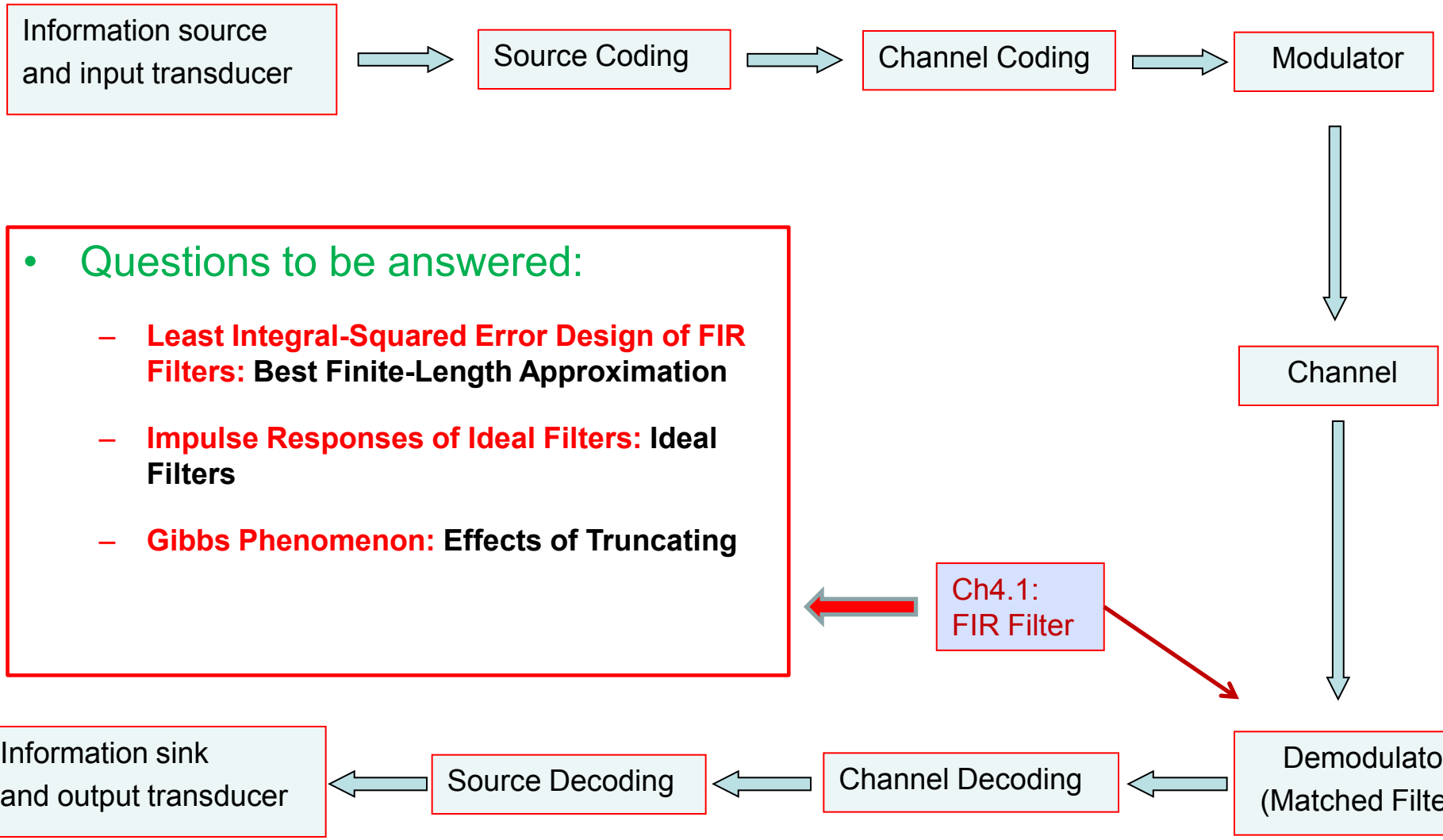
Ch4.1: FIR Filter Design



- Questions to be answered:
  - **Least Integral-Squared Error Design of FIR Filters: Best Finite-Length Approximation**
  - **Impulse Responses of Ideal Filters: Ideal Filters**
  - **Gibbs Phenomenon: Effects of Truncating**

Reading: Mitra (10.1, 10.2)

IIR Vs. FIR Filter

	IIR Filters	FIR Filters
Phase (grp delay)	difficult to control, no particular techniques available	linear phase always possible
Stability	can be unstable, can have limit cycles	always stable, no limit cycles
Order	less	more
History	derived from analog filters	no analog history
Others		polyphase implementation possible can always be made causal

Common method for IIR filter design: Transform a continuous-time filter into a discrete-time filter meeting prescribed specifications I.e., map *s*-plane to *z*-plane to obtain desired IIR filter.

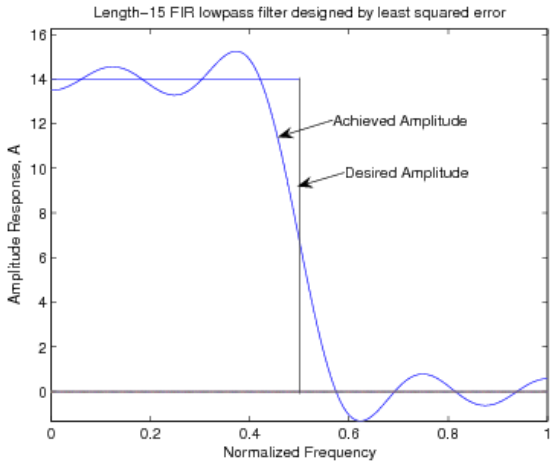
FIR Filter

- FIR filters almost entirely restricted to discrete-time implementations (**unlike IIR implementations**).
- Design techniques for FIR filters are based on **directly approximating** the desired frequency response of the DT system.
- Compared to IIR filters, FIR filters can be designed to have **linear phase**.
- The simplest method of FIR filter design is **window method.**

Ch4.1: FIR Filter Design

- **Least Integral-Squared Error Design of FIR Filters**

- Gibbs Phenomenon



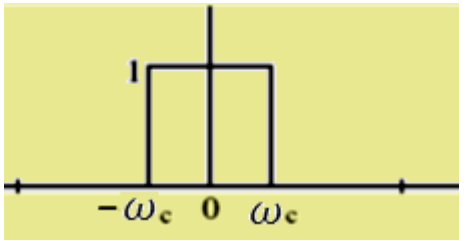
Least Integral-Squared Error Design of FIR Filters

- Design techniques for FIR filters are based on **directly approximating** the desired frequency response of the DT system.
- Let $H_d(e^{j\omega})$ denote the desired frequency response.
- Since $H_d(e^{j\omega})$ is a periodic function of ω, it can be expressed as a Fourier series

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d[n]e^{-j\omega n}$$

where

$$h_d[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})\, e^{j\omega n} d\omega$$



- In general, $H_d(e^{j\omega})$ is **piecewise constant with sharp transitions** between bands. Consider the low-pass filter on the right.
- In this case, $h_d[n]$ is of **infinite length and noncausal**.

Least Integral-Squared Error Design of FIR Filters

- **Objective: Find a finite-duration $h_t[n]$ of length 2*M*+1 whose DTFT $H_t(e^{j\omega})$ approximates the desired DTFT $H_d(e^{j\omega})$ in some sense.**

- Commonly used approximation criterion (Integral-Squared Error)

$$\Phi = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|H_t(e^{j\omega}) - H_d(e^{j\omega})\right|^2 d\omega$$

  where $H_t(e^{j\omega}) = \sum_{n=-M}^{M} h_t[n]e^{-j\omega n}$.

- Using Parseval's relation, we can write

$$\Phi = \sum_{n=-\infty}^{\infty} |h_t[n] - h_d[n]|^2$$

$$= \sum_{n=-M}^{M} |h_t[n] - h_d[n]|^2 + \sum_{n=-\infty}^{-M-1} |h_d[n]|^2 + \sum_{n=M+1}^{\infty} |h_d[n]|^2$$

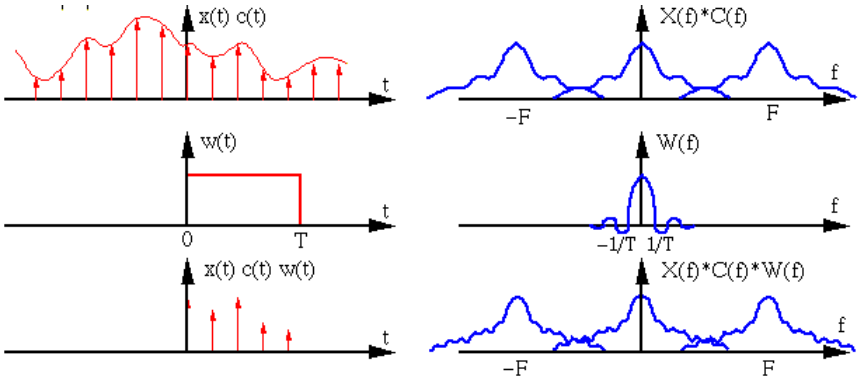
- It follows that Φ is minimum when $h_t[n] = h_d[n]$ for $-M \le n \le M$.

Least Integral-Squared Error Design of FIR Filters

- Best finite-length approximation to ideal infinite-length impulse response in the **mean-square sense** is obtained by **truncation**.
- A causal FIR filter with an impulse response $h[n]$ can be derived from $h_t[n]$ by delaying:

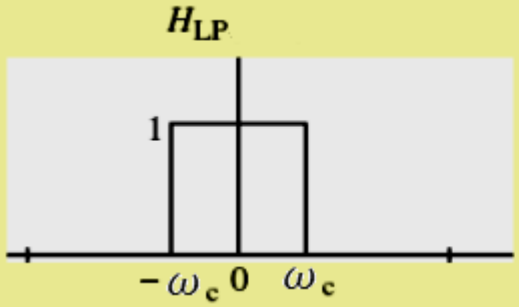
$$h[n] = h_t[n - M]$$

- The causal FIR filter $h[n]$ has the same magnitude response as $h_t[n]$ and its phase response has a linear phase shift of ωM radians with respect to that of $h_t[n]$.



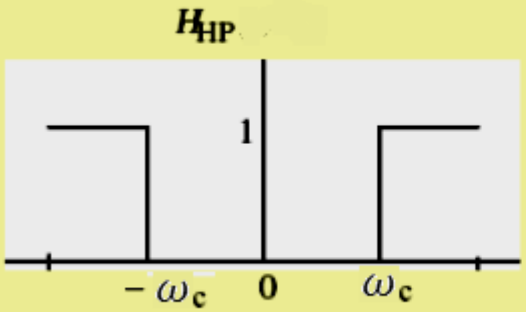
Impulse Responses of Ideal Filters

- Ideal lowpass filter -



$$h_{LP}[n] = \frac{\sin \omega_c n}{\pi n}, \quad -\infty \le n \le \infty$$

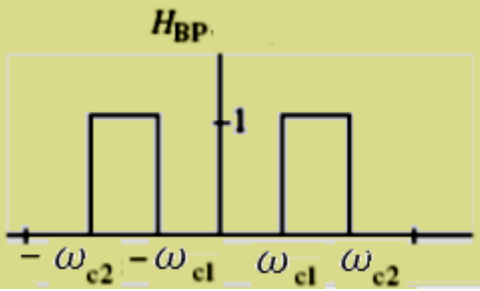
- Ideal highpass filter - Can we get it from the above LPF?



$$h_{HP}[n] = \begin{cases} 1 - \dfrac{\omega_c}{\pi}, & n = 0 \\ -\dfrac{\sin(\omega_c n)}{\pi n}, & n \neq 0 \end{cases}$$

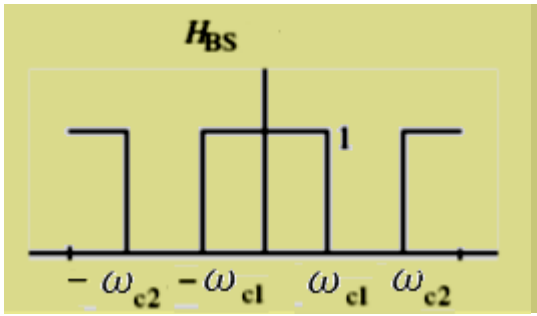
Impulse Responses of Ideal Filters

- Ideal bandpass filter -Can we get it from the above LPF?



$$h_{BP}[n] = \begin{cases} \dfrac{\sin(\omega_{c2}n)}{\pi n} - \dfrac{\sin(\omega_{c1}n)}{\pi n}, & n \neq 0 \\ \dfrac{\omega_{c2}}{\pi} - \dfrac{\omega_{c1}}{\pi}, & n = 0 \end{cases}$$

- Ideal bandstop filter -Can we get it from the LPF and HPF?

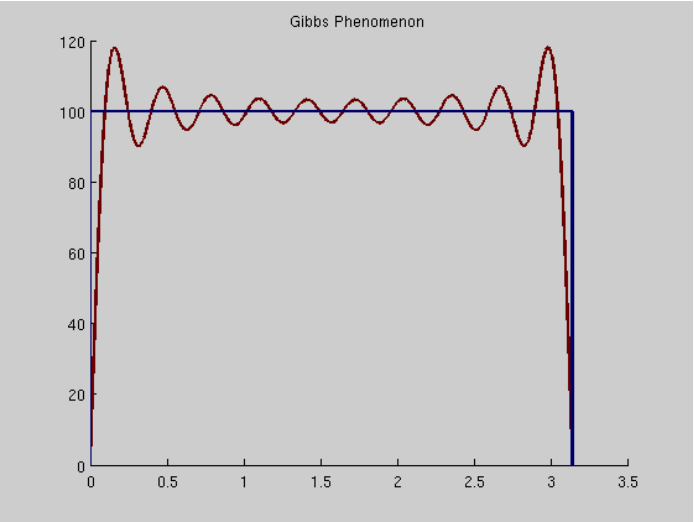


$$h_{BS}[n] = \begin{cases} 1 - \dfrac{(\omega_{c2} - \omega_{c1})}{\pi}, & n = 0 \\ \dfrac{\sin(\omega_{c1}n)}{\pi n} - \dfrac{\sin(\omega_{c2}n)}{\pi n}, & n \neq 0 \end{cases}$$

Ch4.1 : FIR Filter Design

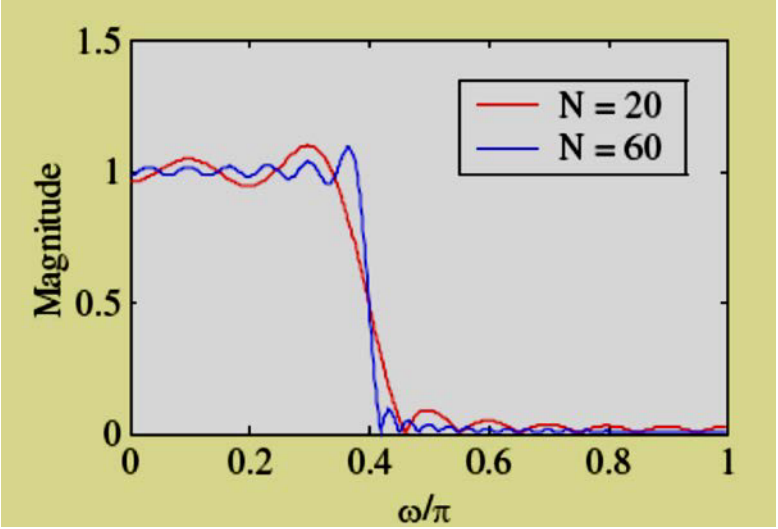
- Least Integral-Squared Error Design of FIR Filters

- **Gibbs Phenomenon**



Gibbs Phenomenon

- **Gibbs phenomenon** - Oscillatory behavior in the magnitude responses of causal FIR filters obtained by truncating the impulse response coefficients of ideal filters.



- As can be seen, as the length of the lowpass filter increases, the **number of ripples** in both passband and stopband increases, with a corresponding decrease in the **ripple widths**.
- Height of the largest ripples remain the same, independent of length.

- Similar oscillatory behavior is observed in the magnitude responses of the truncated versions of other types of ideal filters.

Gibbs Phenomenon: Explanation

- Gibbs phenomenon can be explained by treating the truncation operation as a **windowing operation**: $h_t[n] = h_d[n]w[n]$

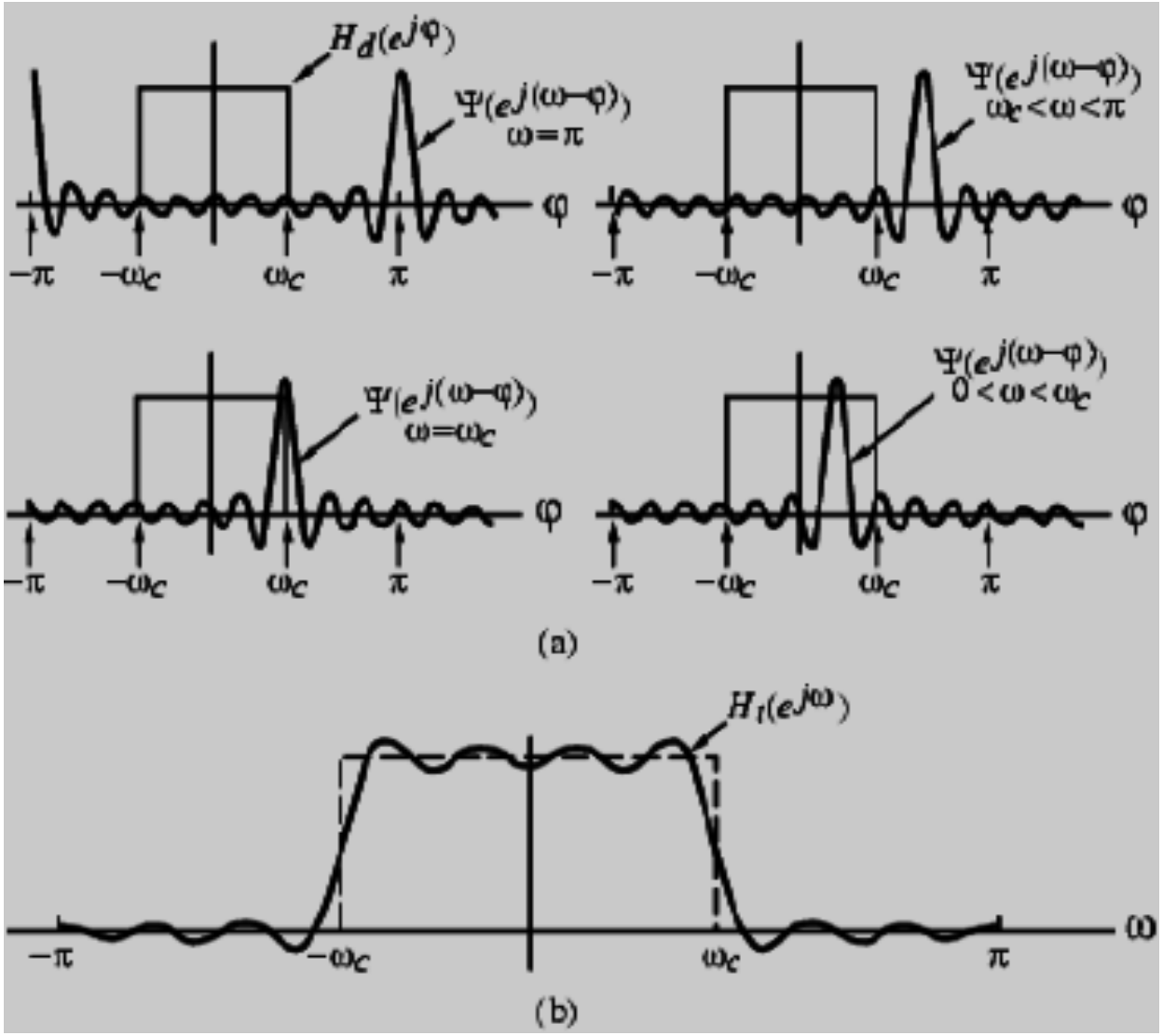
- In the frequency domain,

$$H_t(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d\left(e^{j\omega}\right)\Psi_w\left(e^{j(\omega-\varphi)}\right)d\varphi$$

  where $H_t(e^{j\omega})$ and $\Psi_w\left(e^{j\omega}\right)$ are the DTFTs of $h_t[n]$ and $w[n]$, respectively. Thus, $H_t(e^{j\omega})$ is obtained by a **periodic continuous convolution** of $H_d\left(e^{j\omega}\right)$ and $\Psi_w\left(e^{j(\omega-\varphi)}\right)$.

- If $\Psi_w\left(e^{j\omega}\right)$ is a very narrow pulse centered at $\omega = 0$ (ideally a delta function) compared to variations in $H_d\left(e^{j\omega}\right)$, then $H_t(e^{j\omega})$ will approximate $H_d\left(e^{j\omega}\right)$ very closely.

Gibbs Phenomenon



Gibbs Phenomenon

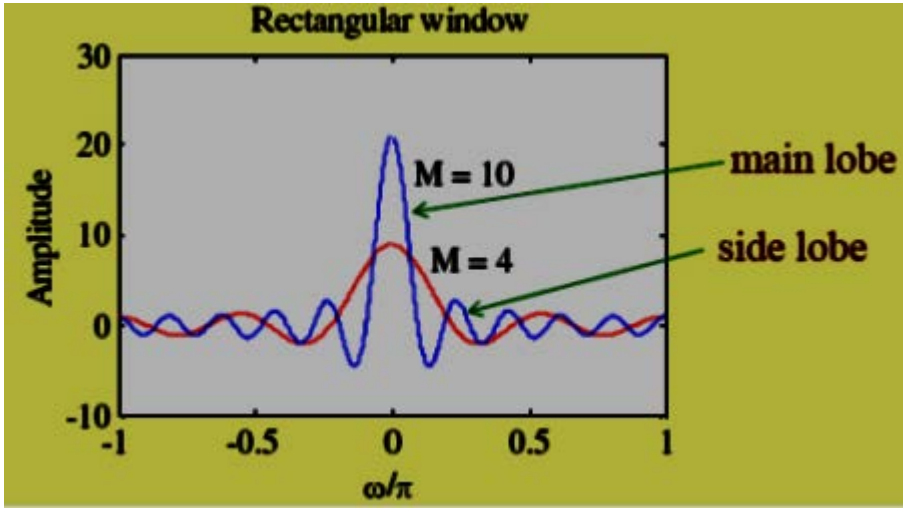
- A **rectangular window** is used to achieve simple truncation:

$$w_R[n] = \begin{cases} 1, 0 \le |n| \le M \\ 0, otherwise \end{cases}$$

- Presence of oscillatory behavior in $H_t(e^{j\omega})$ is basically due to:
  - 1) $h_d[n]$ is infinitely long and may not be absolutely summable, and hence filter is unstable
  - 2) Rectangular window has an abrupt transition

- Oscillatory behavior can be explained by examining the DTFT $\Psi_R(e^{j\omega})$ of $w_R[n]$.

- $\Psi_R(e^{j\omega})$ has a **main lobe** centered at $\omega = 0$. Other ripples are called **sidelobes**



Gibbs Phenomenon

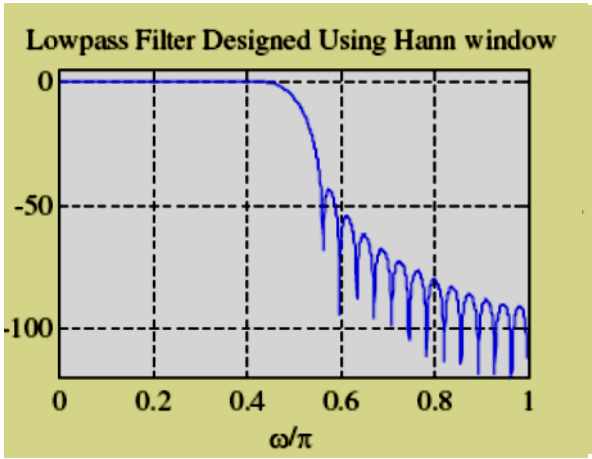
- Main lobe of $\Psi_R(e^{j\omega})$ is characterized by its width $4\pi/(2M+1)$ defined by first zero crossings on both sides of $\omega = 0$.
  - **Result:** As *M* increases, width of main lobe decreases as desired.

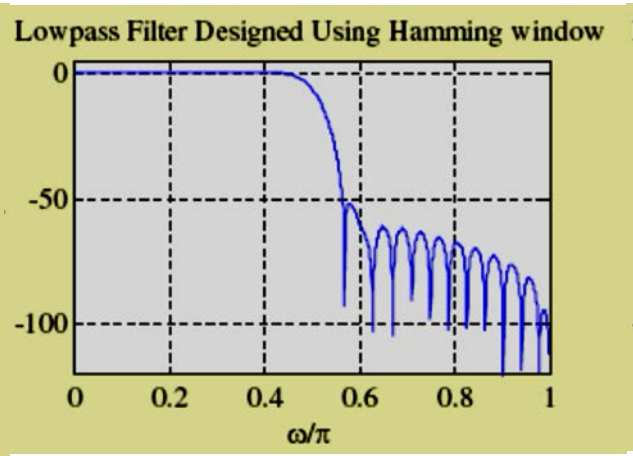
- Area under each lobe remains constant while width of each lobe decreases with an increase in *M.*
  - **Result:** Ripples in around the point of discontinuity occur more closely but with no decrease in amplitude as *M* increases.

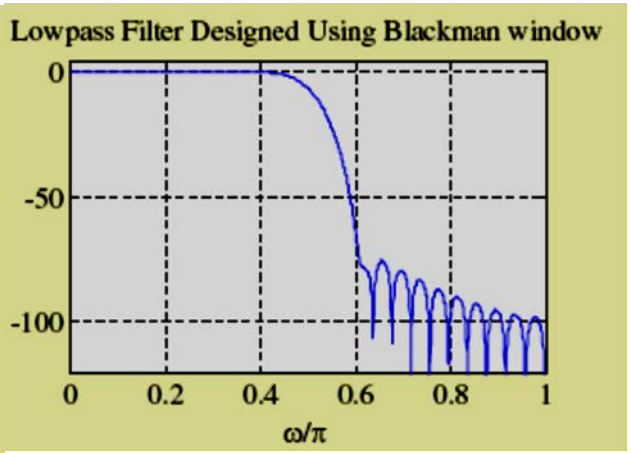
- Rectangular window has an abrupt transition to zero outside the range $-M \leq n \leq M$, which results in Gibbs phenomenon in $H_t(e^{j\omega})$.

Gibbs Phenomenon

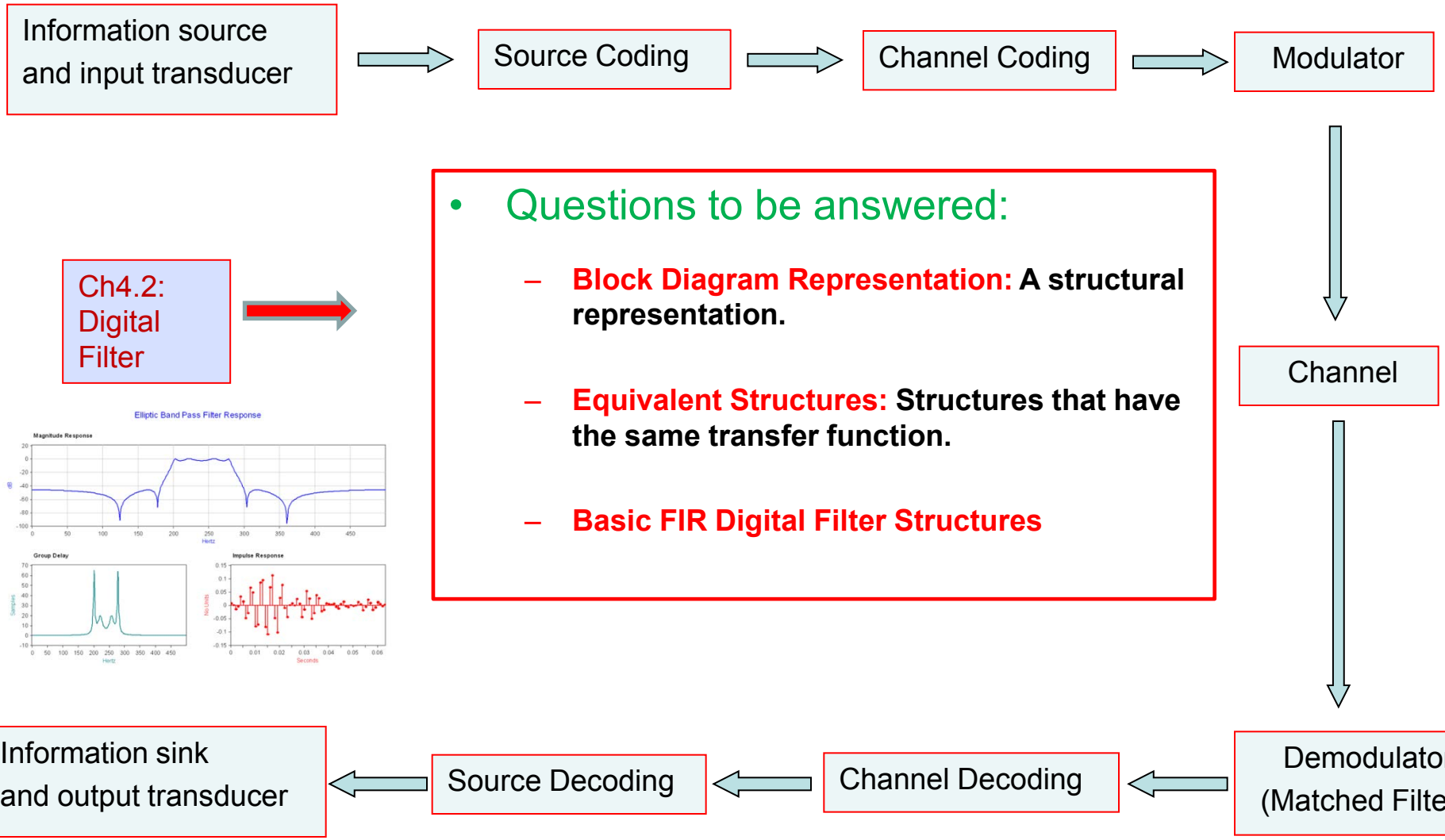
- Gibbs phenomenon can be reduced either:
  - (1) Using a window that tapers smoothly to zero at each end, or
  - (2) Providing a smooth transition from passband to stopband in the magnitude specifications.







Ch4.2: Digital Filter Structure



Ch4.2: Digital Filter

- Questions to be answered:
  - **Block Diagram Representation: A structural representation.**
  - **Equivalent Structures: Structures that have the same transfer function.**
  - **Basic FIR Digital Filter Structures**

Reading: Mitra (chapter 8.1-8.3)

Ch4.2: Digital Filter Structure

- **Block Diagram Representation**
  - Basic Building Blocks
  - Analysis of Block Diagrams
- Equivalent Structures
- Basic FIR Digital Filter Structures

Digital Filter Structure

- The **convolution sum description** of an LTI discrete-time system can, in principle, be used to implement the system.
- For example, an FIR system can be implemented using the convolution sum which is a finite sum of products:

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

- For an IIR finite-dimensional system this approach is not practical as the impulse response is of infinite length. However, a direct implementation of the IIR finite-dimensional system is practical.
- Here, the input-output relation involves a finite sum of products:

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

Digital Filter Structure

- 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.
- Thus, 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.

Block Diagram Representation

- 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.

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

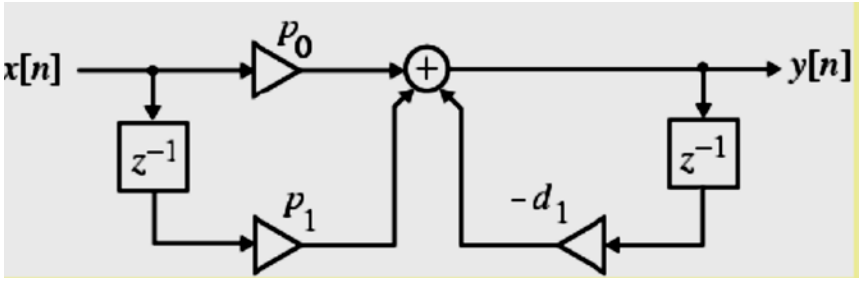
$$y[n] = \sum_{k=0}^{N} h[k]x[n-k]$$

or by the linear constant coefficient difference equation

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

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.



- The filter is described by the difference equation

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

Block Diagram Representation

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

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

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

……

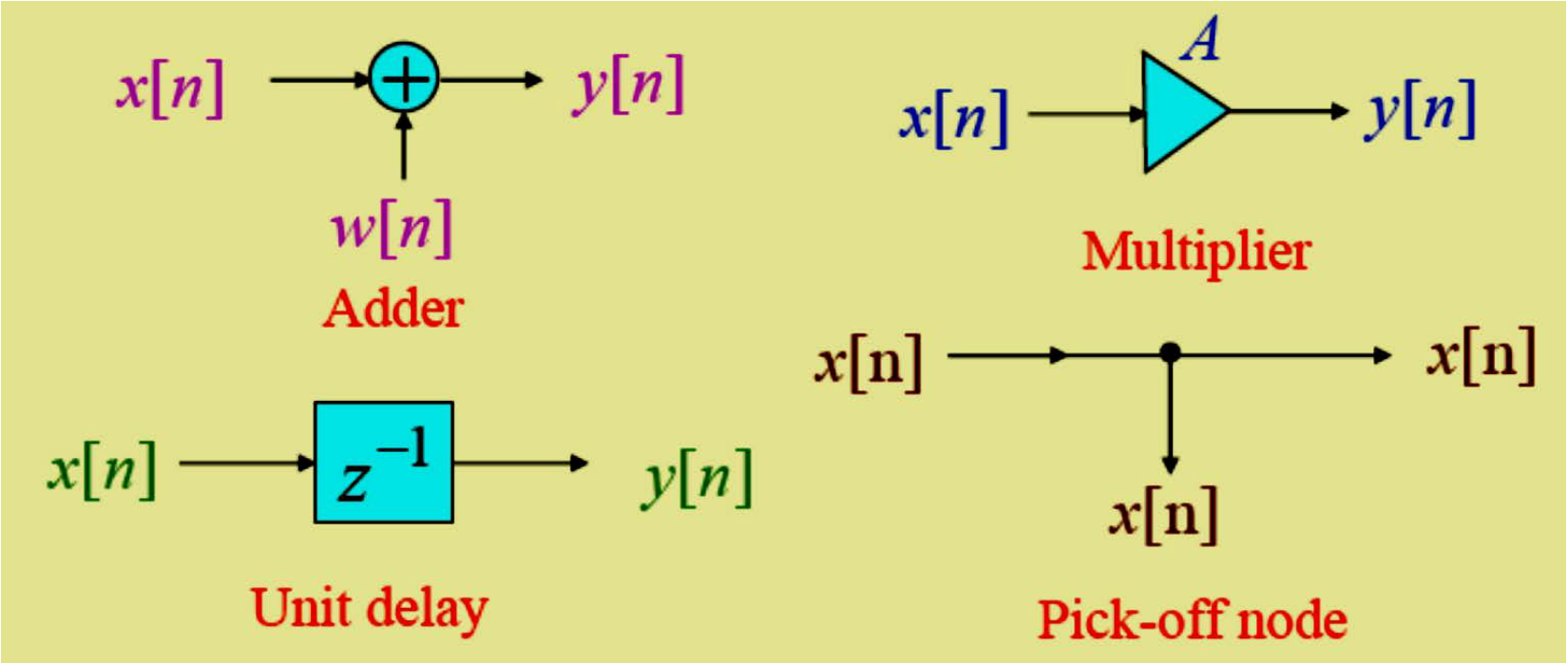
- We can continue this calculation for any n.

- 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:



Advantages of Block Diagram Representation

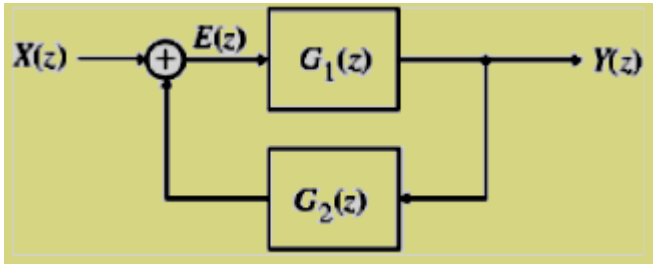
- Advantages of block diagram representation
  - (1) Easy to write down the computational algorithm by inspection
  - (2) Easy to analyze the block diagram to determine the explicit relation between the output and input
  - (3) Easy to manipulate a block diagram to derive other "equivalent" block diagrams yielding different computational algorithms
  - (4) Easy to determine the hardware requirements
  - (5) Easy to develop block diagram representations from the transfer function directly

Analysis of Block Diagrams

- **Question: How to analyze the block diagrams?**
- Analysis is carried out by
  - Writing down the expressions for the output signals of each adder as a sum of its input signals
  - 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 4.2.1

- Consider the single-loop feedback structure shown below

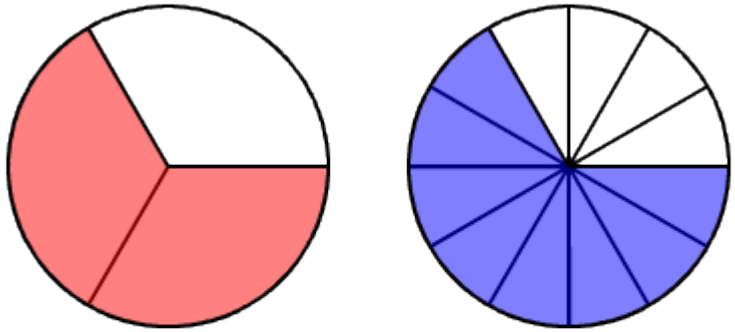


- Writing down the expressions for the output signals of each adder: The output of the adder is $E(z) = X(z) + G_2(z)Y(z)$.
- From the figure, $Y(z) = G_1(z)E(z)$.
- Eliminating $E(z)$ from the previous two equations, we arrive at $[1 - G_1(z)G_2(z)]Y(z) = G_1(z)X(z)$, which leads to

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

Ch4.2: Digital Filter Structure

- Block Diagram Representation
  - Basic Building Blocks
  - Analysis of Block Diagrams
- **Equivalent Structures**
- Basic FIR Digital Filter Structures



$$\frac{2}{3} \times \frac{4}{4} = \frac{8}{12}$$

Equivalent Structures

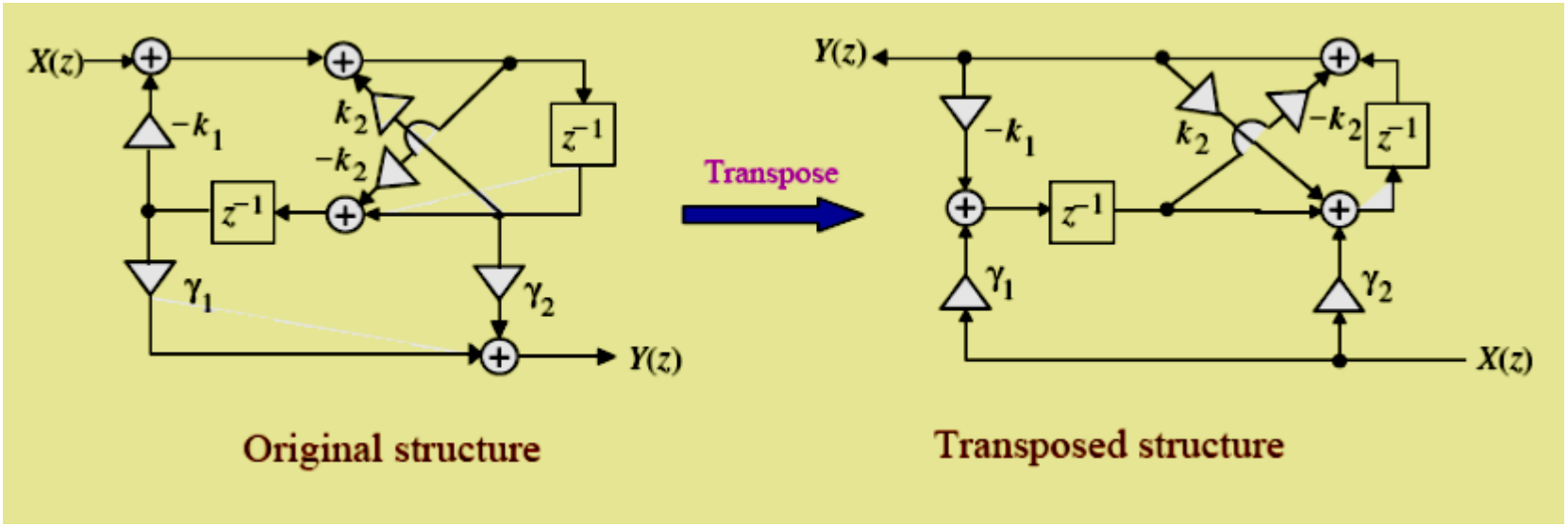
- Two digital filter structures are defined to be **equivalent** if they have the same transfer function.

- A fairly simple way to generate an equivalent structure from a given realization is via the **transpose operation.**

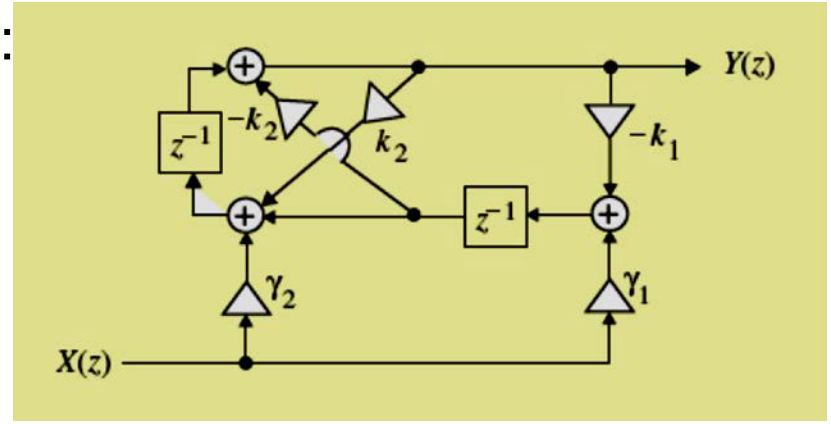
- **Transpose Operation**
  - (1) Reverse all paths
  - (2) Replace pick-off nodes by adders, and vice versa
  - (3) Interchange the input and output nodes

Equivalent Structures: Example 4.2.2

- Transpose operation



- Redrawn transposed structure:



Equivalent Structures

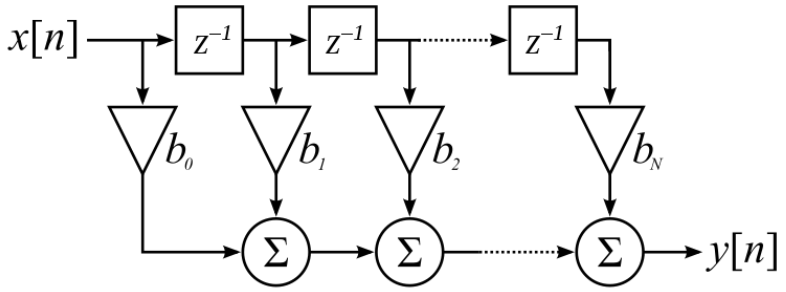
- All other methods for developing equivalent structures are based on a specific algorithm for each structure.
- There are literally an infinite number of equivalent structures realizing the same transfer function.
- Under infinite precision arithmetic any given realization of a digital filter behaves 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.

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 word-length effects in each case, and select the one showing the least effects.
- In certain cases, it is possible to develop a structure that by construction has the least quantization effects.
- Here, we review some simple realizations that in many applications are quite adequate.

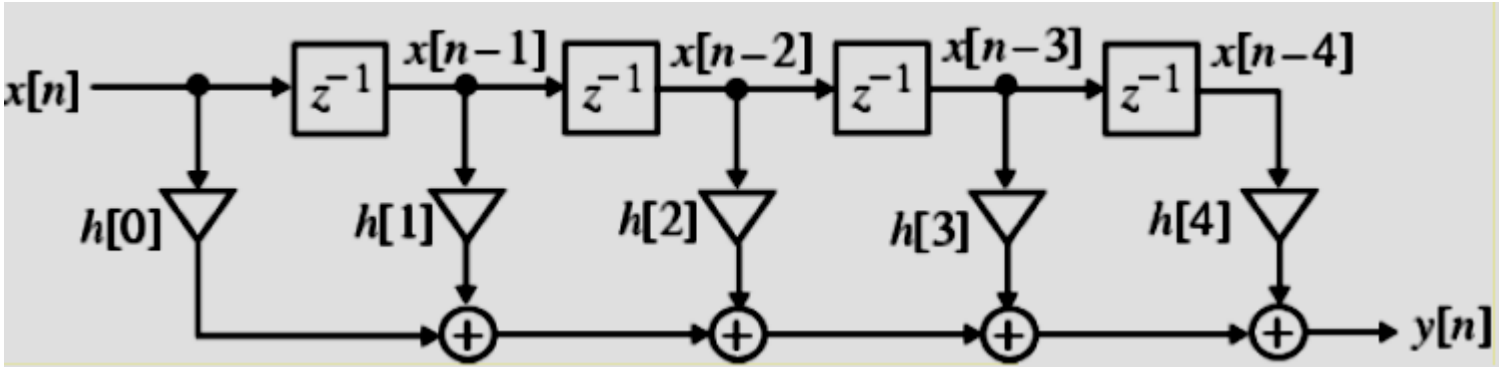
Ch4.2: Digital Filter Structure

- Block Diagram Representation
  - Basic Building Blocks
  - Analysis of Block Diagrams
- Equivalent Structures
- **Basic FIR Digital Filter Structures**



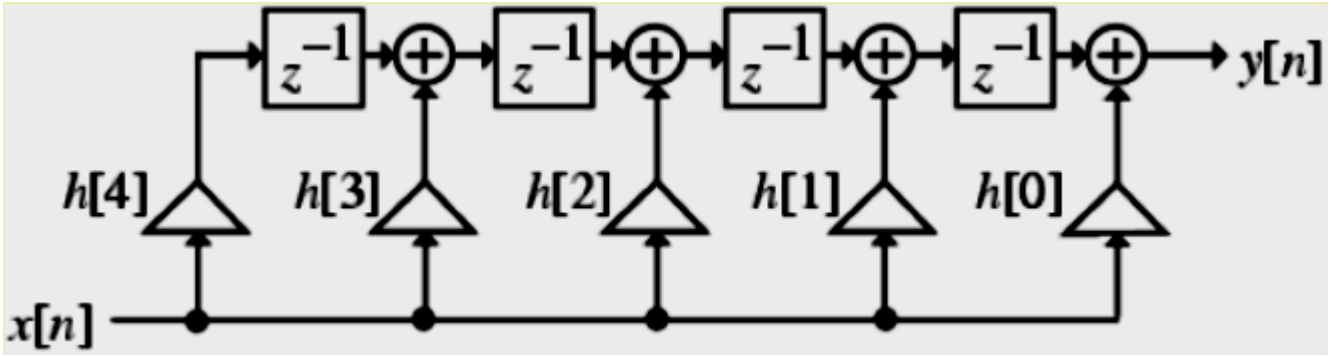
Direct Form FIR Digital Filter Structures

- 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 $y[n] = \sum_{k=0}^{N} h[k]x[n-k]$.
- 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.
- **Example 4.2.3:** Consider an FIR filter with *N*=4.



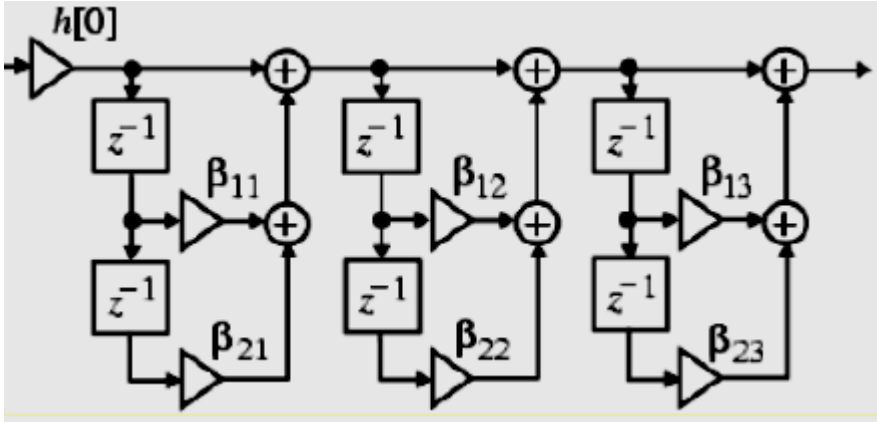
Direct Form FIR Digital Filter Structures

- The direct form structure shown on the previous slide is also known as a **transversal filter.**
- **Transversal filter:** a non-recursive filter using a tapped delay line to implement the basic filter equation.
- The transpose of the direct form structure is indicated below.



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.
- Consider an order-N filter $H(z) = h[0]\prod_{k=1}^{K}(1+\beta_{1k}z^{-1}+\beta_{2k}z^{-2})$ where *K=N/2* if *N* is even and *K=(N+1)/2* if *N* is odd, with $\beta_{2k}=0$.
- 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.

Polyphase FIR Structures

- Consider a causal FIR transfer function $H(z)$ with $N = 8$:

$$H(z) = \prod_{n=0}^{8} h[n]z^{-n}.$$

- $H(z)$ can be expressed as a sum of two terms, with one term containing the even-indexed coefficients and the other containing the odd-indexed coefficients:

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

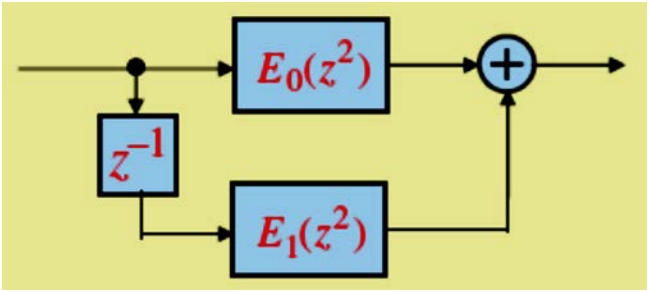
- By using the notation,

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

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

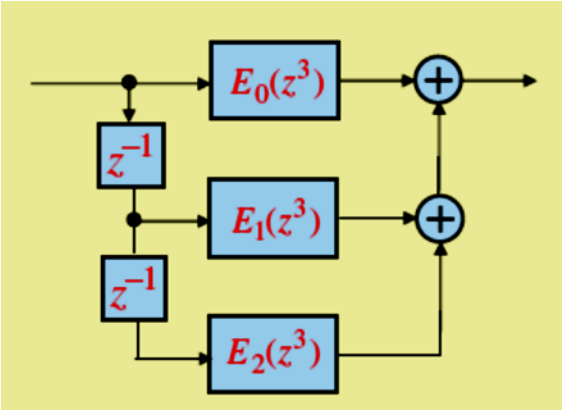
which is commonly known as the

2-branch polyphase decomposition.



Polyphase FIR Structures

- In a similar manner, we can re-express it as
  $H(z) = E_0(z^3) + z^{-1}E_1(z^3) + z^{-2}E_2(z^3)$
  where $E_0 = h[0] + h[3]z^{-1} + h[6]z^{-2}$
  $E_1 = h[1] + h[4]z^{-1} + h[7]z^{-2}$
  $E_0 = h[2] + h[5]z^{-1} + h[8]z^{-2}$



- In general, an *L*-branch polyphase decomposition of an FIR transfer function of order *N* is of the form

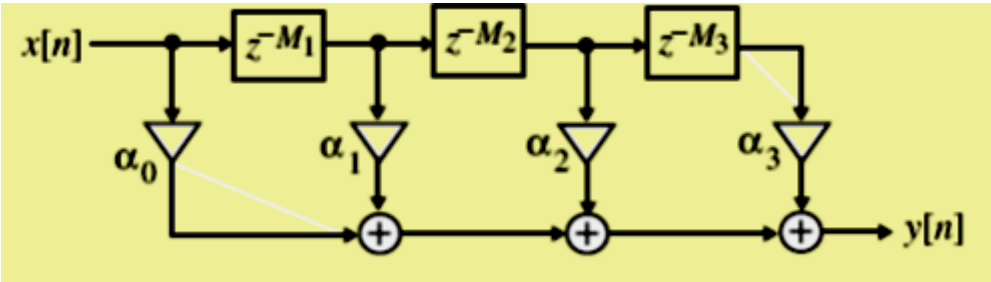
$$H(z) = \sum_{m=0}^{L-1} z^{-m} E_m(z^L)$$

where $E_m(z^L) = \sum_{n=0}^{\lfloor (N+1)/L \rfloor} h[Ln+m]\, z^{-m}$ with $h[n] = 0$ for $n > N$.

- The subfilters $E_m(z^L)$ are also FIR filters and can be realized using any methods described so far.

Tapped Delay Line

- In some applications, such as **wireless communications**, musical, and sound processing, FIR filter structures of the form shown below are employed



- The structure consists of a chain of $M_1, M_2,$ and M_3 unit delays with taps at the input, at the end of first M_1 delays, at the end of next M_2 delays, and at the output.
- Signals at these taps are then multiplied by constants α_0, α_1, α_2, and α_3, and added to form the output.
- Such a structure is usually referred to as the **tapped delay line**.
- The direct form FIR structure in Example 4.2.3 is seen to be a special case of the tapped delay line, where there is a tap after each unit delay.