Realization of Digital Systems

(UNIT-1)

Unit-1: Realization of Digital Systems

- Introduction
- Block Diagram Representation
- Equivalent Structures
- Basic FIR Digital Filter Structures
- Basic IIR Digital Filter Structures
- FIR Cascaded Lattice Structures

1.1 Introduction

- The convolution sum description of an LTI discrete-time system can, in principle, be used to implement the system
- For an IIR finite-dimensional system this approach is not practical as here the impulse response is of infinite length
- Here the input-output relation involves a finite sum of products:

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

Introduction

- The actual implementation of an LTI digital filter can be either in software or hardware form, depending on applications
- In either case, the signal variables and the filter coefficients cannot be represented with infinite precision

Introduction

- However, a direct implementation of a digital filter based on either the difference equation or the finite convolution sum may not provide satisfactory performance due to the finite precision arithmetic
- It is thus of practical interest to develop alternate realizations and choose the structure that provides satisfactory performance under finite precision arithmetic

Introduction

- A structural representation using interconnected basic building blocks is the first step in the hardware or software implementation of an LTI digital filter
- The structural representation provides the key relations between some pertinent internal variables with the input and output that in turn provides the key to the implementation

1.2 Block Diagram Representation

- In the time domain, the input-output relations of an LTI digital filter is given by the convolution sum

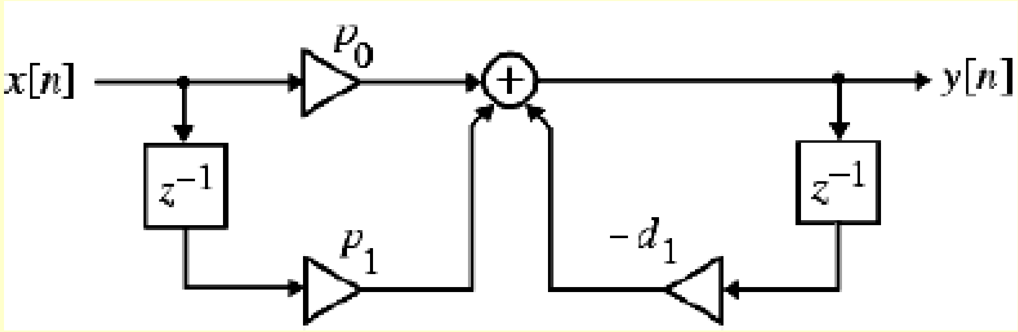
$$y[n]=\sum_{k=-\infty}^{\infty} h[k]x[n-k]$$

or, by the linear constant coefficient difference equation

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

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



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 $y[-1]$ and the input $x[n]$ for $n \geq -1$

Block Diagram Representation

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

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

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

$$\vdots$$

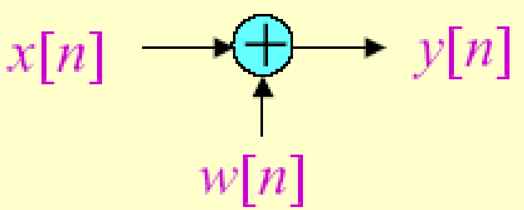
- We can continue this calculation for any value of the time index n we desire

Block Diagram Representation

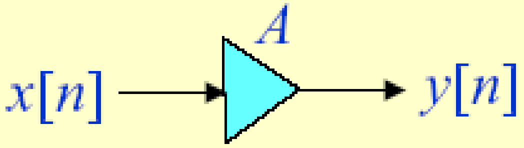
- Each step of the calculation requires a knowledge of the previously calculated value of the output sample (delayed value of the output), the present value of the input sample, and the previous value of the input sample (delayed value of the input)
- As a result, the first-order difference equation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

Basic Building Blocks

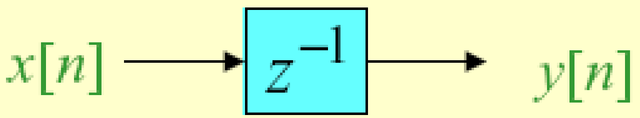
- The computational algorithm of an LTI digital filter can be conveniently represented in block diagram form using the basic building blocks shown below



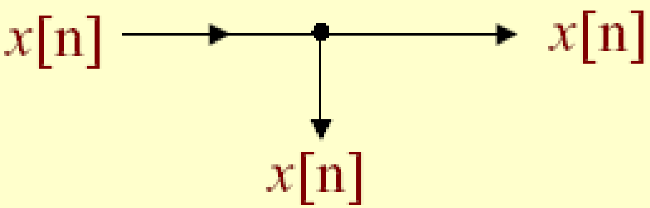
Adder



Multiplier



Unit delay



Pick-off node

Basic Building Blocks

Advantages of block diagrams:

1) Easy to write down the computational algorithm by inspection
2) Easy to analyze the block diagram to determine the explicit relation between the output and input

Basic Building Blocks

3) Easy to manipulate a block diagram to derive other “equivalent” block diagrams yielding different computational algorithms

4) Easy to determine the hardware requirements

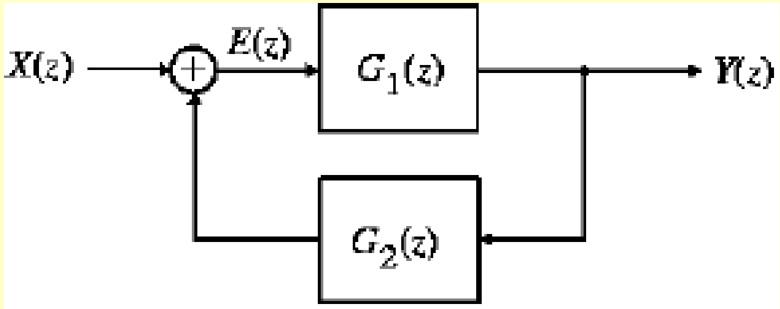
5) Easier to develop block diagram representations from the transfer function directly

Analysis of Block Diagrams

- Block diagrams can be analyzed 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

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

Analysis of Block Diagrams

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

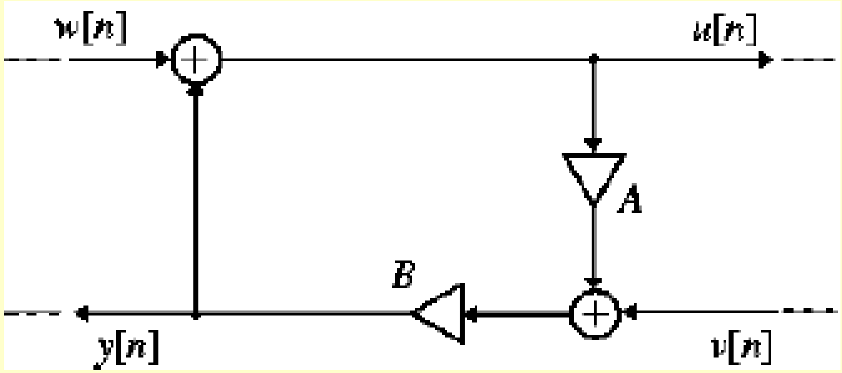
$$[1 - G_1(z)G_2(z)]Y(z) = G_1(z)X(z)$$

which leads to

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

The Delay Free-Loop Problems

- For physical realizability of the digital filter structure, it is necessary that the block diagram contains no delay-free loops
- To illustrate the delay-free loop problem consider the structure below



The Delay Free-Loop Problems

- Analysis of this structure yields

$$u[n] = w[n] + y[n]$$

$$y[n] = B(v[n] + Au[n])$$

which when combined results in

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

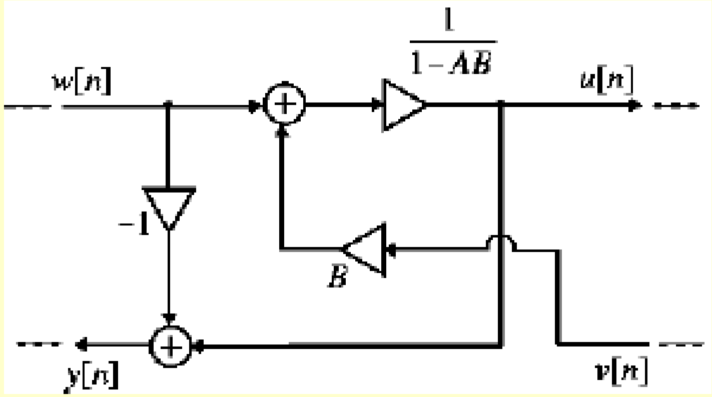
➡ The determination of the current value of $y[n]$ requires the knowledge of the same value

The Delay Free-Loop Problems

- However, this is physically impossible to achieve due to the finite time required to carry out all arithmetic operations on a digital machine
- Method exists to detect the presence of delay-free loops in an arbitrary digital filter structure, along with methods to locate and remove these loops without altering the overall input-output relation

The Delay Free-Loop Problems

- Removal achieved by replacing the portion of the overall structure containing the delay-free loops by an equivalent realization with no delay-free loops
- Figure below shows such a realization of the example structure described earlier



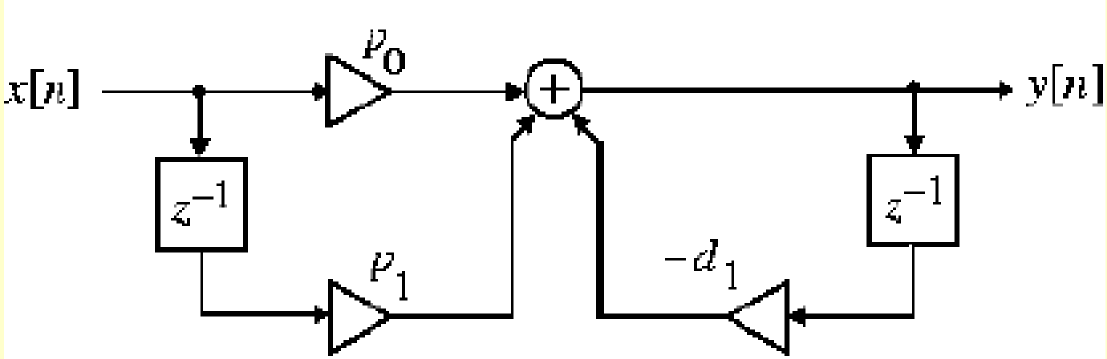
Canonic and Noncanonic Structures

- 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

Canonic and Noncanonic Structures

- The structure shown below is noncanonic as it employs two delays to realize a first-order difference equation

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



1.4 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

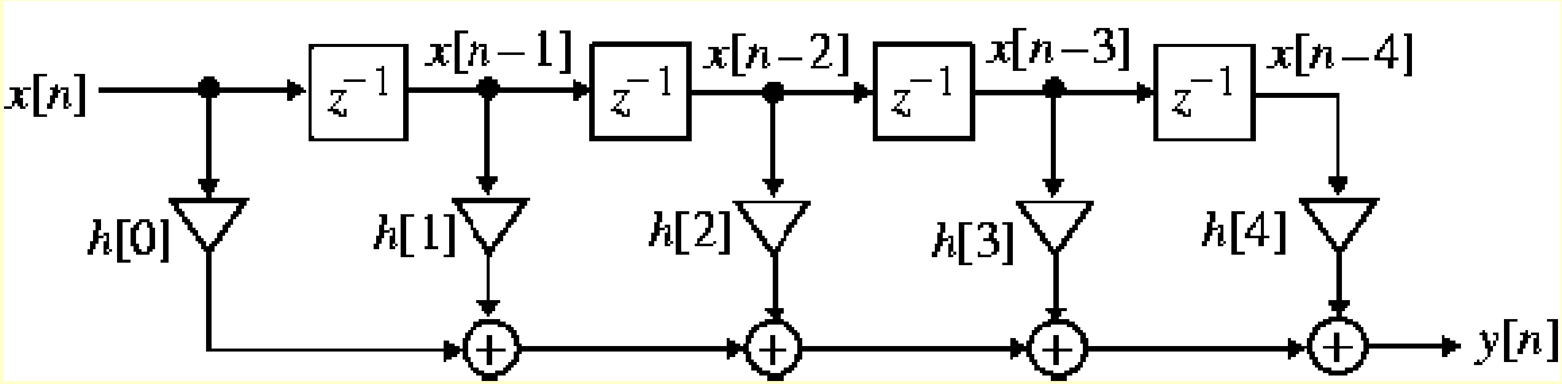
$$y[n] = \sum_{k=0}^{N} h[k] x[n-k]$$

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

Direct Form FIR Digital Filter Structures

- A direct form realization of an FIR filter can be readily developed from the convolution sum description for $N = 4$



Direct Form FIR Digital Filter Structures

- 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 **tapped delay line** or a **transversal filter**

Cascade Form FIR Digital Filter Structures

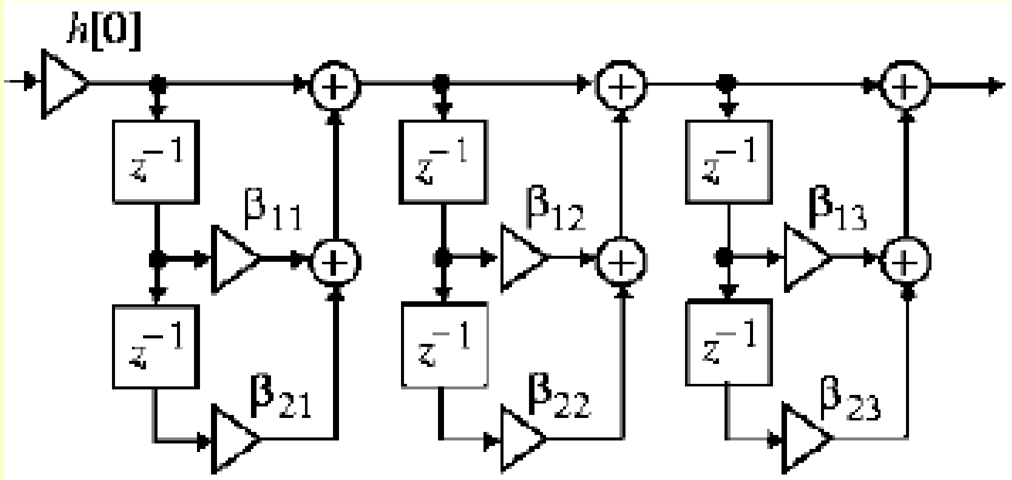
- A higher-order FIR transfer function can also be realized as a cascade of second-order FIR sections and possibly a first-order section
- To this end we express $H(z)$ as

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

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

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 transposed direct form

Linear-Phase FIR Structures

- The symmetry (or antisymmetry) 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:

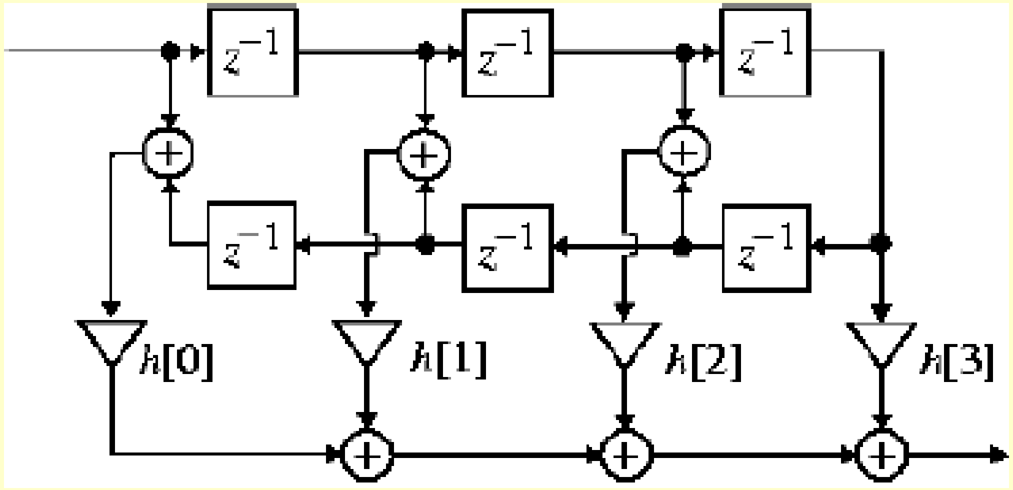
$$H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} \\ + h[2]z^{-4} + h[1]z^{-5} + h[0]z^{-6}$$

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



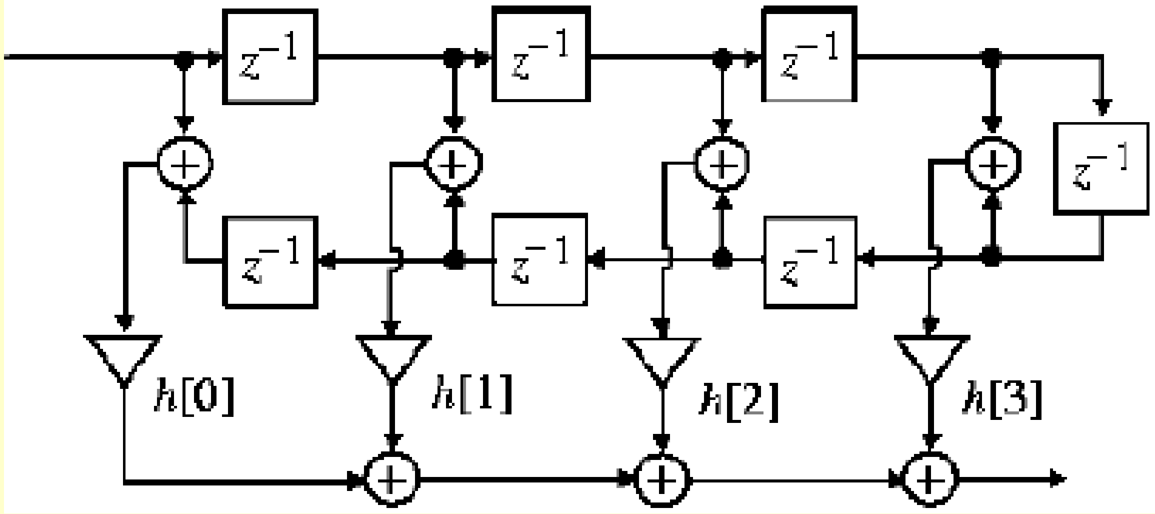
Linear-Phase FIR Structures

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

Linear-Phase FIR Structures

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

leading to the realization shown below



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 antisymmetric impulse responses

1.5 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, it can be seen that the realization of the causal IIR digital filters requires some form of feedback

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

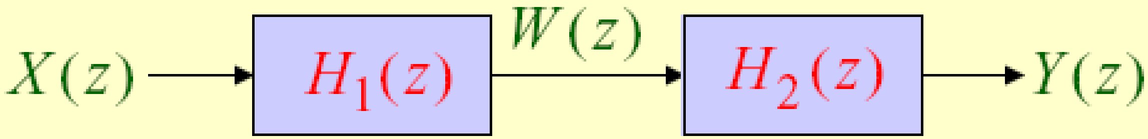
Direct Form IIR Digital Filter Structures

- 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

Direct Form IIR Digital Filter Structures



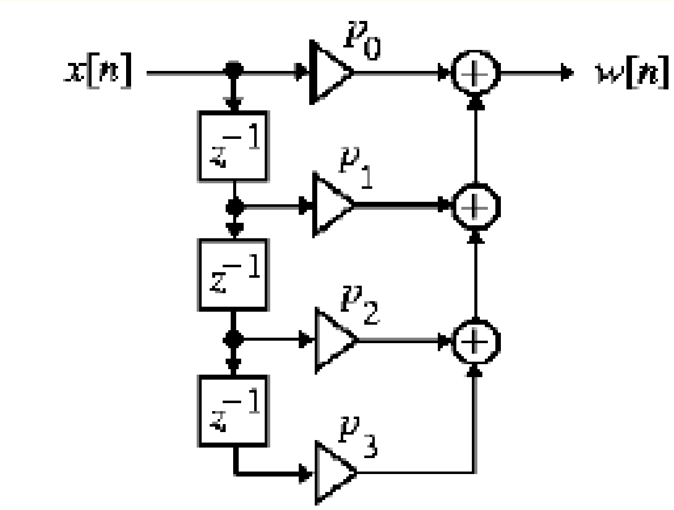
where

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

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

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 below



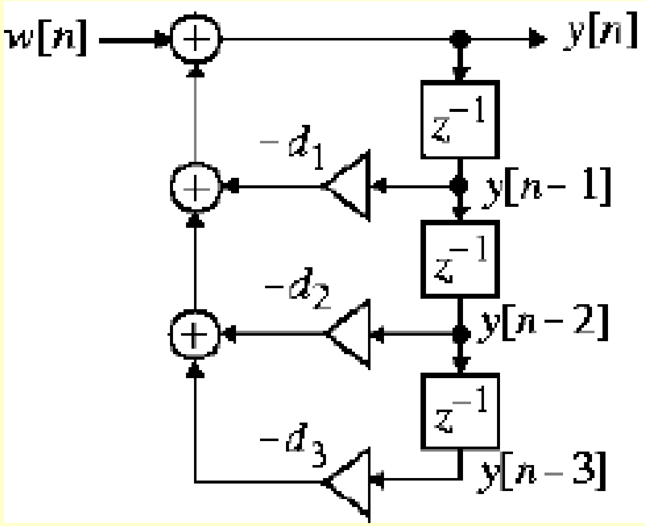
$$w[n] = p_0 x[n] + p_1 x[n-1] + p_2 x[n-2] + p_3 x[n-3]$$

Direct Form IIR Digital Filter Structures

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

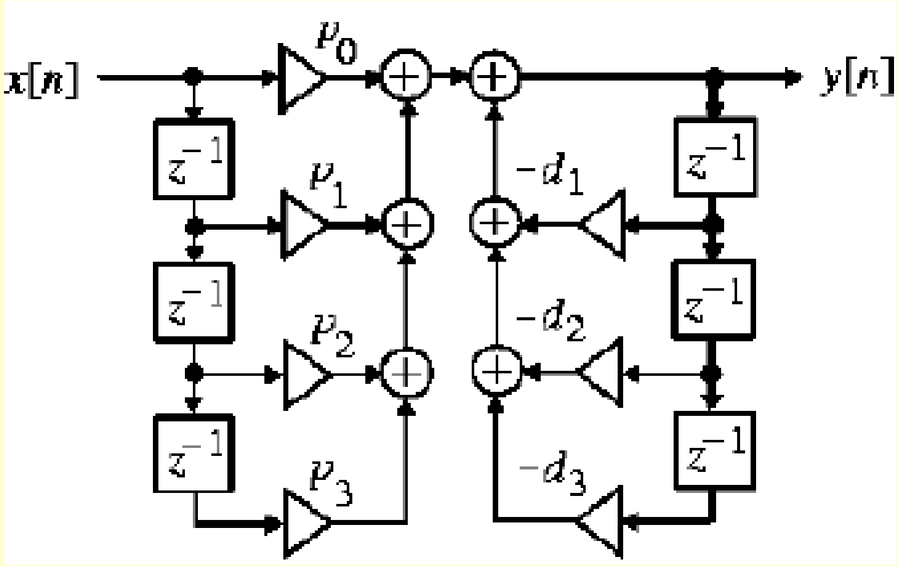
$$y[n] = w[n] - d_1 y[n-1] - d_2 y[n-2] - d_3 y[n-3]$$

- The realization of $H_2(z)$ follows from the above equation and is shown in the figure



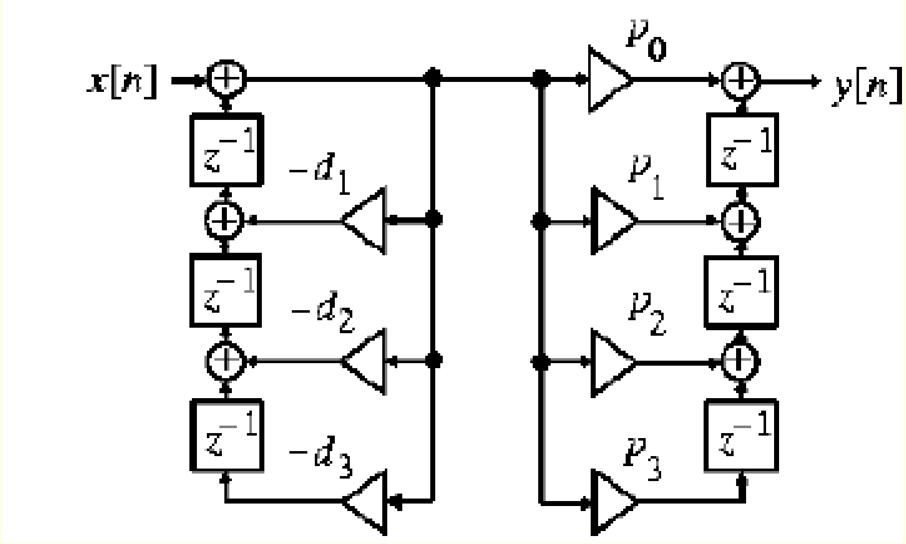
Direct Form IIR Digital Filter Structures

- 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 **direct form I** structure



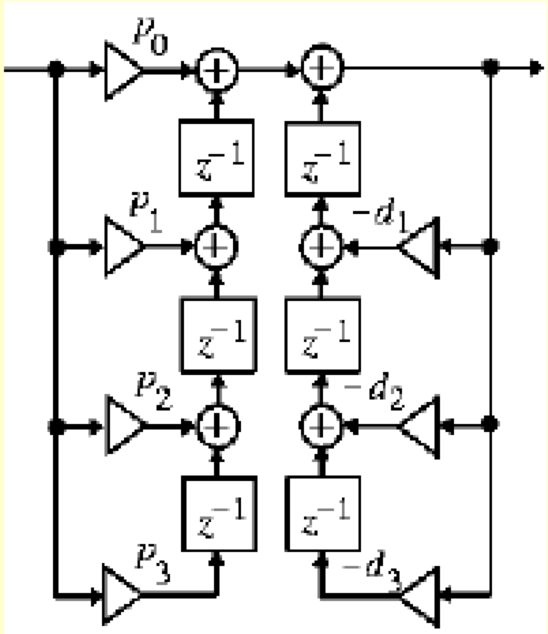
Direct Form IIR Digital Filter Structures

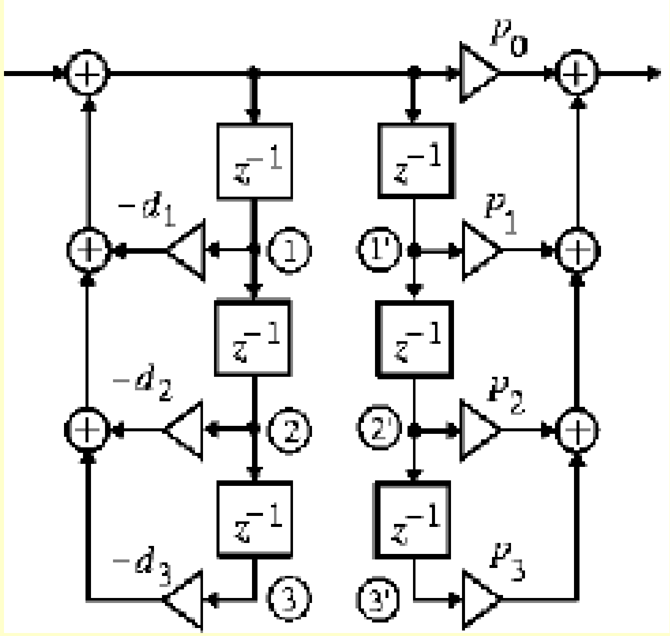
- The direct form I structure is noncanonic as it employs 6 delays to realize a 3rd-order transfer function
- The transpose of the direct form I structure is shown in the figure and it is called the **direct form I_t** structure



Direct Form IIR Digital Filter Structures

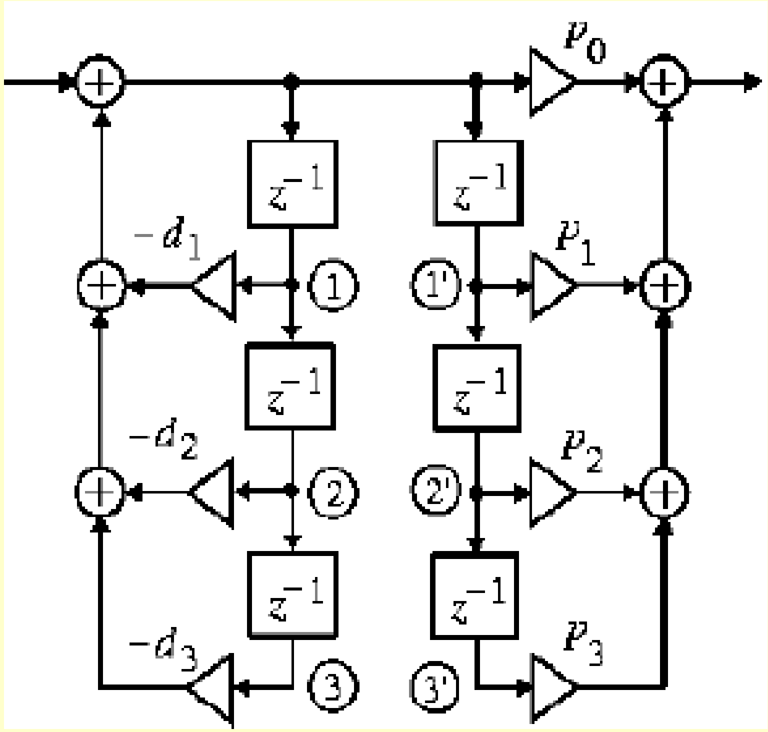
- Various other noncanonic direct form structures can be derived by simple block diagram manipulations as shown below





Direct Form IIR Digital Filter Structures

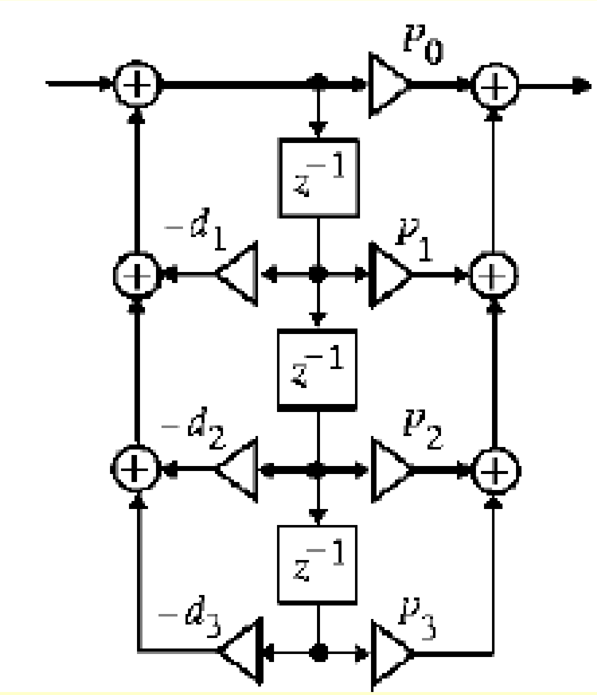
- Observe in the direct form structure shown right, the signal variable at nodes ① and ①' are the same, and hence the two top delays can be shared

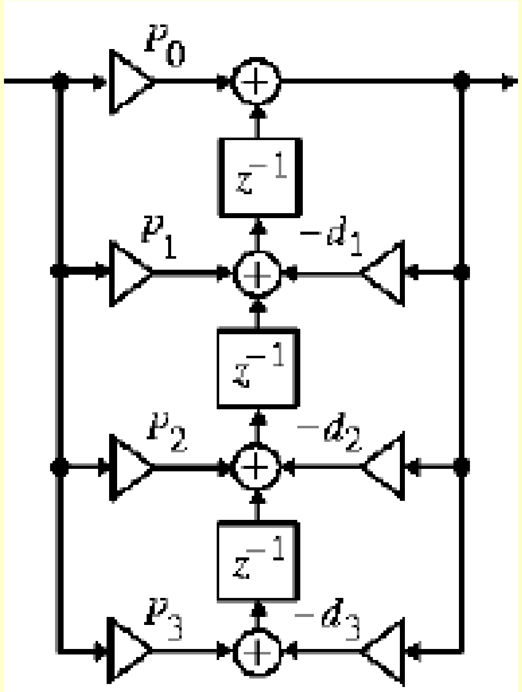


Direct Form IIR Digital Filter Structures

- Likewise, the signal variables at nodes ② and ②′ are the same, permitting the sharing of the middle two delays
- Following the same argument, we can share the bottom two delays leading to the final canonic structure, which is called the **direct form II** structure
- The direct form II and the **direct form II$_t$** structure are shown on the next slide

Direct Form IIR Digital Filter Structures





- Direct form realizations of an N-th order IIR transfer function should be evident

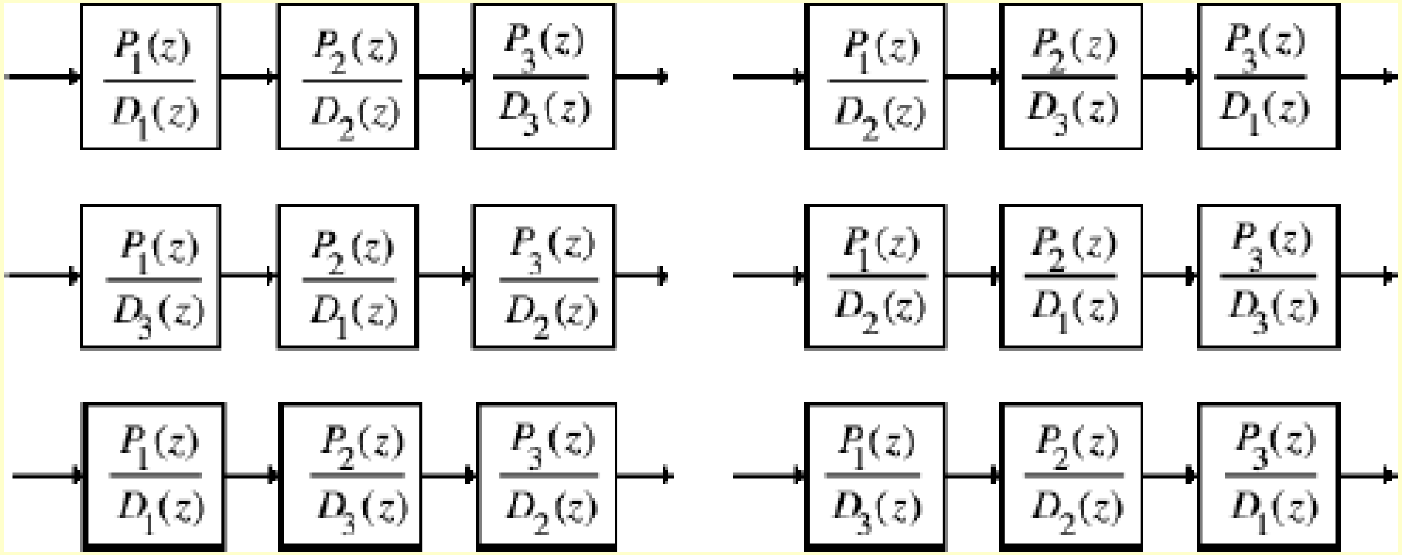
Cascade Form IIR Digital Filter Structures

- 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

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

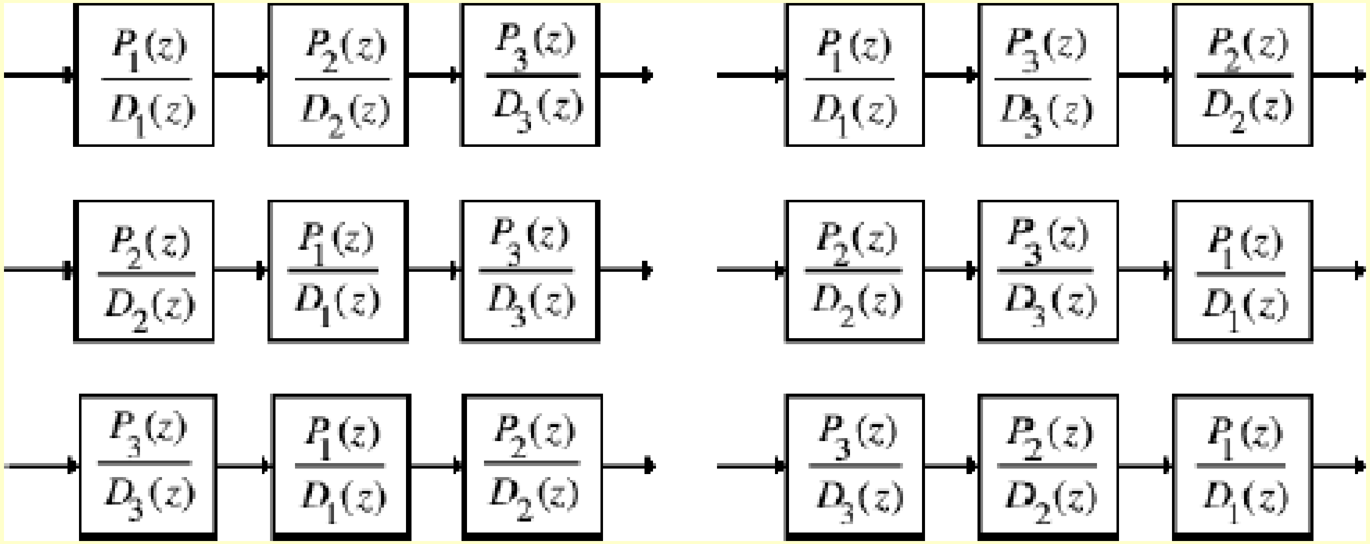
Cascade Form IIR Digital Filter Structures

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



Cascade Form IIR Digital Filter Structures

- Examples of cascade realizations obtained by different ordering of sections are shown below



Cascade Form IIR Digital Filter Structures

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

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

  based on different pole-zero-pairings and different orderings

- Due to finite wordlength effects, each such cascade realization behaves differently from others

Cascade Form IIR Digital Filter Structures

- Usually, the polynomials are factored into a product of 1st-order and 2nd-order polynomials
- In this case $H(z)$ is expressed as

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

- In the above, for a first-order factor

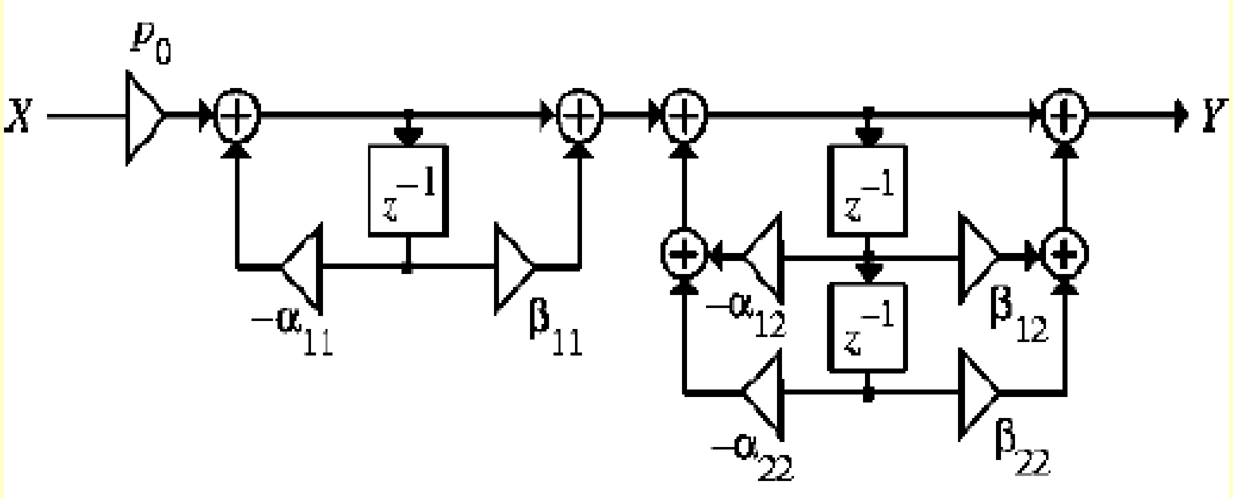
$$\alpha_{2k} = \beta_{2k} = 0$$

Cascade Form IIR Digital Filter Structures

- Consider the 3rd-order transfer function

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

- One possible realization is shown below



Cascade Form IIR Digital Filter Structures

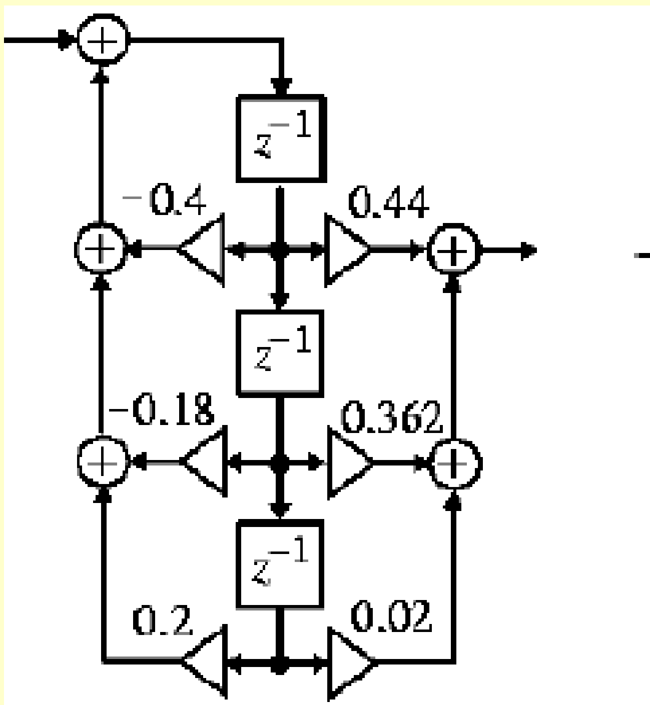
- Example: Direct form II and cascade form realizations of

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

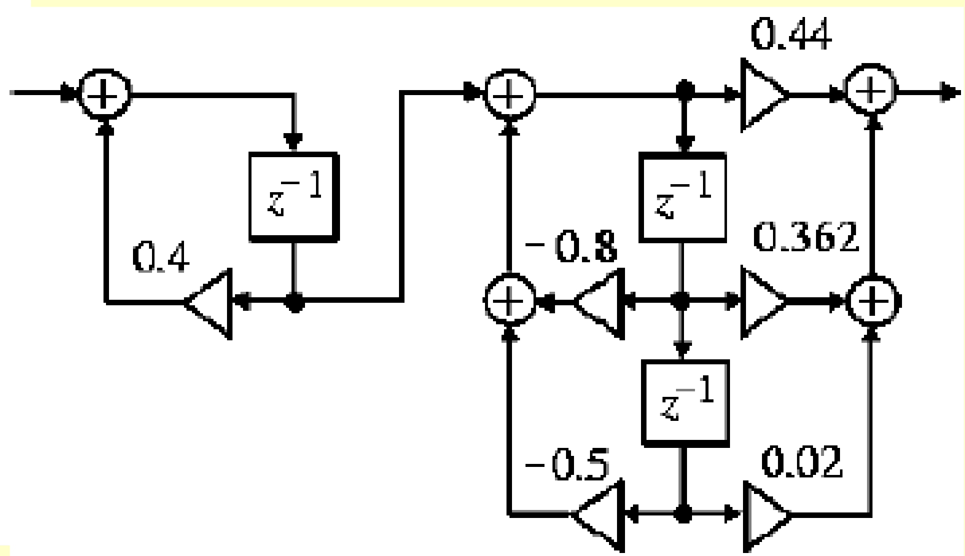
$$=\left(\frac{0.44+0.362z^{-1}+0.02z^{-2}}{1+0.8z^{-1}+0.5z^{-2}}\right)\left(\frac{z^{-1}}{1-0.4z^{-1}}\right)$$

are shown on the next slide

Cascade Form IIR Digital Filter Structures



Direct form II



Cascade form

Parallel Form IIR Digital Filter Structures

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

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

- In the above, for a real pole $\alpha_{2k} = \gamma_{1k} = 0$

Parallel Form IIR Digital Filter Structures

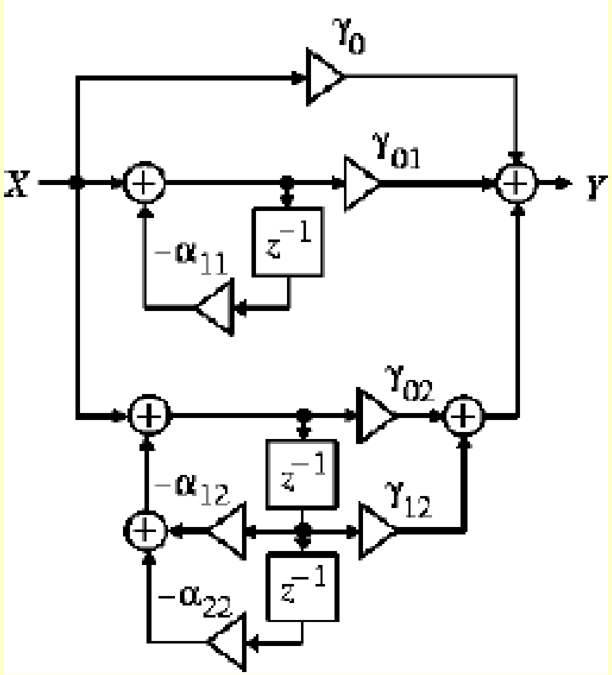
- A direct partial-fraction expansion of the transfer function in z leads to the **parallel form II** structure
- Assuming simple poles, in this case we arrive at

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

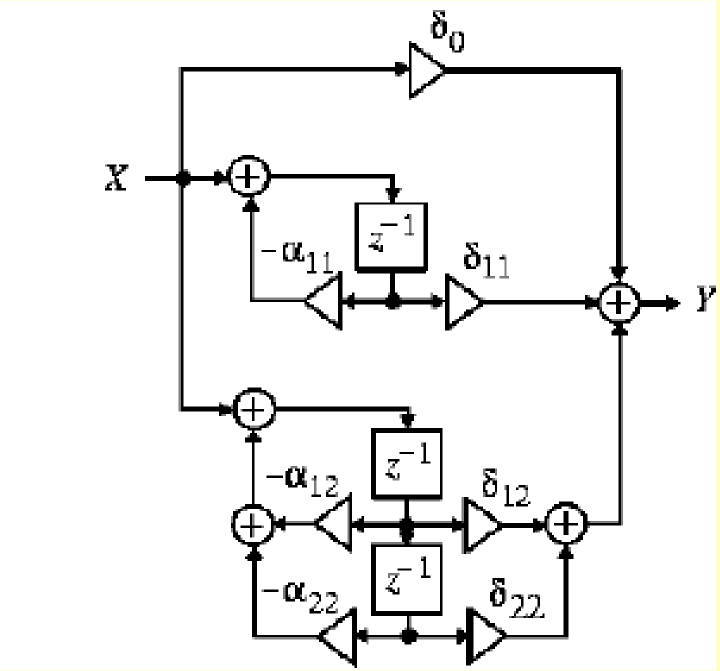
- Here, for a real pole $\alpha_{2k} = \delta_{2k} = 0$

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

Parallel Form IIR Digital Filter Structures

- Example: A partial-fraction expansion of

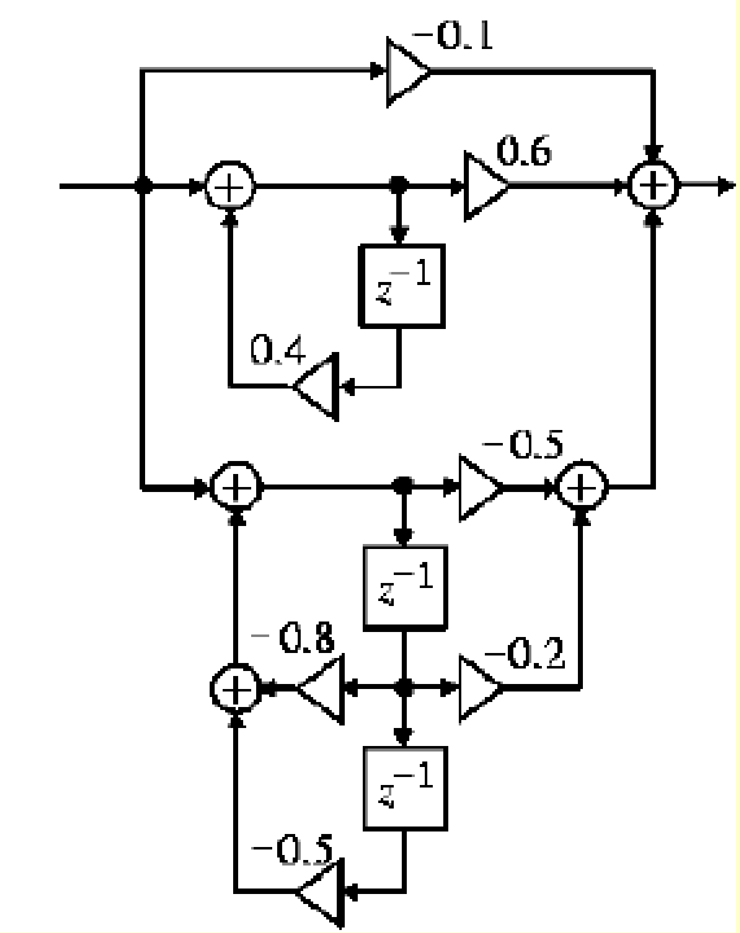
$$H(z) = \frac{0.44z^{-1} + 0.362z^{-2} + 0.02z^{-3}}{1 + 0.4z^{-1} + 0.18z^{-2} - 0.2z^{-3}}$$

in z^{-1} yields

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

Parallel Form IIR Digital Filter Structures

- The corresponding parallel form I realization is shown in the figure

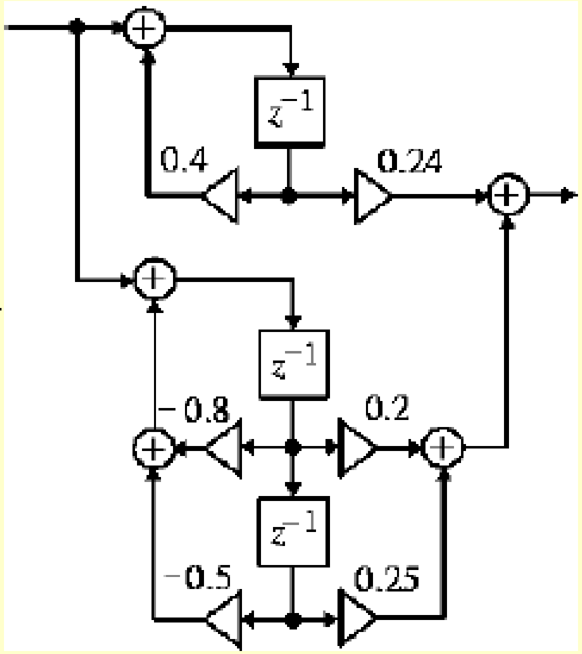


Parallel Form IIR Digital Filter Structures

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

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

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



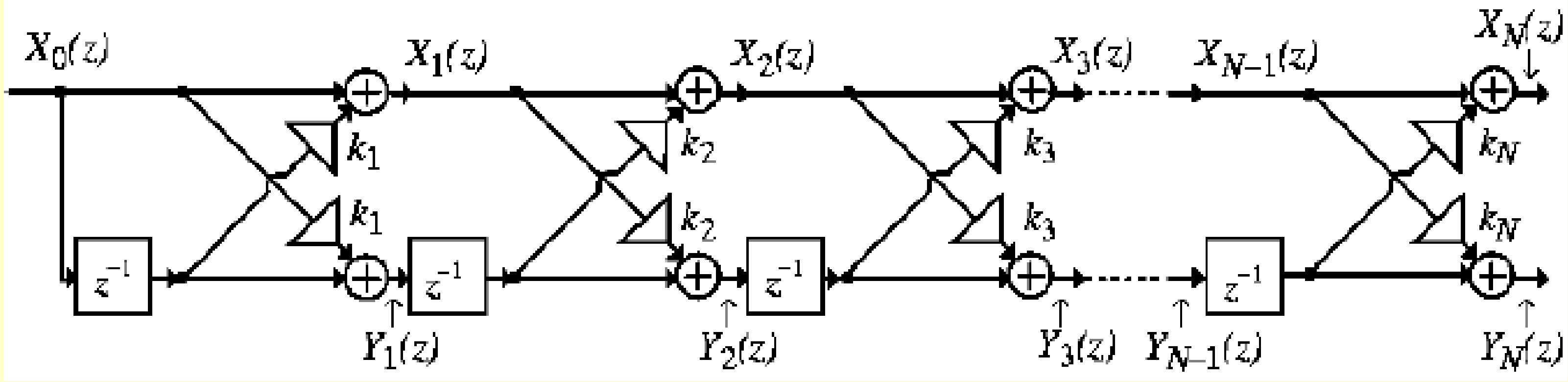
- The corresponding parallel form II realization is shown in the figure

1.6 FIR Cascaded Lattice Structures

- An arbitrary Nth-order FIR transfer function of the form

$$H_N(z) = 1 + \sum_{n=1}^{N} p_n z^{-n}$$

can be realized as a cascaded lattice structure as shown below



FIR Cascaded Lattice Structures

- From figure, it follows that

$$X_m(z) = X_{m-1}(z) + k_m z^{-1} Y_{m-1}(z)$$

$$Y_m(z) = k_m X_{m-1}(z) + z^{-1} Y_{m-1}(z)$$

- In matrix form the above equations can be written as

$$\begin{bmatrix} X_m(z) \\ Y_m(z) \end{bmatrix} = \begin{bmatrix} 1 & k_m z^{-1} \\ k_m & z^{-1} \end{bmatrix} \begin{bmatrix} X_{m-1}(z) \\ Y_{m-1}(z) \end{bmatrix}$$

where $m = 1, 2, \ldots, N$

FIR Cascaded Lattice Structures

- Denote

$$H_m(z) = \frac{X_m(z)}{X_0(z)}, \quad G_m(z) = \frac{Y_m(z)}{X_0(z)}$$

- Then it follows from the input-output relations of the *m*-th two-pair that

$$H_m(z) = H_{m-1}(z) + k_m z^{-1} G_{m-1}(z)$$

$$G_m(z) = k_m H_{m-1}(z) + z^{-1} G_{m-1}(z)$$

FIR Cascaded Lattice Structures

- From the previous equation we observe

$$H_1(z) = 1 + k_1 z^{-1}, \quad G_1(z) = k_1 + z^{-1}$$

where we have used the facts

$$H_0(z) = X_0(z) / X_0(z) = 1$$

$$G_0(z) = Y_0(z) / X_0(z) = X_0(z) / X_0(z) = 1$$

- It follows from the above that

$$G_1(z) = z^{-1}(z\,k_1 + 1) = z^{-1} H_1(z^{-1})$$

⇒ $G_1(z)$ is the mirror-image of $H_1(z)$

FIR Cascaded Lattice Structures

- From the input-output relations of the m-th two-pair we obtain for $m = 2$

$$H_2(z) = H_1(z) + k_2 z^{-1} G_1(z)$$

$$G_2(z) = k_2 H_1(z) + z^{-1} G_1(z)$$

- Since $H_1(z)$ and $G_1(z)$ are 1st-order polynomials, it follows that $H_2(z)$ and $G_2(z)$ are 2nd-order polynomials

FIR Cascaded Lattice Structures

- Substituting $G_1(z) = z^{-1}H_1(z^{-1})$ in the two previous equations we get

$$H_2(z) = H_1(z) + k_2 z^{-2} H_1(z^{-1})$$

$$G_2(z) = k_2 H_1(z) + z^{-2} H_1(z^{-1})$$

- Now we can write

$$G_2(z) = k_2 H_1(z) + z^{-2} H_1(z^{-1})$$

$$= z^{-2}[k_2 z^2 H_1(z) + H_1(z^{-1})] = z^{-2} H_2(z^{-1})$$

⇒ $G_2(z)$ is the mirror-image of $H_2(z)$

FIR Cascaded Lattice Structures

- In the general case, from the input-output relations of the *m*-th two-pair we obtain

$$H_m(z) = H_{m-1}(z) + k_m z^{-1} G_{m-1}(z)$$

$$G_m(z) = k_m H_{m-1}(z) + z^{-1} G_{m-1}(z)$$

- It can be easily shown that

$$G_m(z) = z^{-m} H_m(z^{-1}), \; m = 1, 2, \ldots, N$$

⇨ $G_m(z)$ is the mirror-image of $H_m(z)$

FIR Cascaded Lattice Structures

- To develop the synthesis algorithm, we express $H_{m-1}(z)$ and $G_{m-1}(z)$ in terms of $H_m(z)$ and $G_m(z)$ for $m = N, N-1, \ldots, 1$ arriving at

$$H_{N-1}(z) = \frac{1}{(1-k_N^2)}\{H_N(z) - k_N G_N(z)\}$$

$$G_{N-1}(z) = \frac{1}{(1-k_N^2)z^{-1}}\{-k_N H_N(z) + G_N(z)\}$$

FIR Cascaded Lattice Structures

- Substituting the expressions for

$$H_N(z) = 1 + \sum_{n=1}^{N} p_n z^{-n}$$

and

$$G_N(z) = z^{-N} H_N(z^{-1}) = \sum_{n=1}^{N-1} p_n z^{-N+n} + z^{-N}$$

in the first equation we get

$$H_{N-1}(z) = \frac{1}{1-k_N^2}\{(1 - k_N p_N) + \sum_{n=1}^{N-1}(p_n - k_N p_{N-n})z^{-n} + (p_N - k_N)z^{-N}\}$$

FIR Cascaded Lattice Structures

- If we choose $k_N = p_N$, then $H_{N-1}(z)$ reduces to an FIR transfer function of order $N-1$ and can be written in the form

$$H_{N-1}(z) = 1 + \sum_{n=1}^{N-1} p'_n z^{-n}$$

where

$$p'_n = \frac{p_n - k_N p_{N-n}}{1 - k_N^2}, \; 1 \le n \le N-1$$

- Continuing the above recursion algorithm, all multiplier coefficients of the cascaded lattice structure can be computed

FIR Cascaded Lattice Structures

- Example: Realize the FIR transfer function

$$H_4(z) = 1 + 1.2z^{-1} + 1.12z^{-2} + 0.12z^{-3} - 0.08z^{-4}$$

From the above, we observe $k_4 = p_4 = -0.08$

and using $p'_n = \frac{p_n - k_4 p_{4-n}}{1 - k_4^2}, 1 \leq n \leq 3$

we determine the coefficients of $H_3(z)$ as

$$p'_3 = 0.2173913,\ p'_2 = 1.2173913$$

$$p'_1 = 1.2173913$$

FIR Cascaded Lattice Structures

- As a result,

$$H_3(z) = 1 + 1.2173913\,z^{-1} + 1.2173913\,z^{-2} + 0.2173913\,z^{-3}$$

- Thus, $k_3 = p'_3 = 0.2173913$
- Using

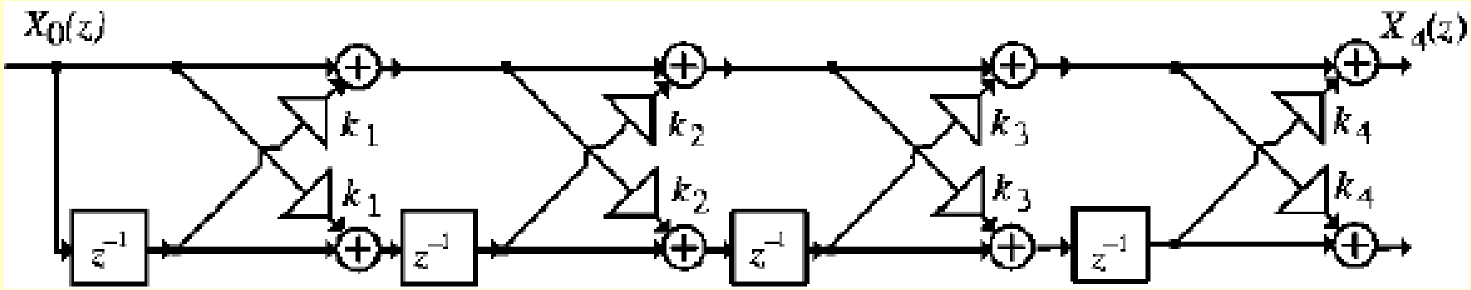
$$p''_n = \frac{p'_n - k_3 p'_{2-n}}{1 - k_3^2},\ 1 \le n \le 2$$

we determine the coefficients of $H_2(z)$ as

$$p''_2 = 1.0, \qquad p''_1 = 1.0$$

FIR Cascaded Lattice Structures

- As a result, $H_2(z) = 1 + z^{-1} + z^{-2}$
- From the above, we get $k_2 = p''_2 = 1$
- The last recursion yields the last multiplier coefficient $k_1 = p''_1/(1+k_2) = 0.5$
- The complete realization is shown below



$$k_1 = 0.5,\ k_2 = 1,\ k_3 = 0.2173913,\ k_4 = -0.08$$