Chapter 8
Digital Filter Structures

Zhiliang Liu

Zhiliang_Liu@uestc.edu.cn

4/25/2019

8.1 Block Diagram Representation

Advantages of block diagram representation

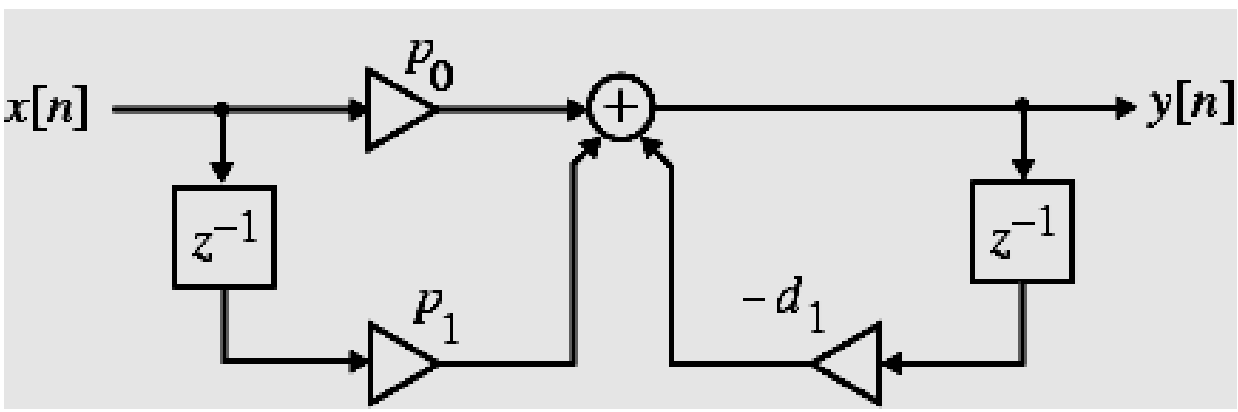
- **(1) Easy to write down the <u>computational</u> algorithm by inspection**
- **(2) Easy to analyze the block diagram to determine the explicit <u>relation between the output and input</u>**

8.1 Block Diagram Representation

– (3) Easy to manipulate a block diagram to derive other "**equivalent**" block diagrams yielding different computational algorithms
– (4) Easy to determine the **hardware requirements**
– (5) Easier to develop block diagram representations from the **transfer function** directly

8.1 Block Diagram Representation

- **For the implementation of an LTI digital filter, the input-output relationship must be described by a valid computational algorithm**
- **To illustrate what we mean by a computational algorithm, consider the causal first-order LTI digital filter shown below**



8.1 Block Diagram Representation

- **The filter is described by the difference equation**

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

- **Using the above equation we can compute $y[n]$ for $n \geq 0$ knowing the initial condition $y[n-1]$ and the input $x[n]$ for $n \geq -1$**

8.1 Block Diagram Representation

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

....

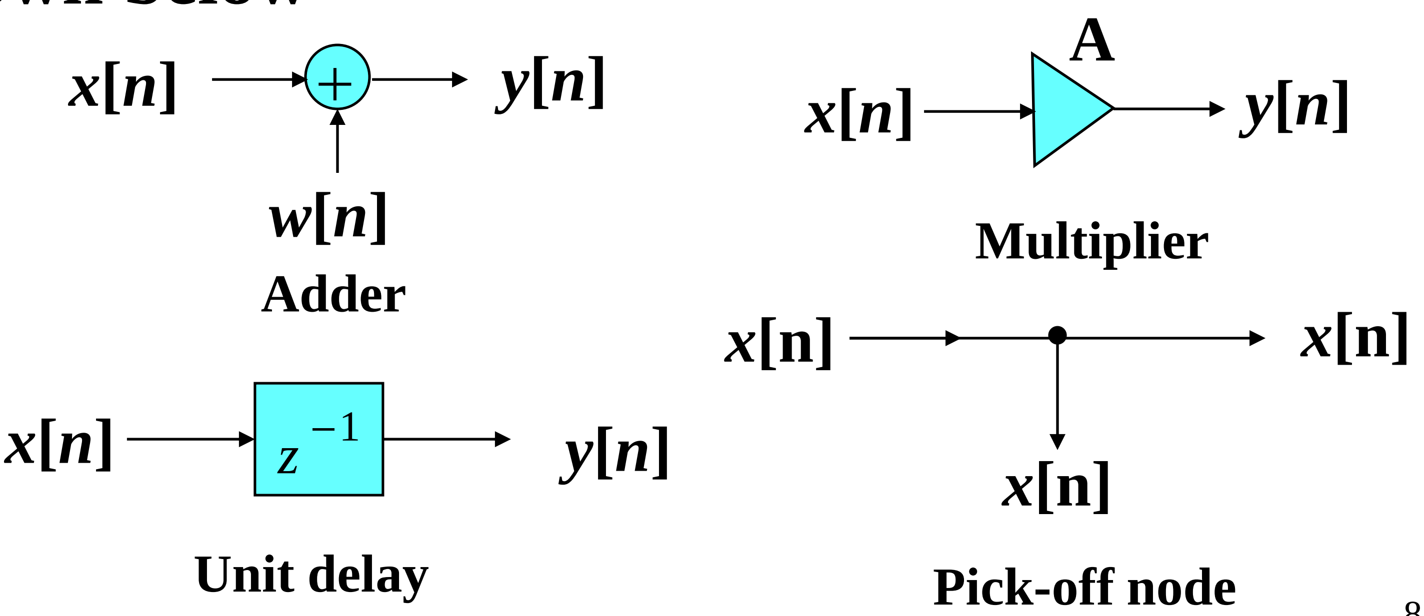
- **We can continue this calculation for any value of the time index *n* we desire**

8.1 Block Diagram Representation

- **Each step of the calculation requires a knowledge of the previously calculated value of the output sample (delayed value of the output), the present value of the input sample, and the previous value of the input sample (delayed value of the input)**
- **As a result, the first-order difference equation can be interpreted as a valid computational algorithm**

§ 8.1.1 Basic Building Blocks

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

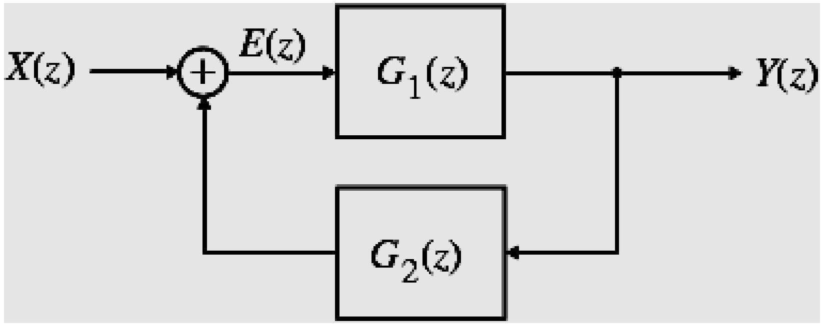


8.1.2 Analysis of Block Diagrams

- **Carried out by writing down the expressions for the output signals of each adder as a sum of its input signals, and developing a set of equations relating the filter input and output signals in terms of all internal signals**
- **Eliminating the unwanted internal variables then results in the expression for the output signal as a function of the input signal and the filter parameters that are the multiplier coefficients**

8.1.2 Analysis of Block Diagrams

Example **- Consider the single-loop feedback structure shown below**



The output $E(z)$ of the adder is

$$E(z) = X(z) + G_2(z)Y(z)$$

- **But from the figure**

$$Y(z) = G_1(z)E(z)$$

8.1.2 Analysis of Block Diagrams

- **Eliminating $E(z)$ from the previous two equations we arrive at**

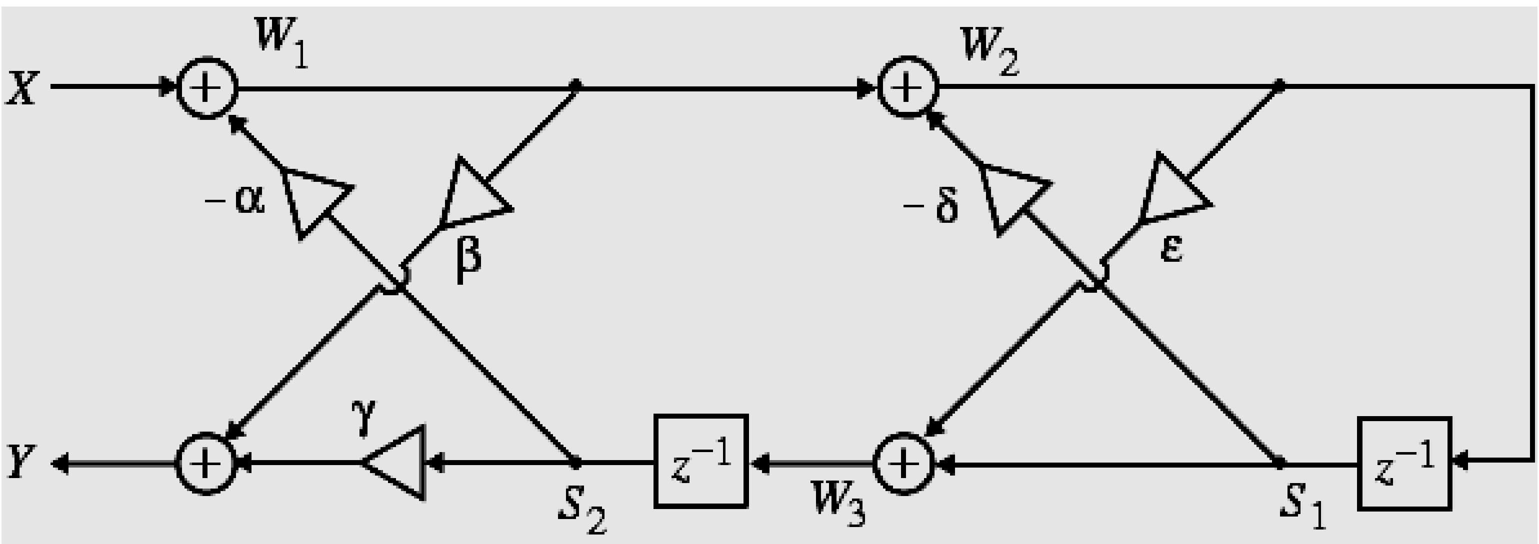
$$[1-G_1(z)G_2(z)]Y(z) = G_1(z)X(z)$$

which leads to

$$H(z) = \frac{Y(z)}{X(z)} = \frac{G_1(z)}{1 - G_1(z)G_2(z)}$$

8.1.2 Analysis of Block Diagrams

- ***Example*** **- Analyze the cascaded lattice structure shown below where the *z*-dependence of signal variables are not shown for brevity**



8.1.2 Analysis of Block Diagrams

- **The output signals of the four adders are given by**

$$W_1 = X - \alpha S_2 \qquad W_2 = W_1 - \delta S_1$$

$$W_3 = S_1 - \varepsilon W_2 \qquad Y = \beta W_1 - \gamma S_2$$

- **From the figure we observe**

$$S_2 = z^{-1} W_3 \qquad S_1 = z^{-1} W_2$$

8.1.2 Analysis of Block Diagrams

- Substituting the last two relations in the first four equations we get

$$W_1 = X - \alpha z^{-1} W_3 \qquad W_2 = W_1 - \delta z^{-1} W_2$$

$$W_3 = z^{-1} W_2 + \varepsilon W_2 \qquad Y = \beta W_1 + \gamma z^{-1} W_3$$

- From the second equation we get $W_2 = W_1 / (1 + \delta z^{-1})$ and from the third equation we get $W_3 = (\varepsilon + z^{-1}) W_2$

8.1.2 Analysis of Block Diagrams

- **Combining the last two equations we get**

$$W_3 = \frac{\varepsilon + z^{-1}}{1 + \delta z^{-1}} W_1$$

- **Substituting the above equation in**

$$W_1 = X - \alpha z^{-1} W_3, \qquad Y = \beta W_1 + \gamma z^{-1} W_3$$

we finally arrive at

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

8.1.2 Analysis of Block Diagrams

***Note*:**

***Delay-free loop*: feedback loops without any delay elements (physically impossible).**

***canonic structure*: The number of delays is equal to the order of the difference equation.**

8.2 Equivalent Structures

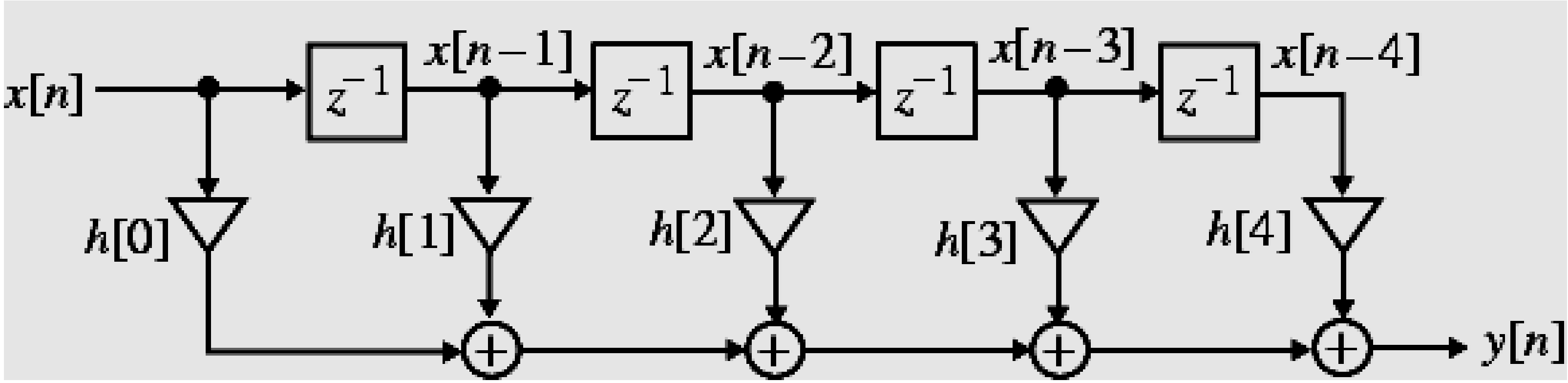
- **Two digital filter structures are defined to be equivalent if they have the same transfer function**
- **We describe next a number of methods for the generation of equivalent structures**
- **However, a fairly simple way to generate an equivalent structure from a given realization is via the *transpose operation***

8.2 Equivalent Structures

Transpose Operation

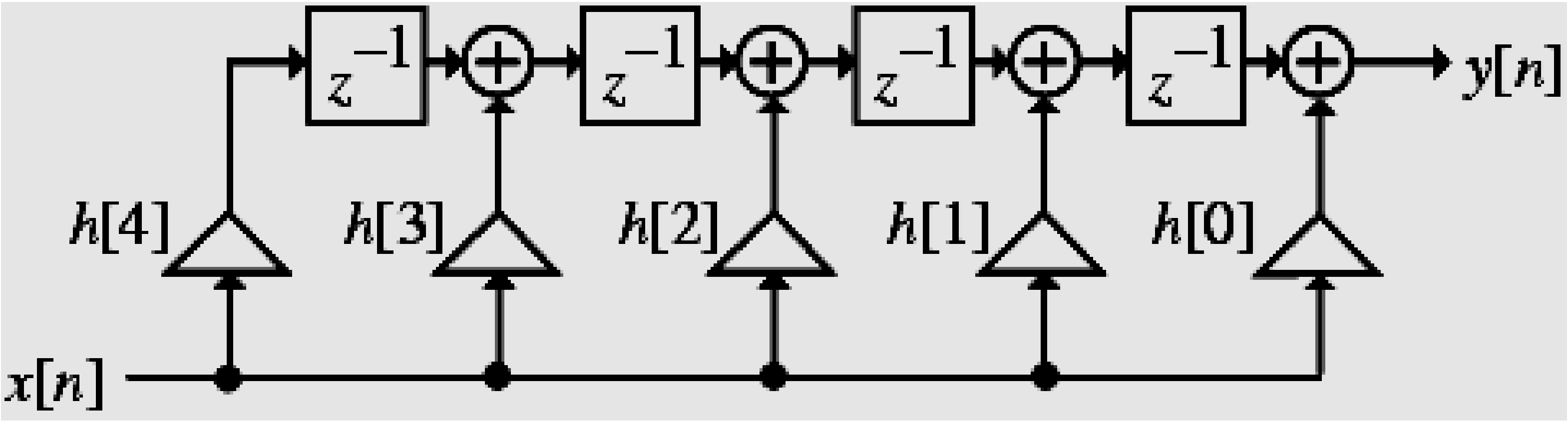
- (1) Reverse all paths
- (2) Replace pick-off nodes by adders, and vice versa
- (3) Interchange the input and output nodes

***Example*:** **Original structure**



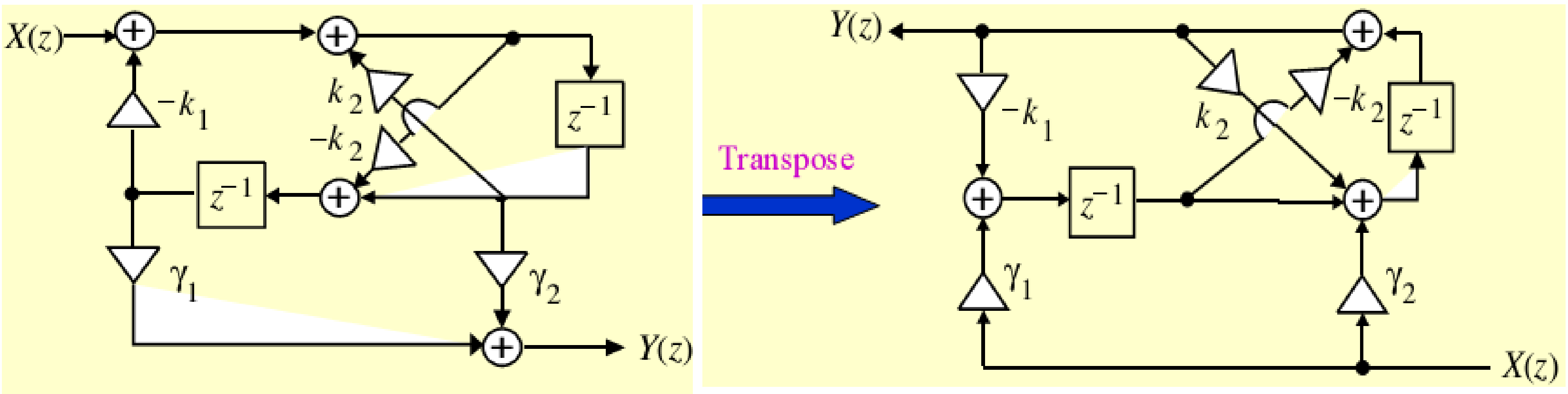
8.2 Equivalent Structures

Transposed structure



8.2 Equivalent Structures

- ***Example***: The transpose operation is illustrated below

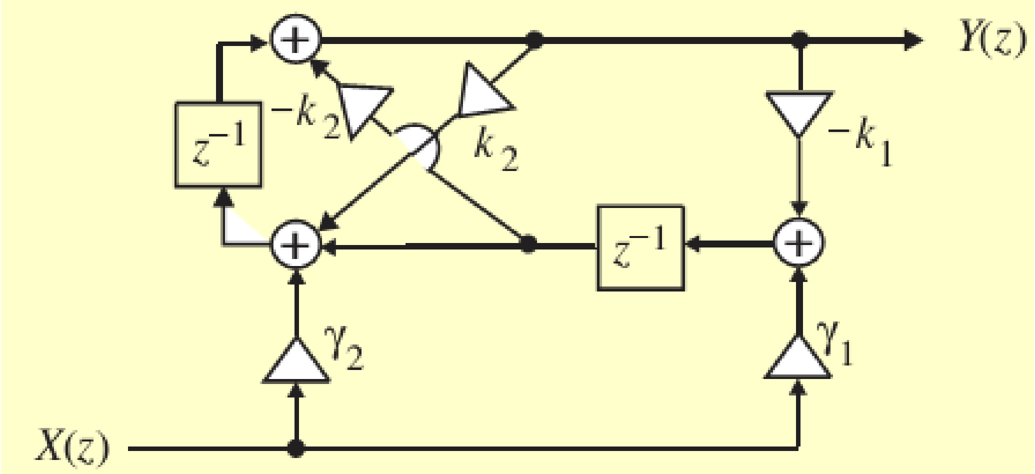


Original Structure

Transposed Structure

8.2 Equivalent Structures

- **Redrawn transposed structure is shown below**



All other methods for developing equivalent structures are based on *a specific algorithm for each structure*

8.2 Equivalent Structures

- **There are literally an infinite number of equivalent structures realizing the same transfer function**
- **It is thus impossible to develop all equivalent realizations**
- **In this course we restrict our attention to a discussion of some commonly used structures**

8.2 Equivalent Structures

- **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 behaves totally differently from its other equivalent realizations**

8.2 Equivalent Structures

- **Hence, it is important to choose a structure that has the *least quantization effects* when implemented using finite precision arithmetic**
- **One way to arrive at such a structure is to determine a large number of equivalent structures, *analyze* the finite wordlength effects in each case, *and select* the one showing the least effects**

8.2 Equivalent Structures

- **In certain cases, it is possible to develop a structure that by construction has the least quantization effects**
- **We defer the review of these structures after a discussion of the analysis of quantization effects**
- **Here, we review some simple realizations that in many applications are quite adequate**

8.3 Basic FIR Digital Filter Structures

- **A causal FIR filter of order *N* is characterized by a transfer function *H*(*z*) given by**

$$H(z) = \sum_{n=0}^{N} h[n] z^{-n}$$

which is a polynomial in z^{-1}

- **In the time-domain the input-output relation of the above FIR filter is given by**

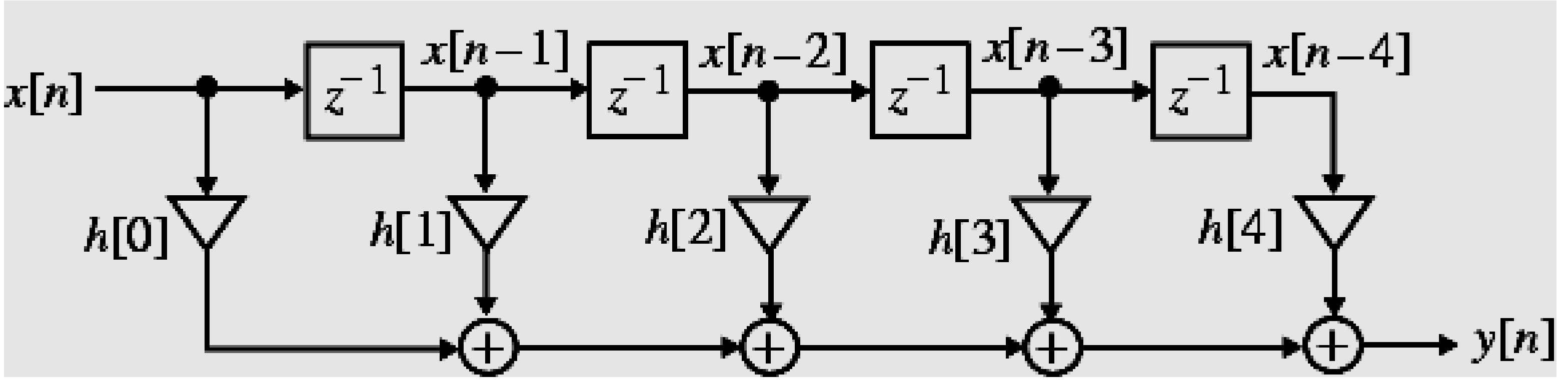
$$y[n] = \sum_{k=0}^{N} h[k] x[n-k]$$

8.3.1 Direct Form FIR Digital Filter Structures

- **An FIR filter of order N is characterized by N+1 coefficients and, in general, require N+1 multipliers and N two-input adders**
- **Structures in which the multiplier coefficients are precisely the coefficients of the transfer function are called direct form structures**

8.3 Basic FIR Digital FilterStructures

- **A direct form realization --- convolution sum description as indicated below for *N* = 4**



8.3 Basic FIR Digital FilterStructures

- **An analysis of this structure yields**

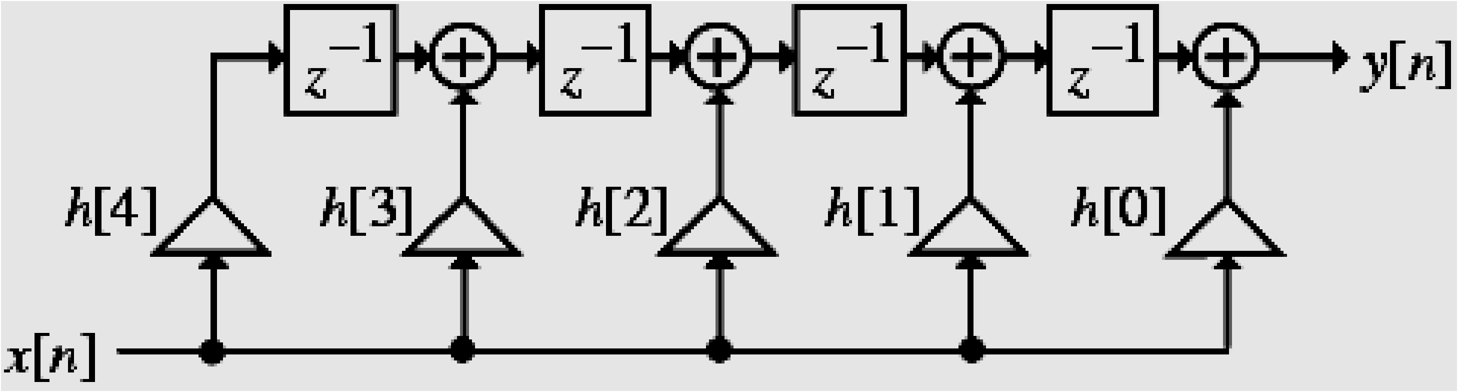
$$y[n] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2]$$
$$+ h[3]x[n-3] + h[4]x[n-4]$$

which is precisely of the form of the convolution sum description

- **The direct form structure shown on the previous slide is also known as a <u>tapped delay line</u> or a <u>transversal filter</u>**

8.3 Basic FIR Digital FilterStructures

- **The transpose of the direct form structure shown earlier is indicated below**



8.3.2 Cascade Form FIR Digital Filter Structures

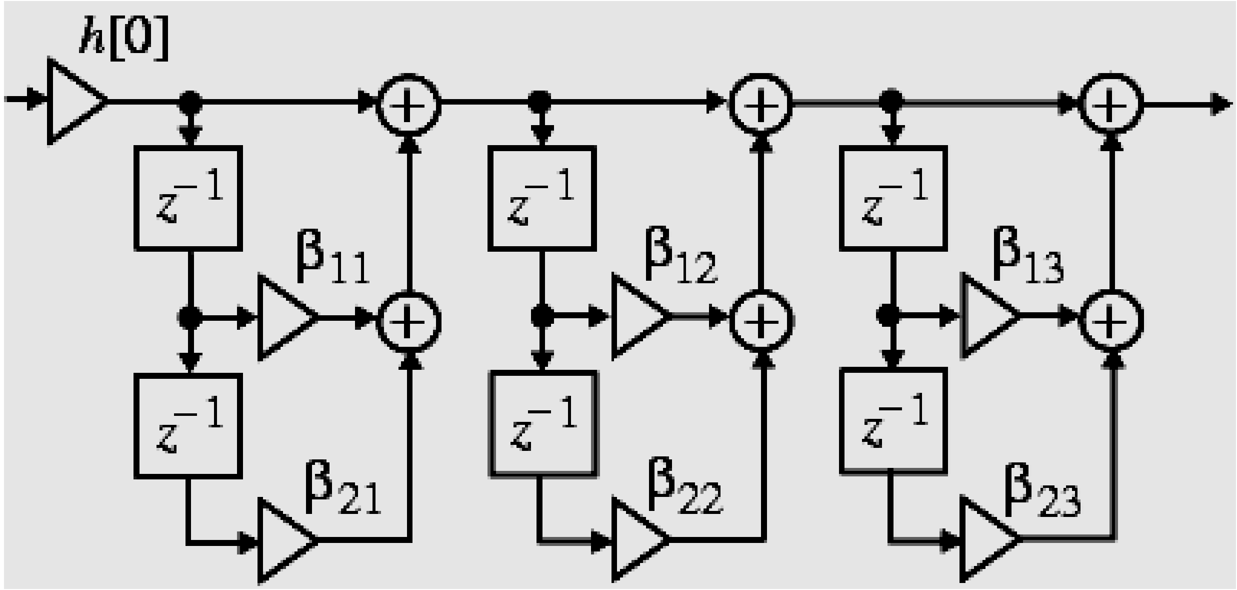
- **A higher-order FIR transfer function can also be realized as a cascade of second-order FIR sections and possibly a first-order section**
- **To this end we express *H*(*z*) as**

$$H(z) = h[0]\prod_{k=1}^{K}(1+\beta_{1k}z^{-1}+\beta_{2k}z^{-2})$$

where $K = N/2$ if N is even, and $K = (N+1)/2$ if N is odd, with $\beta_{2K} = 0$

8.3.2 Cascade Form FIR Digital Filter Structures

- **A cascade realization for *N* = 6 is shown below**



Each second-order section in the above structure can also be realized in the direct form

8.3.3 Linear-Phase FIR Structures

- **The symmetry (or anti-symmetry) property of a linear-phase FIR filter can be exploited to reduce the number of multipliers into almost half of that in the direct form implementations**
- **Consider a length-7 Type 1 FIR transfer function with a symmetric impulse response**

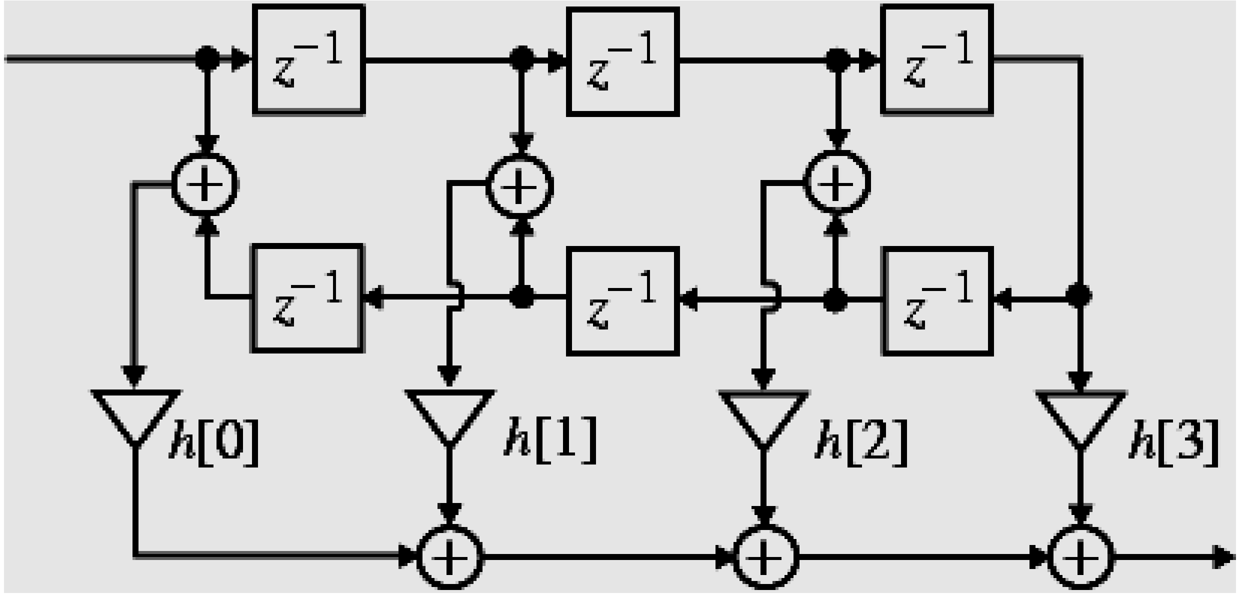
$$H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3}$$
$$+ h[2]z^{-4} + h[1]z^{-5} + h[0]z^{-6}$$

8.3.3 Linear-Phase FIR Structures

- **Rewriting *H*(*z*) in the form**

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

we obtain the realization shown below



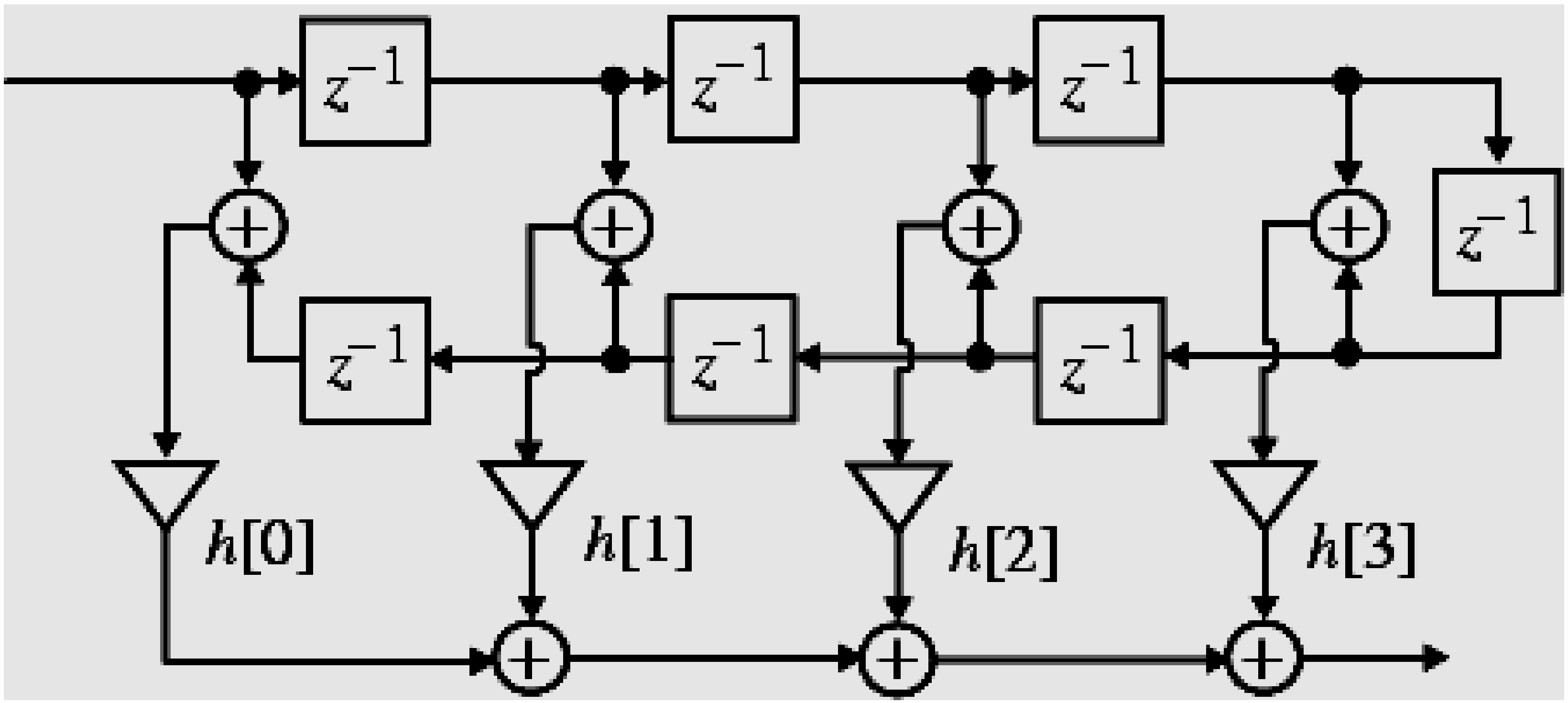
8.3.3 Linear-Phase FIR Structures

- **A similar decomposition can be applied to a Type 2 FIR transfer function**
- **For example, a length-8 Type 2 FIR transfer function can be expressed as**

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

- **The corresponding realization is shown on the next slide**

8.3.3 Linear-Phase FIR Structures



- **Note: The Type 1 linear-phase structure for a length-7 FIR filter requires 4 multipliers, whereas a direct form realization requires 7 multipliers**

8.3.3 Linear-Phase FIR Structures

- **Note: The Type 2 linear-phase structure for a length-8 FIR filter requires 4 multipliers, whereas a direct form realization requires 8 multipliers**
- **Similar savings occurs in the realization of Type 3 and Type 4 linear-phase FIR filters with anti-symmetric impulse responses**

8.4 Basic IIR Digital Filter Structures

- **The causal IIR digital filters we are concerned with in this course are characterized by a real rational transfer function of z^{-1} or, equivalently by a constant coefficient difference equation**
- **From the difference equation representation, it can be seen that the realization of the causal IIR digital filters requires some form of <u>feedback</u>**

8.4 Basic IIR Digital Filter Structures

- **An N-th order IIR digital transfer function is characterized by 2N+1 unique coefficients, and in general, requires 2N+1 multipliers and 2N two-input adders for implementation**
- **Direct form IIR filters: Filter structures in which the multiplier coefficients are precisely the coefficients of the transfer function**

8.4.1 Direct Form IIR Digital Filter Structures

- **Consider for simplicity a 3rd-order IIR filter with a transfer function**

$$H(z) = \frac{P(z)}{D(z)} = \frac{p_0 + p_1 z^{-1} + p_2 z^{-2} + p_3 z^{-3}}{1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}}$$

  - **We can implement *H*(*z*) as a cascade of two filter sections as shown on the next slide**

8.4.1 Direct Form IIR Digital Filter Structures

$$H_1(z) = \frac{W(z)}{X(z)} = P(z) = p_0 + p_1 z^{-1} + p_2 z^{-2} + p_3 z^{-3}$$

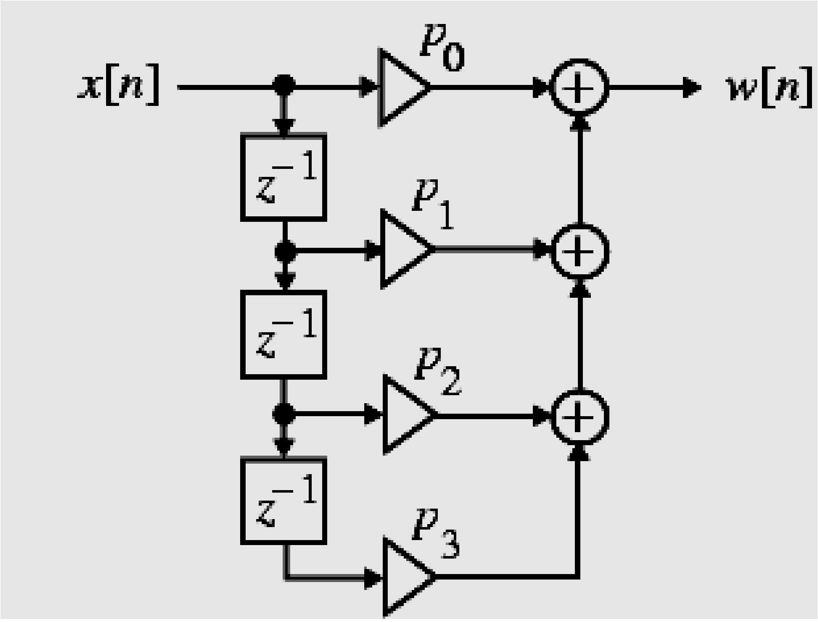
$$H_2(z) = \frac{Y(z)}{W(z)} = \frac{1}{D(z)} = \frac{1}{1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}}$$

- **The time-domain representation of *$H_1(z)$* is given by**

$$w[n] = p_0 x[n] + p_1 x[n-1] + p_2 x[n-2] + p_3 x[n-3]$$

8.4.1 Direct Form IIR Digital Filter Structures

- **The filter section $H_1(z)$ can be seen to be an FIR filter and can be realized as shown right**

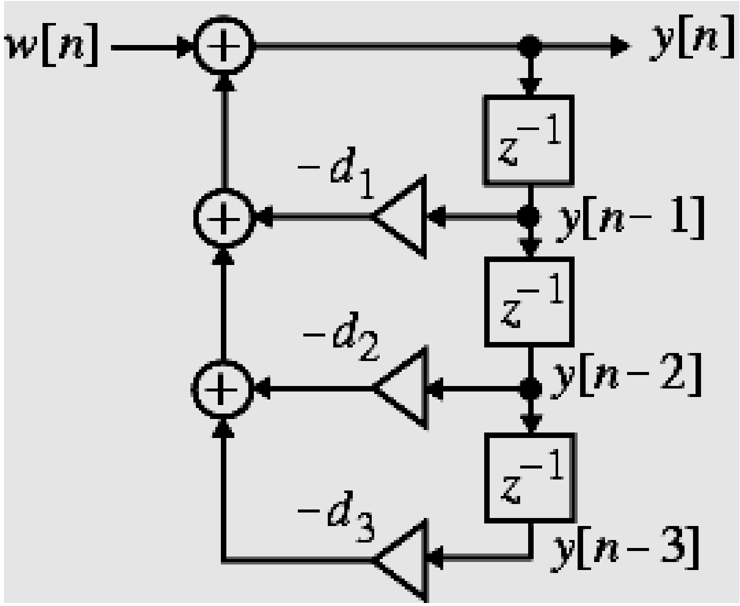


- **The time-domain representation of $H_2(z)$ is given by**

$$y[n] = w[n] - d_1 y[n-1] - d_2 y[n-2] - d_3 y[n-3]$$

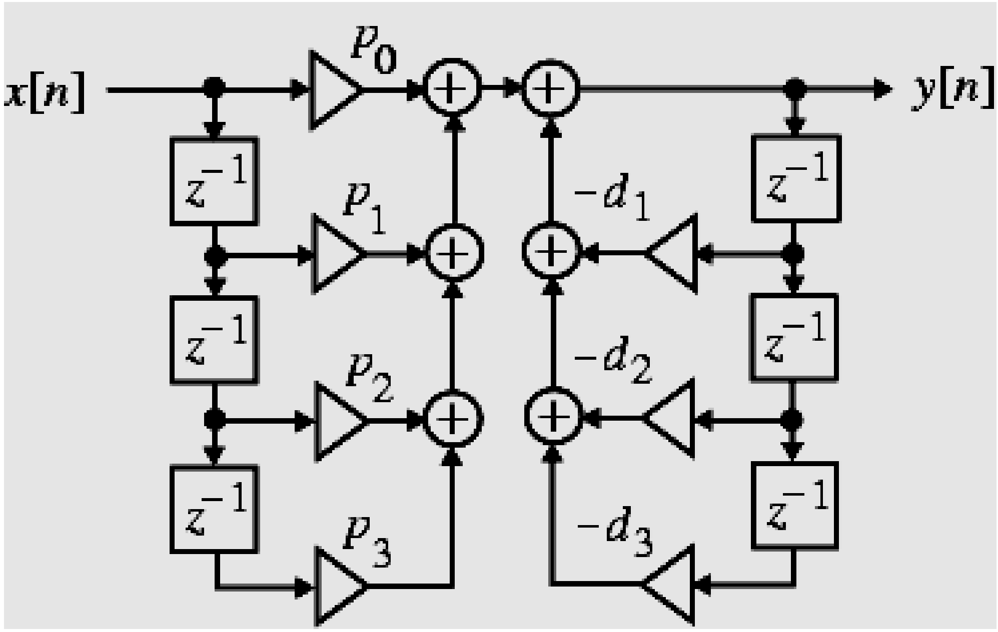
8.4.1 Direct Form IIR Digital Filter Structures

- **Realization of follows from the above equation and is shown on the right**



- **A cascade of the two structures realizing $H_1(z)$ and $H_2(z)$ leads to the realization of $H(z)$ shown below and is known as the <u>Direct Form I</u> structure**

8.4.1 Direct Form IIR Digital Filter Structures

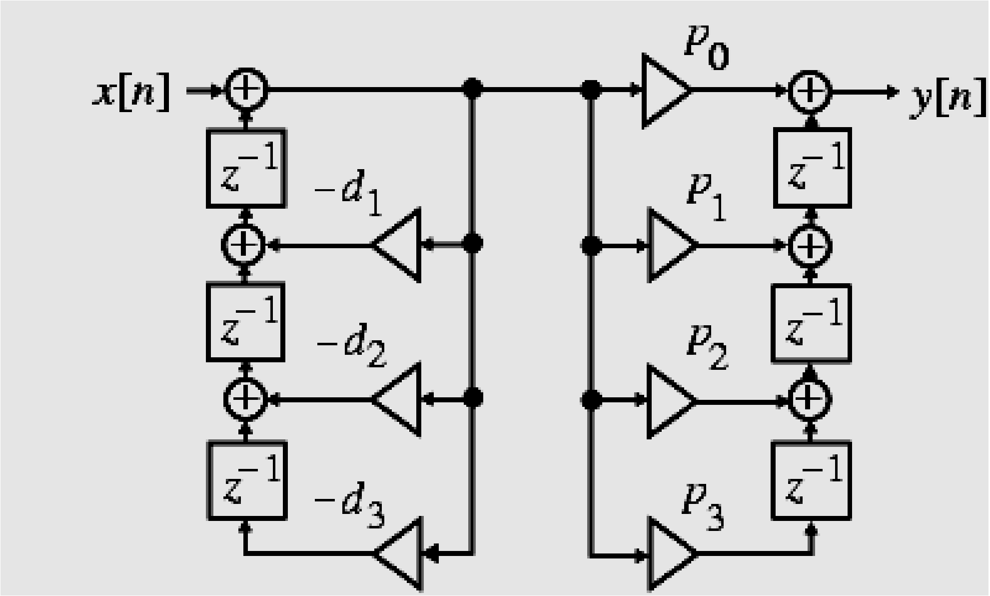


Direct Form I

- **Note: The direct form I structure is <u>noncanonic</u> as it employs 6 delays to realize a 3rd-order transfer function**

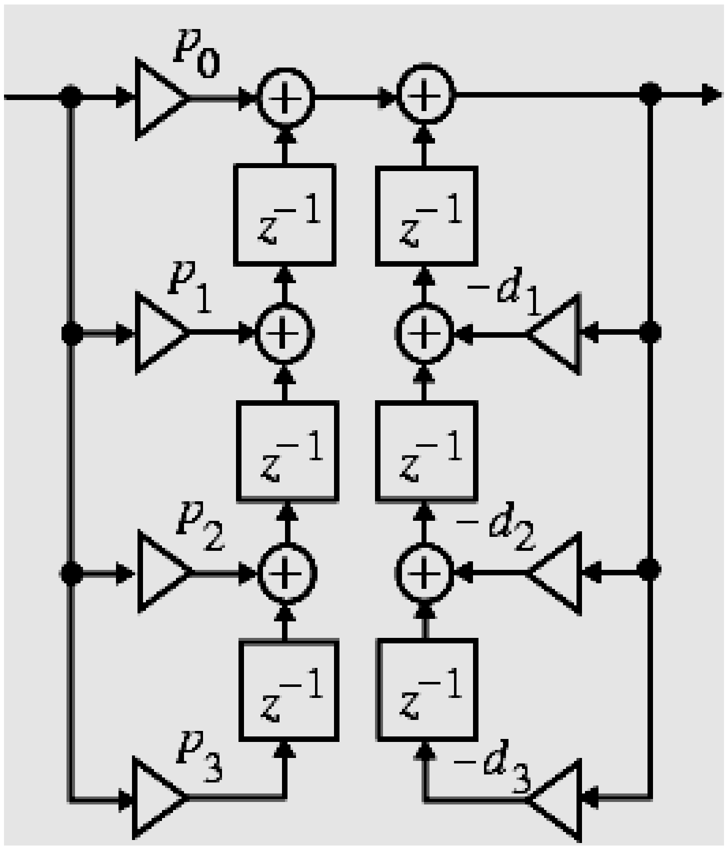
8.4.1 Direct Form IIR Digital Filter Structures

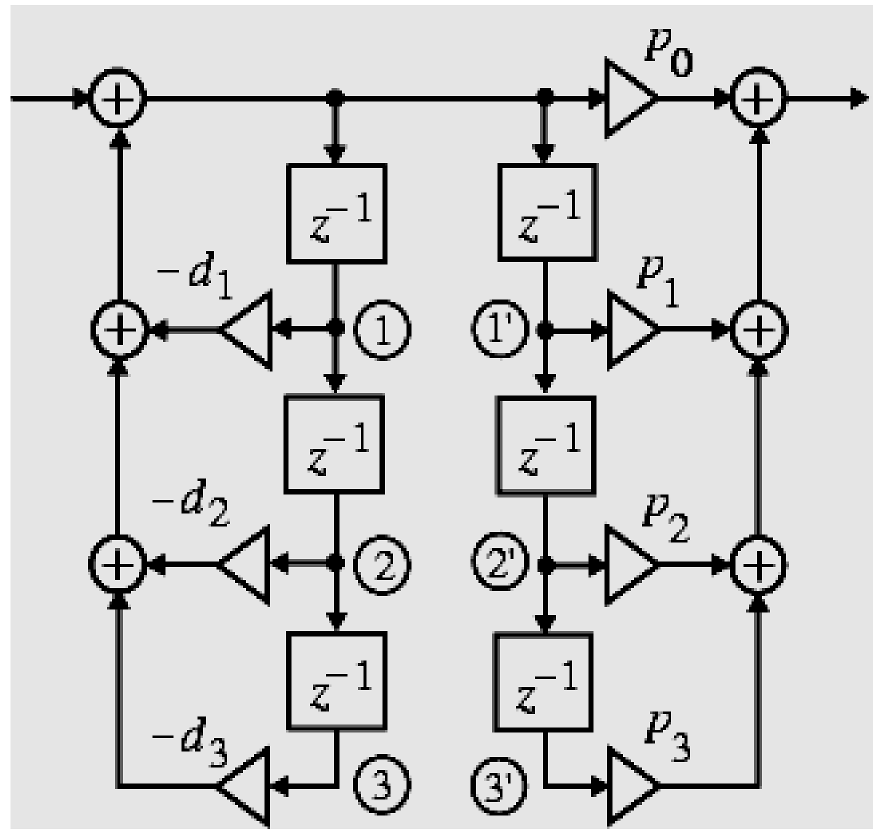
- **A transpose of the direct form I structure is shown on the following figure and is called the direct form It structure**



direct form It

8.4.1 Direct Form IIR Digital Filter Structures

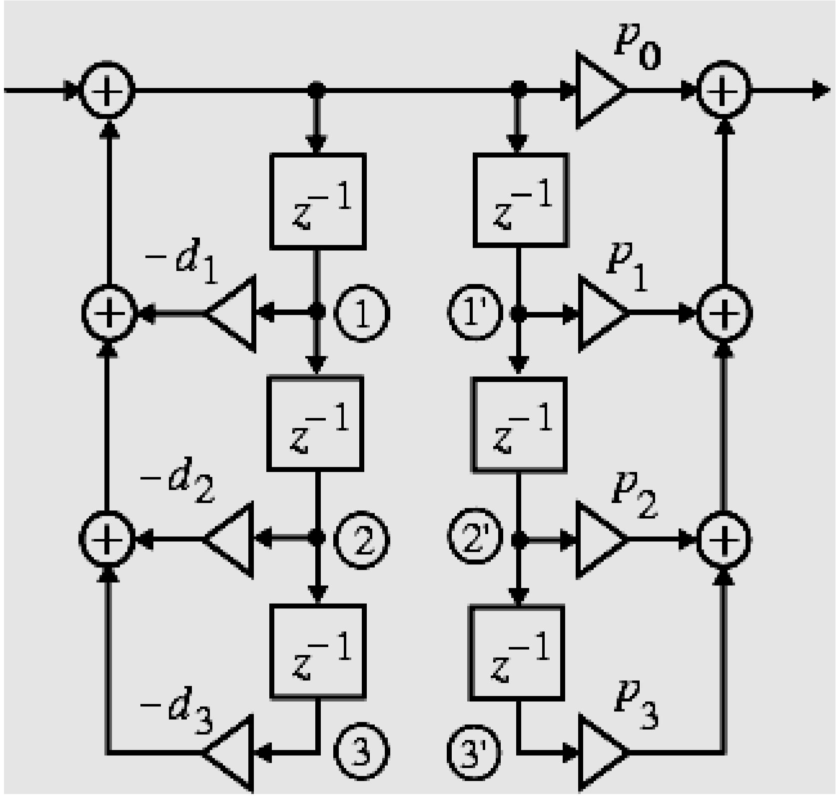




Other noncanonic direct form structures

8.4.1 Direct Form IIR Digital Filter Structures

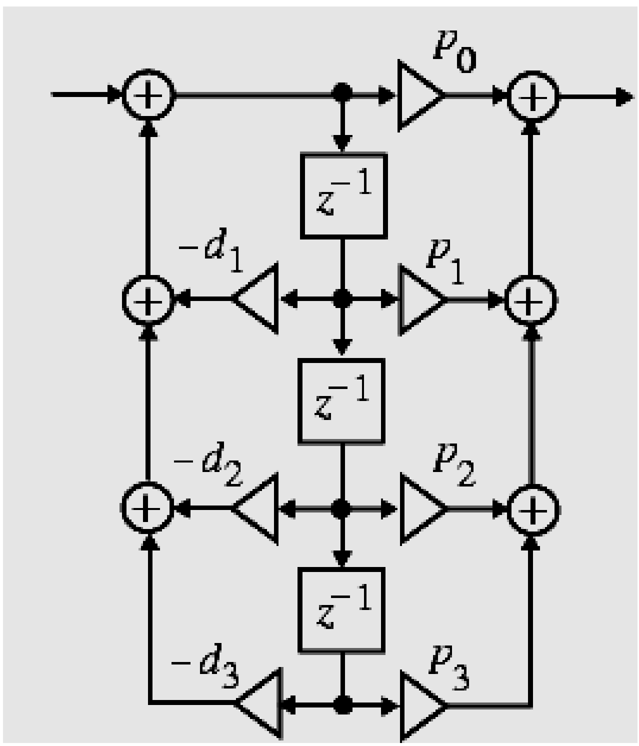
Observe in the direct form structure shown right, the signal variable at nodes 1 and 1' are the same, and hence the two top delays can be shared



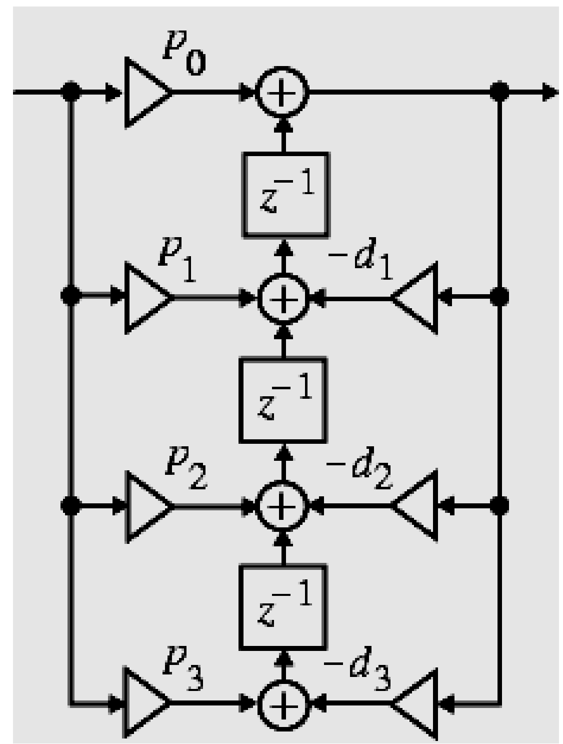
- Likewise, the signal variables at nodes 2 and 2' are the same, permitting the sharing of the middle two delays

8.4.1 Direct Form IIR Digital Filter Structures

Sharing of all delays reduces the total number of delays to 3 resulting in a canonic realization shown in following



Direct form II



Direct form II$_t$

8.4.1 Direct Form IIR Digital Filter Structures

- **It is noted that, Direct form realizations of an N-th order IIR transfer function should be evident from above 3-order structure.**

8.4.2 Cascade Form IIR Digital Filter Structures

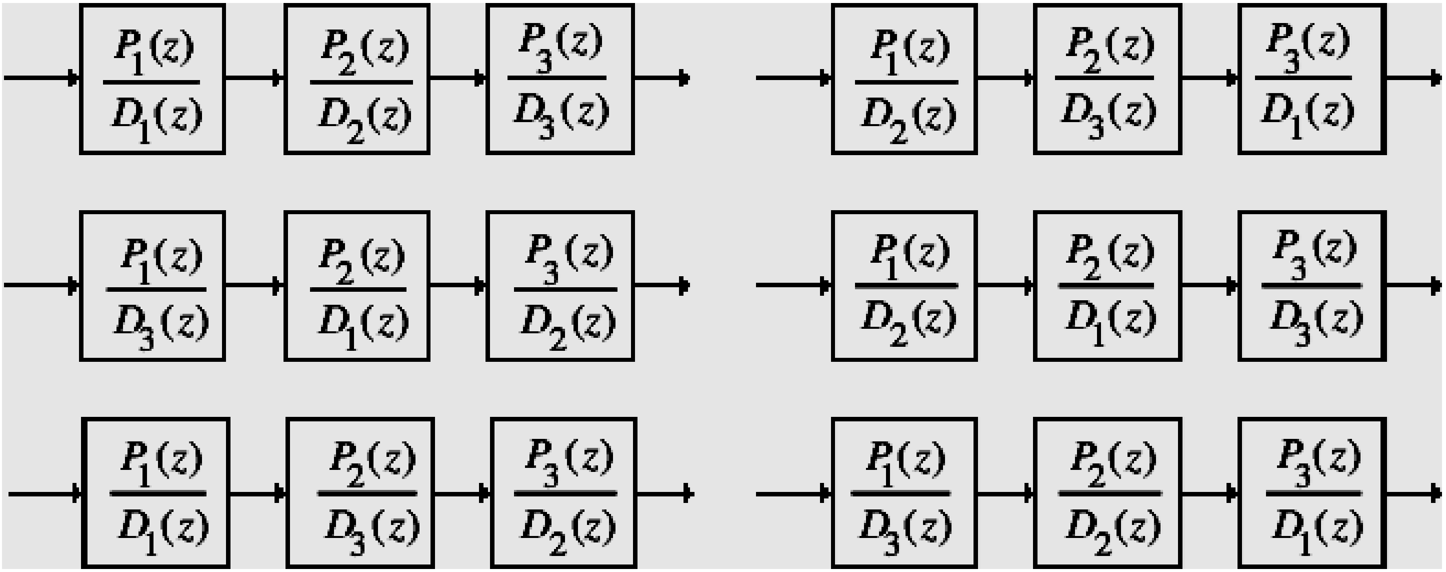
- **By expressing the numerator and the denominator polynomials of the transfer function as a product of polynomials of lower degree, a digital filter can be realized as a cascade of low-order filter sections**

- **Consider, for example, *H*(*z*) = *P*(*z*)/*D*(*z*) expressed as**

$$H(z) = \frac{P(z)}{D(z)} = \frac{P_1(z)P_2(z)P_3(z)}{D_1(z)D_2(z)D_3(z)}$$

8.4.2 Cascade Form IIR Digital Filter Structures

- **Examples of cascade realizations obtained by different pole-zero pairings are shown below**



8.4.2 Cascade Form IIR Digital Filter Structures

- **There are altogether a total of 36 different cascade realizations of**

$$H(z) = \frac{P_1(z)P_2(z)P_2(z)}{D_1(z)D_2(z)D_3(z)}$$

based on <u>pole-zero-pairings</u> and <u>ordering</u>.

Due to finite word-length effects, each such cascade realization behaves differently from others.

8.4.2 Cascade Form IIR Digital Filter Structures

- **Usually, the polynomials are factored into a product of 1st-order and 2nd-order polynomials:**

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

- **In the above, for a first-order factor**

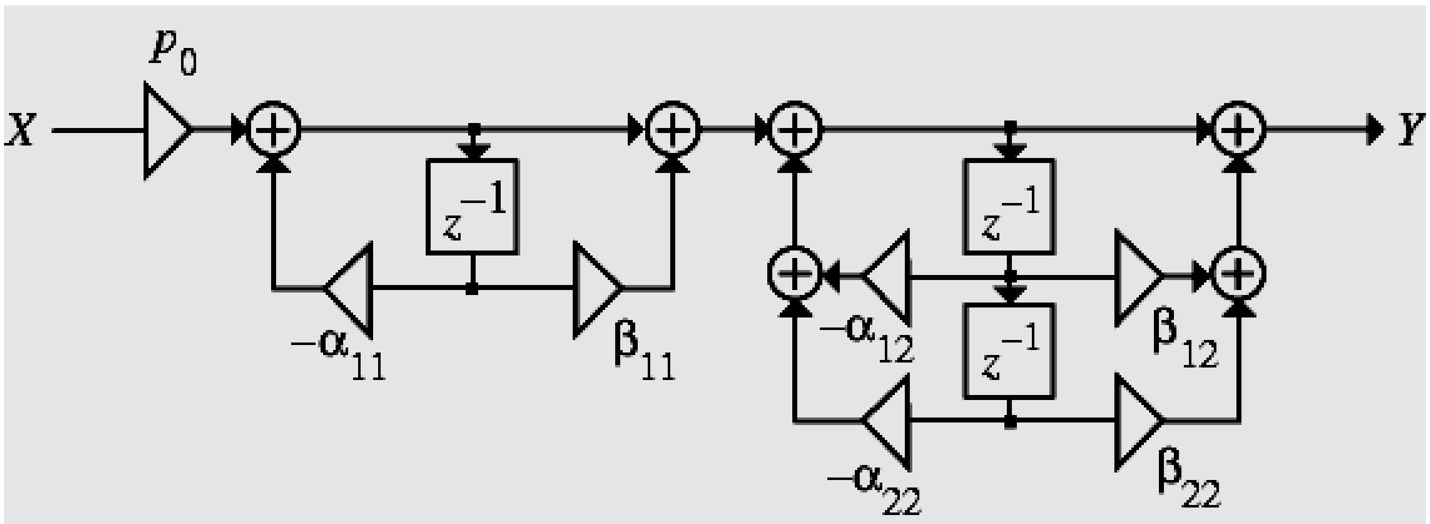
$$\alpha_{2k} = \beta_{2k} = 0$$

8.4.2 Cascade Form IIR Digital Filter Structures

- **Consider the 3rd-order transfer function**

$$H(z) = p_0 \left(\frac{1+\beta_{11} z^{-1}}{1+\alpha_{11} z^{-1}} \right) \left(\frac{1+\beta_{12} z^{-1} + \beta_{22} z^{-2}}{1+\alpha_{12} z^{-1} + \alpha_{22} z^{-2}} \right) \quad \textbf{(8.28)}$$

- **One possible realization is shown below**



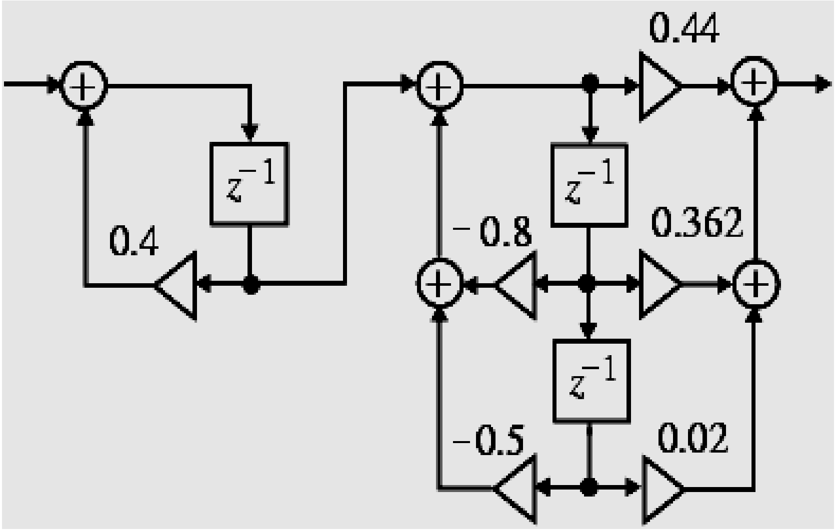
8.4.2 Cascade Form IIR Digital Filter Structures

- ***Example*** **- Direct form II and cascade form realizations of**

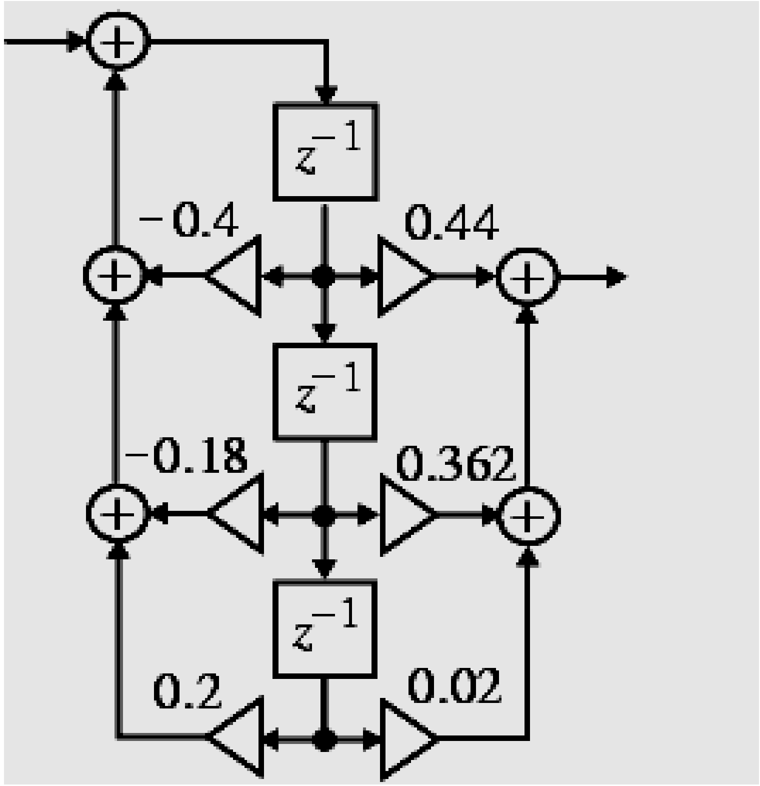
$$H(z) = \frac{0.44 z^{-1} + 0.362 z^{-2} + 0.02 z^{-3}}{1 + 0.4 z^{-1} + 0.18 z^{-2} - 0.2 z^{-3}}$$
$$= \left(\frac{0.44 + 0.362 z^{-1} + 0.02 z^{-2}}{1 + 0.8 z^{-1} + 0.5 z^{-2}} \right) \left(\frac{z^{-1}}{1 - 0.4 z^{-1}} \right) \quad \textbf{(8.29)}$$

are shown on the next slide

8.4.2 Cascade Form IIR Digital Filter Structures



Cascade form



Direct form II

8.4.3 Parallel Form IIR Digital Filter Structures

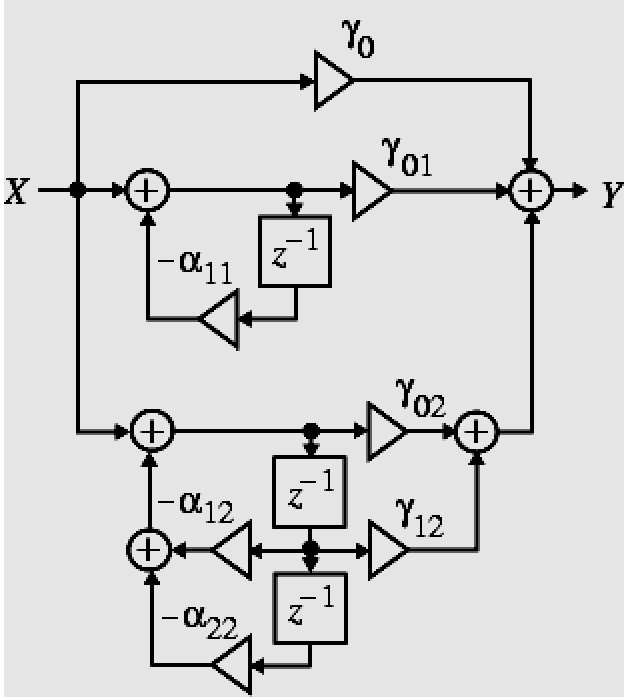
- **A partial-fraction expansion of the transfer function in z^{-1} leads to the parallel form I structure**
- **Assuming simple poles, the transfer function *H*(*z*) can be expressed as**

$$H(z) = \gamma_0 + \sum_k \left(\frac{\gamma_{0k} + \gamma_{1k} z^{-1}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \right) \qquad \textbf{(8.30)}$$

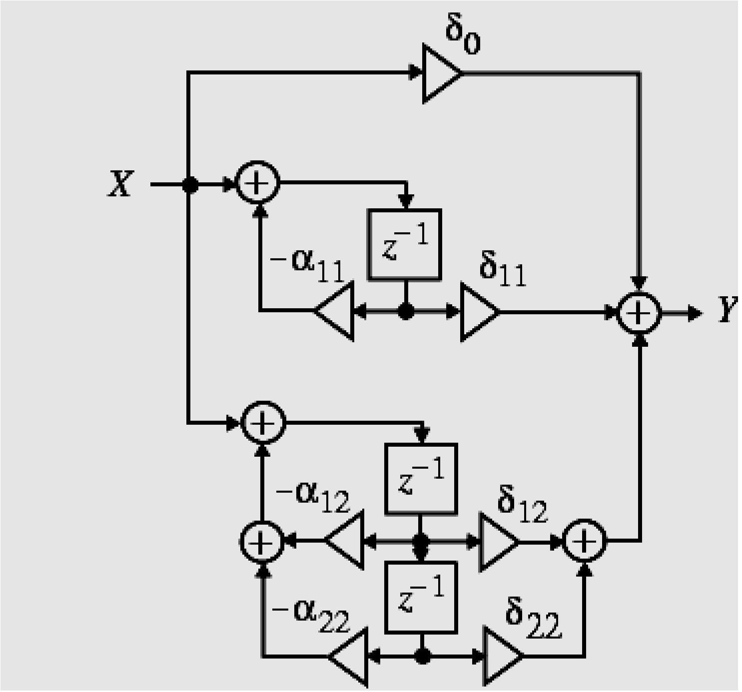
$$H(z) = \delta_0 + \sum_k \left(\frac{\delta_{0k} z^{-1} + \delta_{1k} z^{-2}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \right) \qquad \textbf{(8.31)}$$

8.4.3 Parallel Form IIR Digital Filter Structures

- **The two basic parallel realizations of a 3rd-order IIR transfer function are shown below**



Parallel form I



Parallel form II

8.4.3 Parallel Form IIR Digital Filter Structures

- ***Example*** **- A partial-fraction expansion of**

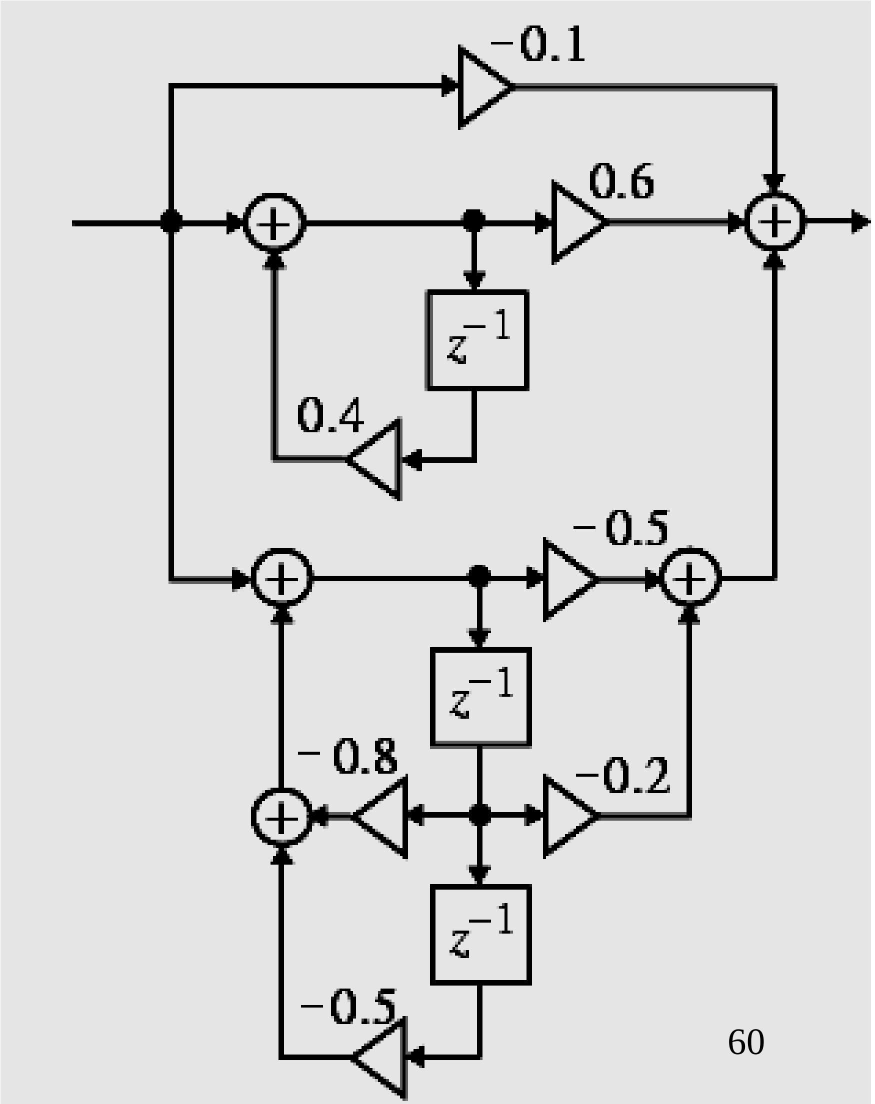
$$H(z) = \frac{0.44z^{-1} + 0.362z^{-2} + 0.02z^{-3}}{1 + 0.4z^{-1} + 0.18z^{-2} - 0.2z^{-3}}$$

in z^{-1} yields

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

8.4.3 Parallel Form IIR Digital Filter Structures

- **The corresponding parallel form I realization is shown right**

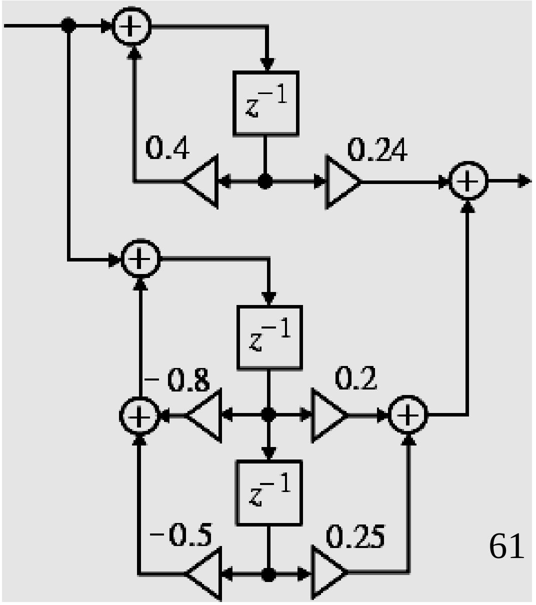


8.4.3 Parallel Form IIR Digital Filter Structures

- **Likewise, a partial-fraction expansion of *H*(*z*) in *z* yields**

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

- **The corresponding parallel form II realization is shown on the right**



8.5 Realization Using MATLAB

- **The cascade form requires the factorization of the transfer function which can be developed using the M-file zp2sos**
- **The statement sos = zp2sos(z, p, k) generates a matrix sos containing the coefficients of each 2nd-order section of the equivalent transfer function $H(z)$ determined from its pole-zero form**

•**sos** **is an** $L\times 6$ **matrix of the form**

$$sos = \begin{bmatrix} p_{01} & p_{11} & p_{21} & d_{01} & d_{11} & d_{21} \\ p_{02} & p_{12} & p_{22} & d_{02} & d_{12} & d_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{0L} & p_{1L} & p_{2L} & d_{0L} & d_{1L} & d_{2L} \end{bmatrix}$$

whose ***i*****-th row contains the coefficients** $\{p_{il}\}$ **and** $\{d_{il}\}$**, of the the numerator and denominator polynomials of the** ***i*****-th 2nd-order section**

- L **denotes the number of sections**

- **The form of the overall transfer function is given by**

$$H(z) = \prod_{i=1}^{L} H_i(z) = \prod_{i=1}^{L} \frac{p_{0i} + p_{1i} z^{-1} + p_{2i} z^{-2}}{d_{0i} + d_{1i} z^{-1} + d_{2i} z^{-2}}$$

Program 8_1 can be used to factorize an FIR and an IIR transfer function

- **Parallel forms I and II can be developed using the functions residuez and residue, respectively**
- **Program 8_2 uses these two functions**

Thanks!

Any questions?