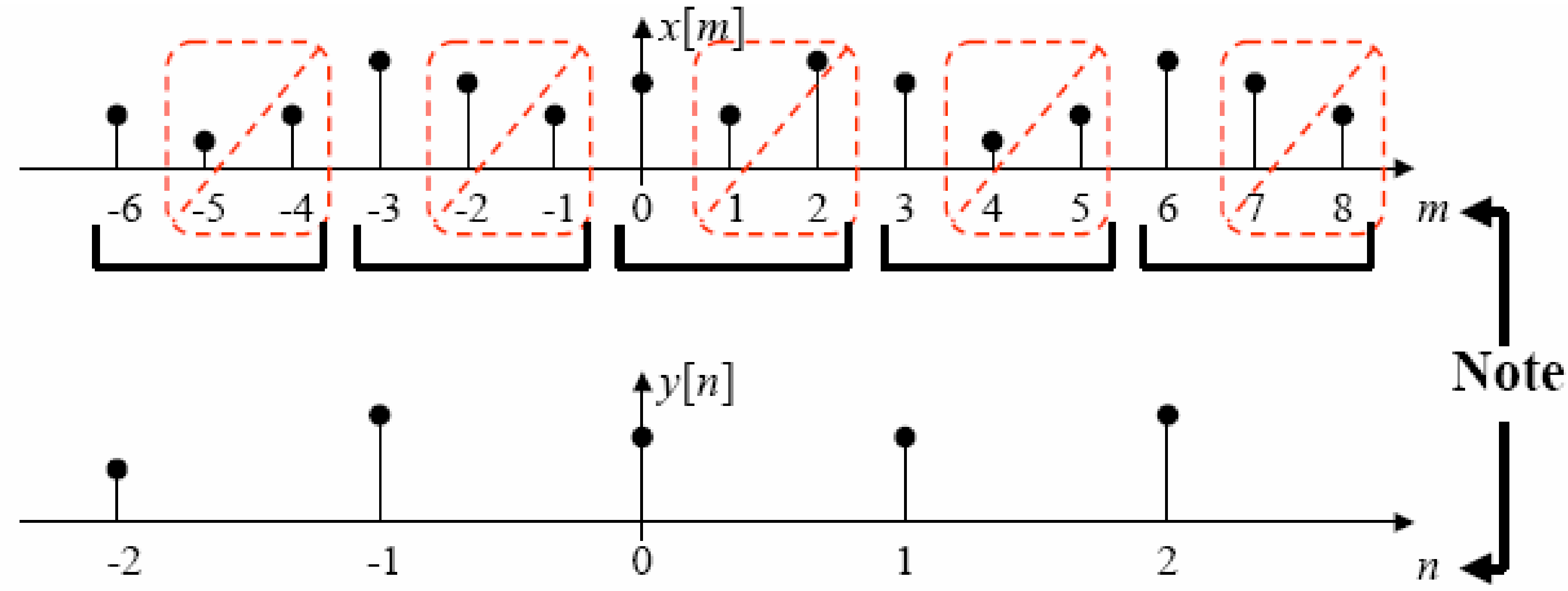
Decimation by M



Time domain representation

Decimator by M

- Z-transform analysis of decimator (continued)

$$U(z) \quad \rightarrow \boxed{\downarrow \mathbf{M}} \rightarrow \quad \frac{1}{M}\sum_{i=0}^{M-1} U(z^{1/M} e^{-j2\pi i/M})$$

  – Note that U(e$^{j\omega}$) is periodic with period 2π , while U(e$^{j\omega/M}$) is periodic with period 2Mπ
  – the summation with i=0…M-1 restores the periodicity with period 2π

- Example:

$$u[k] = \alpha^k, k \geq 0$$

$$\Rightarrow U(z) = \frac{1}{1-\alpha z^{-1}}$$

$$\Rightarrow Y(z) = \frac{1}{M}\sum_{i=0}^{N-1}(\ldots) = \ldots = \frac{1}{1-\alpha^M z^{-1}}$$

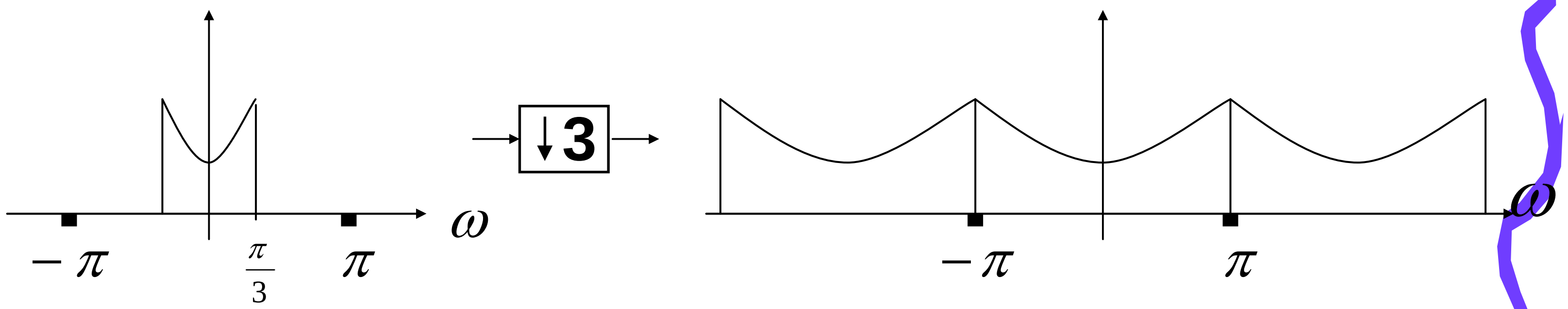
$$\Rightarrow y[k] = (\alpha^M)^k, k \geq 0$$

Decimation by M

- Z-transform (frequency domain) analysis of decimator

u[0],u[1],u[2]... → ↓M → u[0], u[M], u[2M]...

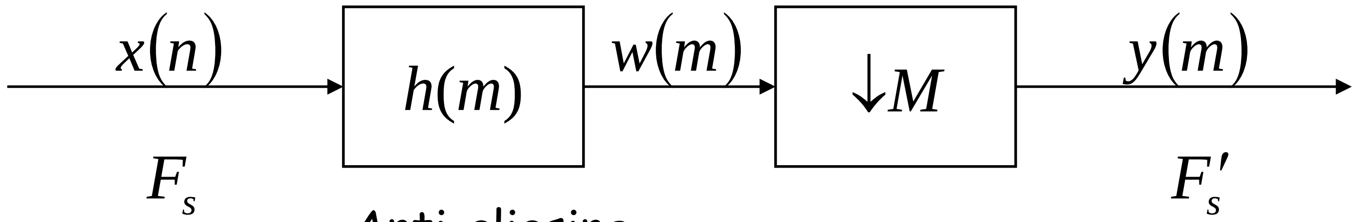
$U(z)$ → ↓M → $\frac{1}{M}\sum_{i=0}^{M-1} U(z^{1/M} e^{-j2\pi i/M})$



decimation introduces ALIASING if input signal occupies frequency band larger than π / M, for $-\pi \leq \omega \leq \pi$

hence decimation mostly preceded by anti-aliasing (decimation) filter

Decimation by M

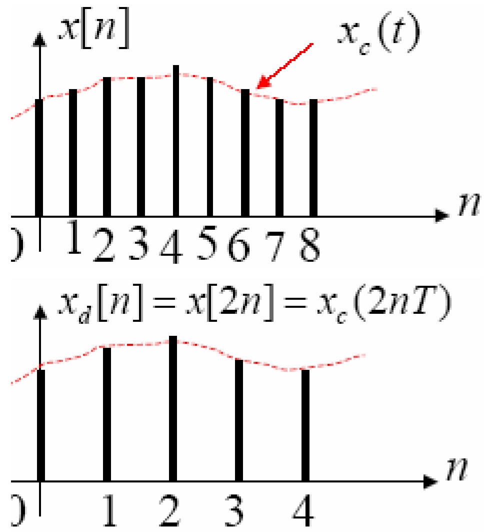


Anti-aliasing filter

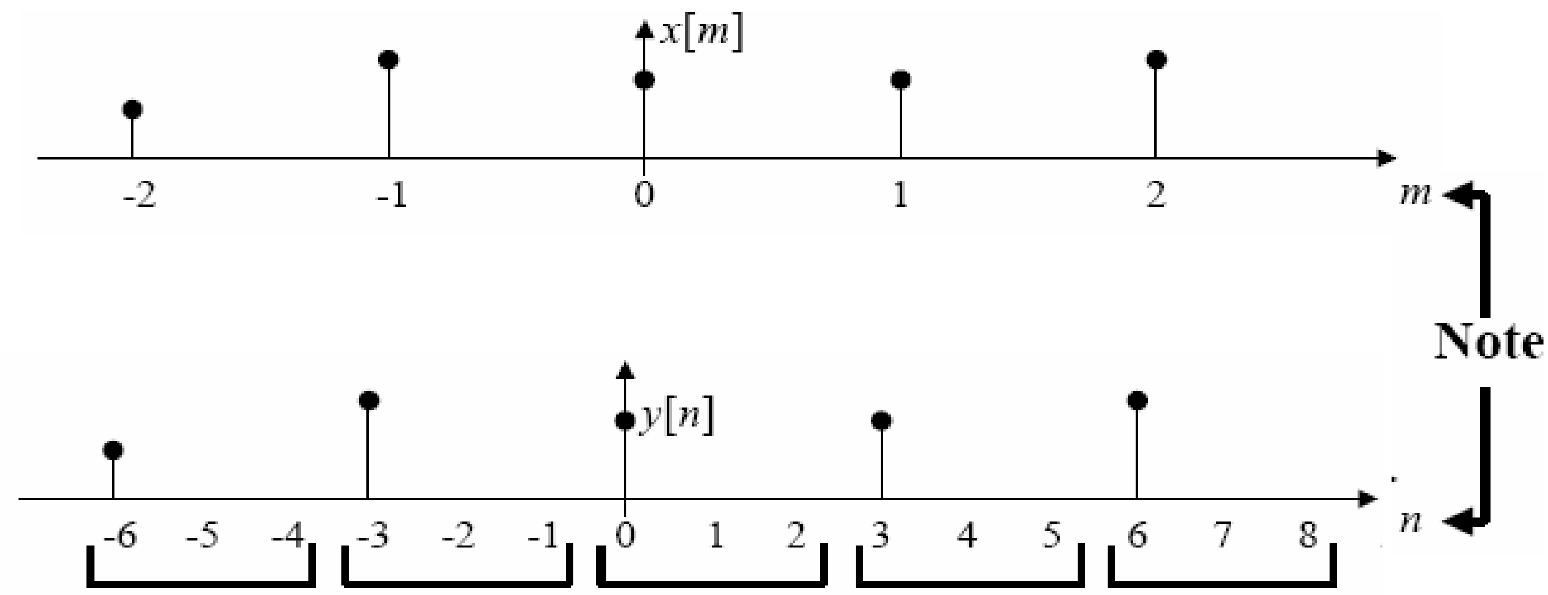
$$F'_s = F_s / M$$

$$\frac{T'}{T} = M = \frac{F_s}{F'_s}$$

$$H_I(\omega) = \begin{cases} 1, & |\omega| < \dfrac{\pi}{M} \\ 0, & \text{otherwise} \end{cases}$$



Interpolation by L



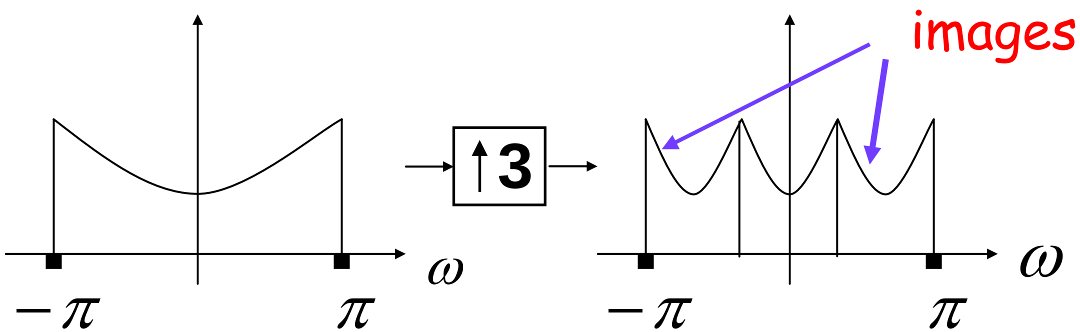
Time domain representation

Interpolation by L

- Z-transform (frequency domain) analysis of expander

u[0], u[1], u[2],... → ↑L → u[0],0,..0,u[1],0,…,0,u[2]...

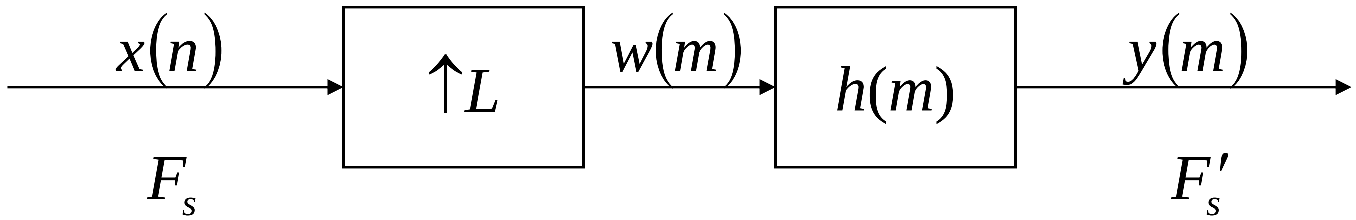
$$U(z) \rightarrow \boxed{\uparrow L} \rightarrow U(z^N)$$



`expansion in time domain ~ compression in frequency domain'

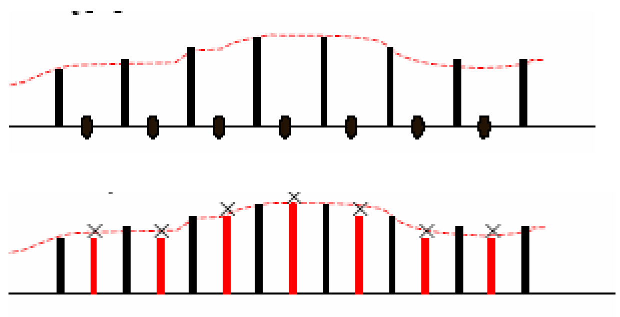
expander mostly followed by interpolation filter to remove all images

Interpolation by L



$$F_s' = L F_s$$

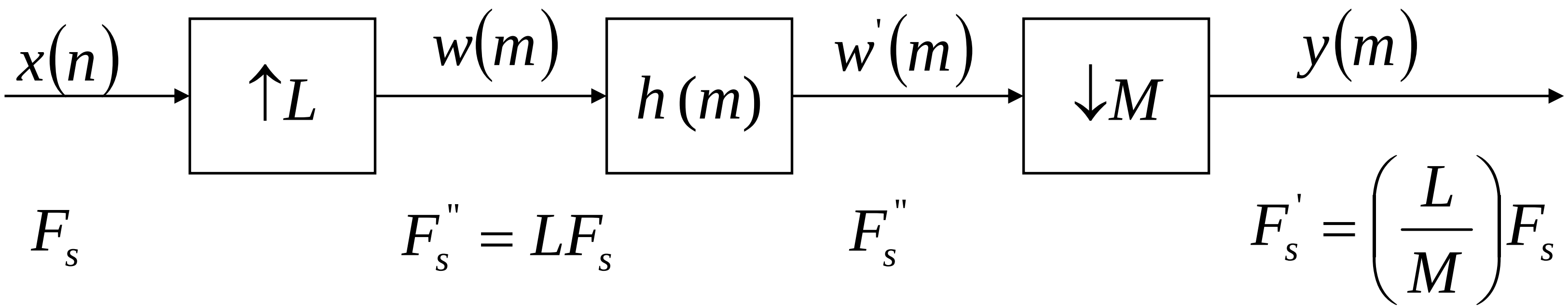
$$\frac{T'}{T} = \frac{1}{L} = \frac{F_s}{F_s'}$$



$$H_I(\omega') = \begin{cases} G, & |\omega'| < \dfrac{\pi}{L} \\ 0, & \text{otherwise} \end{cases}$$

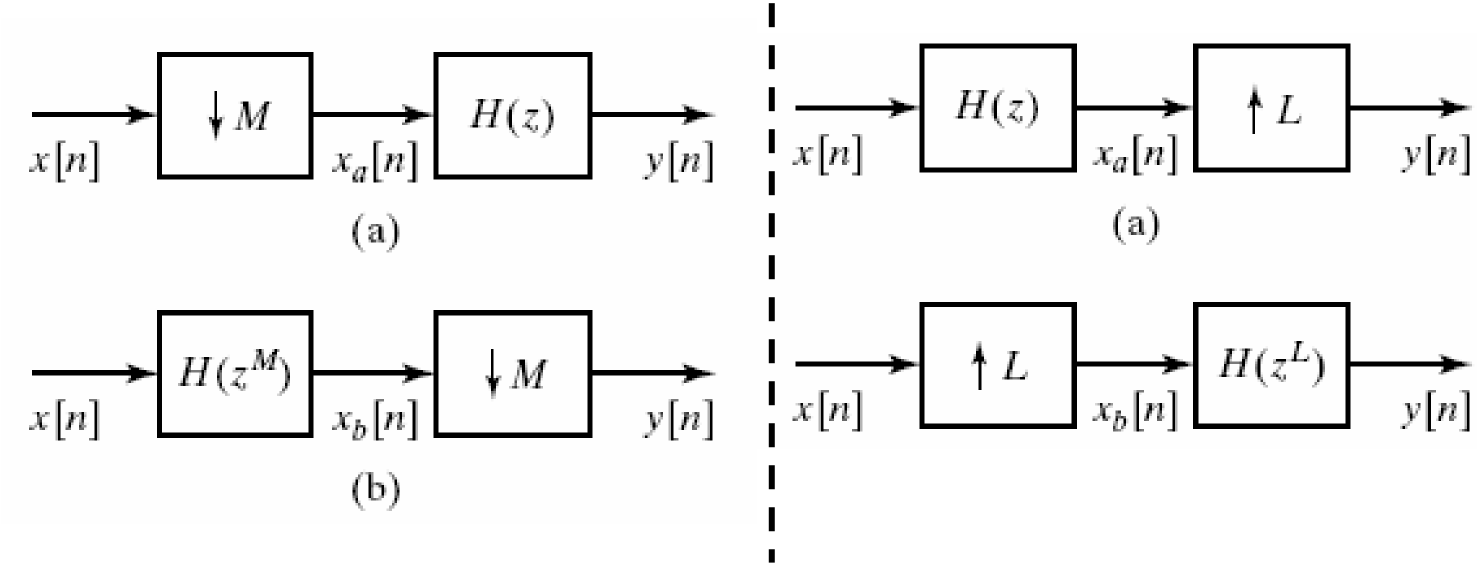
Conversion by a Rational Factor M/L

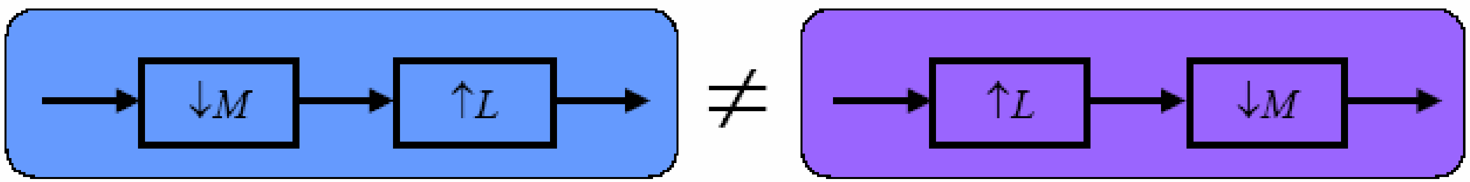
- A more efficiency implementation



$$H_I(\omega'') = \begin{cases} L, & |\omega''| \le \min\left|\frac{\pi}{L}, \frac{\pi}{M}\right| \\ 0, & \text{otherwise} \end{cases}$$

Interchange of Filtering





Proof

$$\text{(b)} \Longrightarrow X_b(e^{j\omega}) = H(e^{jwM})X(e^{j\omega}). \text{ Recall that } Y(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X_b(e^{j(\omega/M-2\pi i/M)})$$

$$\text{Then } Y(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} H(e^{j(\omega-2\pi i)})X(e^{j(\omega/M-2\pi i/M)})$$

$$= \frac{1}{M}\sum_{i=0}^{M-1} H(e^{j\omega})X(e^{j(\omega/M-2\pi i/M)})$$

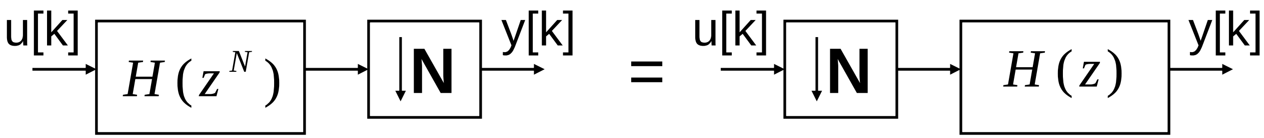
$$= H(e^{j\omega})X_a(e^{j\omega})$$

$$\text{Similarly, (a)} \Longrightarrow Y(e^{j\omega}) = X_a(e^{j\omega L}) = X(e^{j\omega L})H(e^{j\omega L}) = H(e^{j\omega L})X_b(e^{j\omega})$$

$$\text{Since } X_b(e^{j\omega}) = X(e^{j\omega L})$$

Identities (I)

- Nobel identity (I) : (only for rational functions)



Example : N=2

h[0],h[1],0,0,0,…

$$\begin{bmatrix} y[0] \\ y[1] \\ y[2] \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{2-fold downsampling}} \cdot \overbrace{\begin{bmatrix} h[0] & 0 & 0 & 0 \\ 0 & h[0] & 0 & 0 \\ h[1] & 0 & h[0] & 0 \\ 0 & h[1] & 0 & h[0] \\ 0 & 0 & h[1] & 0 \\ 0 & 0 & 0 & h[1] \end{bmatrix}}^{H(z^2)} \cdot \begin{bmatrix} u[0] \\ u[1] \\ u[2] \\ u[3] \end{bmatrix} = \ldots = \underbrace{\begin{bmatrix} h[0] & 0 \\ h[1] & h[0] \\ 0 & h[1] \end{bmatrix}}_{H(z)} \cdot \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{2-fold downsampling}} \cdot \begin{bmatrix} u[0] \\ u[1] \\ u[2] \\ u[3] \end{bmatrix}$$

Identities (II)

- Noble identitiy (II) : (only for rational functions)

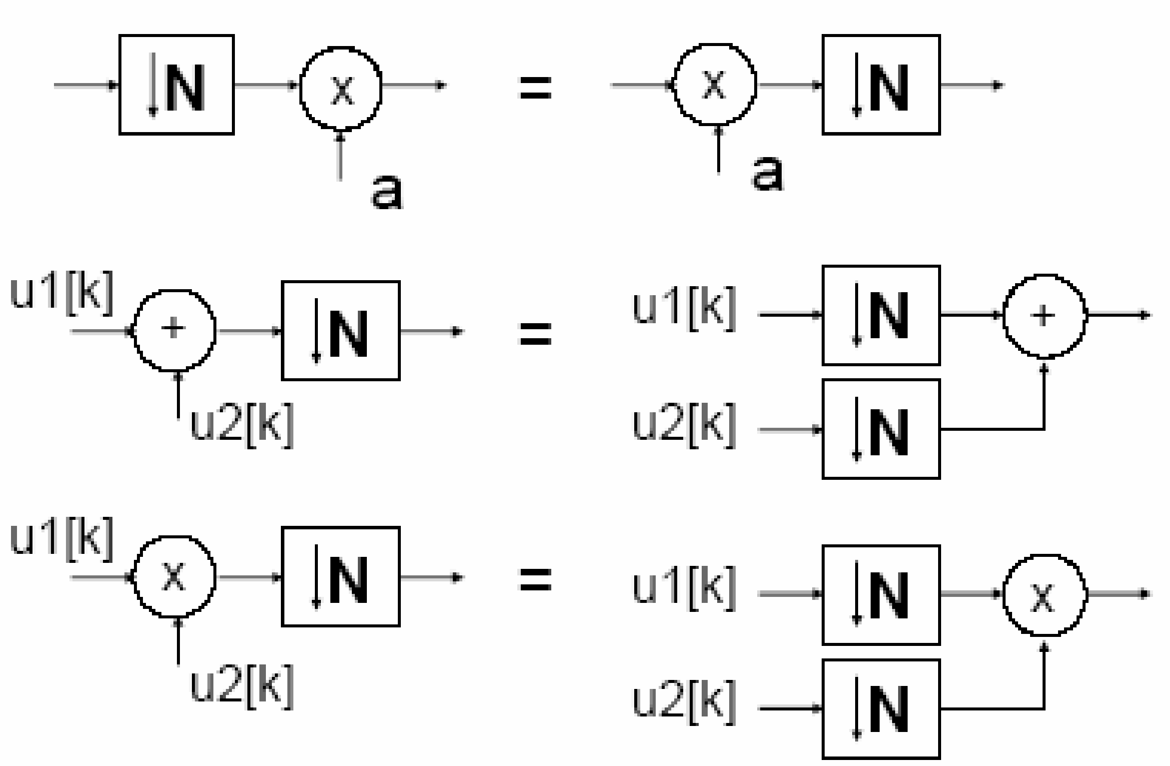
$$u[k] \rightarrow \boxed{H(z)} \rightarrow \boxed{\uparrow N} \rightarrow y[k] \quad = \quad u[k] \rightarrow \boxed{\uparrow N} \rightarrow \boxed{H(z^N)} \rightarrow y[k]$$

Example : N=2

h[0],h[1],0,0,0,…

$$\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ y[3] \\ y[4] \\ y[5] \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{\text{2-fold upsampling}} \cdot \underbrace{\begin{bmatrix} h[0] & 0 \\ h[1] & h[0] \\ 0 & h[1] \end{bmatrix}}_{H(z)} \cdot \begin{bmatrix} u[0] \\ u[1] \end{bmatrix} = \ldots = \underbrace{\begin{bmatrix} h[0] & 0 & 0 & 0 \\ 0 & h[0] & 0 & 0 \\ h[1] & 0 & h[0] & 0 \\ 0 & h[1] & 0 & h[0] \\ 0 & 0 & h[1] & 0 \\ 0 & 0 & 0 & h[1] \end{bmatrix}}_{H(z^2)} \cdot \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\text{2-fold upsampling}} \cdot \begin{bmatrix} u[0] \\ u[1] \end{bmatrix}$$

Interconnection of Multi-rate Building Blocks



Identities also hold if all decimators are replaced by expanders

Identities (III)

$$\rightarrow \boxed{\uparrow L} \rightarrow \boxed{\downarrow N} \rightarrow \quad = \quad \rightarrow \boxed{\downarrow N} \rightarrow \boxed{\uparrow L} \rightarrow$$

if and only if L and N are ***coprime*** !!!!!

Example 1: u[k]=1,2,3,4,5,6,7,8,9,… (L=2,N=3)

a) 2-fold up: 1,0,2,0,3,0,4,0,… | a) 3-fold down: 1,4,7,...

b) 3-fold down:1,0,4,0,7,0,… | b) 2-fold up: 1,0,4,0,7,...

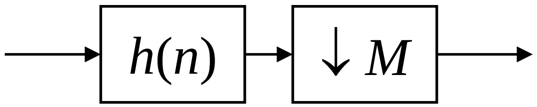
Example 2: u[k]=1,2,3,4,5,6,7,8,9,… (L=2,N=4)

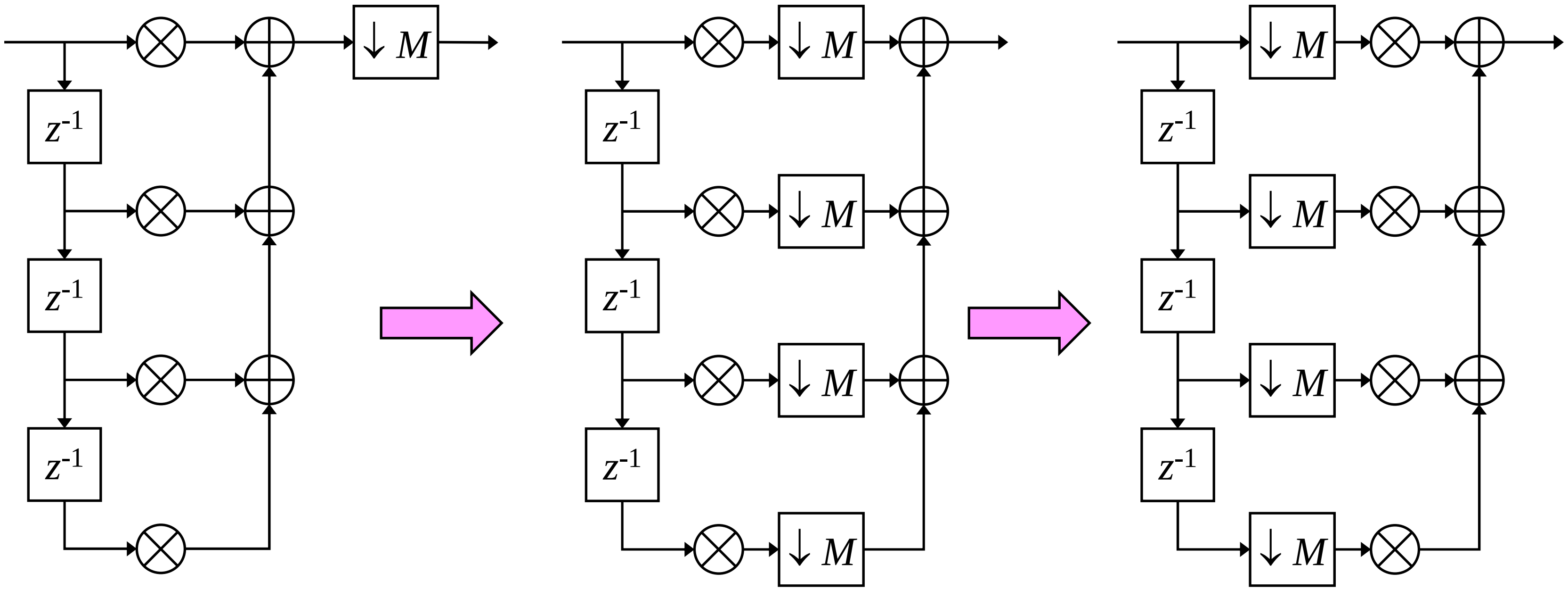
a) 2-fold up: 1,0,2,0,3,0,4,0,… | a) 4-fold down: 1,5,9,...

b) 4-fold down:1,3,5,7,9,… | b) 2-fold up: 1,0,5,0,9,...

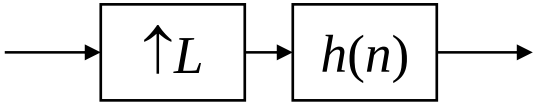
Practical Structure

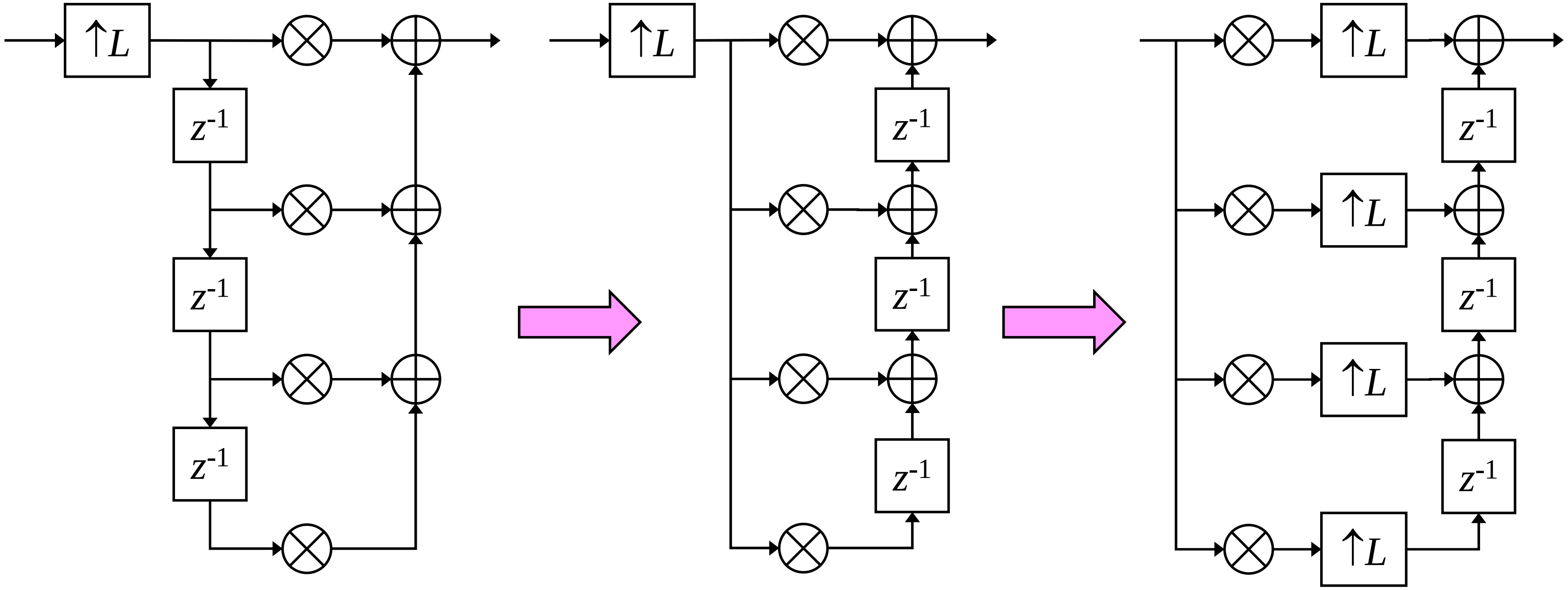
- Decimation





Practical Structure

- Interpolation 



An Efficient Decomposition

- Example: 2-fold decomposition

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

$$= \underbrace{(h[0] + h[2]z^{-2} + h[4]z^{-4} + h[6]z^{-6})}_{E_0(z^2)} + z^{-1}\underbrace{(h[1] + h[3]z^{-2} + h[5]z^{-4})}_{E_1(z^2)}$$

- Example 3-fold decomposition

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

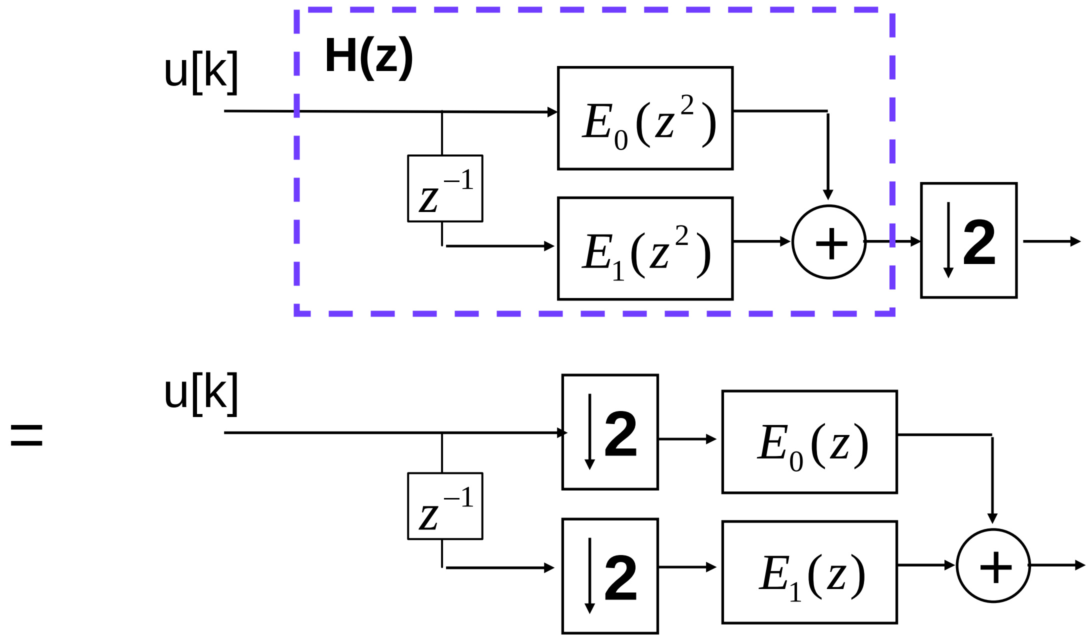
$$= \underbrace{(h[0] + h[3]z^{-3} + h[6]z^{-6})}_{E_0(z^3)} + z^{-1}\underbrace{(h[1] + h[4]z^{-3})}_{E_1(z^3)} + z^{-2}\underbrace{(h[2] + h[5]z^{-3})}_{E_2(z^3)}$$

- General case (N-fold decomposition)

$$H(z) = \sum_{k=-\infty}^{\infty} h[k]z^{-k} = \sum_{l=0}^{N-1} z^{-l} E_l(z^N), \text{ where } E_l(z) = \sum_{k=-\infty}^{\infty} h[Nk + l]z^{-k}$$

Example of Efficient Decomposition

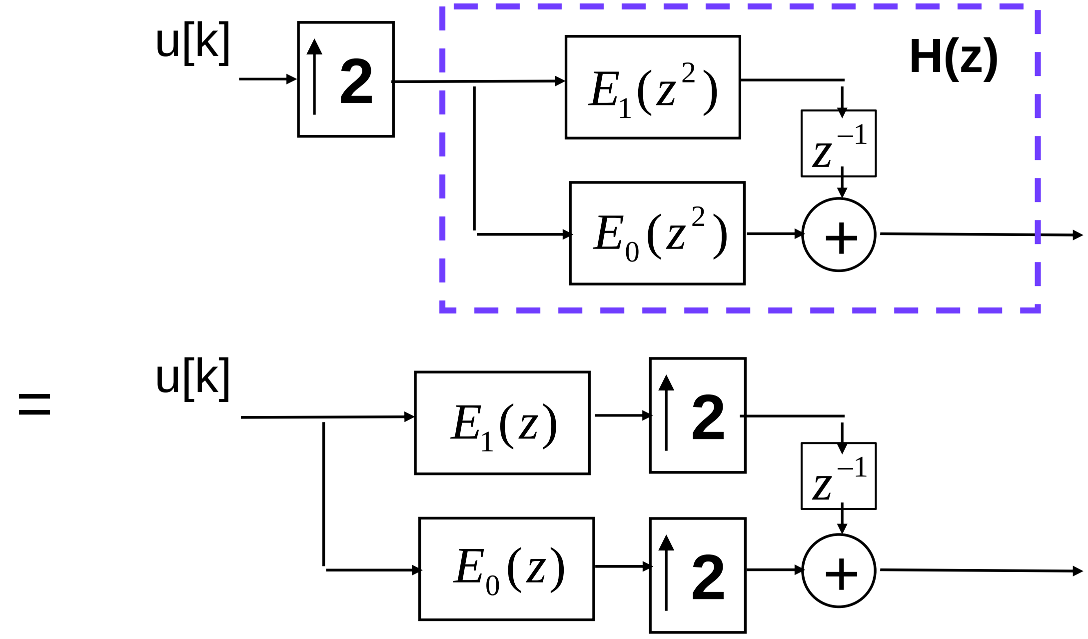
- Efficient implementation of a decimation filter



i.e. all filter operations performed at the lowest rate

Example of Efficient Decomposition

- Efficient implementation of an interpolation filter

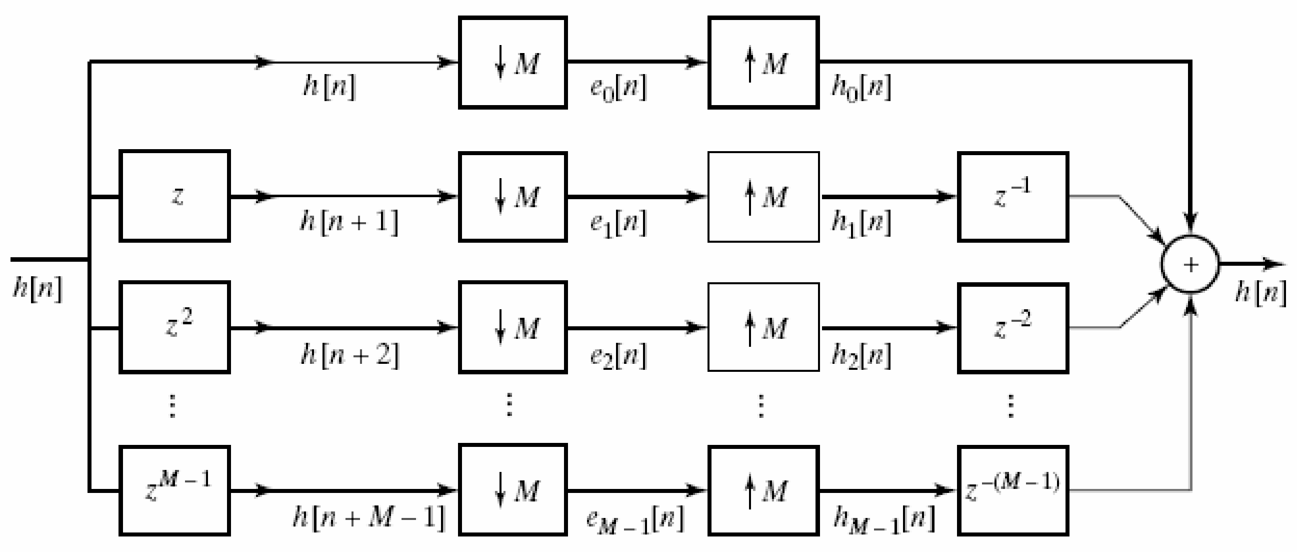


i.e. all filter operations performed at the lowest rate

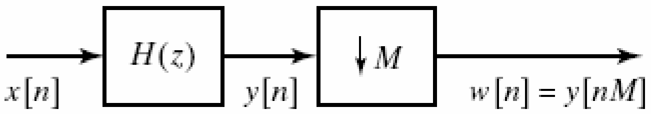
Polyphase Decomposition

- An important advancement in multirate signal processing
  - The polyphase decomposition of a sequence is obtained by representing it as a sequence of M subsequence, each consisting of every Mth value of successively delayed versions of the sequence.
  - Example:

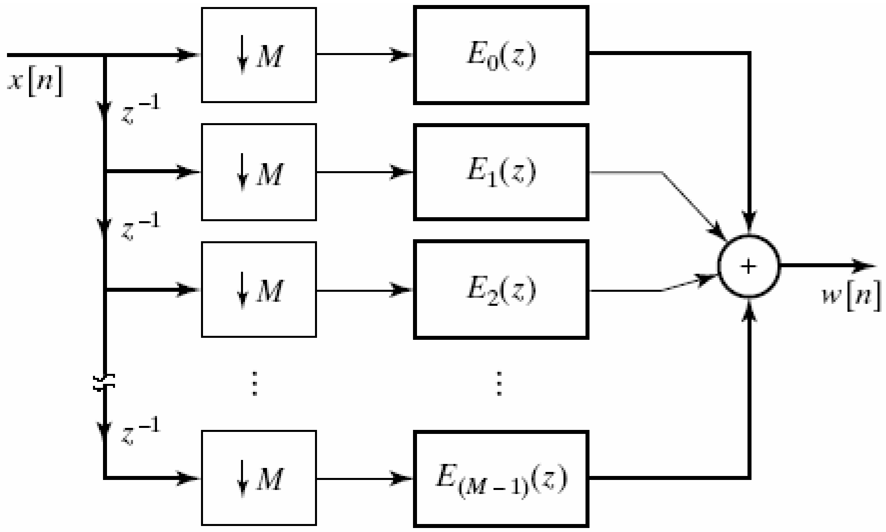
$$h[n]=\sum_{k=0}^{M-1} h_k[n-k], \quad \text{with } h_k[n]=\begin{cases} h[n+k], & n=\text{integer multiple of } M \\ 0, & \text{O.W.} \end{cases}$$



Computation Complexity



N-point FIR



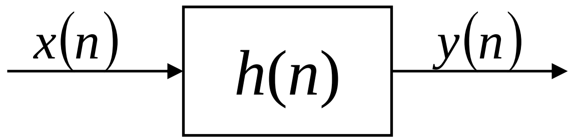
N/M- point FIR

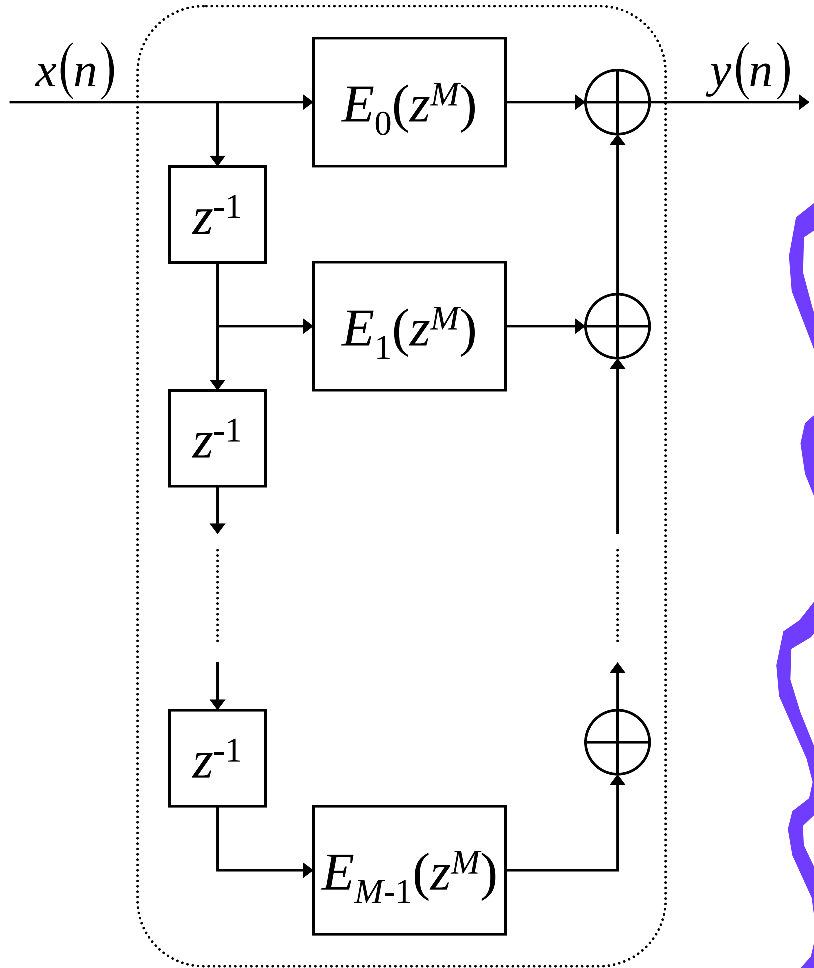
1. N multipliers and N-1 adder
2. One sample in per 1 u.t.
3. One sample out per 1. u.t., but only one of every M outputs is required
4. Computation complexity: N multiplications, N-1 additions

1. For each filter $E_i(z)$, one input is clocked at a rate of 1/M u.t.
2. For each filter Ei(z), it requires (1/M)(N/M) multiplications and (1/M)(N/M-1) additions
3. Computation complexity: N/M multiplications (N/M-1)+M-1 additions

Application: Polyphase FIR Filter

- Polyphase decomposition

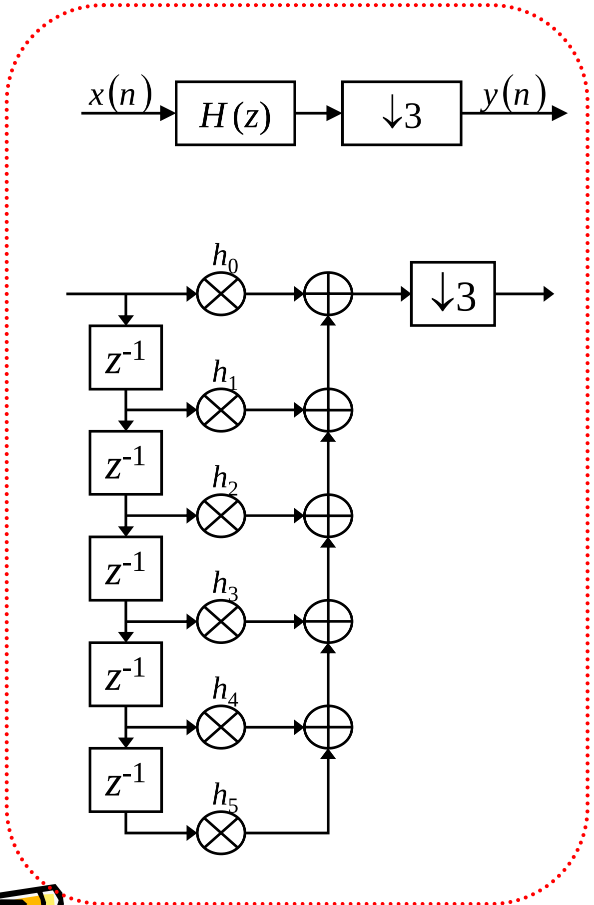


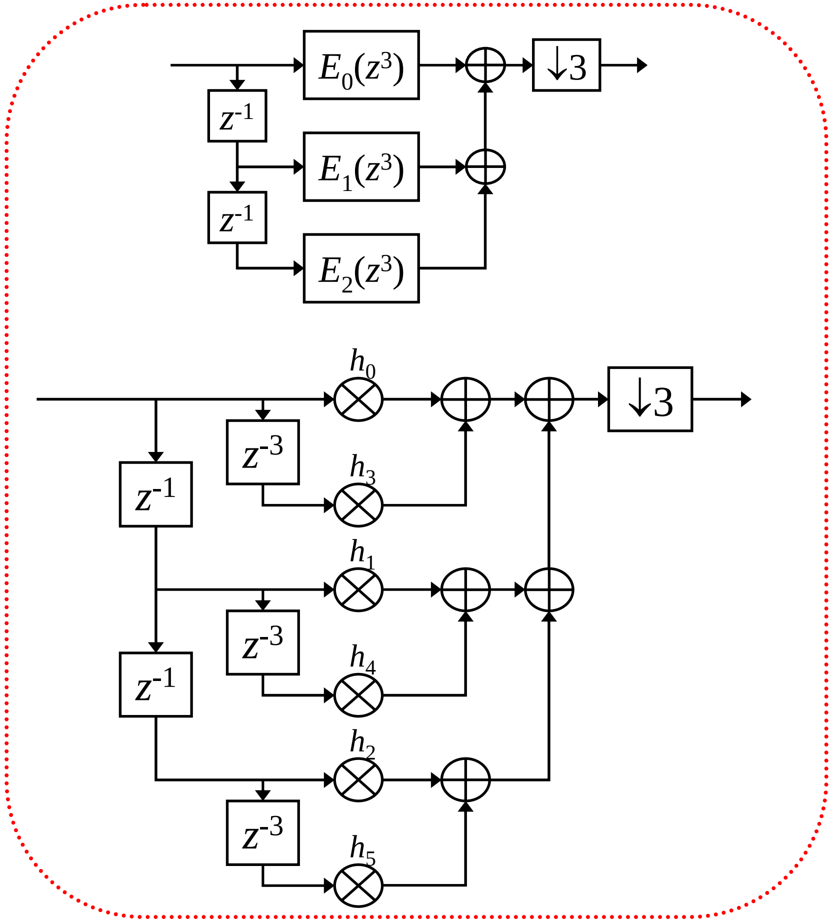


$$H(z)=\sum_{n=-\infty}^{\infty} h(n) z^{-n}$$

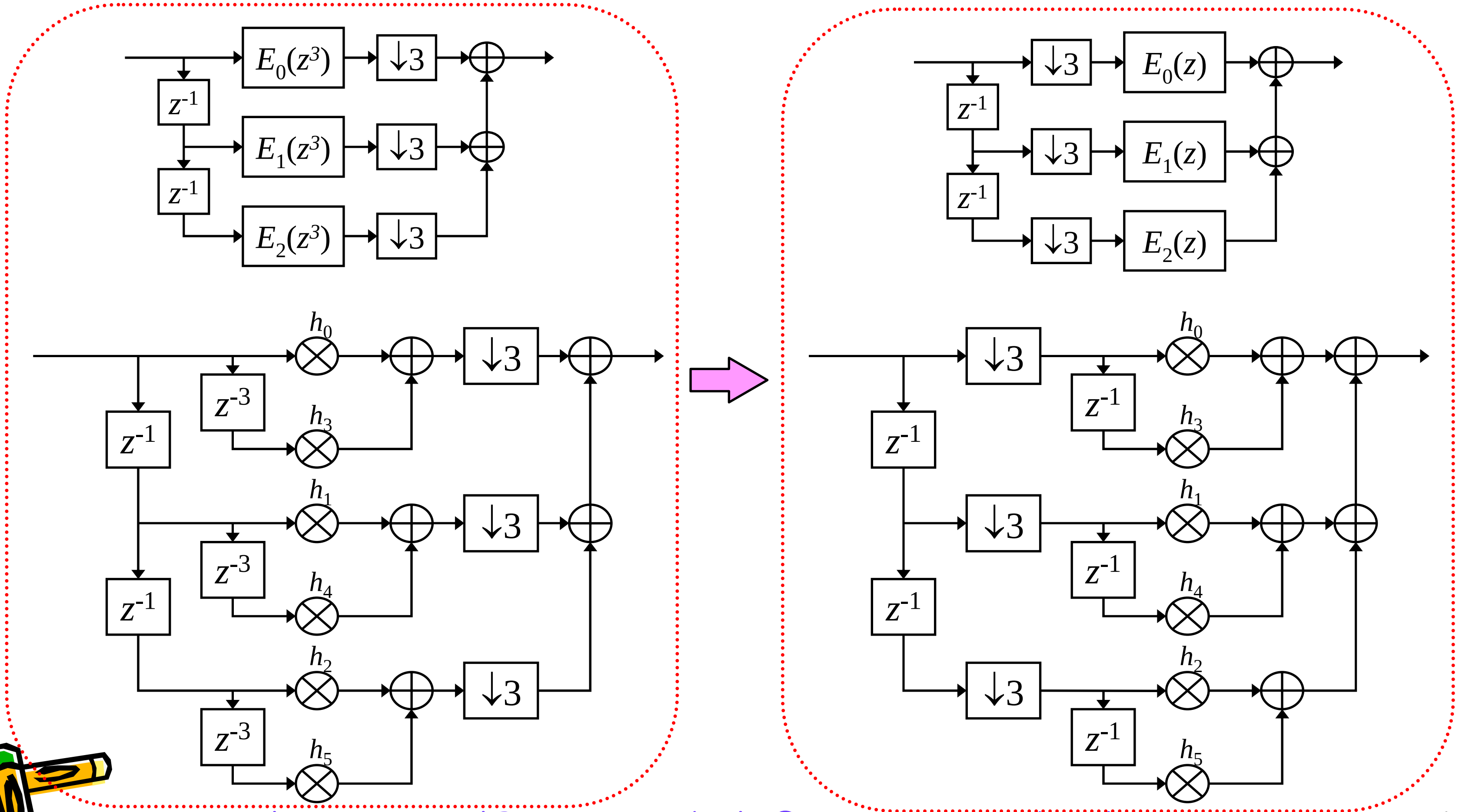
$$\begin{cases} H(z)=\sum_{l=0}^{M-1} z^{-l} E_l\left(z^M\right) \\ E_l(z)=\sum_{n=-\infty}^{\infty} e_l(n) z^{-n} \\ e_l(n) \equiv h(Mn+l) \end{cases}$$

Polyphase FIR Filter

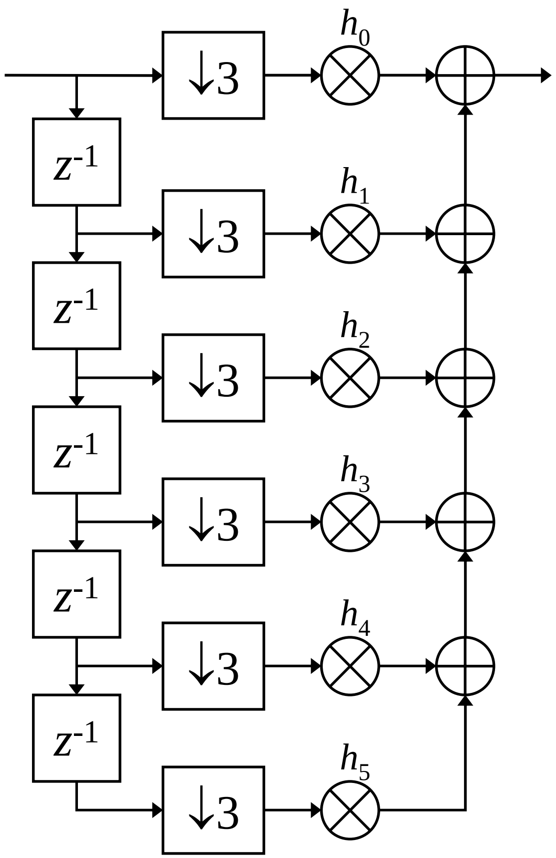




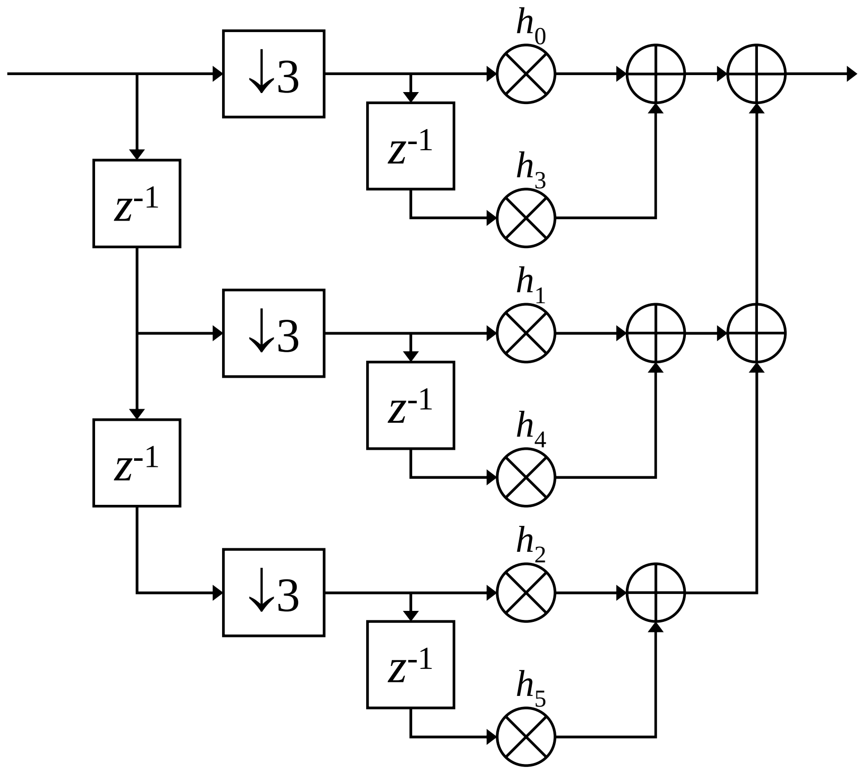
Polyphase FIR Filter



Structure Comparison



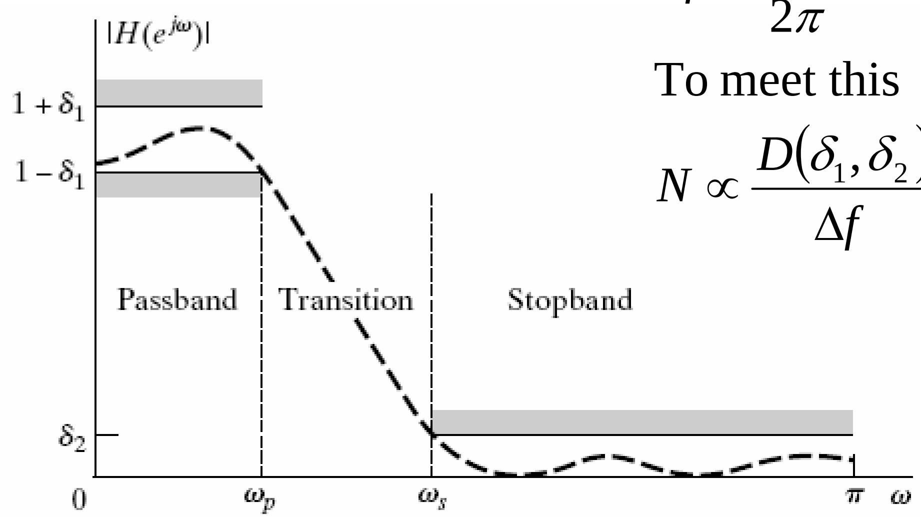
Direct implementation



Polyphase implementation

FIR LPF Design Issue

- FIR lowpass filter



$$\Delta f = \frac{\Delta\omega}{2\pi} = \frac{\omega_s - \omega_p}{2\pi} = \text{normalized transition BW}$$

To meet this spec., the FIR filter has order

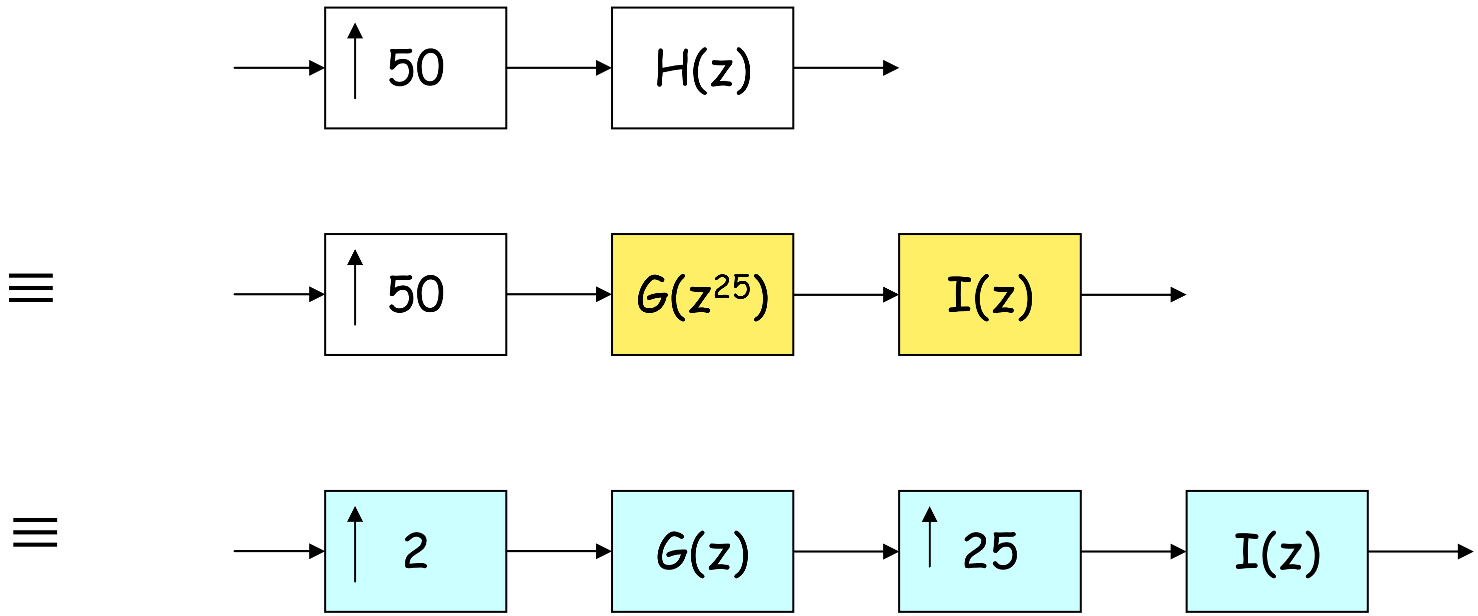
$$N \propto \frac{D(\delta_1, \delta_2)}{\Delta f}$$

\# of multiplications $\propto$N
\# of additions $\propto$N

- Large N requires the LPF to have a small stopband edge of π /N

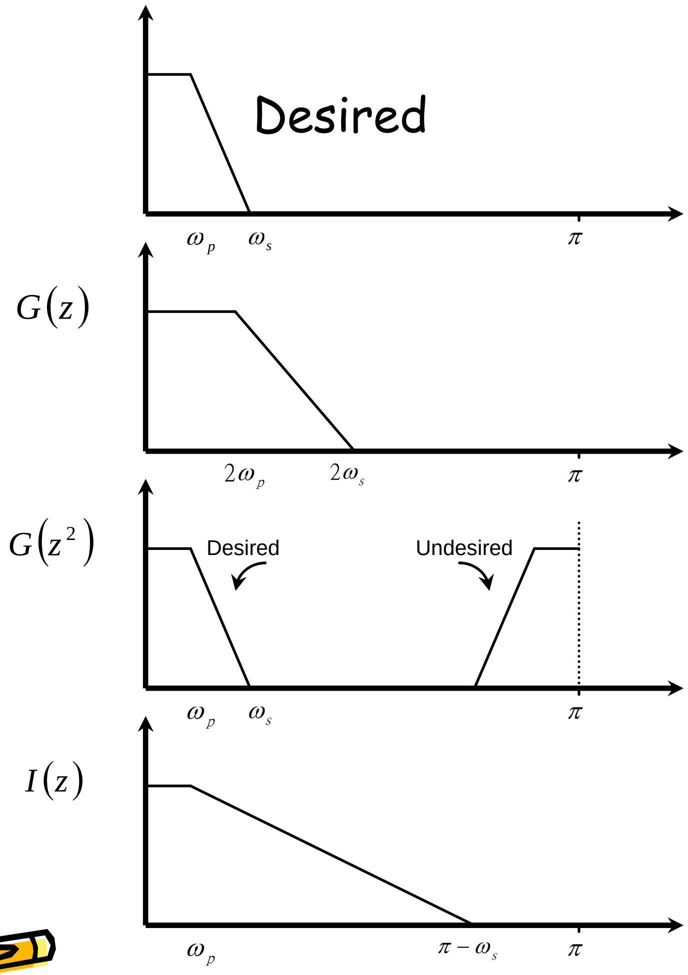
Multistage Implementation

- 2-stage interpolator



Interpolated FIR (IFIR)

with ripple not shown in these figures



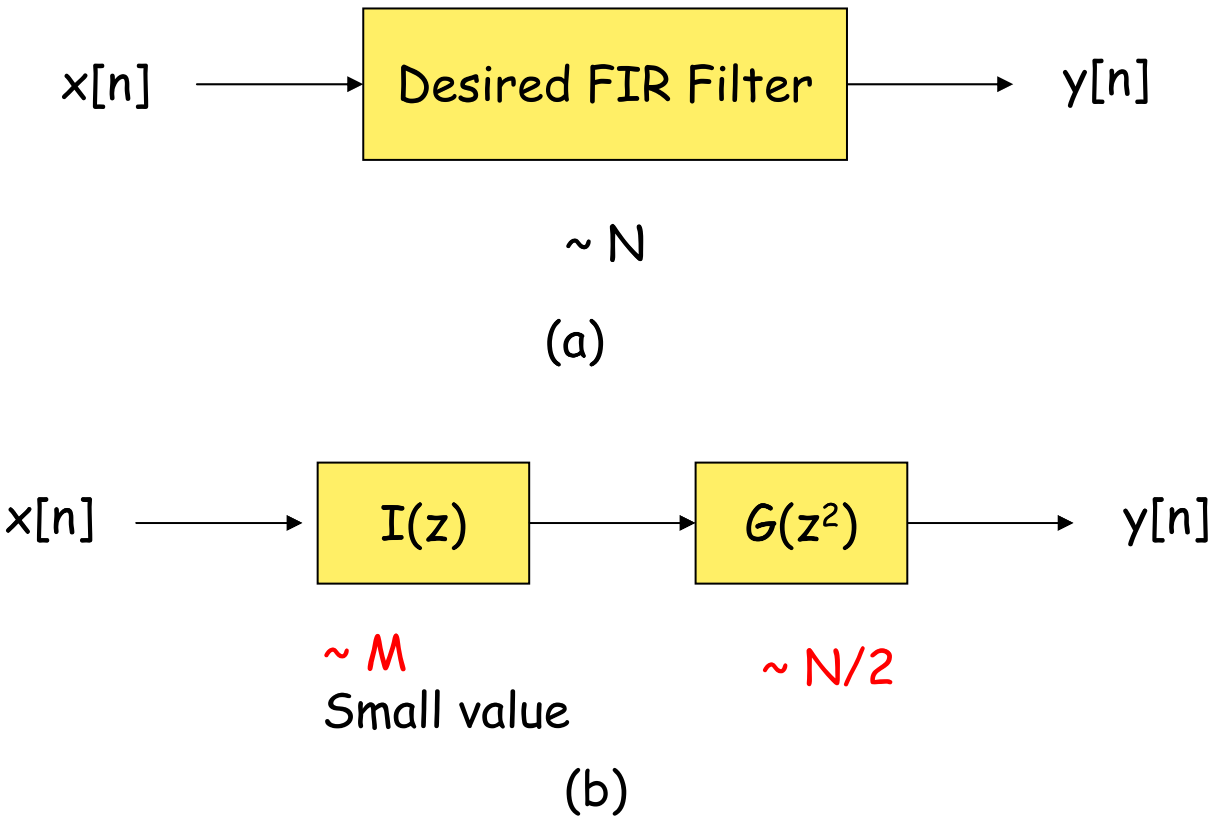
Desired narrowband response
Assume required filter order is N.

2-fold Stretched filter
Required filter order is reduced to $N/2$.

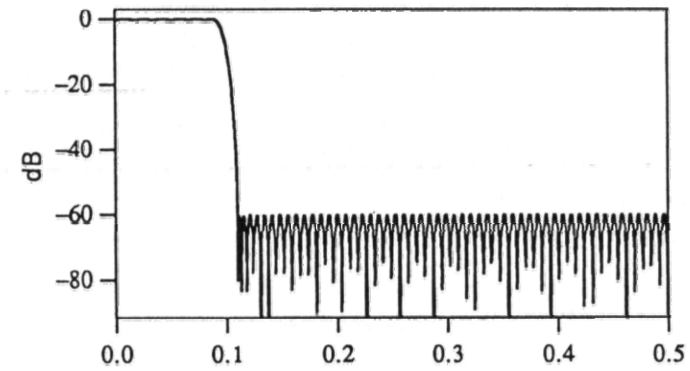
Interpolated version of stretched filter
Required filter order is still $N/2$.

Image suppresser
Required filter order is M.
Order $(N/2+M)$ is needed to implement!
$(N/2+M) << N$ for small M

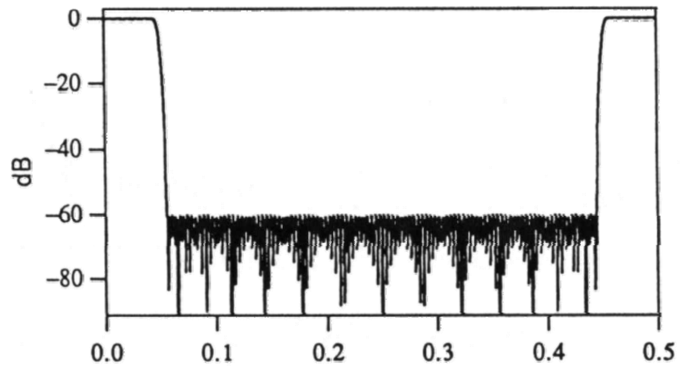
Design of Narrow Band Filter



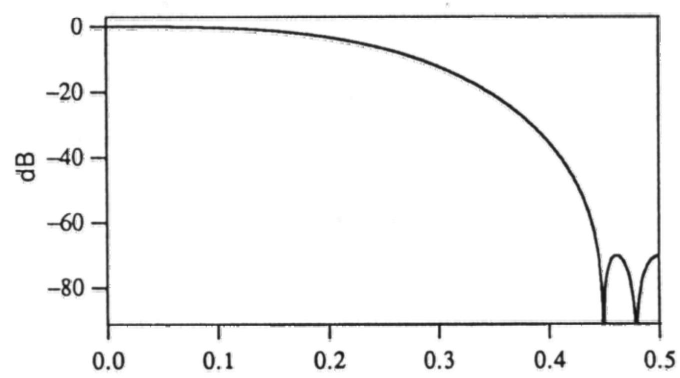
Interpolated FIR (IFIR)



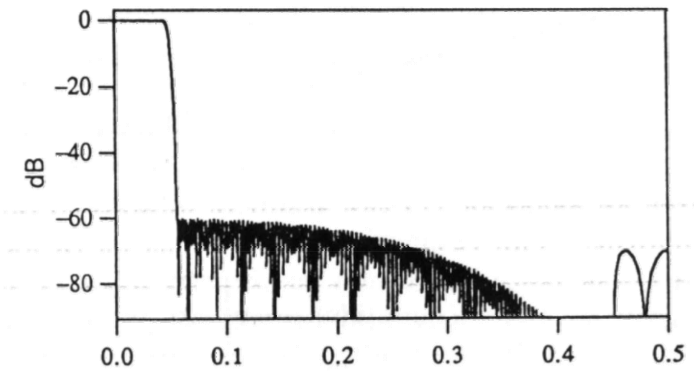
(a) $G(z)$



(a) $G(z^2)$



(b) $I(z)$



(a) $G(z^2)I(z)$

Interpolated FIR (IFIR)

Quantity Compared	Conventional Method	IFIR Method		
		G(z)	I(z)	Total
Filter order	233	131	6	268
Number of Multipliers	**117**	66	4	**70**
Number of Adders	233	131	6	137

Remarks

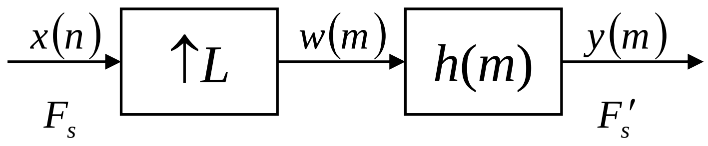
- The overall responses $G(z^2)I(z)$ can have larger ripple sizes. Adjusting ripple sizes is necessary to meet the desired specification.

Questions

- What is the best way to split the integer L (or M) into factors?
- In what order should these factors be arranged?
- It's hard to answer these questions. But, it's easy to mention that the multistage implementations result in more efficient systems
- Consider the following FIR LPF design…

Multistage Implementation

- More efficient implementation of the filter (in terms of # computations per time unit)



$$L = \prod_{i=1}^{I} L_i$$