Zastosowania Procesorów Sygnałowych

Adam Korzeniewski
adamkorz@sound.eti.pg.gda.pl
p. 732 - Katedra Systemów Multimedialnych

Bufory kołowe i szybki splot

Bufory liniowe i kołowe

Filtr FIR jest systemem o transmitancji

$$H(z) = \frac{Y(z)}{X(z)} = b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_M z^{-M}$$

realizującym algorytm opisany następującym równaniem różnicowym

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + \ldots + b_M x[n-M]$$

gdzie M jest rzędem filtru. Współczynniki filtru b_i są zarazem próbkami skończonej odpowiedzi impulsowej filtru

$$h[n] = \{h_0, h_1, h_2, \ldots, h_M\} = \{b_0, b_1, b_2, \ldots, b_M\}$$

Przy zadanym przyczynowym, nieskończonym sygnale wejściowym

$$x[n] = \{x_0, x_1, x_2, \ldots\}$$

Bufory liniowe i kołowe

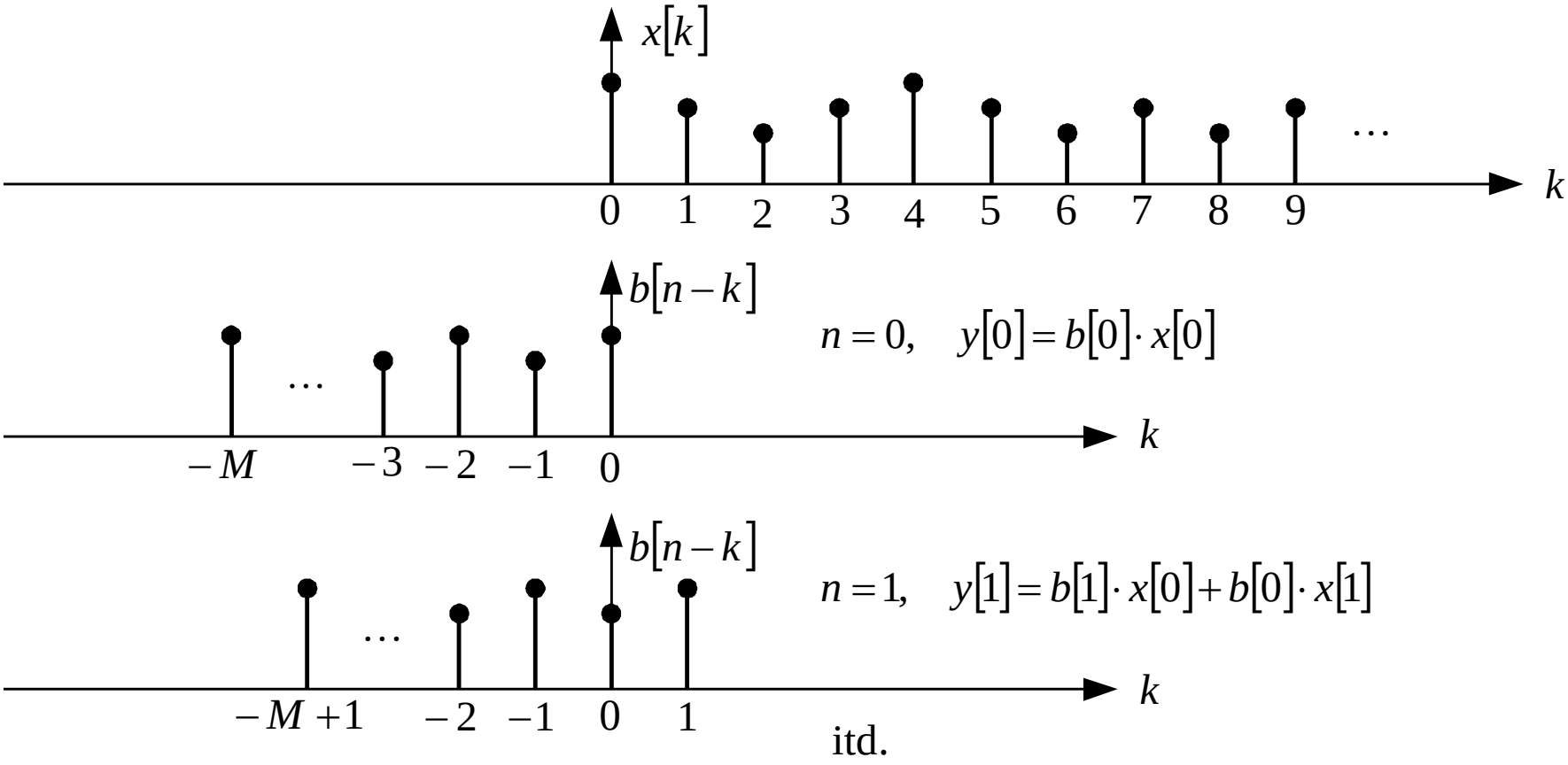
rekurencyjne obliczanie próbek sygnału wyjściowego

$$
\begin{aligned}
y[0] &= b_0 x[0] + b_1 \cdot 0 + b_2 \cdot 0 + \ldots + b_M \cdot 0 \\
y[1] &= b_0 x[1] + b_1 x[0] + b_2 \cdot 0 + \ldots + b_M \cdot 0 \\
y[2] &= b_0 x[2] + b_1 x[1] + b_2 x[0] + \ldots + b_M \cdot 0 \\
y[3] &= b_0 x[3] + b_1 x[2] + b_2 x[1] + \ldots + b_M \cdot 0 \\
&\vdots
\end{aligned}
$$

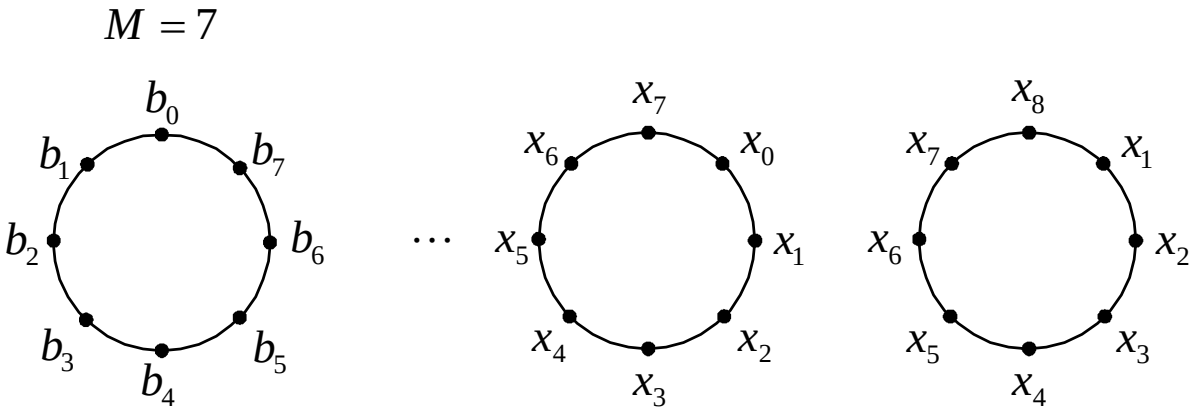
jest niczym innym jak obliczeniem splotu liniowego (sumy iloczynów)

$$y[n] = b[n] * x[n] = \sum_{k=0}^{M} b[n-k]x[k]$$

Bufory liniowe i kołowe



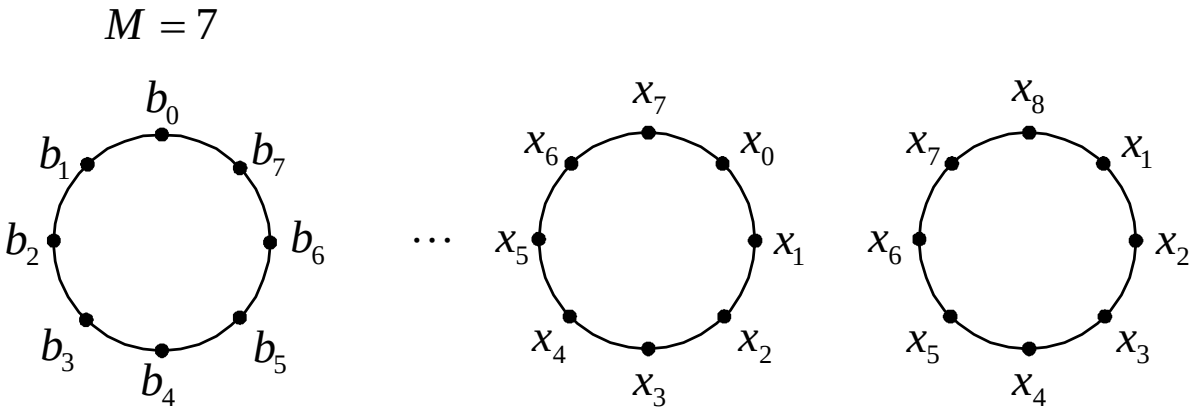
Bufory liniowe i kołowe



Wykonanie obliczeń splotu liniowego w procesorze sygnałowym wymaga wykonania następujących czynności:

1. Umieść *M+1* współczynników filtru *b[k]* w komórkach pamięci (w odwrotnej kolejności).
2. Umieść *M+1* próbek sygnału wejściowego w komórkach pamięci tworzących bufor liniowy odbiorczy.

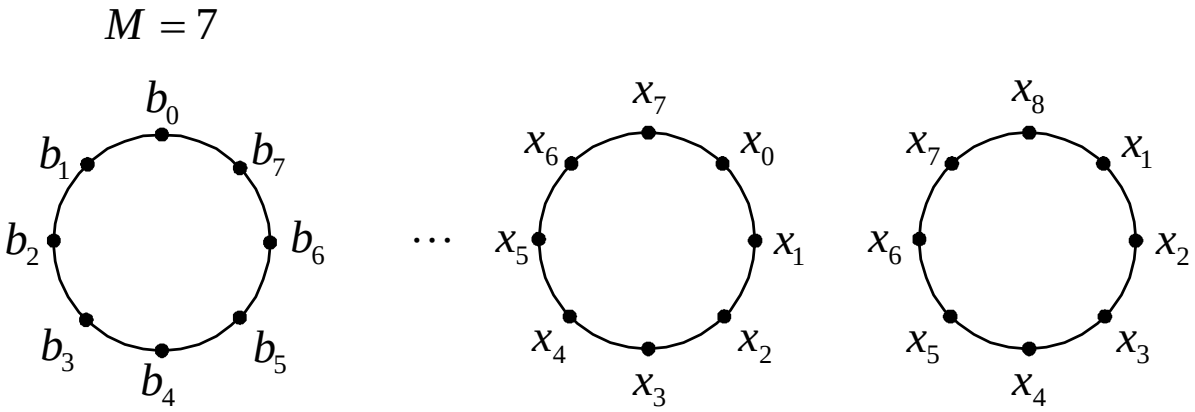
Bufory liniowe i kołowe



3. Oblicz próbkę sygnału wyjściowego $$y[n] = b[n] * x[n] = \sum_{k=0}^{M} b[n-k]x[k]$$

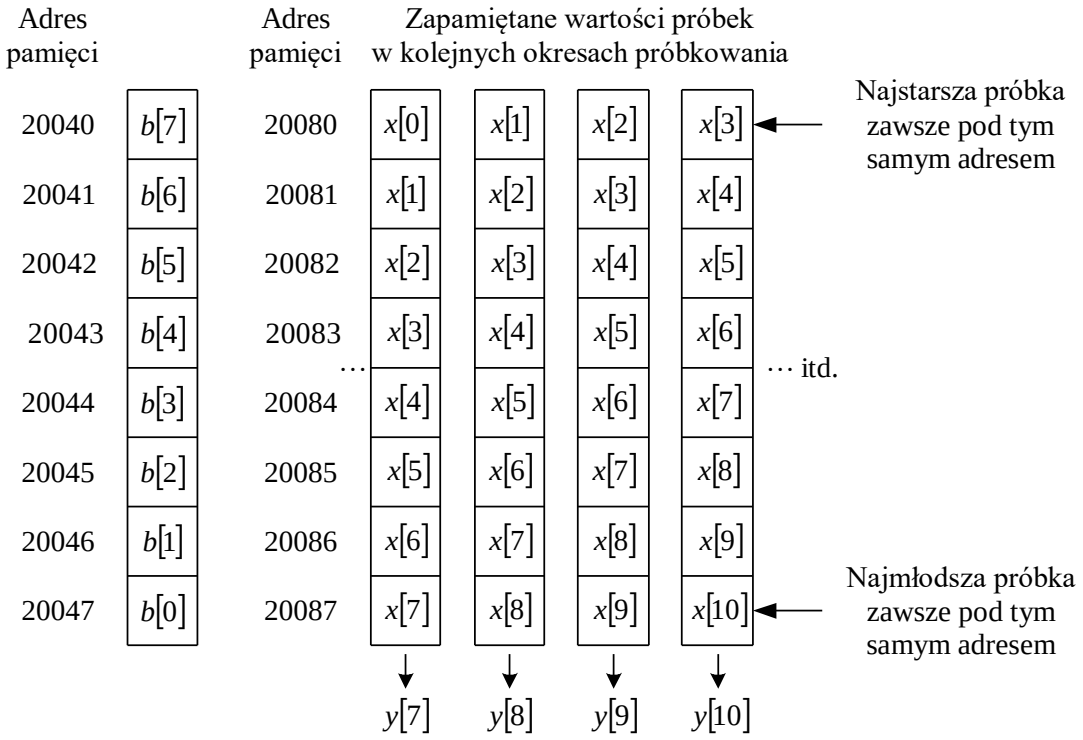
(oblicz sumę iloczynów, mnożone wartości znajdują się w komórkach pamięci „naprzeciw siebie”, jak na przesuwanych linijkach w graficznej interpretacji splotu liniowego).

Bufor liniowy - Filtr FIR



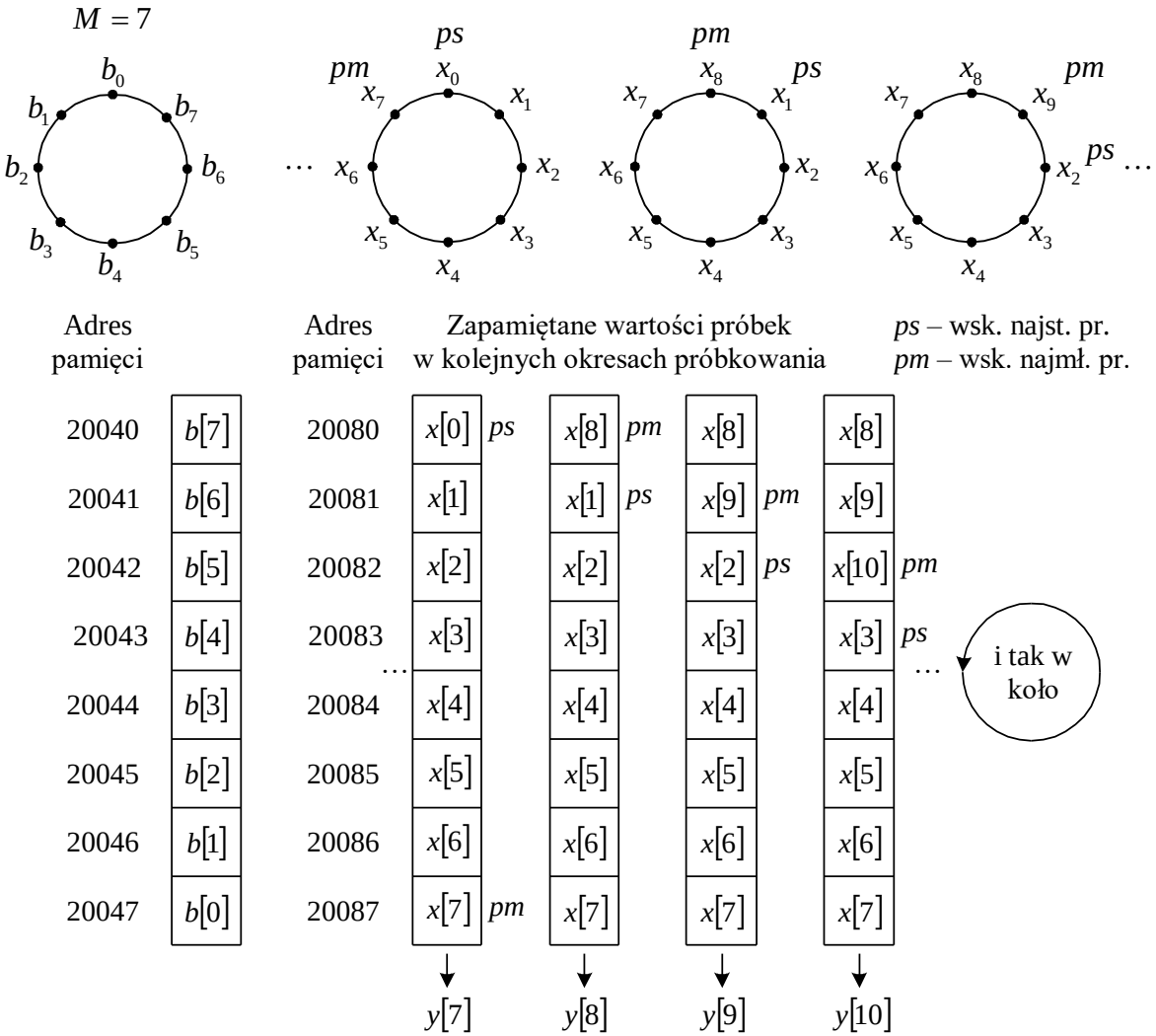
4. Umieść obliczoną próbkę sygnału wyjściowego *y[n]* w komórce pamięci tworzącej bufor liniowy nadawczy.
5. Usuń najstarszą próbkę sygnału wejściowego, przesuń wszystkie próbki tak, aby na miejscu najmłodszej próbki zrobić miejsce i wpisz tam najnowszą próbkę sygnału wejściowego.
6. Powróć do punktu 3.

Bufory liniowe i kołowe



Wadą wykonywania obliczeń według powyższego schematu jest to, że każdorazowo wpisanie najmłodszej próbki pociąga za sobą konieczność zmiany adresów wszystkich starszych próbek. Wady tej nie ma **bufor kołowy**, w którym najmłodszą próbkę wpisuje się w miejsce najstarszej usuwanej próbki, a miejsce kolejnych próbek jest znane dzięki zastosowaniu wskaźników (ang. pointer, jest to liczba integer), wskaźnika najstarszej próbki i wskaźnika najmłodszej próbki .

Bufor kołowy



Bufor kołowy

W przypadku filtru IIR mamy do czynienia z systemem o transmitancji

$$H(z)=\frac{Y(z)}{X(z)}=\frac{b_0+b_1z^{-1}+b_2z^{-2}+\ldots+b_Mz^{-M}}{1+a_1z^{-1}+a_2z^{-2}+\ldots+a_Nz^{-N}}$$

realizującym algorytm opisany następującym równaniem różnicowym

$$y[n]=b_0x[n]+b_1x[n-1]+\ldots+b_Mx[n-M]-a_1y[n-1]-a_2y[n-2]+\ldots-a_Ny[n-N]$$

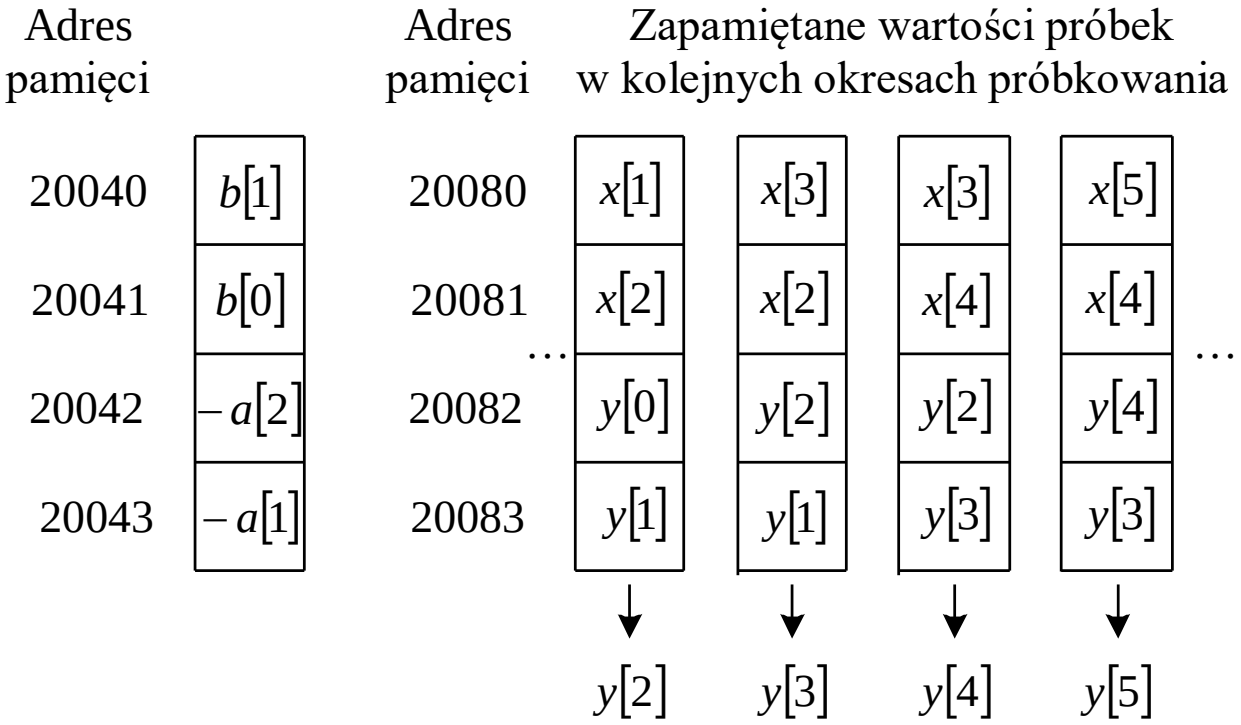
które jest rozwiązywane rekursywnie przy zadanych warunkach początkowych

$$y[-1],\ y[-2],\ \ldots,\ y[-N]$$

$$\begin{aligned}
y[0]&=b_0x[0]+b_1\cdot 0+\ldots+b_M\cdot 0-a_1y[-1]-a_2y[-2]+\ldots-a_Ny[-N]\\
y[1]&=b_0x[1]+b_1x[0]+\ldots+b_Mx[1-M]-a_1y[0]-a_2y[-1]+\ldots-a_Ny[1-N]\\
y[2]&=b_0x[2]+b_1x[1]+\ldots+b_Mx[2-M]-a_1y[1]-a_2y[0]+\ldots-a_Ny[2-N]\\
y[3]&=b_0x[3]+b_1x[2]+\ldots+b_Mx[3-M]-a_1y[2]-a_2y[1]+\ldots-a_Ny[3-N]\\
&\vdots
\end{aligned}$$

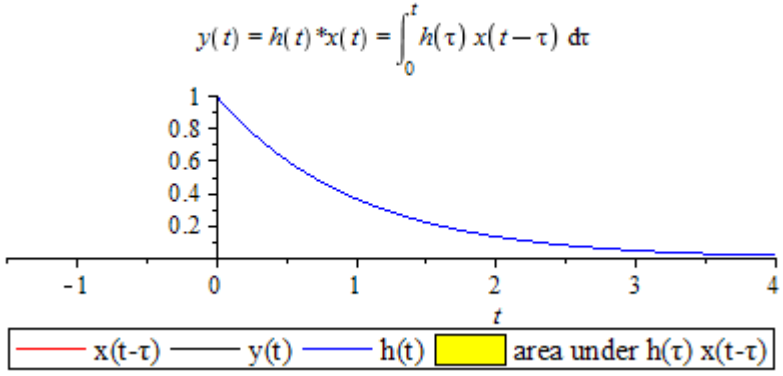
Bufor kołowy - Filtr IIR

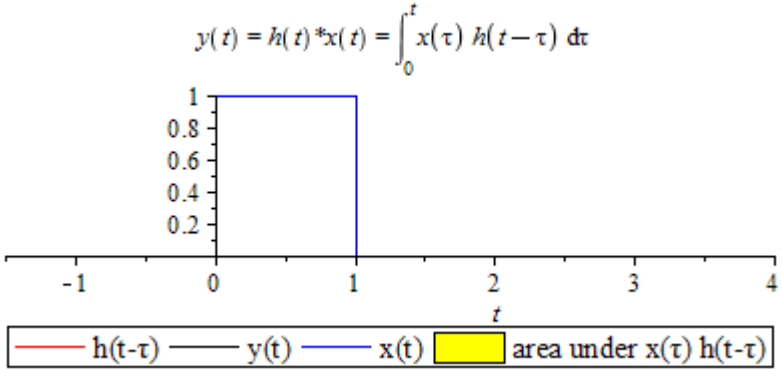
Przyjmując, że schemat obliczeń z buforami kołowymi będzie taki jak na poniższym rysunku (nie naniesiono tam wskaźników). Bufory kołowe filtrów IIR są w praktyce znacznie krótsze niż bufory kołowe filtrów FIR. Charakterystyczne jest to, że trzeba przenosić próbki z bufora nadawczego do bufora odbiorczego.



Splot

Operacja na dwóch funkcjach dająca w wyniku modyfikację oryginalnych funkcji (wynikiem jest iloczyn splotowy). Jest działaniem przemiennym (podobnie jak mnożenie)





Szybki splot - FFT

Algorytm FFT jest tak bardzo skuteczny, że w przypadku obliczania splotu kołowego $y[n] = x[n] \otimes h[n]$ (N^2 mnożeń i $(N-1)^2$ dodawań) opłaca się obliczyć widma sygnałów $x[n] \xrightarrow{FFT} X[k]$, $h[n] \xrightarrow{FFT} H[k]$,

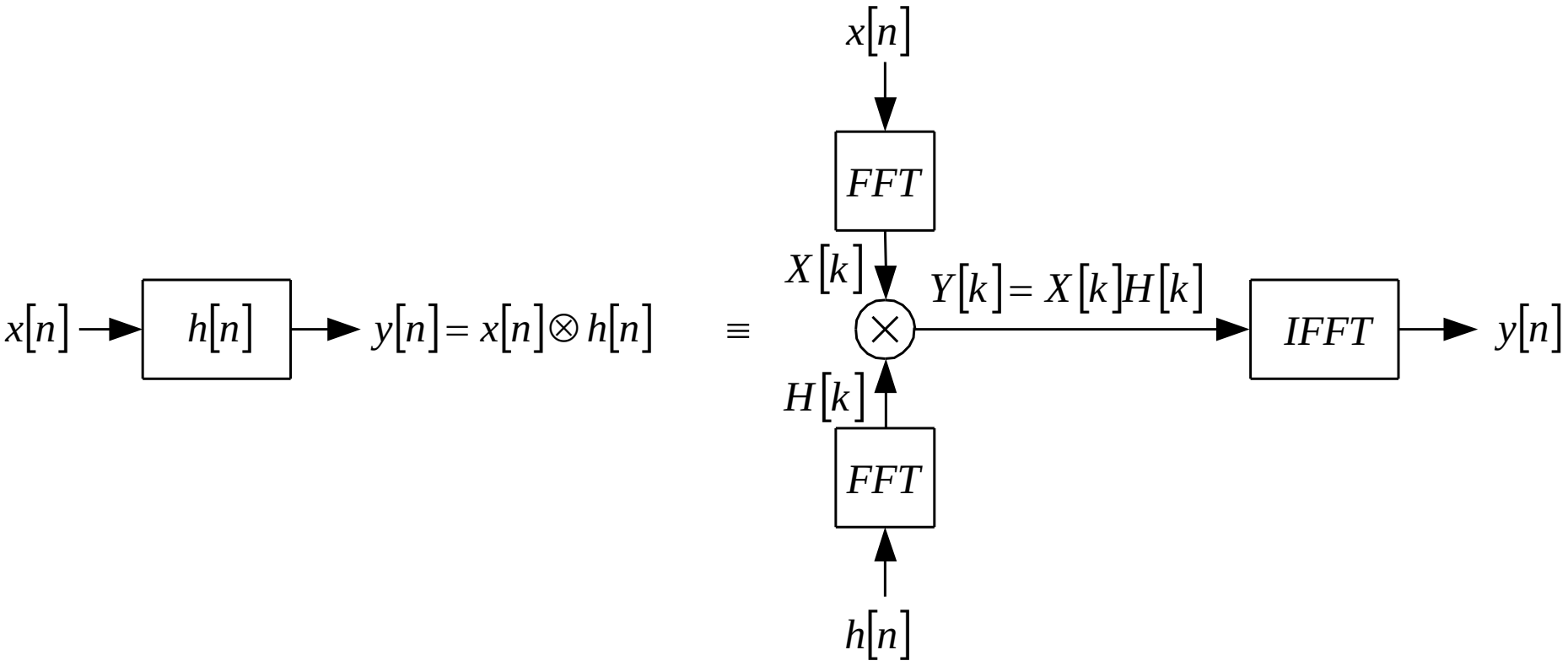
pomnożyć widma $Y[k] = X[k]H[k]$ i obliczyć

odwrotne przekształcenie $Y[k] \xrightarrow{IFFT} y[n]$.

Szybki splot

Ten sposób obliczania splotu nazywa się ***szybkim splotem***, gdyż obliczenia trwają krócej już od $N = 32$ w przypadku splatania sygnałów zespolonych i od $N = 64$ w przypadku splatania sygnałów rzeczywistych. Korzystne jest to, że w tym zastosowaniu FFT nie ma potrzeby przenumerowywania próbek. Dodatkowe korzyści odnosi się w typowej sytuacji, gdy splot jest obliczany dla różnych sygnałów $x[n]$ przy tej samej funkcji $h[n]$. Wystarczy wtedy obliczyć przekształcenie tylko jeden raz.

Szybki splot – schemat blokowy



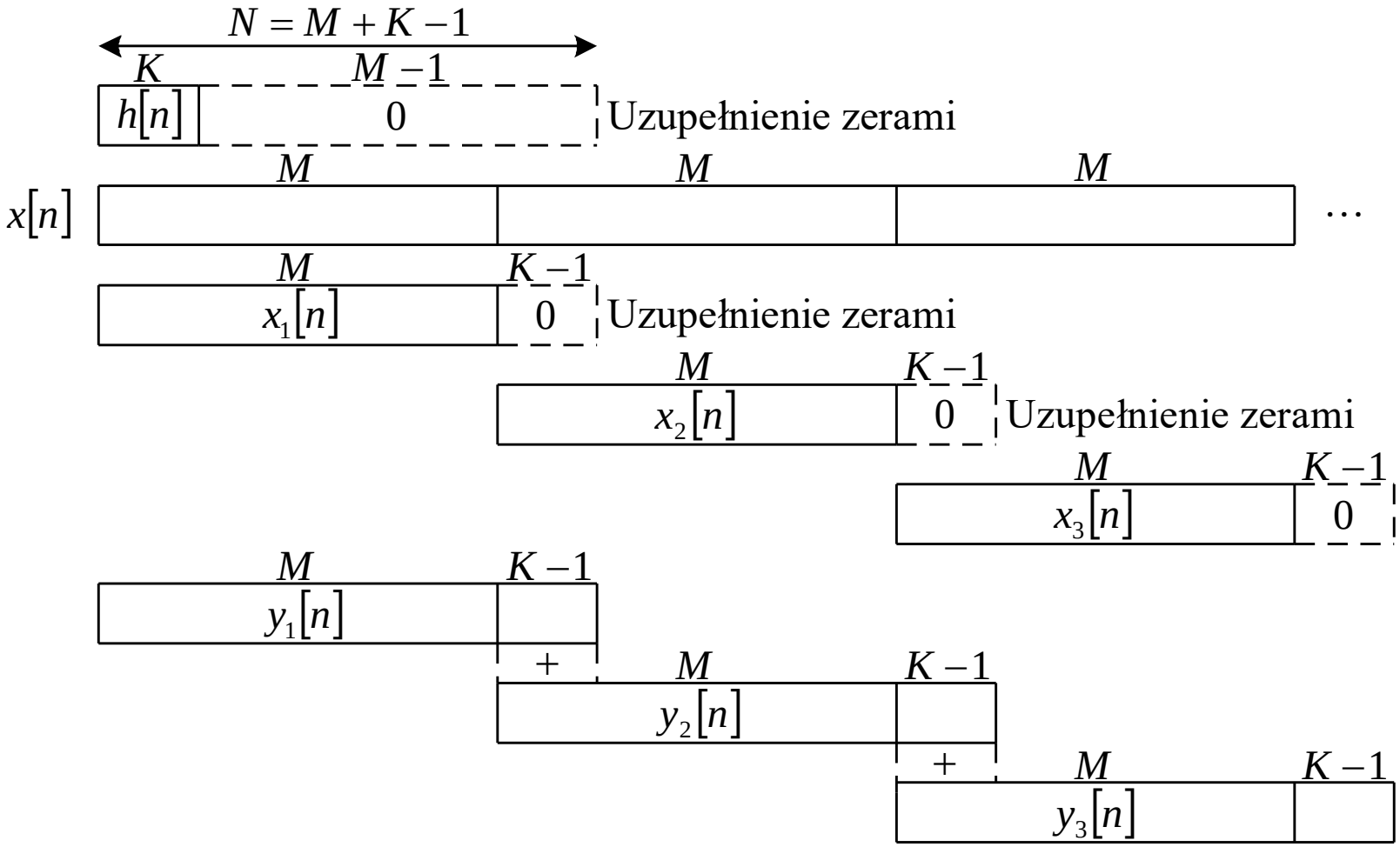
Szybki splot

Jeżeli sygnał wejściowy *x[n]* jest bardzo długi (czy wręcz nieskończony, tak jak to jest przy przetwarzaniu w czasie rzeczywistym sygnałów dźwiękowych, czy obrazów ruchomych), to jego splatanie z odpowiedzią impulsową filtru *h[n]* o długości *K* jest rozbijane na sumę szybkich splotów.

Szybki splot

Sygnał wejściowy należy rozbić na sumę bloków, każdy o długości *M*. Splot liniowy dwóch sygnałów o długościach *M* i *K* daje sygnał o długości *M + K - 1*. Splot kołowy powinien mieć taką właśnie długość *N = M + K - 1*, aby równał się splotowi liniowemu. Dlatego przed wykonaniem splotu kołowego sygnał *h[n]* jest uzupełniany *M-1* zerami do *N* oraz blok sygnału wejściowego $x_i[n]$ jest uzupełniany *K-1* zerami do *N*. Wyniki kolejnych splotów kołowych $y_i[n]$ są sumowane na zakładkę o długości *K-1*, stąd metoda ma angielską nazwę *overlap-add*.

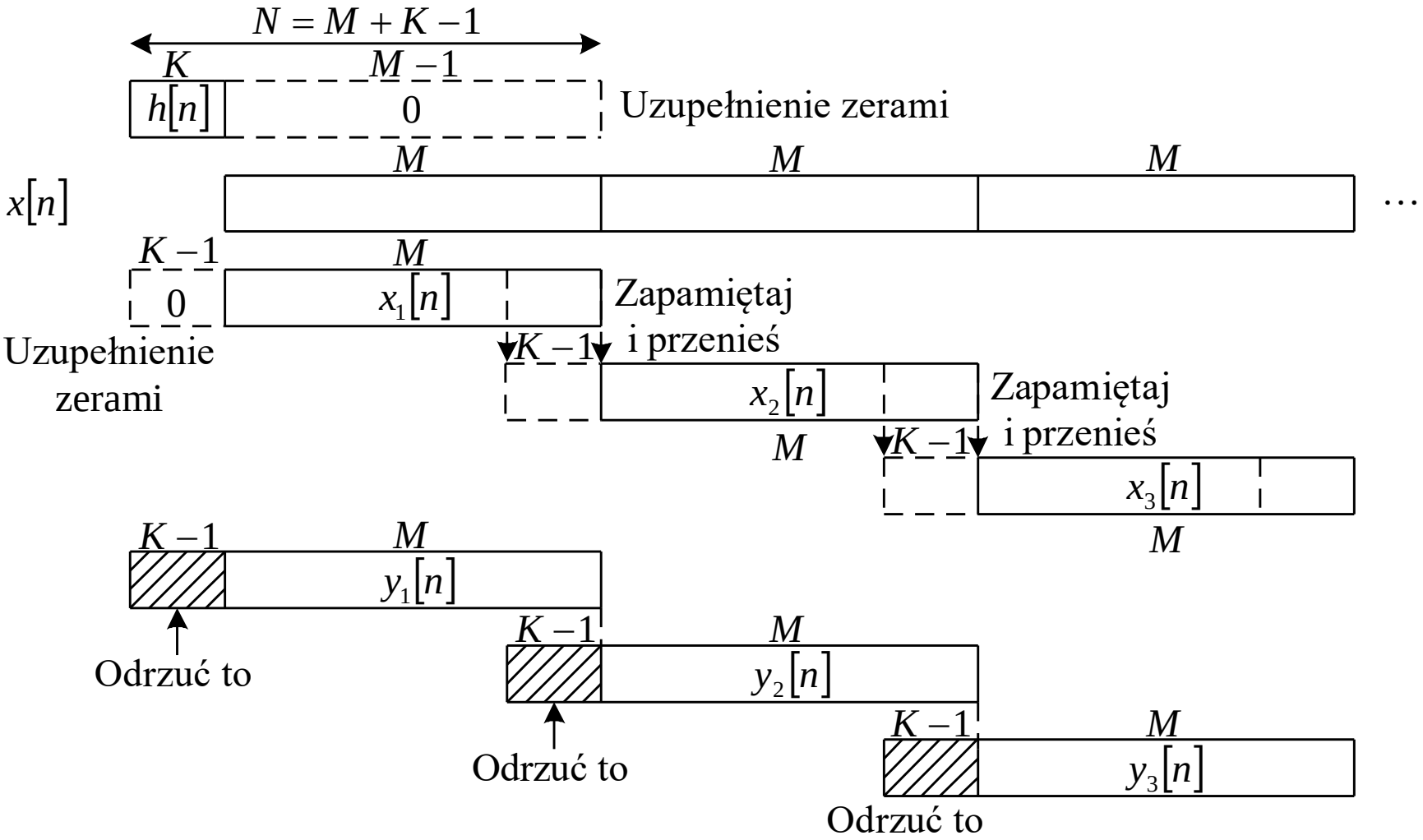
Szybki splot – obliczanie z dodawaniem zakładki (*overlap-add*)



Szybki splot

Równoważny sposób postępowania polega na zapamiętaniu *K-1* końcowych próbek z poprzedniego bloku $x_{i-1}[n]$ i uzupełnieniu tymi próbkami początku następnego bloku $x_i[n]$. Tym razem splot kołowy daje *K-1* początkowych próbek fałszywych (różniących się od wyniku splotu liniowego) i są one odrzucane z sygnału wyjściowego.

Szybki splot – obliczanie z dodawaniem zakładki (*overlap-add*)



Szybki splot

Liczba kolejnych bloków próbek sygnału wejściowego *x[n]* może być nieskończenie wielka. Ten sposób wprowadzania może być stosowany w procesorach sygnałowych przetwarzających sygnały cyfrowe w czasie rzeczywistym. Kolejne próbki ze źródła sygnału (mikrofon, kamera telewizyjna) są wprowadzane do odbiorczego bufora kołowego, a wyniki są wyprowadzane do nadawczego bufora kołowego.

Dziękuję za uwagę

KONIEC