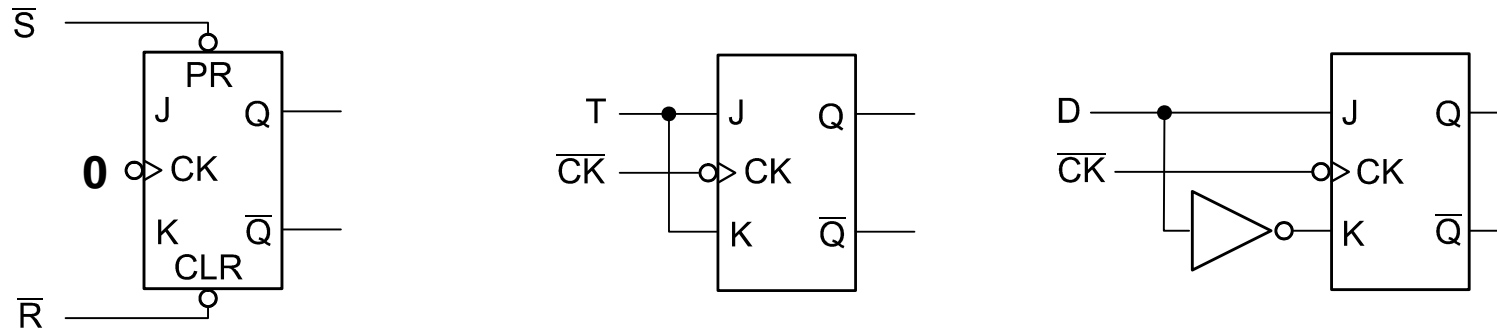
- 7476 – 2 niezależne przerzutniki z dodatkowymi wejściami PR i CLR,
- 7478 – 2 przerzutniki z niezależnymi wejściami PR i wspólnymi CLR.

### Przerzutniki JK wyzwalane ujemnym zboczem $1 \rightarrow 0$

- 74103 – 2 niezależne przerzutniki z dodatkowymi wejściami CLR,
- 74106 – 2 niezależne przerzutniki z dodatkowymi wejściami PR i CLR.

Przerzutniki innego typu mogą wchodzić w skład większych bloków funkcjonalnych ale nie są dostępne jako przerzutniki o niezależnych wyprowadzeniach, np. przerzutniki T, dwutaktowe (Master-Slave) przerzutniki RS oraz D.

## 4.2.6. Konwersje przerzutników

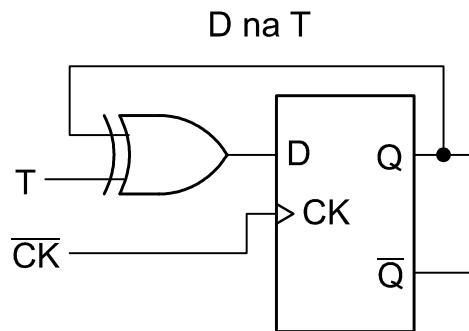


a) przerzutnik prosty RS,

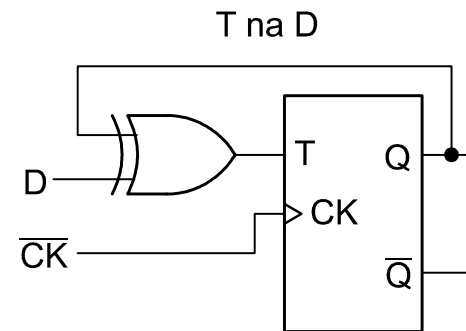
b) przerzutnik T,

c) przerzutnik D.

Rys. 4.23. Konwersje przerzutnika JK z dodatkowymi wejściami bezpośrednimi.



Rys. 4.24. Konwersja przerzutnika typu D w przerzutnik T.



Rys. 4.25. Konwersja przerzutnika typu T w przerzutnik D.

## 4.3. Liczniki

---

Licznik to układ sekwencyjny, którego podstawową funkcją jest zwiększanie (inkrementacja) lub zmniejszanie (dekrementacja) liczby całkowitej reprezentowanej przez jego wewnętrzny stan. W praktyce najczęściej stosowane są liczniki pracujące w kodach NKB (naturalny kod binarny) oraz BCD (*ang. binary-coded decimal*, czyli zapis dziesiętny kodowany dwójkowo). W niektórych licznikach dostępne są także dodatkowe funkcje: zerowanie, ustawianie w stan 1 oraz przepisanie stanu z wejść równoległych. Dodatkowe operacje mogą być realizowane synchronicznie ze zmianami stanu na wejściu zegarowym albo asynchronicznie.

W zależności od struktury liczniki można podzielić na dwie główne grupy:

- liczniki szeregowo (asynchroniczne),
- liczniki równoległe (synchroniczne).

Spotykane są także liczniki o strukturze mieszanej.

### 4.3.1. Liczniki szeregowo (asynchroniczne)

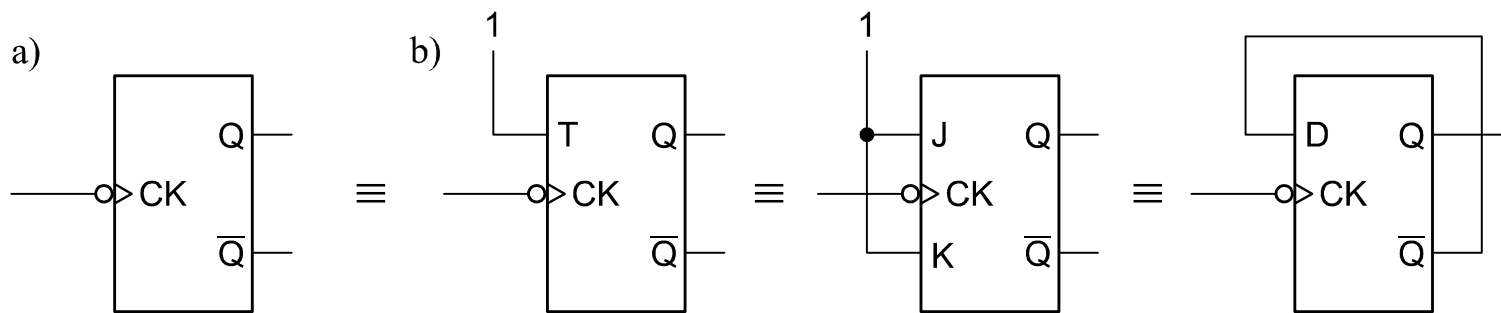
---

W licznikach szeregowych zliczane impulsy doprowadzane są do wejścia zegarowego tylko pierwszego przerzutnika, który odpowiada za najmniej znaczący bit  $Q_0$ . Dalsze przerzutniki mogą przełączyć się dopiero, gdy wszystkie pośrednie przerzutniki zmienią kolejno swój stan. Taka struktura licznika umożliwia maksymalne uproszczenie układu, jednakże prowadzi do względnie długich czasów trwania błędnych stanów przejściowych na wyjściach licznika (stąd druga nazwa – liczniki asynchroniczne).

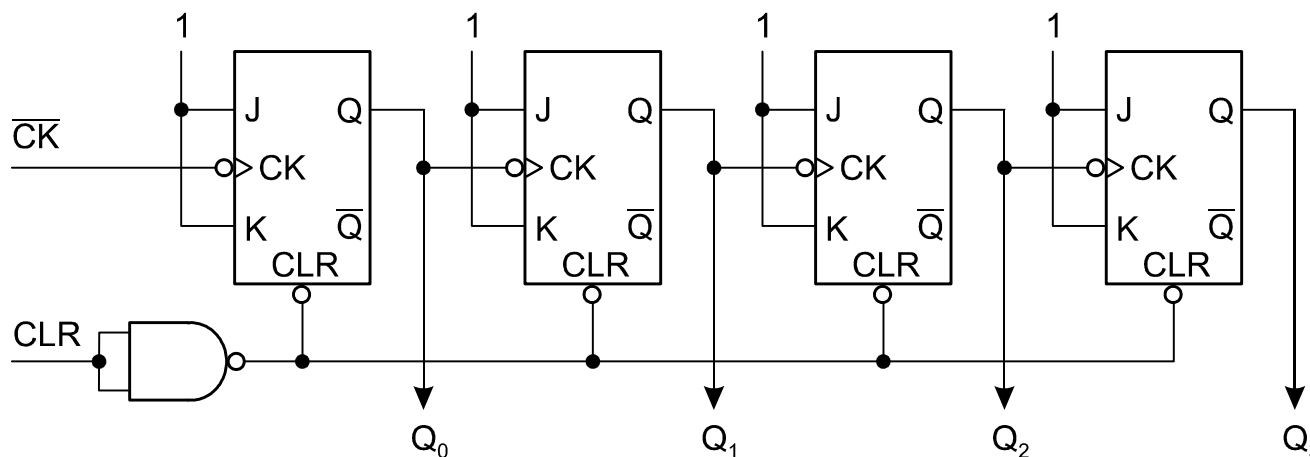
Podstawową komórką licznika szeregowego jest tzw. **dwójka licząca**, w której stan wyjścia Q zmienia się na przeciwny po określonym zboczu na wejściu zegarowym CK (ang. *clock*). Dla wejścia zanegowanego, jak na rys. 26, jest to zbocze opadające 0 → 1.

Najprostszy licznik modulo S, gdzie S jest liczną typu  $2^N$ , buduje się przez szeregowe połączenie dwójek liczących. Zmiany stanu licznika modulo 16 przedstawionego na rys. 4.27 po każdym impulsie zegarowym opisuje wzór:

$$Q^{t+1} := (Q^t + 1)_{\text{mod } 16} \quad (4.5)$$



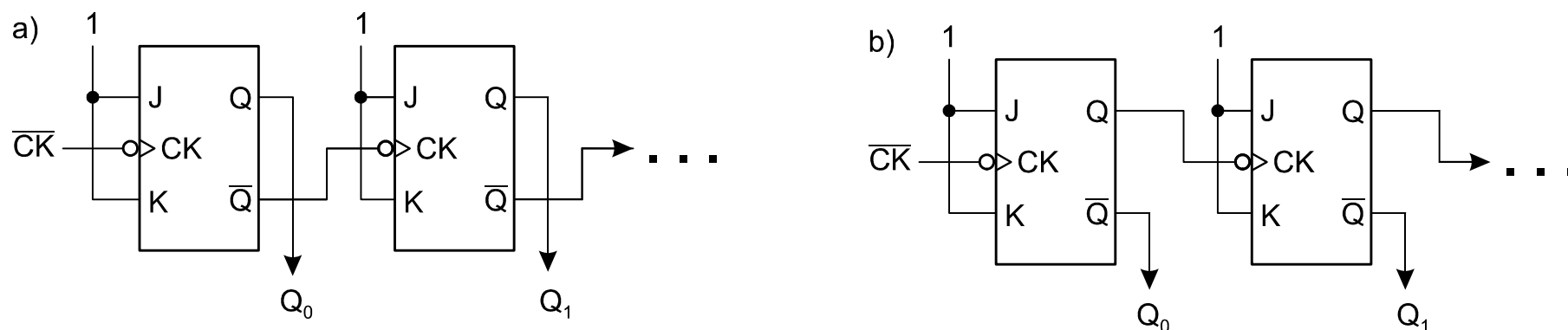
**Rys. 4.26.** Dwójka licząca: (a) symbol; (b) realizacja przy użyciu przerzutników T, JK oraz D.



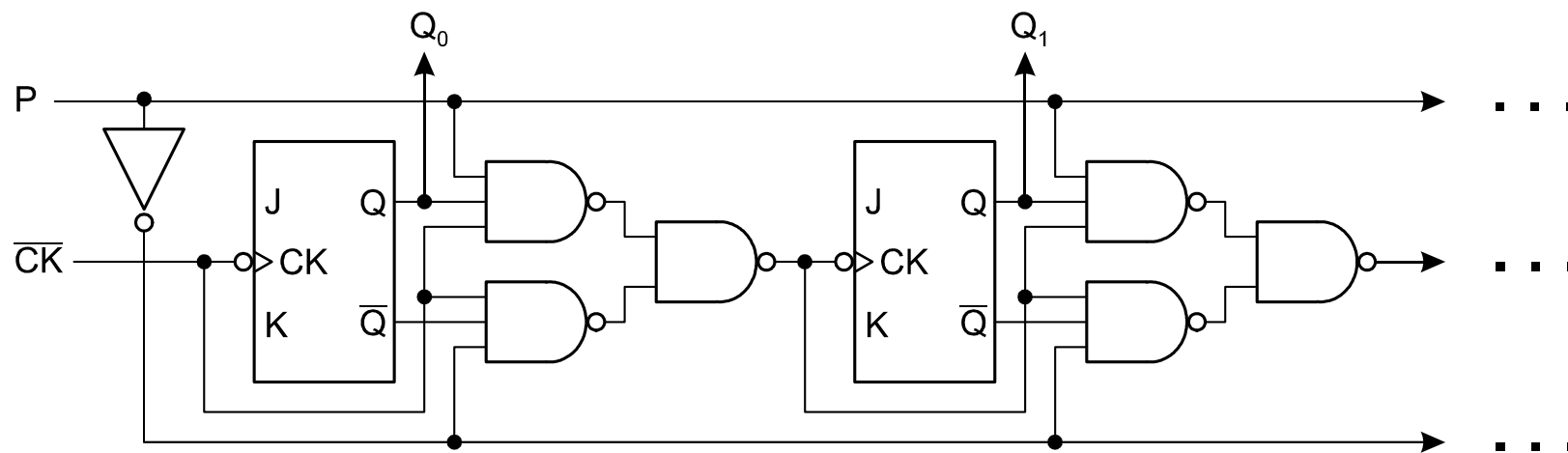
**Rys. 4.27.** Schemat szeregowego licznika z zerowaniem asynchronicznym (układ 7493).

Połączenie wejść NOT( $CK$ ) poszczególnych przerzutników z wyjściami zanegowanymi NOT( $Q$ ) poprzedzających dwójek prowadzi do odwrócenia kierunku zliczania (rys. 4.28.a). Alternatywnie zliczanie do tyłu uzyskuje się przez zanegowanie stanu wszystkich bitów  $Q_i$  na wyjściach licznika zliczającego do przodu (rys. 4.28.b).

Nazwa „licznik rewersyjny” w literaturze jest używana dla liczników o dwóch kierunkach zliczania. Układ taki otrzymujemy dodając w każdym stopniu licznika multiplexer wybierający wyjście  $Q$  albo NOT( $Q$ ) z poprzedniego stopnia (rys. 4.29).

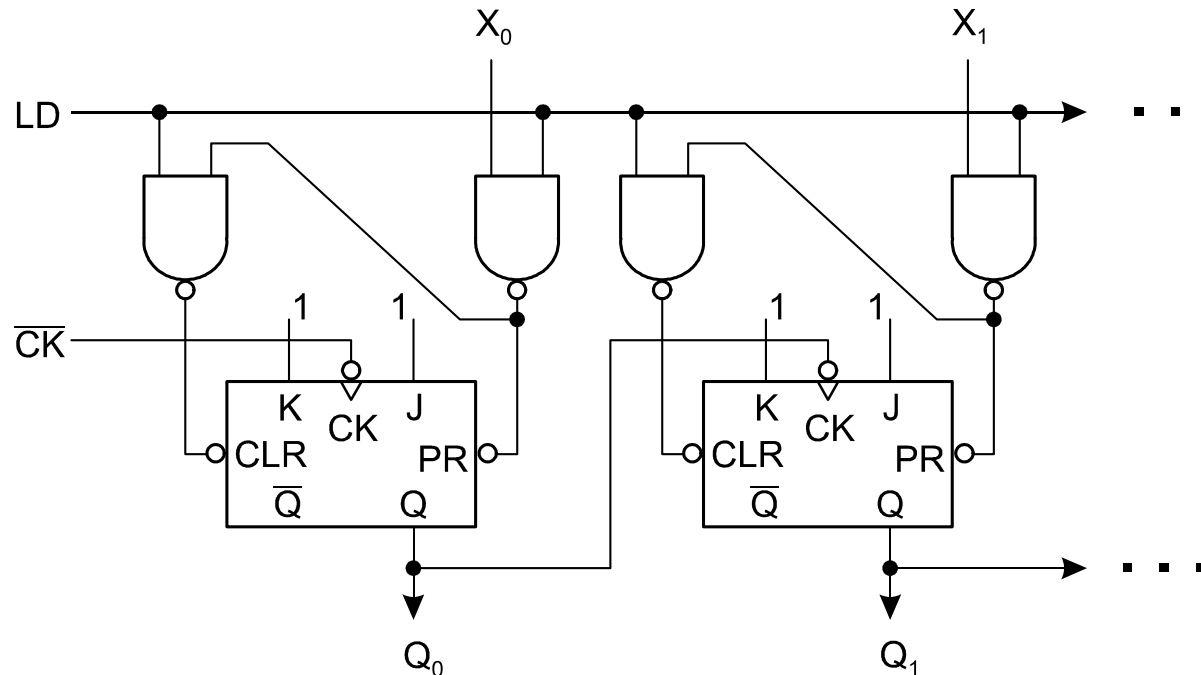


**Rys. 4.28.** Dwa warianty szeregowego licznika dwójkowego liczącego wstecz.



**Rys. 4.29.** Szeregowy licznik rewersyjny. Zliczanie do przodu dla  $P=1$ , do tyłu dla  $P=0$ .

Przepisywanie stanu początkowego licznika o strukturze szeregowej z wejść równoległych typowo jest realizowane w sposób asynchroniczny. Do operacji asynchronicznych wykorzystuje się wejścia bezpośrednio *CLR* (ang. *clear*) i *PR* (ang. *preset*) przerzutników.



**Rys. 4.30.** Asynchroniczne ustawianie stanu początkowego licznika szeregowego na przykładzie licznika liczącego do przodu, który odpowiada w przybliżeniu układowi scalonemu 74197. Wejście *LD* (ang. *load*) wybiera realizowaną mikrooperację:  
 $LD = 0 \Rightarrow \mathbf{Q} := ((\mathbf{Q} + 1) \bmod 16)$  po każdym zboczu  $0 \rightarrow 1$  na wejściu  $\text{NOT}(CK)$ ,  
 $LD = 1 \Rightarrow \mathbf{Q} := \mathbf{X}$  niezależnie od stanu wejścia zegarowego.





## Przykład licznika synchronicznego

Zaprojektować układ licznika synchronicznego modulo 5 liczącego do przodu w naturalnym kodzie binarnym. Zaplanować korektę błędnych stanów licznika.

Wariant 1 – licznik z synchroniczną korektą błędnych stanów



stan wyjść			przerzutniki JK			przerzutniki T			przerzutniki D		
$Q_2$	$Q_1$	$Q_0$	$J_2K_2$	$J_1K_1$	$J_0K_0$	$T_2$	$T_1$	$T_0$	$D_2$	$D_1$	$D_0$
0	0	0	0 –	0 –	1 –	0	0	1	0	0	1
0	0	1	0 –	1 –	– 1	0	1	1	0	1	0
0	1	0	0 –	– 0	1 –	0	0	1	0	1	1
0	1	1	1 –	– 1	– 1	1	1	1	1	0	0
1	0	0	– 1	0 –	0 –	1	0	0	0	0	0
1	0	1	– 1	0 –	– 1	1	0	1	0	0	0
1	1	0	– 1	– 1	0 –	1	1	0	0	0	0
1	1	1	– 1	– 1	– 1	1	1	1	0	0	0

**Tabela. 4.14.** Tablica kolejnych stanów licznika modulo 5. (–) oznacza wartość dowolną.

Stany wejść przygotowujących przerzutników dobieramy porównując bieżący stan licznika  $Q_2Q_1Q_0$  z oczekiwanym stanem następnym. Do realizacji wybierzemy układ wykorzystujący przerzutniki typu JK, dla których blok wejściowy będzie najprostsz.

Po zapisaniu tablicy kolejnych stanów licznika projekt układu sprowadza się do syntezy wejściowego bloku kombinacyjnego.

$$J_2 \quad Q_1 Q_0$$

$Q_2$	00	01	11	10
0	0	0	1	0
1	-	-	-	-

$$J_2 = Q_1 Q_0$$

$$J_1 \quad Q_1 Q_0$$

$Q_2$	00	01	11	10
0	0	1	-	-
1	0	0	-	-

$$J_1 = \overline{Q_2} Q_0$$

$$J_0 \quad Q_1 Q_0$$

$Q_2$	00	01	11	10
0	1	-	-	1
1	0	-	-	0

$$J_0 = \overline{Q_2} \quad (4.6)$$

$$K_2 \quad Q_1 Q_0$$

$Q_2$	00	01	11	10
0	-	-	-	-
1	1	1	1	1

$$K_2 = 1$$

$$K_1 \quad Q_1 Q_0$$

$Q_2$	00	01	11	10
0	-	-	1	0
1	-	-	1	1

$$K_1 = Q_0$$

$$K_0 \quad Q_1 Q_0$$

$Q_2$	00	01	11	10
0	-	1	1	-
1	-	1	1	-

$$K_0 = 1 \quad (4.7)$$

**Rys. 4.32.** Tablica stanów dla licznika modulo 5 z **synchroniczną** korektą błędnych stanów zbudowanego z przerzutników JK-MS po przepisaniu jej w formie tablic Karnaugh.

## Wariant 2 – licznik modulo 5 z asynchroniczną korektą błędnych stanów

Do asynchronicznej korekty błędnych stanów wykorzystuje się wejścia asynchroniczne przerzutników *CLR* (ang. *clear*) oraz *PR* (ang. *preset*). Asynchroniczna korekta zazwyczaj nie wpływa na projekt bloku kombinacyjnego sterującego wejściami przygotowującymi przerzutników (tzn. *J* i *K*, *T* albo *D*).

stan wyjść			$\overline{CLR}$	przerzutniki JK			przerzutniki T			przerzutniki D		
$Q_2$	$Q_1$	$Q_0$		$J_2K_2$	$J_1K_1$	$J_0K_0$	$T_2$	$T_1$	$T_0$	$D_2$	$D_1$	$D_0$
0	0	0	1	0 –	0 –	1 –	0	0	1	0	0	1
0	0	1	1	0 –	1 –	– 1	0	1	1	0	1	0
0	1	0	1	0 –	– 0	1 –	0	0	1	0	1	1
0	1	1	1	1 –	– 1	– 1	1	1	1	1	0	0
1	0	0	1	– 1	0 –	0 –	1	0	0	0	0	0
1	0	1	0	– –	– –	– –	–	–	–	–	–	–
1	1	0	0	– –	– –	– –	–	–	–	–	–	–
1	1	1	0	– –	– –	– –	–	–	–	–	–	–

**Tabela. 4.15.** Tablica kolejnych stanów licznika modulo 5 z asynchroniczną korektą błędnych stanów. Not(*CLR*) oznacza stan wejść zerowania wszystkich przerzutników.

Do realizacji wybierzemy układ wykorzystujący przerzutniki typu JK, dla których blok wejściowy będzie najprostszy.

$J_2$	$Q_1 Q_0$				
		00	01	11	10
$Q_2$					
0		0	0	1	0
1		-	-	-	-

$$J_2 = Q_1 Q_0$$

$J_1$	$Q_1 Q_0$				
		00	01	11	10
$Q_2$					
0		0	1	-	-
1		0	-	-	-

$$J_1 = Q_0 \text{ zmiana względem wer. 1}$$

$J_0$	$Q_1 Q_0$				
		00	01	11	10
$Q_2$					
0		1	-	-	1
1		0	-	-	-

$$J_0 = \overline{Q_2} \quad (4.8)$$

$K_2$	$Q_1 Q_0$				
		00	01	11	10
$Q_2$					
0		-	-	-	-
1		1	-	-	-

$$K_2 = 1$$

$K_1$	$Q_1 Q_0$				
		00	01	11	10
$Q_2$					
0		-	-	1	0
1		-	-	-	-

$$K_1 = Q_0$$

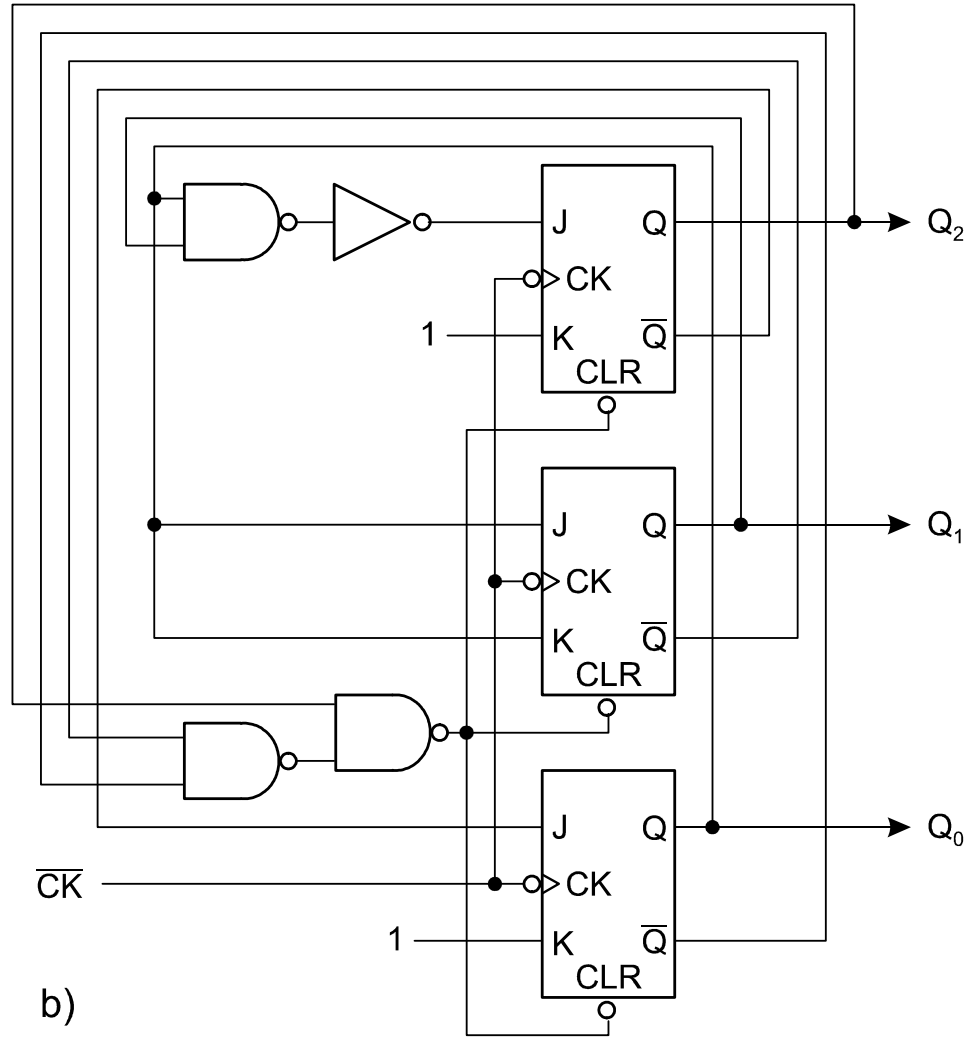
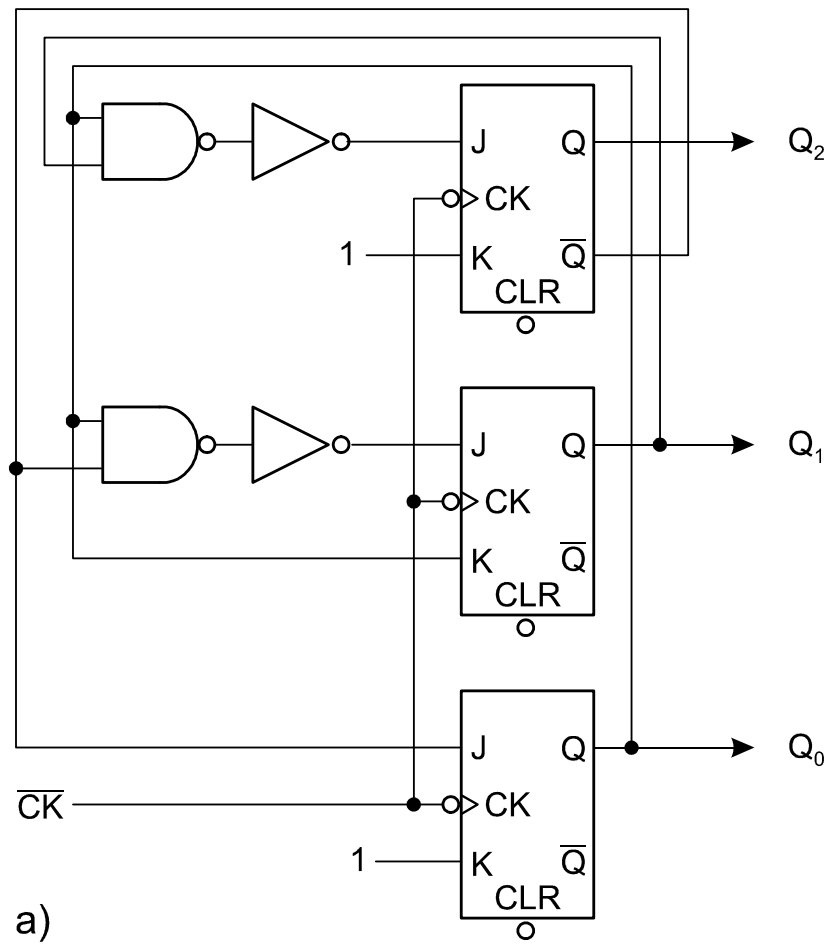
$K_0$	$Q_1 Q_0$				
		00	01	11	10
$Q_2$					
0		-	1	1	-
1		-	-	-	-

$$K_0 = 1 \quad (4.9)$$

$\overline{CLR}$	$Q_1 Q_0$				
		00	01	11	10
$Q_2$					
0		1	1	1	1
1		1	0	0	0

$$\overline{CLR} = \overline{Q_2} + \overline{Q_1} \overline{Q_0} = \overline{Q_2} \overline{\overline{Q_1} \overline{\overline{Q_0}}} \quad (4.10)$$

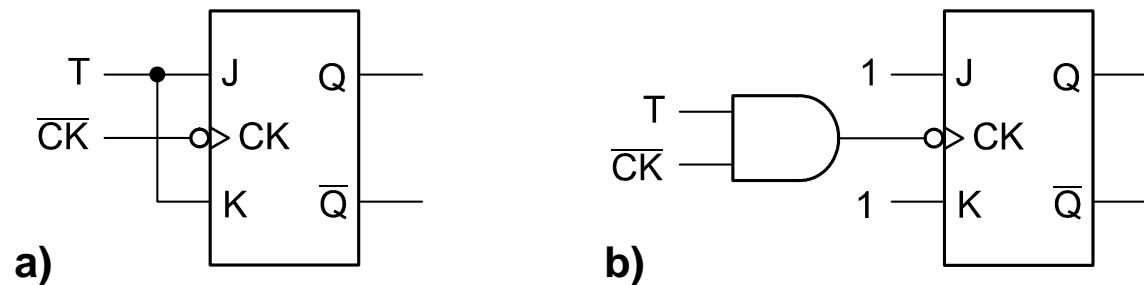
**Rys. 4.33.** Tablica stanów dla licznika modulo 5 z **asynchroniczną** korektą błędnych stanów zbudowanego z przerzutników JK-MS po przepisaniu jej w formie tablic Karnaugh.



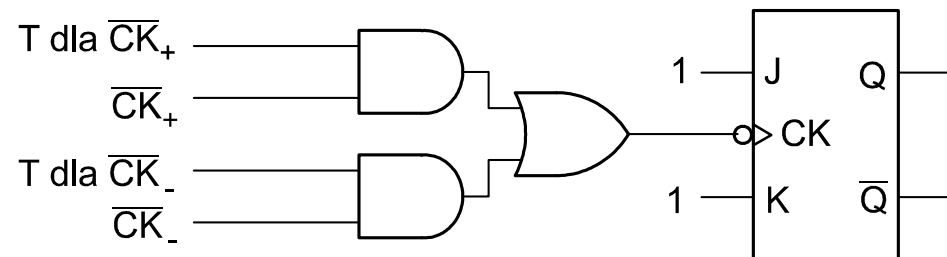
**Rys. 4.34.** Schemat układu synchronicznego licznika modulo 5 z korektą błędnych stanów, (a) korekta synchroniczna, (b) korekta asynchroniczna.

Wejście zegarowe w rewersyjnych (dwukierunkowych) licznikach synchronicznych może być zaprojektowane na dwa sposoby:

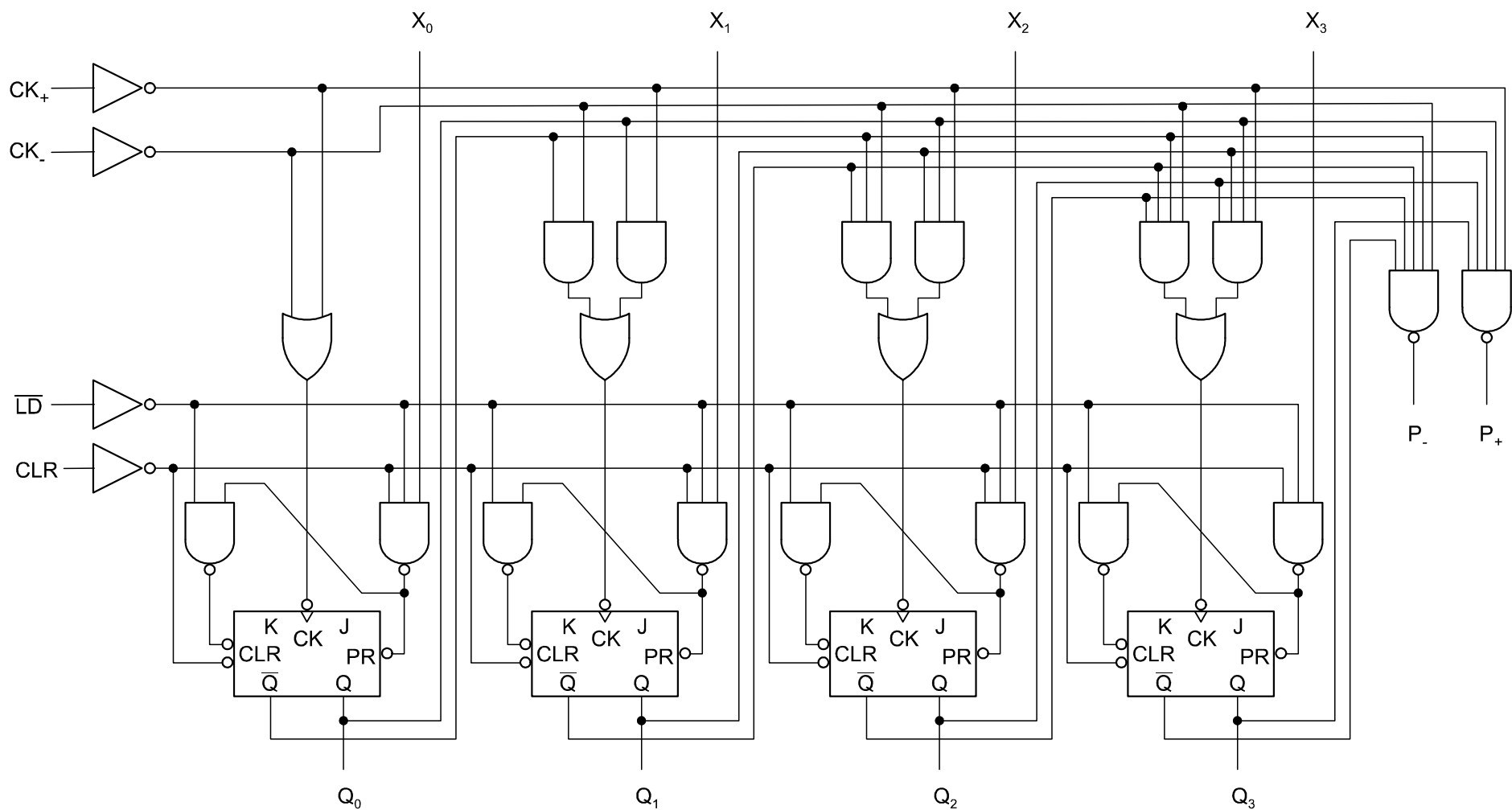
- jedno wejście zegarowe i pomocnicze wejście (prowadzące do wejściowego bloku kombinacyjnego), które określa kierunek zliczania,
- niezależne wejścia dla impulsów zliczanych do przodu oraz do tyłu. Wykorzystanie przerzutników T o rozdzielonych wejściach zegarowych (rys. 4.36) umożliwia zaprojektowanie dwóch niezależnych wejściowych bloków kombinacyjnych związanych z poszczególnymi kierunkami zliczania. Takie rozwiązanie zastosowano np. w rewersyjnych licznikach synchronicznych 74192 (modulo 10) oraz 74193 (modulo 16).



**Rys. 4.35.** Dwie możliwości konwersji przerzutnika JK w przerzutnik T: **(a)** dokładny odpowiednik przerzutnika T; **(b)** przerzutnik pracujący jako dwójka licząca przekształcony w przerzutnik T - równoważność pod warunkiem zmiany stanu  $T$  tylko podczas  $\text{NOT}(CK)=0$ .



**Rys. 4.36.** Przykład realizacji przerzutnika typu T o rozdzielonych wejściach dla dwóch sygnałów zegarowych.



**Rys. 4.37.** Schemat synchronicznego licznika rewersyjnego modulo 16 w układzie scalonym 74193.

**Tabela. 4.16.** Tablica przejść licznika z rys. 4.37.

$CLR$	$\overline{LD}$	$CK_+$	$CK_-$	mikrooperacja
1	—	—	—	$\mathbf{Q} := 0$
0	0	—	—	$\mathbf{Q} := \mathbf{X}$
0	1	↑	1	$\mathbf{Q} := \mathbf{Q}+1$
0	1	1	↑	$\mathbf{Q} := \mathbf{Q}-1$



## **4.4. Rejestry**

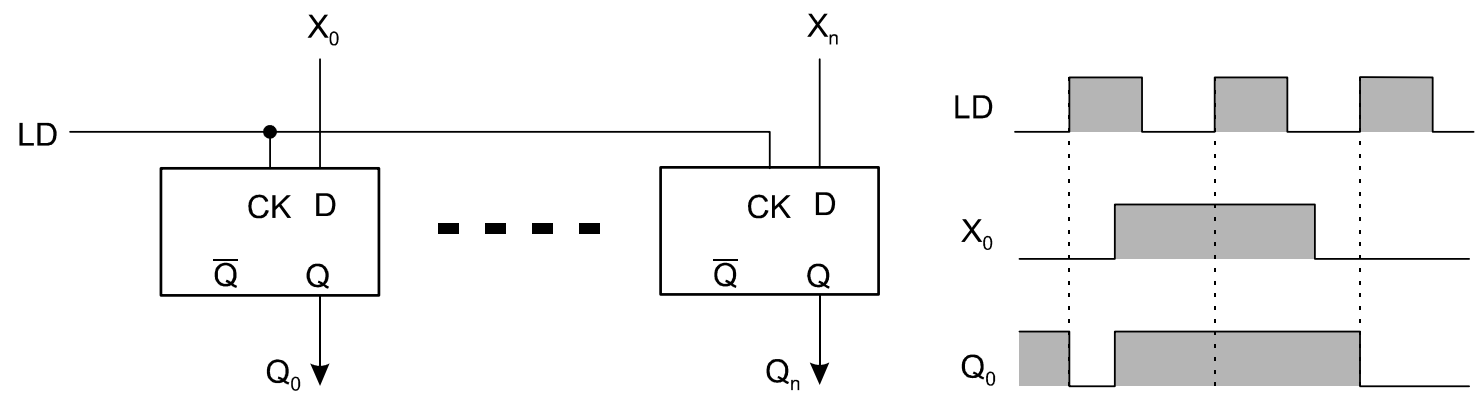
---

Rejestry są synchronicznymi układami sekwencyjnymi, w których podstawowe operacje to wprowadzanie, przechowywanie oraz wyprowadzanie informacji cyfrowej. W odróżnieniu od bloków pamięci, dane w rejestrach nie są identyfikowane przez adres. Ze względu na sposób wprowadzania oraz wyprowadzania danych wyróżnia się rejestry:

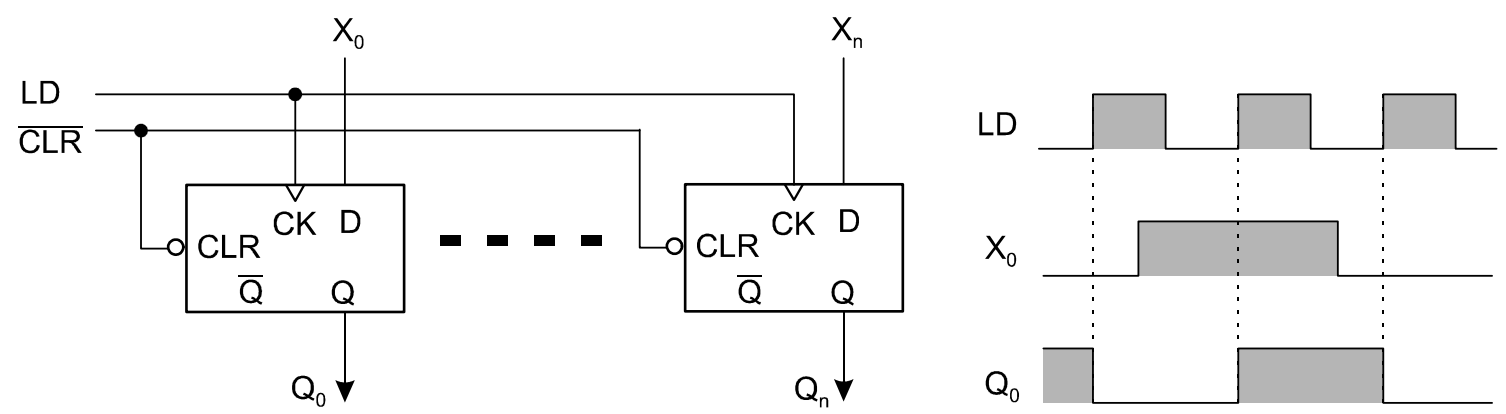
- **PIPO** (*ang. parallel input – parallel output*) – wprowadzanie i wyprowadzanie informacji odbywa się po liniach równoległych.
- **PISO** (*ang. parallel input – serial output*) – wprowadzanie informacji odbywa się po liniach równoległych, wyprowadzanie w sposób szeregowy.
- **SIPO** (*ang. serial input – parallel output*) – wprowadzanie informacji odbywa się w sposób szeregowy, wyprowadzanie po liniach równoległych.
- **SISO** (*ang. serial input – serial output*) – wprowadzanie i wyprowadzanie informacji odbywa się w sposób szeregowy.
- **Rejestry uniwersalne** – łączą możliwości rejestrów PIPO, PISO, SIPO i SISO oraz umożliwiają przesuwanie danych w obie strony.

## Rejestry PIPO

W rejestrach PIPO operacje na każdym bicie danych wykonywane są niezależnie (równolegle) od pozostałych bitów. W praktyce spotyka się zarówno rejestry PIPO zapisywane poziomem na wejściu sterującym (tzw. rej. zatrzaskowe) jak i rejestry zapisywane zboczem.



**Rys. 4.38.** Budowa i przebiegi czasowe w rej. PIPO zapisywanym poziomem 1. Rejestr tego typu jest dostępny w układach scalonych 7475.



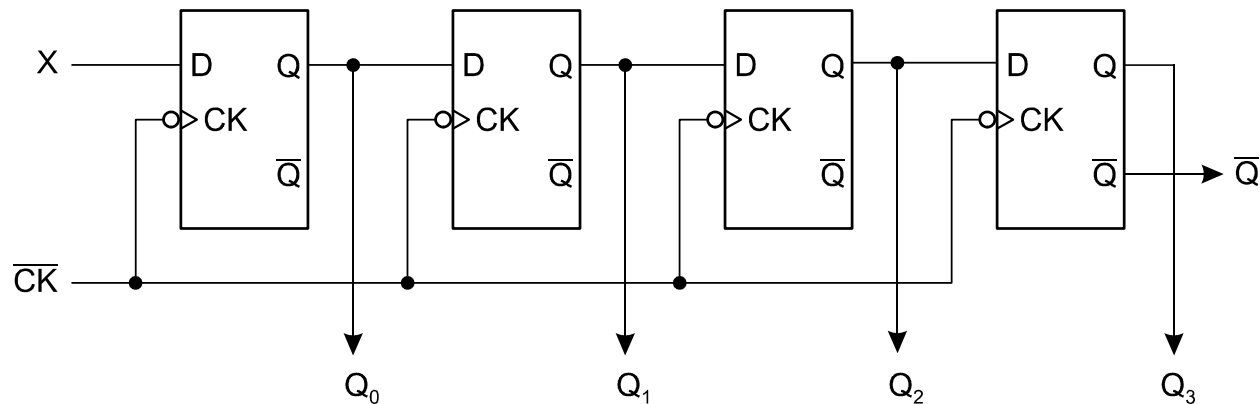
**Rys. 4.39.** Budowa i przebiegi czasowe w rej. PIPO zapisywanym dodatnim zboczem. Rejestr tego typu jest dostępny w układach scalonych 74174.

## Rejestry SIPO i SISO

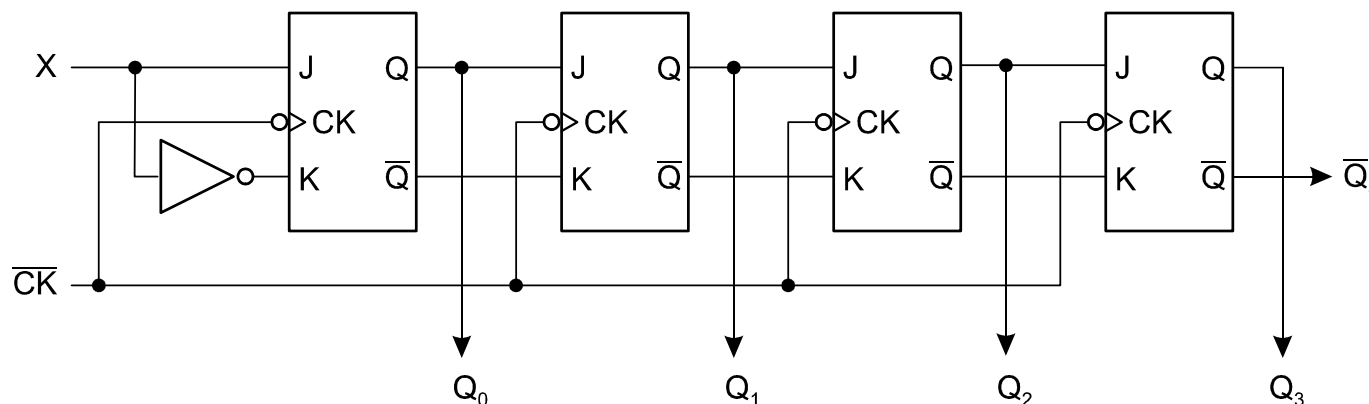
Rejestry SIPO i SISO są rejestrami przesuwającymi. Rejestry tego typu buduje się najczęściej z przerzutników D wyzwalanych zboczem.

➤ Jeżeli na zewnątrz bloku funkcjonalnego wyprowadzono wyjście tylko ostatniego przerzutnika, to układ może pełnić tylko funkcję rejestru SISO (np. 8-bitowy rejestr w układzie scalonym 7491).

➤ W rejestrze SIPO wyprowadzone są wyjścia wszystkich przerzutników (np. 8-bitowy rejestr w układzie scalonym 74164).



**Rys. 4.40.** Schemat podstawowego układu rejestru SIPO zbudowanego z przerzutników D.

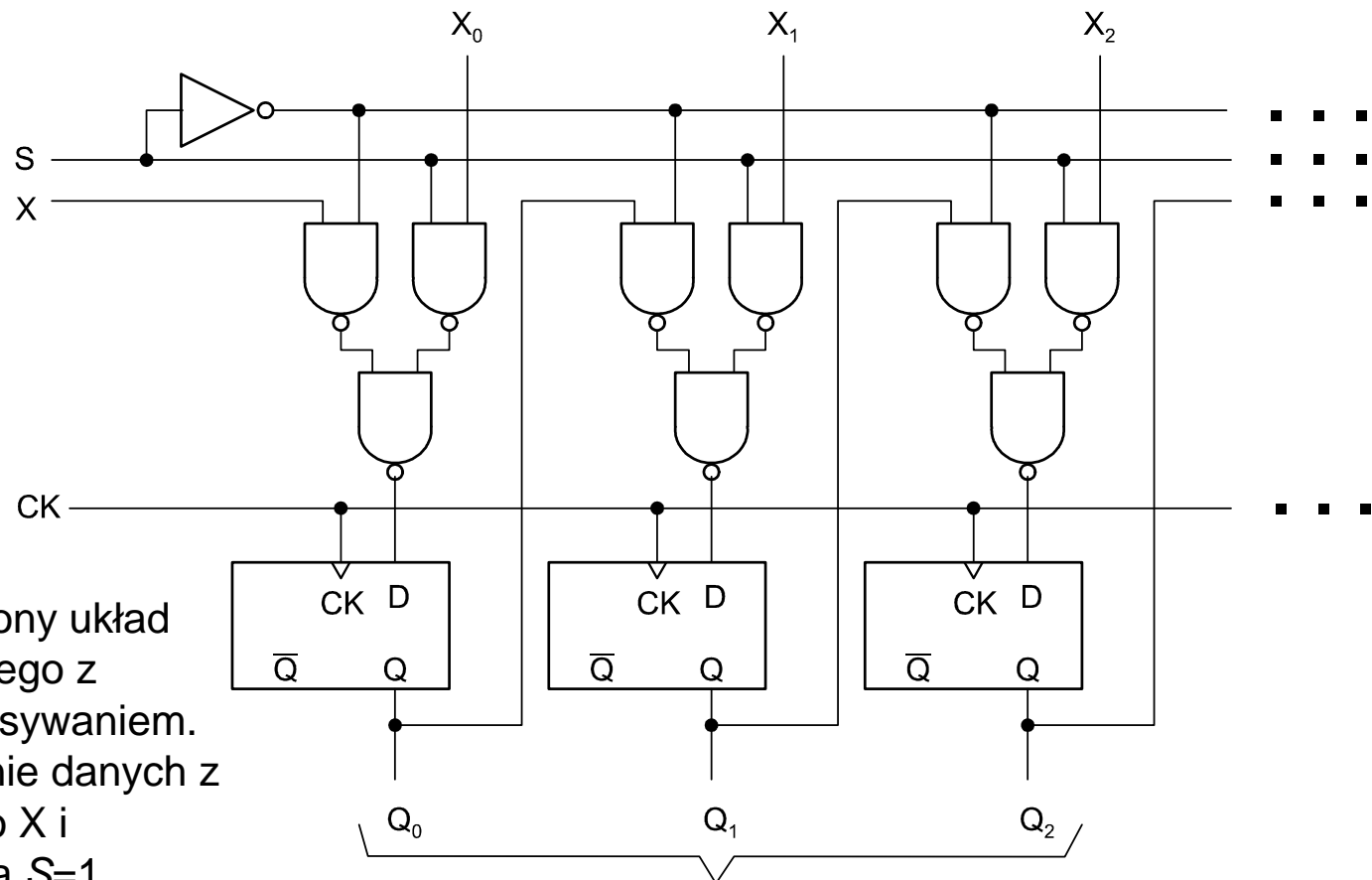


**Rys. 4.41.** Realizacja rejestru SIPO z przerzutników JK-MS.

## Rejestry z wprowadzaniem równoległym (m.in. PISO)

Możliwe są dwa warianty wprowadzania danych z wejść równoległych:

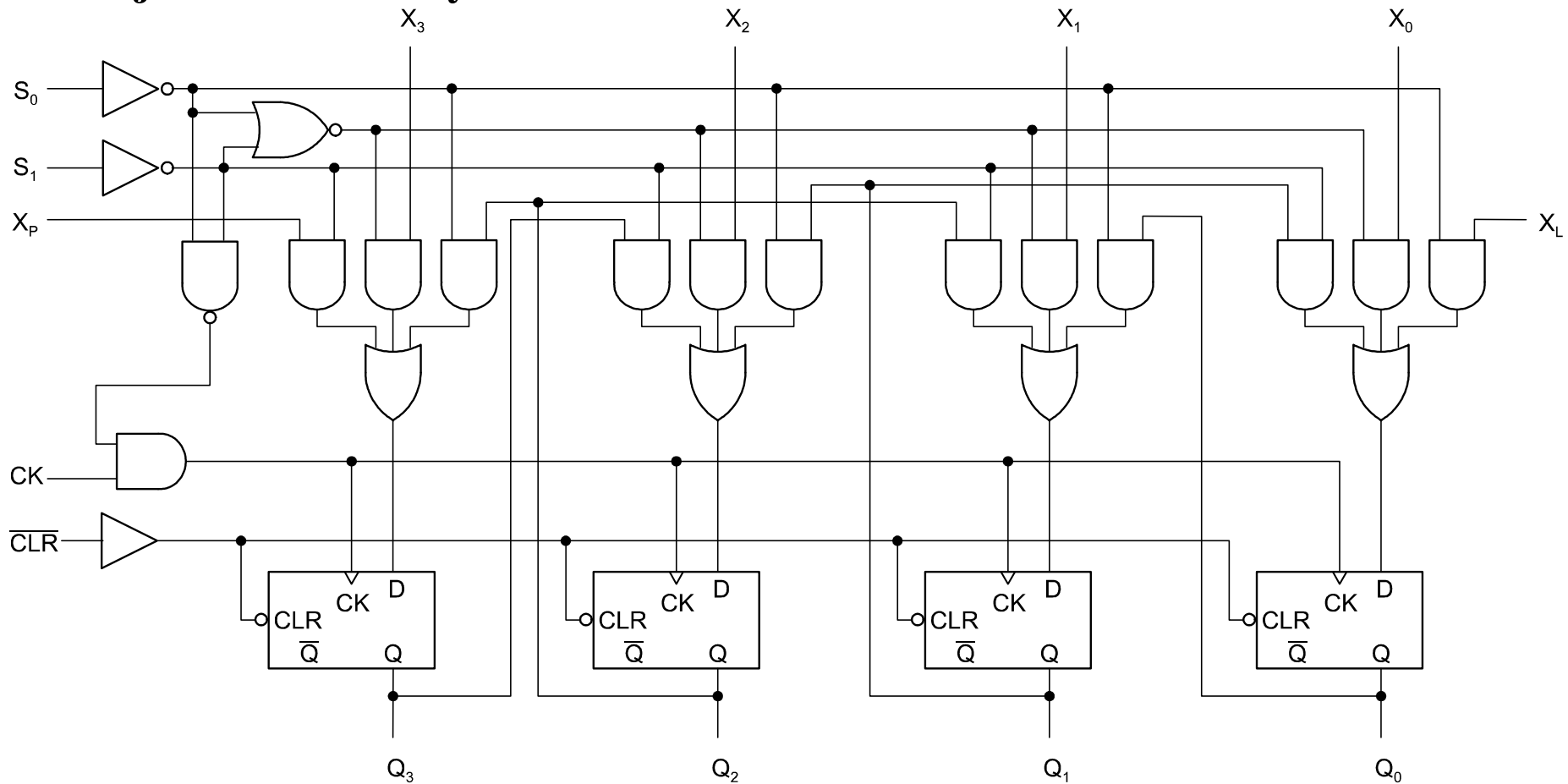
- asynchroniczne – przy wykorzystaniu wejść przerzutników *CLR* i *PR* jak w licznikach z asynchronicznym ustawianiem stanu początkowego (rys. 4.30),
- synchroniczne – wymaga multiplekserów przyłączających wejścia *D* przerzutników do wyjść *Q* poprzedniego stopnia albo do linii *X* wejścia równoległego.



**Rys. 4.42.** Uproszczony układ rejestru przesuwającego z synchronicznym zapisywaniem. Dla  $S=0$  wprowadzenie danych z wejścia szeregowego  $X$  i przesuw w prawo; dla  $S=1$  przepisanie stanów wejść  $X_0X_1X_2\dots$  na wyjścia  $Q_0Q_1Q_2\dots$ .

W rej. PISO wyprowadzone tylko wyjście ostatniego stopnia, w rej. wielofunkcyjnym - wszystkie wyjścia  $Q_i$

# Rejestr uniwersalny



**Rys. 4.43.** Schemat rejestru uniwersalnego 74194. Zerowanie jest asynchroniczne, wprowadzanie z wejść równoległych - synchroniczne.

**Tabela. 4.17.** Tablica przejść rejestru z rys. 4.43.

$\overline{CLR}$	$S_0$	$S_1$	$CK$	mikrooperacja
0	—	—	—	$\mathbf{Q} := 0$
1	0	0	↑	nic nie rób
1	0	1	↑	przesuw w lewo
1	1	0	↑	przesuw w prawo
1	1	1	↑	$\mathbf{Q} := \mathbf{X}$

## 4.6. Układy arytmetyczne

Do grupy podstawowych układów arytmetycznych należą sumatory, komparatory cyfrowe, jednostki arytmetyczno logiczne oraz układy kontroli parzystości. Bardziej złożone operacje akumulacji sumy, mnożenia i dzielenia wykonywane są przez układy zbudowane z szeregu prostszych układów arytmetycznych.

### 4.6.1. Sumator

Elementarny sumator jednobitowy z uwzględnieniem przeniesienia arytmetycznego wykonuje następując operacje:

$$S_i = A_i \oplus B_i \oplus P_i, \quad (4.11)$$

$$P_{i+1} = A_i B_i + A_i P_i + B_i P_i. \quad (4.12)$$

gdzie:

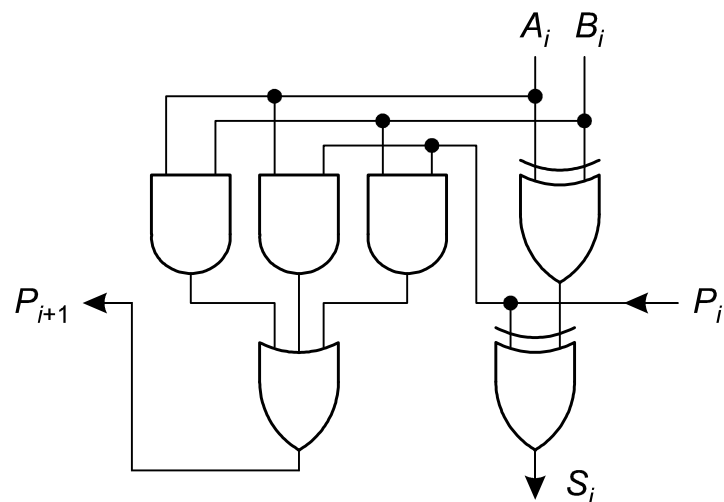
$A_i, B_i$  – wejścia  $i$ -tego bitu liczb **A** i **B** reprezentujących liczby całkowite w NKB lub U2,

$P_i$  – przeniesie arytmetyczne wygenerowane przez sumator dodający bity ( $i-1$ ),

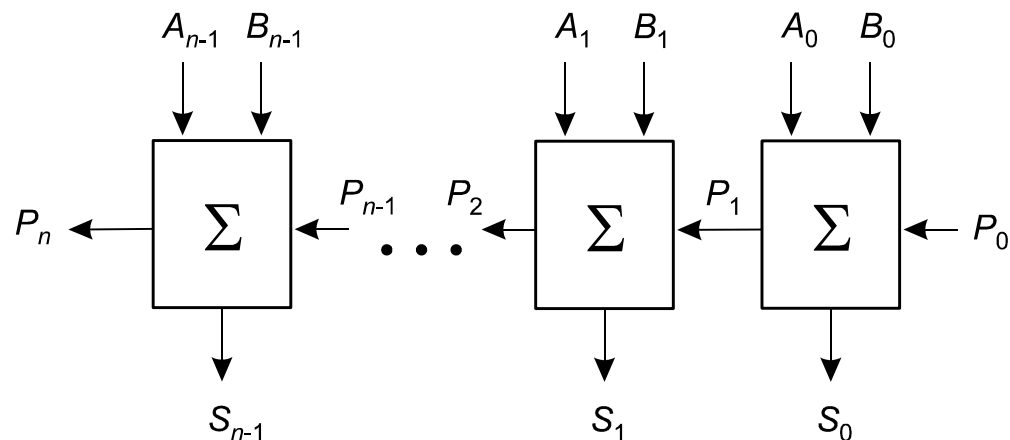
$P_{i+1}$  – wyjście przeniesienia arytmetycznego dla sumatora dodającego bity ( $i+1$ ),

$\oplus$  – suma modulo dwa, funkcja EXOR,

$+$  – suma logiczna, alternatywa, funkcja OR.



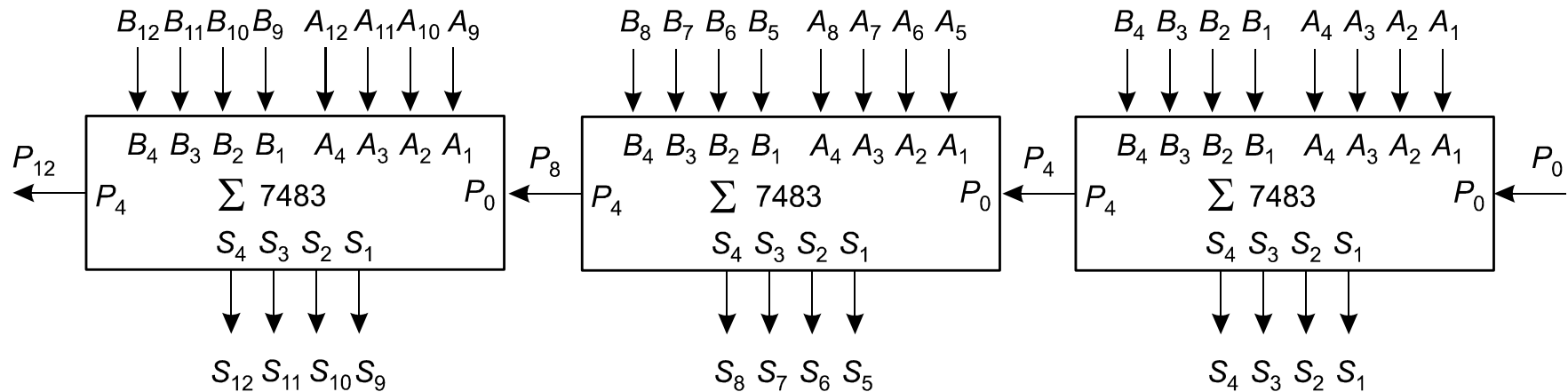
**Rys. 4.44.** Realizacja sumatora jednobitowego.



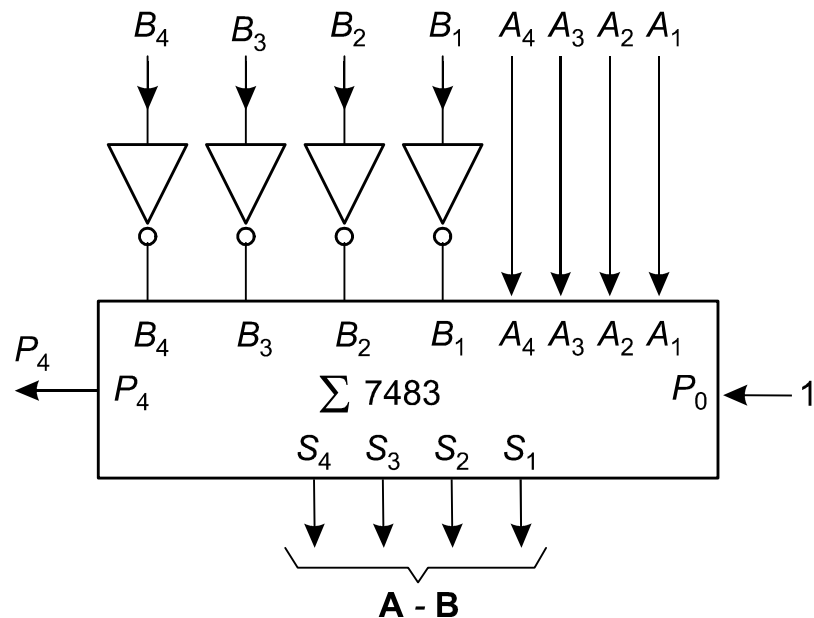
**Rys. 4.45.** Realizacja sumatora wielobitowego z przeniesieniami szeregowymi przy wykorzystaniu sumatorów jednobitowych.

W układach połączonych w wielobitowe sumatory z przeniesieniami szeregowymi dochodzi do sumowania się czasów propagacji sygnałów od wejść do wyjścia przeniesienia wewnątrz elementarnych sumatorów 1-bitowych. W praktyce w układach wielobitowych stosuje się przeniesienia równoległe lub rozwiązania pośrednie (szeregowo-równoległe).

Układ scalony 7483 jest 4-bitowym sumatorem z przeniesieniami równoległymi wewnątrz układu. Realizacja przeniesień równoległych pomiędzy różnymi układami scalonymi wymaga dodatkowych bloków kombinacyjnych.



**Rys. 4.46.** Schemat logiczny sumatora 12-bitowego z przeniesieniami szeregowymi pomiędzy czterobitowymi sumatorami 7483.

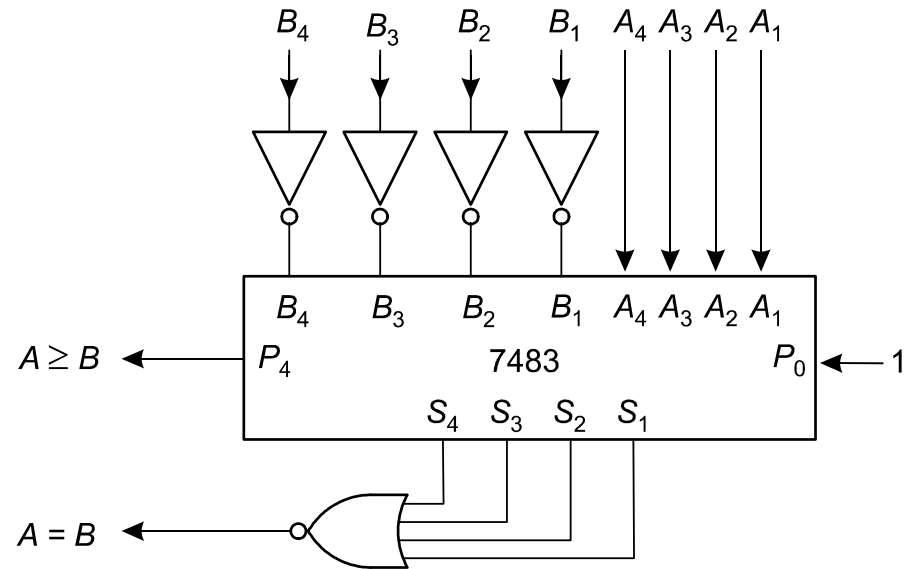


**Rys. 4.47.** Realizacja układu odejmującego (subtraktora) przy wykorzystaniu sumatora. Układ pracuje z liczbami w kodzie NKB (naturalnym kodzie binarnym) lub U2 (kodzie uzupełnień do dwóch).

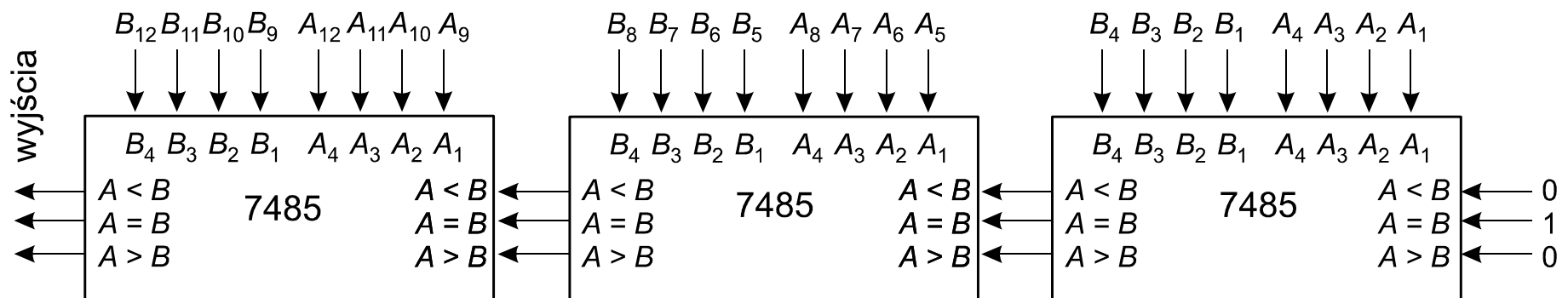


## 4.6.2. Komparator

Komparatory cyfrowe służą do porównywania wielkości dwóch liczb. Komparator można zrealizować przy wykorzystaniu sumatora, np. 7483. Produkowane są także gotowe komparatory do porównywania dwóch liczb bez znaku w NKB, które można łączyć szeregowo w zespoły o wymaganej liczbie bitów.



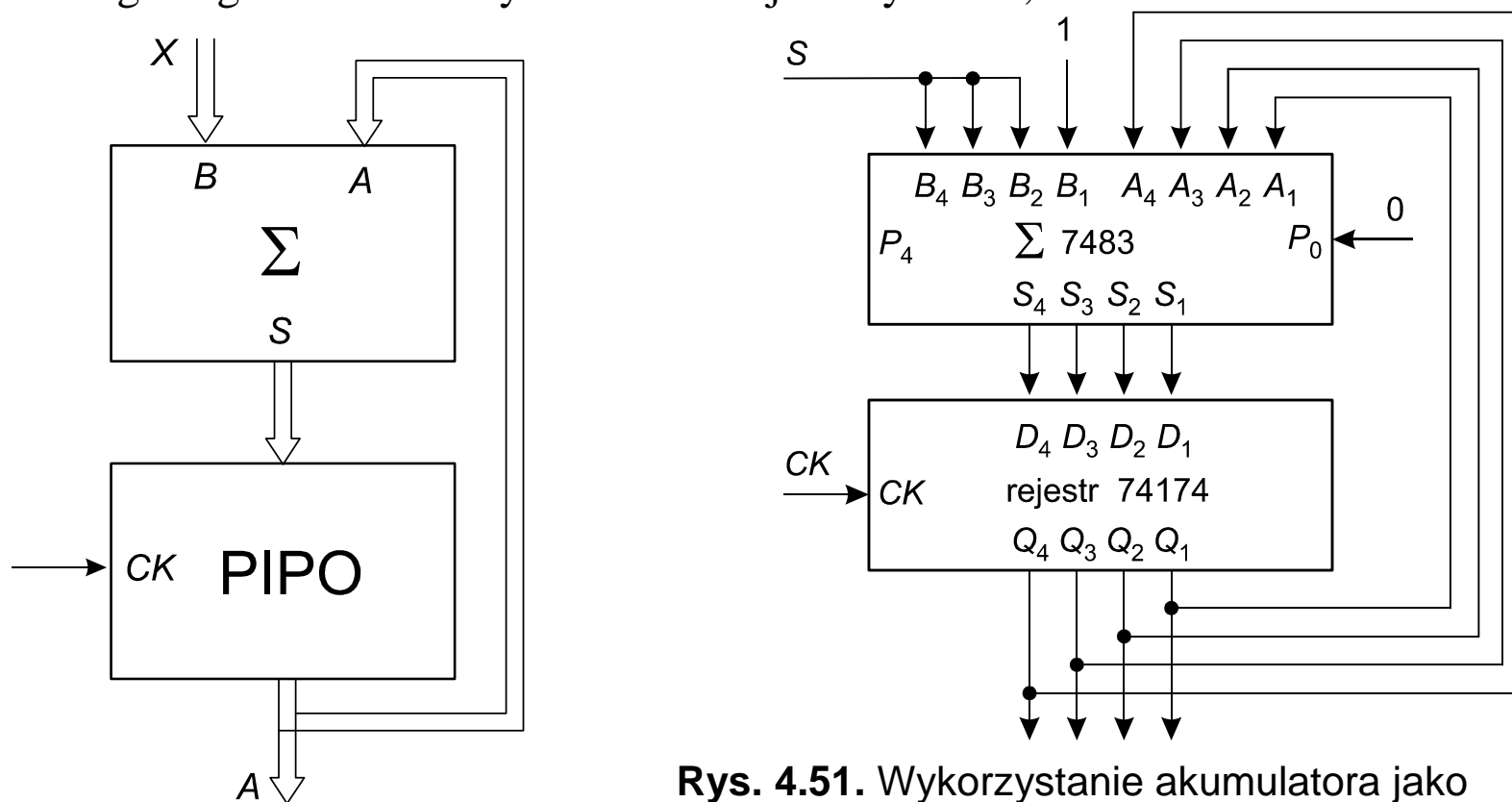
**Rys. 4.48.** Realizacja komparatora 4-bitowego w NKB przy wykorzystaniu sumatora 7483.



**Rys. 4.49.** Realizacja komparatorów o dowolnej liczbie bitów przez szeregowe łączenie komparatorów 4-bitowych typu 7485.

## 4.6.3. Akumulator

Akumulator, zwany także sumatorem akumulacyjnym, jest połączeniem sumatora z rejestrem równoległym zapisywanym zboczem zegarowym. W akumulatorze po podaniu odpowiedniego zbocza (lub impulsu) na wejście sterujące wykonywana jest mikrooperacja  $A := A + X$ . Akumulator w połączeniu z układami sterowania może pełnić liczne funkcje: licznika o zadanym kroku, rejestru przesuwanego, sumatora szeregowego liczb złożonych z dowolnej liczby bitów, itd.



Rys. 4.50. Struktura akumulatora

Rys. 4.51. Wykorzystanie akumulatora jako licznika rewersyjnego, który dla  $S=0$  zlicza do przodu, zaś dla  $S=1$  do tyłu.

## 4.7. Literatura

---

- [1] P. Misiurewicz, *Układy automatyki cyfrowej*, Wydawnictwa Szkolne i Pedagogiczne, Warszawa, 1984.
- [2] H. Kamionka-Mikuła, H. Małysiak, B. Pochopień, *Synteza i analiza układów cyfrowych*, Wydawnictwo Pracowni Komputerowej Jacka Skamierskiego, Gliwice 2006.
- [3] W. Sasal, *Układy scalone serii UCA64/UCY74. Parametry i zastosowania*, WKiŁ, Warszawa 1990.
- [4] C. Zieliński, *Podstawy projektowania układów cyfrowych*, PWN, Warszawa 2003.
- [5] W. Traczyk, *Układy cyfrowe. Podstawy teoretyczne i metody syntezy*, WNT, Warszawa 1986.
- [6] J. Kalisz, *Podstawy elektroniki cyfrowej*, WKiŁ, Warszawa 2002.
- [7] M. Molski, *Wstęp do techniki cyfrowej*, WKiŁ, Warszawa 1989.
- [8] A. Barczak, J. Florek, T. Sydoruk, *Elektroniczne techniki cyfrowe*, VIZJA PRESS&IT Sp. z o.o., Warszawa 2006.
- [9] W. Głocki, *Układy cyfrowe*, Wydawnictwa Szkolne i Pedagogiczne, Warszawa 2008.