

# Zastosowania Procesorów Sygnałowych

Adam Korzeniewski

adamkorz@sound.eti.pg.gda.pl

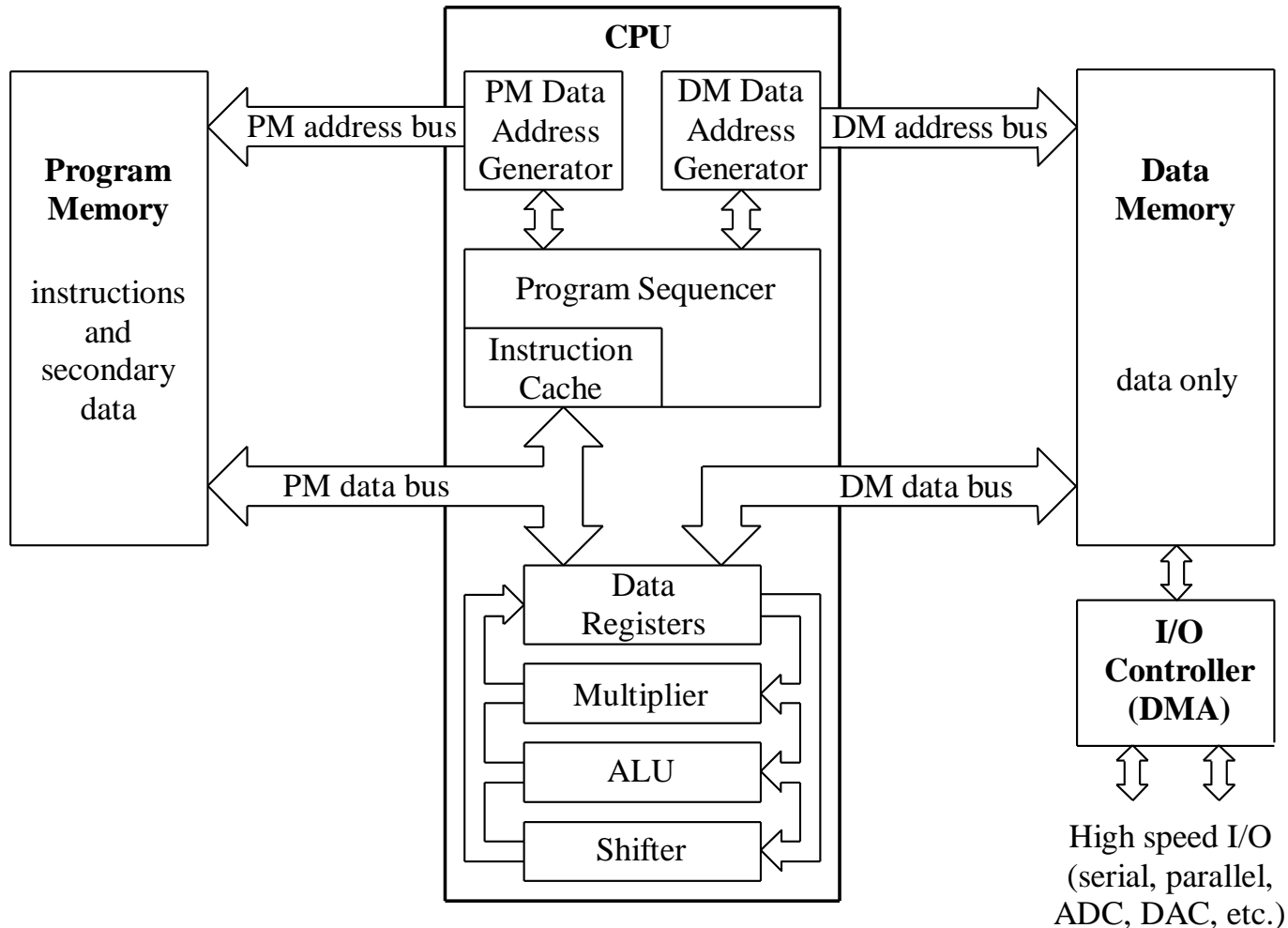
p. 732 - Katedra Systemów Multimedialnych

## Architektura typowego PS

## Systemy wieloprocessorowe

## Bufor kołowy

# Architektura typowego procesora sygnałowego



# Architektura typowego procesora sygnałowego

W zwykłym mikroprocesorze mnożenie dwóch liczb (dwóch operandów) odbywa się w sposób podobny do mnożenia ręcznego „w słupku”. Po pobraniu instrukcji mnożenia jest wywoływany mikro-kod wykonujący mnożenie jako sekwencję przesunięć i dodawań w kolejnych cyklach zegara. Będzie to wymagało np. aż ok. 80 cykli zegara do wymnożenia dwóch liczb 16-bitowych.

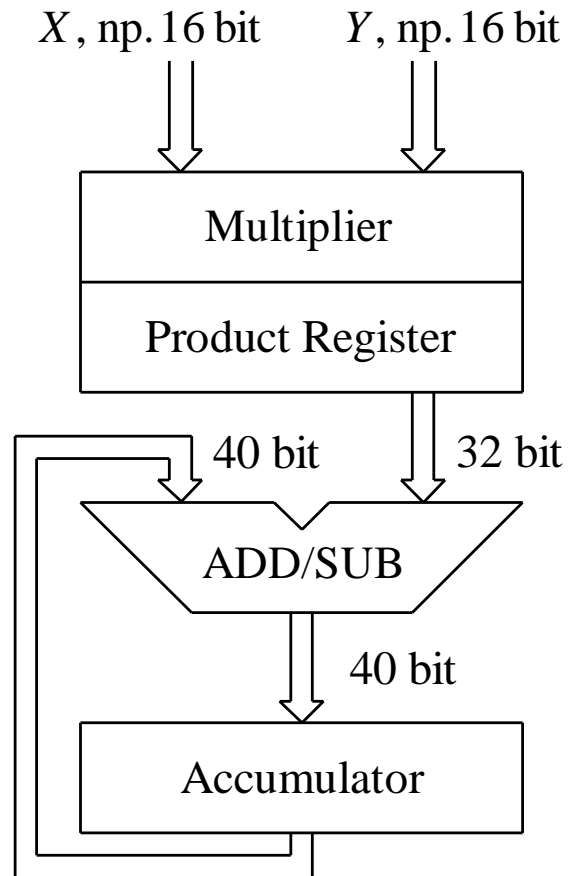
# Architektura typowego procesora sygnałowego

Cyfrowe przetwarzanie sygnałów zasadniczo sprowadza się do obliczania sum iloczynów (jest obliczana bieżąca suma, akumulata). Dlatego procesory sygnałowe zapewniają sprzętowo (hardwarowo) wykonanie operacji MAC (Multiply Accumulate operation) w jednym cyklu zegara. Dzięki temu przykładowo przy realizacji filtru FIR o 100 współczynnikach należy oczekiwać około 105-110 cykli zegara na czas między dwiema kolejnymi próbkami sygnału wejściowego. Operacja MAC jest przykładem operacji superskalarnej, gdzie wykonywane są dwie instrukcje równolegle. W informatyce mikroprocesor superskalarny, to mikroprocesor sprzętowo przystosowany do wykonania kilku instrukcji równolegle w jednym cyklu zegara.

# Architektura typowego procesora sygnałowego

MAC

$$A := X \cdot Y + A$$



# Miary prędkości procesorów

**MIPS** (Million Instructions Per Second).

Jest to liczba instrukcji na jedną sekundę. Prędkość wyrażona w tych jednostkach jest mierzona poprzez wykonanie wielu różnych programów testowych, zsumowanie liczby wykonanych instrukcji i podzielenie przez czas wykonania programów. Niestety, w imię dobrej reklamy produktu, podawane w danych katalogowych prędkości są zazwyczaj wartościami maksymalnymi. Na przykład dla procesora o częstotliwości zegara 200 MHz z pięcioma jednostkami operacyjnymi działającymi równolegle w jednym cyklu zegara, producent poda prędkość . W przeciętnym zastosowaniu procesora sygnałowego prędkość będzie pięciokrotnie mniejsza. Na przykład procesor Intel 8080 (1974 r., częstotliwość zegara 2 MHz) miał prędkość 0,5 MIPS. Procesor Intel Core i7 Extreme Edition 3960X (2011 r., częstotliwość zegara 3,33 GHz) ma prędkość 177 730 MIPS.

# Miary prędkości procesorów

**MFLOPS** (Million FLOating point operations Per Second). Jest to liczba operacji zmiennoprzecinkowych na jedną sekundę. Niestety pojęcie operacji zmiennoprzecinkowej nie jest jednoznaczne. Procesory sygnałowe są przystosowane do wykonania operacji MAC (Multiply-ACcumulate, obliczenie iloczynu i zsumowanie, podstawowa operacja przy obliczaniu akumulacji, obliczaniu sumy iloczynów) i dodatkowo „w locie” przesunięcie wyniku o zadaną liczbę bitów w lewo lub w prawo. Nie wiadomo, czy liczyć to jako jedną operację, czy trzy operacje. Ocenia się, że mózg ludzki pracuje z ogromną prędkością 10 petaFLOPS, , osiąganą tylko w superkomputerach.

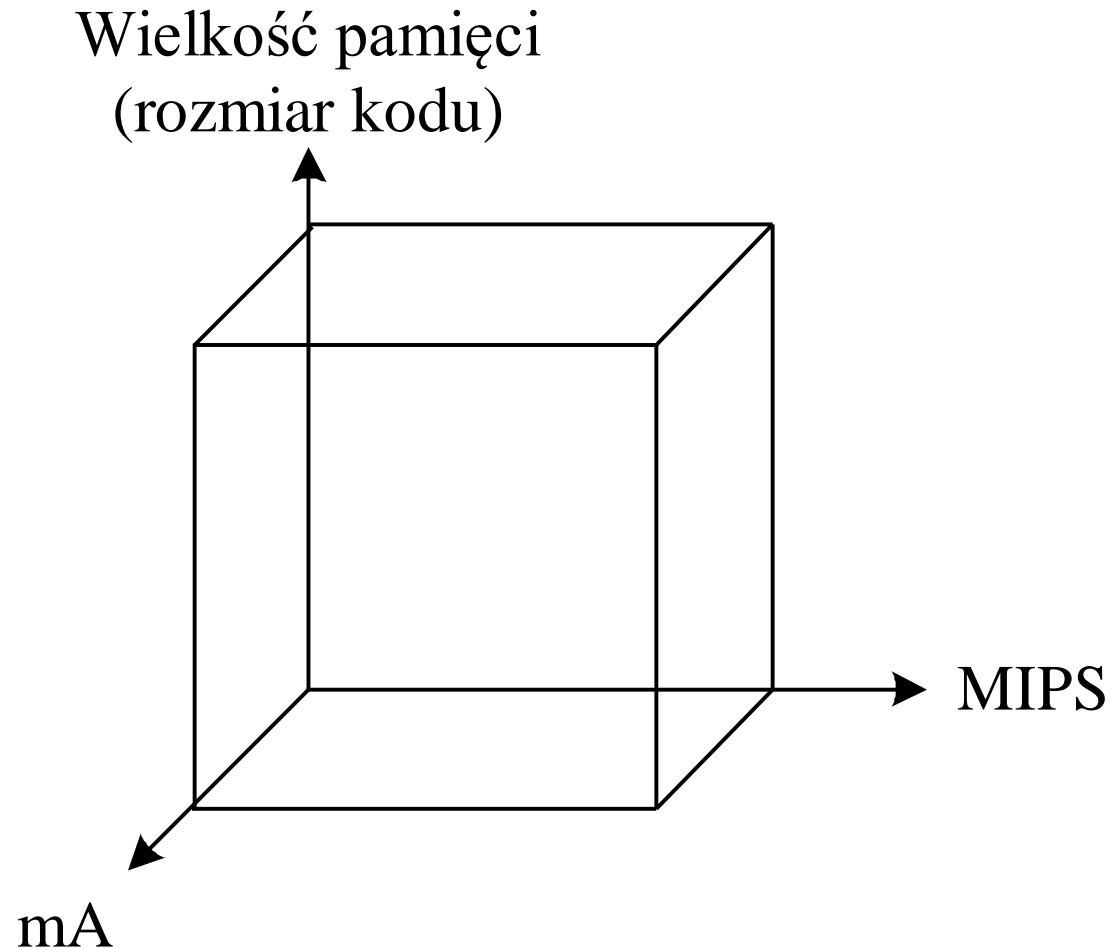
# Miary prędkości procesorów

**MMACS** (Million Multiply-ACcumulates per Second). Jest to liczba operacji MAC na jedną sekundę. Obliczana jako liczba jednostek MAC dzielona przez czas cyklu jednej instrukcji.

Procesor sygnałowy	Prędkość	Czas cyklu instrukcji (częstotliwość zegara)
Analog Devices ADSP-21065L SHARC	66 MIPS 198 MFLOPS 66 MMACS	15 ns (66 MHz)
Analog Devices ADSP-TS201S TigerSHARC	4.8 GIPS 3.6 GFLOPS 4.8 GMACS	1,67 ns (600 MHz)
Lucent Technologies DSP 16000	100 MIPS 200 MFLOPS 200 MMACS	10 ns (100 MHz)
Motorola DSP56302	66 MIPS 66 MFLOPS 66 MMACS	15 ns (66 MHz)

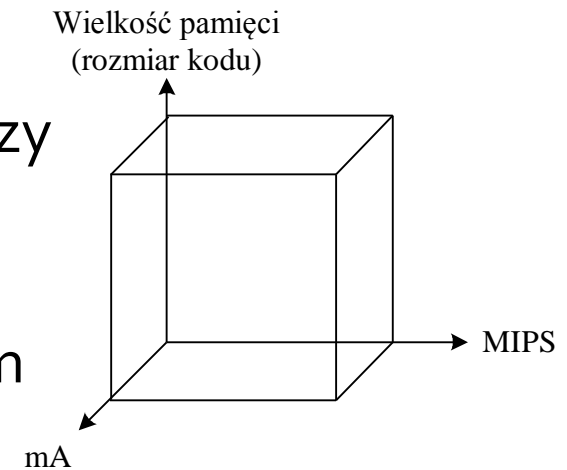


# Prostopadłościan zastosowań



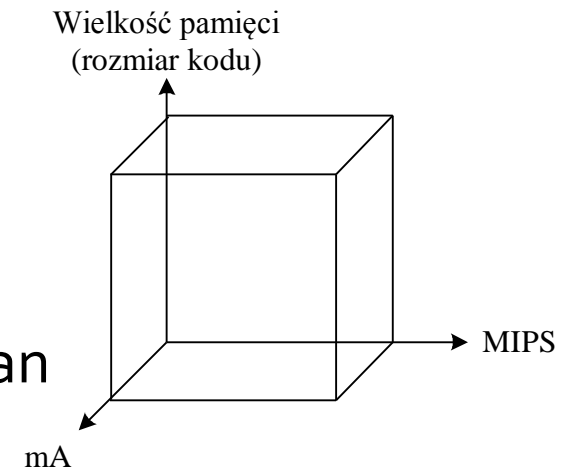
# Prostopadłościan zastosowań

Prędkość działania procesora jest bardzo ważnym parametrem, gdyż decyduje o tym, czy będziemy w stanie zrealizować przetwarzanie sygnału w czasie rzeczywistym. Drugim ważnym parametrem jest wielkość pamięci, im większa pamięć programu, tym większy program, o większej liczbie instrukcji, o większym rozmiarze kodu można zrealizować w procesorze sygnałowym. Trzecim ważnym parametrem jest pobór prądu przez procesor sygnałowy. Wyobrażamy sobie, iż te trzy parametry tworzą prostopadłościan w przestrzeni trzywymiarowej.



# Prostopadłościan zastosowań

Jesteśmy zainteresowani tym, aby prędkość i wielkość pamięci były jak największe przy jak najmniejszym poborze prądu, a więc chcemy, aby prostopadłościan przyjął postać plastra. Niestety są to wymagania sprzeczne, bez zmian technologii pobór prądu będzie z dobrym przybliżeniem wprost proporcjonalny do wzrostu prędkości i wielkości pamięci. Mały pobór prądu jest istotny zwłaszcza w urządzeniach przenośnych. Jednak nawet w urządzeniach stacjonarnych jesteśmy zainteresowani małym poborem prądu po to, aby ograniczyć wydzielanie ciepła, podwyższyć niezawodność.



# Systemy wieloprocessorowe

Istnieją tak złożone zadania obliczeniowe, że ich wykonanie w czasie rzeczywistym jest niemożliwe z użyciem pojedynczego procesora sygnałowego.

Przykładem są systemy przestrzennego zobrazowania w radarach i sonarach. W takim przypadku złożony algorytm (kompletne zadanie) należy rozbić na mniejsze podzadania i realizować je w procesorach sygnałowych połączonych w system wieloprocessorowy. Większość procesorów sygnałowych jest tak skonstruowanych, aby ułatwić ich łączenie w system wieloprocessorowy.

# Systemy wieloprocessorowe

Konstruując system wieloprocessorowy trzeba podjąć decyzję:

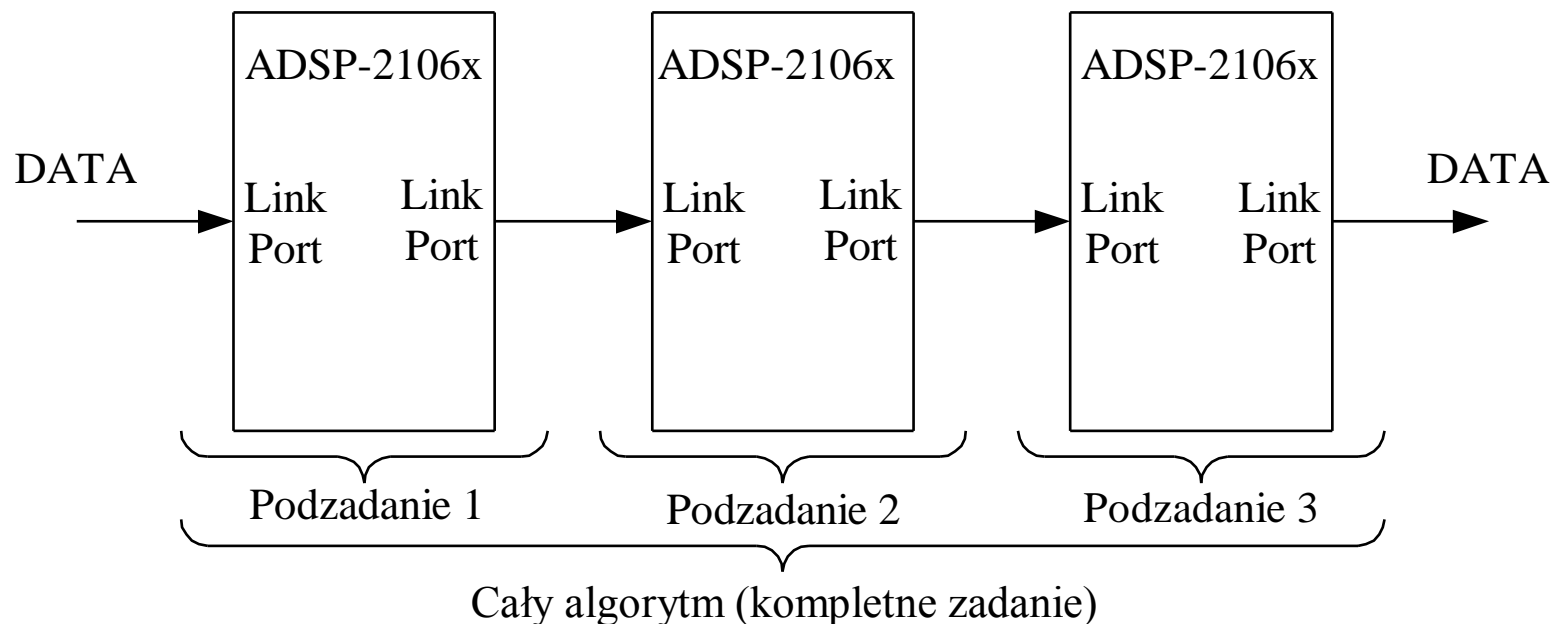
- ile procesorów sygnałowych zastosować
- jakie mają być parametry procesorów sygnałowych
- w jakiej konfiguracji połączyć procesory sygnałowe

# Systemy wieloprocessorowe

## Połączenie od punktu do punktu

(ang. *point-to-point communication, data flow multiprocessing*)

Procesory są połączone jeden za drugim jak na linii produkcyjnej. To połączenie jest też nazywane połączeniem potokowym. W tej konfiguracji pojedynczemu procesorowi sygnałowemu wystarcza dostęp do danych tylko od lewego sąsiada.

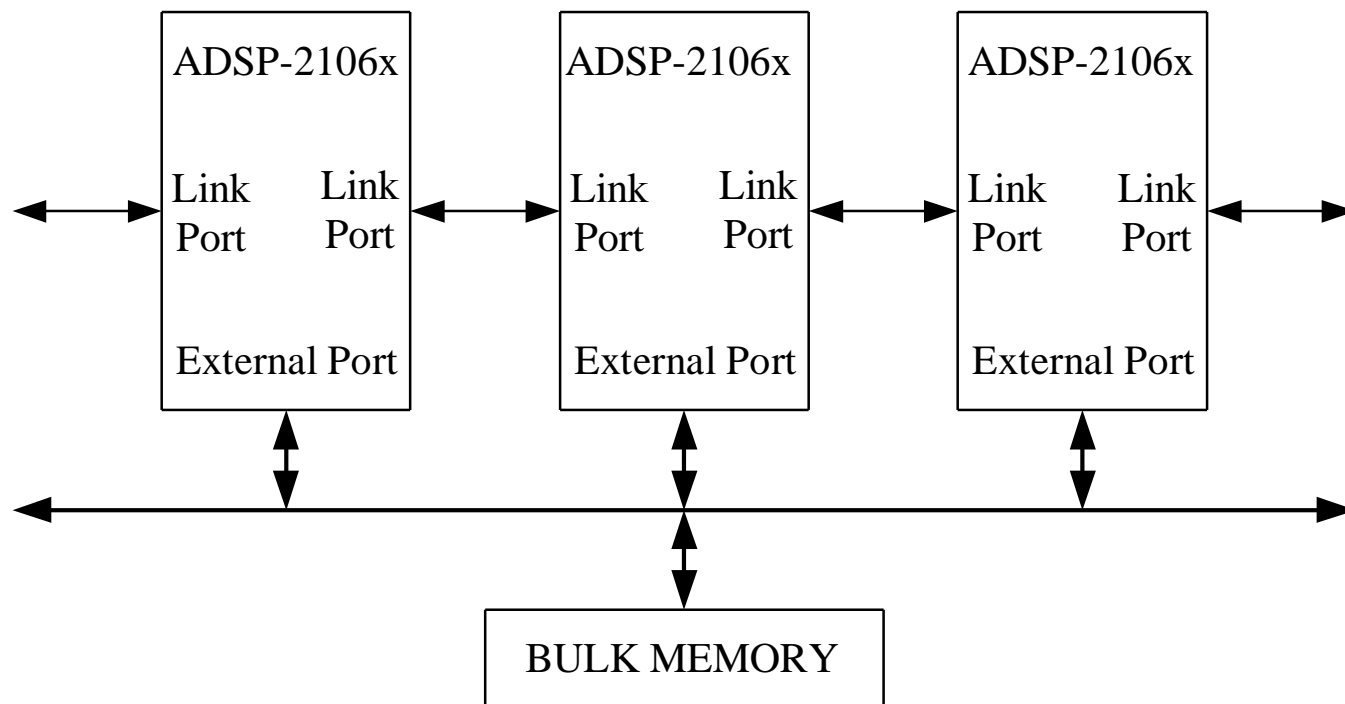


# Systemy wieloprocessorowe

## Połączenie klastrowe z dostępem do wspólnej globalnej pamięci

(ang. *global memory shared access communication, cluster multiprocessing*)

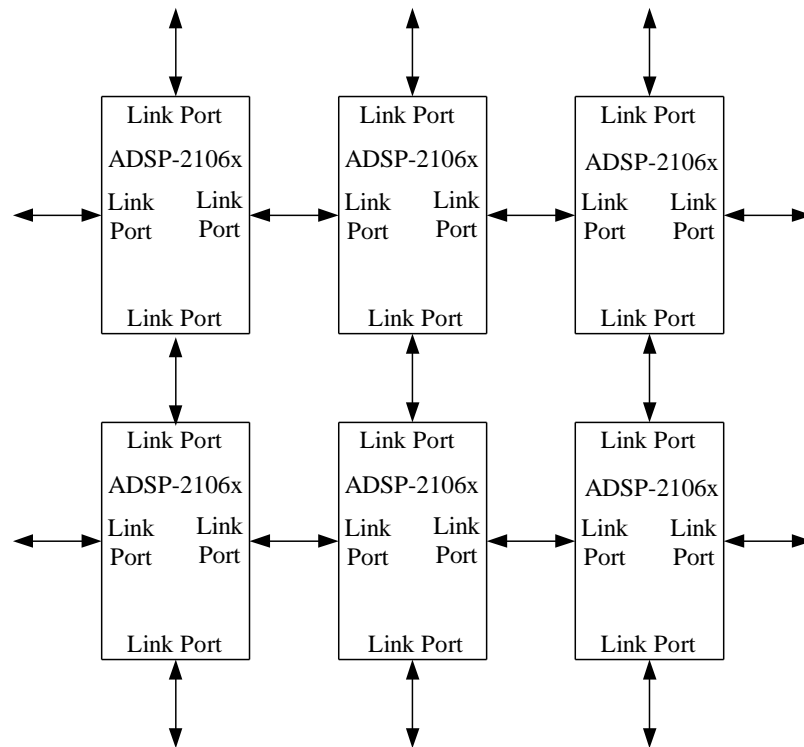
Procesory współpracują ze sobą poprzez zewnętrzne porty za pośrednictwem wspólnej równoległej szyny zapewniającej dostęp do globalnej pamięci masowej. W danej chwili czasu tylko jeden procesor ma dostęp do globalnej pamięci.



# Systemy wieloprocessorowe

## Połączenie oczkowe (ang. *mesh communication*)

To połączenie znajduje zastosowanie w przestrzennym zobrazowaniu 2D (jak na poniższym rysunku) lub 3D

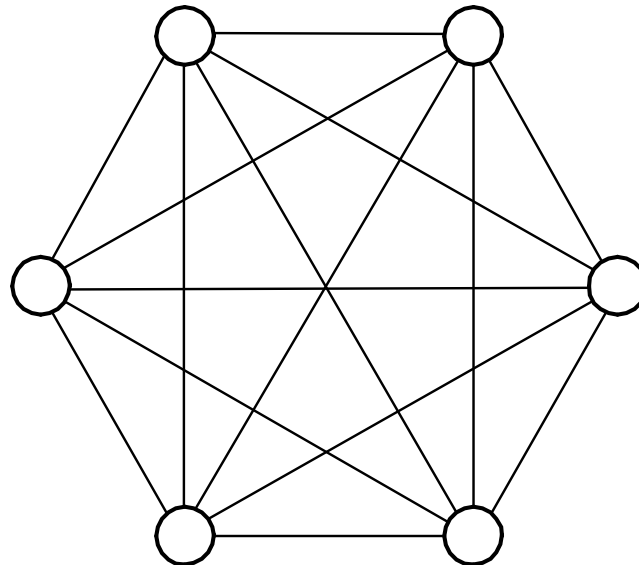




# Systemy wieloprocessorowe

**Połączenie kępowe** (ang. *BusCluster modules communication*)

Jest to system wieloprocessorowy wyższego rzędu. Każda kępa (ang. *BusCluster module*) jest systemem wieloprocessorowym, klastrem. Kępy (węzły) są łączone w dowolnej konfiguracji w system wyższego rzędu.



# Bufory liniowe i kołowe

Filtr FIR jest systemem o transmitancji

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

realizującym algorytm opisany następującym równaniem różnicowym

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

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

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

Przy zadanym przyczynowym, nieskończonym sygnale wejściowym

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

# Bufory liniowe i kołowe

rekurencyjne obliczanie próbek sygnału wyjściowego

$$y[0] = b_0 x[0] + b_1 \cdot 0 + b_2 \cdot 0 + \dots + b_M \cdot 0$$

$$y[1] = b_0 x[1] + b_1 x[0] + b_2 \cdot 0 + \dots + b_M \cdot 0$$

$$y[2] = b_0 x[2] + b_1 x[1] + b_2 x[0] + \dots + b_M \cdot 0$$

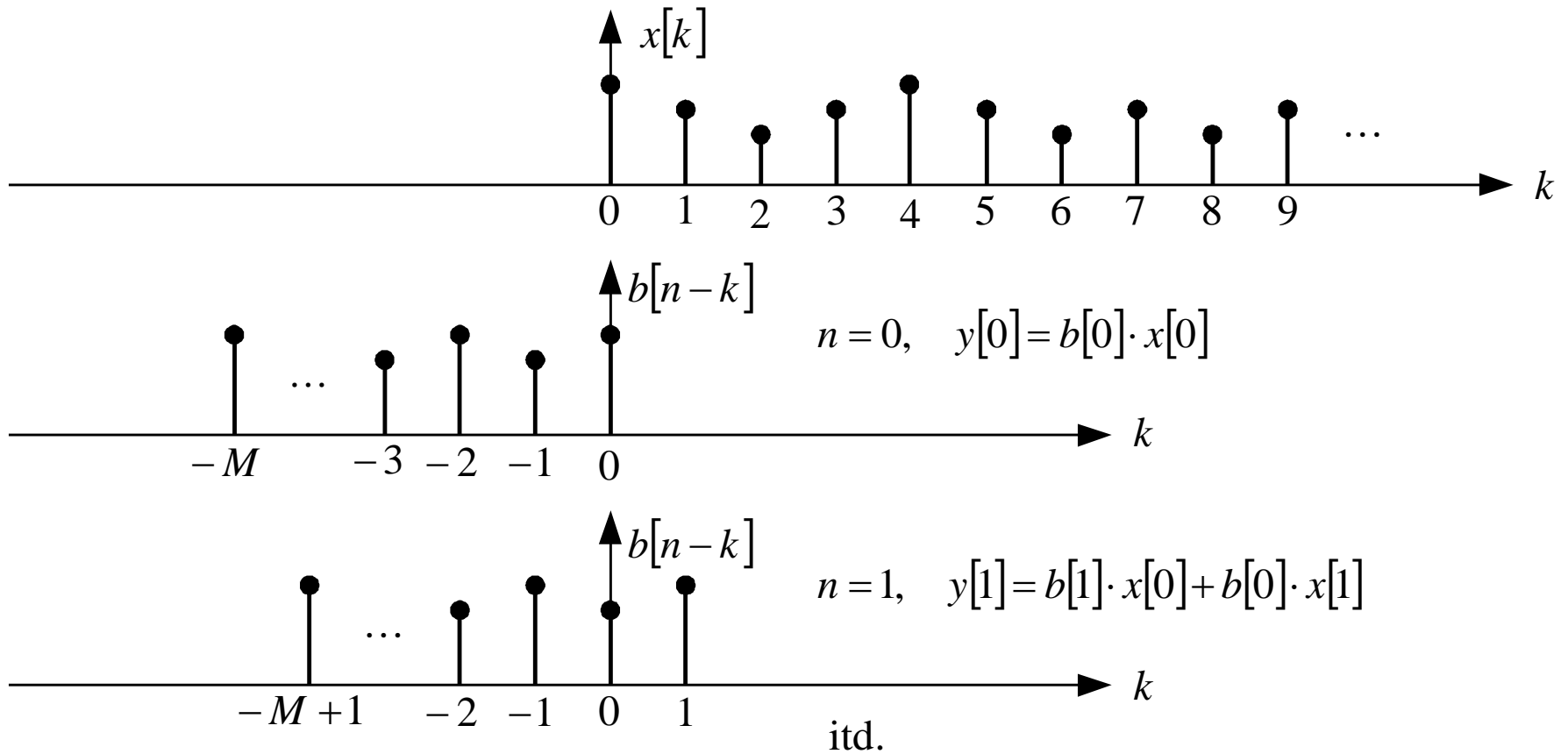
$$y[3] = b_0 x[3] + b_1 x[2] + b_2 x[1] + \dots + b_M \cdot 0$$

⋮

jest niczym innym jak obliczeniem splotu liniowego (sumy iloczynów)

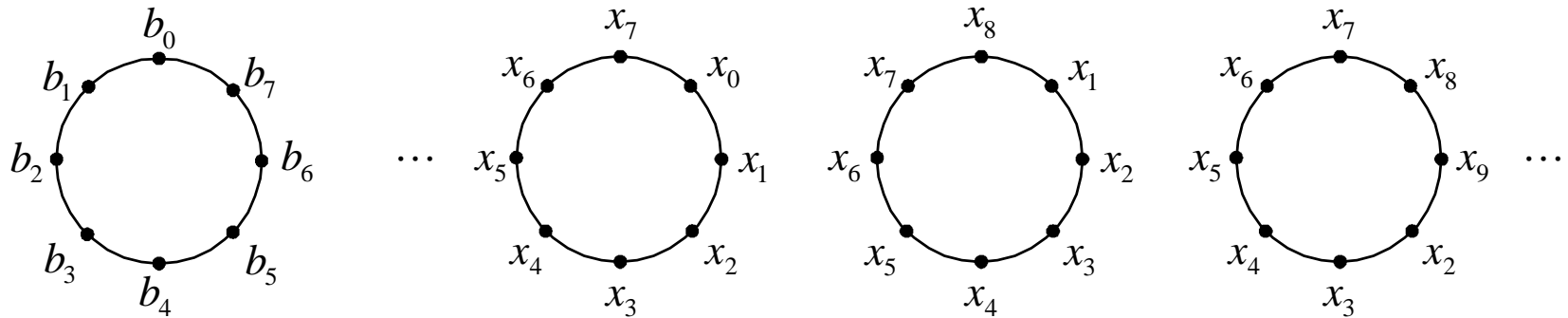
$$y[n] = b[n] * x[n] = \sum_{k=0}^M b[n-k] x[k]$$

# Bufory liniowe i kołowe



# Bufory liniowe i kołowe

$$M = 7$$

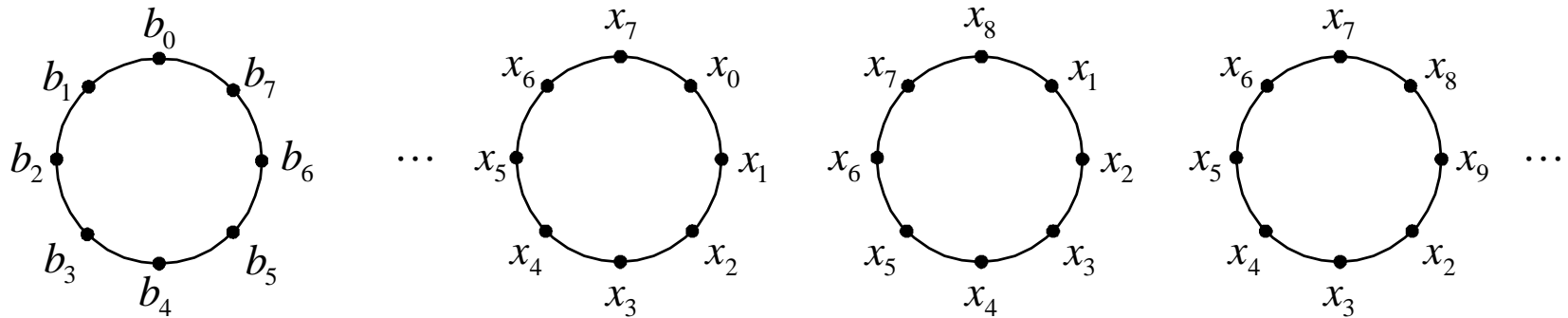


Wykonanie obliczeń splotu liniowego w procesorze sygnałowym wymaga wykonania następujących czynności:

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

# Bufory liniowe i kołowe

$$M = 7$$

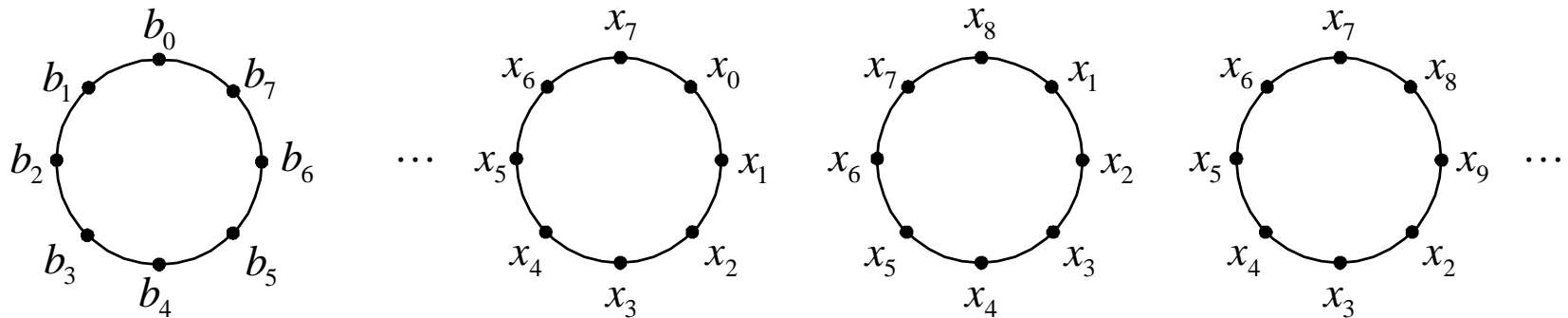


3. Oblicz próbkę sygnału wyjściowego
- $$y[n] = b[n] * x[n] = \sum_{k=0}^M b[n-k]x[k]$$

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

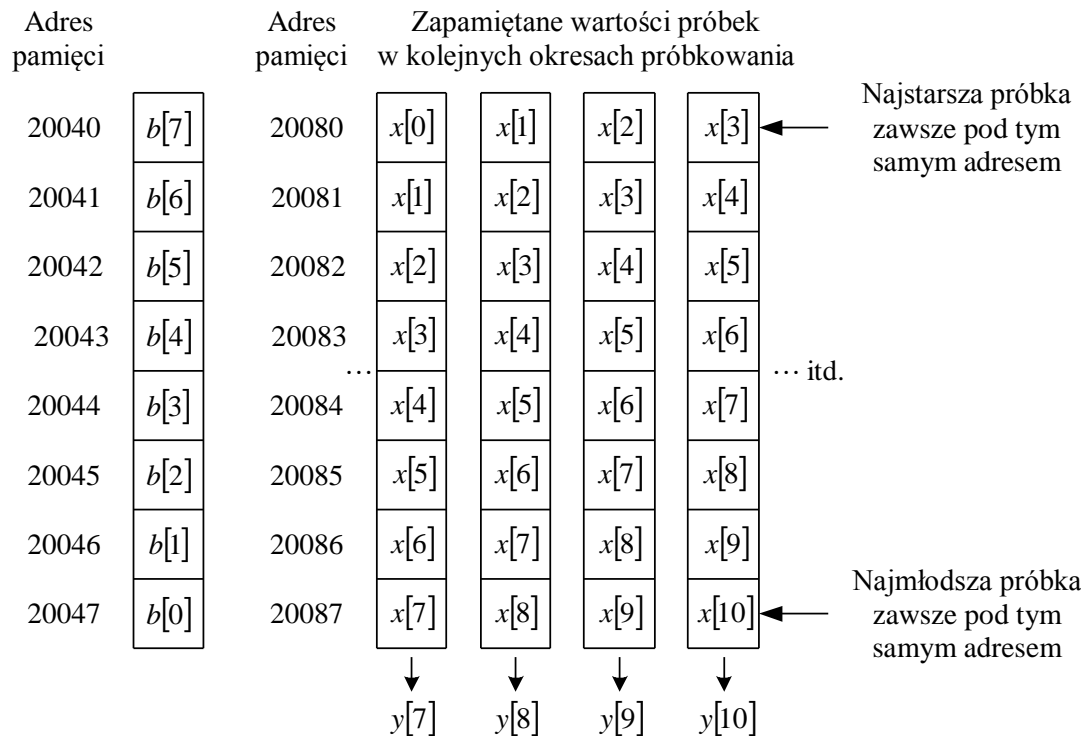
# Bufor liniowy - Filtr FIR

$$M = 7$$



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

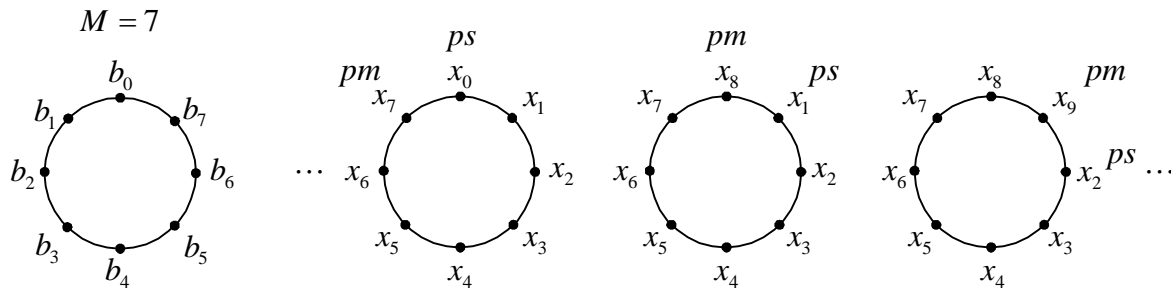
# Bufory liniowe i kołowe



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



Adres  
pamięci

20040	$b[7]$
20041	$b[6]$
20042	$b[5]$
20043	$b[4]$
20044	$b[3]$
20045	$b[2]$
20046	$b[1]$
20047	$b[0]$

Adres  
pamięci

20080	$x[0]$
20081	$x[1]$
20082	$x[2]$
20083	$x[3]$
...	...
20084	$x[4]$
20085	$x[5]$
20086	$x[6]$
20087	$x[7]$

↓  
 $y[7]$

Zapamiętane wartości próbek  
w kolejnych okresach próbkowania

$x[8]$
$x[1]$
$x[2]$
$x[3]$
$x[4]$
$x[5]$
$x[6]$
$x[7]$

↓  
 $y[8]$

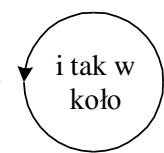
$x[8]$
$x[9]$
$x[2]$
$x[3]$
$x[4]$
$x[5]$
$x[6]$
$x[7]$

↓  
 $y[9]$

$x[8]$
$x[9]$
$x[10]$
$x[3]$
$x[4]$
$x[5]$
$x[6]$
$x[7]$

↓  
 $y[10]$

$ps$  – wsk. najst. pr.  
 $pm$  – wsk. najmł. pr.



# Bufor kołowy

W przypadku filtru IIR mamy do czynienia z systemem o transmitancji

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}}$$

realizującym algorytm opisany następującym równaniem różnicowym

$$y[n] = b_0 x[n] + b_1 x[n-1] + \dots + b_M x[n-M] - a_1 y[n-1] - a_2 y[n-2] + \dots - a_N y[n-N]$$

które jest rozwiązywane rekursywnie przy zadanych warunkach początkowych

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

$$y[0] = b_0 x[0] + b_1 \cdot 0 + \dots + b_M \cdot 0 - a_1 y[-1] - a_2 y[-2] + \dots - a_N y[-N]$$

$$y[1] = b_0 x[1] + b_1 x[0] + \dots + b_M x[1-M] - a_1 y[0] - a_2 y[-1] + \dots - a_N y[1-N]$$

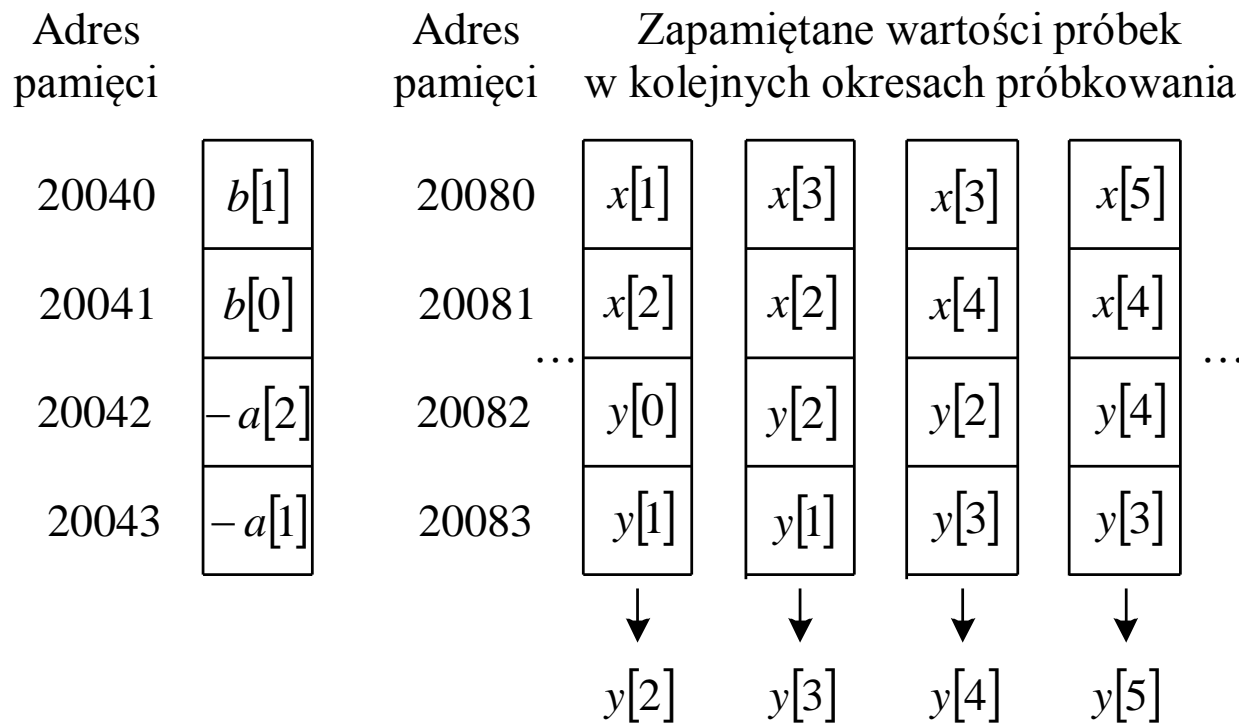
$$y[2] = b_0 x[2] + b_1 x[1] + \dots + b_M x[2-M] - a_1 y[1] - a_2 y[0] + \dots - a_N y[2-N]$$

$$y[3] = b_0 x[3] + b_1 x[2] + \dots + b_M x[3-M] - a_1 y[2] - a_2 y[1] + \dots - a_N y[3-N]$$

⋮

# 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.



**Dziękuję za uwagę**

---

**KONIEC**