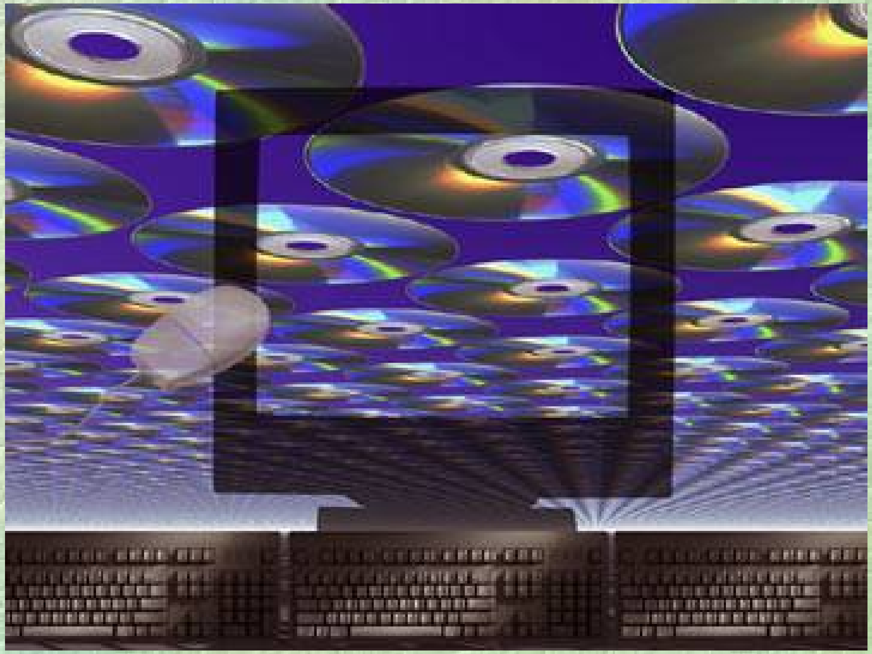
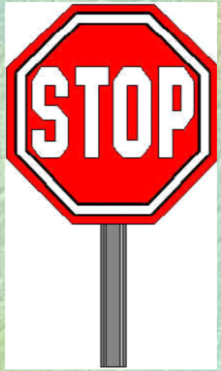
Systemy przetwarzania sygnałów

Filtry cyfrowe



wykład №4



Wykorzystano materiały książki Marven C., Ewers G.: „Zarys cyfrowego przetwarzania sygnałów” Warszawa WKŁ 1999, a więc mogą być stosowane oraz traktowane wyłącznie jako pomoc dydaktyczna.

Przez wiele ostatnich lat filtry cyfrowe były najbardziej rozpowszechnioną dziedziną stosowania cyfrowego przetwarzania sygnałów.

Jak już podkreślałem „ucyfrowienie” dowolnego projektu umożliwia odtworzenie każdego układu elektronicznego w dowolnym czasie i z identyczną dokładnością jego parametrów.

W odniesieniu do filtrów istnieją jeszcze dwie inne istotne zalety stosowania rozwiązań cyfrowych.

Po pierwsze możliwe jest przeprogramowanie urządzenia DSP i dzięki temu całkowita zmiana jego charakterystyki amplitudowej i fazowej.

Po drugie, możemy zmieniać współczynniki filtru podczas pracy urządzenia, tzn. możemy konstruować filtry „adaptacyjne".

Na przykład możemy zmienić konkretny program danego filtru i z filtru dolnoprzepustowego utworzyć filtr gómoprzepustowy bez jakiejkolwiek zmiany sprzętowej.

Dwie podstawowe postaci filtrów cyfrowych

filtry o Skończonej Odpowiedzi Impulsowej (SOI)

/**FIR** - finite impulse responce/

filtry o Nieskończo-nej Odpowiedzi Impulsowe (NOI)

/**IIR**- infinite impulse responce/

Najprostszy typ filtru – **filtr o skończonej odpowiedzi impulsowej**, określane wzórem:

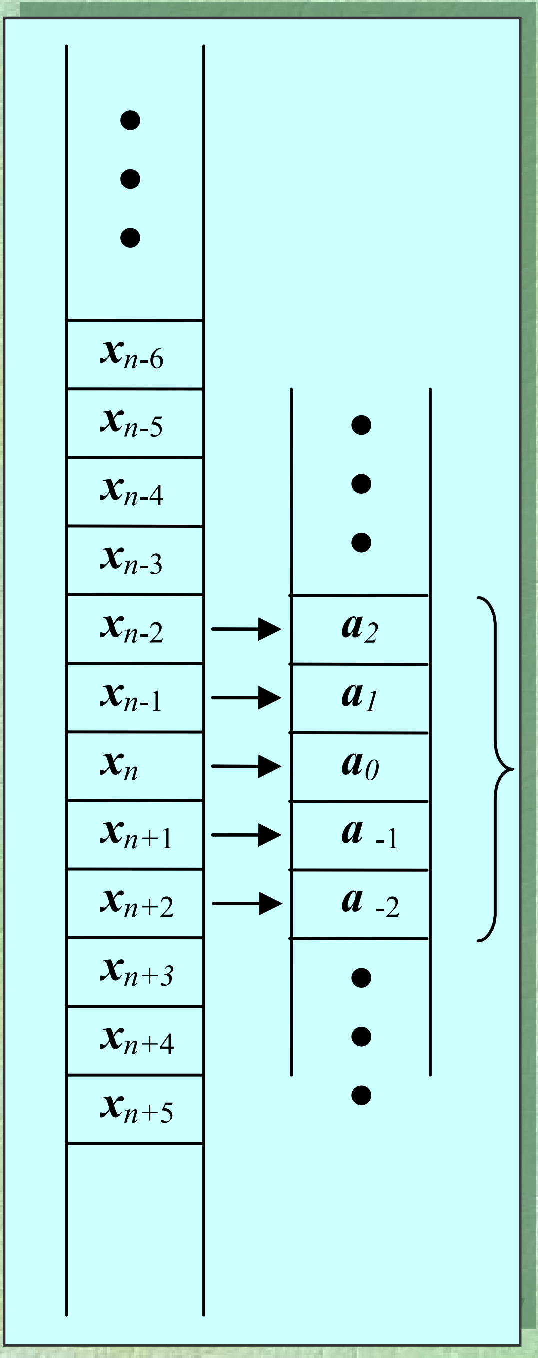
$$y_n = \sum_{-\infty}^{+\infty} a_k x_{n-k}$$

gdzie $\boldsymbol{a_k}$ – są współczynnikami filtru (wartości stałe), $\boldsymbol{x_{n-k}}$ – są danymi wejściowymi (próbki sygnału przed filtracją), $\boldsymbol{y_n}$ – dane wyjściowe (próbki sygnału po filtracji).

Wyobraźcie sobie dwie taśmy papierowe.

Na jednej taśmie znajdują się wartości danych x_{n-k}, zapisane jedno pod drugim.

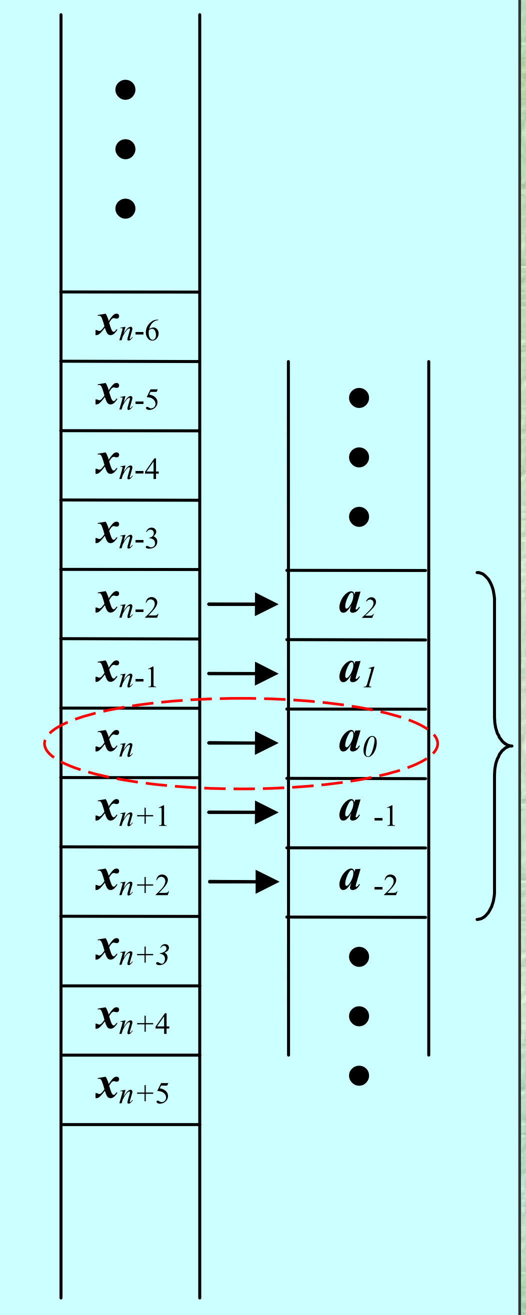
Na drugiej taśmie znajdują się współczynniki filtru a_k, które zapisane są w odwrotniej kolejności (od dołu w górę).

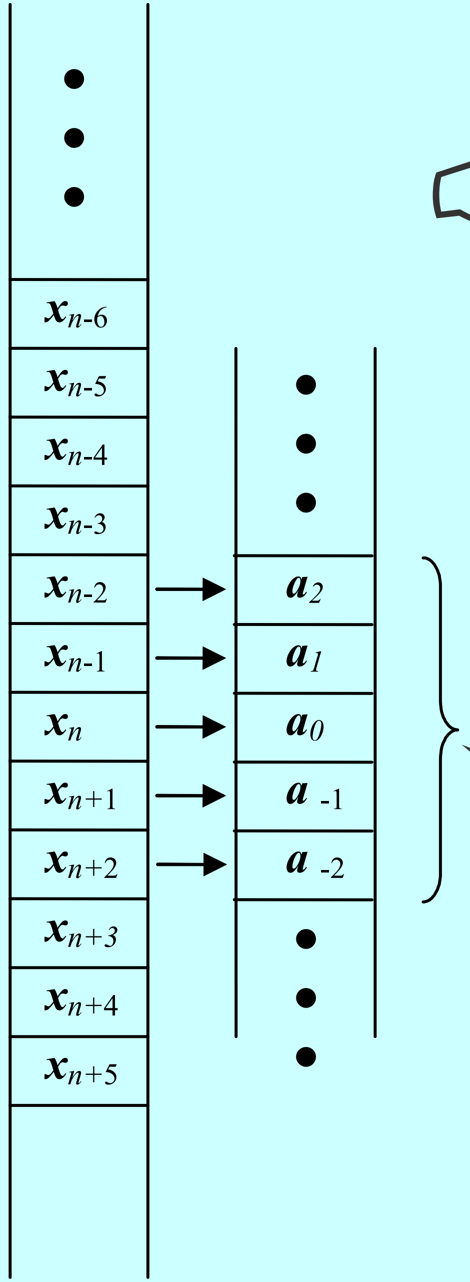


Wartość o zerowym indeksie znajduje się na przeciwko wartości o indeksie ***n***. Wynikiem jest suma iloczynów

$$\boldsymbol{a_k\, x_{n-k}}.$$

Po wyznaczeniu jednej wartości przesuńmy jedną z taśm (na przykład taśmę ze współczynnikami filtru) o jedną pozycję w dół i obliczmy nowy zestaw iloczynów, których suma daje nową wartość na wyjściu filtra - $\boldsymbol{y_{n+1}}$.





Procedura filtracji nierekursywnej polega na cyklicznem wykonaniu dwoch czynności: obliczenju sumy iłoczynów odpowiednich elementów obydwu taśm i przesuniencju jednej z taśm o jedną pozycję w dół.

Każda wartość wyjściowa jest wynikiem sumowania wszystkich iloczynów, wytwarzanych w wyniku przesuniencia taśm.

Opisany proces jest podstawą filtracji i nazywa się **splotem** sekwencji wejściowej ze współczynnikami filtru.

Obojętnie jest jaka z taśm będzie zapisana w kolejności odwrotniej – wynik będzie ten sam.

$$y_n = \sum_{-\infty}^{+\infty} a_k x_{n-k}$$

W praktyce liczba iloczynów z którymi mamy do czynienia powinna być skonczona. Zazwyczaj zakłada się, iż długość ciągu niezerowych współczynników $\boldsymbol{a_k}$ jest mniejsza od ciągu $\boldsymbol{x_{n-k}}$ danych wejściowych.

Czasem korzystnie jest rozpatrywać współczynniki $\boldsymbol{a_k}$ jako część ciągu nieskończonego, zawierającego wiele współczynników zerowych.

Zazwyczaj jednak najbardziej stosowne jest uznawać zbior $\{\boldsymbol{a_k}\}$ jako skończony i nie uwzględniać elementów zerowych znajdujących się po za jego zakresem.

Wówczas wzór przyjmuje postać:

$$\boxed{y_n = \sum_{-N}^{+N} a_k x_{n-k}}$$

Filtry **nierekursywne** do uzyskania bieżącej wartości elementu sekwencji danych wyjściowych wykorzystują elementy bieżące i elementy przeszłe sekwencji wejściowej nie uwzględniając zaś elementów sekwencji danych wyjściowych.

Jest to również powodem nazywania filtrów tego typu filtrami nierekursywnymi.

Dysponując skończonym ciągiem różnych od zera elementów sekwencji danych wejściowych filtr tego typy zawsze ma na wyjściu ciąg o skończonej długości niezerowych elementów sekwencji danych wyjściowych.

Stąd bierze się druga nazwa filtru tego typu – **filtr o skończonej odpowiedzi impulsowej (SOI)**.

Zatem jeżeli sekwencja wejściowa filtru **SOI** staje się nagle ciągiem wartości zerowych, to sekwencja wyjściowa też będzie ciągiem elementów zerowych.

Jeśli do wyznaczania wyjściowych wartości y_n wykorzystuje się nie tylko dane **wejściowe** ale równierz wartości **wyjściowe**, to otrzymamy następujący wzór:

$$y_n = \sum_{-\infty}^{+\infty} a_k x_{n-k} + \sum_{-\infty}^{+\infty} b_k y_{n-k}$$

gdzie $\boldsymbol{a_k}$, $\boldsymbol{b_k}$ – są wartości stałe.

W tym przypadku zdecydowano ograniczać zakres niezerowych współczynników **bieżącymi** i **przeszłymi** wartościami sekwencji danych wejściowych $\boldsymbol{x_{n-k}}$ oraz tylko **przeszłymi** wartościami sekwencji danych wyjściowych $\boldsymbol{y_n}$. Ponadto liczba wyznaczanych współczynników powinna być skończona.

Dła tego wzór przyjmuje postać:

$$y_n = \sum_0^N a_k x_{n-k} + \sum_1^M b_k y_{n-k}$$

Takie filtry nazywają się **rekursywnymi**.

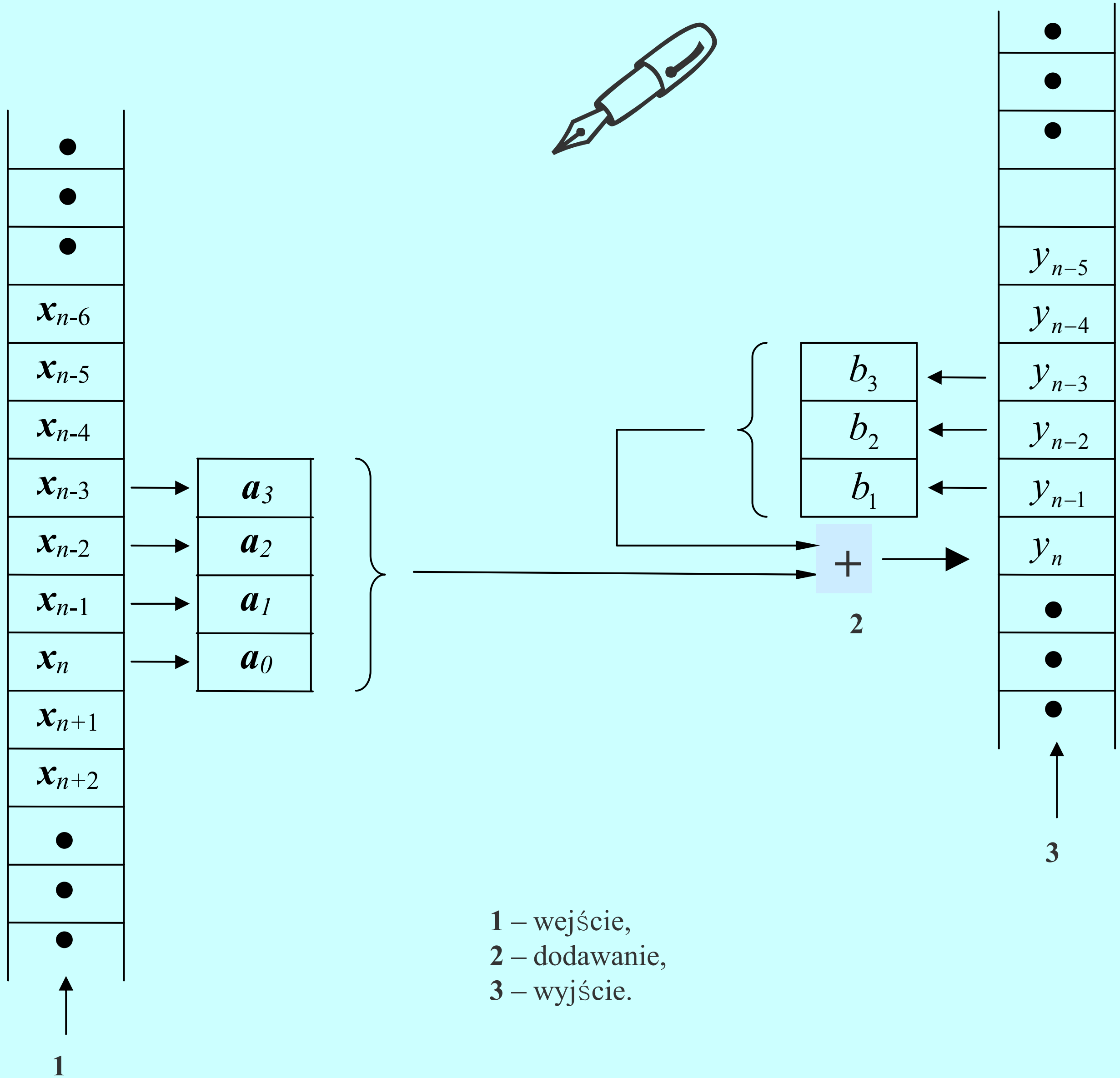
Filtry rekursywne są zasadniczo różne od od filtrów nierekursywnych ponieważ te filtry zawsze wymagają **„sprzężenia zwrotnego”**.

Jak już wspomniano, w odróżnieniu od filtrow **SOI** w filtrach tego typu każdy element sekwencji wyjściowej zależy od wartości bieżących oraz poprzednich elementów sekwencji wejściowej zarówno jak i od wartości poprzednich elementów sekwencji wyjściowej.

Efekt jest taki, że pewien skończony ciąg niezerowych wartości wejściowych może spowodować pojawienie się na wyjściu filtru nierekursywnego ciągu niezerowych wartości o nieskończonem czasie trwania.

Zatem, jeżeli sekwencja wejściowa filtru tego typu stałaby się nagle ciągiem wartości zerowych, to sekwencja wyjściowa mógłaby (i to w pewnych warunkach już na zawsze) pozostać się siągiem elementów niezerowych.

Stąd bierze się druga nazwa filtru tego typu – **filtr o nieskończonej odpowiedzi impulsowej (NOI)**.



1 – wejście,
2 – dodawanie,
3 – wyjście.

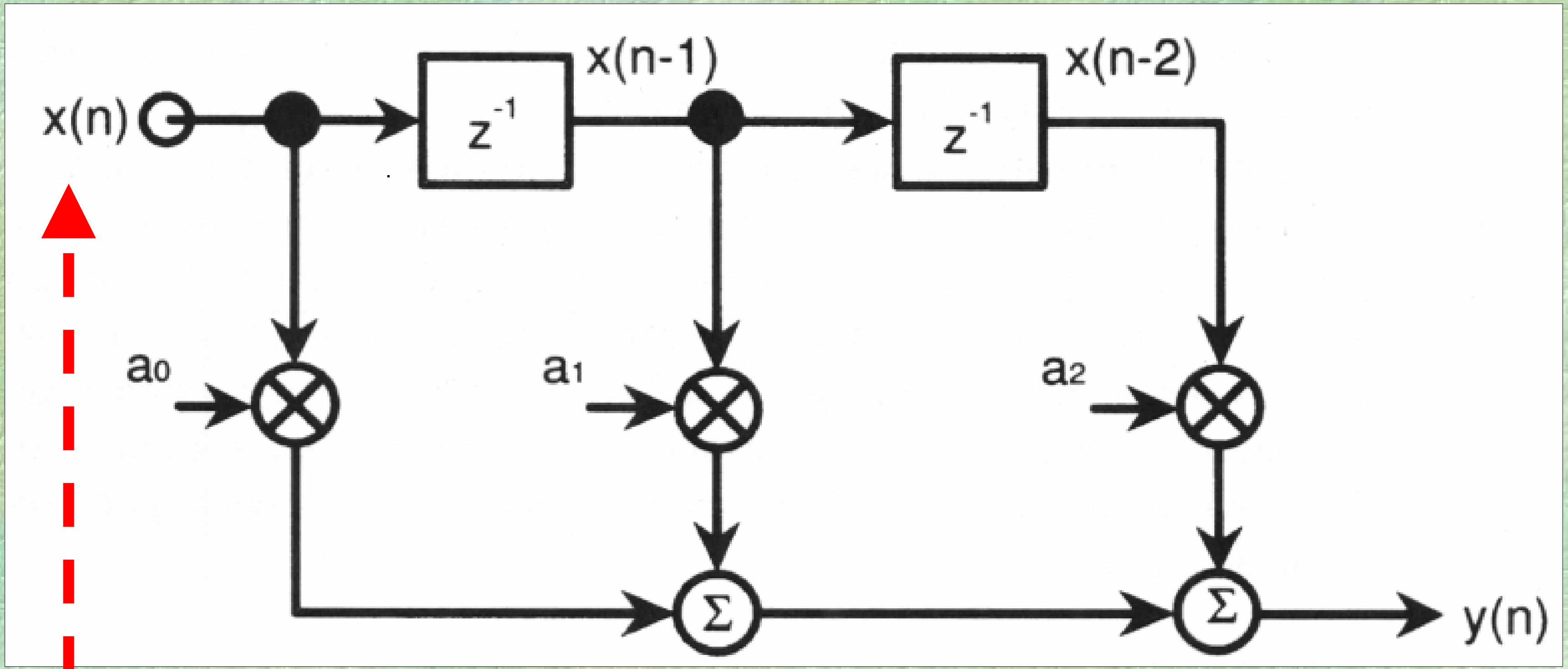
Zazwyczaj ***shematy blokowe*** filtrów **SOI** i **NOI** przedstawia się za pomocą struktur składających się z **bloków opóźnienia**, **układów mnożących** oraz **sumatorów** połączonych ze sobą w odpowiedni sposób.

Podczas zapoznawania się z omówieniami filtrów w rożnych książkach napotkacie tam ogromną liczbę funkcji matematycznych nazywanych **przekształceniem Z** i służących do opisu funkcjonowania filtrów cyfrowych.

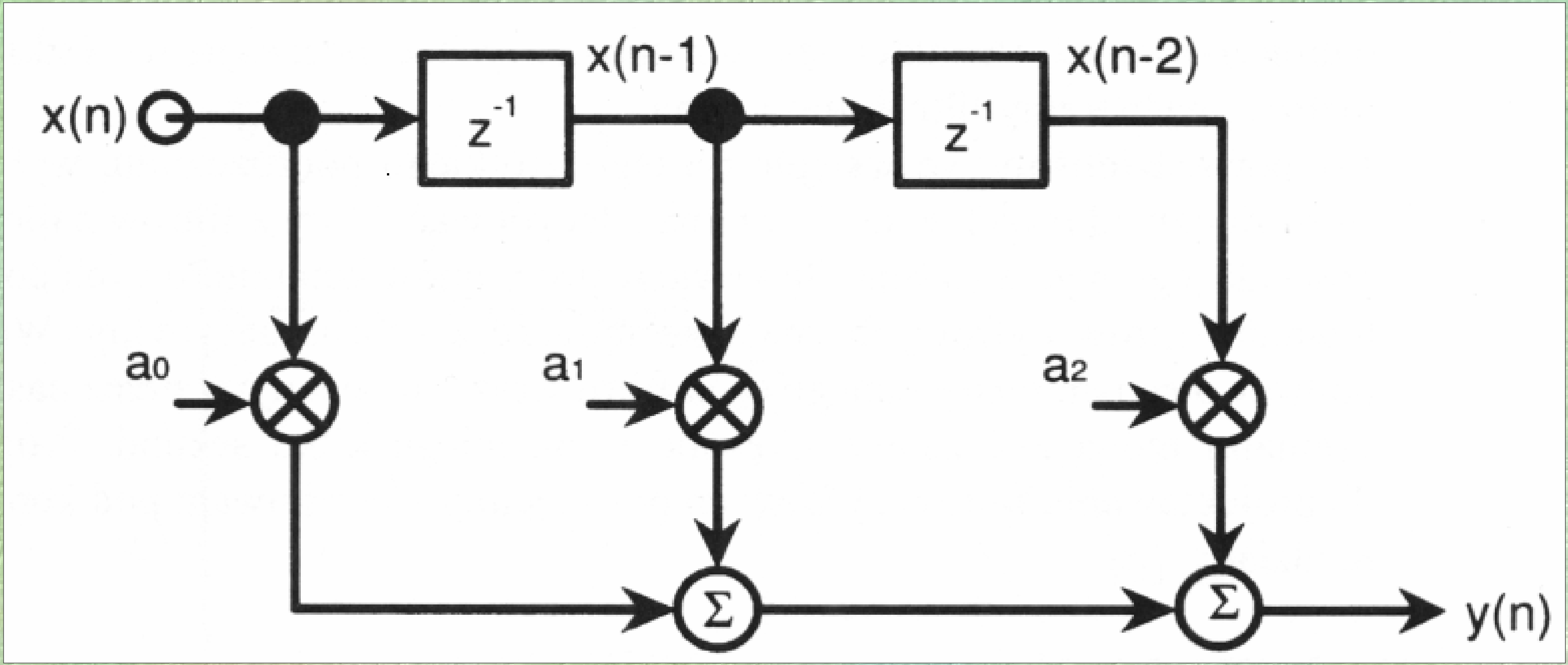
Dla naszych celów wystarczające będzie jedynie zrozumienie aplikacji wynikających ze stosowania przekształcenia Z i opisanie sposobów implementacji filtrów w urządzeniach DSP dnia dzisiejszego.

Filtry o Skończonej Odpowiedzi Impulsowej

Schemat blokowy filtru o skończonej odpowiedzi impulsowej (SOI)



Sygnał wejściowy ***x(n)*** jest ciągiem dyskretnych wartości otrzymanych przez próbkowanie przebiegu analogowego.



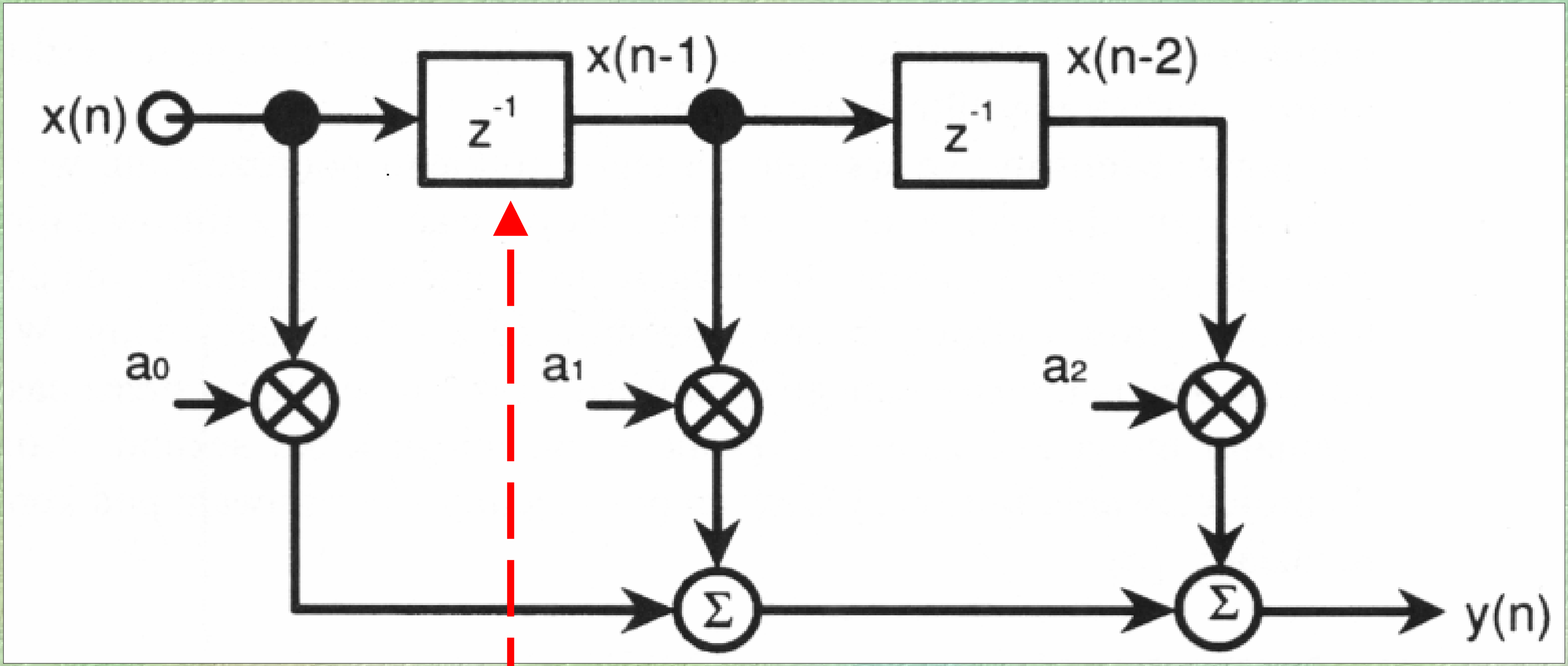
W ciągu tym

$x(0)$ odpowiada wartości wejściowej w momencie t=0;

$x(1)$ jest wartością w momencie $t = t_s$;

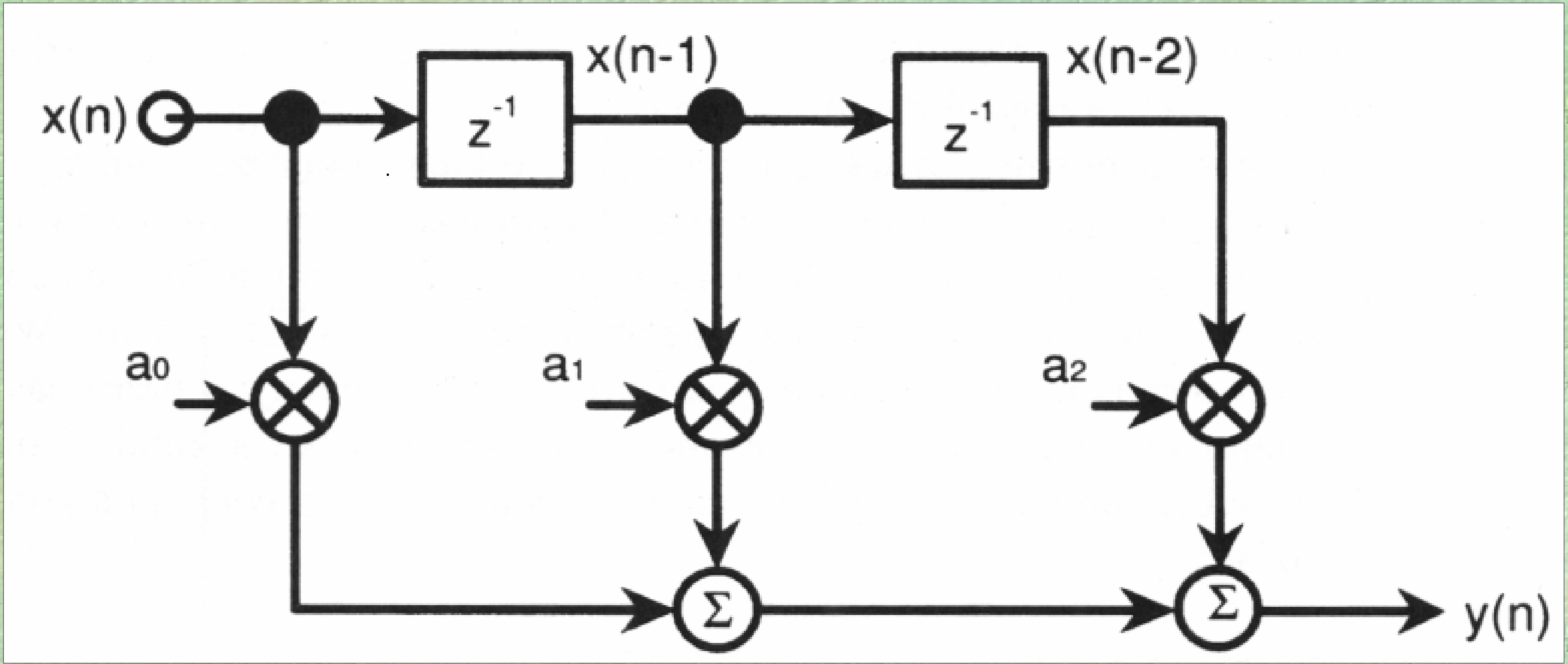
$x(2)$ -wartością w momencie t=2t_s, itp.

Wielkość $\boldsymbol{t_s}$ jest **okresem próbkowania** sygnału.



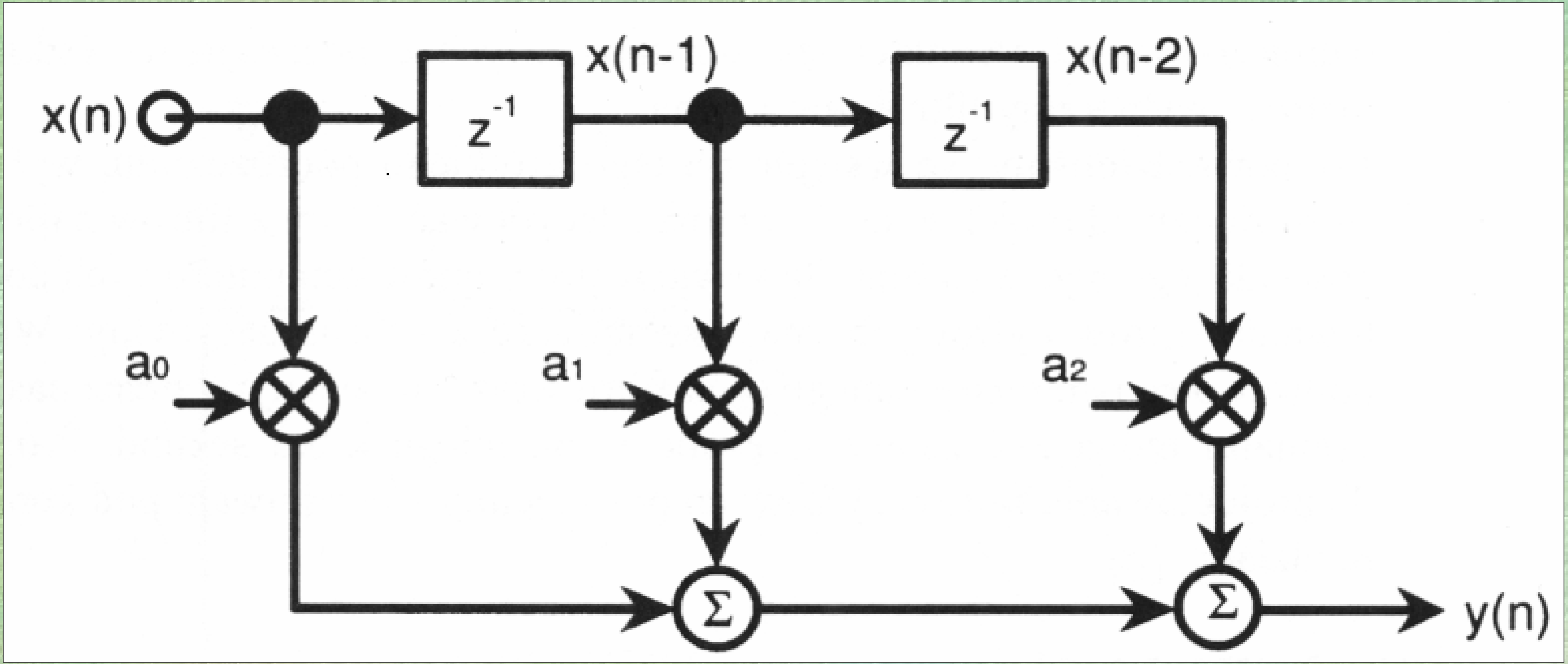
Wielkość *„z”* jest ***przekształceniem Z***, o którym mówiliśmy już wcześniej.

Dla naszych celów przyjmijmy, że kwadrat oznaczony na rysunku przez $\mathbf{z^{-1}}$ jest opóźnieniem czasowym sygnału o jeden okres próbkowania ($\boldsymbol{t_s}$), nazywany takie **jednostką opóźnienia**.



Wynika z tego, że wartość ***x*(*n-1*)** jest wartością ***x*(*n*)**, która wystąpiła o jeden okres próbkowania wcześniej, tzn. jest to poprzednie wejście.

Podobnie ***x(n-2)*** jest wartością wejściową wcześniejszą o dwa okresy próbkowania.



Z przykładu wynika, że sygnał wyjsciowy ***y(n)*** jest zawsze kombinacja składowych co najmniej **trzech** próbek wejściowych.

Każda próbka mnożona jest przez odpowiedni współczynnik a_R, co daje:

$$y(n) = a_0x(n) + a_1x(n-1) + a_2x(n-2)$$

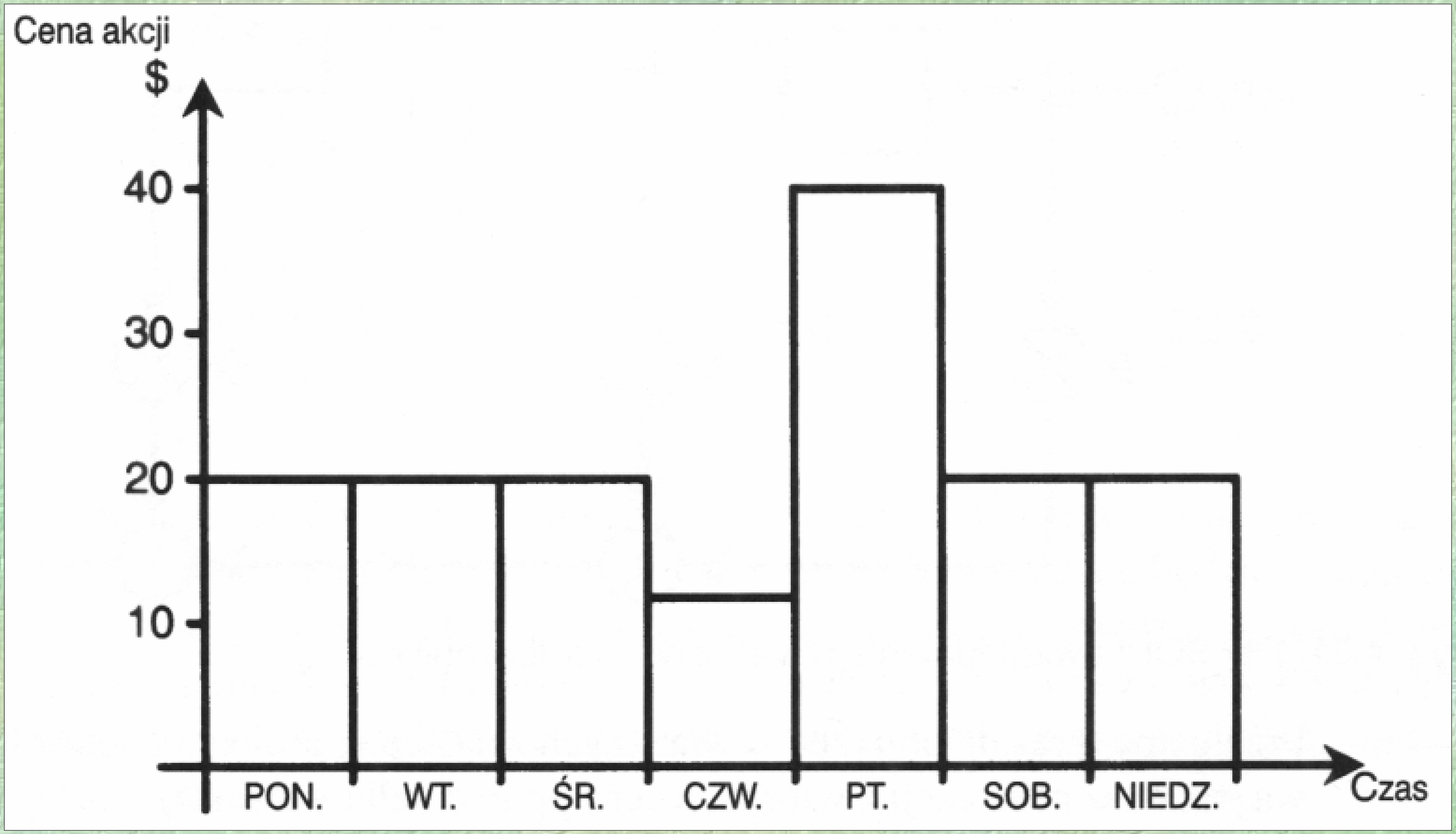
W celu zilustrowania pracy tego filtru przyjrzyjmy się konkretnemu przykładowi.

ceny akcji giełdowych za okres jednego tygodnia

Dzień	**Okres**	***x(n)***	**Cena($)**
Poniedziałek	0	$x(0)$	20
Wtorek	1	$x(1)$	20
Środa	2	$x(2)$	20
Czwartek	3	$x(3)$	12
Piątek	4	$x(4)$	40
Sobota	5	$x(5)$	20
Niedziela	6	$x(6)$	20

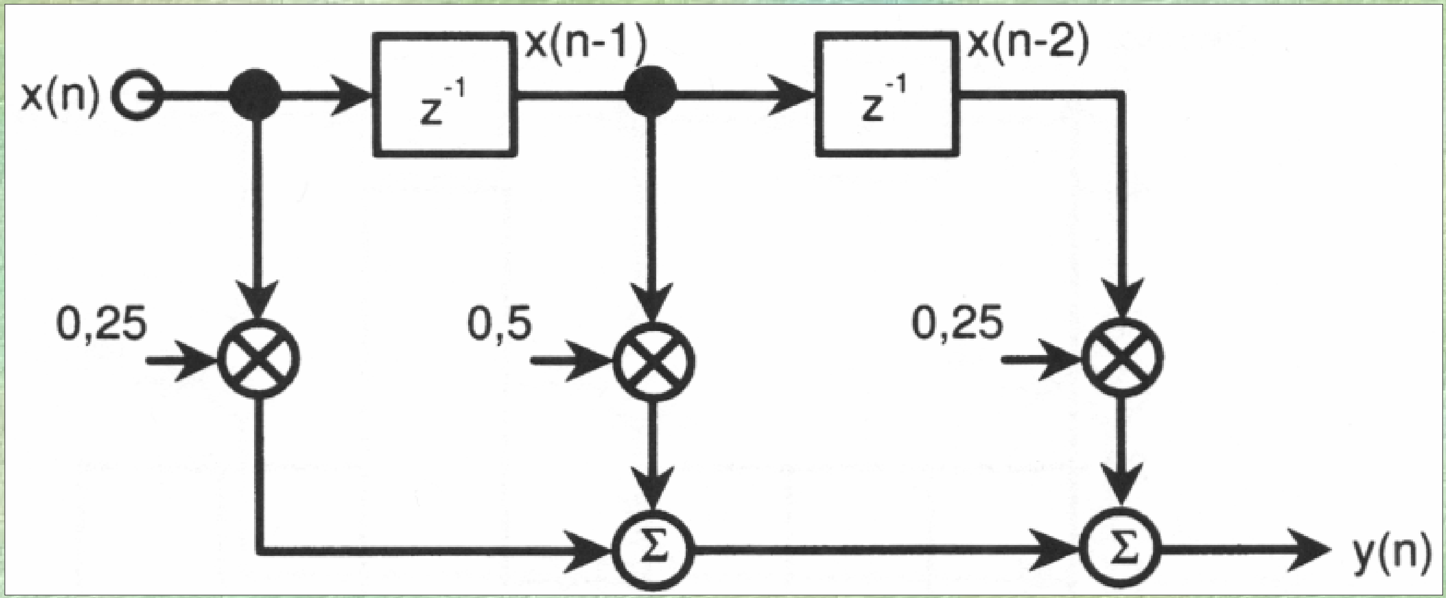
Użyjemy wartości tych cen w charakterze próbek wejściowych, tzn. ***x(n)***.

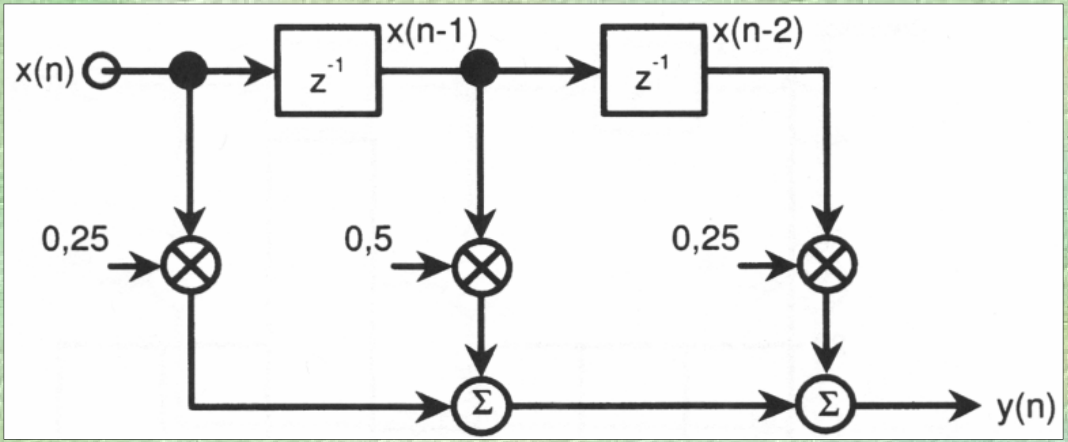
Na podstawie podanych danych możemy sporządzić histogram:



Zastosujmy teraz filtr z następującymi współczynnikami:

a_R	Wartośc
a_0	0,25
a_1	0,50
a_2	0,25





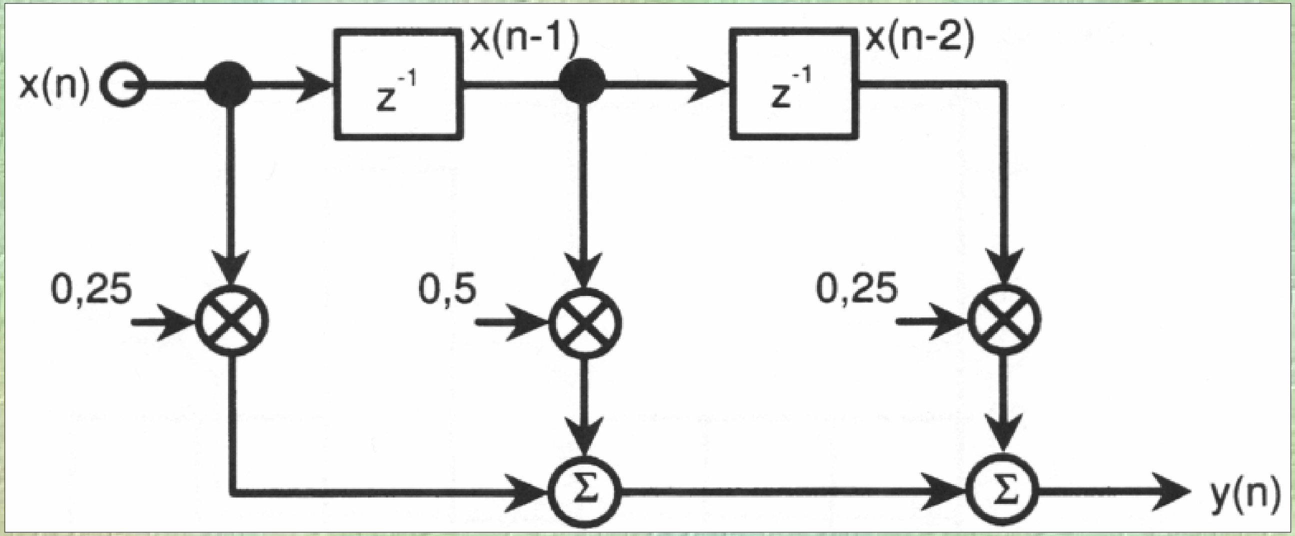
Jeśli ***n* = 0**, wówczas pierwsza wartość ***x(n)***, tzn. ***x(0)*** zostaje podana na wejście.

Przy założeniu, że w dwóch poprzednich dniach ceny akcj i nie były notowane, próbki ***x(n)***, ***x(n -1)*** i ***x(n -2)*** przybierają przy ***n* = 0** następujące wartości:

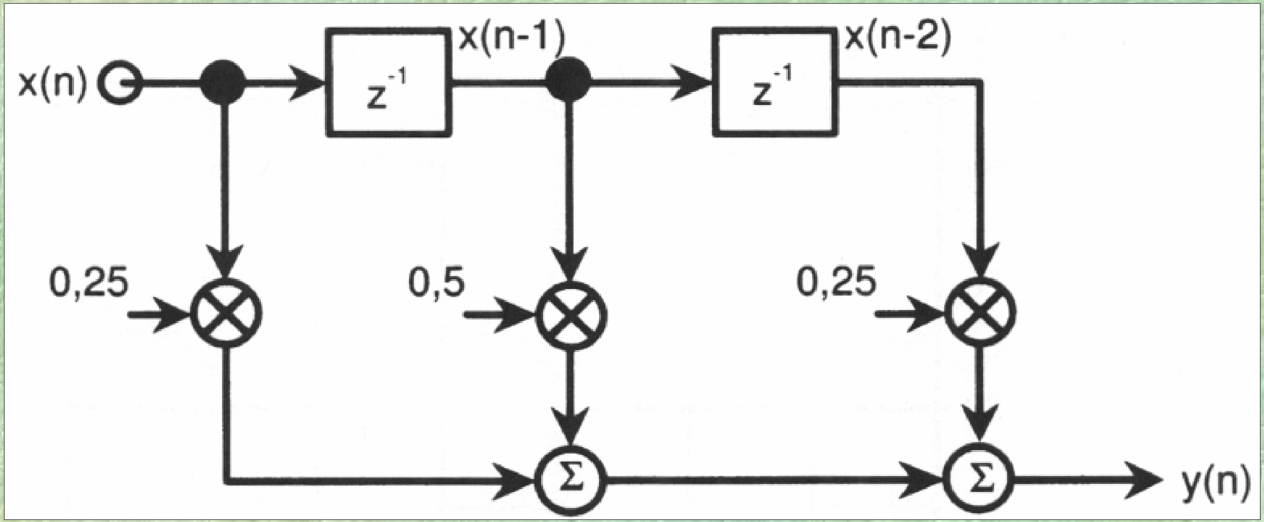
x(0)	**20**
x(-1)	**0**
x(-2)	**0**

$$y(0) = 0{,}25 \times 20 + 0{,}5 \times 0 + 0{,}25 \times 0 = 5$$

Przejdźmy teraz do obliczeń zawiązanych z następną próbką.



Wartością wejściową jest teraz cena akcji z wtorku, a cena poniedziałkowa zostaje zastąpiona przez ***x(n-1)***, tzn. *cena poniedziałkowa stała się wartością* ***przedawnioną*** *o jeden okres próbkowania*.



W tym przykładzie okres próbkowania ***t*** wynosi jeden dzień, więc ***n*** = 1, a wartości ***x(n)***, ***x(n-1)*** i ***x(n-2)*** są następujące:

x(0)	**20**
x(-1)	**20**
x(-2)	**0**

$y(1) = 0{,}25 \times 20 + 0{,}5 \times 20 + 0{,}25 \times 0 = 15$

Dla środy, ***n*** **= 2**, mamy:

$x(0)$	**20**
$x(-1)$	**20**
$x(-2)$	**20**

$$y(2) = 0{,}25 \times 20 + 0{,}5 \times 20 + 0{,}25 \times 20 = 20$$

Dla czwartku, ***n*** **= 3**:

$x(0)$	**12**
$x(-1)$	**20**
$x(-2)$	**20**

Zauważmy, że poniedziałkowe ceny akcji znikły z równania !

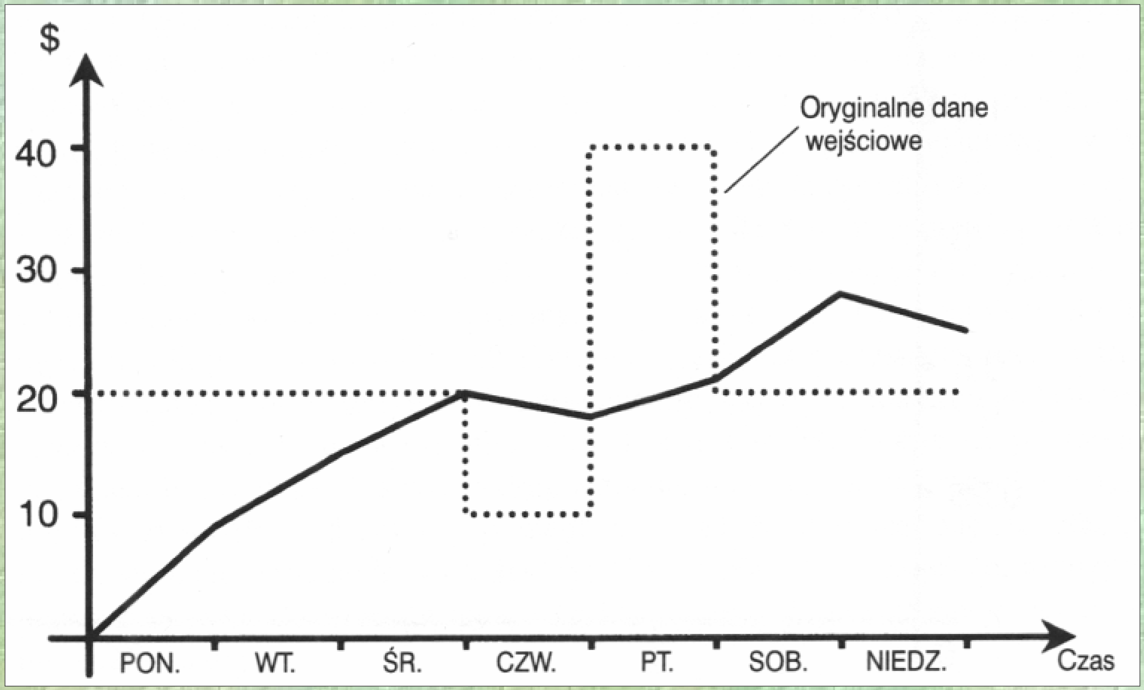
$$y(2) = 0{,}25 \times 12 + 0{,}5 \times 20 + 0{,}25 \times 20 = 18$$

Powtarzając te obliczenia dla pozostałych dni tygodnia, otrzymamy wartości ***y(n)***, które przedstawiamy poniżej:

Dzień	***y(n)***
Poniedziałek	5
Wtorek	15
Śrioda	20
Czwartek	18
Piątek	21
Sobota	28
Niedziela	25

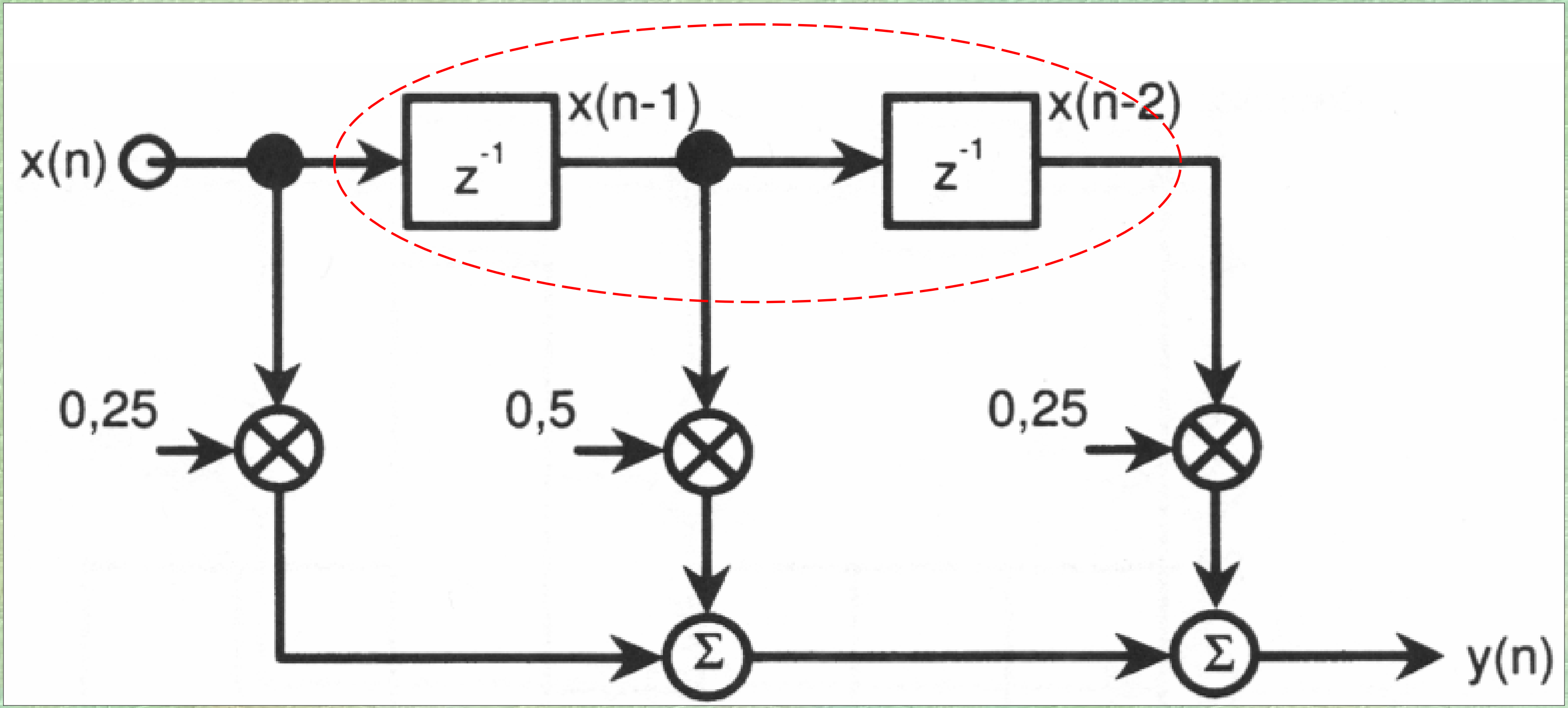
Teraz możemy otrzymane wartości wyjściowe przedstawić w postaci graficznej, co pokazano na rysunku.

Założyliśmy że wyjście z układu będzie podane na przetwornik c/a, a następnie na filtr rekonstrukcyjny, a dyskretne punkty połączyliśmy liniami prostymi.

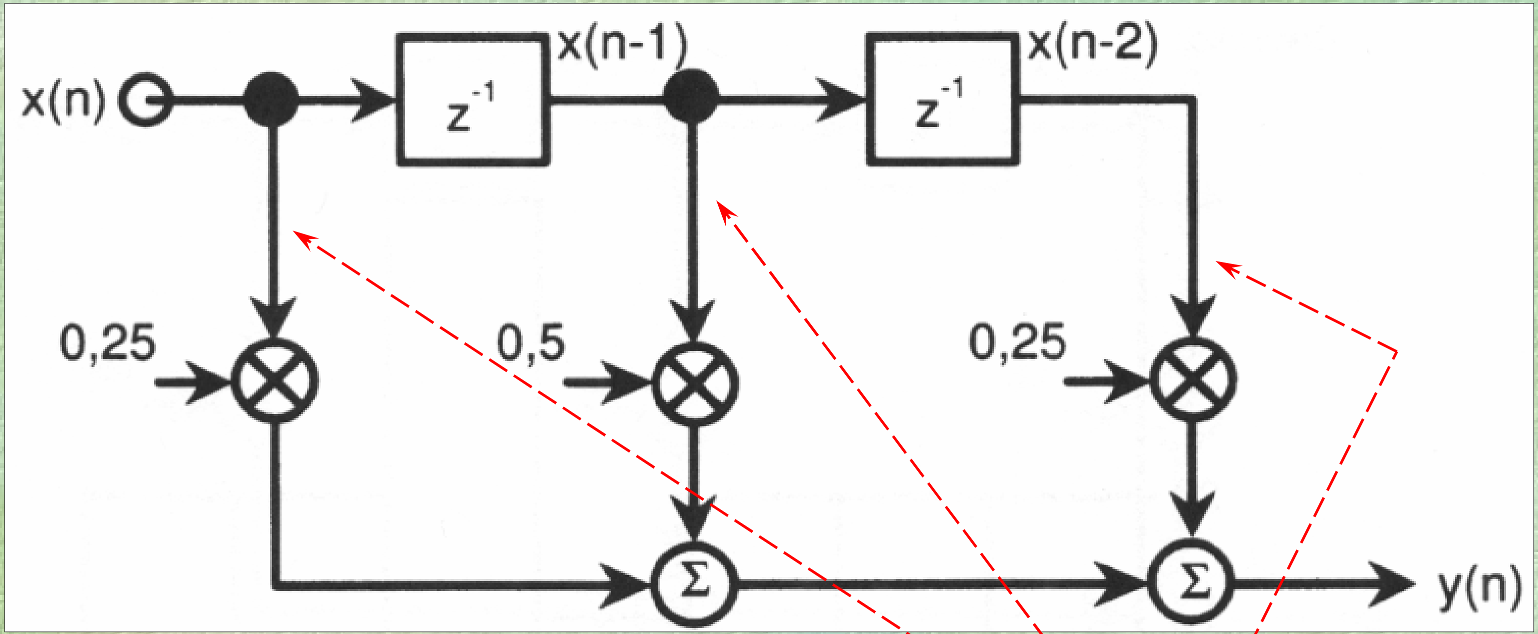


Widzimy, że filtr dokonuje obliczenia ***kroczącej średniej***.

W powszechnym użyciu jest kilka **terminów charakterystycznych** dla filtrów cyfrowych.



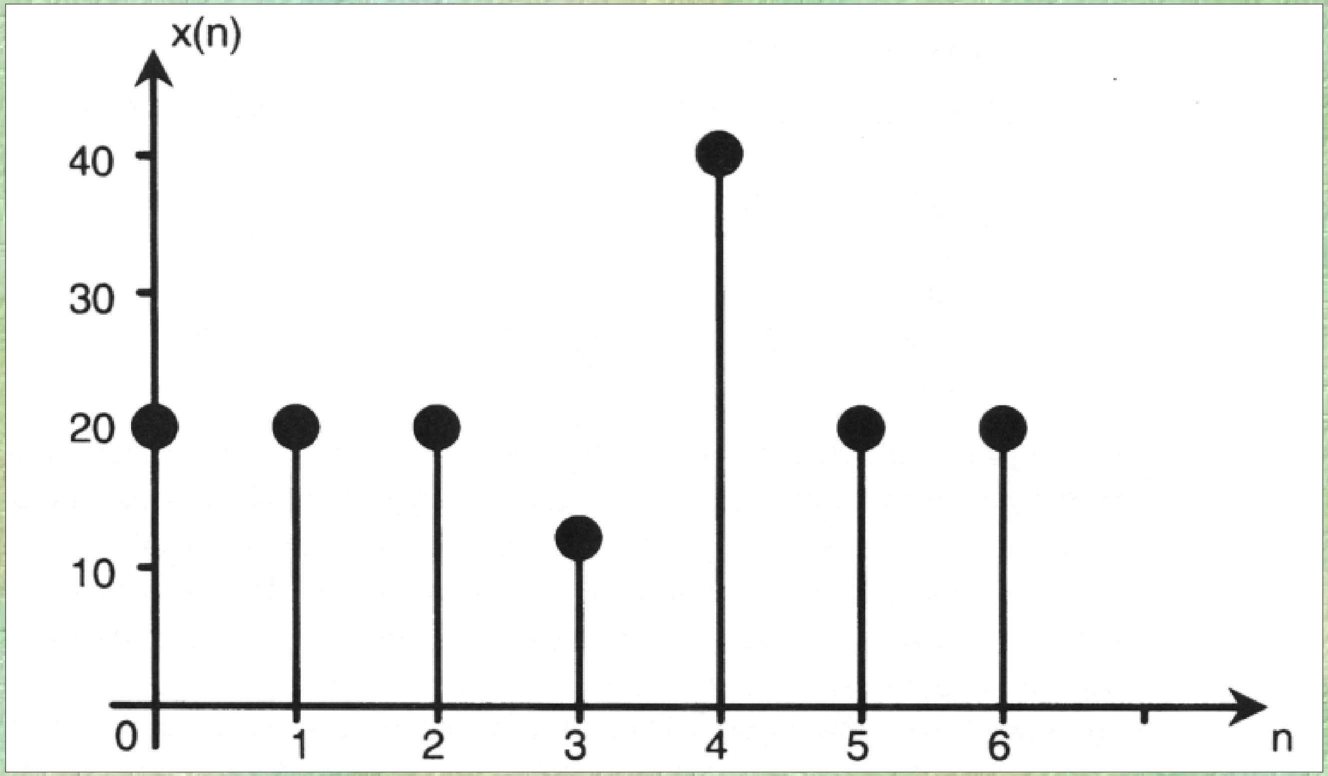
Ciąg funkcji **z^{-1}** nazywany jest **łańcuchem opóźniającym**, podczas gdy pojedynczy obwód **z^{-1}** to **opóźnienie** o jeden okres próbkowania.



Odgałęzienia wychodzące z łańcucha opóźniającego nazywane są **odprowadzeniami**, gdyż są punktami z których odprowadzana jest określona cześć sygnału.

Tak więc nasz układ teraz może być opisany jako filtr cyfrowy o trzech odprowadzeniach.

Na poprzedniem wykladzie poświęciliśmy nieco uwagi ważonej funkcji impulsowej.



Sygnał wejściowy ***x(n)*** filtru cyfrowego może być przedstawiony jako ciąg owych impulsów ważonych.

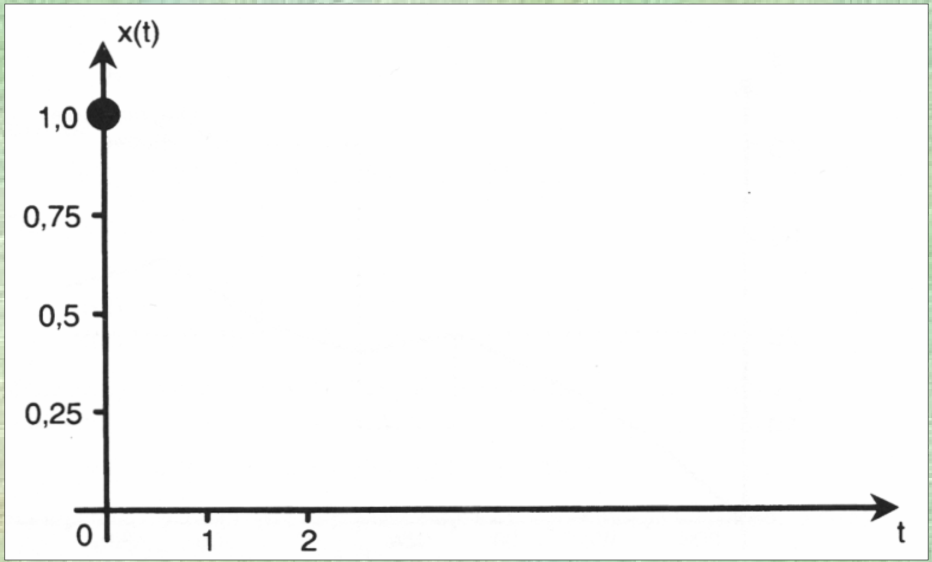
Wartość wejściową dla poniedziałku możemy więc zapisać jak:

$$\int_{-\infty}^{+\infty} 20\delta(t) = 20$$

Dzień	Cena($)
Poniedziałek	20
Wtorek	20
Śrioda	20
Czwartek	12
Piątek	40
Sobota	20
Niedziela	20

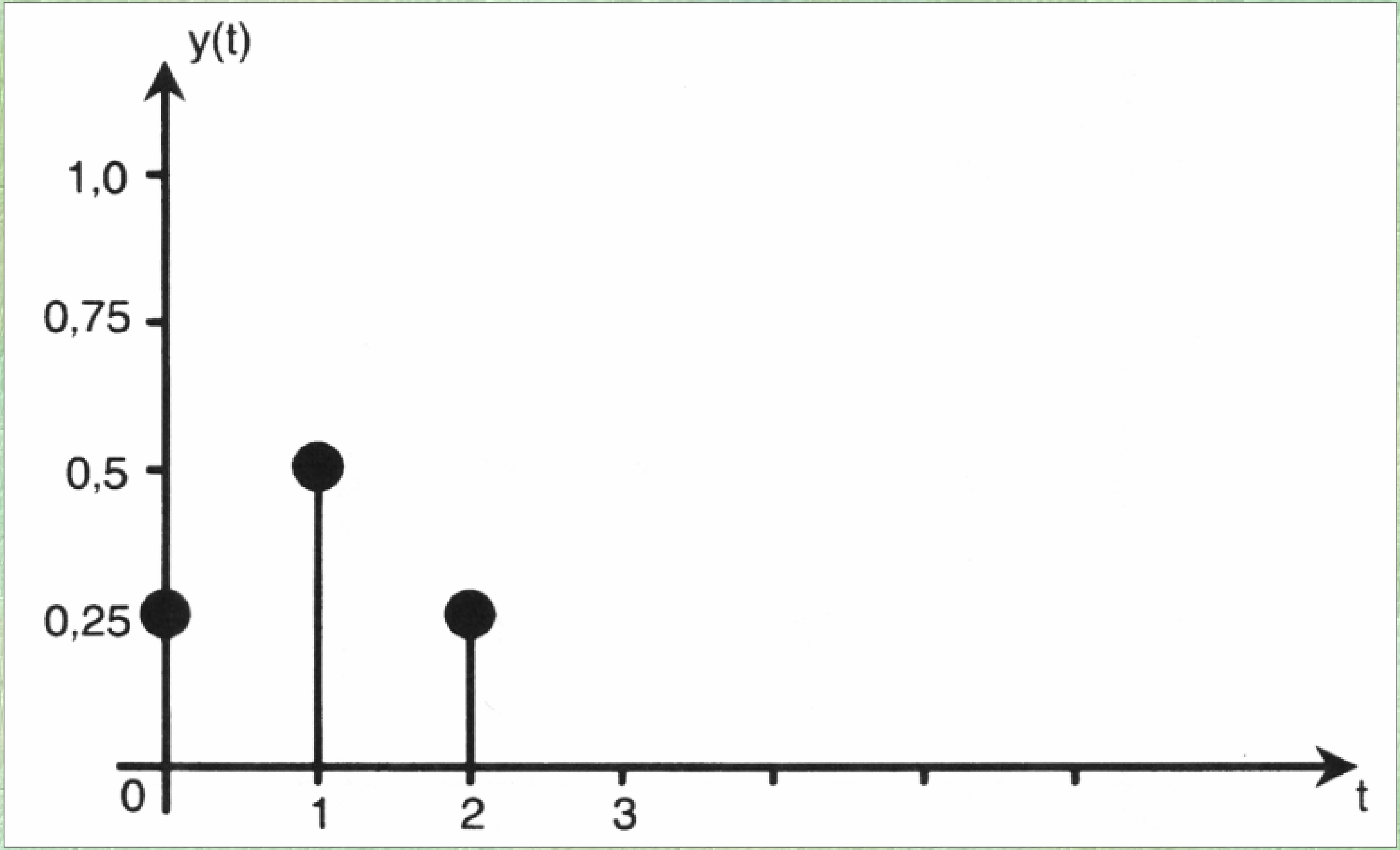
Wszystkie pozostałe wartości wejściowe mogą być zapisane w podobny sposób.

Ważonej funkcji impulsowej możemy także użyć do opisu samego filtru cyfrowego.



Odpowiedź impulsową filtru możemy zdefiniować jako przebieg wyjściowy filtru otrzymany w wyniku podania na jego wejście w chwili $i=0$ pojedynczej jednostkowej funkcji impulsowej.

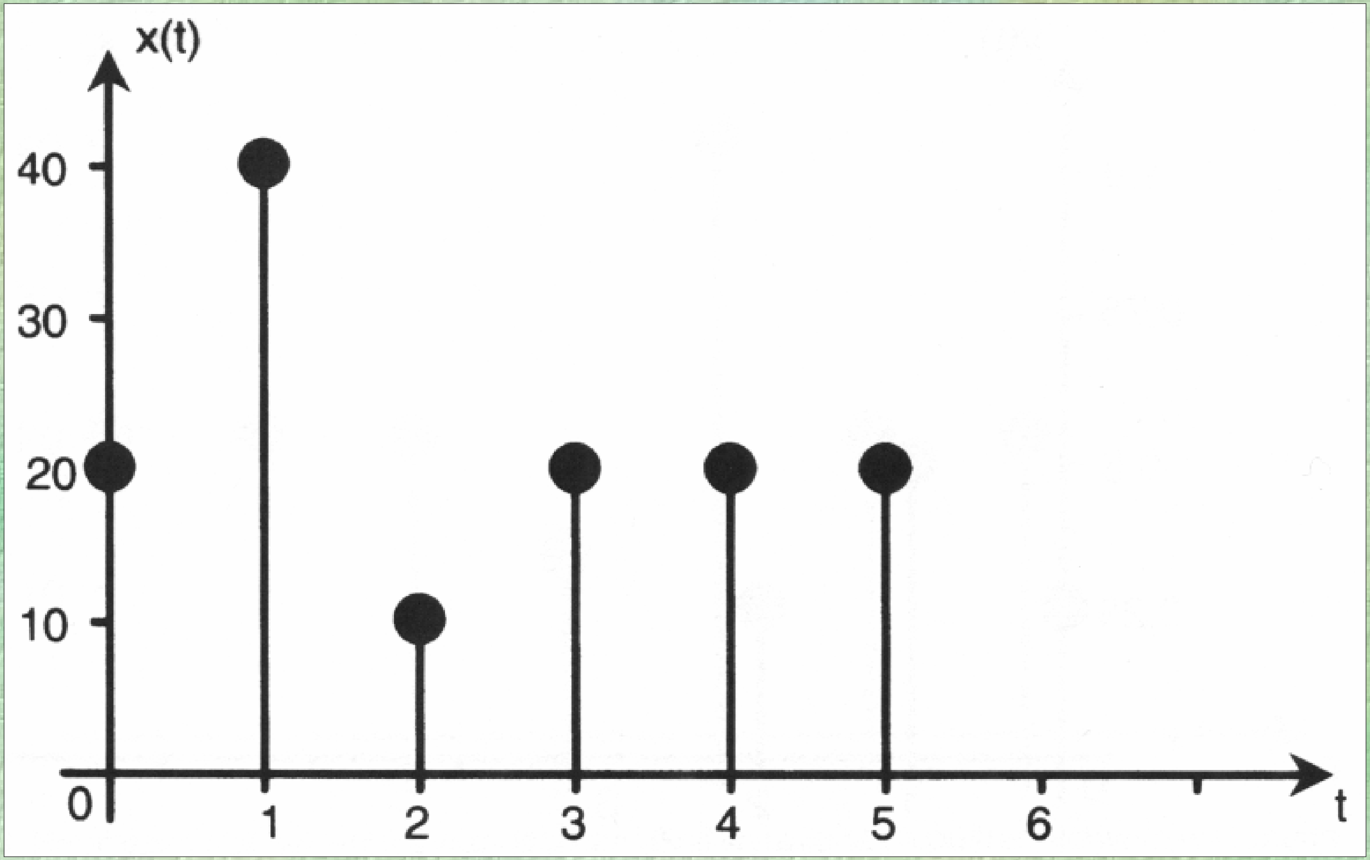
Jeśli powtórzymy przeprowadzoną powyżej analizę w stosunku do pojedynczej wartości wejściowej, wówczas otrzymamy ciąg impulsów zademonstrowanych na rysunku:



Ważona funkcja impulsowa ma tę zaletę, że pozwala nam przedstawić filtr w postaci graficznej.

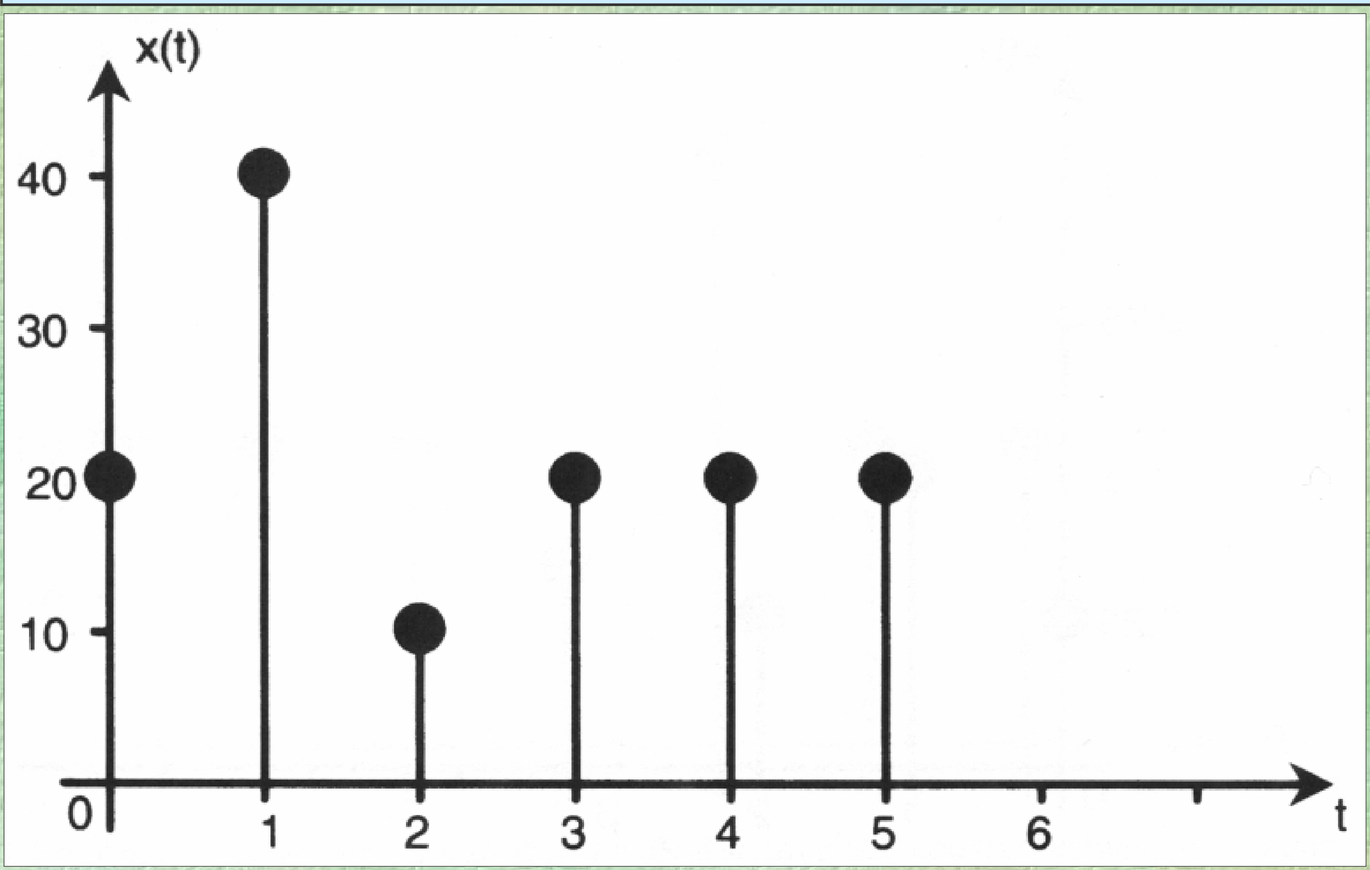
Dzięki temu możliwe staje się obliczenie sygnału wyjściowego w dowolnym czasie za pomocą prostego mnożenia odpowiedzi impulsowej filtru przez ciąg impulsów wejściowych z interesującego nas zakresu czasowego.

Rozpatrzmy zatem jeden z przypadków, powiedzmy przy $\boldsymbol{n} = 5$, dla którego ciąg wejściowy $\boldsymbol{x(t)}$ pokazano na rysunku:

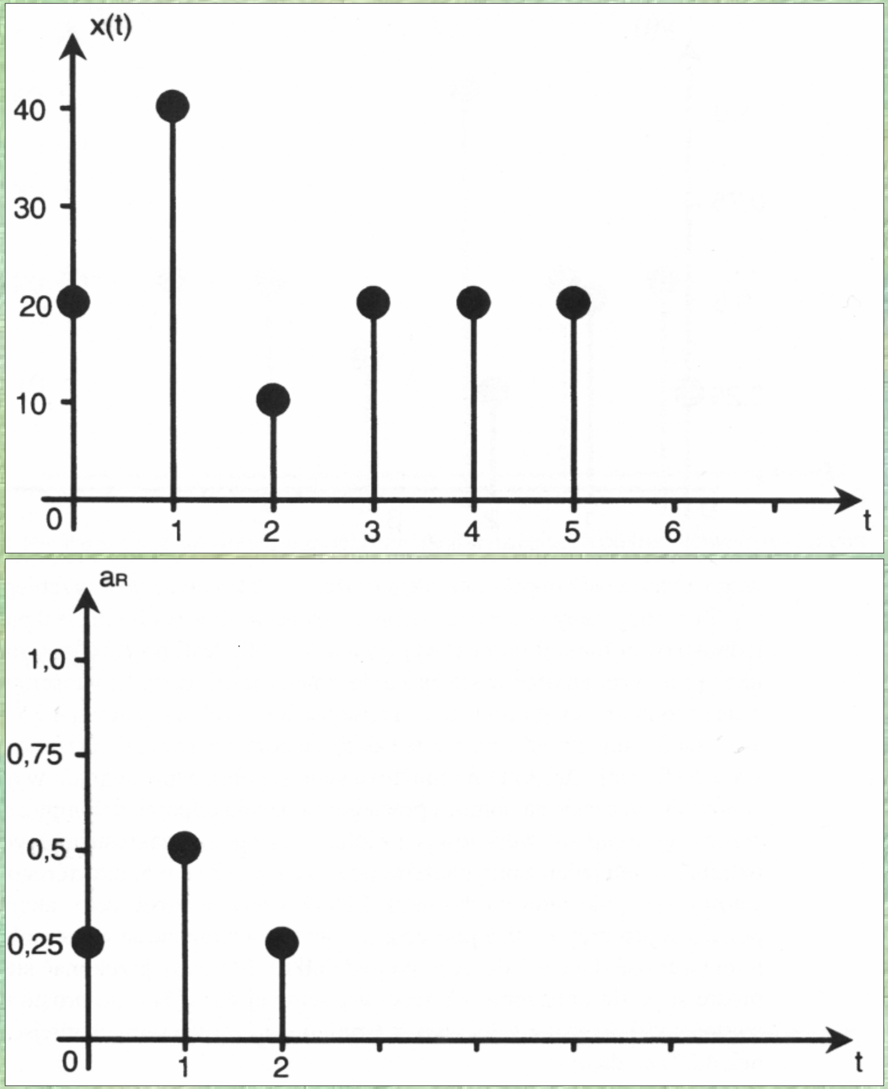


Dzień	**Cena ($)**
Pn	20
W	20
Ś	20
C	12
Pt	40
S	20
Ni	20

Bieżąca wartość ceny akcji (sobota) pokazana jest przy *t=0*, a poprzednie wartości wyrażone są w kierunku wstecznym (wartość dla *t = 5* dotyczy poniedziałku).

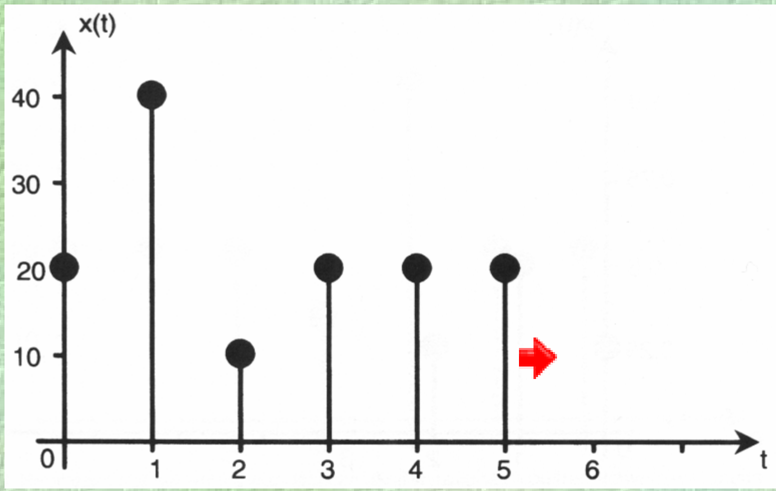


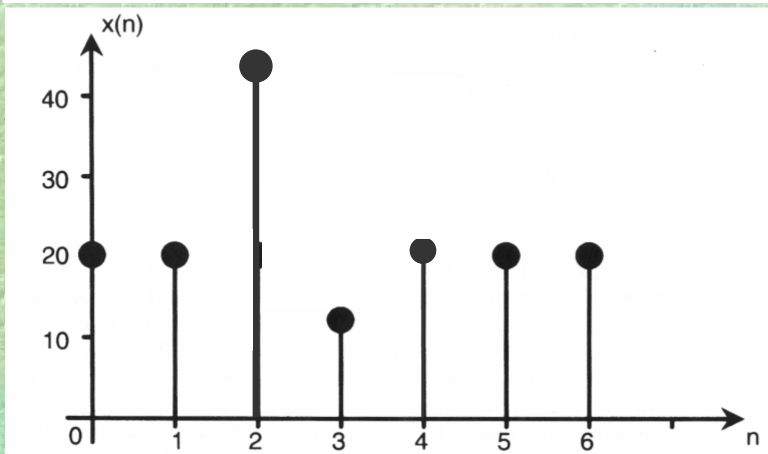
Dzień	Cena ($)
Pn	20
W	20
Ś	20
C	12
Pt	40
S	20
Ni	20

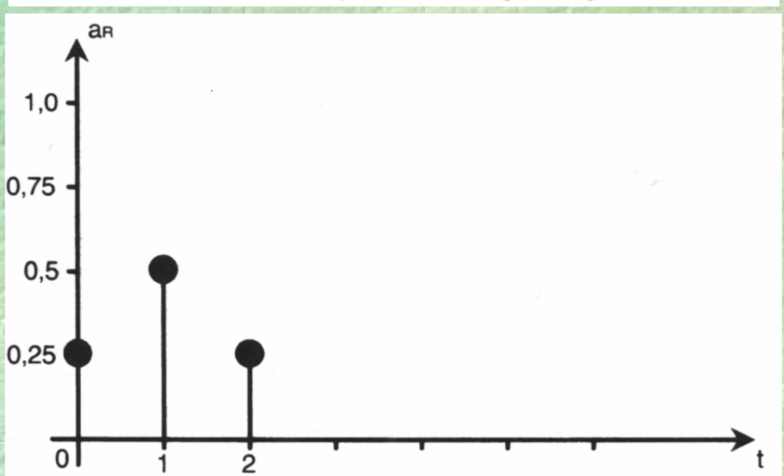


Możemy przekonać się teraz jak proste staje się obliczenie wartości wyjściowej dla ***y(5)*** - po prostu mnożymy wartości ***x(t)*** przez wartości współczynników a_R w tych samych miejscach co daje:

$$y(5) = 0{,}25 \times 20 + 0{,}5 \times 40 + 0{,}25 \times 12 = 28$$







Przechodząc do niedzieli - po prostu przesuwamy sekwencję wejściową o jeden stopień w prawo, przypisując wartości cen akcji dla niedzieli czas ***t* =0** i wykonujemy obliczenie.

Należy tu zauważyć, że dla przeprowadzenia omówionych przed chwilą obliczeń **niezbędne jest odwrócenie** sekwencji wejściowej ***x(n)*** przed wykonaniem mnożenia przez odpowiedź impulsową filtru.

Dzięki temu, przed obliczeniem każdej kolejnej wartości wyjściowej **po prostu przesuwamy sekwencję o jedną pozycję w prawo** i przypisujemy nowej wartości czas **t = 0**.

Na wstępie naszych rozważań filtr opisaliśmy następującym równaniem:

$$y(n) = a_0 \cdot x(n) + a_1 \cdot x(n-1) + a_2 \cdot x(n-2)$$

Teraz nieco rozwiniemy opis naszego filtru.

Powiedzieliśmy już, że jednostka opóźnienia jest wyrażona przez $\mathbf{z^{-1}}$.

Z jej pomocą zdefiniujemy opóźnione próbki wejściowe, co pokazano poniżej:

$$x(n-1) = x(n) \cdot z^{-1}$$

$$x(n-2) = x(n) \cdot z^{-1} \cdot z^{-1} = x(n) \cdot z^{-2}$$

Równanie filtru możemy teraz zapisać w nowej postaci:

$$y(n) = (a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}) \cdot x(n),$$

czy wreszcie:

$$H(n) = y(n)/x(n) = a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}.$$

Funkcja ***H(n)*** jest matematyczną reprezentacją impulsowej odpowiedzi filtru.

Częściej nazywana jest ona **funkcją przeniesienia.**

Zasadą, o której należy pamiętać jest to, że jeśli $\boldsymbol{z^{-n}}$ (gdzie $\boldsymbol{n}$ jest dowolną liczbą całkowitą) występuje tylko w liczniku ułamka funkcji przeniesienia, wówczas filtr jest wewnętrznie stabilny.

W przeciwieństwie do wielu analogowych rozwiązań filtrów ten filtr nigdy nie stanie się oscylatorem.

Jedną z podstawowych cech filtru **SOI** jest możliwość jego przystosowania do realizacji liniowej odpowiedzi fazowej.

W celu osiągnięcia liniowości fazy powinniśmy przyswoić współczynnikom filtru wartości symetryczne względem środkowego odprowadzenia filtru, tak by:

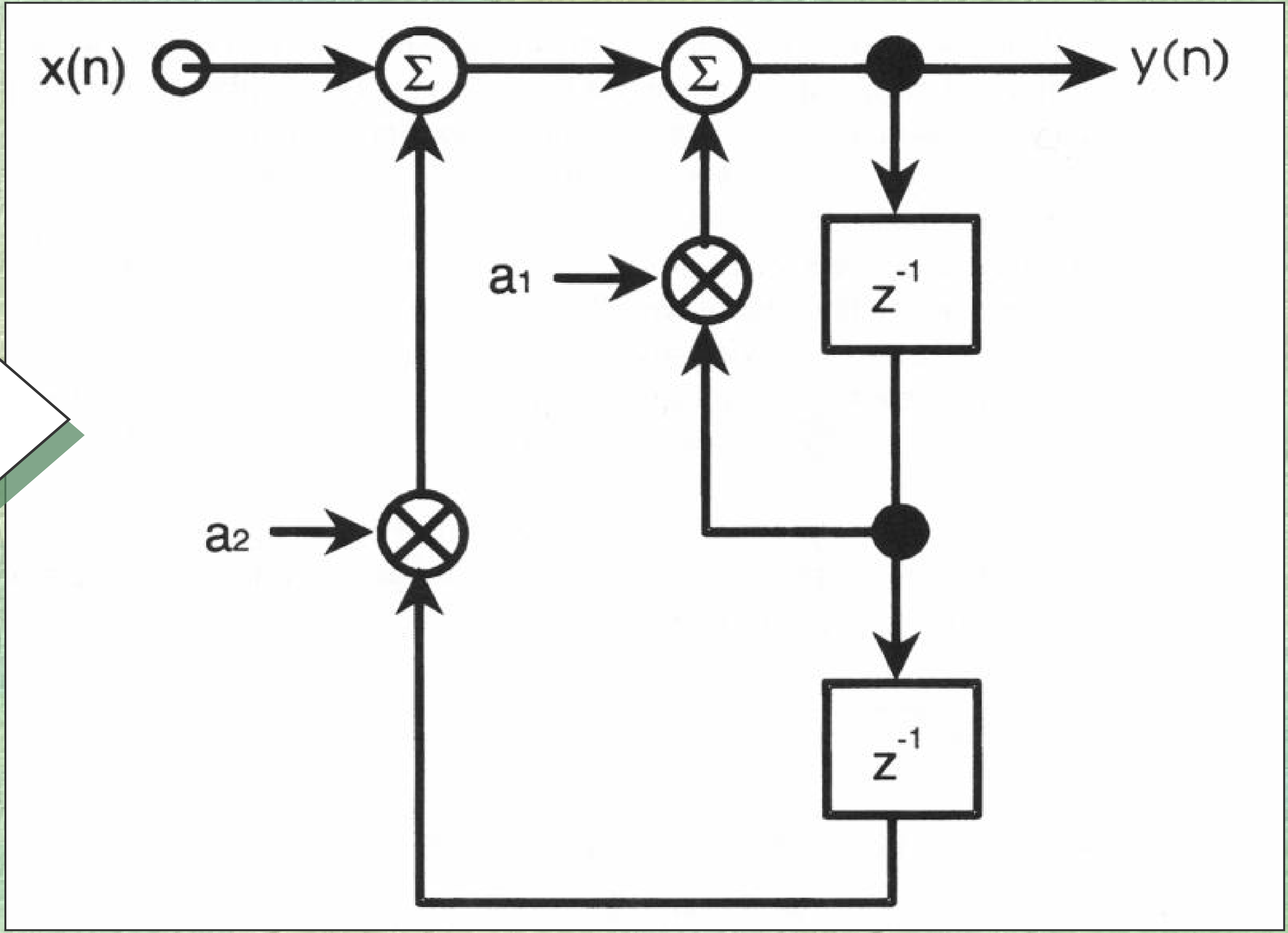
$$\boldsymbol{a_0 = a_R\ ,\ a_1 = a_{R-1}}\ \text{i tp.}$$

gdzie ***R+1*** jest liczbą odprowadzeń filtru.

Filtry o Nieskończonej Odpowiedzi Impulsowej

Inna podstawowa forma filtrów cyfrowych charakteryzuje się strukturą o **Nieskończonej Odpowiedzi Impulsowej (NOI).**

Uproszczona postać filtru NOI



Stosując identyczny zapis, jaki wykorzystywaliśmy przy omawianiu filtrów **SOI**, otrzymujemy:

$$y(n) = x(n) + a_1 \cdot y(n-1) + a_2 \cdot y(n-2) =$$

$$x(n) + [a_1 \cdot z^{-1} + a_2 \cdot z^{-2}] \cdot y(n) =$$

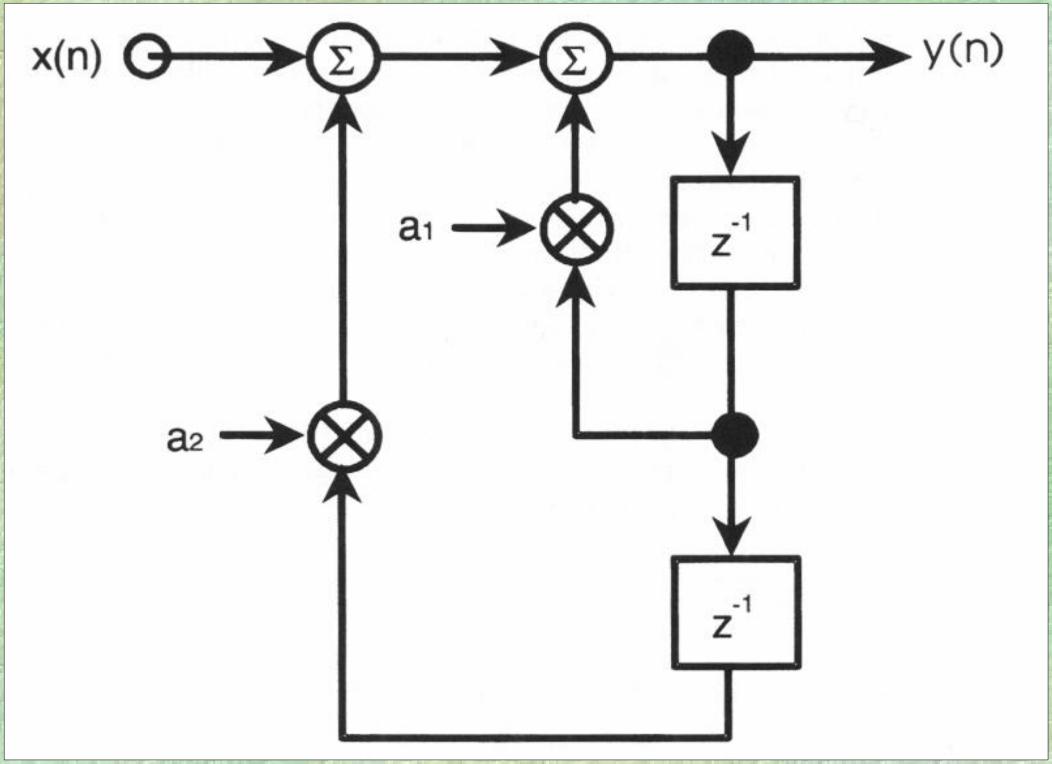
$$= x(n) \frac{1}{1 - a_1 z^{-1} + a_1 z^{-2}}$$

Pomińmy tu wyprowadzenie tego wyrażenia - jest to prosta operacja podstawiania.

Na podstawie powyższych wyrażeń **funkcja przeniesienia** może być przedstawiona przez:

$$\boxed{H(n) = \frac{y(n)}{x(n)} = \frac{1}{1 - a_1 z^{-1} + a_1 z^{-2}}}$$

Jeśli przyjrzymy się ponownie rysunkowi wówczas zobaczymy, że filtr ten jest w istocie ciągiem pętli sprzężenia zwrotnego i w związku z tym, przy zaistnieniu pewnych warunków, może on stać się niestabilny.



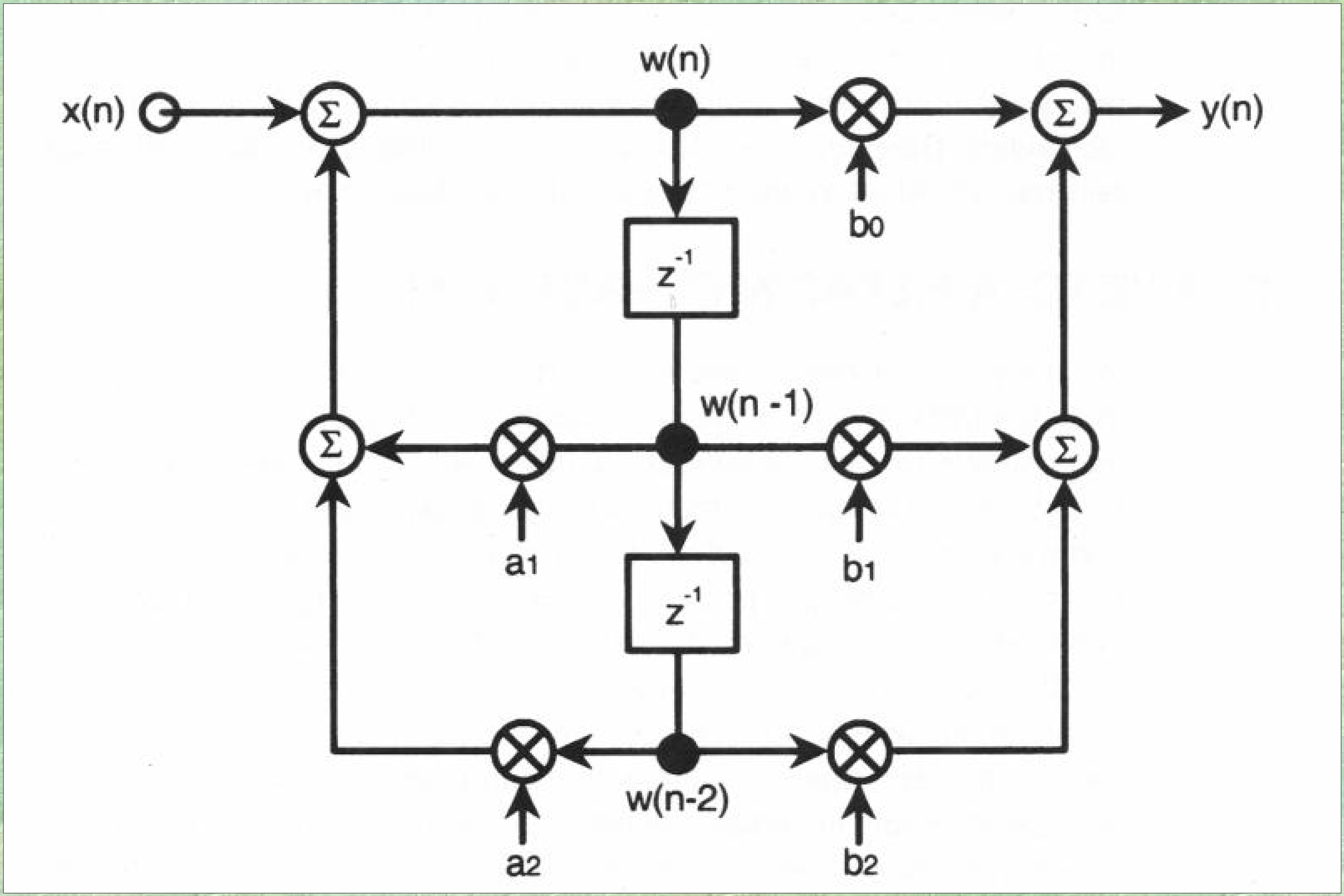
Aczkolwiek pojawienie się niestabilności jest możliwe w filtrach **NOI**, to mają one taką zaletę, że dla takiej samej charakterystyki nachylenia wymagają mniejszej liczby odprowadzeń, niż filtry **SOI**.

Oznacza to, że jeśli nasze zasoby obliczeniowe są ograniczone, możemy zastosować filtr **NOI** zwracając jednak baczną uwagę na staranne zaprojektowanie jego stabilności.

Filtry **NOI** są powszechnie stosowane przy realizacji sprzężeń międzyukładowych dla prądu zmiennego oraz w celu wygładzania (uśredniania) przebiegów.

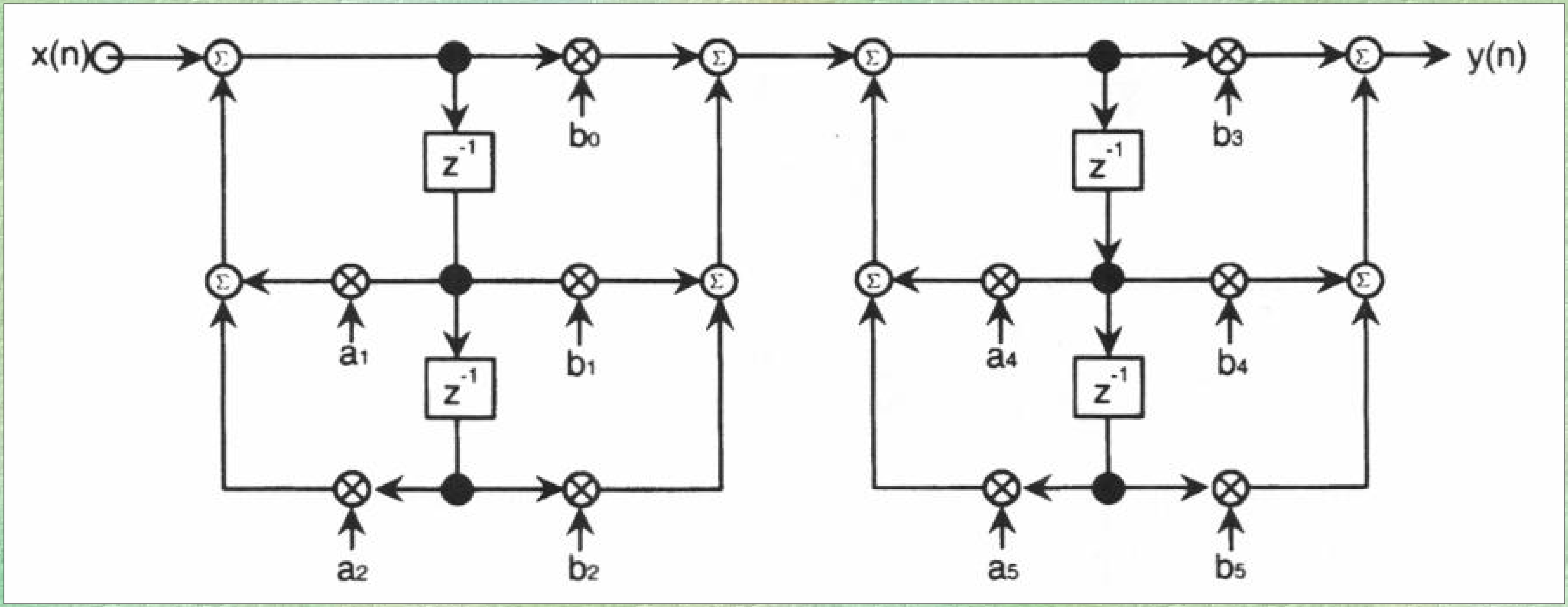
Najczęściej jednak wykorzystywane są łączone struktury filtrów **SOI** i **NOI**.

Na przykład popularna jest struktura:



Jeśli potrzebujemy filtrów o większej liczbie odprowadzeń, wówczas po prostu łączymy kaskadowo omówiony filtr drugiego rzędu (dwa odprowadzenia).

Na przykład poniżej pokazano **kaskadowy** filtr czwartego rzędu.



Zaletą takiego właśnie podejścia do projektowania filtrów jest ich modulamość, co powoduje znacznie łatwiejszą implementacje.

Istnieje także wiele innych postaci filtrów **NOI**.

W celu realizacji struktur filtracyjnych wyższych rzędów stosowane są różnorodne sposoby łączenia miedzy sobą sekcji filtrów drugiego rzędu.

Zademonstrowaliśmy tu jedynie połączenie kaskadowe - innym oczywistym sposobem połączenia jest struktura równoległa.

Każda z tych metod ma swoje wady jak i zalety.

PORÓWNANIE FILTRÓW SOI i NOI

W zastosowaniach, gdzie podstawowym wymaganiem jest liniowość charakterystyki fazowej preferowane są filtry **SOI**.

Mogą być one zaprojektowane z dokładnie liniową charakterystyką fazową, gdyż są stabilne i ich zachowanie się można dokładnie przewidzieć.

Stosujemy je także w sytuacjach, gdzie wymagany jest duży stopień kontroli odpowiedzi fazowej systemu.

Filtry **NOI** natomiast charakteryzują się kiepską odpowiedzią fazową, która jest wyjątkowo nieliniowa na brzegach pasm.

Niektóre z programów komputerowego projektowania filtrów nawet nie biorą pod uwagę odpowiedzi fazowej filtru jako parametru projektowego.

Wynika to z założenia, że stosując filtry **NOI** staramy się osiągnąć jak najbardziej strome nachylenie charakterystyki z użyciem jak najmniejszej liczby odprowadzeń - w tym celu poświęcamy jakość odpowiedzi fazowej.

Pod względem nachylenia charakterystyki amplitudowej filtr **NOI** jest pięć razy bardziej efektywny niż filtr **SOI** przy takiej samej liczbie odprowadzeń.

Biorąc pod uwagę względy ekonomiczne, tzn. złożoność realizacji sprzętowej oraz względy czasowe, czyli szybkość wykonywania obliczeń okazuje się, że filtr **NOI** także osiąga znaczną przewagę.

Natomiast czysty filtr **SOI**, tzn. filtr z elementami skomponowanymi w porządku liniowym jest ze swej natury stabilny i pewny w działaniu.

Filtry **NOI** zawsze zawierają jakąś składową sprzężenia zwrotnego i, jak wiemy, przy zaistnieniu określonych warunków może to powodować wzbudzanie się układu.

Wymaga to znacznie większej uwagi podczas projektowania filtrów **NOI**.

SZUMY W UKŁADACH FILTRÓW

Kwantyzacja sygnału

Wiemy, że wszystkie typy przetworników **a/c** wprowadzają do przychodzącego sygnału błędy kwantyzacji.

Jeśli rozdzielczość wyjściowa przetwornika **c/a** jest niższa niż wewnętrzna rozdzielczość DSP, mamy do czynienia z pojawianiem się dodatkowych szumów w sygnale wyjściowym.

Trzeba tu jednak dodać, że ten typ szumu nie jest rozprzestrzeniany wzdłuż filtru i nie może zależeć od jego rozwiązania układowego.

Jeśli do 16-to bitowego DSP zastosujemy **10-** lub **12**-bitowe przetworniki **a/c** i **c/a**, wówczas maksymalny możliwy poziom szumów na wejściu filtru wynosi $\mathbf{2^{-10}}$ lub $\mathbf{2^{-12}}$.

Przy założeniu, że filtr ma wzmocnienie niższe niż 0 dB w całym zakresie częstotliwościowym dodane szumy nie są znaczące.

Kwantyzacja współczynników

Wpływ kwantyzacji współczynników na błędy stałoprzecinkowych układów DSP jest bardziej istotny.

Większe zmiennoprzecinkowe urządzenia DSP o 32 i więcej bitach są o wiele bardziej dokładne i w ich przypadku wpływ kwantyzacji współczynników może być pominięty.

Stałoprzecinkowe urządzenia DSP mają z reguły 16-bitową szerokość słowa i w celu uniknięcia przepełnienia akumulatora DSP wszystkie dane wyrażane są w formie ułamkowej.

Dokładność współczynników jest bardzo istotna.

Powód konieczności stosowania dużej dokładności współczynników jest taki, że podczas mnożenia dwóch liczb stałoprzecinkowych wynikłe błędy zwielokrotniają się.

Na przykład dokonajmy mnożenia wartości wejściowej 0,39 przez 0,4:

$$\begin{array}{r} \times \quad 0{,}39 \\ 0{,}4 \\ \hline 0{,}156 \end{array}$$

Przekłamanie o jeden znak dziesiętny współczynnika może spowodować zamianę 0,4 na 0,3 i rezultat mnożenia będzie teraz inny:

$$\begin{array}{r} \times \quad 0{,}39 \\ 0{,}3 \\ \hline 0{,}117 \end{array}$$

!

co daje błąd o wartości **25%.**

W zasadzie jednak, przy zastosowaniu właściwego skalowania, 16-bitowe współczynniki są wystarczająco dokładne.

Błąd najmniej znaczącego bitu (LSB) wynosi wówczas mniej niż 0,002%.

W przypadku filtrów **NOI** błąd najmniej znaczącego bitu może okazać się bardziej niebezpieczny.

Pojedyncze odchylenie LSB w jednym elemencie sprzężenia zwrotnego może spowodować po przejściu pierwszej pętli tylko nieznaczny błąd, lecz po wielu pętlach sprzężenia błąd ten może się nawarstwić do tego stopnia, że spowoduje wzbudzenie się filtru.

Większość programów do komputerowego projektowania filtrów umożliwia sprawdzenie stabilności filtrów NOI.

Przed wprowadzeniem filtru w strukturę półprzewodnikową warto przekonać się, czy małe odchylenie wartości współczynnika nie spowoduje problemów w wyrobie końcowym.

Szumy obcięcia

W celu pełnego zapamiętania rezultatu podczas mnożenia przez siebie dwóch liczb ***n***-bitowych wymagana jest przestrzeń ***2n***-bitowa.

Zademonstrujemy to na przykładzie mnożenia dziesiętnych oraz binarnych:

$$\begin{array}{r} 0{,}64 \\ \times\; 0{,}73 \\ \hline 192 \\ 4480 \\ \hline 0{,}\underbrace{4672} \end{array} \qquad\qquad \begin{array}{r} 11 \\ \times\; 11 \\ \hline 11 \\ 110 \\ \hline \underbrace{1001} \end{array}$$

Z tego powodu wszystkie stałoprzecinkowe urządzenia DSP- rejestr iloczynu oraz akumulator mają podwójne długości.

Na przykład w 16-bitowym układzie TMS320C25 rejestr iloczynu (P) i akumulator (ACC) mają długość 32-bitową.

Ponieważ do akumulatora dodawany jest iloczyn 32-bitowy, więc w przeciągu pełnego cyklu pracy tego podprogramu pod uwagę brane są wszystkie 32 bity.

Po wykonaniu cyklu 32-bitowy rezultat powinien być zapamiętany w pamięci 16-bitowej.

Możliwe jest to z wykorzystaniem dwóch odrębnych cykli zapisu do dwóch kolejnych komórek pamięci, co jednak podwaja czas operacji, zajmuje większy rozmiar pamięci oraz podwaja czas odczytu rezultatu podczas jego wykorzystania do kolejnych obliczeń.

Z tego powodu zwykle zapamiętywana jest górna, bardziej znacząca połowa rezultatu i pełny rezultat zostaje obcięty.

Błąd wprowadzany przez opisane obcięcie jest istotny wyłącznie dla 16-tego bitu i stanowi mniej niż 0,001 %.

W urządzeniach DSP TMS320Cxx istnieje jeszcze inna opcja zaokrąglania rezultatu, powodująca dodatkową redukcję błędu.

W większości realnych zastosowań DSP zapewnia to w pełni wystarczającą dokładność.

„Overflow” oraz „Underflow”

W większości procesorów zachodzi zjawisko przepełnienia wewnętrznego **(overflow)** polegające na tym, że jeśli podczas operacji matematycznej rezultat wyjściowy przekracza zakres akumulatora, wówczas rezultat zostaje obcięty od góry.

Na przykład, podczas dodawania liczb 0110 i 1111 rezultat powinien wynosić 10101.

W procesorach 4-bitowych zachowywane są jedynie cztery najmłodsze bity i bit najbardziej znaczący zostaje utracony, a rezultatem dodawania staje się liczba 0l0l, co jest oczywiście błędem.

Identycznie sprawa się ma z niedoborem arytmetycznym **(underflow)** powodującym obcięcie rezultatu w kierunku ujemnym.

Podobnie do innych przedyskutowanych już rodzajów błędów efekt przepełnienia lub niedoboru nie jest dla filtrów SOI katastrofą, gdyż jest ograniczony do pojedynczego przejścia przez filtr.

Natomiast w filtrach NOI błąd na wyjściu będzie w swojej części wprowadzany ponownie na wejście co powoduje nieustanne jego nawarstwianie.

W celu zapobieżenia tym efektom w DSP przewidziany jest **specjalny tryb pracy** nazywany **trybem nasycenia**.

W tym trybie, w przypadku przekroczenia maksymalnego dopuszczalnego zakresu akumulatora DSP, wyjście układu przechodzi do stanu nasycenia i wszystkie bity akumulatora zostają ustawione w stan jedynkowy.

Natomiast, gdy wynik działania jest mniejszy od minimalnego dopuszczalnego zakresu, wówczas akumulator nasyca się ustawiając wszystkie swoje bity do stanu zerowego.

Metoda ta zapobiega podawaniu niewłaściwych rezultatów w sprzężeniu zwrotnym, lecz nie jest oczywiście idealna, gdyż algorytm filtrujący nie oblicza w takim przypadku dokładnych rezultatów wyjściowych.

Wynika stąd wniosek, **że filtry powinny być projektowane w ten sposób, by unikać pojawienia się przepełnienia lub niedoboru.**

Producenci urządzeń DSP załączają zawsze programy symulacyjne, których możemy użyć w celu modelowania działania filtru.

Możemy sprawdzić każdą możliwość wystąpienia wewnętrznego przepełnienia i odpowiednio przeskalować sygnały wejściowe usuwając w ten sposób większość problemów.

Ograniczenia zakresu dynamicznego

Zakres dynamiczny urządzeń DSP może ograniczać ogólną wydajność filtru, gdyż jest bezpośrednio związany z szerokością słowa procesora.

Liczba dostępnych bitów jest związana z minimalnym rozmiarem kroku kwantyzacyjnego, a zatem z granulacją filtru.

W przypadku ograniczonej liczby bitów urządzenia w celu zapewnienia akwizycji pełnej znaczącej informacji sygnału wejściowego należy stosować bardziej skomplikowane metody kwantyzacyjne.

Urządzenia stałoprzecinkowe rodziny DSP TMS320Cxx mają słowa o długości 16 bitów.

Wynikający z tego zakres dynamiczny wynosi:

$$20log_{10}(2^{16}) = 20log_{10}(65536) = 96 \text{ dB}.$$

Akumulator i rejestr iloczynu jednostki arytmetycznej tych urządzeń zostały rozszerzone do 32 bitów, zapewniając tym samym dla obliczeń pośrednich zakres dynamiczny równy 192 dB.

Jest to wielkość zupełnie wystarczająca dla większości zastosowań.