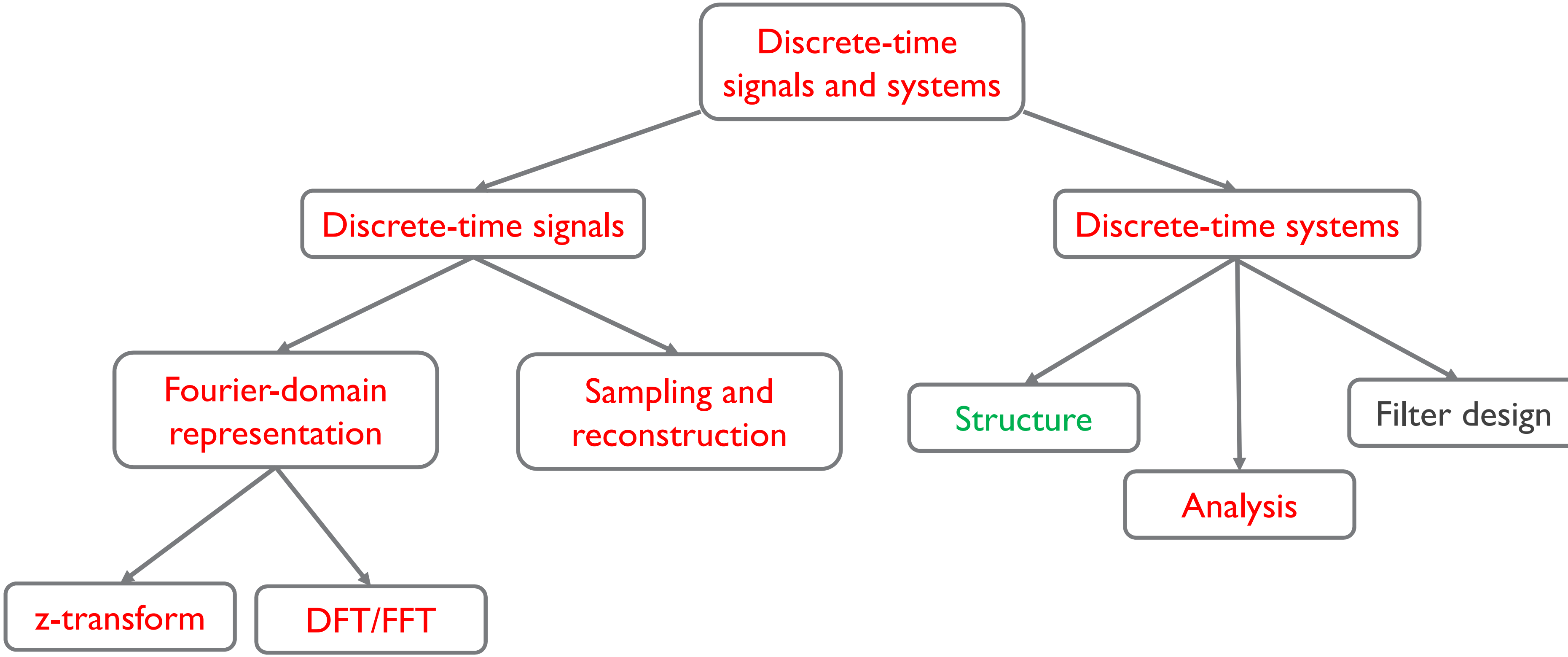
Digital Signal Processing

POSTECH
Department of Electrical Engineering
Junil Choi

Course at glance



Structures for Discrete-Time Systems

System implementation

- Consider LTI system with rational function

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 - az^{-1}}, \quad |z| > |a|$$

- Impulse response

$$h[n] = b_0 a^n u[n] + b_1 a^{n-1} u[n-1]$$

- Linear constant-coefficient difference equation (with initial rest)

$$y[n] - ay[n-1] = b_0 x[n] + b_1 x[n-1]$$

- They are three equivalent representations
- How to implement this system?

Block Diagram Representation

System implementation concept

- Hard to implement the system using impulse response

$$h[n] = b_0 a^n u[n] + b_1 a^{n-1} u[n-1]$$

Infinite duration

- Linear constant-coefficient difference equation

$$y[n] - ay[n-1] = b_0 x[n] + b_1 x[n-1]$$

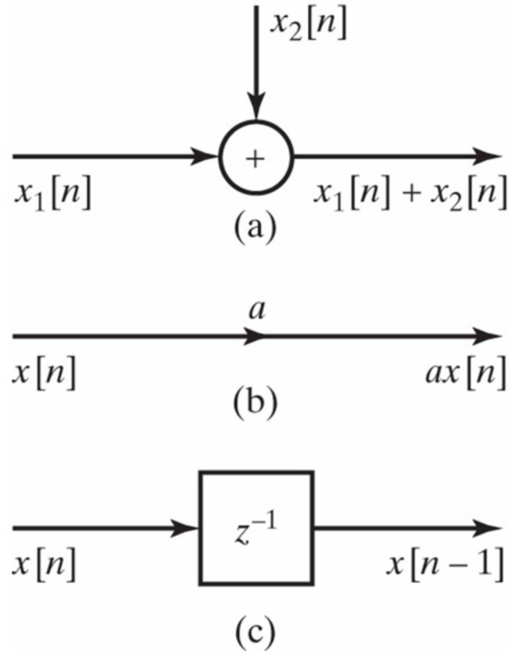
$$y[n] = ay[n-1] + b_0 x[n] + b_1 x[n-1]$$

provides the basis for an algorithm for recursive computation of the output at any time *n*

Basic elements for implementation

- Consider $y[n] = ay[n-1] + b_0 x[n] + b_1 x[n-1]$

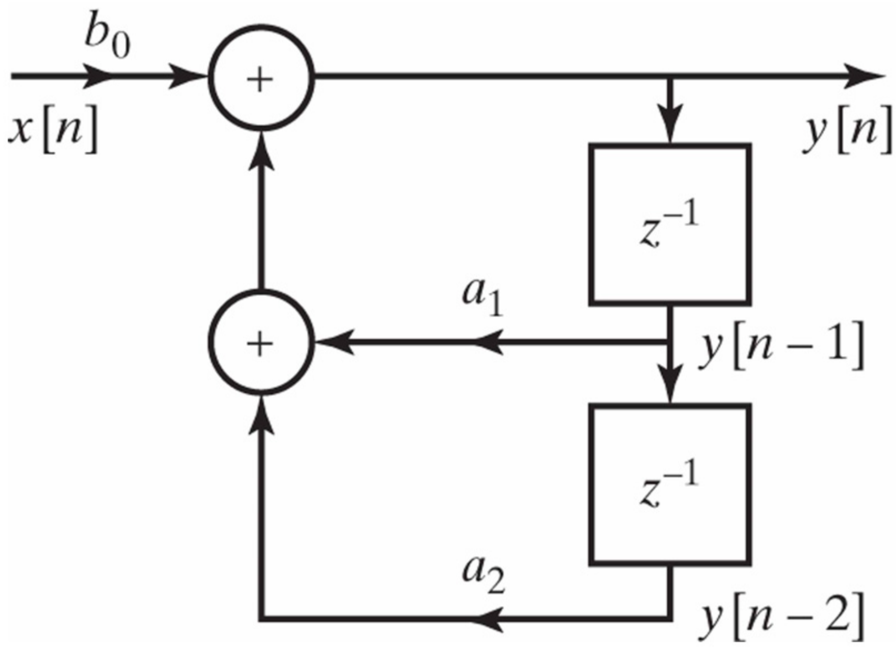
- It needs
  - Adders
  - Multipliers
  - Unit delays



Block diagram example

- Consider $y[n] = a_1 y[n-1] + a_2 y[n-2] + b_0 x[n]$

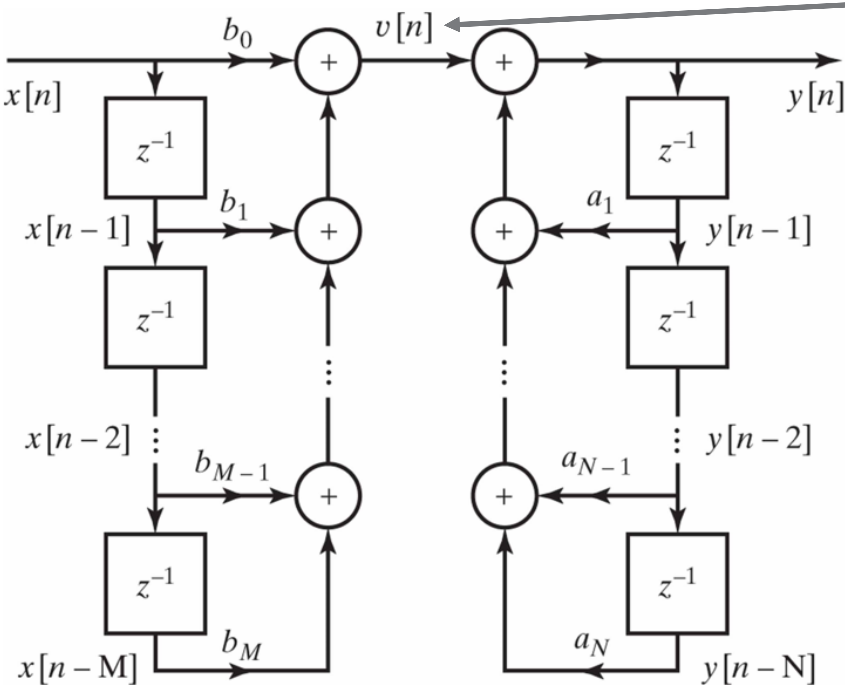
- The system can be implemented as



General higher-order difference equations

- Consider $y[n] = \sum_{k=1}^{N} a_k y[n-k] + \underbrace{\sum_{k=0}^{M} b_k x[n-k]}_{v[n]}$

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}}$$



A cascade of two LTI systems

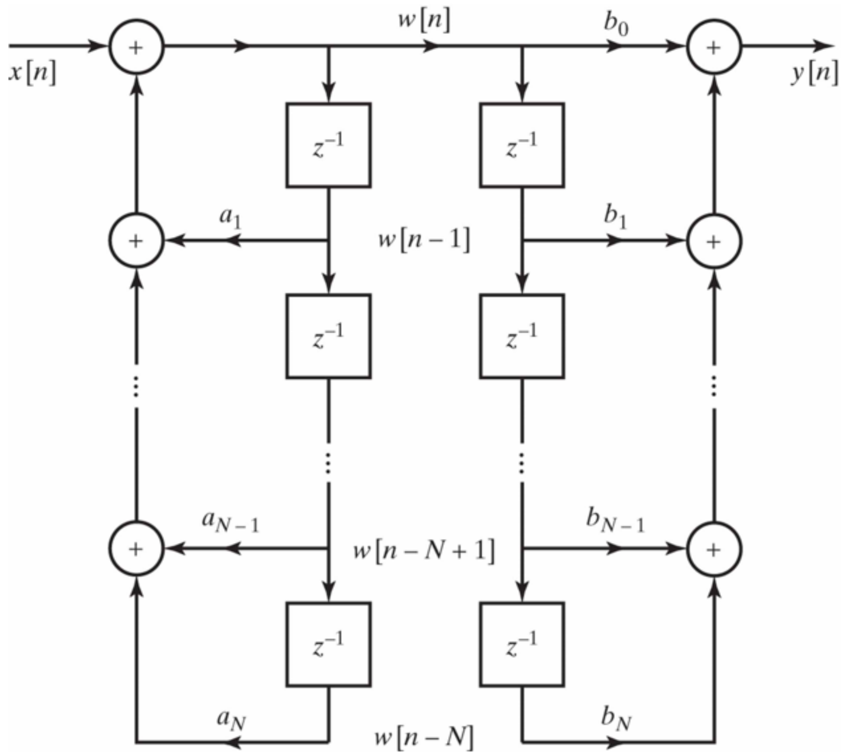
x[n]➔v[n], v[n]➔y[n]

$$v[n] = \sum_{k=0}^{M} b_k x[n-k]$$

$$y[n] = \sum_{k=1}^{N} a_k y[n-k] + v[n]$$

Rearrangement of block diagram

- Since convolution is commutative, the order of two LTI systems can be reversed while having the same output

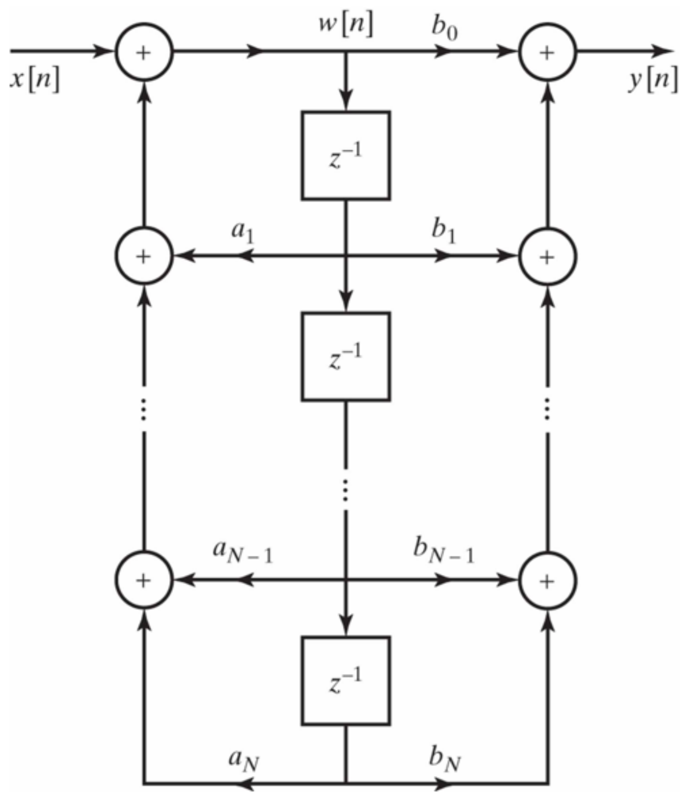


$$w[n] = \sum_{k=1}^{N} a_k w[n-k] + x[n]$$

$$y[n] = \sum_{k=0}^{M} b_k w[n-k]$$

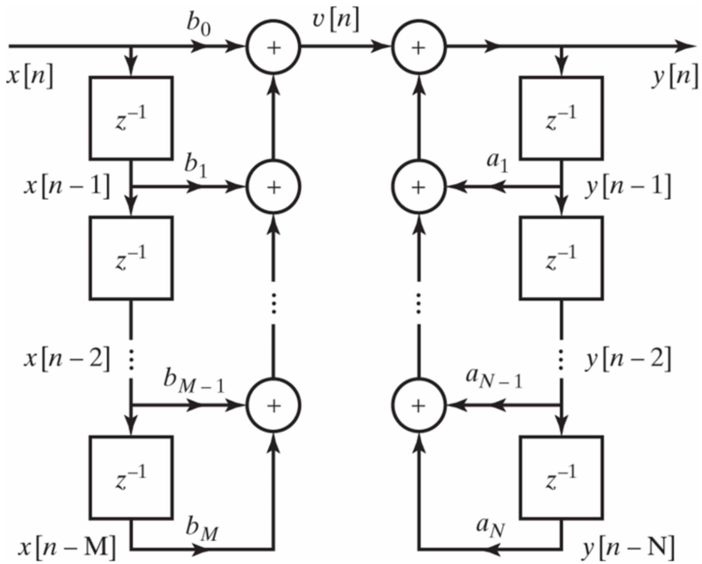
Minimum delay implementation

- The minimum number of delay elements: $\max(N, M)$

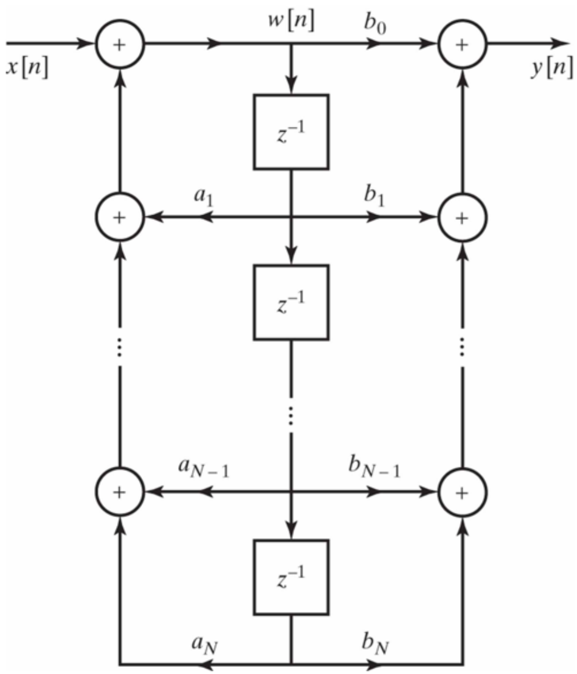


Direct forms I and II

- Direct form I
  - Direct realization of difference equation



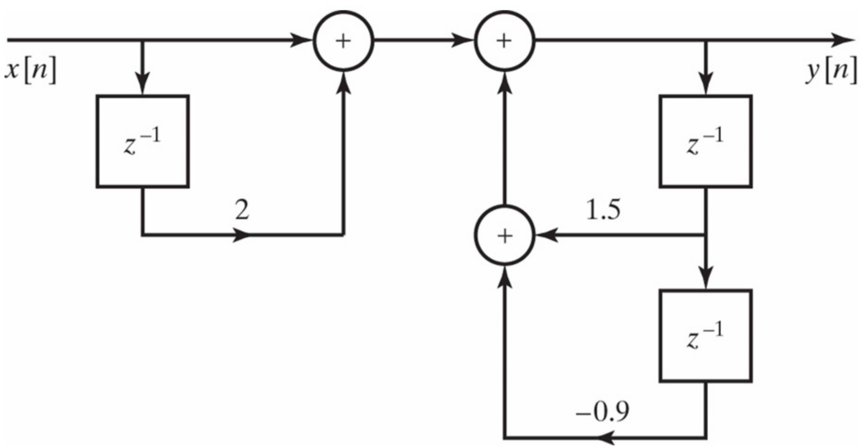
- Direct form II (canonic form)
  - With minimum number of delays



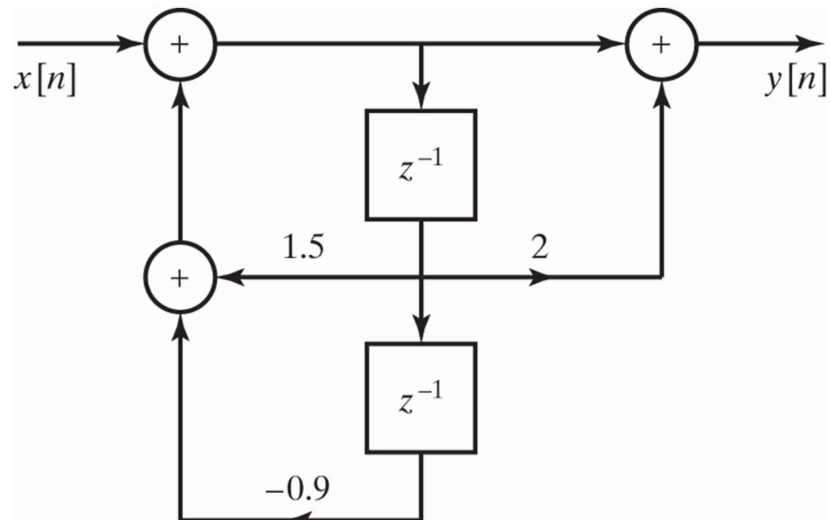
Example

- Consider $H(z) = \dfrac{1 + 2z^{-1}}{1 - 1.5z^{-1} + 0.9z^{-2}}$

$$y[n] = 1.5y[n-1] - 0.9y[n-2] + x[n] + 2x[n-1]$$



Direct form I

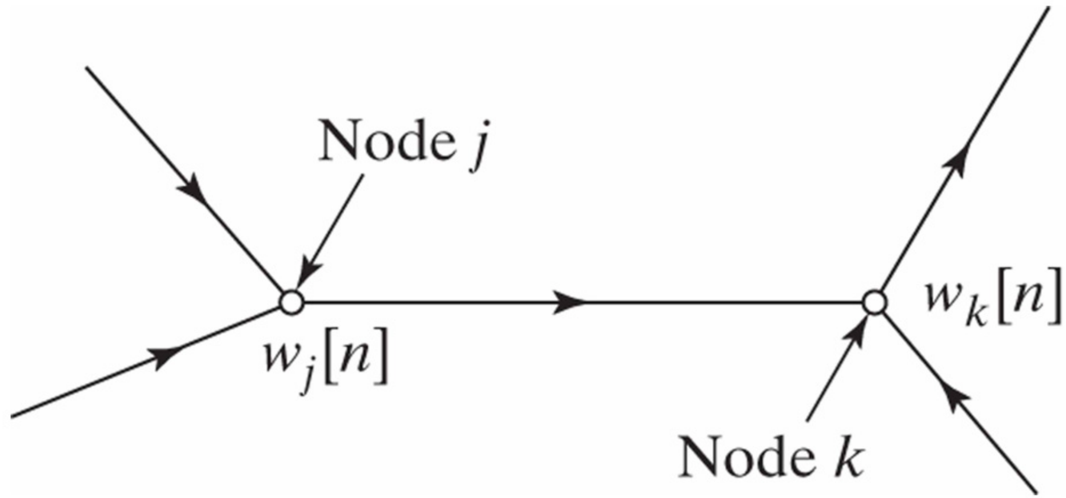


Direct form II

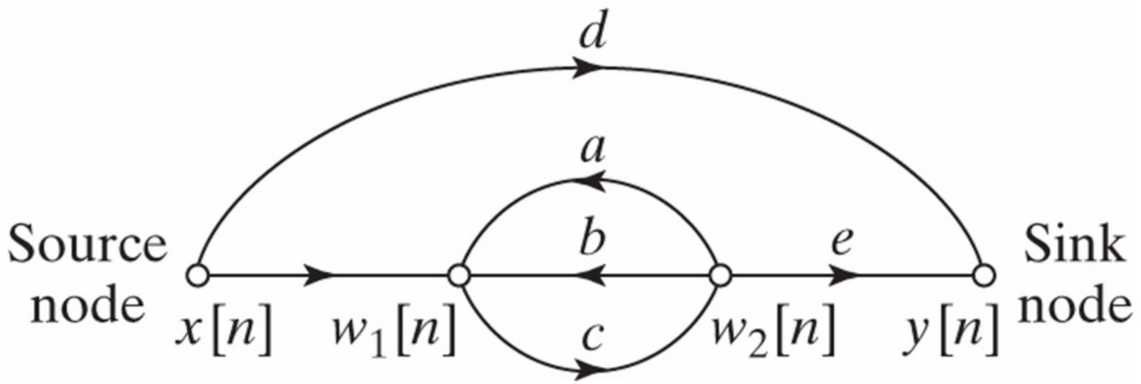
Signal Flow Graph Representation

Signal flow graph

- Essentially the same as block diagram representation
  - Exist a few notational differences
  - Represent a network with nodes and branches



Example of signal flow graph

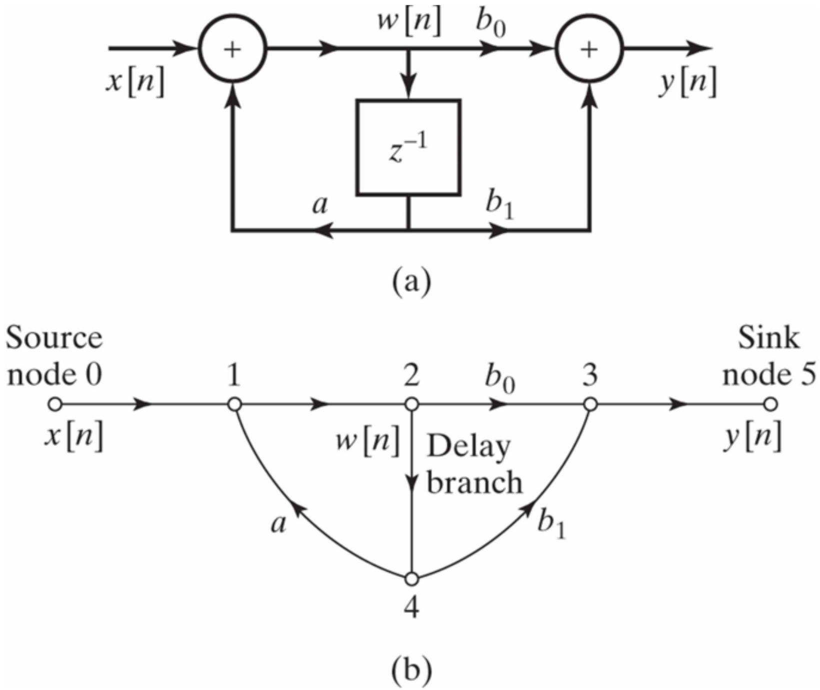


- Note $w_1[n] = x[n] + aw_2[n] + bw_2[n]$

$$w_2[n] = cw_1[n]$$

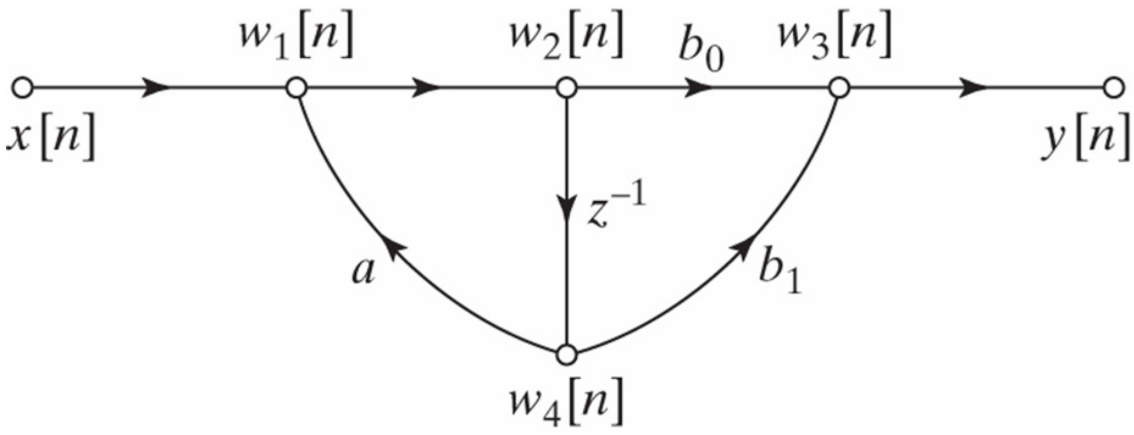
$$y[n] = dx[n] + ew_2[n]$$

Block vs. signal flow graph representation



- Nodes in signal flow graph represent both branching points and adders
- In the block diagram a special symbol is used for adders and a node has only one incoming branch.

Actual signal flow graph representation



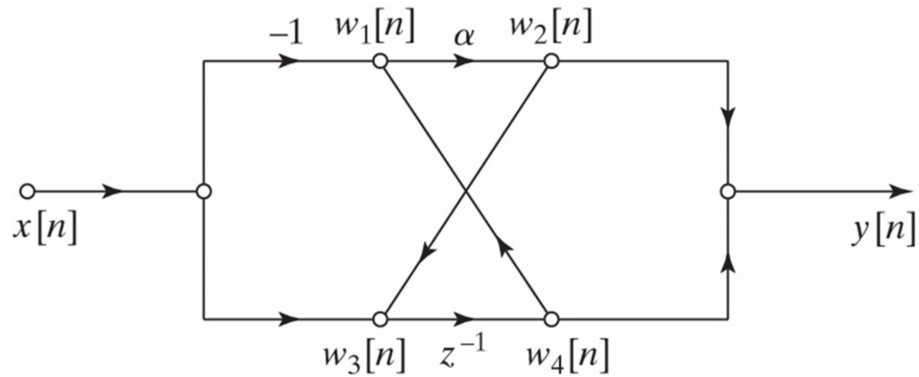
$$w_1[n] = aw_4[n] + x[n]$$
$$w_2[n] = w_1[n]$$
$$w_3[n] = b_0 w_2[n] + b_1 w_4[n]$$
$$w_4[n] = w_2[n-1]$$
$$y[n] = w_3[n]$$

$$\Rightarrow$$

$$w_2[n] = aw_2[n-1] + x[n]$$
$$y[n] = b_0 w_2[n] + b_1 w_2[n-1]$$

Signal flow graph with z-transformation

- Consider the graph
  - Not a direct form
  - Cannot obtain H(z) by inspection
  - How to obtain H(z)?



- Each node is $w_1[n] = w_4[n] - x[n]$

$$w_2[n] = \alpha w_1[n]$$
$$w_3[n] = w_2[n] + x[n]$$
$$w_4[n] = w_3[n-1]$$
$$y[n] = w_2[n] + w_4[n]$$

Difficult to solve due to delay
Use z-transform!

Signal flow graph with z-transformation

- z-transform equations

$$
\begin{aligned}
W_1(z) &= W_4(z) - X(z) \\
W_2(z) &= \alpha W_1(z) \\
W_3(z) &= W_2(z) + X(z) \\
W_4(z) &= z^{-1} W_3(z) \\
Y(z) &= W_2(z) + W_4(z)
\end{aligned}
$$

- After removing some variables

$$
\begin{aligned}
W_2(z) &= \alpha(W_4(z) - X(z)) \\
W_4(z) &= z^{-1}(W_2(z) + X(z)) \\
Y(z) &= W_2(z) + W_4(z)
\end{aligned}
\quad\Rightarrow\quad
\begin{aligned}
W_2(z) &= \frac{\alpha(z^{-1} - 1)}{1 - \alpha z^{-1}} X(z) \\
W_4(z) &= \frac{z^{-1}(1 - \alpha)}{1 - \alpha z^{-1}} X(z)
\end{aligned}
$$

Signal flow graph with z-transformation

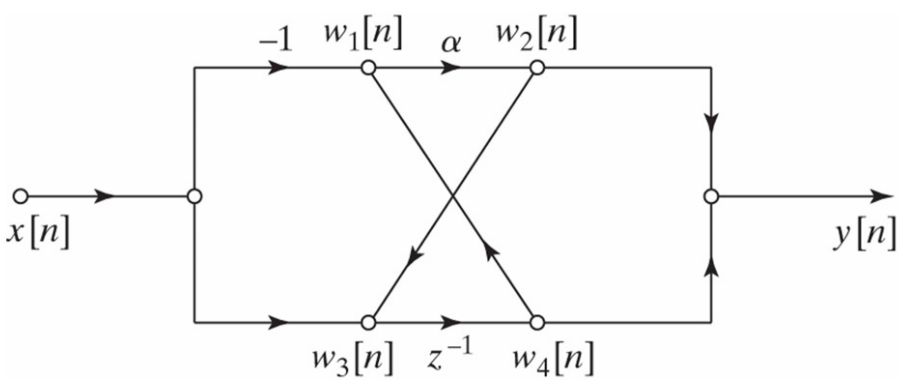
- Output becomes $Y(z) = \left(\dfrac{z^{-1} - \alpha}{1 - \alpha z^{-1}}\right) X(z)$

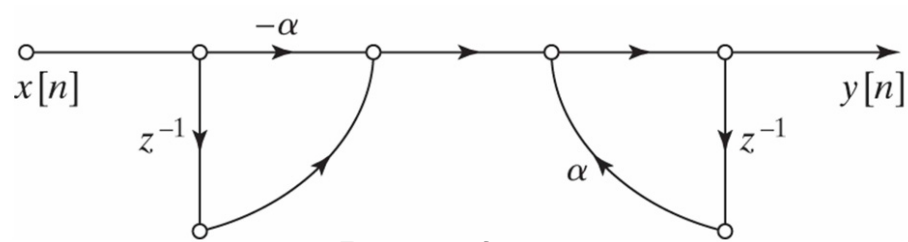
  All-pass system with real α

- System function and corresponding impulse response

$$H(z) = \left(\frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}\right), \quad h[n] = \alpha^{n-1}u[n-1] - \alpha^{n+1}u[n]$$

- Comparing two representations: requires different computational resources





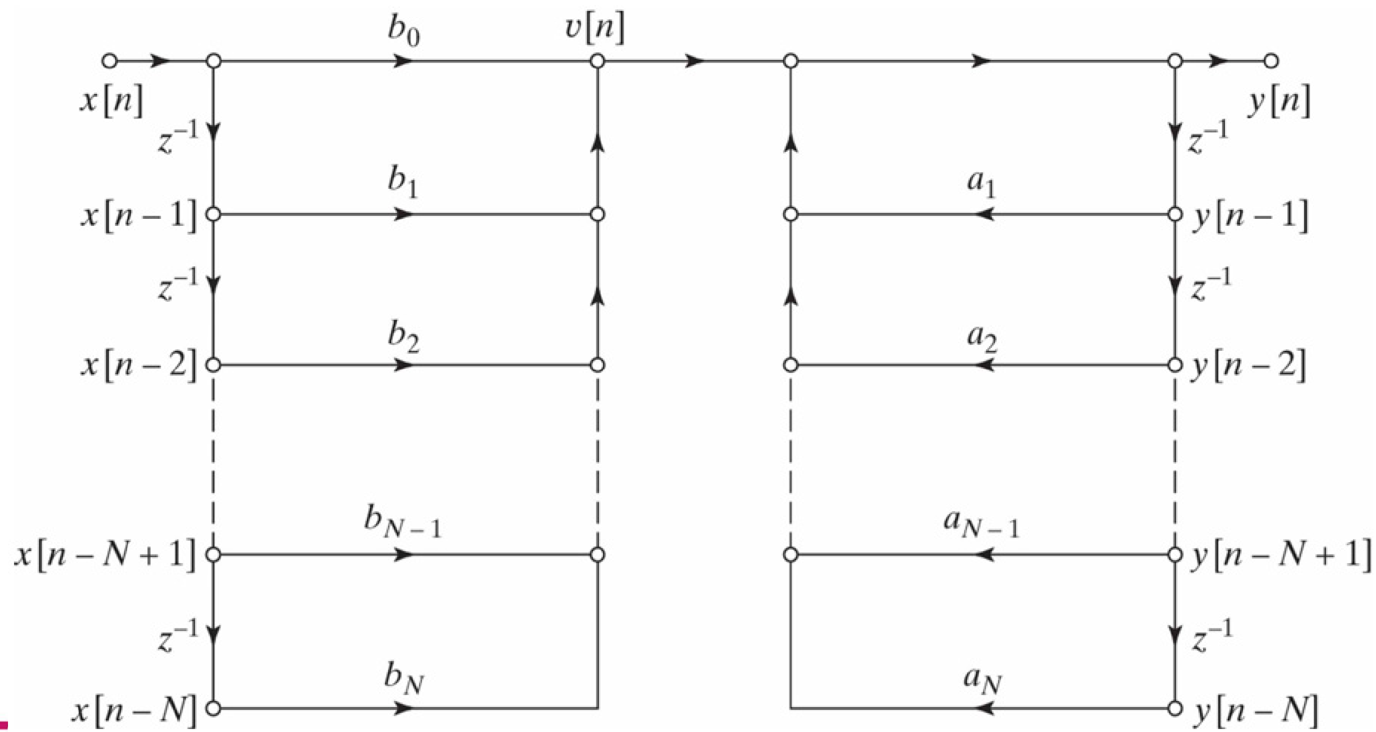
Direct form I

Basic structure for IIR systems

- Similar to block-diagram representation, there can be various ways to represent a system using signal flow graph
  - Direct form I
  - Direct form II (canonic direct form)
  - Cascade form
  - Parallel form

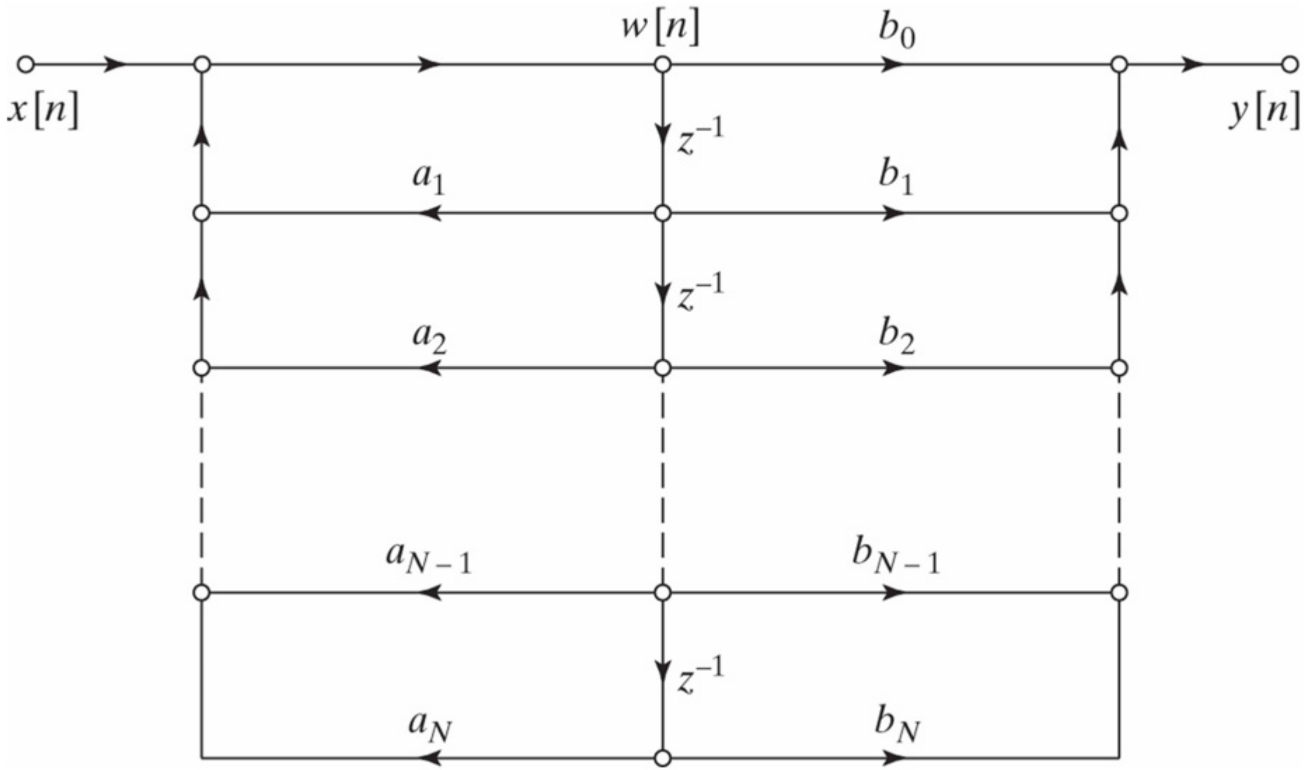
Direct form I

- Consider $y[n] = \sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k]$



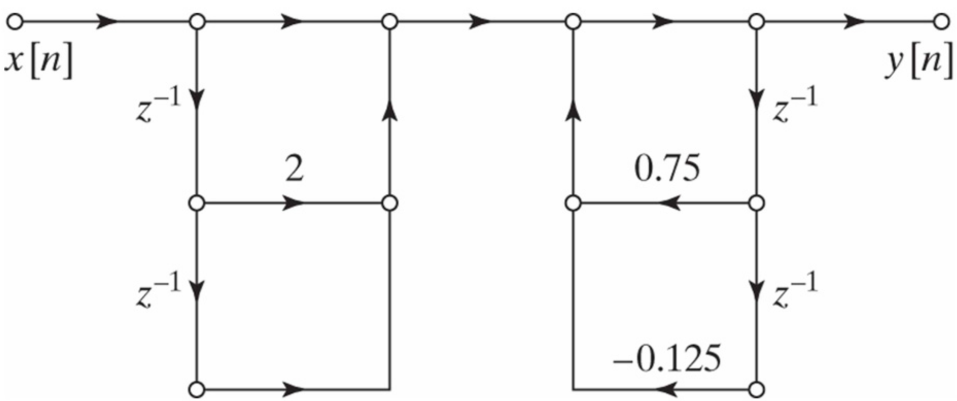
Direct form II

- Consider $y[n] = \sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k]$

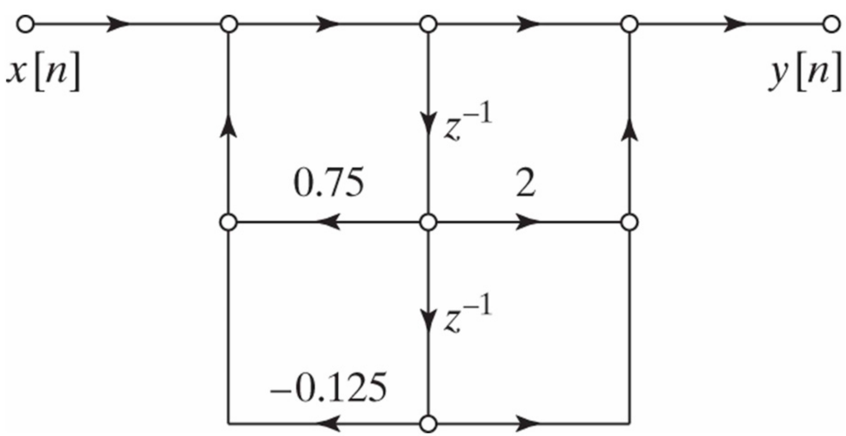


Direct forms example

- Consider $H(z) = \dfrac{1 + 2z^{-1} + z^{-2}}{1 - 0.75z^{-1} + 0.125z^{-2}}$



Direct form I



Direct form II

Cascade form

- Note $H(z) = \dfrac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}}$

- Consider the most general factorization when all coefficients are real

$$H(z) = A \frac{\underbrace{\prod_{k=1}^{M_1}(1 - f_k z^{-1})}_{} \underbrace{\prod_{k=1}^{M_2}(1 - g_k z^{-1})(1 - g_k^* z^{-1})}_{}}{\underbrace{\prod_{k=1}^{N_1}(1 - c_k z^{-1})}_{\text{Real poles and zeros}} \underbrace{\prod_{k=1}^{N_2}(1 - d_k z^{-1})(1 - d_k^* z^{-1})}_{\text{Conjugate pairs of poles and zeros}}}$$

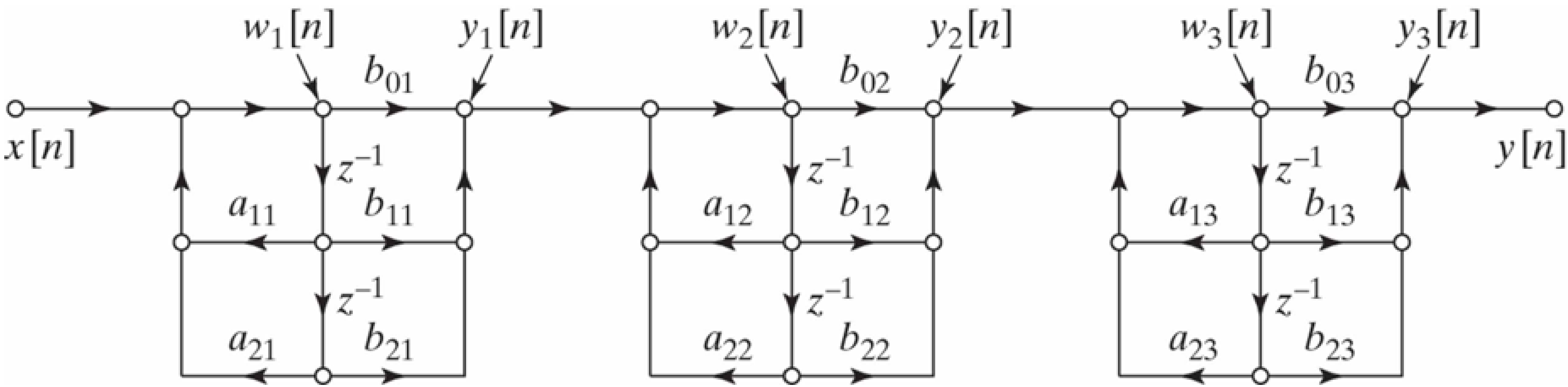
- Combine pairs of real factors and complex conjugate pairs into 2nd-order factors

$$H(z) = \prod_{k=1}^{N_s} \frac{b_{0k} + b_{1k} z^{-1} + b_{2k} z^{-2}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}}$$

$\Rightarrow$ Can efficiently implement 2nd-order subsystems

Cascade form

- Example of 6-th order system



- Many ways to combine pairs of poles and zeros with the same overall system function **with infinite precision**
  - With finite precision, the results can be quite different

Parallel form

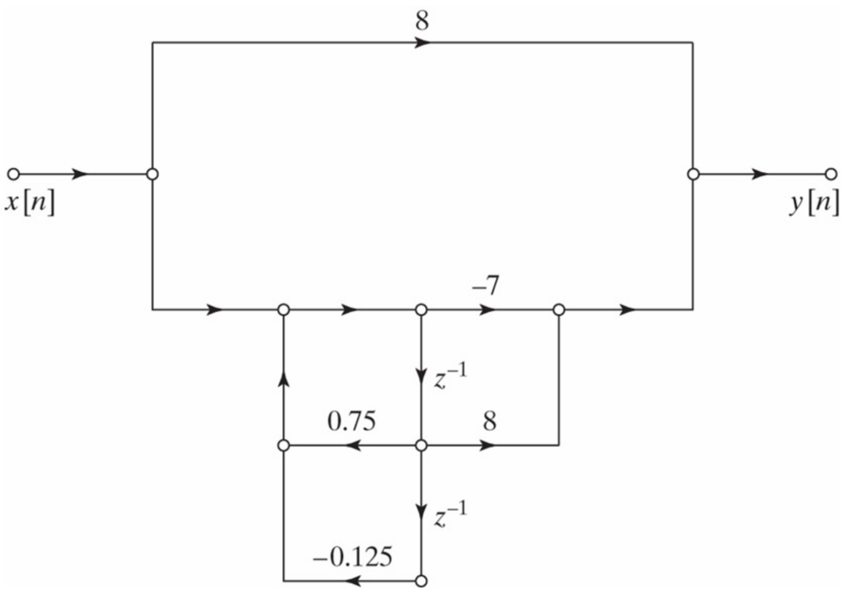
- Using partial fraction expansion

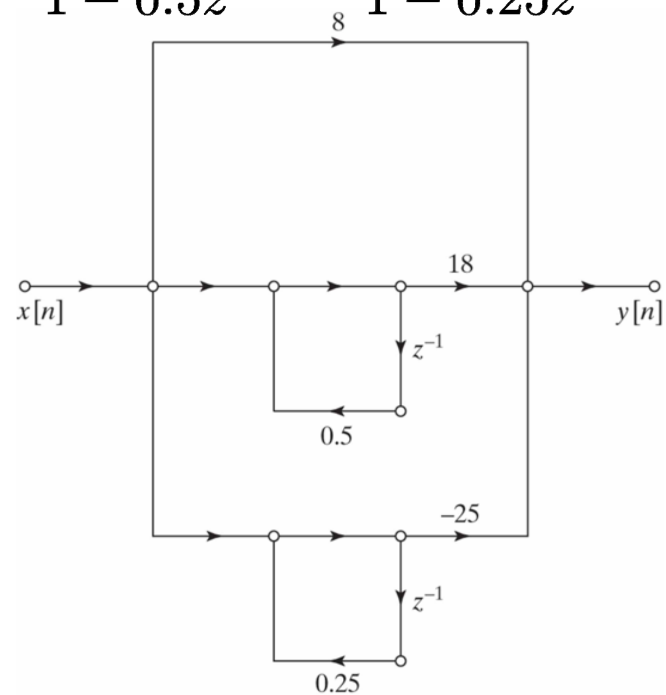
$$H(z) = \sum_{k=0}^{N_p} C_k z^{-k} + \sum_{k=1}^{N_1} \frac{A_k}{1 - c_k z^{-1}} + \sum_{k=1}^{N_2} \frac{B_k(1 - e_k z^{-1})}{(1 - d_k z^{-1})(1 - d_k^* z^{-1})}$$

$$= \sum_{k=0}^{N_p} C_k z^{-k} + \sum_{k=1}^{N_s} \frac{e_{0k} + e_{1k} z^{-1}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}}$$

Parallel form example

- $$H(z)=\frac{1+2z^{-1}+z^{-2}}{1-0.75z^{-1}+0.125z^{-2}}=8+\frac{-7+8z^{-1}}{1-0.75z^{-1}+0.125z^{-2}}$$
$$=8+\frac{18}{1-0.5z^{-1}}-\frac{25}{1-0.25z^{-1}}$$

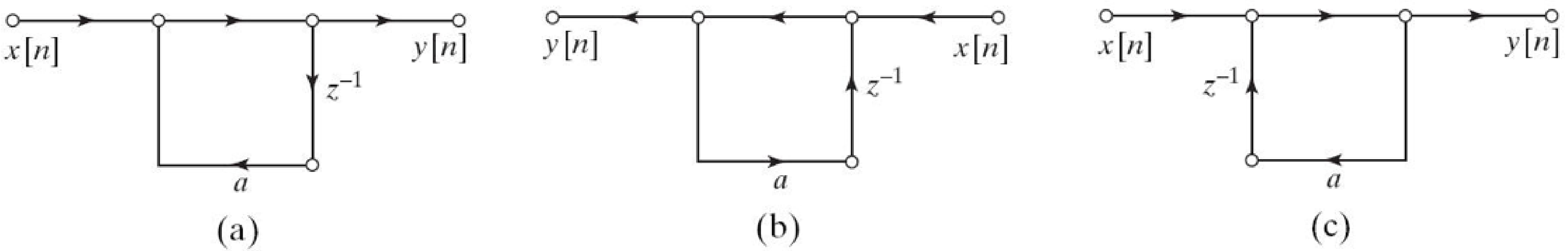




Transposed forms

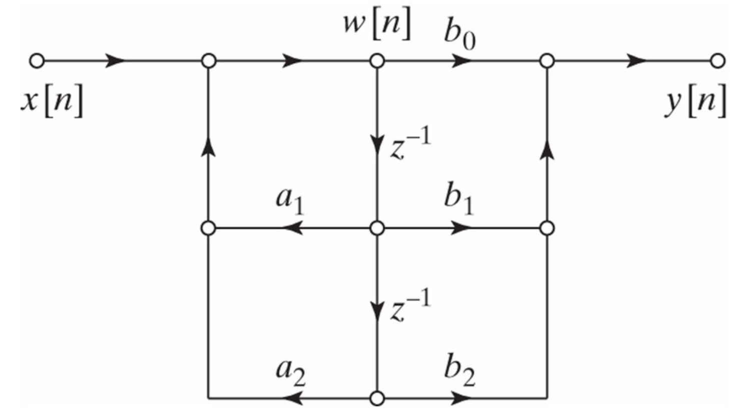
- Reverse the directions of all branches in the network
- Keep functions on branches (multiplications, delays, etc) the same
- Reverse the input and output
  ➔ Obtain the same system!

- Simple example $H(z) = \dfrac{1}{1 - az^{-1}}$

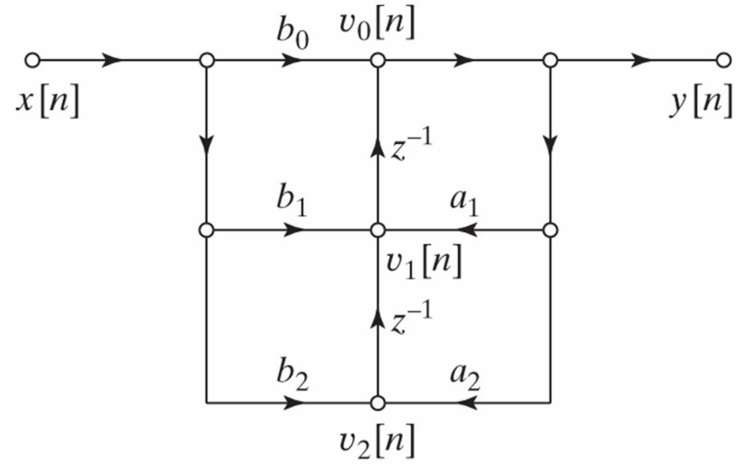


Another example of transposed form

- $w[n] = a_1 w[n-1] + a_2 w[n-2] + x[n]$
  $y[n] = b_0 w[n] + b_1 w[n-1] + b_2 w[n-2]$



- $v_0[n] = b_0 x[n] + v_1[n-1]$
  $y[n] = v_0[n]$
  $v_1[n] = a_1 y[n] + b_1 x[n] + v_2[n-1]$
  $v_2[n] = a_2 y[n] + b_2 x[n]$



- Both systems represent
  $y[n] = a_1 y[n-1] + a_2 y[n-2]$
  $\quad + b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$

Structures for FIR systems

- FIR system functions have only zeros (except for poles at z=0)

- The difference equation reduces to

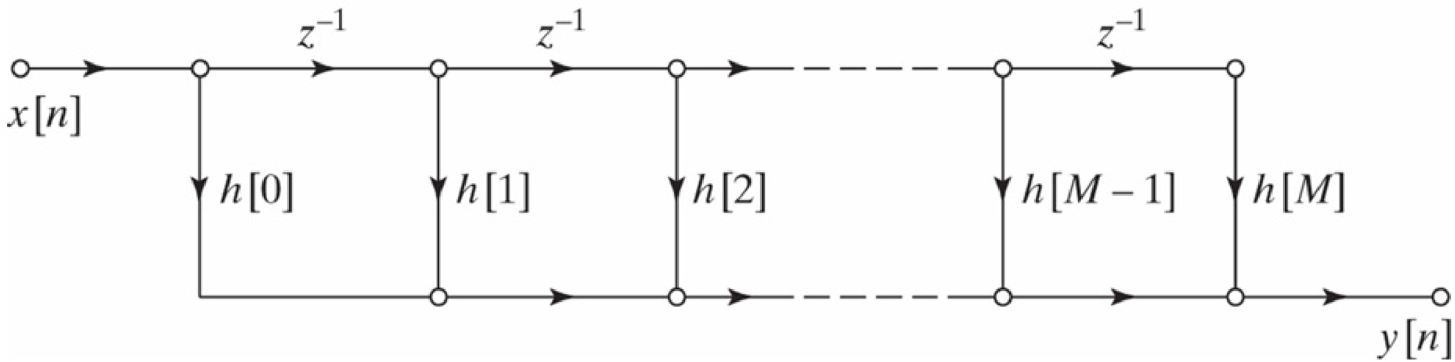
$$y[n] = \sum_{k=0}^{M} b_k x[n-k]$$

with the impulse response

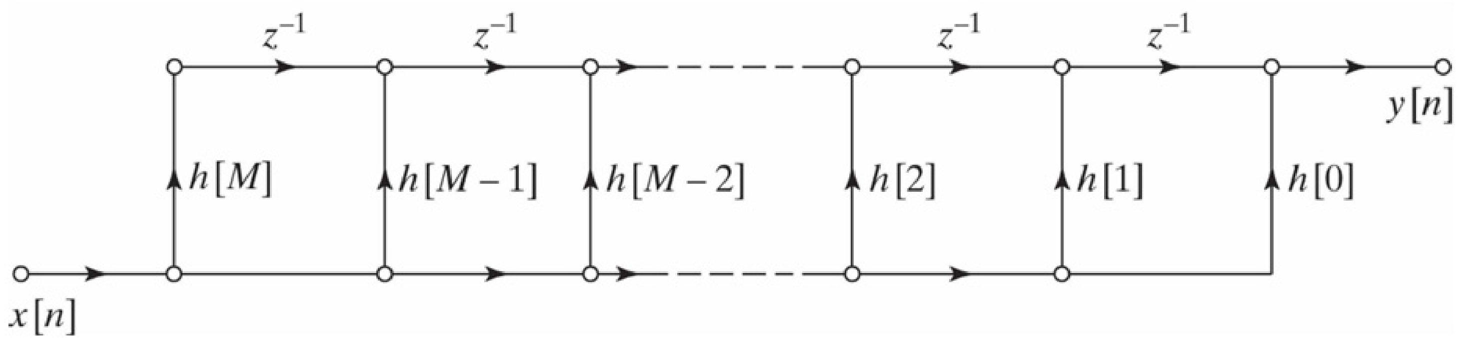
$$h[n] = \begin{cases} b_n & n = 0, 1, \ldots, M \\ 0 & \text{otherwise} \end{cases}$$

Direct form

- Direct form



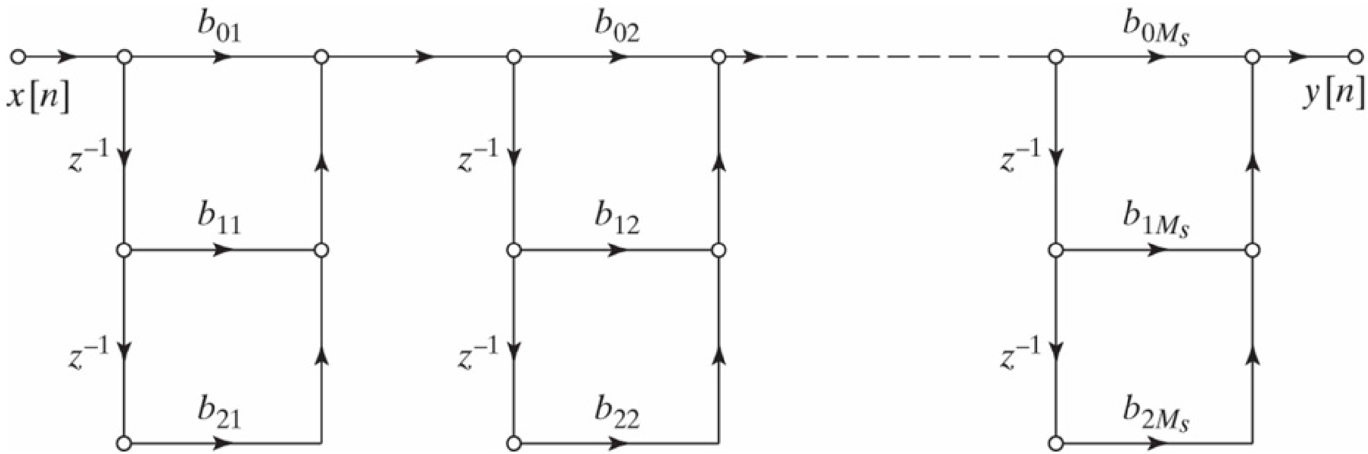
- Transposed direct form



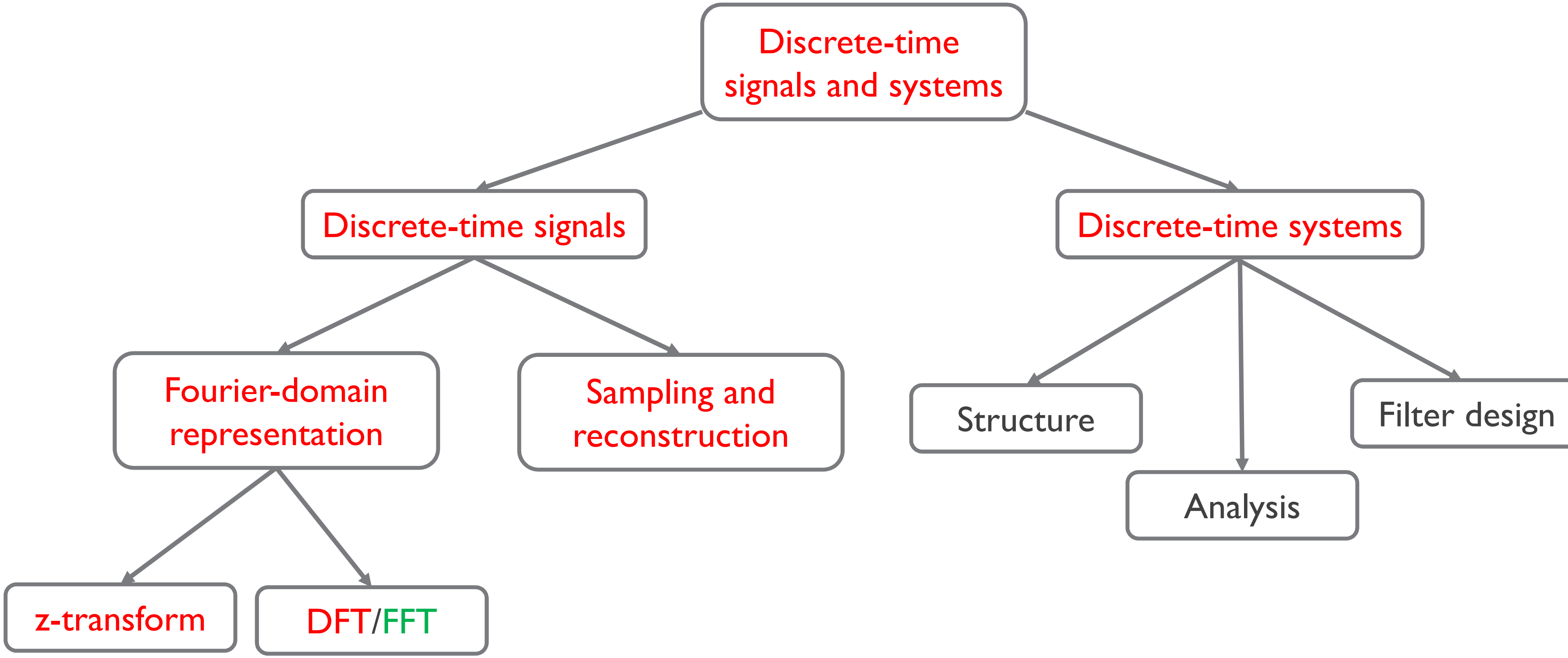
Cascade form

- Factoring the polynomial system function

$$H(z) = \sum_{n=0}^{M} h[n]z^{-n} = \prod_{k=1}^{M_s} (b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2})$$



Course at glance



Discrete Fourier Transform (DFT)

- Both time-domain sequence $x[n]$ and its DFT $X[k]$ are discrete sequences
  ➔ Appropriate for digital processing

- The complexity of direct computations

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \quad 0 \le k \le N-1$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}, \quad 0 \le n \le N-1$$

proportional to N^2
➔ May not be feasible for large *N*

Complexity of direct DFT

- $x[n]$ is complex sequence in general

- To compute DFT

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} = \sum_{n=0}^{N-1} \left[(\mathrm{Re}\{x[n]\}\mathrm{Re}\{W_N^{kn}\} - \mathrm{Im}\{x[n]\}\mathrm{Im}\{W_N^{kn}\}) \right.$$
$$\left. + j(\mathrm{Re}\{x[n]\}\mathrm{Im}\{W_N^{kn}\} - \mathrm{Im}\{x[n]\}\mathrm{Re}\{W_N^{kn}\}) \right], \quad 0 \le k \le N-1$$

  - Each component of $X[k]$ requires N complex ($4N$ real) multiplications and (N-1) complex (($4N$-2) real) additions
    ➔ $X[k]$ requires N^2 complex multiplications and $N(N-1)$ complex additions

How to reduce complexity?

- Note $W_N^{k(N-n)} = W_N^{-kn} = (W_N^{kn})^*$

- Grouping

$$\mathrm{Re}\{x[n]\}\mathrm{Re}\{W_N^{kn}\} + \mathrm{Re}\{x[N-n]\}\mathrm{Re}\{W_N^{k(N-n)}\}$$
$$= (\mathrm{Re}\{x[n]\} + \mathrm{Re}\{x[N-n]\})\mathrm{Re}\{W_N^{kn}\}$$

$$-\mathrm{Im}\{x[n]\}\mathrm{Im}\{W_N^{kn}\} - \mathrm{Im}\{x[N-n]\}\mathrm{Im}\{W_N^{k(N-n)}\}$$
$$= -(\mathrm{Im}\{x[n]\} - \mathrm{Im}\{x[N-n]\})\mathrm{Im}\{W_N^{kn}\}$$

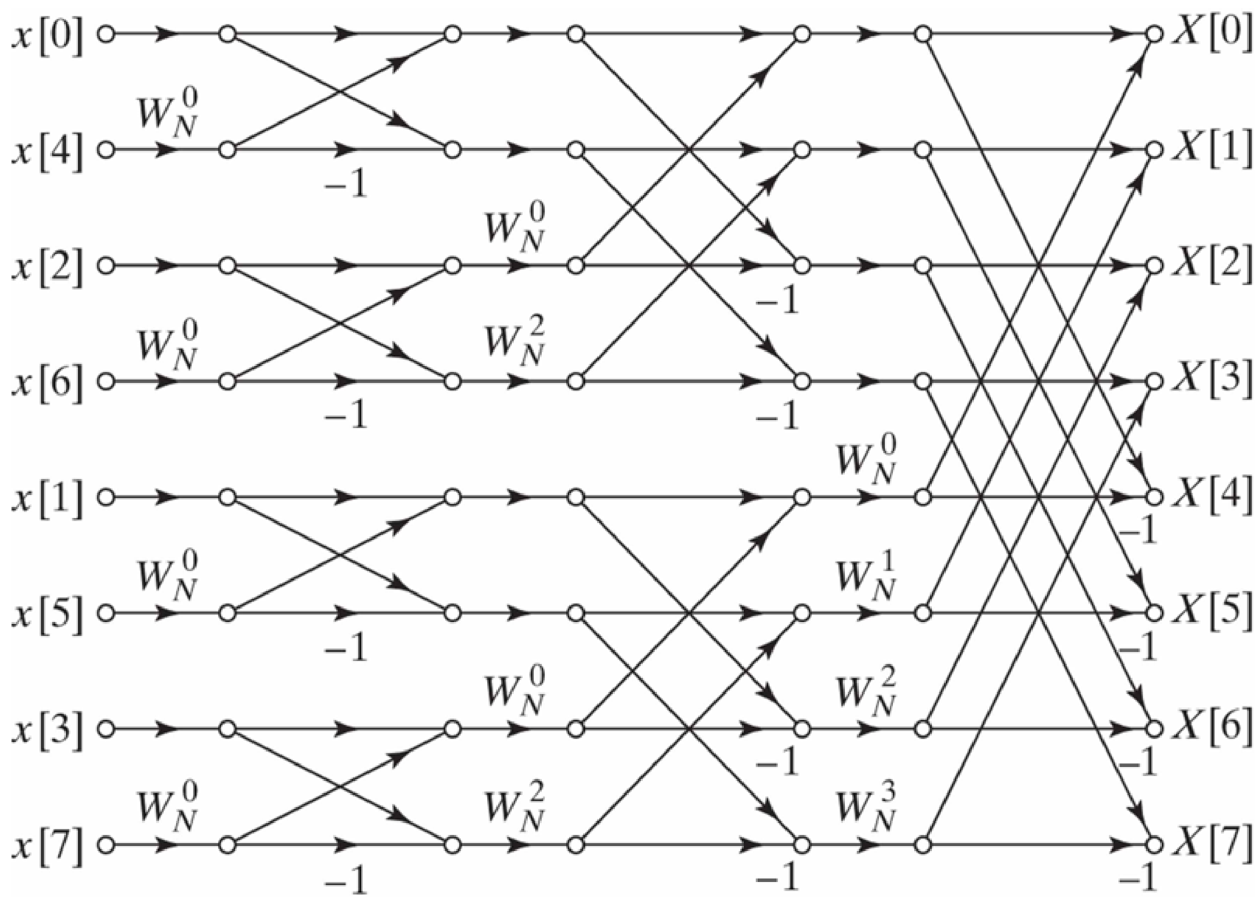
➔ Number of multiplications can be reduced by approximately a factor of 2

Fast Fourier Transform (FFT)

- A class of algorithms for efficient computation of DFT
- Computation proportional to $N \log_2 N$
- FFT algorithms not applicable for all values of *N*
- In general, FFT works for $N = 2^v$ for arbitrary positive integer v

Butterfly computation

- 8-point DFT



Complexity comparison

Figure 9.26 Number of floating-point operations as a function of N for MATLAB fft ( ) function (revision 5.2).