

Selekcyjne bloki funkcjonalne

Rejestr - układ złożony z tylu przerzutników ile bitów zapamiętuje

Rejestr 4-bitowy

z asynchronicznym zerowaniem i ustawianiem

D_3, D_2, D_1, D_0 - stano wejściowe, które zostaną zapamiętane przez rejestr
(wejścia informacyjne)

Q_3, Q_2, Q_1, Q_0 - stano wyjściowe, pamiętane przez rejestr

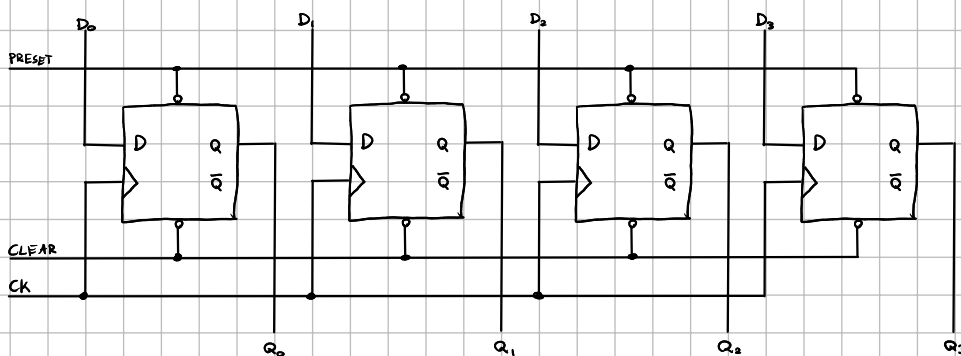
Wspólny sygnał zegarowy dociera do wszystkich przerzutników

Clear - niezależnie od stanu i wejścia wpisuje 0 do przerzutnika

Preset - niezależnie od stanu i wejścia wpisuje 1 do przerzutnika

Clear i Preset działają asynchronicznie (natychmiast, niezależnie od zegara)

Aktywowane stanem niskim (negacja przy wejściach)

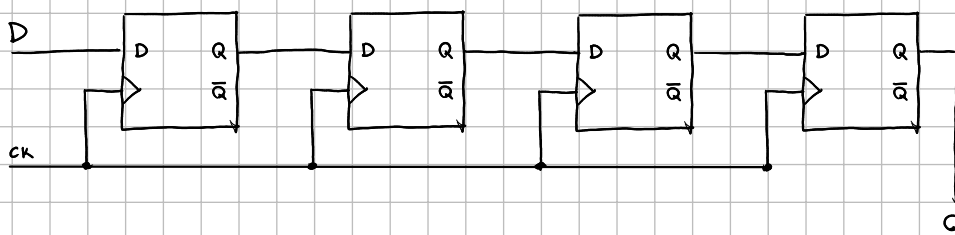


Preset i Clear mogą też być zrealizowane synchronicznie, wtedy nadpisują stan przerzutnika w momencie jego aktywacji sygnałem zegara

Rejestr przesuwający

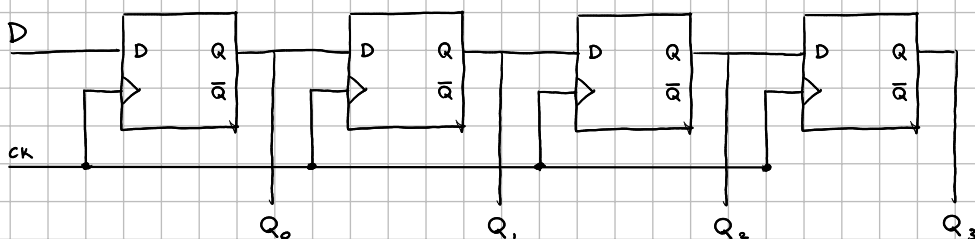
W takcie zegara, wszystkie bity przesuwają się o 1 bit

Stano usuwa się po 1 bicie na takt zegara



Rejestr przesuwający z dostępem równoległym

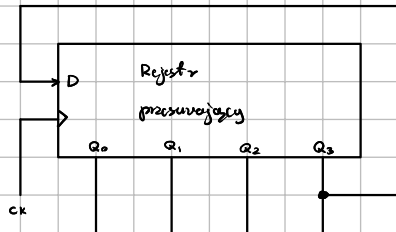
Umożliwia dodatkowo odczyt



Rodzaje rejestrów

- równoległy - równoległy równoległy zapis i odczyt
- szeregowo - równoległy szeregowy zapis, równoległy odczyt
- równoległy - szeregowy równoległy zapis, szeregowy odczyt
- szeregowo - szeregowy szeregowy zapis i odczyt

Licznik pierścieniowy



Zapętla się w ten sposób, zależnie od początkowo zapisanego słowa, np.

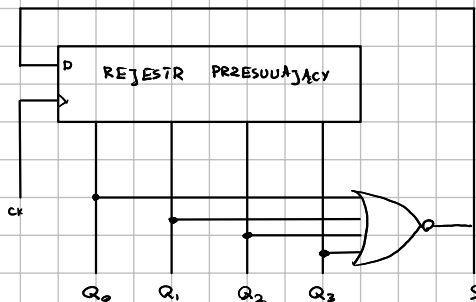
1	0	0	0	0	1	1	1
0	1	0	0	1	0	1	1
0	0	1	0	1	1	0	1
0	0	0	1	1	1	1	0
1	0	0	0	0	1	1	1

Licznik pierścieniowy sumujący

z krążącą jedynką

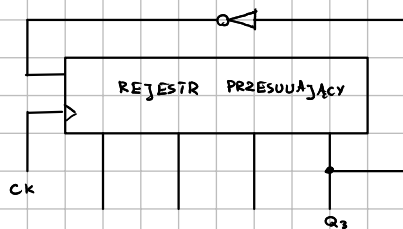
Odporny na problem z utratą zasilania

Upisuje 1 na wejście kiedy wszystkie wyjścia dające 0 i cykl się powtórzy



				S
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1
1	0	0	0	0

Licznik Johnsona

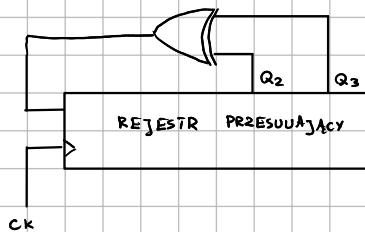


Stopniowo uzupełnia się jedynkami po czym stopniowo uzupełnia się zerami

1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1
0	0	0	0

kod Johnsona

Licznik liniowy (liczącowy)



Zapętla z powrotem wybrane bity,
powstające przez jakąś funkcję logiczną

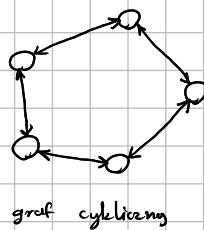
Generator liczb pseudolosowych
(lepszy dla dłuższych słów)

Liczniki bardzo często buduje się z rejestrów

Licznik - automat opisany przez graf cykliczny

Liczniki zliczające impulsy wejściowe

- synchroniczne (równoległe)
- asynchroniczne (szeregowe)



Liczniki zliczające cykle zegara

- Timery

• Następnikowe $\rightarrow 1, 2, 3, 4, \dots$

• Poprzednikowe $\rightarrow 4, 3, 2, 1, \dots$

• Reversyjne \rightarrow kierunku definiowany przez wejście

Realizacja licznika synchronicznego następnikowego na przerzutnikach T

Tabela przejść

Mapy Karnaugh'a funkcji wzbudzeń dla przerzutników T
(dla kolejnych bitów słowa wyjściowego)

x Q_3, Q_2, Q_1, Q_0	0	1
000	000	001
001	001	010
010	010	011
011	011	100
100	100	101
101	101	110
110	110	111
111	111	000

x Q_3, Q_2, Q_1, Q_0	0	1
000	0	0
001	0	0
011	0	1
010	0	0
110	0	0
111	0	1
101	0	0
100	0	0

x Q_3, Q_2, Q_1, Q_0	0	1
000	0	0
001	0	1
011	0	1
010	0	0
110	0	0
111	0	1
101	0	1
100	0	0

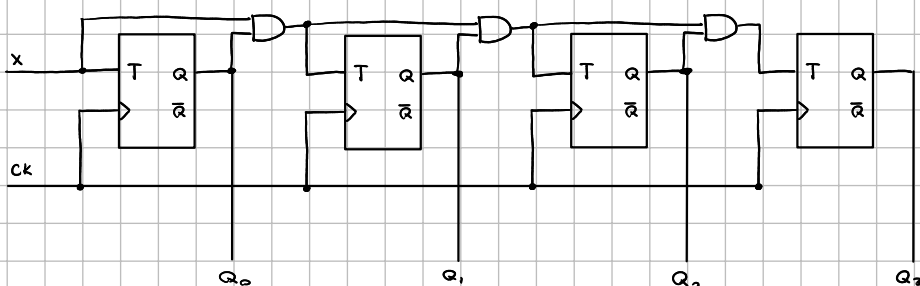
x Q_3, Q_2, Q_1, Q_0	0	1
000	0	1
001	0	1
011	0	1
010	0	1
110	0	1
111	0	1
101	0	1
100	0	1

$$T_2 = x Q_1 Q_0$$

$$T_1 = x Q_0$$

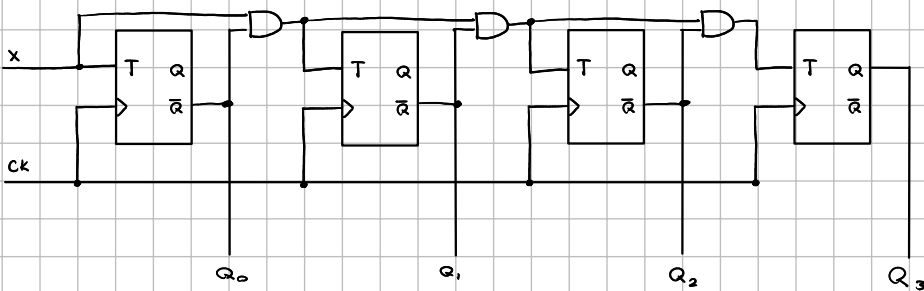
$$T_0 = x$$

$$T_i = x Q_0 Q_1 \dots Q_{i-1}$$

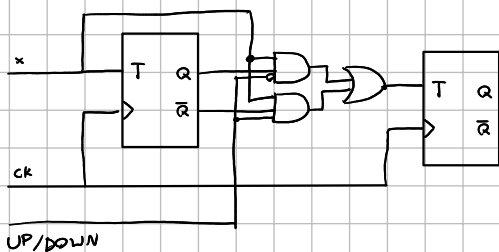


Analogicznie można zrobić licznik poprzedniowy

$$T_i = x \overline{Q_0} \overline{Q_1} \dots \overline{Q_{i-1}}$$

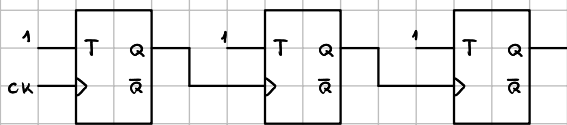


Licence reversing

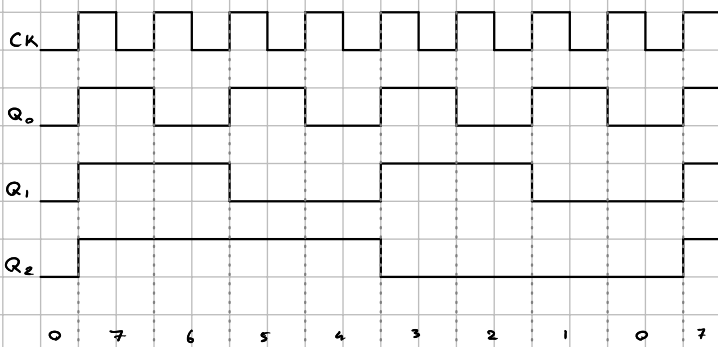


Polazenie następnikowego z poprzedniwym
z sygnałem wybierającym

Licznik szeregowy poprzednikowy (asynchroniczny)



Każdy kolejny blok zmienia się z 2 razy dłuższym czasem



Opoznięcia wynikające z czasu propagacji psują efekt przy asynchronicznym odczycie

Częstotliwość graniczna licznika

Maksymalna częstotliwość sygnału zegarowego, przy której licznik przyjmuje kolejne różne stany.

Najszerszy zegar, przy którym licznik nadaje

Dla licznika równoległego

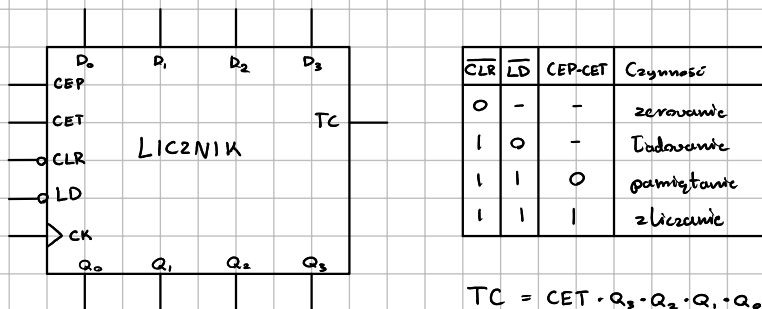
Dla licznika szeregowego
 $T_{min} = n \cdot t$

τ - czas propagacji

W wejścia i wyjścia liczników synchronicznych

- wejścia dostępu CEP i CET - aktywne 1 (Count Enable Parallel, Count Enable Trickle)
- wejście ładowania LD - aktywne 0 (Load)
- wejście zerowania CLR - aktywne 0 (Clear)
- wyjście przeniesienia TC (Terminal Count)

Oznaczenie abstrakcyjnego licznika 4-bitowego

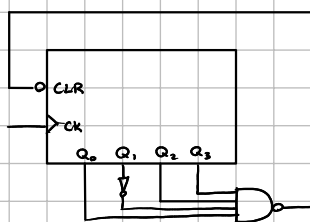


Doświ sygnali TC można konstruować liczniki (połączając dwa 4-bitowe w jeden 8-bitowy)

CLR ma najwyższy priorytet

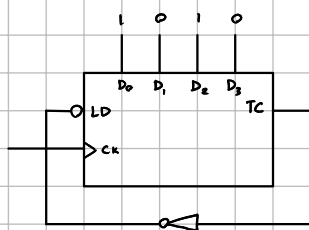
CLR i LD mogą być zrealizowane synchronicznie lub asynchronicznie

Stworzenie cyklu licznika przez CLR



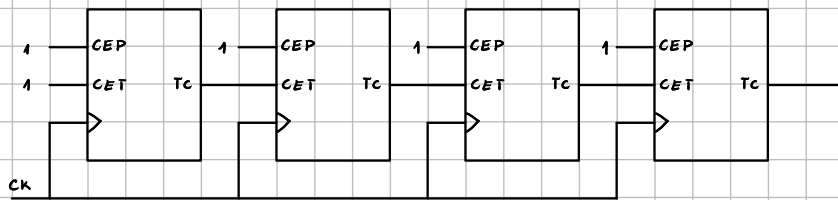
Stworza cykl od 0 do 13
Kiedy na wyjściu pojawi się
 $1101_2 = 13_{\text{dec}}$ to zacznie
liczyć od nowa

Stworzenie cyklu licznika przez LD



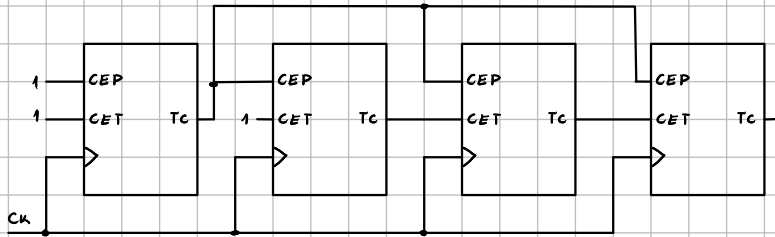
Stworza cykl licznika od 0 do 15
Po dojściu do $15 = 1111_2$ wygeneruje przeniesienie
po czym weźmie $0101_2 = 5$ i zacznie liczyć dalej

Składanie liczników



Problematyczne, bo TC musi przepływać przez wszystkie liczniki
a to ogranicza t_{max}

Szybsze połączenie liczników



TC ma 1b sygnału zegara na propagację do ostatniego członu