SEL4223 Digital Signal Processing

LTI System Function

Musa Mohd Mokji

System Function $H(z)$

- Also called transfer function
- It is a function in z-domain that relates the input signal $x[n]$ with the output signal $y[n]$ where

$$H(z) = \frac{Y(z)}{X(z)}$$

- Because $H(z)$ holds the relationship between $x[n]$ and $y[n]$, difference equation can be obtained from $H(z)$
- Difference equation is important in designing discrete system because it will be the last product before the system can be implemented into hardware. For Analog system, differential equation is obtained from Laplace transform before it is used to construct the analog circuit (R,L,C circuit).

Example 1

- Obtain difference equation for $H(z) = \frac{1+z^{-1}}{1+2z^{-1}+0.5z^{-2}}$

Solution:

$$H(z) = \frac{Y(z)}{X(z)}$$

$$\frac{Y(z)}{X(z)} = \frac{1+z^{-1}}{1+2z^{-1}+0.5z^{-2}}$$

$$Y(z)(1 + 2z^{-1} + 0.5z^{-2}) = X(z)(1 + z^{-1})$$

$$Y(z) + 2z^{-1}Y(z) + 0.5z^{-2}Y(z) = X(z) + z^{-1}X(z)$$

Example 1 (cont.)

- Using the delay property of z-transform, the difference equation is

$$y[n] + 2y[n-1] + 0.5y[n-2] = x[n] + x[n-1]$$

$$y[n] = x[n] + x[n-1] - 2y[n-1] - 0.5y[n-2]$$

- Note that the difference equation not only has the current and past input values but also past output value. In general, the difference equation can be written as

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

Example 1 (cont.)

- When the equation is transformed into z-domain, we obtain

$$Y(z)\sum_{k=0}^{N} a_k z^{-k} = X(z)\sum_{k=0}^{M} b_k z^{-k}$$

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

- Thus, in order to obtain difference equation from $H(z)$, make sure the $H(z)$ is in the form of above equation.

Example 2

- Find difference equation for $H(z) = \frac{z^{-1}}{(1-z^{-1})(1-0.5z^{-1})}$

Solution:

- $H(z) = \frac{z^{-1}}{1-1.5z^{-1}+0.5z^{-2}}$
- $H(z) = \frac{Y(z)}{X(z)}$

$$Y(z)(1 - 1.5z^{-1} + 0.5z^{-2}) = X(z)(z^{-1})$$

$$Y(z) - 1.5z^{-1}Y(z) + 0.5z^{-2}Y(z) = z^{-1}X(z)$$

$$y[n] - 1.5y[n-1] + 0.5y[n-2] = x[n-1]$$

Example 3

- Find difference equation for $h[n] = 0.5^n u[n]$

Solution:

$$H(z) = \frac{1}{1-0.5z^{-1}}$$

$$\frac{Y(z)}{X(z)} = \frac{1}{1-0.5z^{-1}}$$

$$Y(z)(1 - 0.5z^{-1}) = X(z)$$

$$y[n] - 0.5y[n-1] = x[n]$$

$$y[n] = x[n] + 0.5y[n-1]$$

Example 4

- Find difference equation for $h[n] = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$

Solution:

a) Because this is an FIR system, convolution can be used to obtain the difference equation.

$$y[n] = \sum_{k=0}^{9} h[k]x[n-k]$$

$$= h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + h[3]x[n-3] + h[4]x[n-4] + h[5]x[n-5] + h[6]x[n-6] + h[7]x[n-7] + h[8]x[n-8] + h[9]x[n-9]$$

Example 4 (cont.)

$$y[n] = x[n] + x[n-1] + x[n-2] + x[n-3] + x[n-4] + x[n-5] + x[n-6] + x[n-7] + x[n-8] + x[n-9]$$

The sequence of the difference equation is too long. Let's look at different technique.

b) Now, rearrange the equation, do z-transform and obtain the difference equation

$$h[n] = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1] = u[n] - u[n-10]$$

Example 4 (cont.)

$$H(z) = \frac{1}{1-z^{-1}} - \frac{z^{-10}}{1-z^{-1}}$$

$$= \frac{1-z^{-10}}{1-z^{-1}}$$

$$\frac{Y(z)}{X(z)} = \frac{1-z^{-10}}{1-z^{-1}}$$

$$Y(z)(1-z^{-1}) = X(z)(1-z^{-10})$$

$$y[n] - y[n-1] = x[n] - x[n-10]$$

$y[n] = x[n] - x[n-10] + y[n-1]$ → The sequence is shorter.

Discrete System Realization

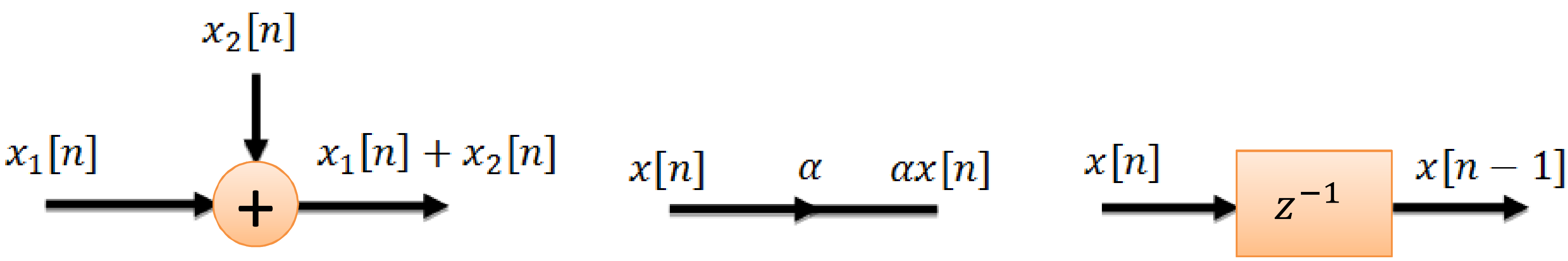
- After a filter is designed, it must be *realized* by developing a signal flow diagram that describes the filter in terms of operations on sample sequences.

- A given transfer function may be realized in many ways. Consider how a simple expression such as $ax + bx + c$ could be evaluated – one could also compute the equivalent $x(a + b) + c$.

- Specifically, some realizations are more efficient in terms of the number of operations or storage elements required for their implementation, and others provide advantages such as improved numerical stability and reduced round-off error.

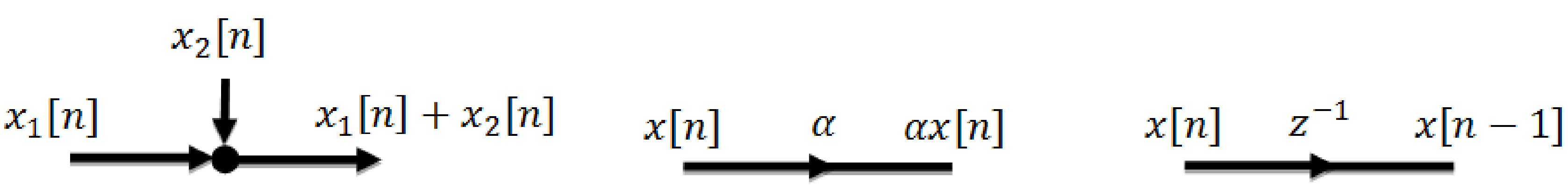
Realization Symbols

Block Diagram:



Addition *Multiplication* *Delay*

Signal Flow Graph:



Addition *Multiplication* *Delay*

Discrete System Realization (cont.)

- In general, system function of a system is represents by

$$\frac{Y(z)}{X(z)} = H(z) = \frac{N(z)}{D(z)} = H_1(z)H_2(z)$$

- Where

$$H_1(z) = N(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_N z^{-M}$$

$$H_2(z) = \frac{1}{D(z)} = \frac{1}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_M z^{-N}}$$

- By manipulating above equation, discrete system can be realized in several ways

FIR: Direct-Form Structure

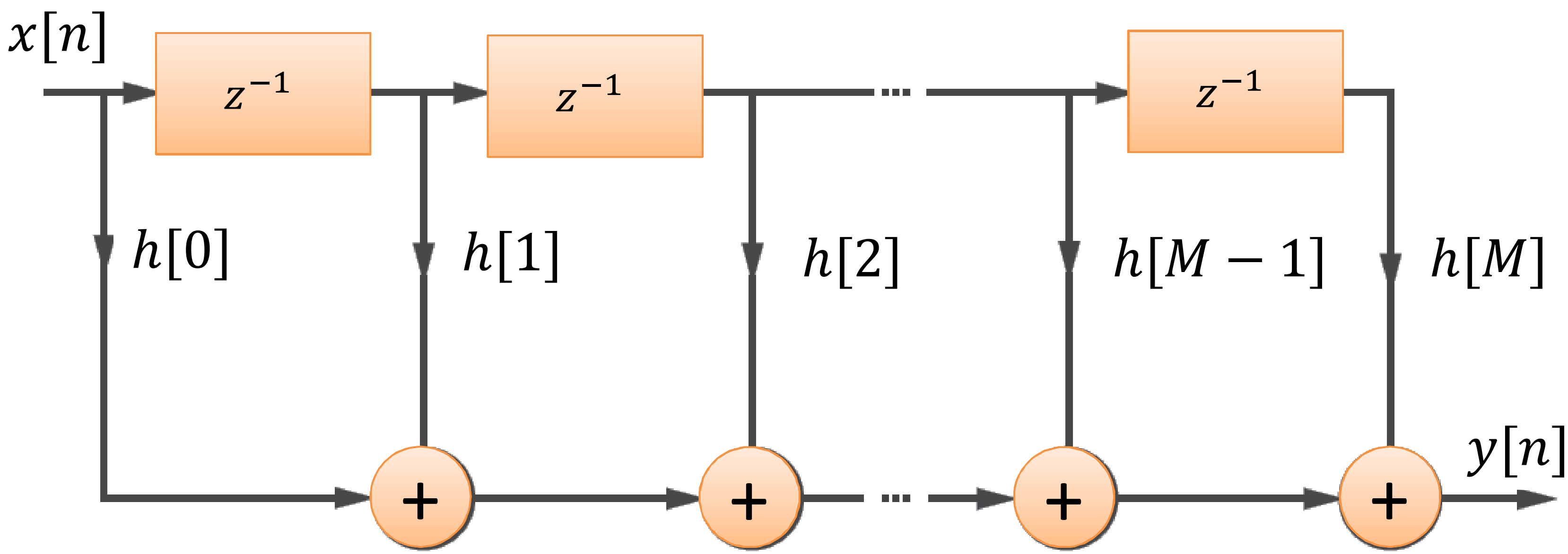
- For FIR system, there will be no past and future output. Thus the system does not have $h_2[n]$ where it can be represented by

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

- Where $h[n] = h_1[n]$ and $M + 1$ is the length of the $h[n]$

- Based on above equation, there will be M number of addition and $M + 1$ number of multiplication. Number of memory needed is M. The next figure shows the structure.

FIR: Direct-Form Structure



FIR: Direct-Form Structure (linear phase)

- For linear phase FIR system where $h[n]$ is always either symmetry or anti-symmetry

$$h[n] = \begin{cases} h[M-n] & for\ Type\ I\ and\ Type\ II \\ -h[M-n] & for\ Type\ III\ and\ Type\ IV \end{cases}$$

- Thus,

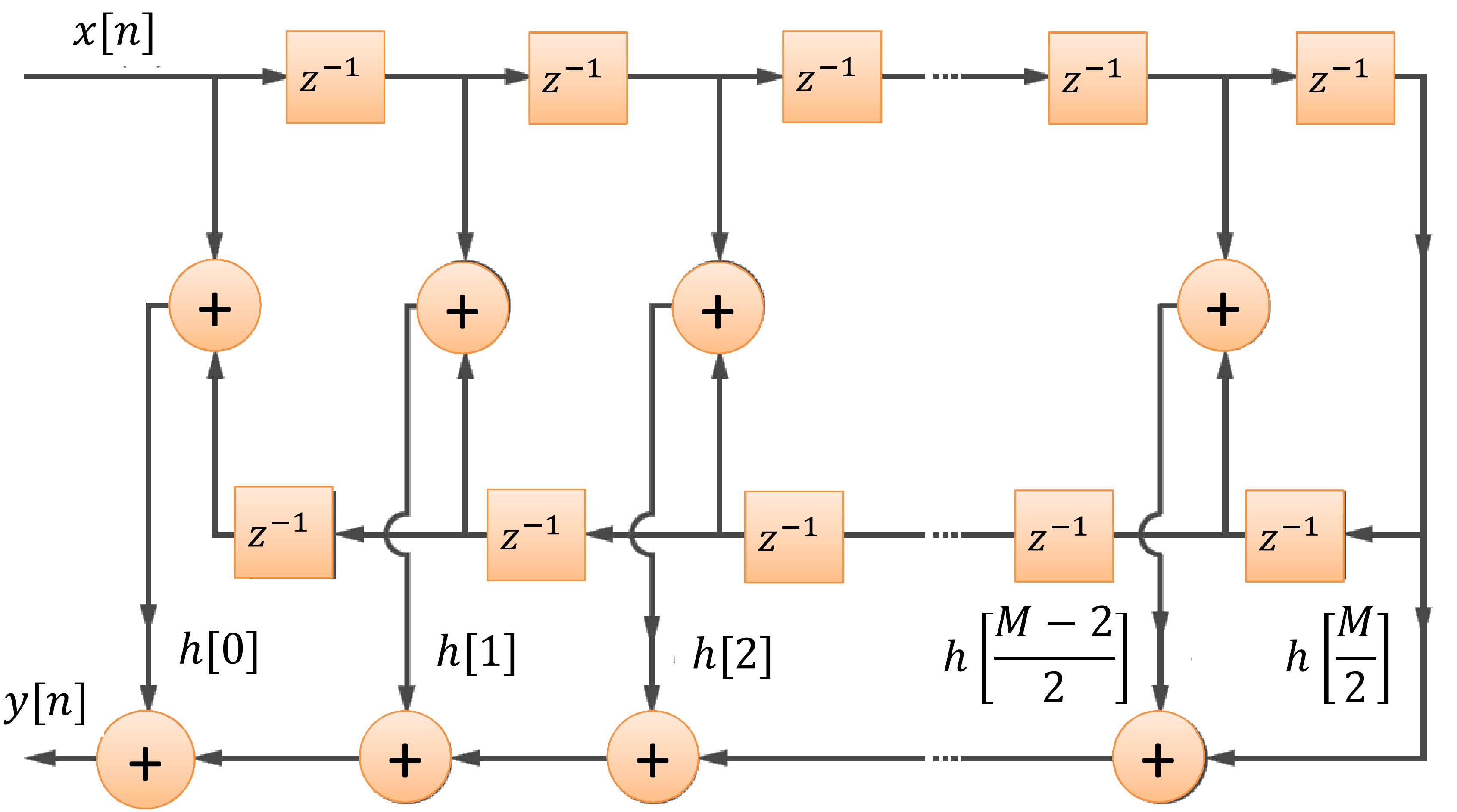
$$y[n] = \sum_{k=0}^{P} h[k](x[k] + x[n-M+k])$$

Where $P = \frac{M}{2}$ for Type I and III (M even) and $P = \frac{M-1}{2}$ for Type II and IV (M odd).

FIR: Direct-Form Structure (linear phase)

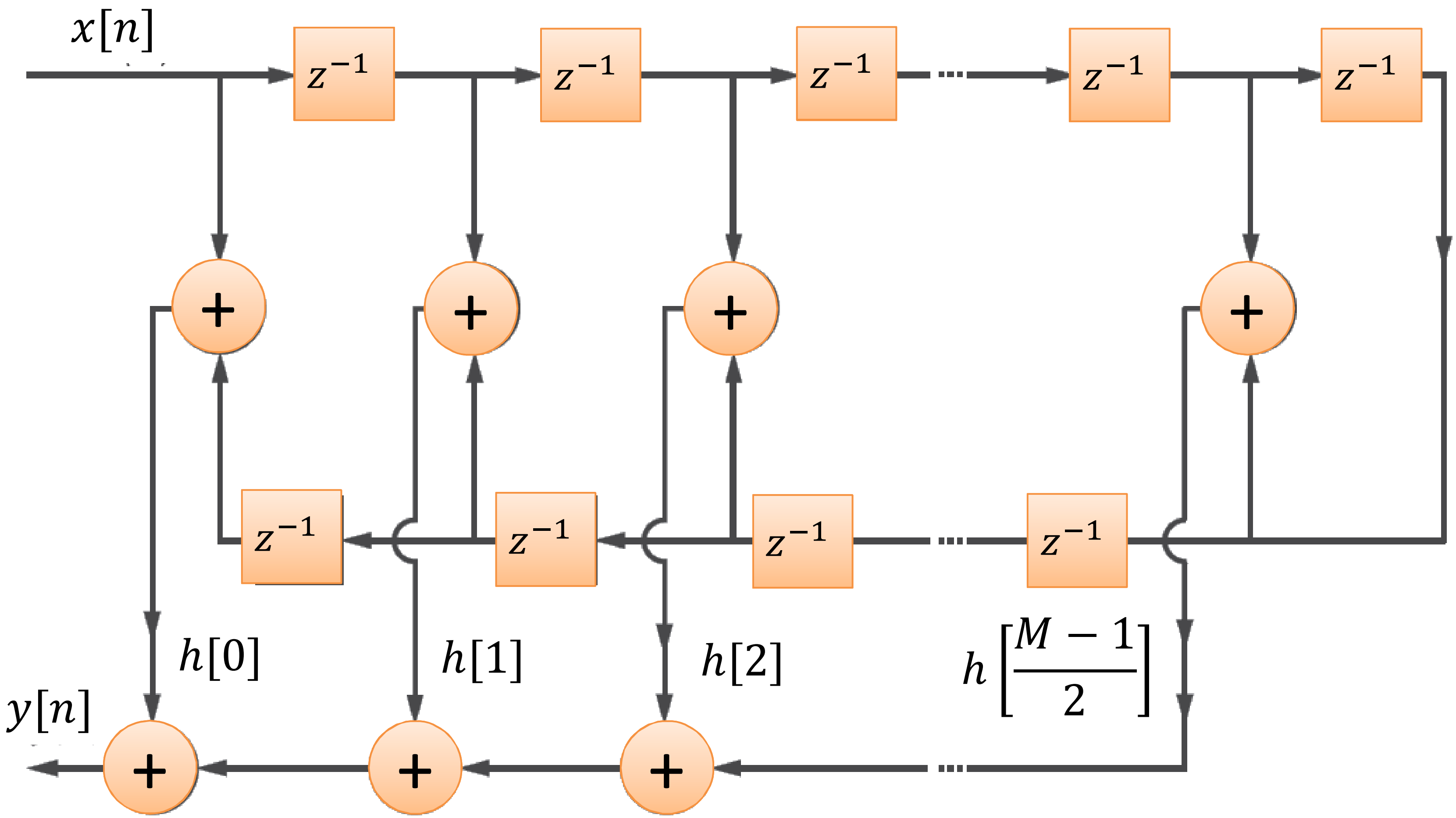
- From the previous equation:
  - The number of multiplications is reduced to $M/2$ for M even and to $(M-1)/2$ for M odd compare to the previous general FIR system.
  - The number of addition and memory needed are similar which both are M.

FIR: Direct-Form Structure (linear phase)



FIR Linear Phase Type I and III realization

FIR: Direct-Form Structure (linear phase)



FIR Linear Phase Type II and IV realization

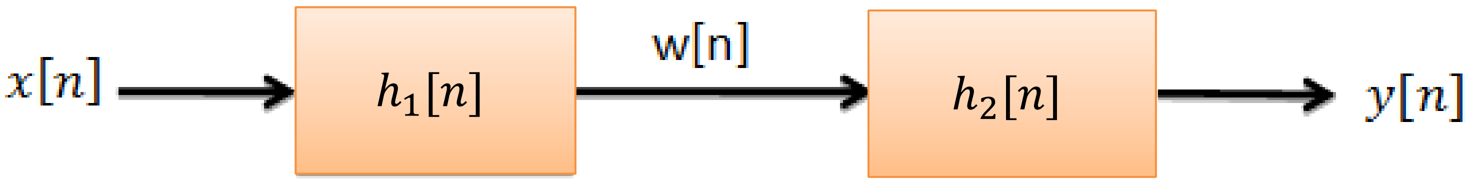
IIR: Direct Form I

- The realization of the system is straight forward where input will go through $H_1(z)$ first and then the $H_2(z)$

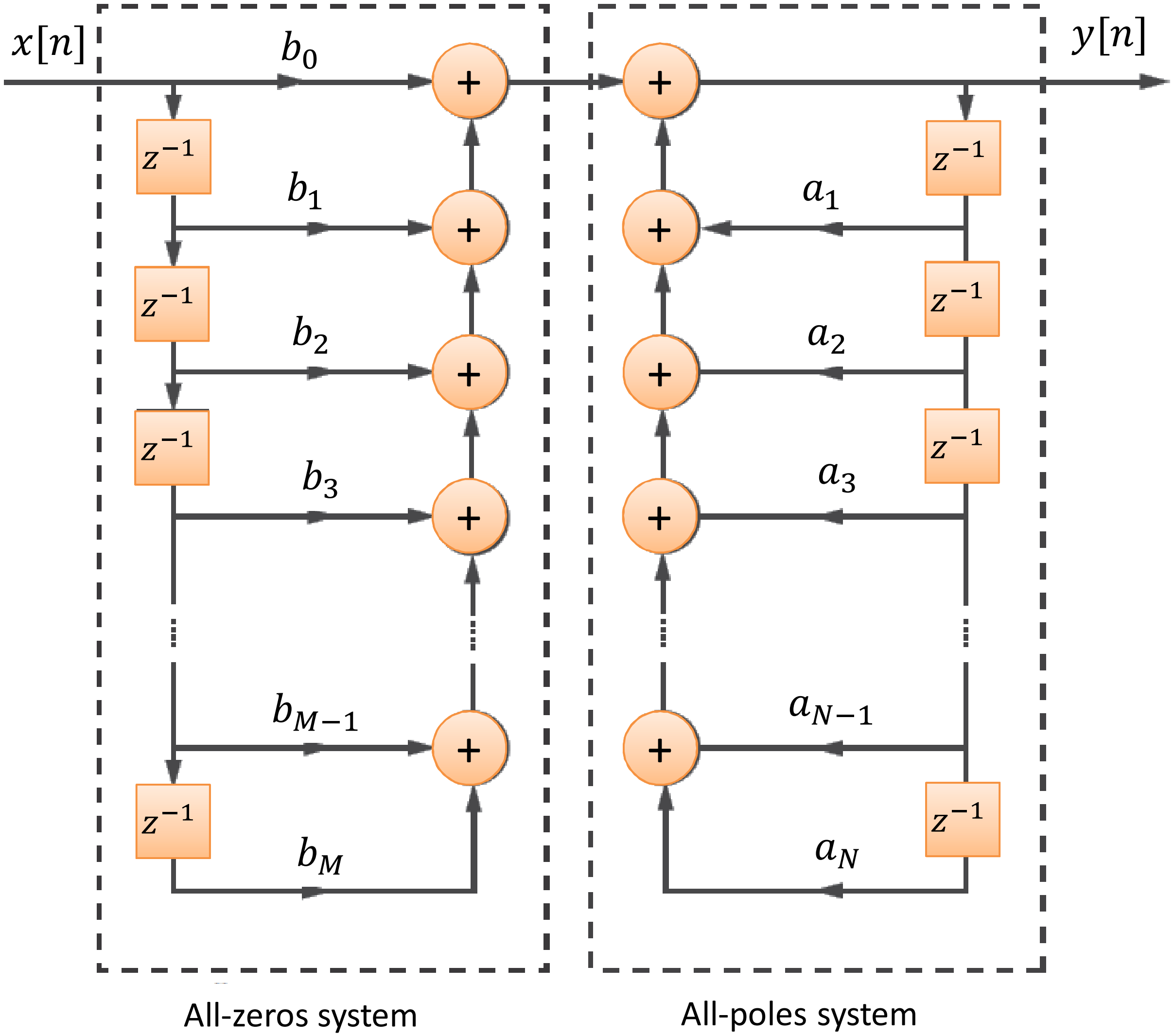
$$Y(z) = W(z)H_2(z) \quad or \quad y[n] = w[n] - a_1 y[n-1] - \cdots - a_M y[n-N]$$

- Where

$$W(z) = X(z)H_1(z) \quad or \quad w[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_N x[n-M]$$



IIR: Direct Form I



Number of operation and memory:

- Addition: $M + N$
- Multiplication: $M + N + 1$
- Memory: $M + N$

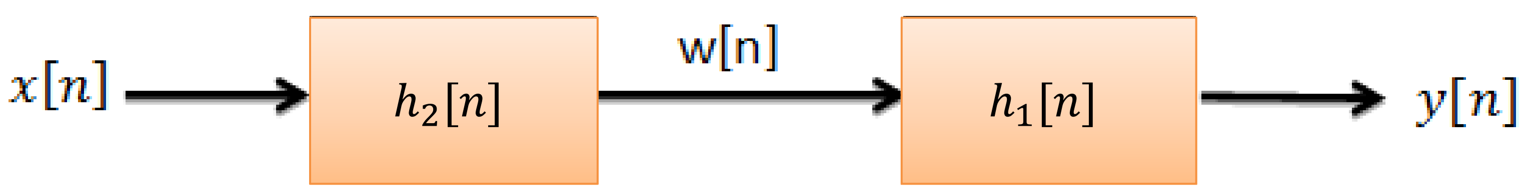
IIR: Direct Form II

- For Direct Form II, input sequence will go through $h_2[n]$ first and then $h_1[n]$

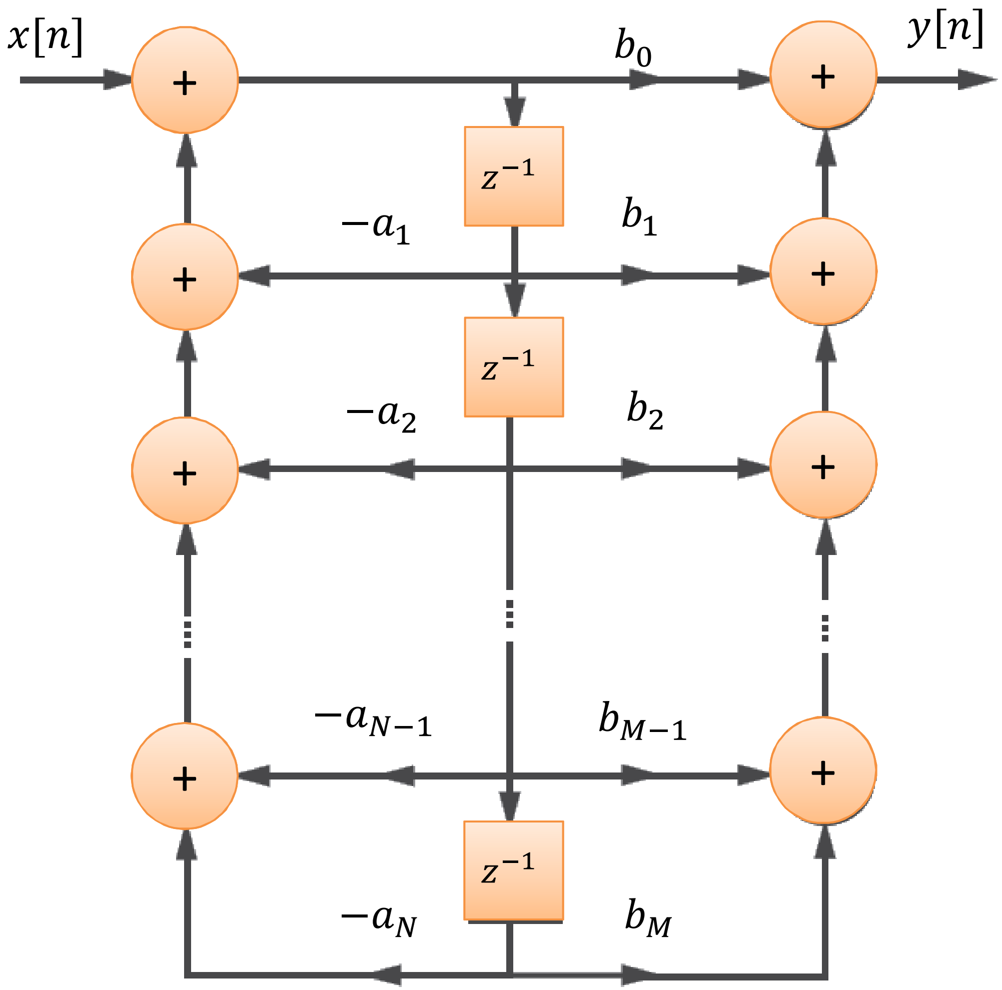
$$Y(z) = W(z)H_1(z) \quad or \quad y[n] = b_0 w[n] + b_1 w[n-1] + \cdots + b_N w[n-M]$$

- Where

$$W(z) = X(z)H_2(z) \quad or \quad w[n] = x[n] - a_1 w[n-1] - \cdots - a_M w[n-N]$$



IIR: Direct Form II

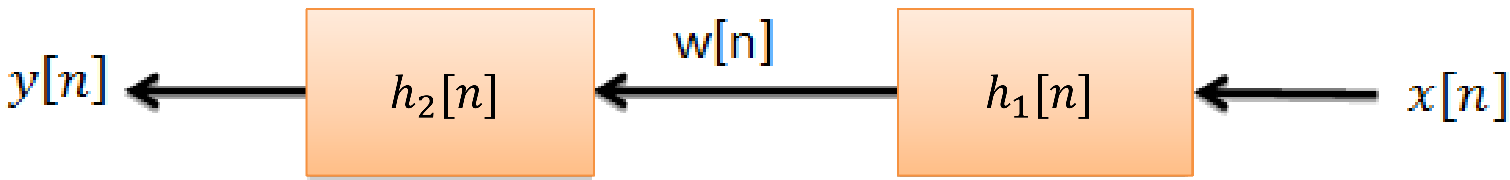


Number of operation and memory:

- Addition:
  $M + N$
- Multiplication:
  $M + N + 1$
- Memory:
  $M \; or \; N$
  whichever is higher

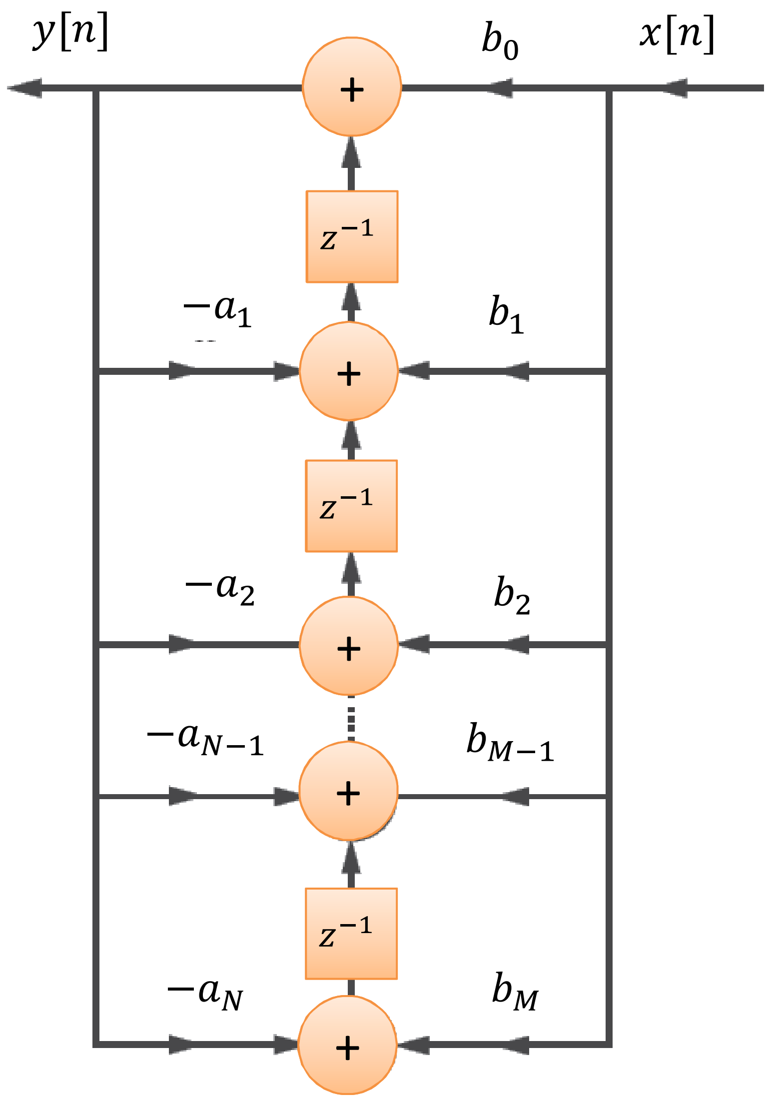
IIR: Transpose Direct Form II

- In this form, the difference equation is rearranged so that current input will be at the right hand side of the equation instead of the current output as seen in the previous two forms.



- From figure above, this realization seems to be similar to the Direct Form I where $x[n]$ will go through $h_1[n]$ first and then $H_2[n]$. However the structure is different as shown in the next figure.

IIR: Transpose Direct Form II



- Number of operation and memory:
  - Addition:
    $M+1 \;\; or \;\; N+1$
    whichever is higher
  - Multiplication:
    $M+N+1$
  - Memory:
    $M \; or \; N$
    whichever is higher

IIR: Transpose Direct Form II (cont.)

- When $M = N$, the mathematical manipulation of this system is as follow

$$y[n] = b_0 x[n] + w_1[n-1]$$

$$w_k[n] = w_{k+1}[n-1] + b_k x[n] - a_k y[n], \qquad k = 1, 2, \dots, N-1$$

$$w_N[n] = b_N x[n] - a_N y[n]$$

- If $M > N$, weight $a_{N+1}\ to\ a_M$ are equals to zero.
- Otherwise, if $N > M$, weight $b_{M+1}\ to\ b_N$ are equals to zero.

References

1) John G. Proakis, Dimitris K Manolakis, "Digital Signal Processing: Principle, Algorithm and Applications", Prentice-Hall, 4th edition (2006).

2) Sanjit K. Mitra, "Digital Signal Processing-A Computer Based Approach", McGraw-Hill Companies, 3rd edition (2005).

3) Alan V. Oppenheim, Ronald W. Schafer, "Discrete-Time Signal Processing", Prentice-Hall, 3rd edition (2009).