

Multipleksery i rejestry

Multipleksery – wiadomości ogólne

Multipleksier – jest układem posiadającym: wejścia danych, wejście adresowe oraz wyjście. Za pomocą wejścia adresowego można dokonać wyboru, które z wejść ma zostać przekazane na wyjście, najczęściej używanym formatem adresu jest naturalny kod binarny lub „1 z N”. Multipleksery mają często dodatkowe wejście sterujące, zwane wejściem strobowym lub zezwalającym. Jeśli wejście to jest w stanie niskim, to multipleksier działa tak jak określono powyżej, natomiast gdy jest w stanie wysokim, to niezależnie od wejść stan wyjścia jest stały i równy „0”.

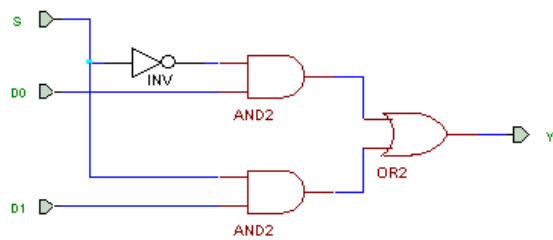
Najprostszym przykładem multipleksiera jest multipleksier 2:1. W zależności od stanu na wejściu adresowym S, przekazuje on na wyjście stan z wejścia D0 lub D1.

Jego działanie można opisać funkcją:

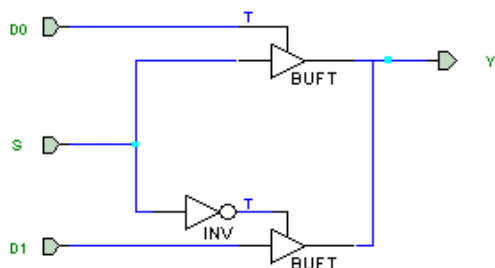
$$Y = \begin{cases} D_0 & \text{dla } S = 0 \\ D_1 & \text{dla } S = 1 \end{cases} \quad \text{lub zapisując inaczej:}$$

$$Y = D_0 \bar{S} + D_1 S$$

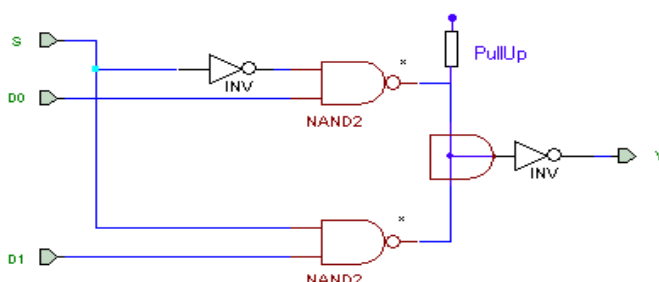
Poniżej przedstawiono układ złożony z bramek trójstanowych, spełniający identyczną



funkcję. Połączenie wyjść bramek trójstanowych, jest możliwe ponieważ, nigdy nie są one jednocześnie w stanie aktywnym (zawsze jedna z nich jest w stanie wysokiej impedancji). Pozwala to na uproszczenie całego układu.



Identycznie działający układ można utworzyć używając bramek NAND z wyjściami typu open collector. Układ ten umożliwia utworzenie tzw. sumy montażowej.

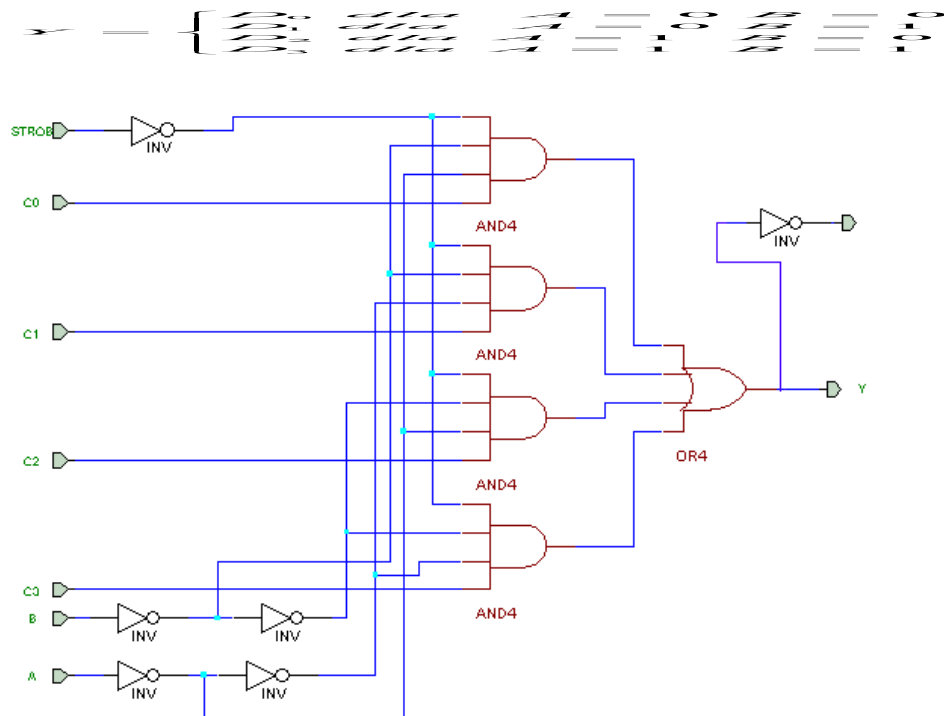


Potencjał wyjścia przyjmuje stan wysoki tylko wówczas, gdy wyjścia wszystkich bramek są w stanie wysokim, co w logice dodatniej odpowiada funkcji AND. Korzystając z prawa de Morgana można postać ilorazową przekształcić w sumę.

$\overline{SD_0} \cdot \overline{SD_1} = \overline{SD_0 + SD_1}$ jeżeli zanegujemy to wyrażenie to otrzymamy funkcję logiczną:

$Y = D_0\overline{S} + D_1S$ czyli identyczną jak dla poprzednich dwóch układów.

Multiplekser 4:1



Zbudowany multiplekser ma 4 wejścia i 2 wyjścia (wyjście Y oraz jego negacja). Zapiszmy tabele możliwych stanów.

wejścia		Wyjścia
Adresowe	Strob.	
BA	S	Y
XX	1	0
00	0	X_0
01	0	X_1
10	0	X_2
11	0	X_3

$D_0 \backslash AB$	00	01	11	10
0	0	0	0	0
1	1	0	0	0

$$Y_0 = \overline{AB}D_0$$

Analogicznie rozpatrując wszystkie wyjścia otrzymujemy :

$$Y_1 = \overline{AB}D_1 ; Y_2 = \overline{AB}D_2 ; Y_3 = \overline{AB}D_3$$

Uwzględniając także wejście strobujujące S otrzymujemy

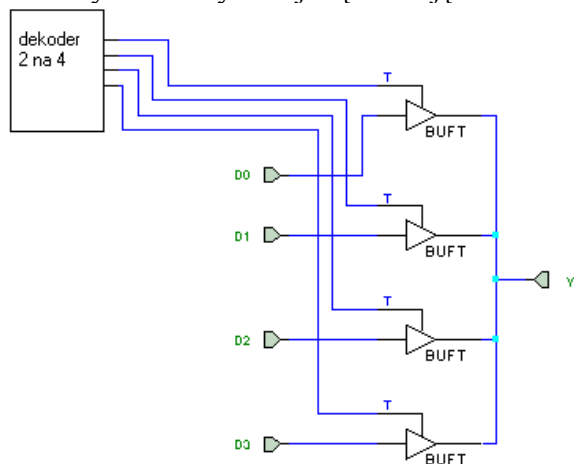
$$Y_1 = \overline{A}BD_1\overline{S} ; Y_2 = A\overline{B}D_2\overline{S} ; Y_3 = ABD_3\overline{S}$$

Wyjście W jest sumą logiczną stanów Y_n .

A zatem funkcja uzyskuje postać:

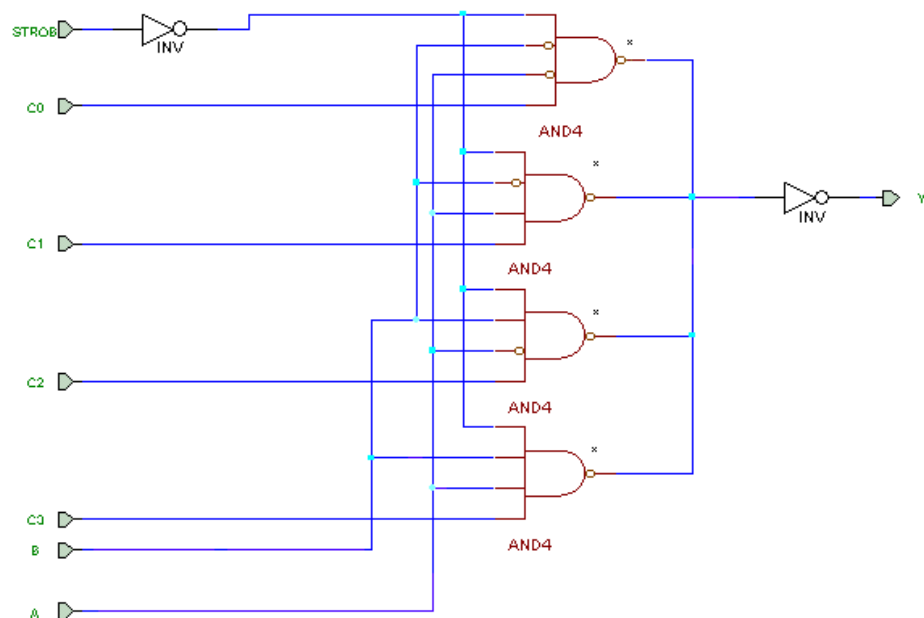
$$Y = Y_0 + Y_1 + Y_2 + Y_3 = \overline{A}BD_0\overline{S} + \overline{A}BD_1\overline{S} + A\overline{B}D_2\overline{S} + ABD_3\overline{S}$$

Poniższy układ wykonuje tę funkcję na bramkach trójstanowych.



Takie uproszczenie układu umożliwia budowa bramek trójstanowych. Sterowanie powoduje, że w danym momencie tylko jedna bramka jest aktywna, a pozostałe pozostają w stanie wysokiej impedancji. Pozwoliło to na połączenie ich wyjść.

Układ przyjmuje jeszcze inną formę gdy użyjemy bramek typu open collector.



Demultiplekser 1:n

Demultiplekser przekazuje daną wejściową C na tylko jedno z wyjść, które jest określone przez wejście adresowe. Często jest wyposażony w wejście strobowe.

Działanie najprostszego demultipleksera 1:2 można przedstawić za pomocą funkcji:

$$Y_0 = \begin{cases} C & \text{dla } A = 0 \\ 1 & \text{dla } A = 1 \end{cases}$$

$$Y_1 = \begin{cases} C & \text{dla } A = 1 \\ 1 & \text{dla } A = 0 \end{cases}$$

Tabela stanów demultipleksera 1:4

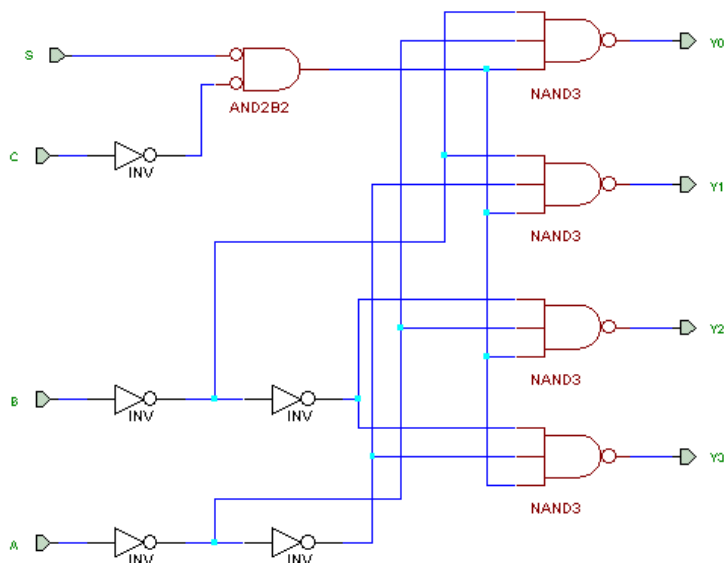
B	A	S	C	Y ₀	Y ₁	Y ₂	Y ₃
X	X	1	X	1	1	1	1
0	0	0	1	0	1	1	1
0	1	0	1	1	0	1	1
1	0	0	1	1	1	0	1
1	1	0	1	1	1	1	0
X	X	X	0	1	1	1	1

Zapiszmy tablice Carnaughta dla Y₀;

AB \ SC	00	01	11	10
00	1	1	1	1
01	0	1	1	1
11	1	1	1	1
10	1	1	1	1

Skąd otrzymujemy: $\overline{Y_0} = \overline{ABSC}$;

Rozpatrując analogicznie z kolejnymi wyjściami A_n konstruujemy układ:



REJESTRY

Wprowadzenie :

Rejestrem nazywany układ służący do przechowywania informacji.

Ze względu na sposób wyprowadzania informacji dzielimy rejestry na :

- Szeregowe umożliwiające szeregowe wprowadzenie i wyprowadzenie danych (tzn. bit po bicie), są to tak zwane rejestry SISO (Serial In Serial Out)
- Równoległe, umożliwiające równoległe wprowadzenie i wyprowadzenie informacji jednocześnie do wszystkich pozycji rejestru, PIPO (Parallel In Parallel Out)
- Szeregowo-równoległe, umożliwiające szeregowe wprowadzenie i równoległe wyprowadzenie informacji, SIPO (Serial In Parallel Out)
- Równoległo – szeregowe, umożliwiające równoległe wprowadzenie i szeregowe wyprowadzenie informacji. PISO (Parallel In Serial Out)

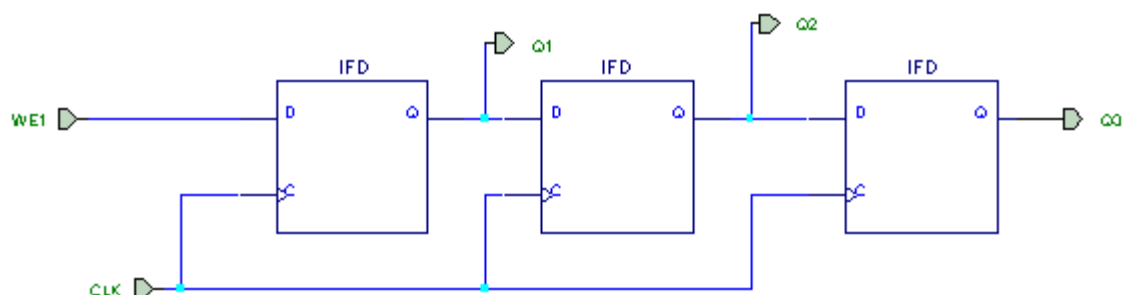
Rejestry można podzielić także na asynchroniczne i synchroniczne . Jednakże najczęściej stosowane są rejestry synchroniczne.

Parametrami charakteryzującymi rejestr są:

- Długość rejestru, równa liczbie przerzutników N
- Pojemność rejestru, równa 2^N
- Szybkość rejestru; w przypadku rejestru równoległego będzie to czas wprowadzania lub wyprowadzania informacji , natomiast w przypadku rejestru szeregowego maksymalna możliwa częstotliwość impulsów przesuwających, przy której nie następuje zniekształcenie informacji.

REJESTR - Układ SIPO

Projektujemy rejestr o wejściu szeregowym i wyjściu równoległym. Najprostszy taki rejestr można uzyskać łącząc przerzutniki. Niech to będą w naszym przypadku przerzutniki typu D (można również zaprojektować taki układ na przerzutnikach np. JK)

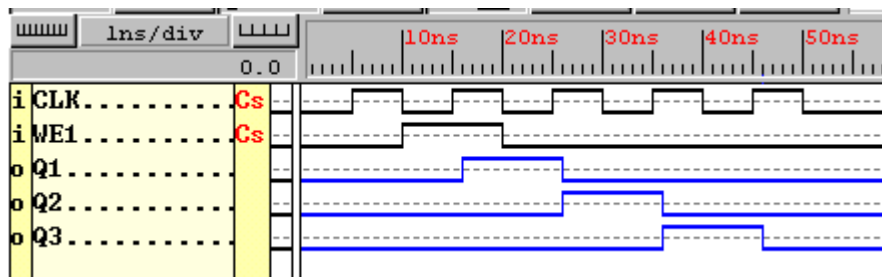


Jeżeli na wejściu WE1 podamy jedynkę a potem cały czas stan niski, to ta jedynka będzie wędrować wraz z kolejnymi taktami zegara poprzez wszystkie przerzutniki i pojawi się kolejno na wyjściach Q1, Q2 i Q3.

Jeśli zasymulujemy działanie takiego układu podając na wejściu WE1 impuls wysoki czyli zapisując kod programu:

```
delete_signals
restart
wfm CLK @0=0 (5ns=1 5ns=0)*5
wfm WE1 @0=0 @10ns=1 @20ns=0
watch Q1 Q2 Q3
sim 100ns
```

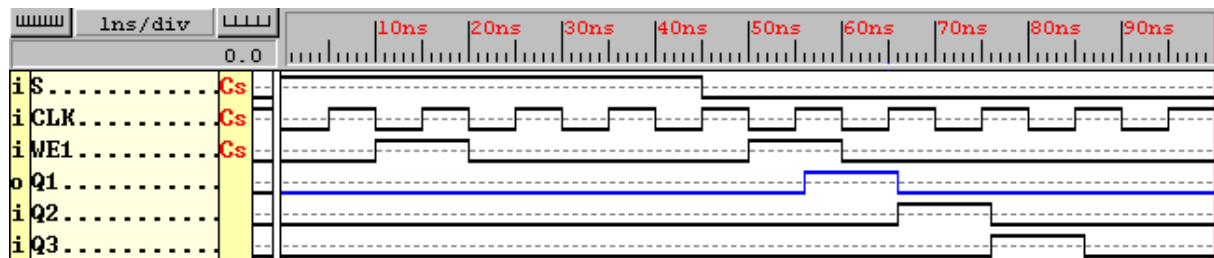
to przebiegi będą wyglądały jak poniżej:



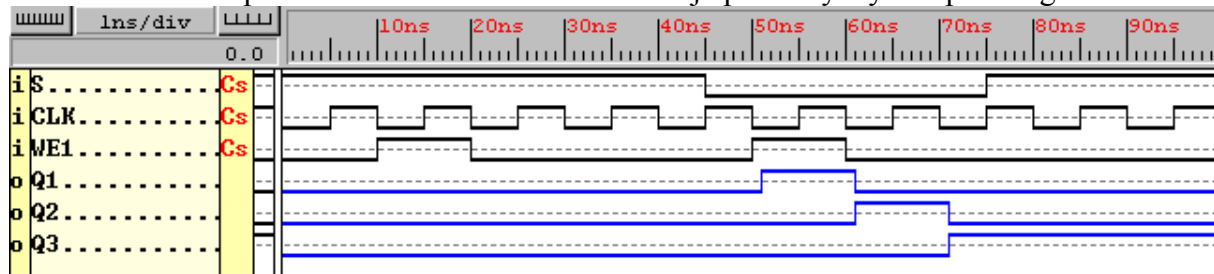
Widać więc, że stan wysoki przesuwa się (stąd nazwa rejestrów – **rejestry przesuwne**). W ten sposób można skonstruować dzielnik częstotliwości łącząc wyjście Q3 z wejściem przerzutnika WE1. Otrzymamy wtedy pętlę powtarzającą impuls wysoki co trzeci takt zegarowy. Współczynnik wypełnienia przebiegu wyjściowego wynosi 1/3. Gdybyśmy wpisali do rejestru dwie jedynki otrzymalibyśmy współczynnik wypełnienia 2/3.

Układ SIPO z dodatkowym sygnałem zezwolenia zegara .

Jest to układ podobny jak powyższy, z tą różnicą, że dodajemy sterowanie sygnałem zegara. Takie sterowanie można wykonać za pomocą bramki sterującej sygnałem zegara (np. bramki AND). Jak na poniższym schemacie, gdzie zastosowano taktowanie jednego zegara dla całego układu. Wejście na blok rejestru jest oznaczone W1 i jest sterowane z pomocniczego przerzutnika w celu pokazania faktycznych opóźnień w układzie. Sterowanie sygnałem zegara odbywa się poprzez bramkę AND. Zegar jest aktywny co drugi takt zegara (bramka AND jest sterowana z licznika modulo 2).

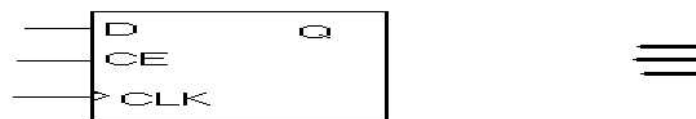


Jeżeli wyłączymy zegar w chwili t (dla naszego przykładu w 75ns) to otrzymamy zachowanie stanu przerzutników w chwili t co ilustruje poniższy wykres przebiegów.

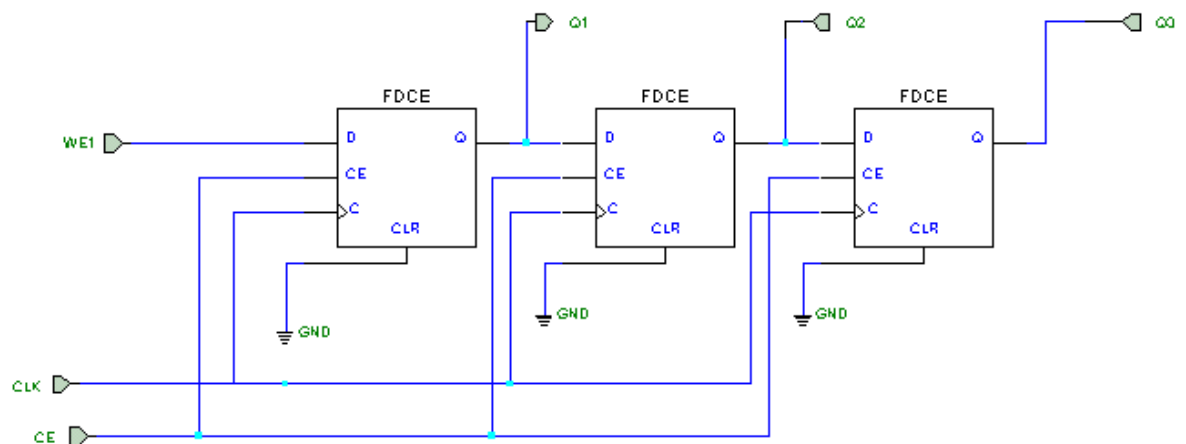


Ponownie załączając zegar (podając na wejście S stan niski) Otrzymamy na wyjściach przerzutników stany z chwili t (w której zegar został wyłączony – tu 75 ns).

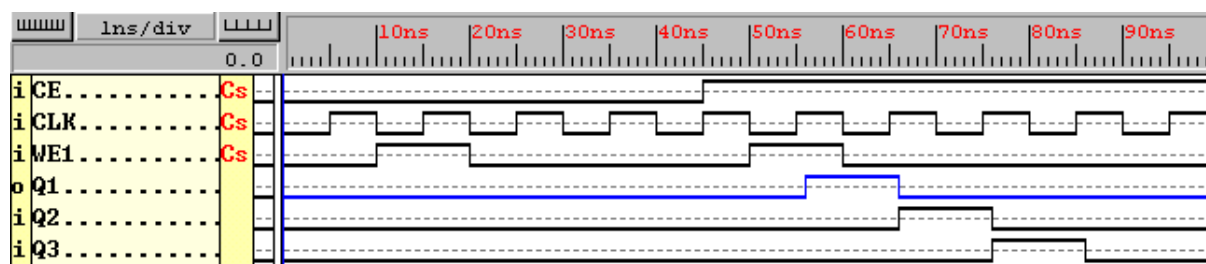
Należy zdać sobie sprawę, iż przerzutnik D z wejściem CE jest zbudowany z prostego przerzutnika typu D z dołączonym multiplexserem 2 na 1:



Tworząc rejestr oparty na przerzutnikach z wejściem CE będzie on wyglądał następująco:



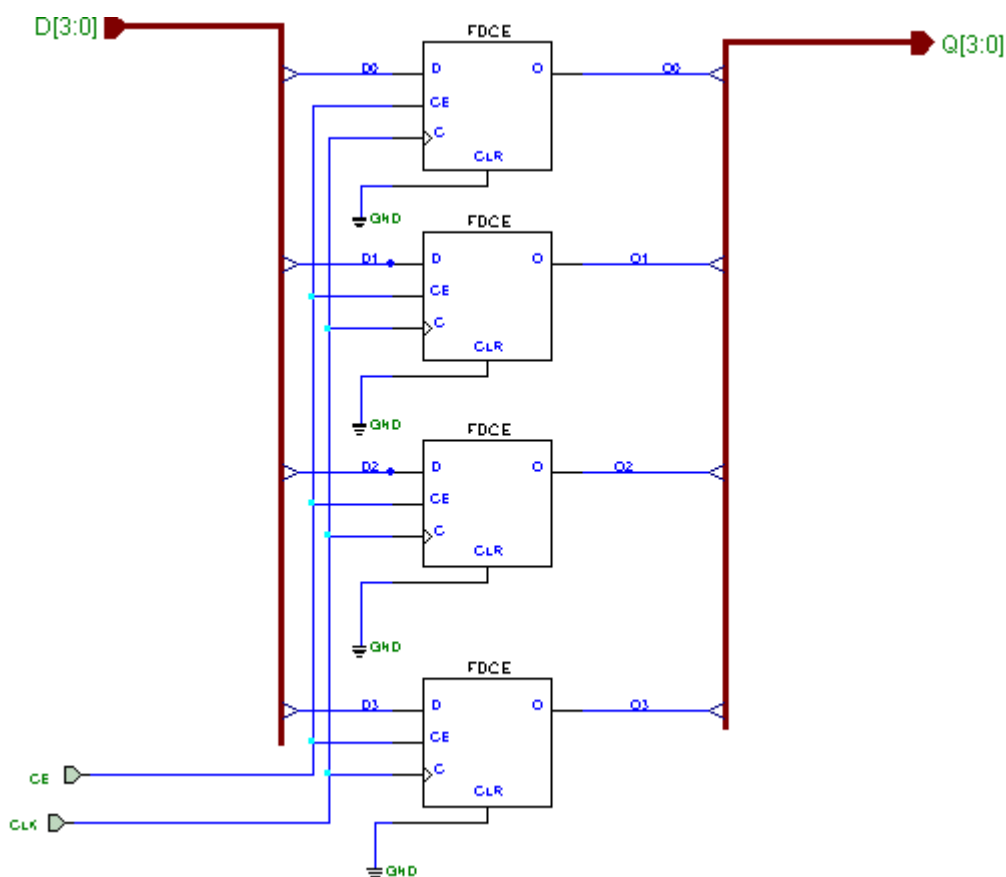
Załączanie i odłączanie zegara będzie się wtedy odbywało poprzez podanie na CE sygnałów – odpowiednio wysokiego i niskiego.



Układ z przerzutnikami posiadającymi wejście CE jest wygodniejszy i pozbawiony wad takich jak powstawanie hazardów czy zjawisko wyścigów. Jednak z takim układem związana jest dodatkowa logika (tzn. multiplexery) i przełączanie układu (układ ciągle reaguje na zegar), a zatem dodatkowy pobór mocy. W układach, w których zależy nam na jak najmniejszym poborze mocy (szczególnie wtedy kiedy bramkowany (CE) układ jest bardzo duży) często stosuje się układy oparte na zasadzie podanej z bramką AND czy bramką OR oraz dodatkowymi rozwiązaniami zapobiegającymi hazardom czy zjawisku wyścigów.

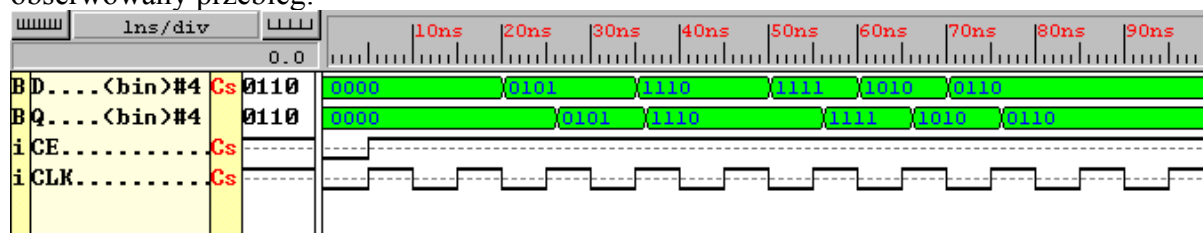
UKŁAD PIPO

Przykładem takiego rejestru (PIPO – wpisywanie i odczytywanie równoległe) może być połączenie równoległe przerzutników. Otrzymujemy wtedy rejestr równoległy z wpisywaniem synchronicznym 74175



Podając na wejścia magistrali D wymuszenia oraz załączając zegar możemy obserwować przeniesienia sygnałów na równoległe wyjścia z przerzutników (magistrala Q).

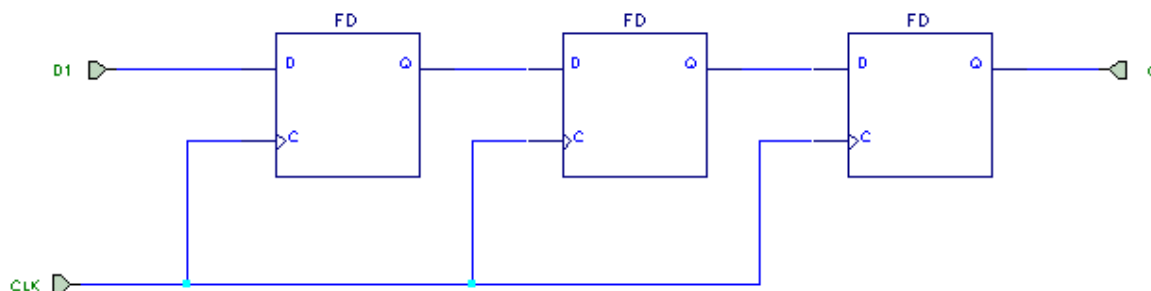
obserwowany przebieg:



Powyższy schemat przedstawia rejestr równoległy z wpisywaniem synchronicznym 74175. Wpisywaniem synchronicznym informacji do rejestru można sterować wejściem CE, które ustawia nam sygnał Clock Enable. Układ zbudowany na przerzutnikach typu D. Wpisanie informacji następuje na narastającym zboczu sygnału C, stan wejściowy na magistrali D musi być ustalony przed zmianą sygnału C (o czas ustawiania t_s) i może się zmienić dopiero po pewnym czasie od licząc od chwili sygnału zmiany C (czas utrzymywania t_h). Na wykresie powyższym widać też opóźnienia jakie wnoszą układy (czasy propagacji - opcja *set_mode unit* jest włączona).

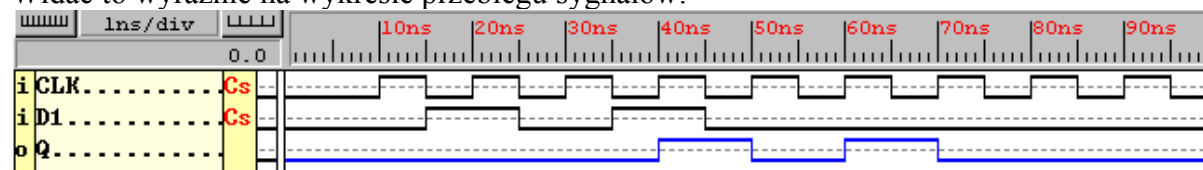
REJESTR SISO

Najprostszym przykładem realizacji rejestru typu SISO (szeregowe wejście i wyjście) jest rejestr zbudowany z przerzutników synchronicznych typu D. Jest to rejestr przesuwny. Jego realizację przedstawia poniższy schemat:



Wejście D1 jest wejściem pierwszego z przerzutników połączonych szeregowo, tak, że wejście następnego jest wyjściem poprzedniego. W ten sposób możemy połączyć dowolną ilość przerzutników, a opóźnienie sygnału na wyjściu będzie uzależnione od ich liczby. I tak – w naszym przypadku – będziemy mieli wyjście opóźnione o trzy takty względem wejścia.

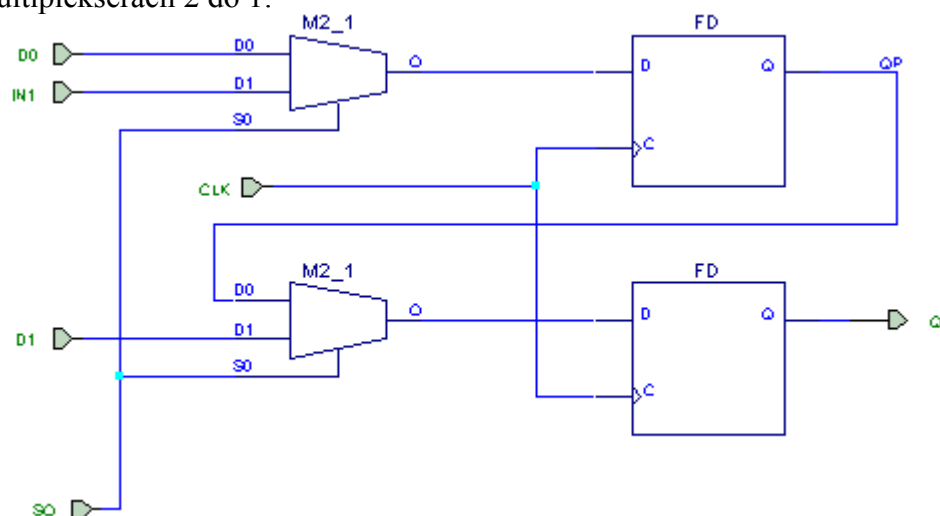
Widać to wyraźnie na wykresie przebiegu sygnałów:



REJESTR PISO

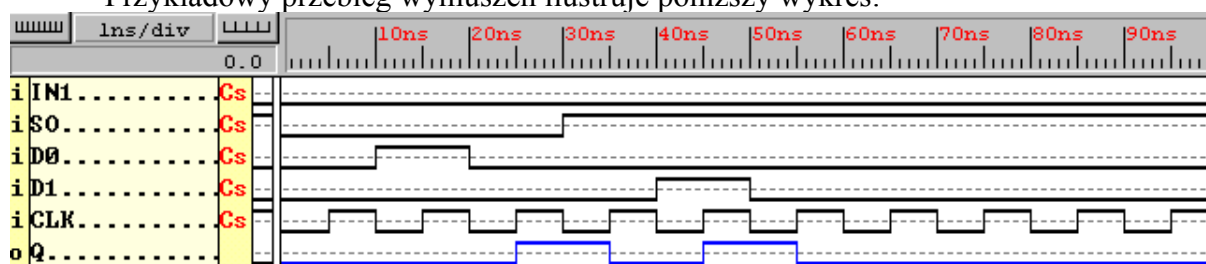
Rejestr również przesuwający, tym razem z równoległym wejściem i szeregowym wyjściem. Taki rejestr można również zaprojektować na podstawowych przerzutnikach typu D (lub JK).

Schemat najprostszego rejestru PISO na dwóch przerzutnikach typu D oraz dwóch multiplexerach 2 do 1:



Zastosowane tutaj multiplexery służą temu, aby oba wejścia (zarówno wyjście z przerzutnika D0 jak i wejście D1) mogły być przekazane na wejście drugiego przerzutnika.

Przykładowy przebieg wymuszeń ilustruje poniższy wykres:

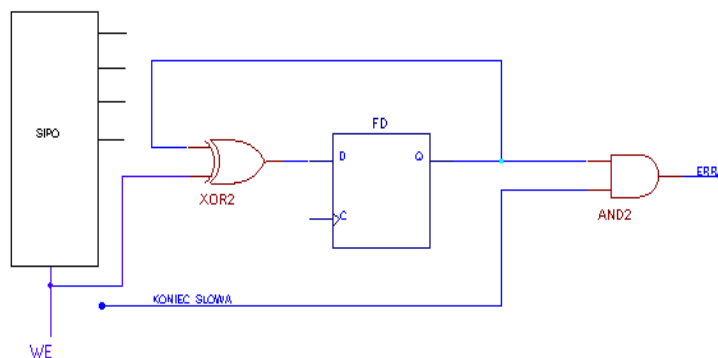


Widać, że stan z wejścia D0 na pierwszy przerzutnik jest w tym układzie opóźniony o 2 takty zegara, natomiast stan z wejścia D1 tylko o jeden takt. Przełączenie na wejściu sterującym S0 powoduje dostarczenie sygnału raz z wejścia D0 raz z D1.

Układ SIPO, w którym nie obserwuje się na wyjściu stanów przejściowych.

Na wyjściu należy umieścić rejestr typu równoległego (PIPO). Wejście zezwolenia zapisu powinno być sterowane za pomocą licznika modulo N gdzie N jest liczbą taktów jaką potrzebuje rejestr SIPO na wprowadzenie całego słowa. Przy takim rozwiązaniu na wyjściu całego układu stany przejściowe nie będą widoczne.

Kontrola bitu parzystości

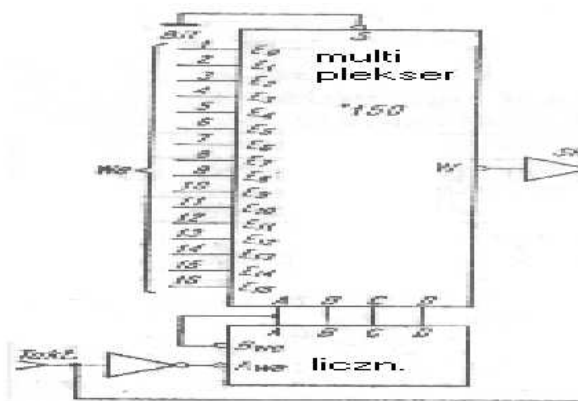


Na wyjściu rejestru szeregowo - równoległego należy umieścić powyższy układ. Ważne jest by kontrolować stan początkowy przerzutnika D. Wyjście Q będzie pokazywało czy dotąd zliczone bity był parzyste jeżeli taka wartość zostanie na wyjściu po zliczeniu ostatniego bitu słowa zostanie włączony sygnał error. Funkcję końca słowa bitowego można zapisać używając licznika (CC)

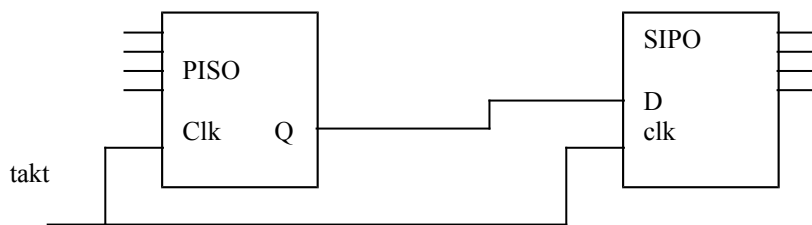
Przesyłanie danych.

Skutecznym sposobem przesyłania szeregowego danych (na małe odległości) jest użycie multiplexera i demultiplexera sterowanych dwoma identycznymi licznikami do których doprowadzamy identyczny sygnał zegarowy. W ten sposób dokonujemy najprostszej realizacji zwielokrotnienia z podziałem czasu. (Time Division Multiplexing).

Równoważnym rozwiązaniem tego problemu jest układ z użyciem rejestru SIPO,

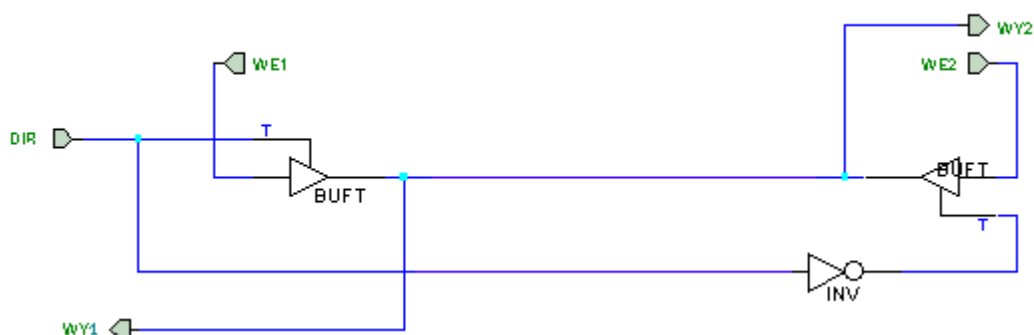


Zamiast używanego w powyższym przykładzie multiplexera oraz sterującego nim licznika. Natomiast zamiast demultiplexera i licznika można użyć rejestru PISO. Powoduje to w większości przypadków uproszczenie układu oraz zwiększenie maksymalnej częstotliwości układu.



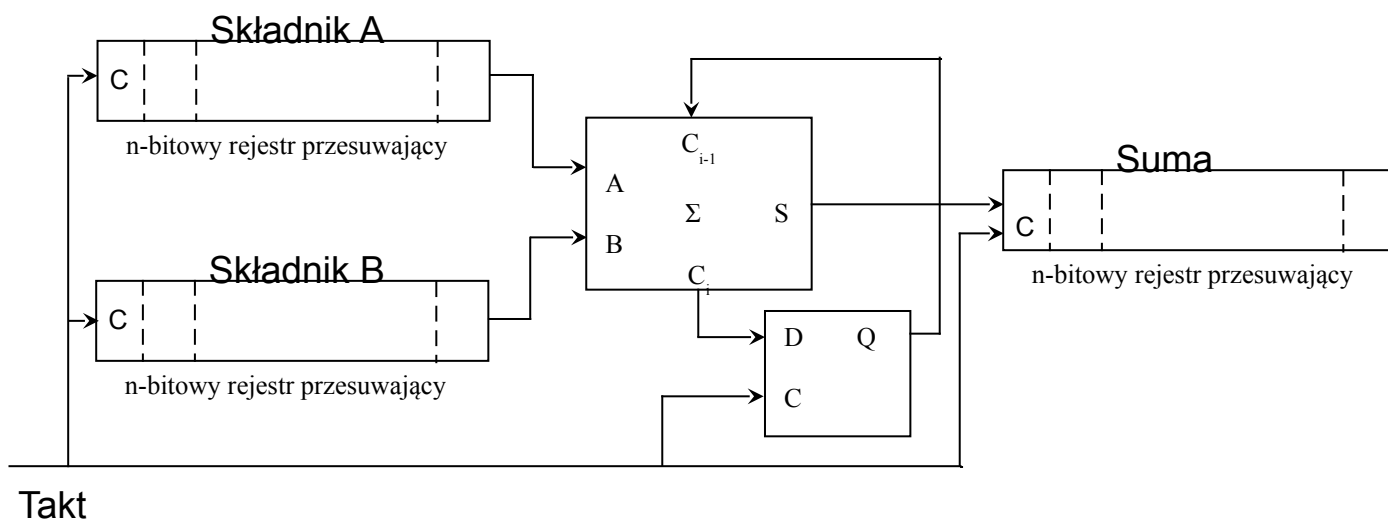
Przykład dwukierunkowej transmisji danych.

Zależnie od wartości wejścia dir dane są przekazywane z WE1 lub WE2 (gdy jedno z wejść przekazuje dane drugie jest w stanie wysokiej impedancji).



Układ wykonujący szeregowo operację dodawania dwóch liczb.

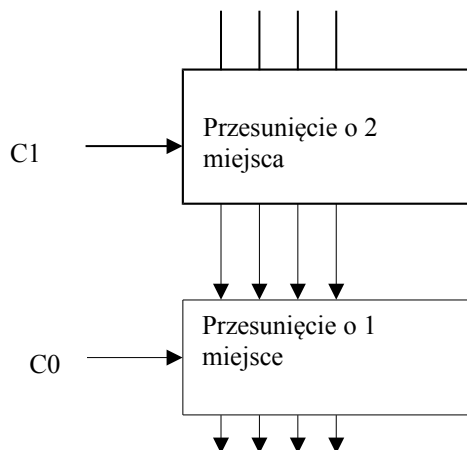
W sumatorze **szeregowym** (ang. *Serial Adder*) w każdym kroku dodawane są, poczynając od pozycji najmniej znaczącej, dwa bity składników oraz bit przeniesienia z poprzedniej pozycji. Przeniesienie z poprzedniej pozycji jest zapamiętywane, np. za pomocą przerzutnika D. Schemat ideowy takiego sumatora jest przedstawiony na rysunku poniżej. Zaletami sumatorów szeregowych są ich prostota i mała liczba układów potrzebnych do realizacji. Natomiast wadą jest mała szybkość działania – czas trwania dodawania składników nT (T - takt zegara).



Układy wielopoziomowe.

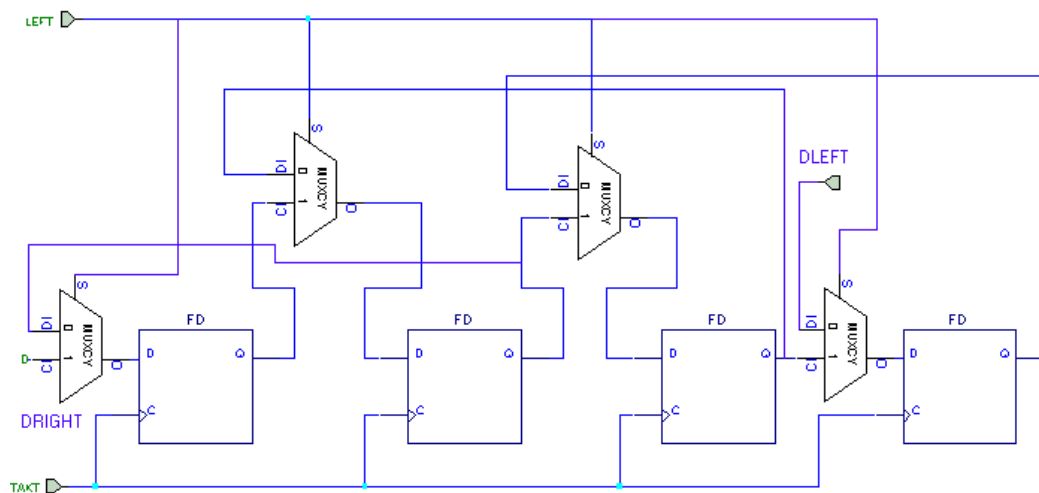
Najprostszą metodą implementacji n-bitowego rejestru przesuwającego, jest stworzenie go z $\log_2 n$ poziomów n-bitowych rejestrów przesuwających, w taki sposób by i-ty poziom rejestru przesuwającego przesunął o 2^i (lub o 0) miejsc; poziom ten jest kontrolowany przez i-ty bit sterowania (w podanym przykładzie wejście C).

Jako przykład takiego rozwiązania przedstawiamy czterobitowy wielopoziomowy rejestr przesuwający.



Wartość przesunięcia określa równanie $m = C_0 + C_1 * 2$ wzór ogólny to: $m = \sum C_n 2^n$

Układ rejestru przesuwnego w lewo i w prawo w zależności od sygnału kontrolnego „Left”.

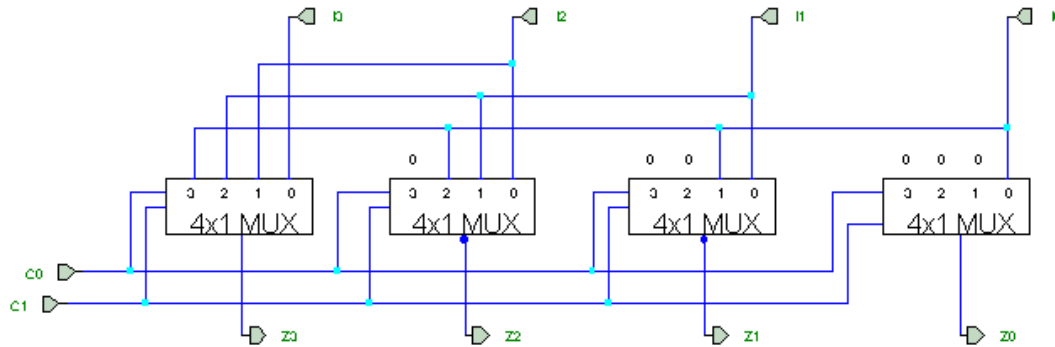


Powyższy rejestr rewersyjny-szeregowy umożliwia przesuwanie jego zawartości w lewo bądź w prawo zależnie od wartości na wejściu „Left”.

Układy szybkich rejestrów przesuwających (ang. barrel shifter).

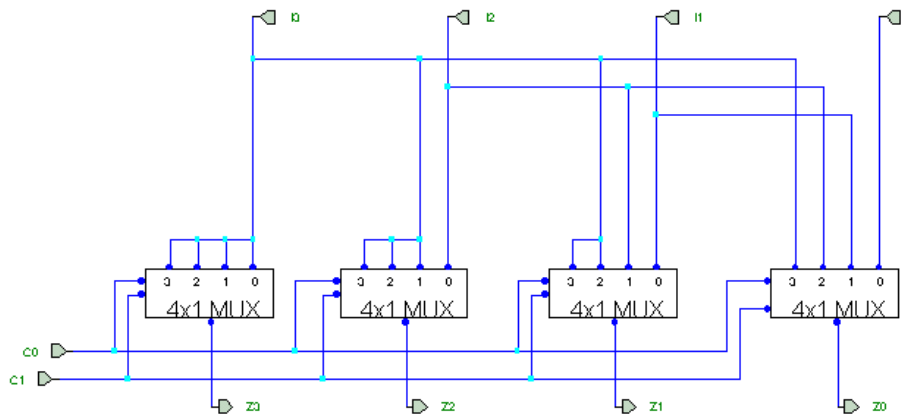
Są to układy oparte na multiplekserach, dzielimy je jedno oraz wielo poziomowe. Przykładem jednopozomowego rejestru przesuwającego jest układ:

Układ mnożący



Jest to 4 bitowy przesuwnik mogący przesunąć swoją zawartość o dowolną ilość miejsc z zakresu 0-3. Wielkość przesunięcia jest wpisywana kodem binarnym na wejścia C0,C1. Ponieważ przesunięcie bitów o N-pozycji w kierunku MSB jest mnożeniem o 2^N natomiast przesunięcie w kierunku przeciwnym dzieleniem o 2^N . Układ ten możemy nazwać rejestrem przesunym arytmetycznym.

Układ dzielący



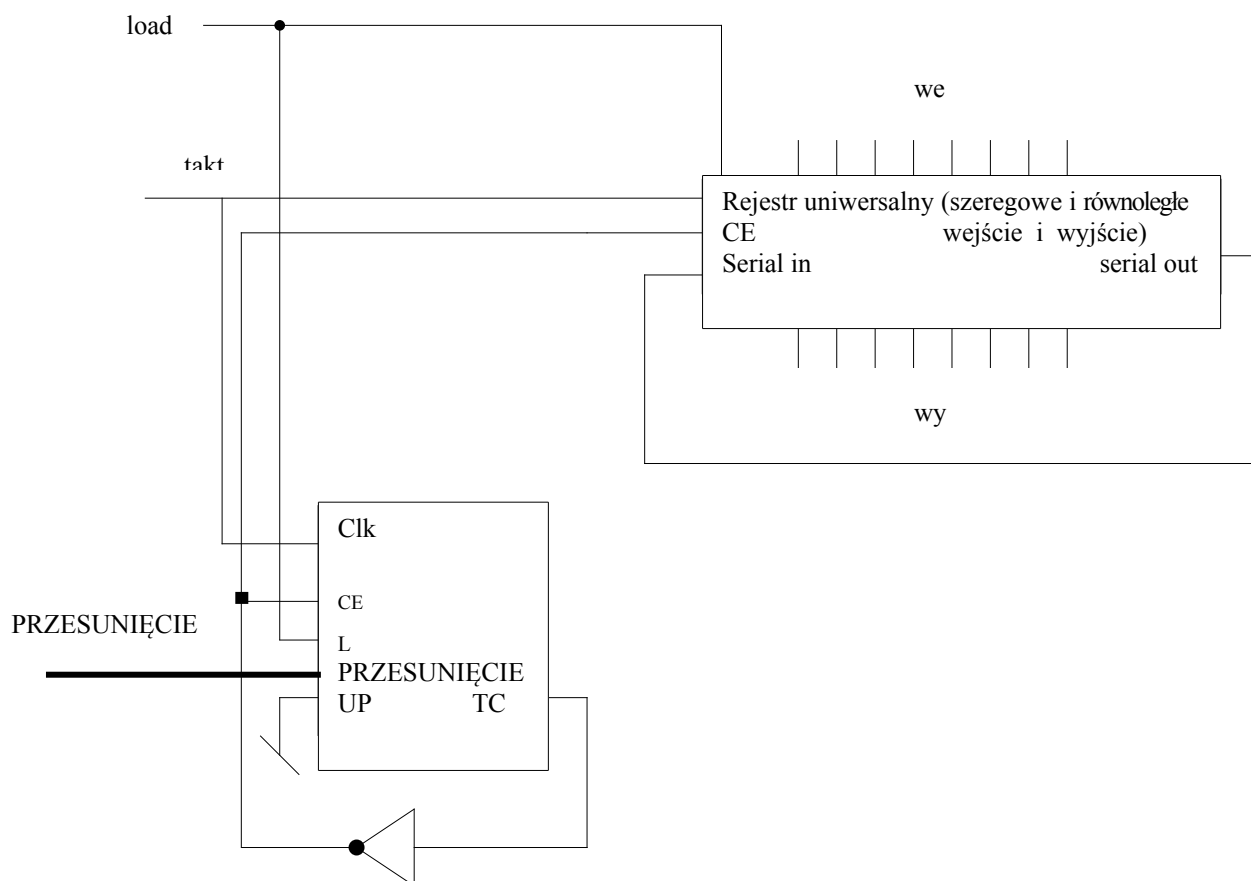
Powyższy układ może przesunąć swoją zawartość o dowolną ilość miejsc z zakresu 0-3. Przesuwa w kierunku LSB a więc dzieli o 2^N .

Z przedstawionego powyżej układu można utworzyć szybki rejestr przesuwny krążący.

Rejestr przesuwny krążący, sekwencyjny z wejściem Load

Jest to rejestr pozwalający na przesuwanie swojej zawartości o określoną ilość miejsc, tą ilość określamy za pomocą wejścia oznaczonego na schemacie jako „przesunięcie”, które jest podawane na wejście wpisu równoległego licznika zliczającego w dół. W przeciwieństwie do szybkich układów przesuwających, układ ten przesuwa tylko o jeden bit w jednym taktie zegara, przez co potrzebuje aż ‘przesunięcie’ taktów zegara w celu dokonania przesunięcia.

Połączenie wyjścia szeregowego z szeregowym wejściem pozwala na rotacje danych (o jedno miejsce w cyklu zegara). Przesunięcie następuje jedynie gdy na wyjściu TC licznika jest stan niski, czyli stan licznika jest większy od zera. Wpisanie danych wejściowych oraz wartości przesunięcia następuje w chwili podania stanu wysokiego na wejściu ‘load’.



Bibliografia:

- 0 Układy scalone TTL w systemach cyfrowych.
- 1 Układy scalone serii UCA64/UCY74
- 2 Cyfrowe układy MOS-LSI
- 3 Cyfrowe układy CMOS
- 4 Elementy i układy cyfrowe
- 5 Computer arithmetic systems.

Jan Pieńkos Janusz Turszyński
Włodzimierz Sasal
Pieńkos Piestryński
Piotr Gajewski Janusz Turczyński
J. Piecha
J.Omondi Marenia