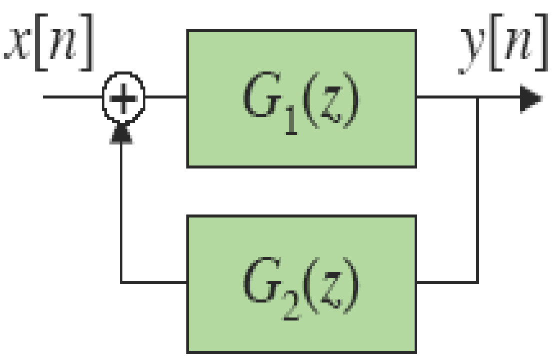
Block Diagrams

- Useful way to illustrate implementations
- Z-transform helps analysis:

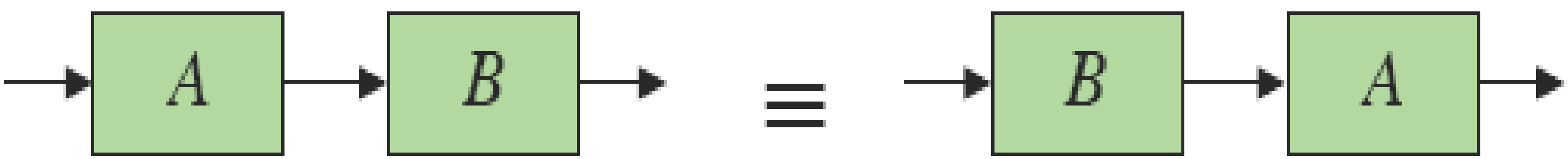


$$Y(z) = G_1(z)\left[X(z) + G_2(z)Y(z)\right]$$
$$\Rightarrow Y(z)\left[1 - G_1(z)G_2(z)\right] = G_1(z)X(z)$$
$$\Rightarrow H(z) = \frac{Y(z)}{X(z)} = \frac{G_1(z)}{1 - G_1(z)G_2(z)}$$

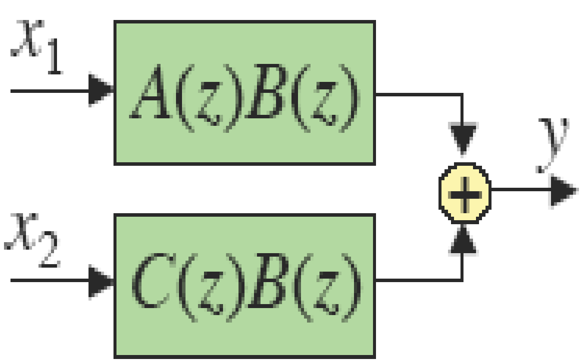
- Approach
  - Output of summers as dummy variables
  - Everything else is just multiplicative

Equivalent Structures

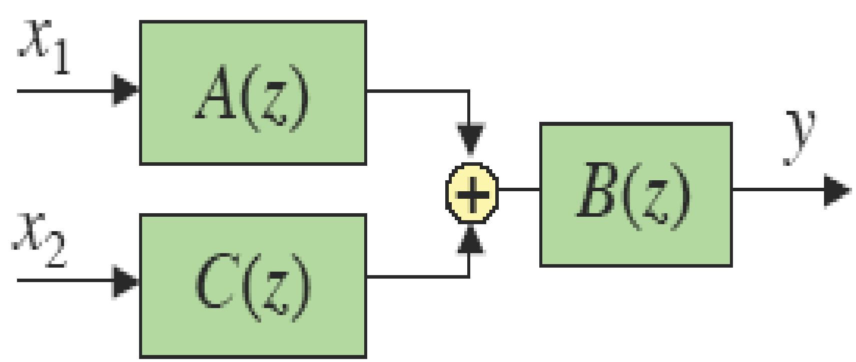
- Modifications to block diagrams that do not change the filter
- e.g. Commutation $H = AB = BA$



- Factoring $AB + CB = (A+C) \cdot B$



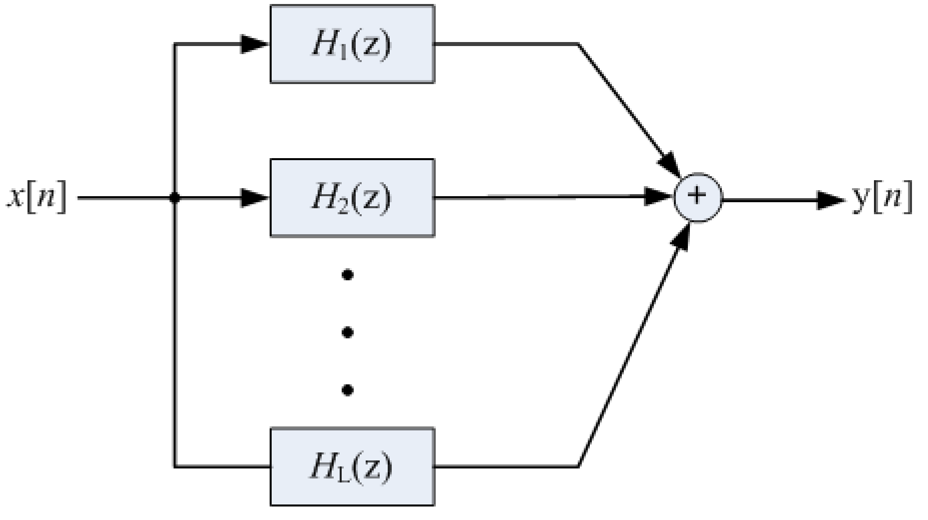
fewer blocks



less computation

- The Transfer Function of LTI system can be connected in 2 ways :

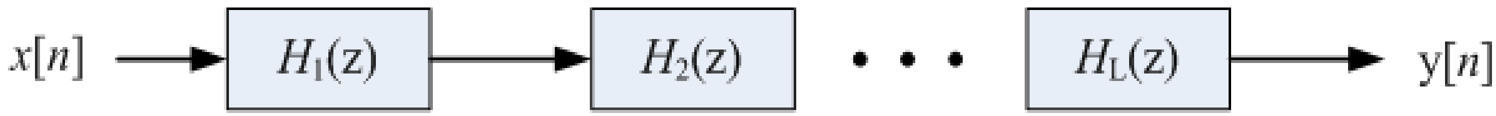
a. Parallel Connection :



The overall transfer function,

$$H(z) = H_1(z) + H_2(z) + \ldots + H_L(z)$$

- b. Cascade connection :



The overall transfer function :

$$H(z) = H_1(z).H_2(z). \ldots H_L(z)$$

Each one of them can be implemented using any of the Direct Forms

- Canonic

  - number of delays in the block diagram representation is equal to the order of the difference equation

- Non-canonic
  - otherwise

FIR FILTER STRUCTURES

FIR FILTERS

- These are realized using only two Forms:
- (as it only has the Numerator part i.e. ALL ZERO SYSTEMS)

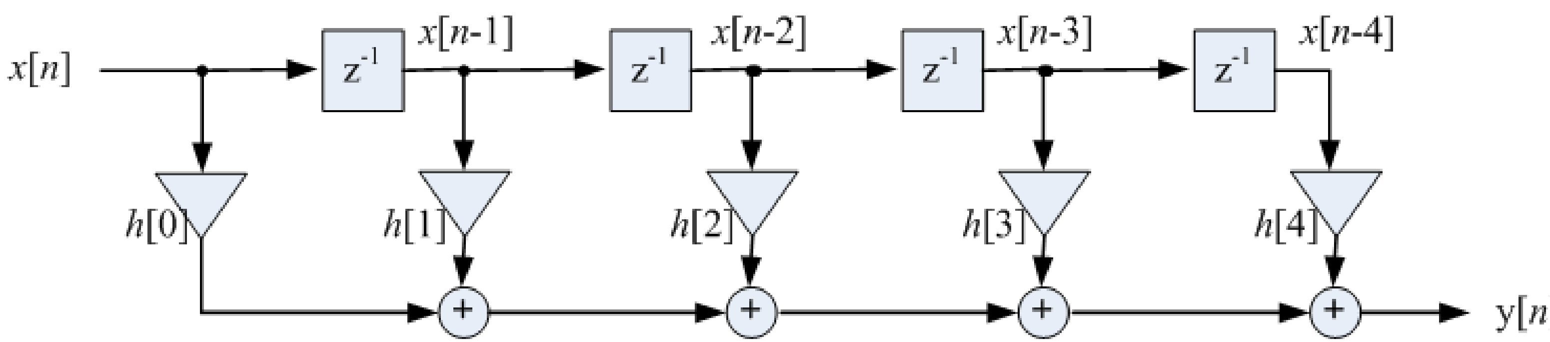
- 1. Direct Form 1 or Tapped delay Line or Transversal delay Line Filter.

- 2. Cascade form

FIR Filter Structures

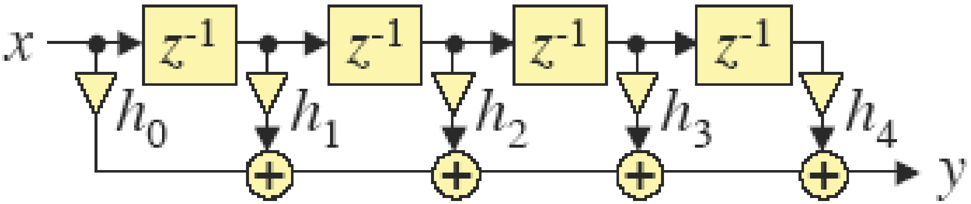
- Direct form
  - An FIR filter of order N requires N + 1 multipliers, N adders and N delays.
  - An FIR filter of order 4

$y[n] = h[0]x[n] + h[1]x[n\text{-}1] + h[2]x[n\text{-}2] + h[3]x[n\text{-}3]+ \quad h[4]x[n\text{-}4]$



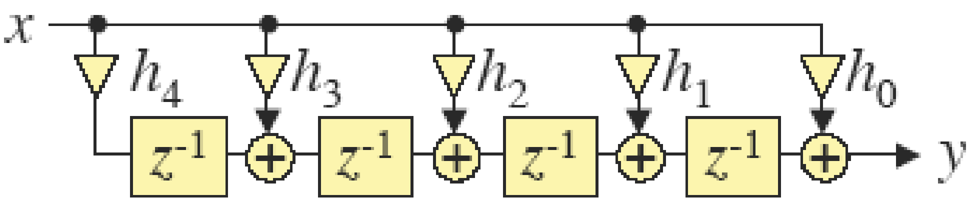
FIR Filter Structures

- Direct form "Tapped Delay Line"



$$y[n] = h_0 x[n] + h_1 x[n-1] + \dots$$
$$= \sum_{k=0}^{4} h_k x[n-k]$$

- Transpose



- Re-use delay line if several inputs x_i for single output y ?

- Cascade form
  - Transfer function H(z) of a causal FIR filter of order N

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

  - Factorized form

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

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

Example…

- Determine the Direct Form & Cascade Form Realization for the transfer Function of an FIR Digital filter which is given by

$$H(z)= (1-1/4\ Z +3/8\ \ Z^2)(1-1/8Z-1/2Z^2)$$

Direct Form

- We Simply Expand the equation to get this form as

Let us expand the transfer function H(z) i.e., equation (i) as under :

$$H(z) = 1 - \frac{3}{8}z^{-1} - \frac{3}{32}z^{-2} + \frac{5}{64}z^{-3} - \frac{3}{16}z^{-4}$$

This function can be realised in FIR dirct form as depicted in figure 8.72.

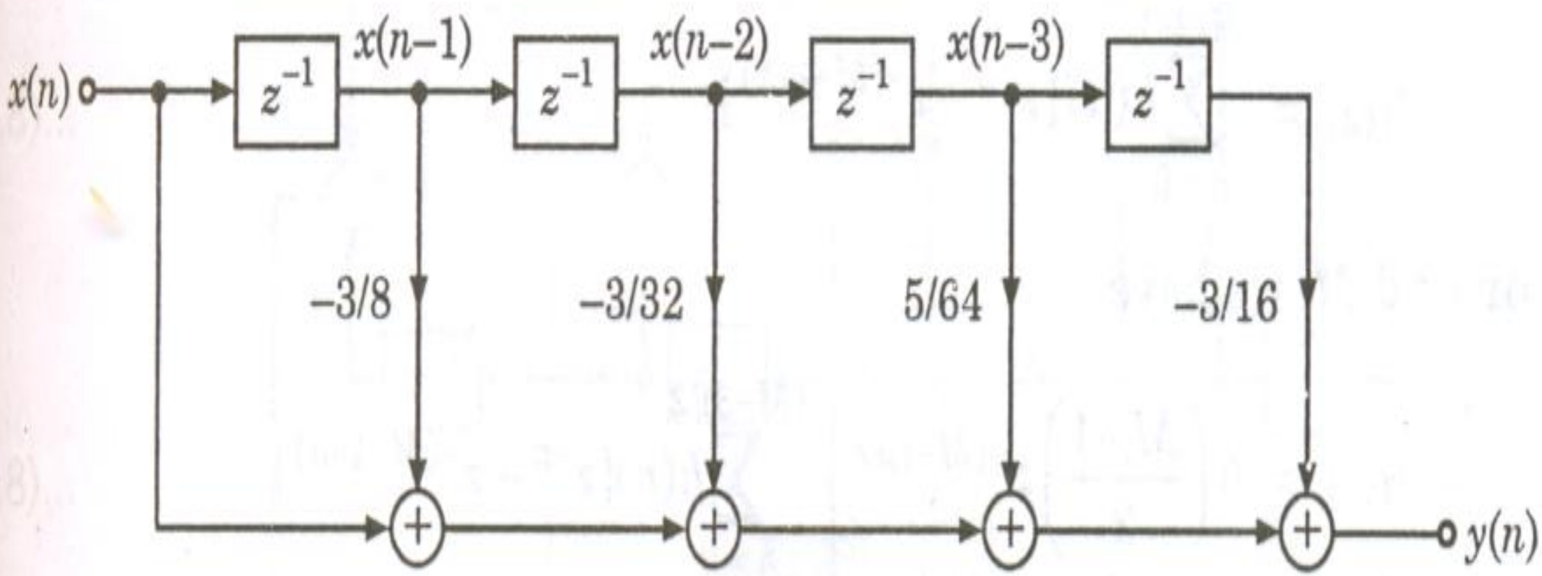
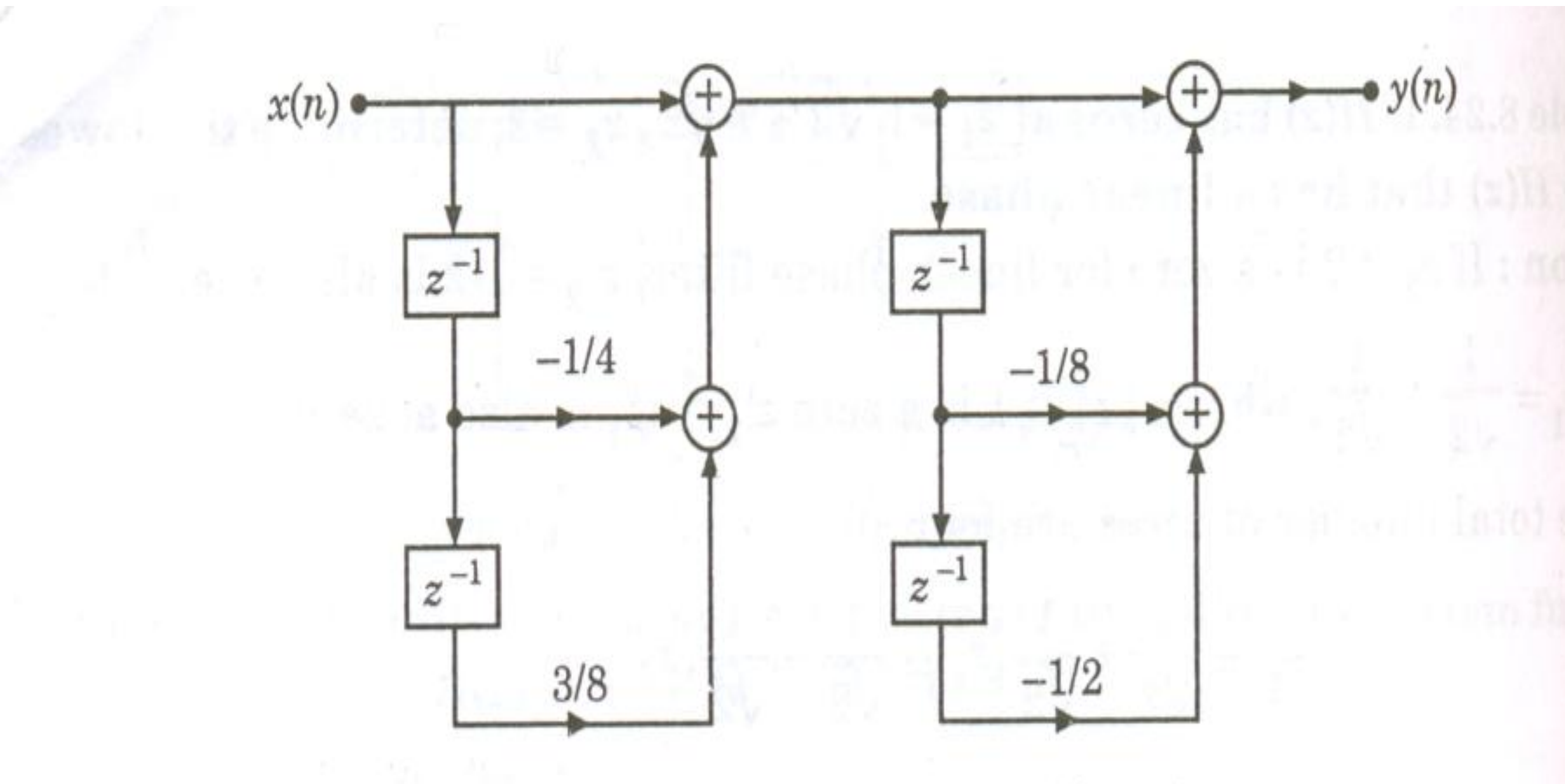


Fig. 8.72 Direct form realisation

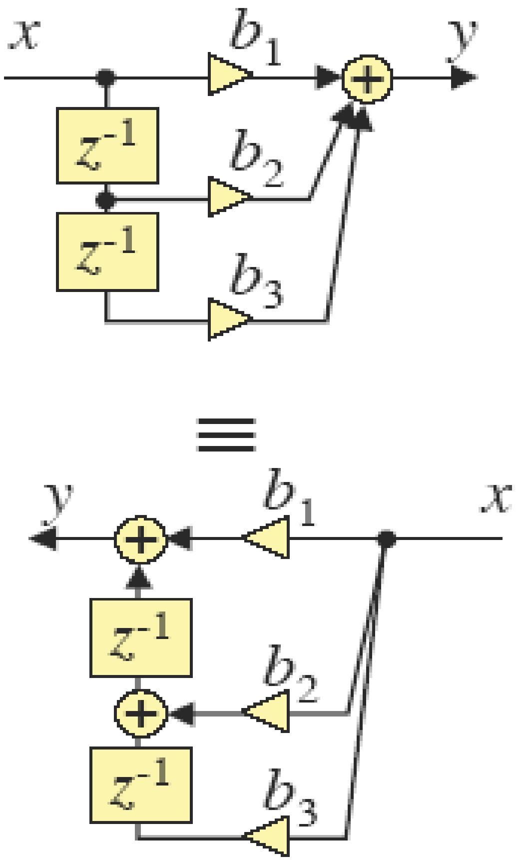
Cascade Form

- H(z)=H1(z)* H2(z) & hence



Equivalent Structures

- Transpose
  - reverse paths
  - adders↔nodes
  - input↔output



$$
\begin{aligned}
Y &= b_1 X + b_2 z^{-1} X + b_3 z^{-2} X \\
&= b_1 X + z^{-1}\left(b_2 X + z^{-1} b_3 X\right)
\end{aligned}
$$

IIR FILTER STRUCTURES

- IIR system/filter can be realized in several structures:

1. DIRECT FORM I
2. DIRECT FORM II (CANONIC)
3. CASCADE FORM
4. PARALLEL FORM

IIR System Function

$$H(z) = \frac{Y(z)}{S(z)} = \frac{\sum_{k=0}^{M-1} B_k z^{-k}}{1 + \sum_{k=1}^{N-1} A_k z^{-k}}$$

where M and N are integer numbers.

Bifurcation of H(z) into H1(z) & H2(z)

$$H(z) = \frac{\sum_{k=0}^{M-1} B_k z^{-k}}{1 + \sum_{k=1}^{N-1} A_k z^{-k}} = H_1(z) \,.\, H_2(z) \qquad \ldots(7.3)$$

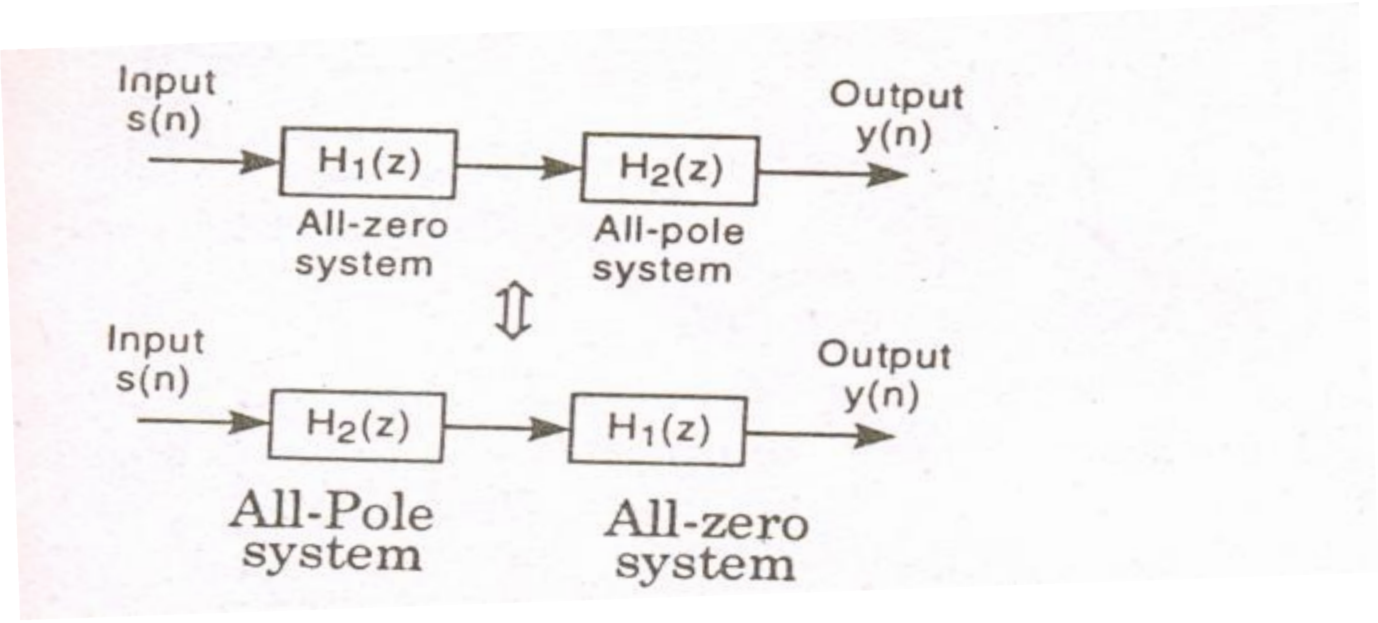
$$H_1(z) = \sum_{k=0}^{M-1} B_k z^{-k} \qquad \ldots(7.4)$$

$$H_2(z) = \frac{1}{1 + \sum_{k=1}^{N-1} A_k z^{-k}}$$

$$= \left[1 + \sum_{k=1}^{N} A_k z^{-k}\right]^{-1}$$

$$= 1 - \sum_{k=1}^{N} A_k z^{-k} = 1 + \sum_{k=1}^{N} (-A_k) z^{-k} \qquad \ldots(7.5)$$

Block Diagram of Direct Form I & II

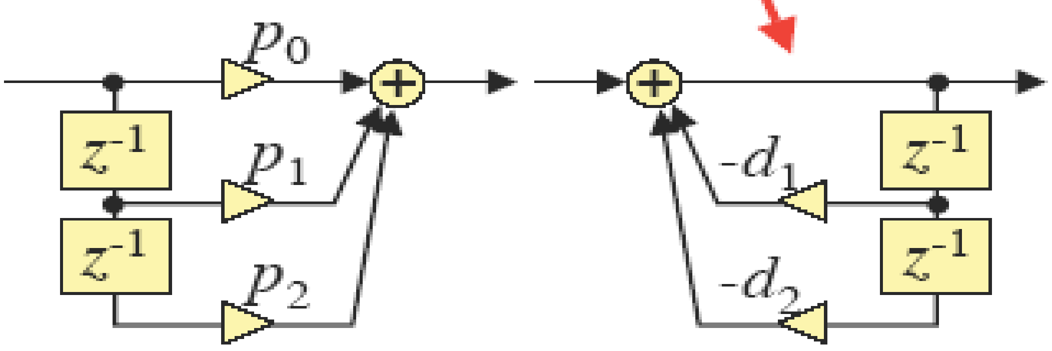


IIR Filter Structures

- IIR: numerator + denominator

$$H(z) = \frac{p_0 + p_1 z^{-1} + p_2 z^{-2} + \dots}{1 + d_1 z^{-1} + d_2 z^{-2} + \dots}$$

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

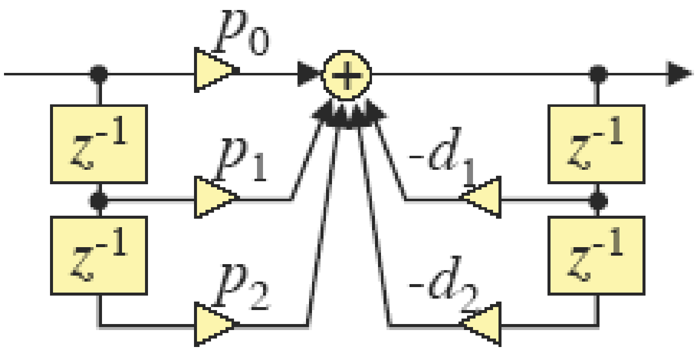


FIR

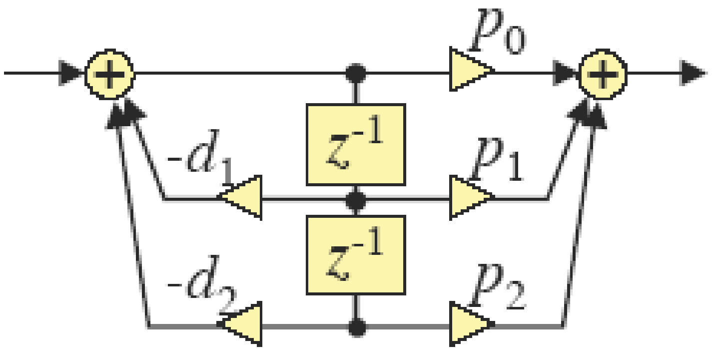
all-pole IIR

IIR Filter Structures

- Hence, Direct form I



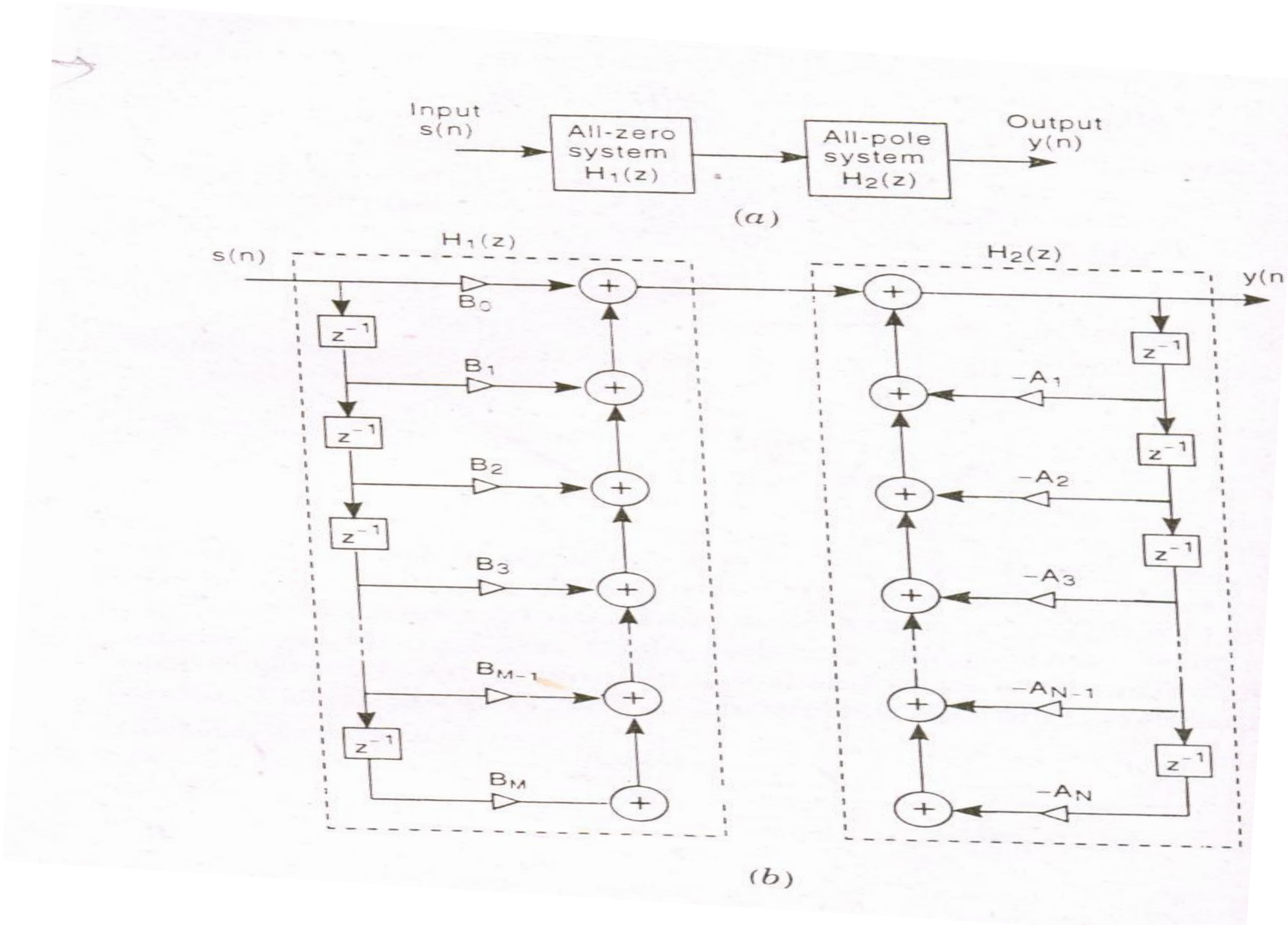
- Commutation $\rightarrow$ Direct form II (DF2)



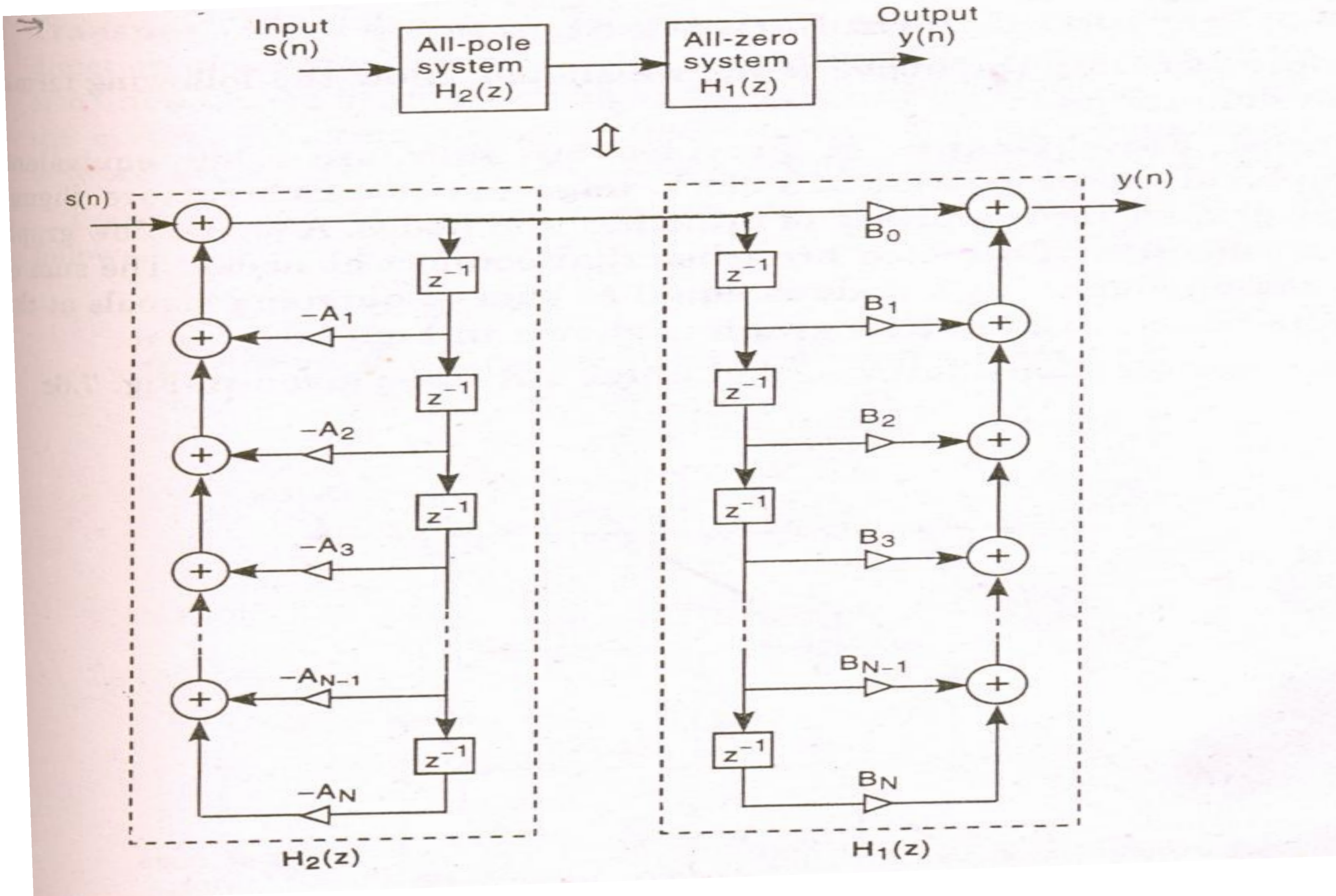
• same signal
∴ delay lines merge

• "canonical"
= min. memory usage

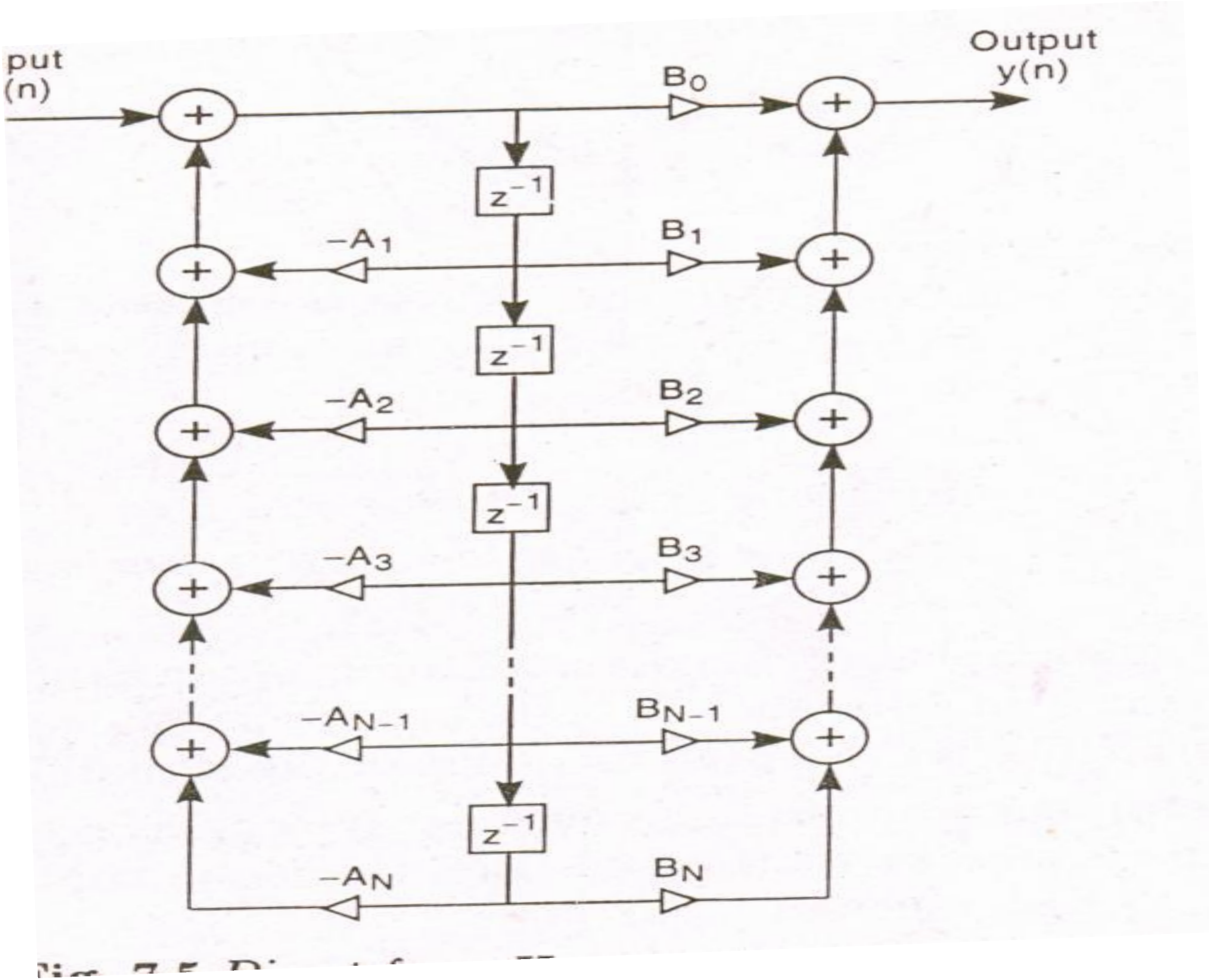
Direct Form I Realization



Direct Form II Realization



Canonic Direct Form II Realization



Parallel Form Realization

$$H_k(z) = \frac{B_{k0} + B_{k1}z^{-1}}{1 + A_{k1}z^{-1} + A_{k2}z^{-2}} \quad \text{...(7.9)}$$

Coefficients B_{ki} ard A_{ki} real-valued system parameters.

Parallel form network structures are shown in Fig. 7.9 and Fig. 7.10.

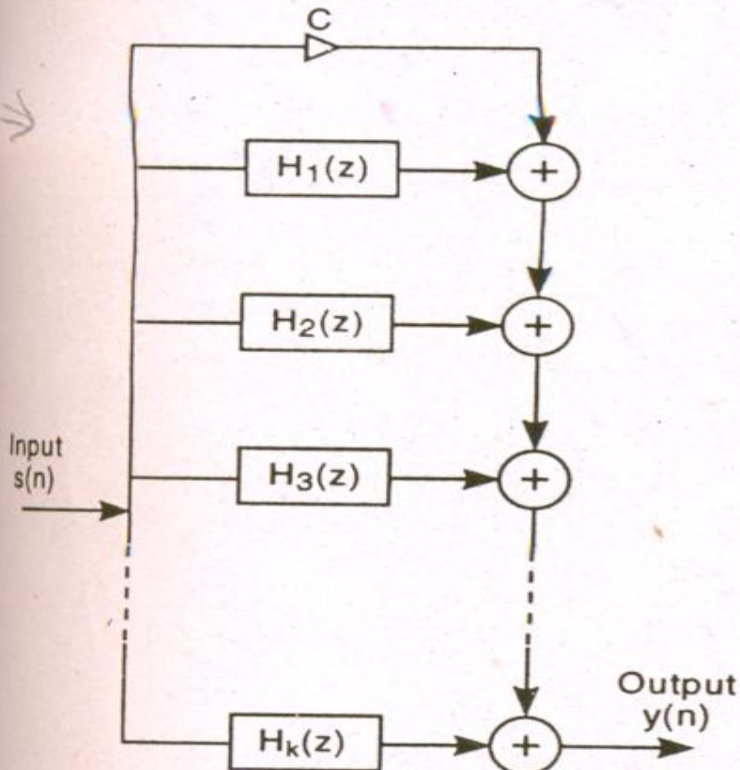


Fig. 7.9 *Paralled-form network structure of IIR system.*

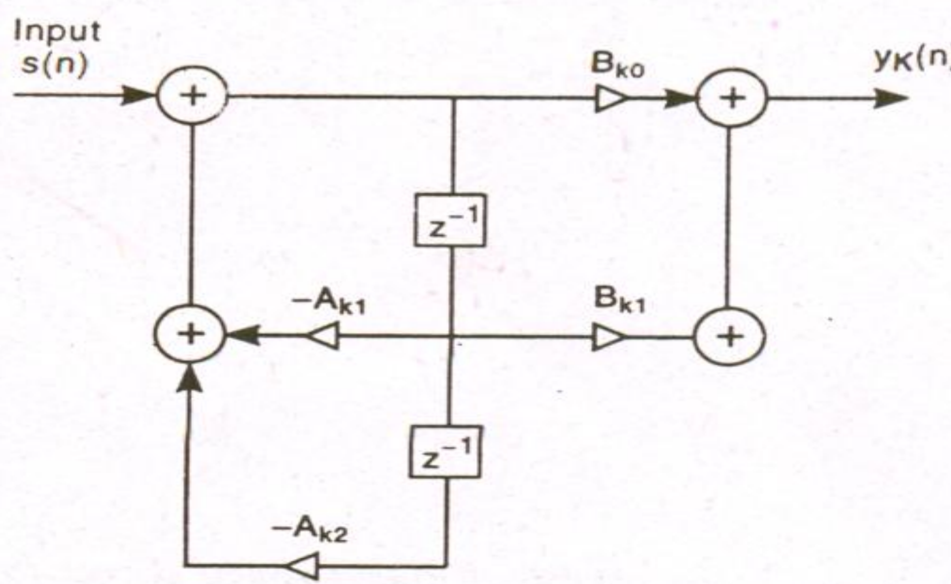


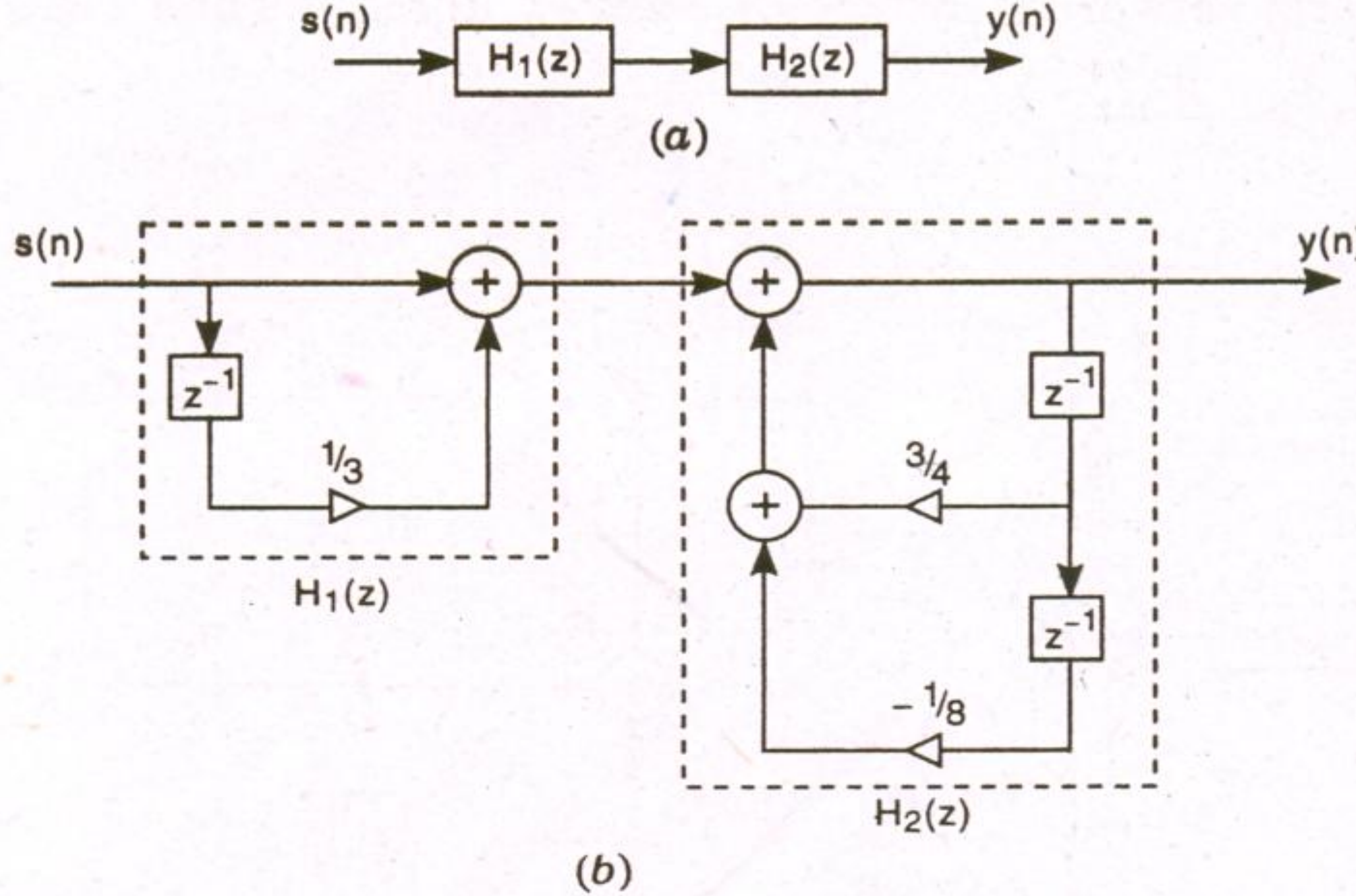
Fig. 7.10 *Structure of IInd order section in a parallel-form network structure realization.*

Example..

Example 7.2 Sketch the direct form-I, direct form-II, cascade and parallel-form network structures for the system characterized by following difference equations.

$$y(n) = \frac{3}{4}y(n-1) - \frac{1}{8}y(n-2) + s(n) + \frac{1}{3}s(n-1)$$

Direct Form I



(a) Block diagram of direct form-I of above problem.

(b) Direct form-I, network structure of above filter.

Direct Form II & Cascade Form

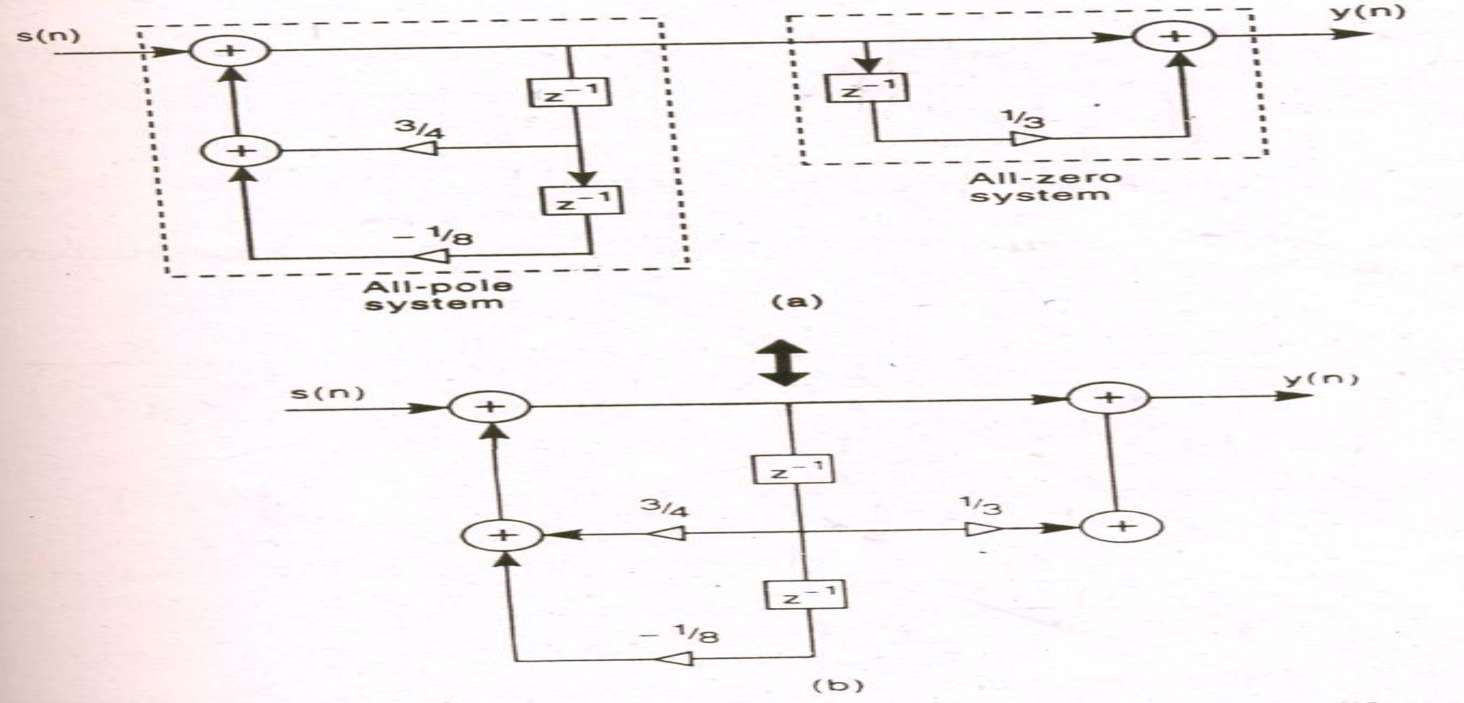


Fig. 7.12 *Direct form-II realization of above filter.*

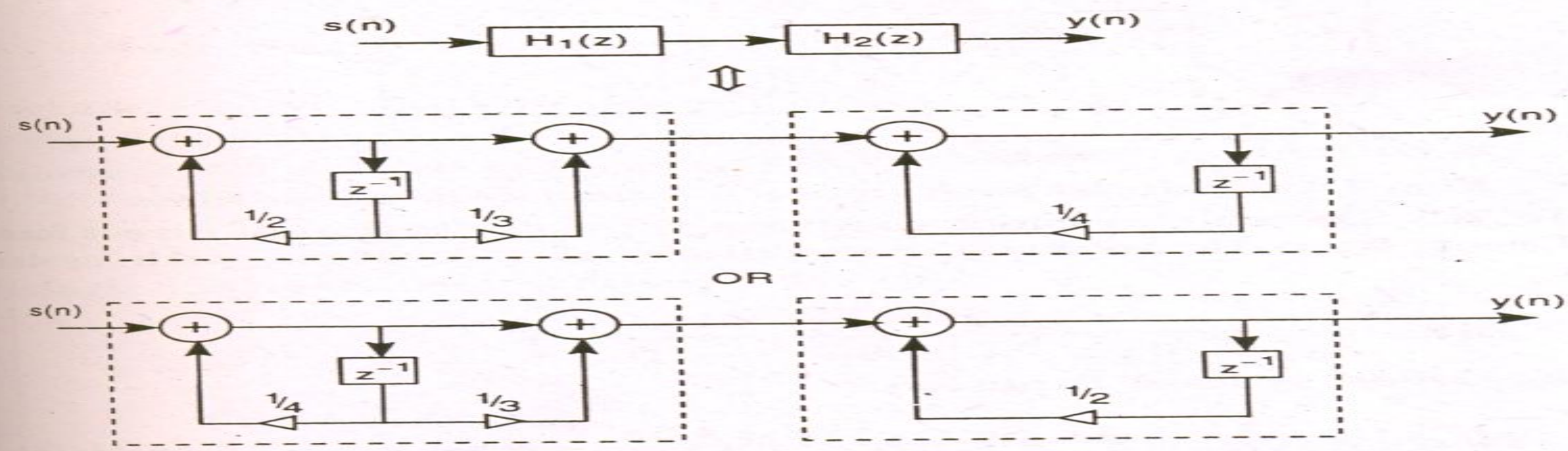


Fig. 7.13 *Cascade-form network structure of above filter.*

Parallel Form

- To get this we use PFE method :
- H(z)= Y(z)/X(z)
- Giving us A1=-7/3 & A2= 10/3 so implementing it we have -----

Parallel Form

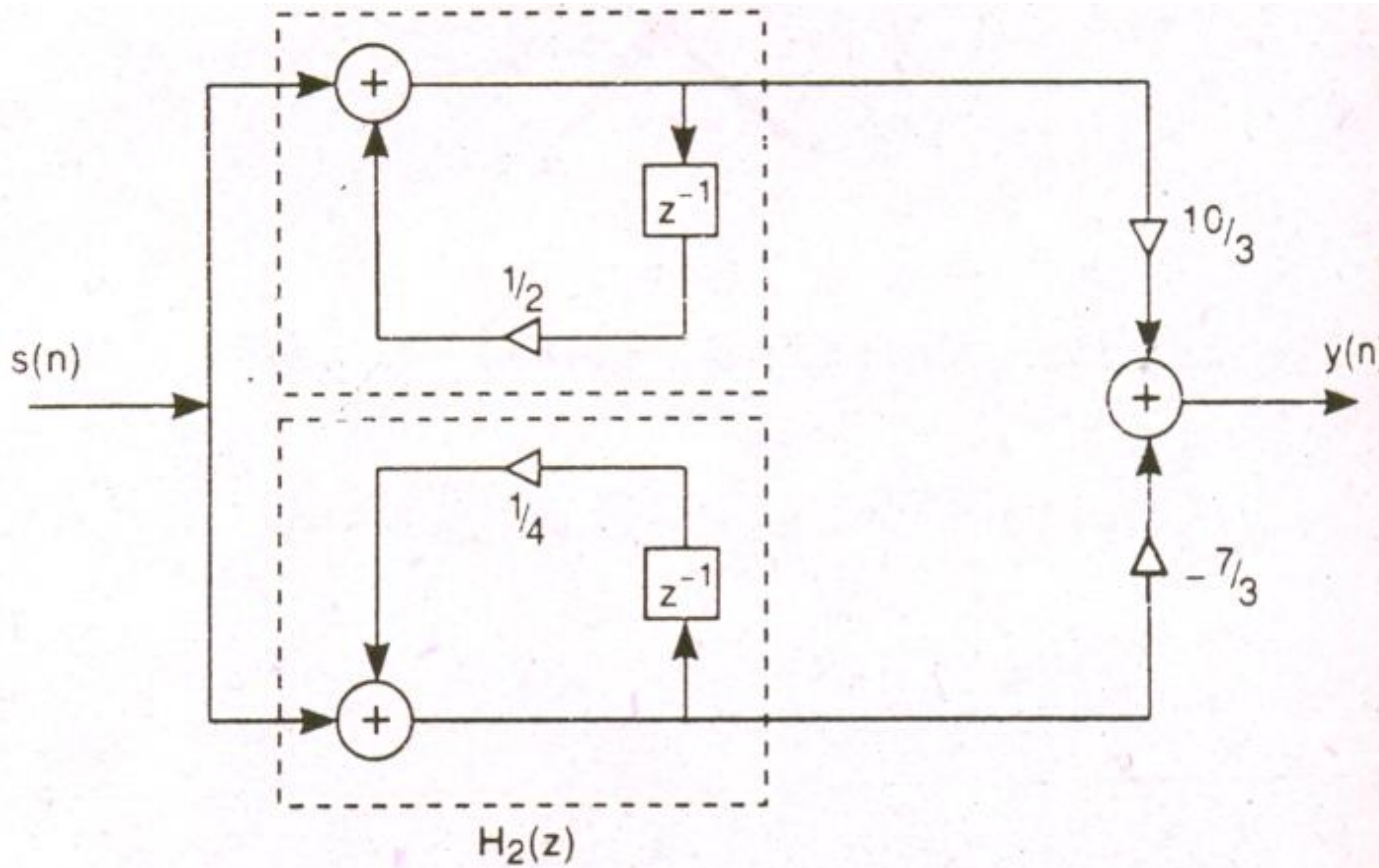
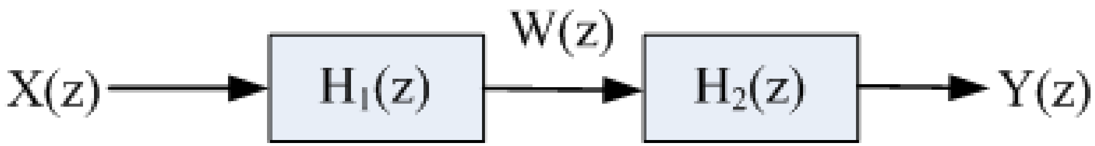


Fig. 7.14 *Parallel form network structure of above filter.*

- Direct Form I
  - Consider a third order IIR described by transfer function

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

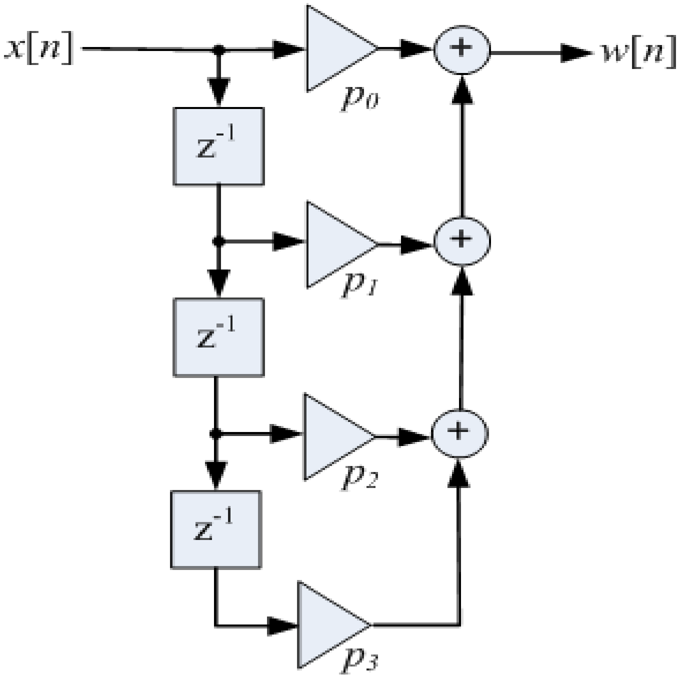
  - Implement as a cascade of two filter section

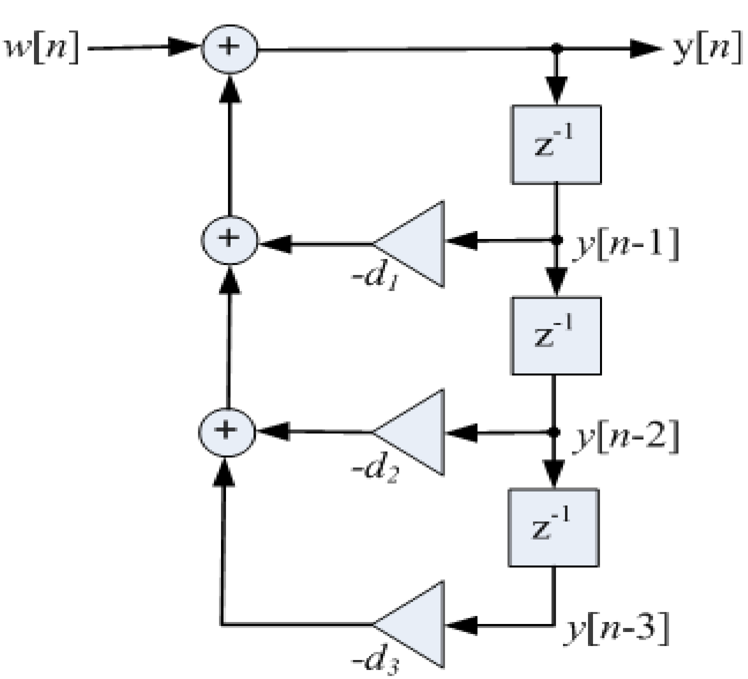


Where $$H_1(z) = \frac{W(z)}{X(z)} = P(z) = p_0 + p_1 z^{-1} + p_2 z^{-2} + p_3 z^{-3}$$

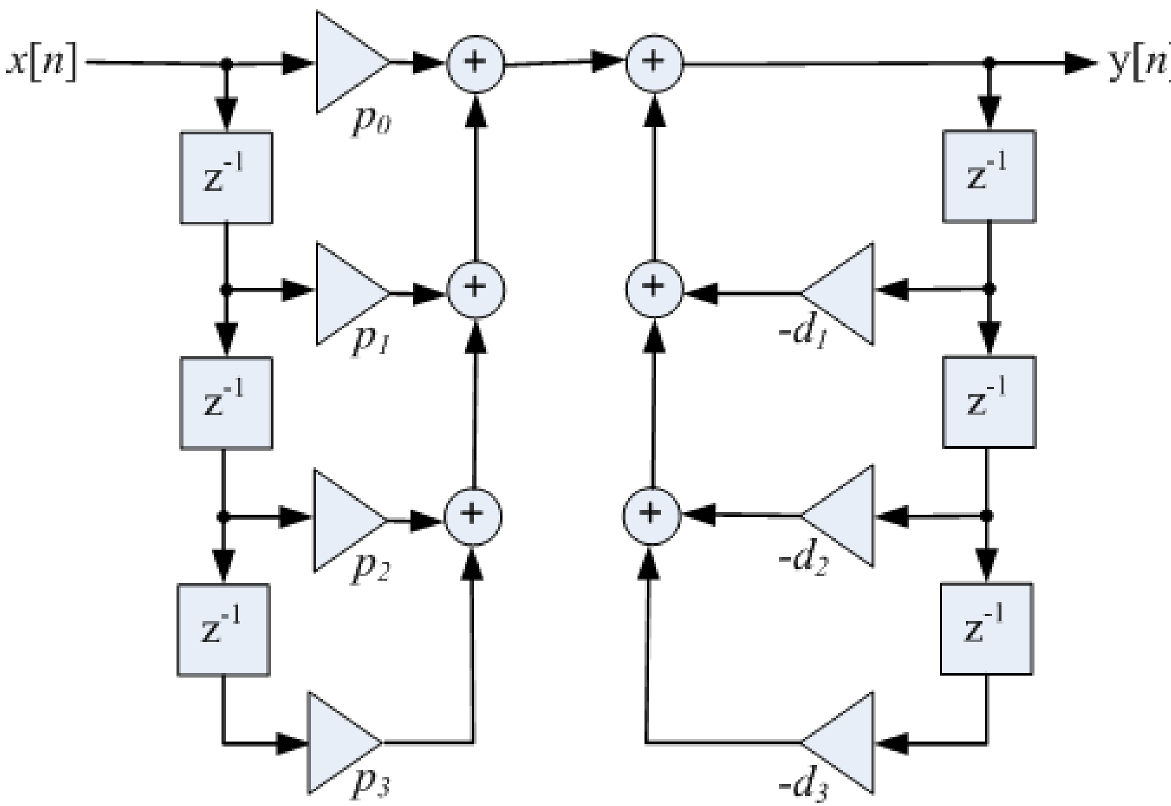
and $$H_2(z) = \frac{Y(z)}{W(z)} = \frac{1}{D(z)} = \frac{1}{1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}}$$

- Resulting in realization indicated below



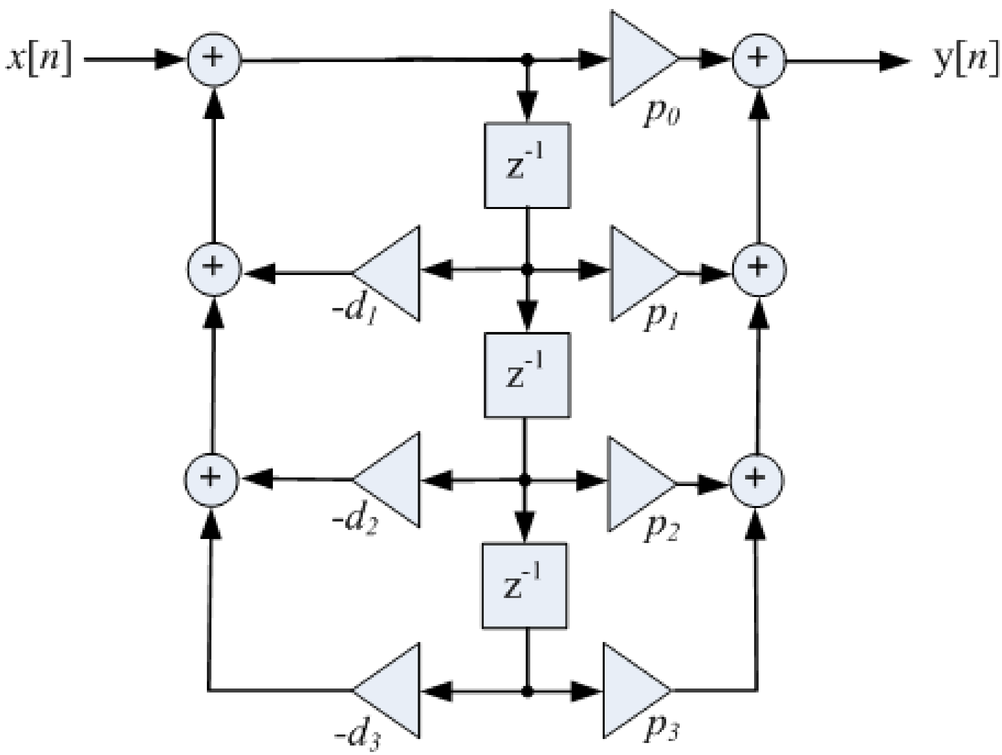


- Direct Form I



- Direct Form II (Canonic)

  – The two top delays can be shared



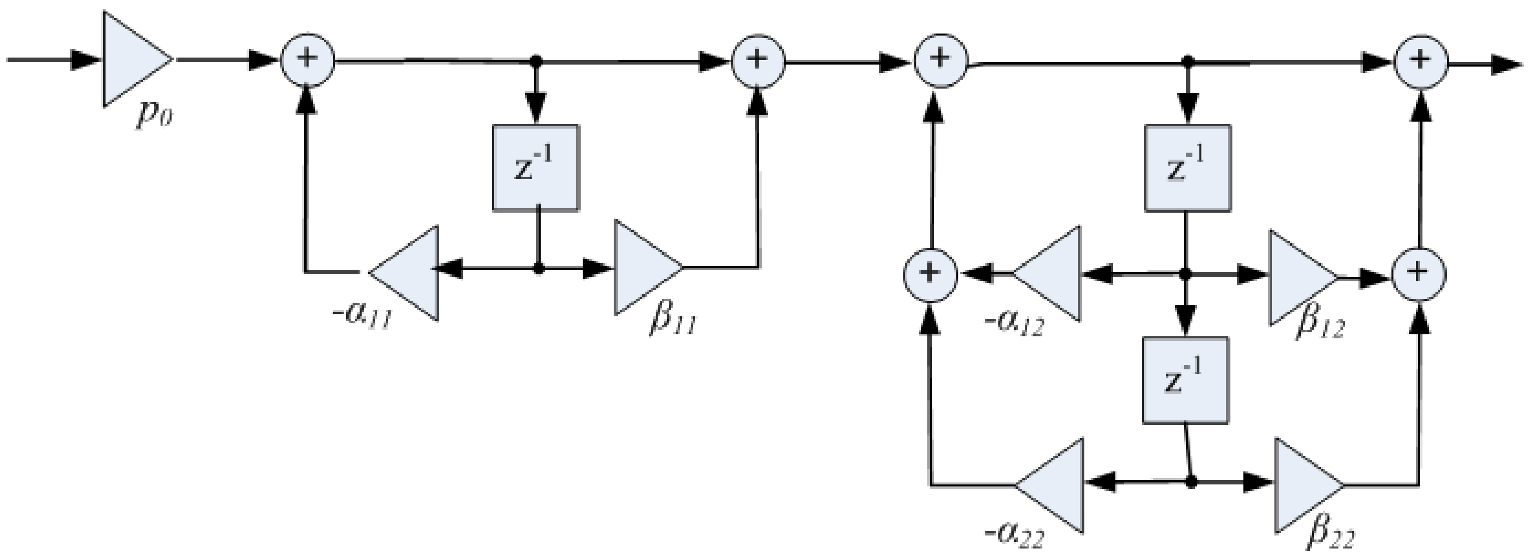
- Cascade Form

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

- A third order transfer function

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

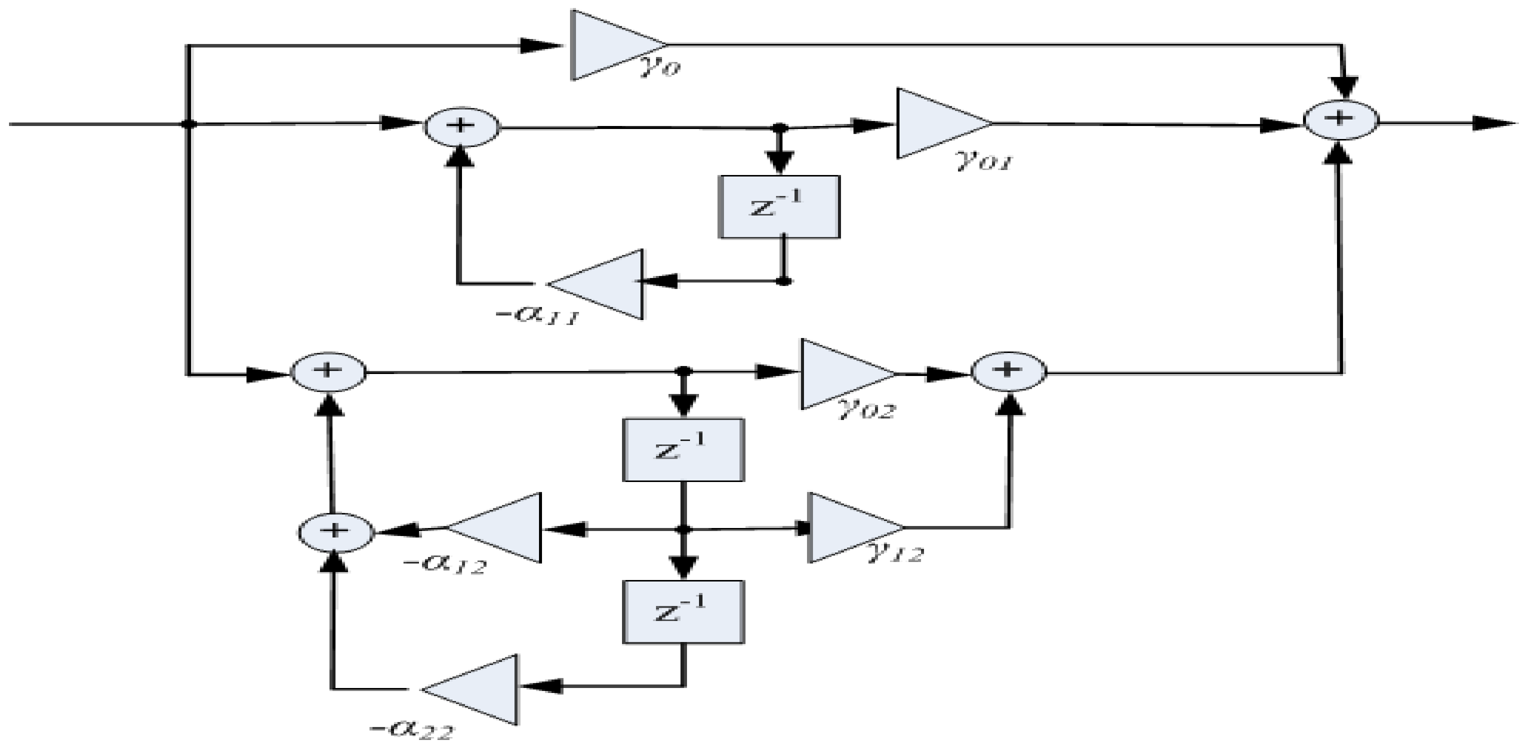
- Cascade Form



- Parallel Form: Use Partial Fraction Expansion Form to realize them

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

- Parallel Form: used in High Speed Filtering applications(as operated parallely)



Parallel IIR Structures

- Can express $H(z)$ as sum of terms (IZT)

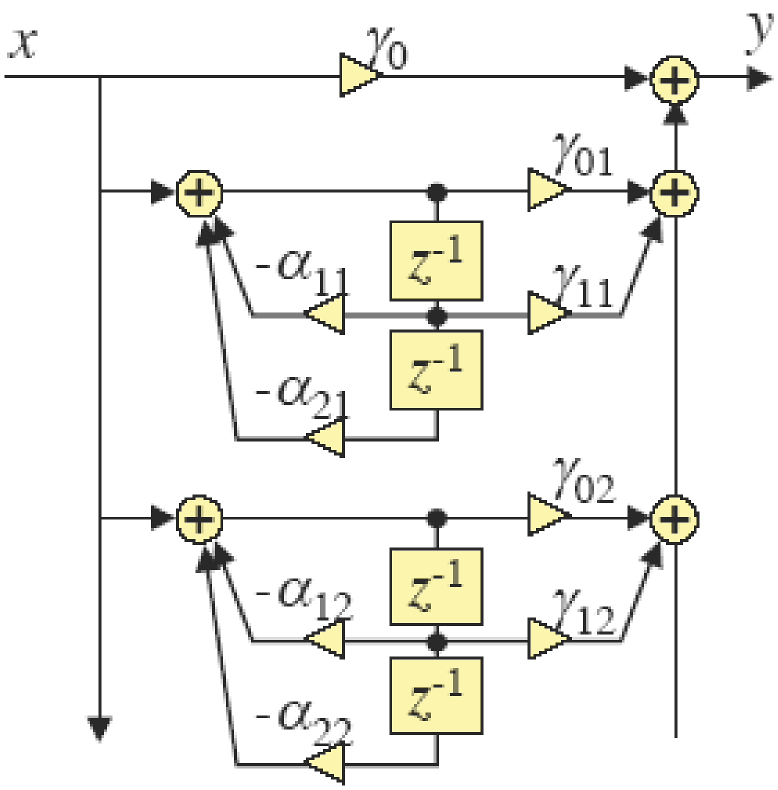
$$H(z) = \text{consts} + \sum_{\ell=1}^{N} \frac{\rho_\ell}{1 - \lambda_\ell z^{-1}} \qquad \rho_\ell = \left(1 - \lambda_\ell z^{-1}\right) F(z)\Big|_{z=\lambda_\ell}$$

- Or, second-order terms:

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

- Suggests parallel realization...

Parallel IIR Structures



- Sum terms become parallel paths
- Poles of each SOS are from full TF
- System zeros arise from output sum
- Why do this?
  - stability/sensitivity
  - reuse common terms