EEE336 Signal Processing and Digital Filtering

Lecture 13 Digital Filters Structures

Lect_13_1 Building blocks and Block diagram

Zhao Wang

Zhao.wang@xjtlu.edu.cn

Room EE322

Time domain characterization

- The Input-output relation of an LTI digital filter in real world are implemented in time domain.
  - Convolution Sum:

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$$

  - Constant Coefficient Linear Difference Equation:

$$y[n] = -\sum_{k=1}^{N} d_k y[n-k] + \sum_{k=0}^{M} p_k x[n-k]$$

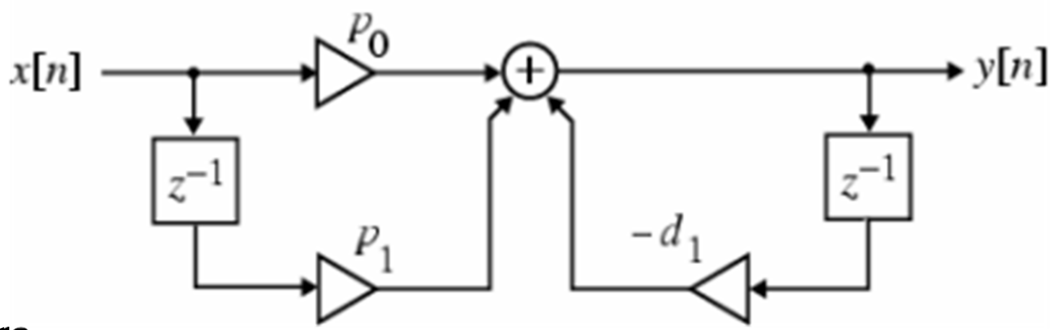
- A structural representation using interconnected basic building blocks is the first step in the hardware or software implementation of an LTI digital filter.

Time domain characterization

- For the implementation of an LTI digital filter, the input-output relationship must be described by a valid *computational algorithm*
  - For example, consider a causal first-order LTI system:

$$y[n] = -d_1 y[n-1] + p_0 x[n] + p_1 x[n-1]$$

can be implemented as follows:

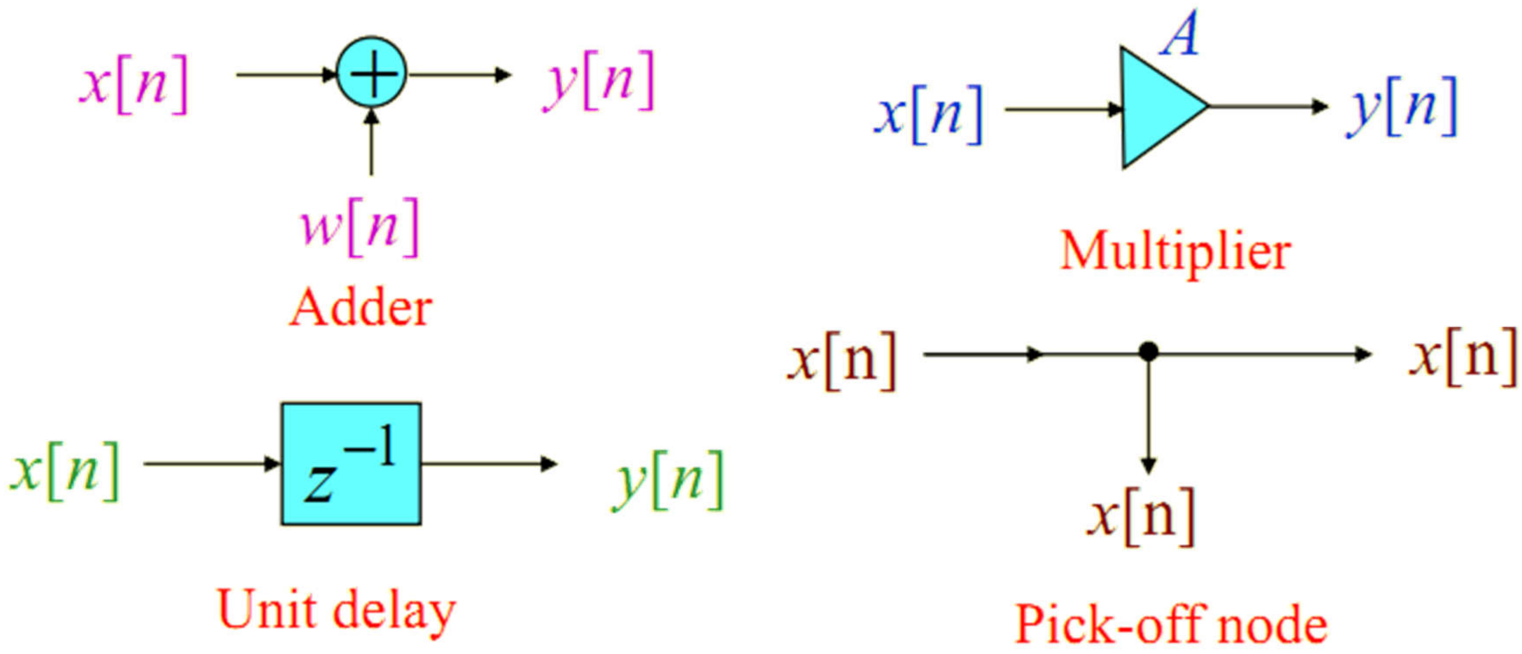


- Knowing the initial condition y[-1] and the input x[n], we can calculate y[n], n≥0.

$$y[0] = -d_1 y[-1] + p_0 x[0] + p_1 x[-1]$$
$$y[1] = -d_1 y[0] + p_0 x[1] + p_1 x[0]$$
$$y[2] = -d_1 y[1] + p_0 x[2] + p_1 x[1]$$

Basic building blocks

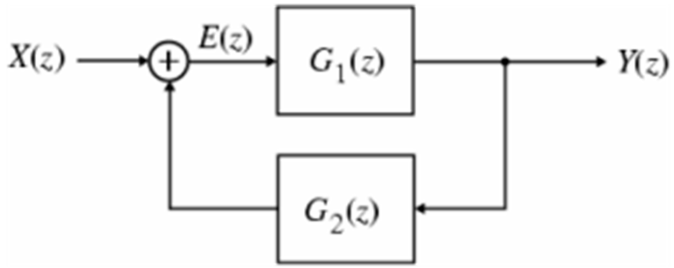
- The computational algorithm of an LTI digital filter can be conveniently represented in block diagram form using the basic building blocks:



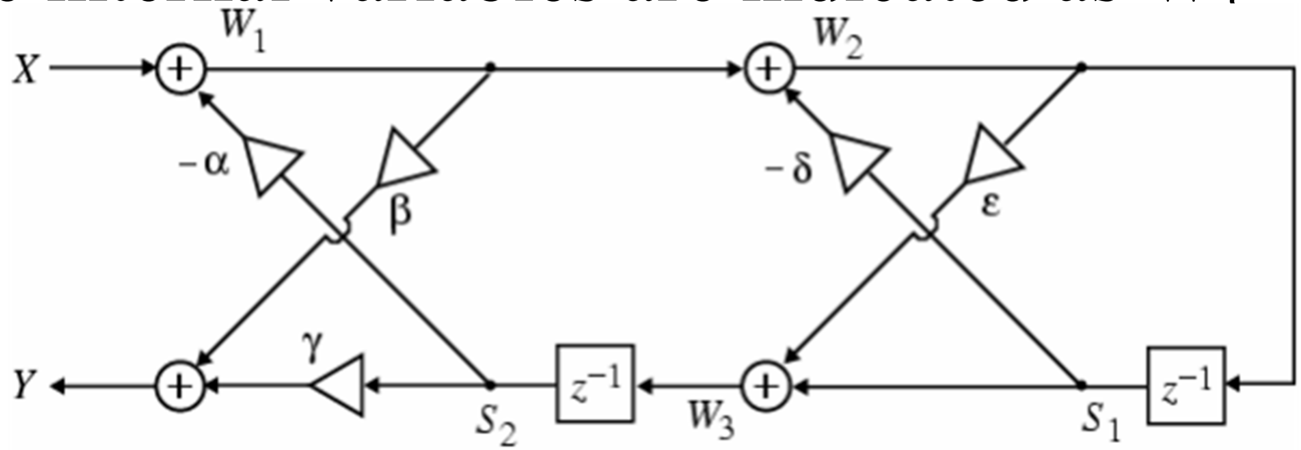
Analysis of block diagrams

- Given a block diagram, the filter implemented by that diagram can be obtained by
  - writing down the input / output equations on key points of the block diagram;
  - eliminating the internal variables;
  - finally obtaining the main input / output expression.

- Example: consider the single-loop feedback structure:



- Example: consider the following (cascaded lattice) structure, where the internal variables are indicated as $W_1 \sim W_3$ and $S_1 \sim S_2$



- We can write the following expressions:

$$W_1 = X - \alpha S_2$$
$$W_2 = W_1 - \delta S_1$$
$$W_3 = \varepsilon W_2 + S_1$$
$$Y = \beta W_1 + \gamma S_2$$
$$S_2 = z^{-1} W_3$$
$$S_1 = z^{-1} W_2$$

$\Rightarrow$

$$W_1 = X - \alpha z^{-1} W_3$$

$$\left.\begin{array}{l} W_2 = W_1 - \delta z^{-1} W_2 \\ W_3 = z^{-1} W_2 + \varepsilon W_2 \end{array}\right\} \Rightarrow \left.\begin{array}{l} W_2 = W_1 / (1 + \delta z^{-1}) \\ W_3 = (\varepsilon + z^{-1} \ W_2) \end{array}\right\} W_3 = \frac{\varepsilon + z^{-1}}{1 + \delta z^{-1}} W_1$$

$$Y = \beta W_1 + \gamma z^{-1} W_3$$

$\Rightarrow$

$$\boxed{H(z) = \frac{Y}{X} = \frac{\beta + (\beta\delta + \gamma\varepsilon) z^{-1} + \gamma z^{-2}}{1 + (\delta + \alpha\varepsilon) z^{-1} + \alpha z^{-2}}}$$

13_1 Wrap up

- Fundamental building blocks
- Feedback loop

- Transfer function → Block diagram
- Block diagram → Transfer function

EEE336 Signal Processing and Digital Filtering

Lecture 13 Digital Filters Structures

Lect_13_2 Important Concepts

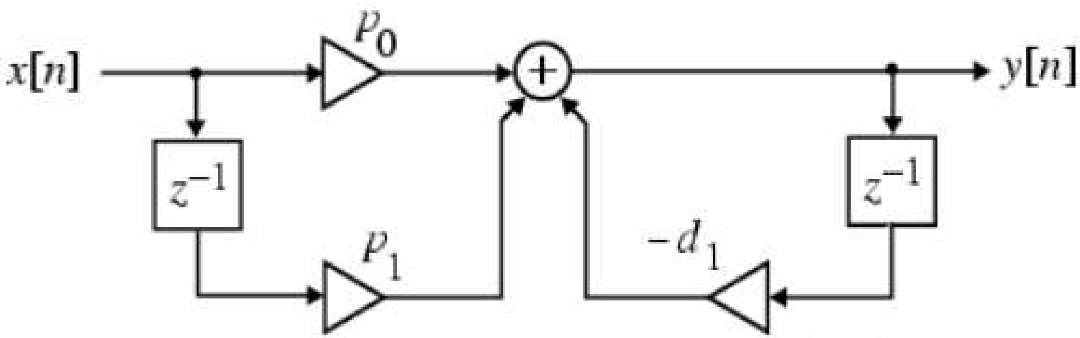
Zhao Wang

Zhao.wang@xjtlu.edu.cn

Room EE322

Canonic and noncanonic structures

- A digital filter structure is said to be **canonic** if the number of delays in the block diagram representation is equal to the order of the transfer function. Otherwise, it is a **noncanonic** structure.
  - Example: this structure is not canonic, since it uses two delay elements for a first order filter



$$y[n] = -d_1 y[n-1] + p_0 x[n] + p_1 x[n-1]$$

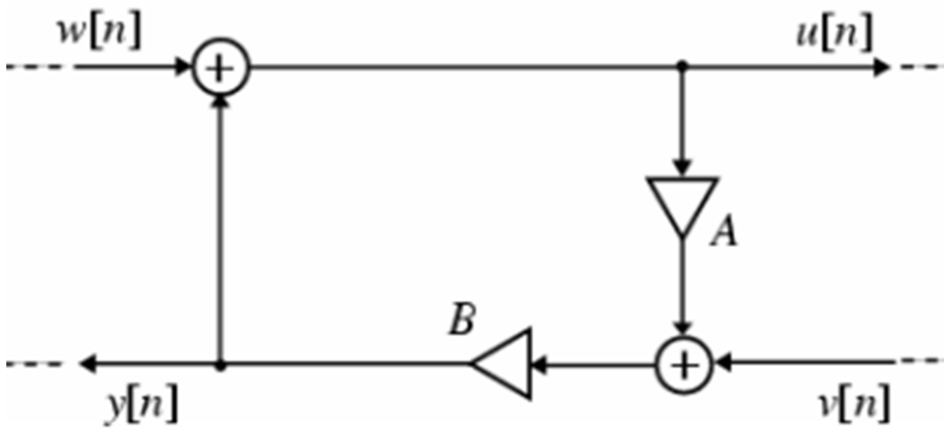
- Two digital filter structures are called ***equivalent*** if they have the same transfer function.

Equivalent structure

- Why do we need equivalent structures?
  - To avoid delay-free loop problem
    - A structure that includes delay free loops is physically impossible to implement.
  - To deal with finite precision problems
    - Under infinite precision arithmetic any given realization of a digital filter behaves identically to any other equivalent structure
    - However, in practice, due to the finite word-length limitations, a specific realization may behave very different than its other "equivalent" realizations.
    - Hence, it is important to choose a structure that has the least quantization effects when implemented using finite precision arithmetic

Delay-free loop

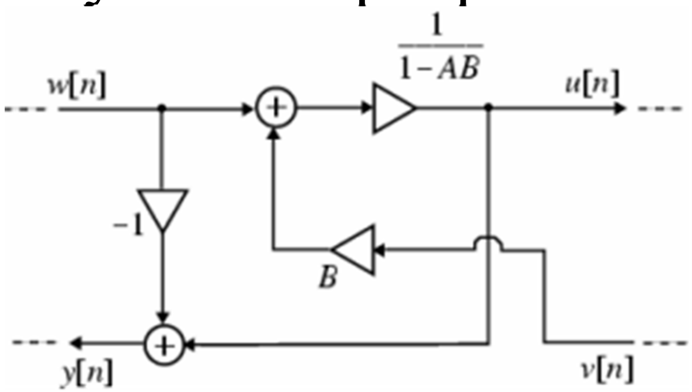
- A block diagram containing delay-free loops is physically non-realizable.



$$u[n] = w[n] + y[n]$$
$$y[n] = B(v[n] + Au[n])$$
$$\Downarrow$$
$$y[n] = B\big(v[n] + A(w[n] + y[n])\big)$$

The determination of the current value of y[n] requires the knowledge of itself.

  – Delay-free loop equivalent realization:



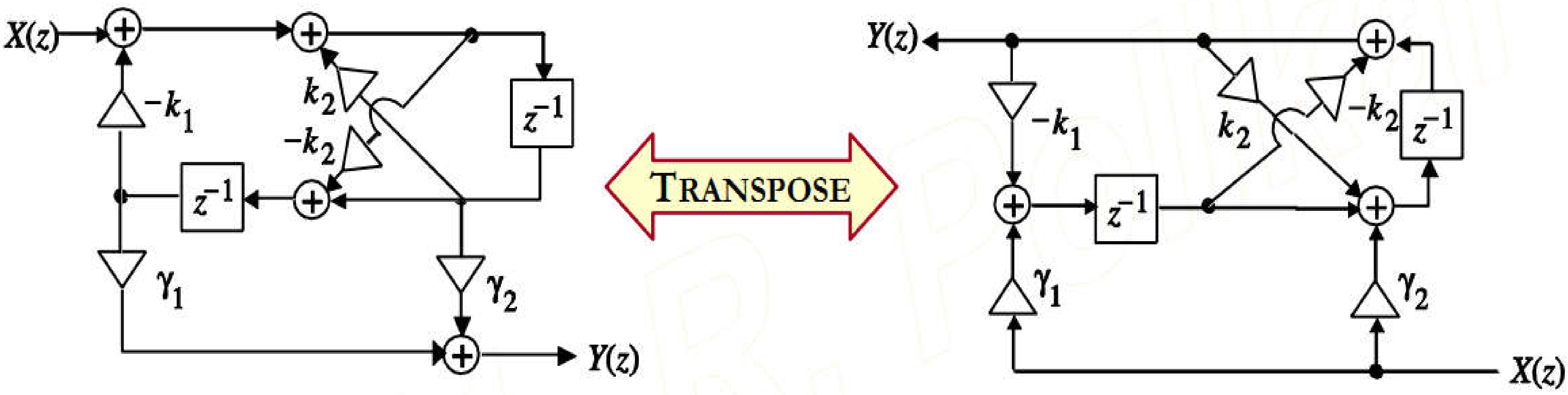
$$y[n] = \frac{AB}{1-AB} w[n] + \frac{B}{1-AB} v[n]$$

Equivalent structure

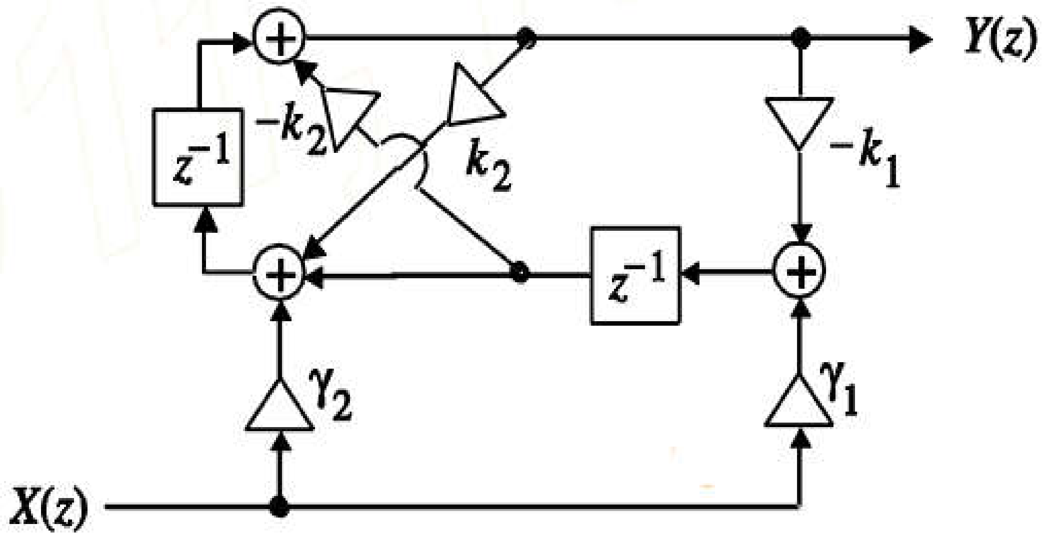
- Two digital filter structures are called ***equivalent*** if they have the same transfer function.

- There are a number of methods for the generation of equivalent structures.
- One of many equivalent structures (the most simple one) is obtained using the **transpose** operation.
  - Reverse all paths
  - Replace pick-off nodes by adders, and vice versa
  - Interchange the input and output nodes

- For example:



Redraw the transposed structure for left – to – right input / output representation



13_2 Wrap up

- Equivalent structures
- Canonic structures
- Delay-free loop
- Transpose

EEE336 Signal Processing and Digital Filtering

Lecture 13 Digital Filters Structures

Lect_13_3 FIR System Structures

Zhao Wang

Zhao.wang@xjtlu.edu.cn

Room EE322

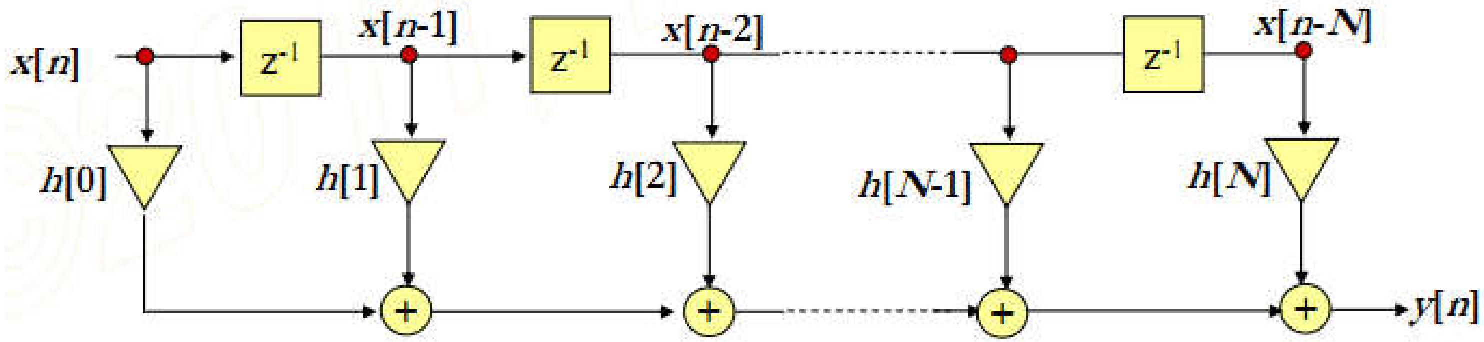
Basic FIR structures

- A causal FIR filter can be represented in time domain with its CCLDE equation, which is equivalent to its impulse response representation:

$$y[n] = x[n] * h[n] = \sum_{k=0}^{N} h[k]x[n-k]$$
$$= h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + \cdots + h[N]x[n-N]$$

- This Nth order (N+1-coefficient) filter can be implemented directly by using N+1 multipliers and N two-input adders

Direct form:

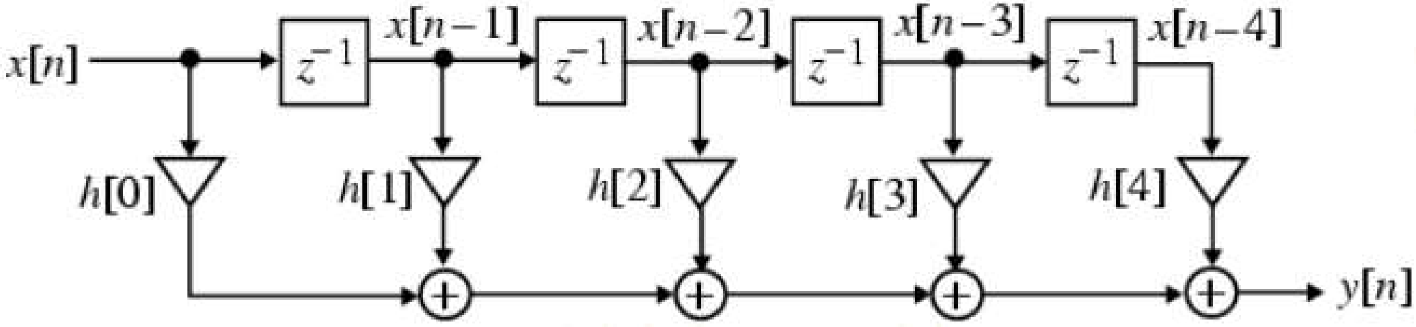


- Structures in which the multiplier coefficients are precisely the coefficients of the transfer function are called **direct form** structures

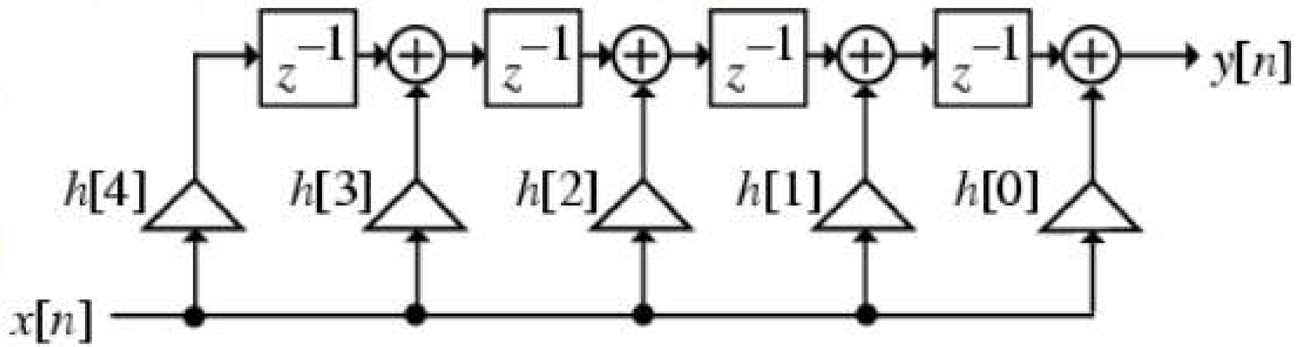
Basic FIR structures

- For an order N=4 filter:

$$y[n] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + h[3]x[n-3] + h[4]x[n-4]$$



- And its transpose:



- Both are canonic structures. These structures are also known as <u>*tapped delay line*</u> or <u>*transversal*</u> filter structures

Linear phase FIR structures

- Recall that a linear phase filter must have either symmetry or anti-symmetry property, which can be exploited to reduce the number of multipliers into almost half of that in the direct form implementations
  - For example, consider a length-7 Type 1 FIR transfer function with a symmetric impulse response:

$$H(z)=h[0]+h[1]z^{-1}+h[2]z^{-2}+h[3]z^{-3}+h[2]z^{-4}+h[1]z^{-5}+h[0]z^{-6}$$

  - Rewrite this expression as

$$H(z)=h[0](1+z^{-6})+h[1](z^{-1}+z^{-5})+h[2](z^{-2}+z^{-4})+h[3]z^{-3}$$

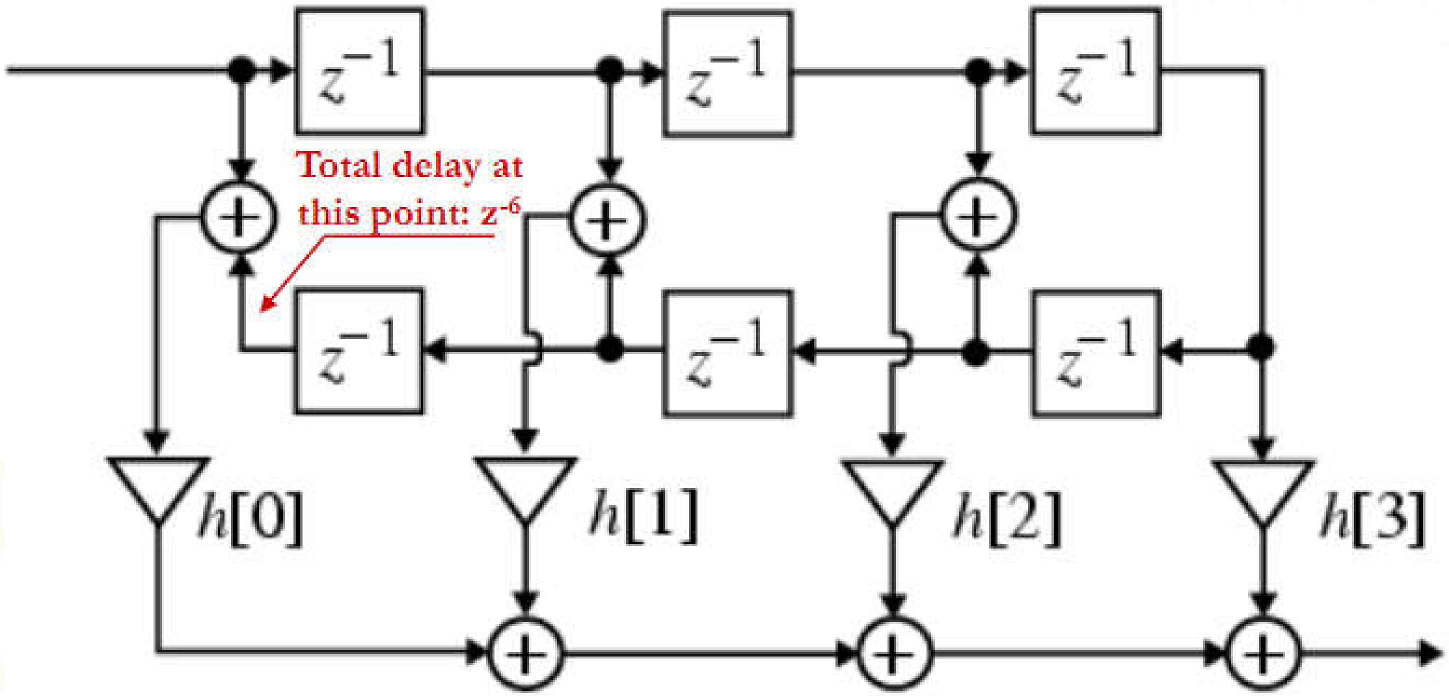
    Which only needs four multipliers instead of seven

Linear phase FIR structures

- Linear phase length-7 Type 1 FIR transfer function:

$$H(z) = h[0]\left(1+z^{-6}\right)+h[1]\left(z^{-1}+z^{-5}\right)+h[2]\left(z^{-2}+z^{-4}\right)+h[3]z^{-3}$$

- Block diagram can be modified as:

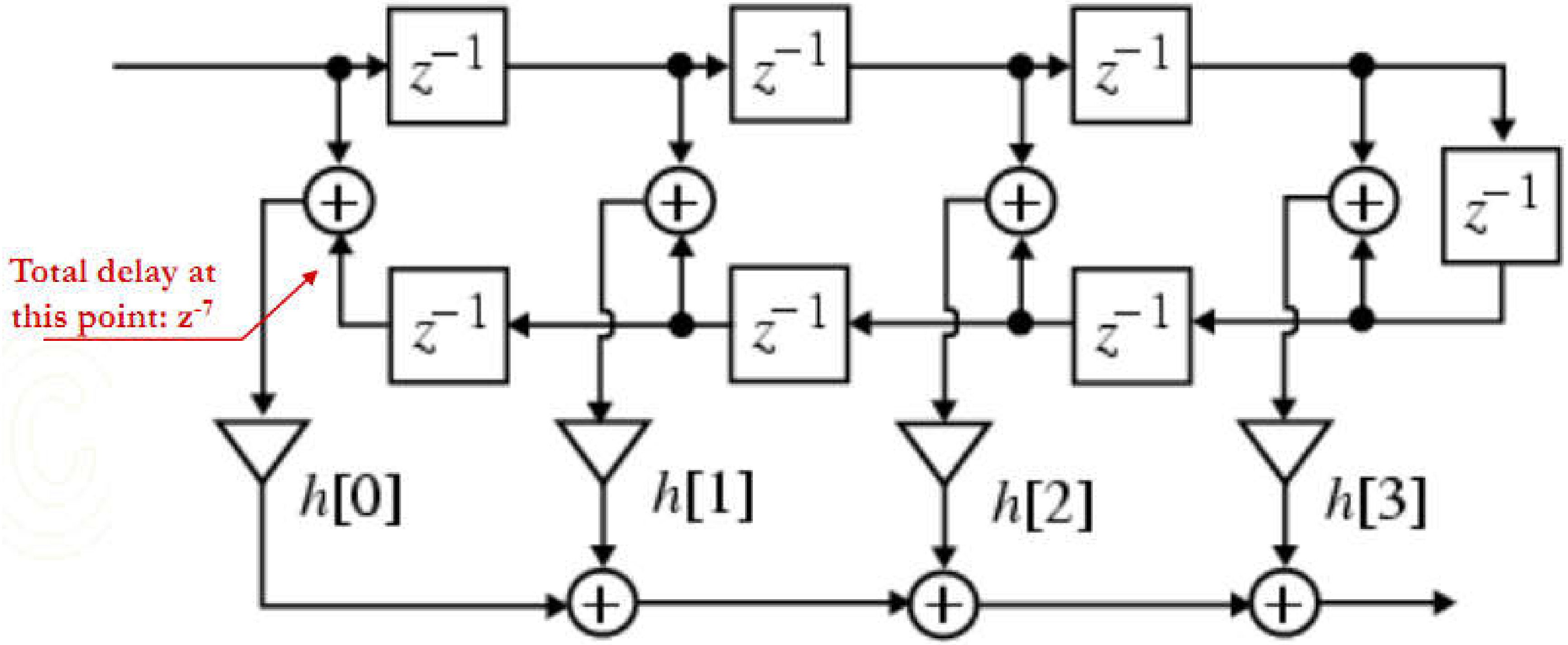


Linear phase FIR structures

- Similarly, and FIR type 2 filter (with even length, say 8)

$$H(z) = h[0](1+z^{-7}) + h[1](z^{-1}+z^{-6}) + h[2](z^{-2}+z^{-5}) + h[3](z^{-3}+z^{-4})$$

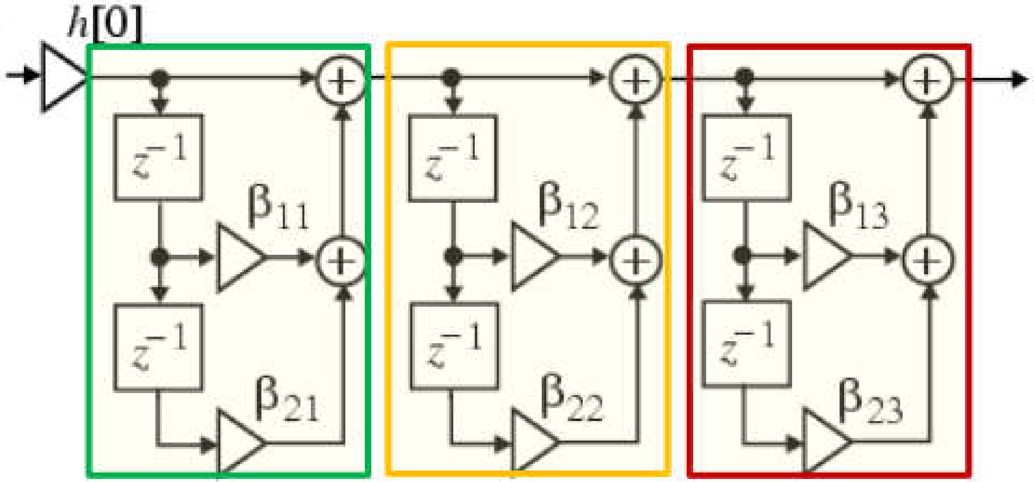
can be implemented with 4 multipliers, instead of 8.



  – Note that it still needs 7 delay elements.

Cascade form FIR structures

- A higher order FIR filter can be realized from a cascade of lower (first or second) order structures.
  - The general structure of an FIR filter can be factored into a product of second order functions:
$$H(z) = h[0]\prod_{k=1}^{K}(1+\beta_{1k}z^{-1}+\beta_{2k}z^{-2})$$
  - where K=N/2 if N is even and K=(N+1)/2 if N is odd, with $\beta_{2K}=0$
- For example: a 6th order FIR system can be designed as a cascade of three second order systems.



Parallel form - Polyphase realization

- Polyphase decomposition of the FIR transfer function results in a parallel structure of an FIR filter
- For example, consider a length-9 FIR filter

$$H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} + h[4]z^{-4}$$
$$+ h[5]z^{-5} + h[6]z^{-6} + h[7]z^{-7} + h[8]z^{-8}$$

- Expressing the above equation as a sum of two terms, one containing the even-indexed coefficients and the other containing the odd-indexed coefficients

$$H(z) = (h[0] + h[2]z^{-2} + h[4]z^{-4} + h[6]z^{-6} + h[8]z^{-8})$$
$$+ z^{-1}(h[1] + h[3]z^{-2} + h[5]z^{-4} + h[7]z^{-6})$$

Polyphase realization

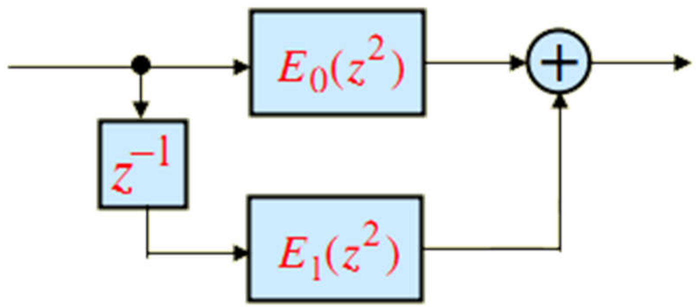
- Using the notations

$$E_0(z) = h[0] + h[2]z^{-1} + h[4]z^{-2} + h[6]z^{-3} + h[8]z^{-4}$$
$$E_1(z) = h[1] + h[3]z^{-1} + h[5]z^{-2} + h[7]z^{-3}$$

- H(z) can be written as:

$$H(z) = E_0(z^2) + z^{-1}E_1(z^2)$$

- This is more commonly known as the 2-branch polyphase decomposition as shown:



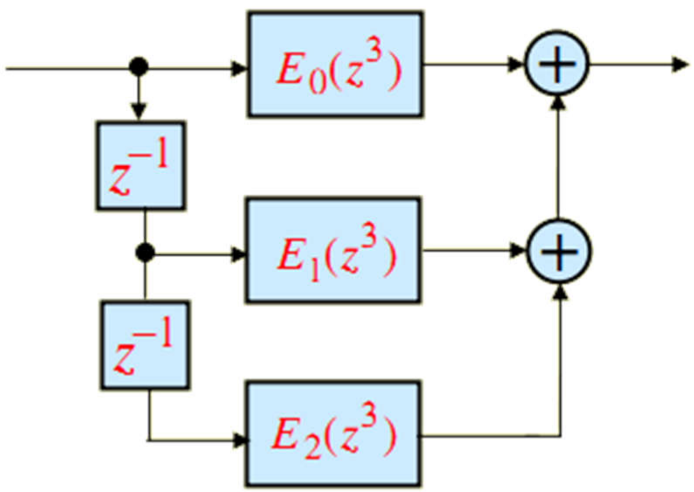
Polyphase realization

- In a similar manner, by grouping the terms in the original expression H(z) differently, we can have

$$H(z) = E_0(z^3) + z^{-1}E_1(z^3) + z^{-2}E_2(z^3)$$

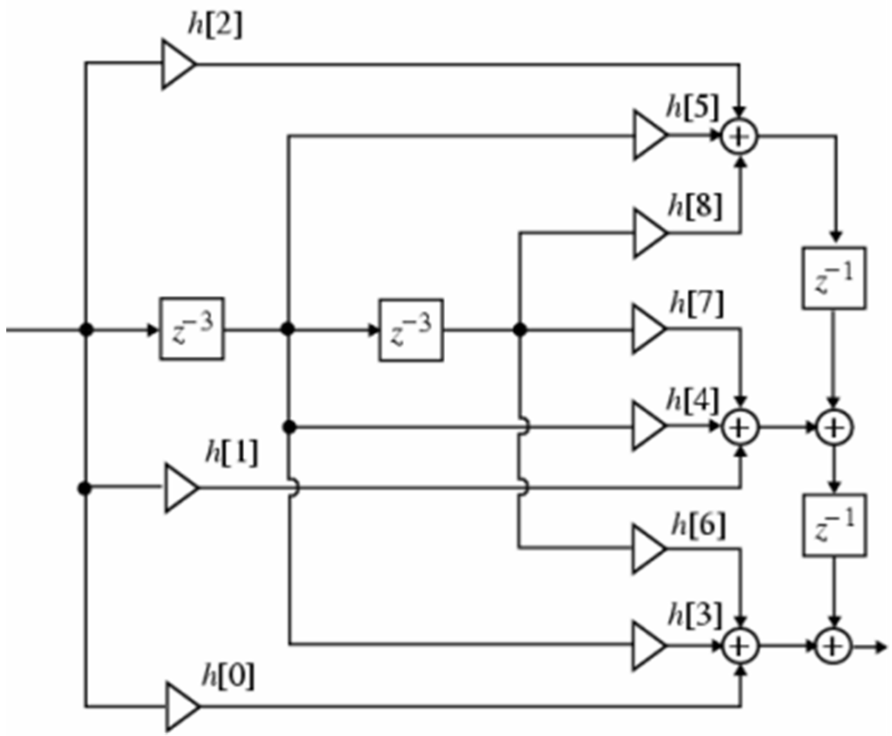
- Where $E_0(z) = h[0] + h[3]z^{-1} + h[6]z^{-2}$
  $E_1(z) = h[1] + h[4]z^{-1} + h[7]z^{-2}$
  $E_2(z) = h[2] + h[5]z^{-1} + h[8]z^{-2}$

- As known as the 3-branch polyphase decomposition illustrated by



Polyphase realization

- To obtain a canonic realization of the overall structure, the delays in all subfilters must be shared.
- For example, the 3-branch realization of the length-9 FIR filter can be obtained using delay sharing as follows:



13_3 Wrap up

- General FIR systems: direct form
  - Special realizations: linear phase FIR systems
    - Symmetry → save multiplier

- Cascade structure: simplify the subsystems

- Parallel structure: polyphase realization

EEE336 Signal Processing and Digital Filtering

Lecture 13 Digital Filters Structures

Lect_13_4 IIR System Structures

Zhao Wang

Zhao.wang@xjtlu.edu.cn

Room EE322

IIR filter structures

- From the difference equation representation of IIR filters, it can be seen that the realization of the causal IIR digital filters requires some form of feedback.

$$\sum_{k=0}^{N-1} a_0 y[n-k] = \sum_{l=0}^{M-1} b_0 x[n-l]$$

$$y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_{M-1} x[n-M+1] - a_1 y[n-1] - \cdots - a_{N-1} y[n-N+1]$$

  - Furthermore, an Nth order IIR digital transfer function is characterized by 2N+1 unique (a and b) coefficients, and in general, requires 2N+1 multipliers and 2N two-input adders for implementation

- Direct forms: Coefficients are directly the transfer function coefficients

Direct form I

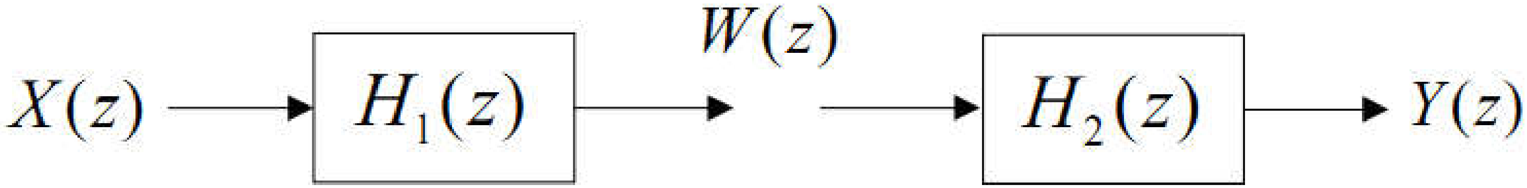
- Consider a 3rd order example:

$$H(z)=\frac{P(z)}{D(z)}=\frac{p_0+p_1z^{-1}+p_2z^{-2}+p_3z^{-3}}{1+d_1z^{-1}+d_2z^{-2}+d_3z^{-3}}$$

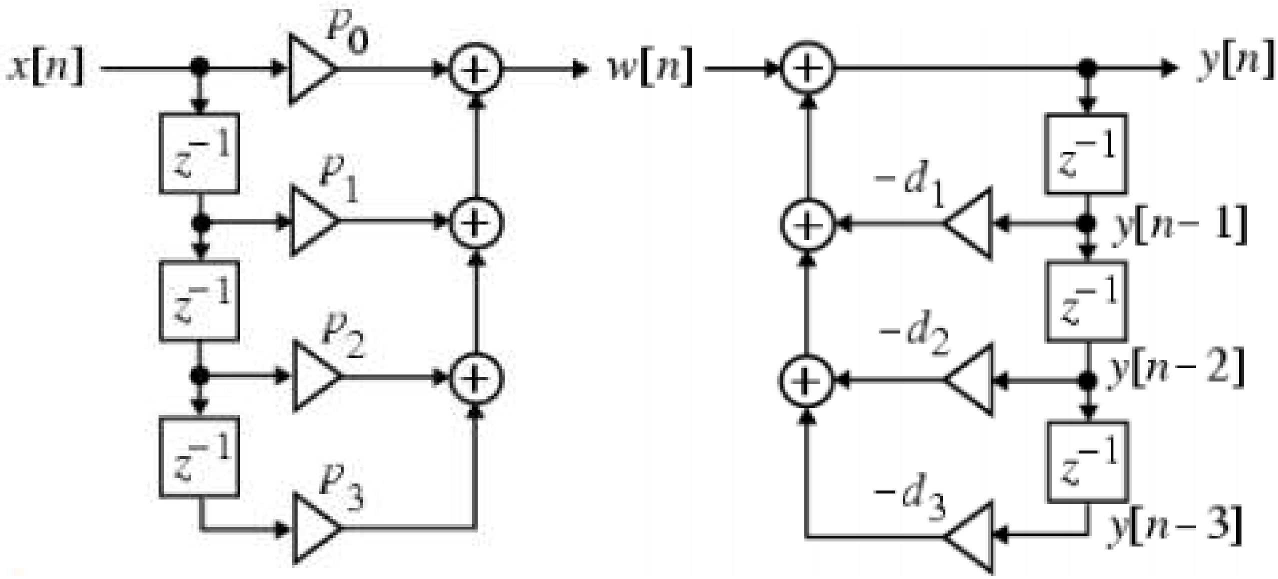
$$\rightarrow H_1(z)=\frac{W(z)}{X(z)}=P(z)=p_0+p_1z^{-1}+p_2z^{-2}+p_3z^{-3}$$

$$\rightarrow H_2(z)=\frac{Y(z)}{W(z)}=\frac{1}{D(z)}=\frac{1}{1+d_1z^{-1}+d_2z^{-2}+d_3z^{-3}}$$

  – Considering the numerator and denominator separately



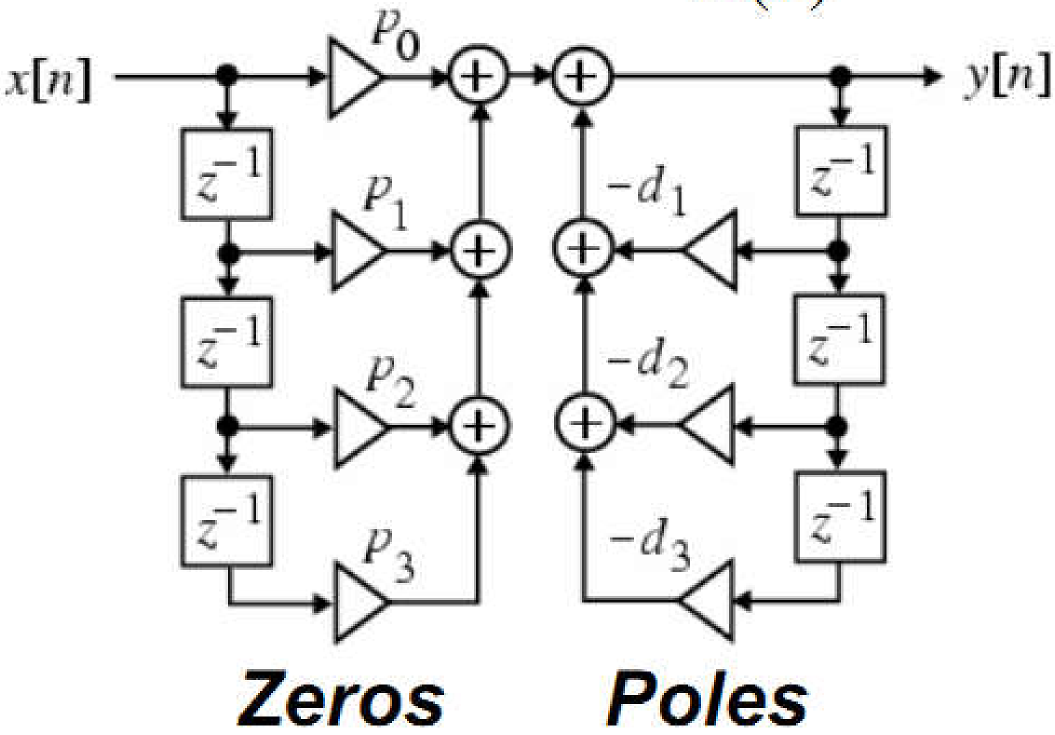
$H_1(z)$ realizes the zeros;
$H_2(z)$ realizes the poles.



Direct form I

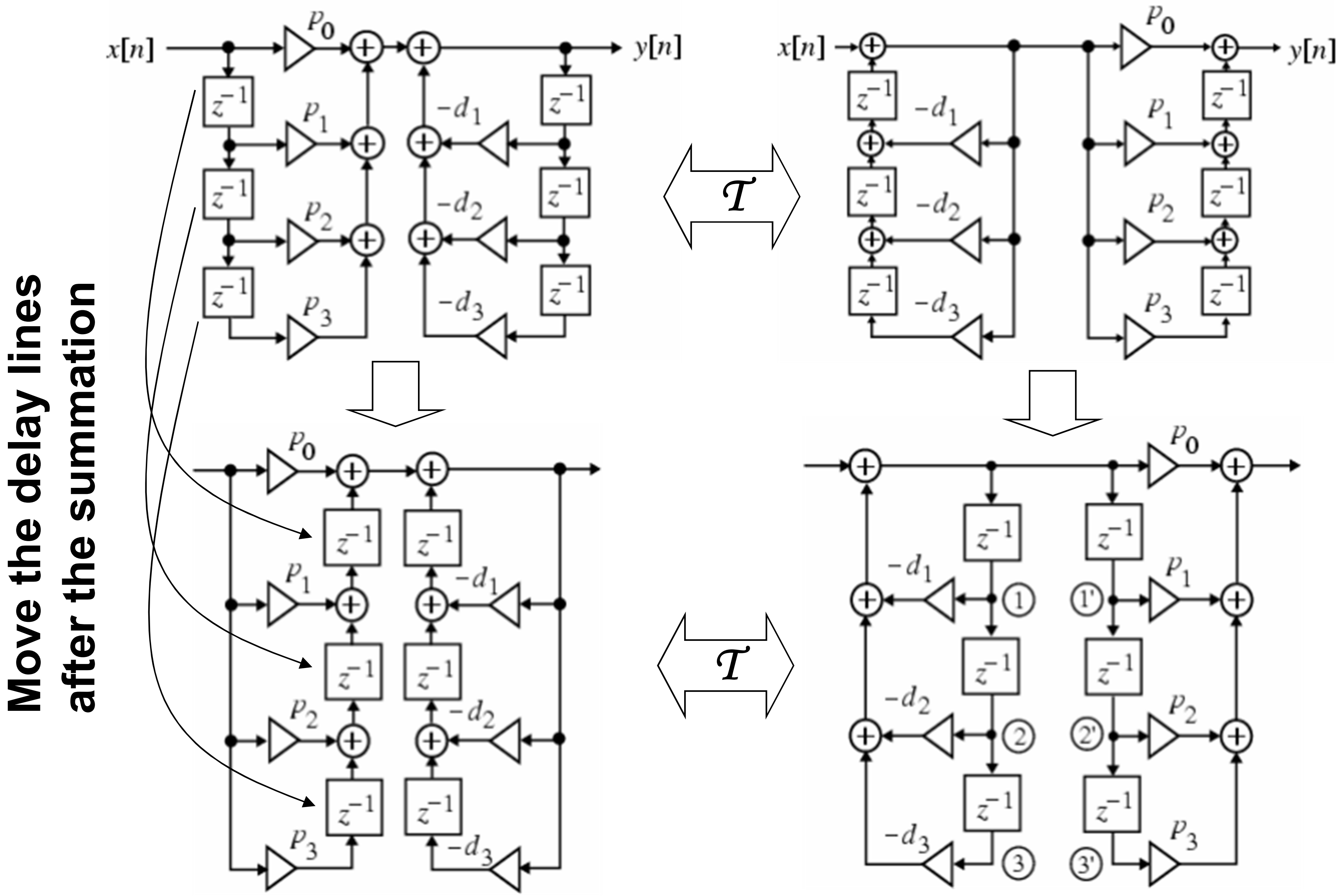
- A cascade of the two then gives us the overall H(z), whose implementation is known as <u>**Direct Form I**</u> implementation

$$H(z) = P(z) \cdot \frac{1}{D(z)}$$



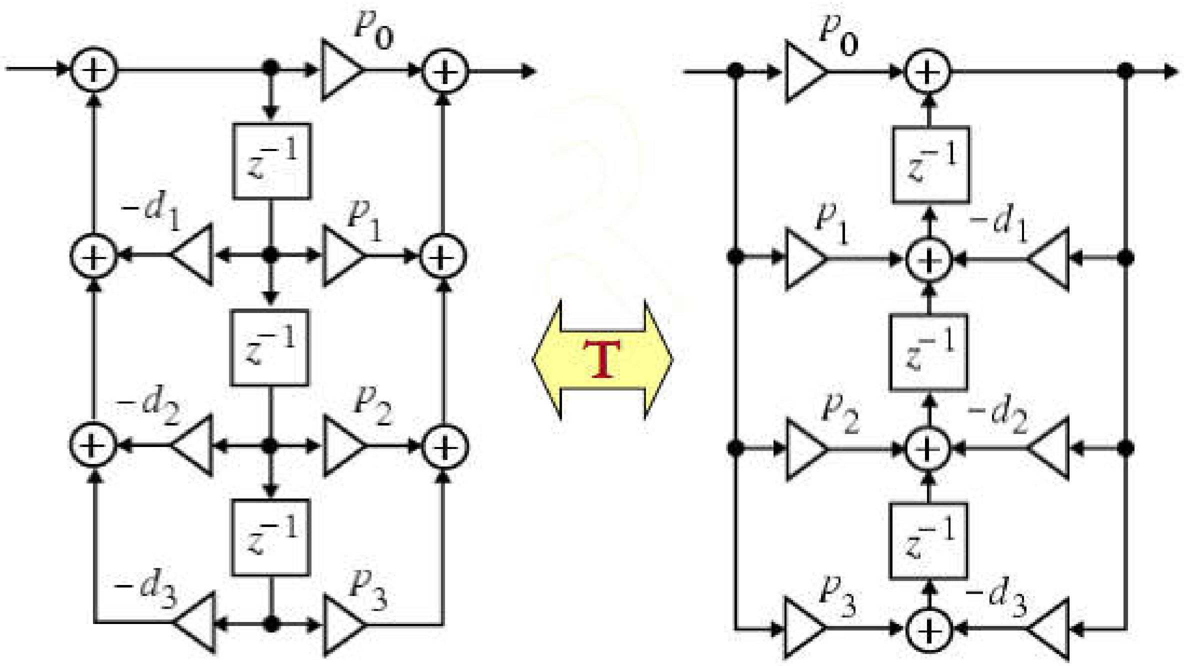
Note that this structure is noncanonic since it employs 6 delays to realize a 3rd-order transfer function

- The transpose of this implementation can also be obtained:



Direct form II

- Now notice that the points indicated as 1 and 1', 2 and 2', 3 and 3' are really indistinguishable from each other.
- => The delay elements can be shared.



3rd order

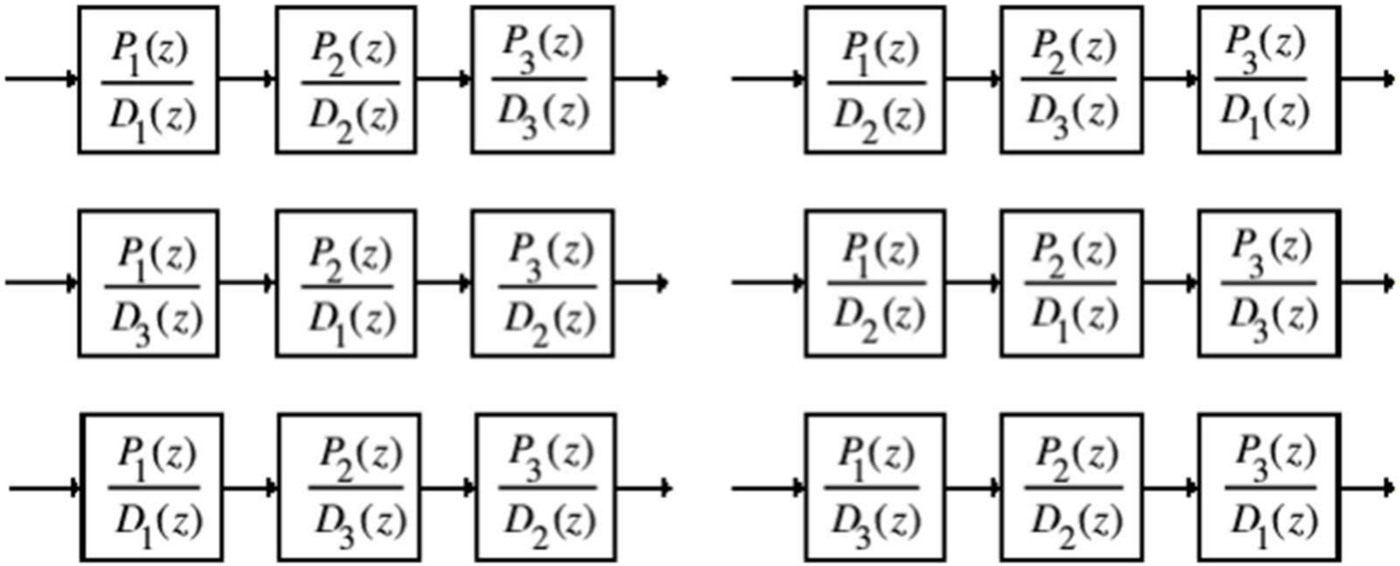
3 delays

=> Canonic!

- This particular implementation is called the Direct Form II realization, and requires half the number of delay elements!

Cascade form

- By expressing the numerator and the denominator polynomials of the transfer function as a product of polynomials of lower degree, a digital filter can be realized as a cascade of low-order filter sections
- Consider H(z)=P(z)/D(z) expressed as $H(z)=\frac{P(z)}{D(z)}=\frac{P_1(z)P_2(z)P_3(z)}{D_1(z)D_2(z)D_3(z)}$
- A total of 36 cascade realizations
can be achieved based on different pole-zero parings and ordering.
- Different realizations behave differently under finite word-length constraints



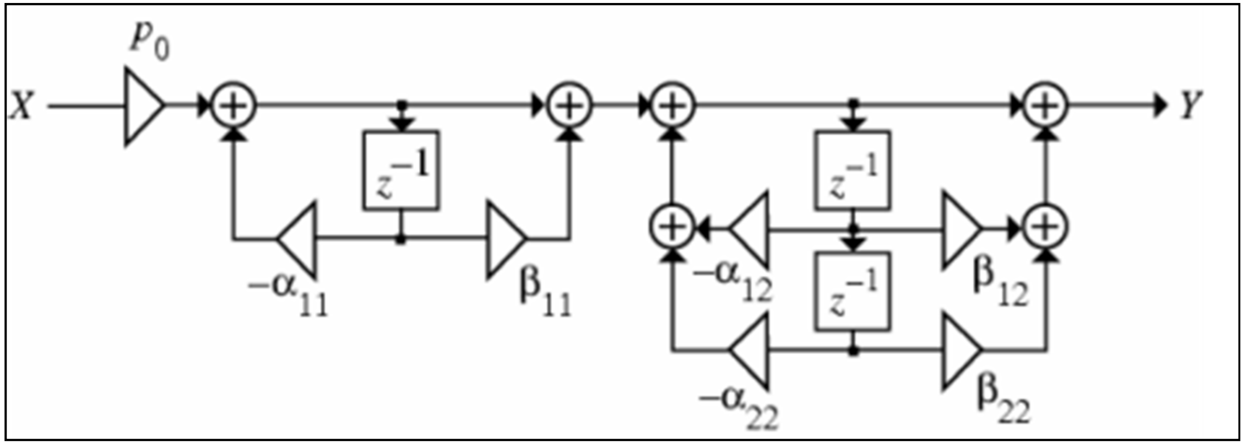
Cascade form

- Usually the polynomials are factored into a product of first and second order polynomials

$$H(z) = p_0 \prod_k \left(\frac{1 + \beta_{1k} z^{-1} + \beta_{2k} z^{-2}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \right)$$

  – Where $\alpha_{2k}=\beta_{2k}=0$ for first order forms.
  – For example, a third order system can be written and realized as

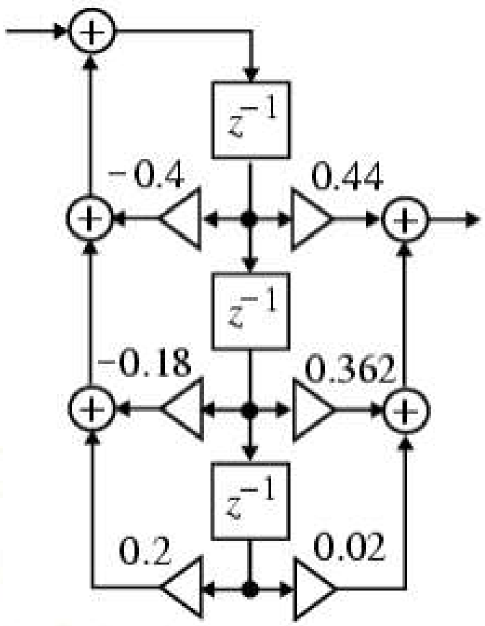
$$H(z) = p_0 \left(\frac{1+\beta_{11} z^{-1}}{1+\alpha_{11} z^{-1}} \right) \left(\frac{1+\beta_{12} z^{-1} + \beta_{22} z^{-2}}{1 + \alpha_{12} z^{-1} + \alpha_{22} z^{-2}} \right)$$



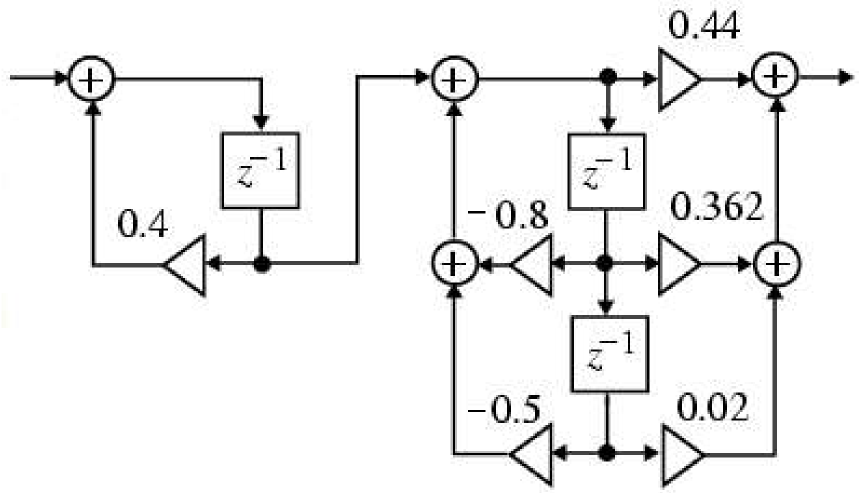
Cascade form

- An example: find the direct II and cascade form of H(z)

$$H(z)=\frac{0.44z^{-1}+0.362z^{-2}+0.02z^{-3}}{1+0.4z^{-1}+0.18z^{-2}-0.2z^{-3}}=\left(\frac{0.44+0.362z^{-1}+0.02z^{-2}}{1+0.8z^{-1}+0.5z^{-2}}\right)\left(\frac{z^{-1}}{1-0.4z^{-1}}\right)$$



Direct form II



Cascade form

Parallel form

- We can also realize IIR filters through direct partial fraction expansion, where each term is then implemented separately.
  - If the partial fraction expansion is done in terms of z^{-1}, parallel form I realization is obtained, leading to terms in the form of:

$$H(z) = \gamma_0 + \sum_k \left(\frac{\gamma_{0k} + \gamma_{1k} z^{-1}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \right)$$

  - where for real poles, $\alpha_{2k}=\gamma_{1k}=0$
  - If the partial fraction expansion is done in terms of z, parallel form II realization is obtained, leading to terms in the form of

$$H(z) = \delta_0 + \sum_k \left(\frac{\delta_{1k} z^{-1} + \delta_{2k} z^{-2}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \right)$$

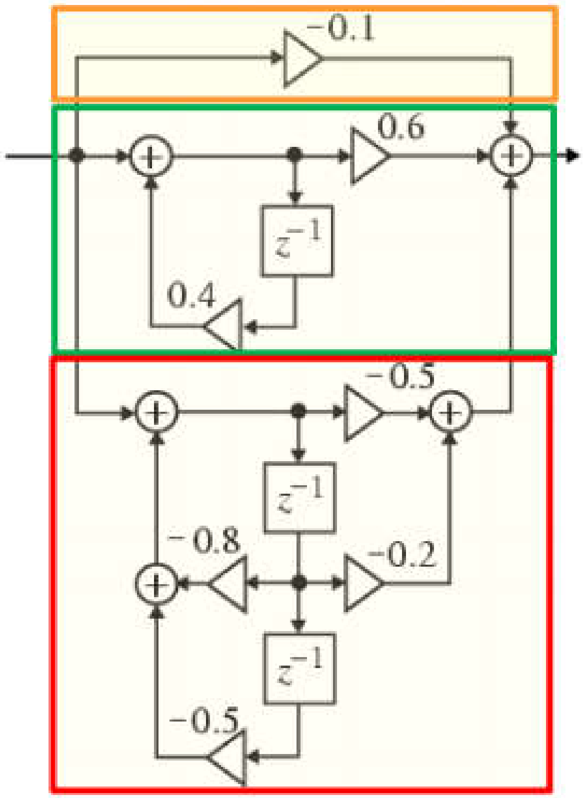
  - where for real poles, $\alpha_{2k}=\delta_{2k}=0$

- Example, consider the same H(z) as in the previous example:

$$H(z)=\frac{0.44z^{-1}+0.362z^{-2}+0.02z^{-3}}{1+0.4z^{-1}+0.18z^{-2}-0.2z^{-3}}$$

$$H(z)=\boxed{-0.1}+\boxed{\frac{0.6}{1-0.4z^{-1}}}+\boxed{\frac{-0.5-0.2z^{-1}}{1+0.8z^{-1}+0.5z^{-2}}}$$

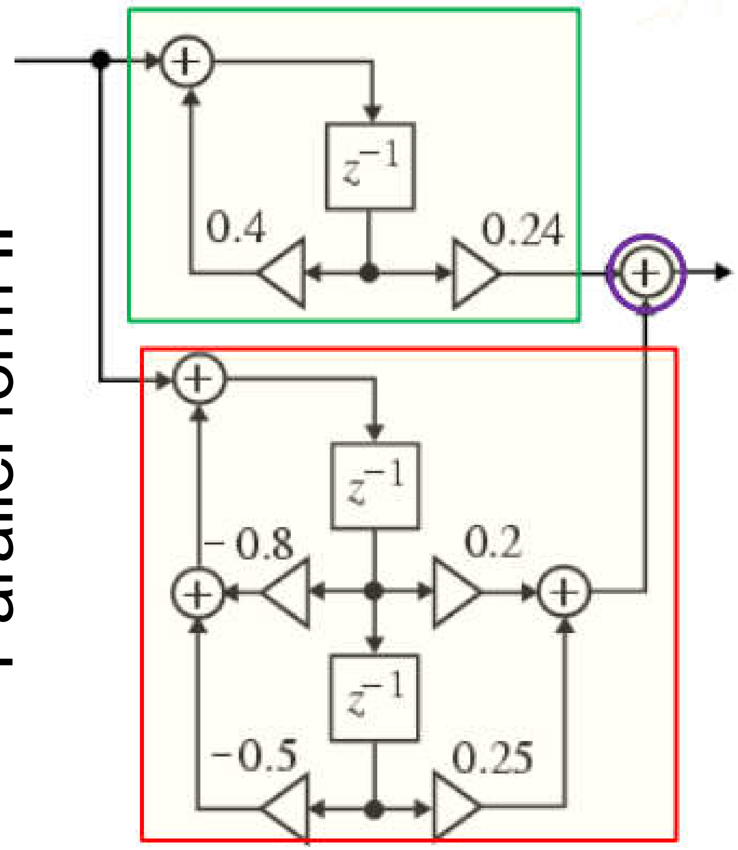
Parallel form I



$$H(z)=\frac{0.24}{z-0.4}+\frac{0.2z+0.25}{z^2+0.8z+0.5}$$

$$=\boxed{\frac{0.24z^{-1}}{1-0.4z^{-1}}}+\boxed{\frac{0.2z^{-1}+0.25z^{-1}}{1+0.8z^{-1}+0.5z^{-2}}}$$

Parallel form II



13_4 Wrap up

- Direct Forms
  - Direct Form I: Noncanonic
  - Direct Form II: Canonic
- Cascade form
- Parallel form

Chapter 13 Summary

- Block diagram
  - Transfer function → Block diagram
  - Block diagram → Transfer function
- Equivalent structures
  - Canonic VS Noncanonic
  - Transpose
- FIR Filter structures
  - Direct form
    - Linear phase FIR filters
- IIR Filter structures
  - Direct form I and Direct form II