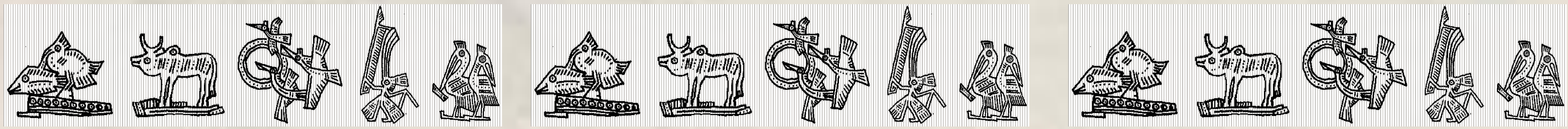
DCT – Wavelet – Filter Bank



Trac D.Tran
Department of Electrical and Computer Engineering
The Johns Hopkins University
Baltimore, MD 21218

Outline

- Reminder
  - Linear signal decomposition
  - Optimal linear transform: KLT, principal component analysis
- Discrete cosine transform
  - Definition, properties, fast implementation
- Review of multi-rate signal processing
- Wavelet and filter banks
  - Aliasing cancellation and perfect reconstruction
  - Spectral factorization: orthogonal, biorthogonal, symmetry
  - Vanishing moments, regularity, smoothness
  - Lattice structure and lifting scheme
  - *M*-band design – Local cosine/sine bases

Reminder: Linear Signal Representation

input signal

transform coefficient

basis function

Representation $$\mathbf{x} = \sum_{i=0}^{N} c_i \boldsymbol{\psi}_i$$

Decomposition $$c_i = \langle \mathbf{x}, \boldsymbol{\psi}_i \rangle$$

Approximation $$\hat{\mathbf{x}} = \sum_{i=0}^{L<<N} c_i \boldsymbol{\psi}_i$$

using as few coefficients as possible

Motivations

- Fundamental question: what is the best basis?
  - energy compaction: minimize a pre-defined error measure, say MSE, given *L* coefficients
  - maximize perceptual reconstruction quality
  - low complexity: fast-computable decomposition and reconstruction
  - intuitive interpretation
- How to construct such a basis? Different viewpoints!
- Applications
  - compression, coding
  - signal analysis
  - de-noising, enhancement
  - communications

KLT: Optimal Linear Transform

$$\mathbf{R}_{\mathbf{xx}}\mathbf{\Phi}_{\mathbf{i}} = \lambda_i \mathbf{\Phi}_{\mathbf{i}}$$

$$E\left[\mathbf{x}\mathbf{x}^T\right]$$

eigenvectors

$$KLT = \begin{bmatrix} | & | & & | \\ \mathbf{\Phi}_{\mathbf{0}} & \mathbf{\Phi}_{\mathbf{1}} & \cdots & \mathbf{\Phi}_{\mathbf{N-1}} \\ | & | & & | \end{bmatrix}$$

- Signal dependent
- Require stationary signals
- How do we communicate bases to the decoder?
- How do we design “good” signal-independent transform?

Discrete Cosine Transforms

- Type I

$$K_i = \begin{cases} 1/\sqrt{2}, & i = 0, M \\ 1, & \text{otherwise} \end{cases}$$

$$\left[C^{I}\right] = \sqrt{\frac{2}{M}}\left[K_m K_n \cos\left(\frac{mn\pi}{M}\right)\right], \qquad m,n \in \{0,1,\cdots,M\}$$

- Type II

$$\boxed{\left[C^{II}\right] = \sqrt{\frac{2}{M}}\left[K_m \cos\left(\frac{m(n+1/2)\pi}{M}\right)\right], \qquad m,n \in \{0,1,\cdots,M-1\}}$$

- Type III

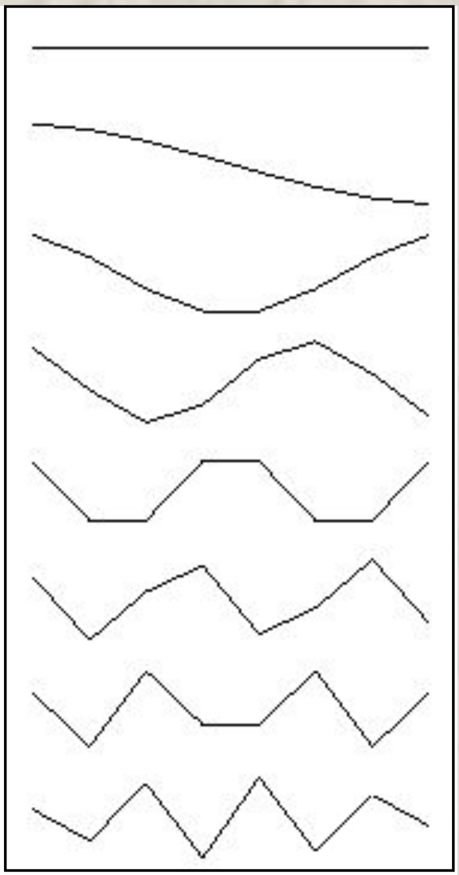
$$\left[C^{III}\right] = \sqrt{\frac{2}{M}}\left[K_n \cos\left(\frac{(m+1/2)n\pi}{M}\right)\right], \qquad m,n \in \{0,1,\cdots,M-1\}$$

- Type IV

$$\left[C^{IV}\right] = \sqrt{\frac{2}{M}}\left[\cos\left(\frac{(m+1/2)(n+1/2)\pi}{M}\right)\right], \qquad m,n \in \{0,1,\cdots,M-1\}$$

DCT Type-II

DCT basis

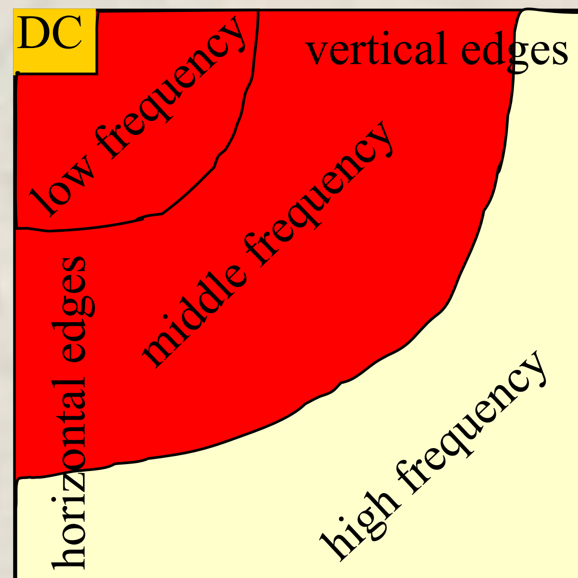


$$\begin{cases} X[m] = \sqrt{\dfrac{2}{M}}\, K_m \displaystyle\sum_{n=0}^{M-1} x[n] \cos\left[\dfrac{(2n+1)m\pi}{2M}\right] \\ x[n] = \sqrt{\dfrac{2}{M}}\, K_n \displaystyle\sum_{m=0}^{M-1} X[m] \cos\left[\dfrac{(2m+1)n\pi}{2M}\right] \end{cases}$$

$$m, n = 0, 1, \dots, M-1$$

$$K_i = \begin{cases} \dfrac{1}{\sqrt{2}}, & i = 0 \\ 1, & i \neq 0 \end{cases}$$

8 x 8 block



- orthogonal
- real coefficients
- symmetry
- near-optimal
- fast algorithms

DCT Symmetry

$$\cos\left(\frac{m(2(M-1-n)+1)\pi}{2M}\right)$$

$$=\cos\left(\frac{(2M-2-2n+1)m\pi}{2M}\right)$$

$$=\cos\left[\frac{2Mm\pi}{2M}-\frac{(2n+1)m\pi}{2M}\right]$$

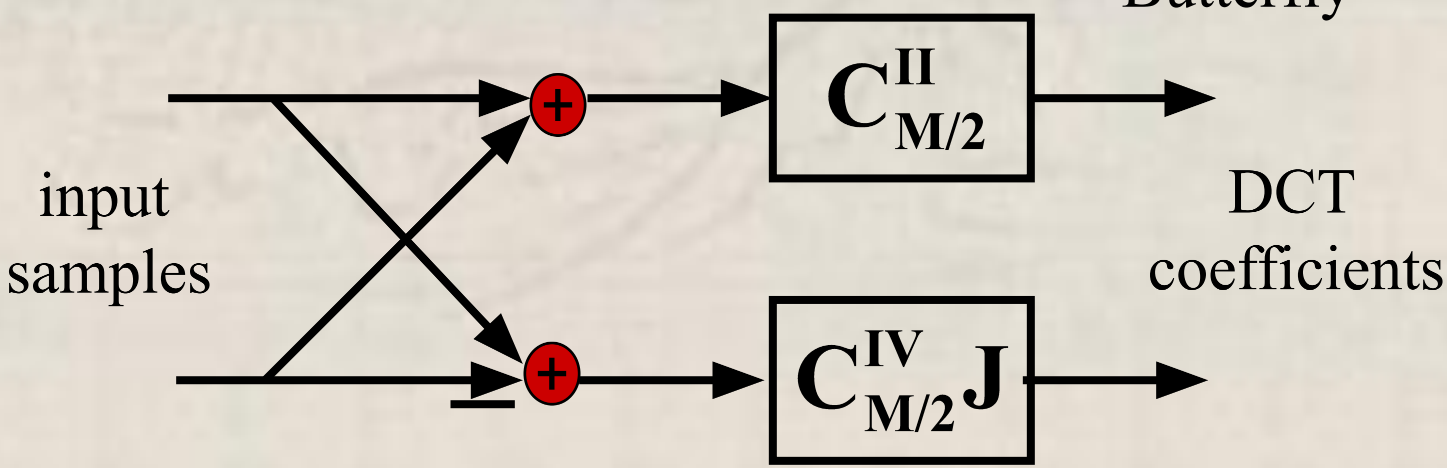
$$=\pm\cos\left[\frac{(2n+1)m\pi}{2M}\right]$$

DCT basis functions are either symmetric or anti-symmetric

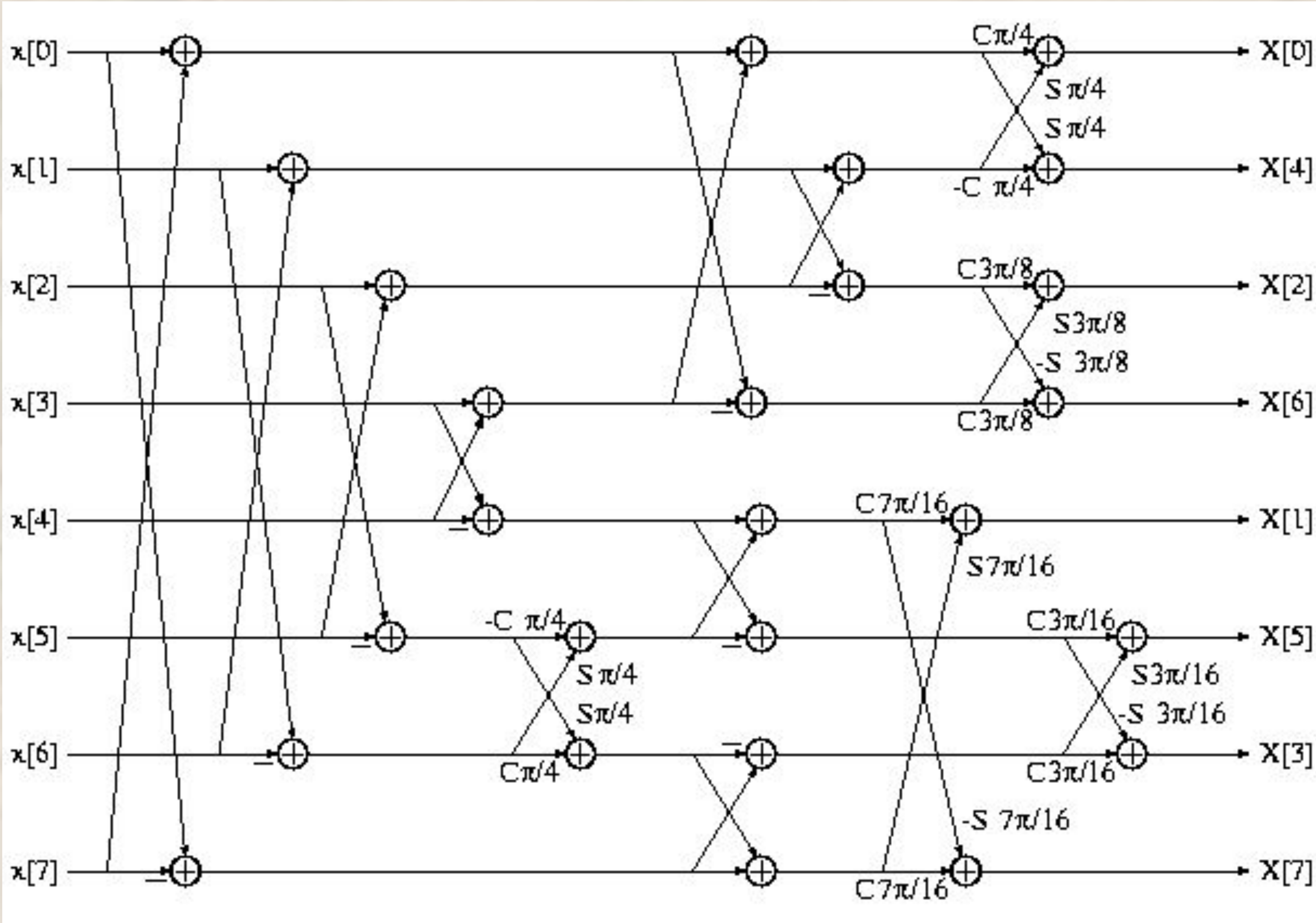
DCT: Recursive Property

- An M-point DCT–II can be implemented via an M/2-point DCT–II and an M/2-point DCT–IV

$$\left[C_M^{II}\right] = \frac{1}{\sqrt{2}} \begin{bmatrix} \boxed{C_{M/2}^{II}} & 0 \\ 0 & \boxed{C_{M/2}^{IV}} J \end{bmatrix} \underbrace{\begin{bmatrix} I & J \\ J & -I \end{bmatrix}}_{\text{Butterfly}}$$



Fast DCT Implementation



13 multiplications and 29 additions per 8 input samples

Block DCT

$$\begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_N \end{bmatrix} = \begin{bmatrix} \mathbf{C}_M^{II} & \mathbf{0} & & & \\ \mathbf{0} & \mathbf{C}_M^{II} & \mathbf{0} & & \\ & \mathbf{0} & \mathbf{C}_M^{II} & \mathbf{0} & \\ & & & \mathbf{0} & \mathbf{C}_M^{II} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}$$

output blocks
of DCT coefficients,
each of size M

input blocks,
each of size M

Filtering

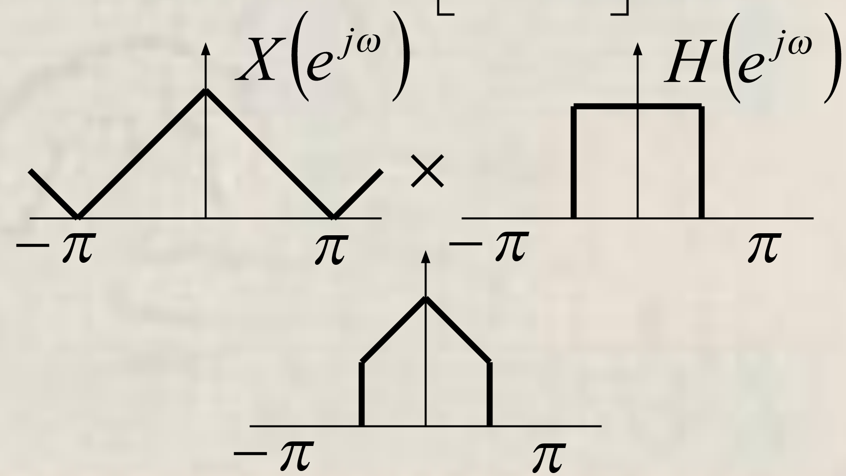
$$x[n] \longrightarrow \boxed{h[n]} \longrightarrow y[n]$$

$$y[n] = \sum_k h[n-k]x[k] = \sum_k h[k]x[n-k]$$

$$\begin{bmatrix} \vdots \\ y[n-1] \\ \boxed{y[n]} \\ \boxed{y[n+1]} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & h[2] & h[1] & h[0] & 0 & 0 & \vdots \\ \vdots & 0 & \boxed{h[2] \quad h[1] \quad h[0]} & & & 0 & \vdots \\ \vdots & 0 & 0 & \boxed{h[2] \quad h[1] \quad h[0]} & & & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \vdots \\ x[n-2] \\ x[n-1] \\ x[n] \\ x[n+1] \\ \vdots \end{bmatrix}$$

z – Transform : $H(z) = \sum_n h[n] z^{-n}$

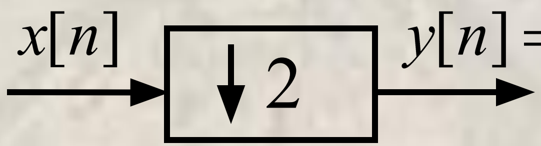
Fourier : $H(e^{j\omega}) = \sum_n h[n] e^{-j\omega n}$



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

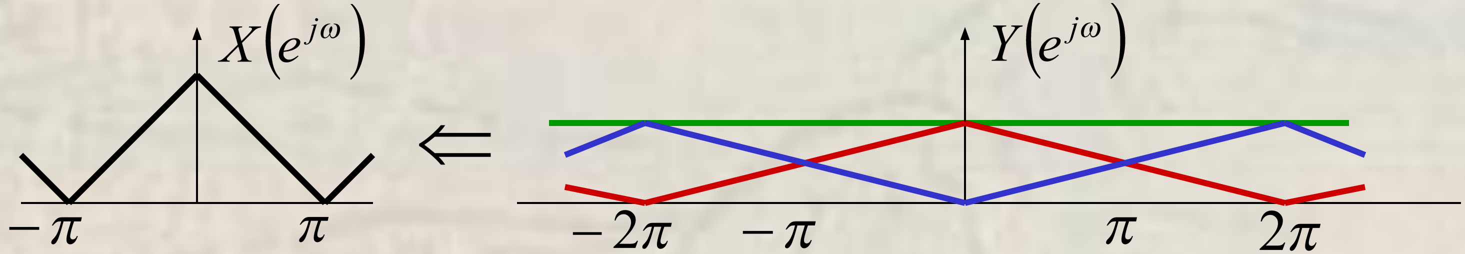
LTI Operator

Down-Sampling



$$
\begin{bmatrix} \vdots \\ y[-1] \\ \boxed{y[0]} \\ \boxed{y[1]} \\ \vdots \end{bmatrix}
=
\begin{bmatrix}
\ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
\cdots & 1 & 0 & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & \boxed{1} & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & \boxed{1} & \cdots \\
\ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
\begin{bmatrix} \vdots \\ x[-1] \\ \boxed{x[0]} \\ x[1] \\ \boxed{x[2]} \\ \vdots \end{bmatrix}
$$

Linear Time-Variant Lossy Operator



$$Y\left(e^{j\omega}\right) = \frac{1}{2}\left[X\left(e^{j\omega/2}\right) + X\left(e^{j(\omega-2\pi)/2}\right)\right]$$

$$\boxed{Y(z)} = \frac{1}{2}\left[\boxed{X\left(z^{1/2}\right)} + \boxed{X\left(-z^{1/2}\right)}\right]$$

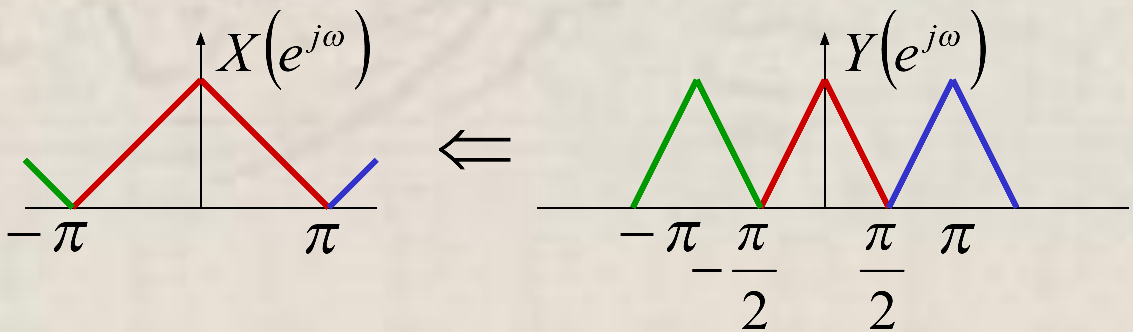
Up-Sampling

$$x[n] \rightarrow \boxed{\uparrow 2} \rightarrow y[n] = \begin{cases} x[n/2], & n \text{ even} \\ 0, & n \text{ odd} \end{cases}$$

$$\begin{bmatrix} \vdots \\ y[-1] \\ \boxed{y[0]} \\ y[1] \\ \boxed{y[2]} \\ \vdots \end{bmatrix} = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \\ \cdots & 1 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & \cdots \\ \cdots & 0 & \boxed{1} & 0 & \cdots \\ \cdots & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & \boxed{1} & \cdots \\ & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ x[-1] \\ \boxed{x[0]} \\ x[1] \\ \boxed{x[2]} \\ \vdots \end{bmatrix}$$

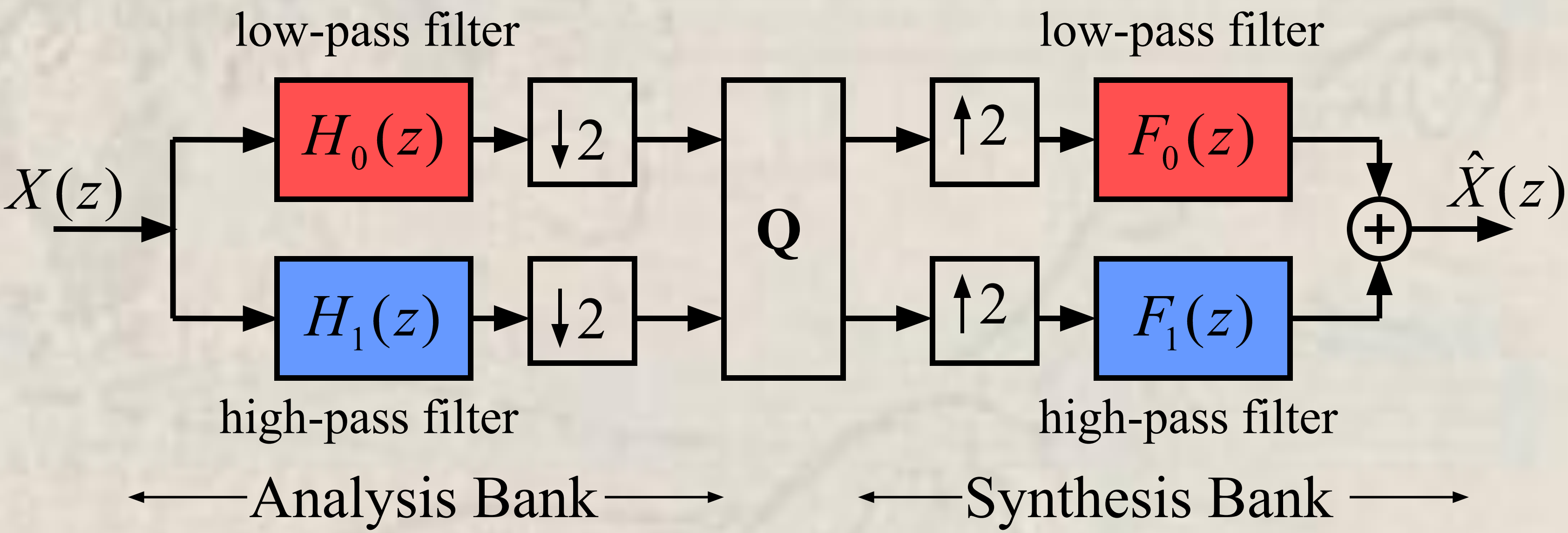
$$Y\left(e^{j\omega}\right) = X\left(e^{j2\omega}\right)$$

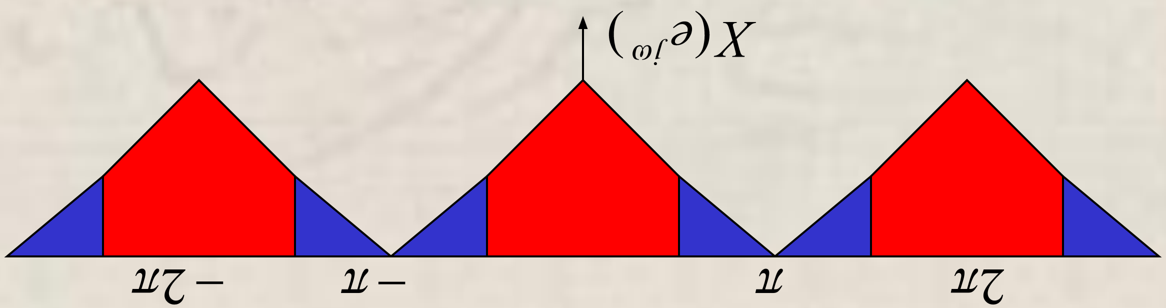
$$Y(z) = X\left(z^2\right)$$



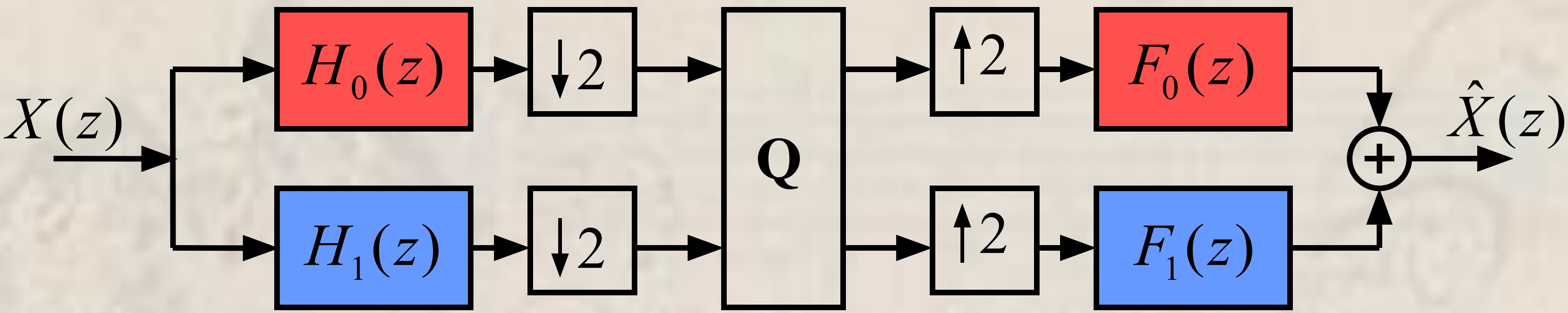
Filter Bank

- First FB designed for speech coding, [Croisier-Esteban-Galand 1976]
- Orthogonal FIR filter bank, [Smith-Barnwell 1984], [Mintzer 1985]





FB Analysis



$$\hat{X}(z) = \frac{1}{2}\overbrace{\left[F_0(z)H_0(z) + F_1(z)H_1(z)\right]}^{\text{for Distortion Elimination, set to } 2z^{-l}}X(z) \qquad = z^{-l}X(z)$$

$$+\frac{1}{2}\underbrace{\left[F_0(z)H_0(-z) + F_1(z)H_1(-z)\right]}_{\text{for Aliasing Cancellation, set to } 0!}X(-z)$$

$$\boxed{\begin{aligned} F_0(z) &= H_1(-z) \\ F_1(z) &= -H_0(-z) \end{aligned}}$$

Alternating-Sign Construction

Perfect Reconstruction

- With **Aliasing Cancellation**

$$\boxed{\begin{aligned} F_0(z) &= H_1(-z) \\ F_1(z) &= -H_0(-z) \end{aligned}}$$

- **Distortion Elimination** becomes

$$\boxed{F_0(z)H_0(z) - F_0(-z)H_0(-z) = 2z^{-l}}$$

$$\Rightarrow \boxed{P_0(z) - P_0(-z) = 2z^{-l}} \quad \text{where} \quad P_0(z) \equiv F_0(z)H_0(z)$$

Half-band Filter

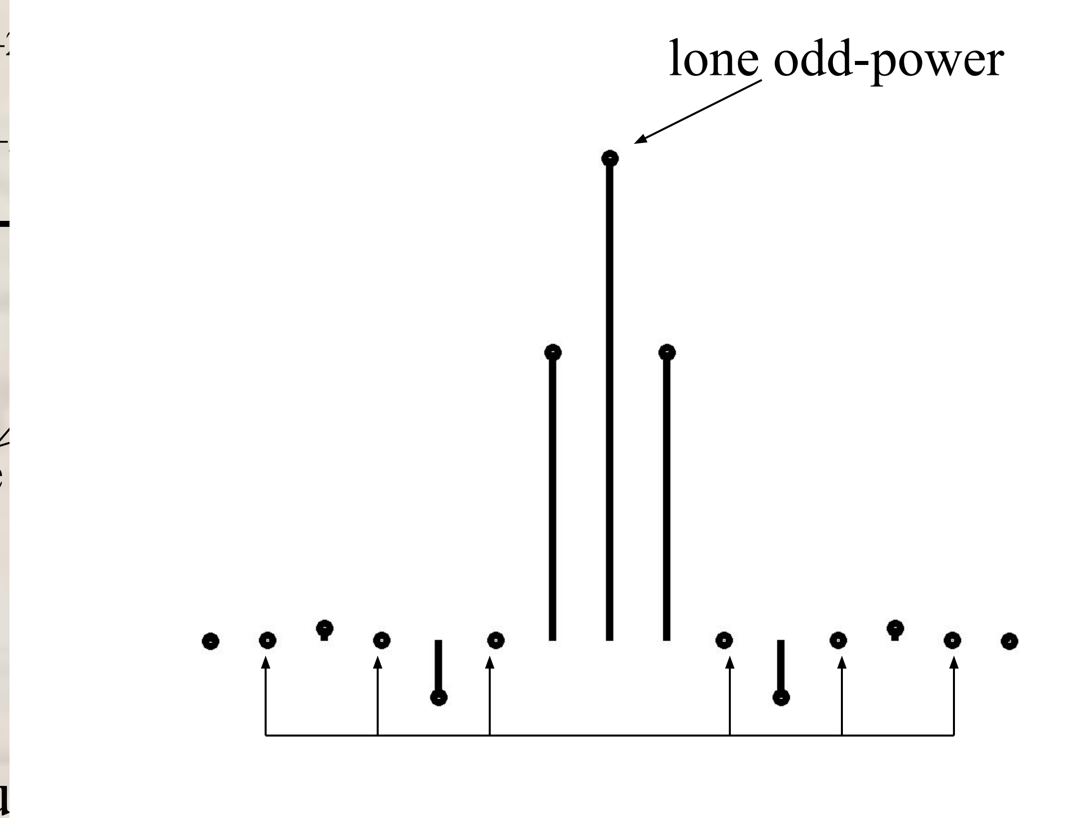
Half-band Filter

$$P_0(z) = a + bz^{-1} + cz^{-\cdots}$$

$$-\quad P_0(-z) = a - bz^{-1} + cz^{-\cdots}$$

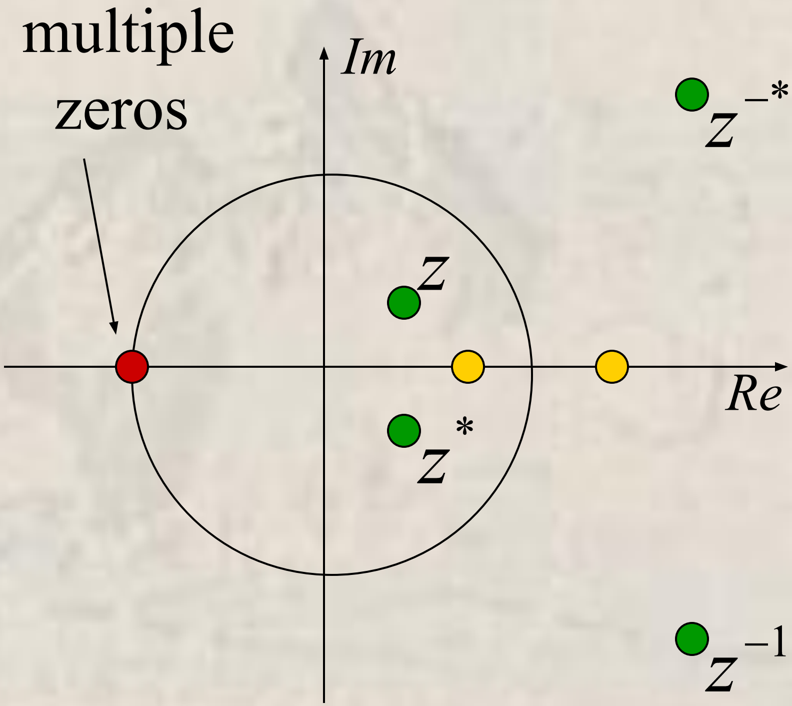
$$P_0(z) - P_0(-z) = 0 + 2bz^{-1} + 0 \cdots$$

can only have



- Standard design procedu
  - Design a good low-pass half-band filter $P_0(z)$
  - Factor $P_0(z)$ into $H_0(z)$ and $F_0(z)$
  - Use the aliasing cancellation condition to obtain $H_1(z)$ and $F_1(z)$

Spectral Factorization



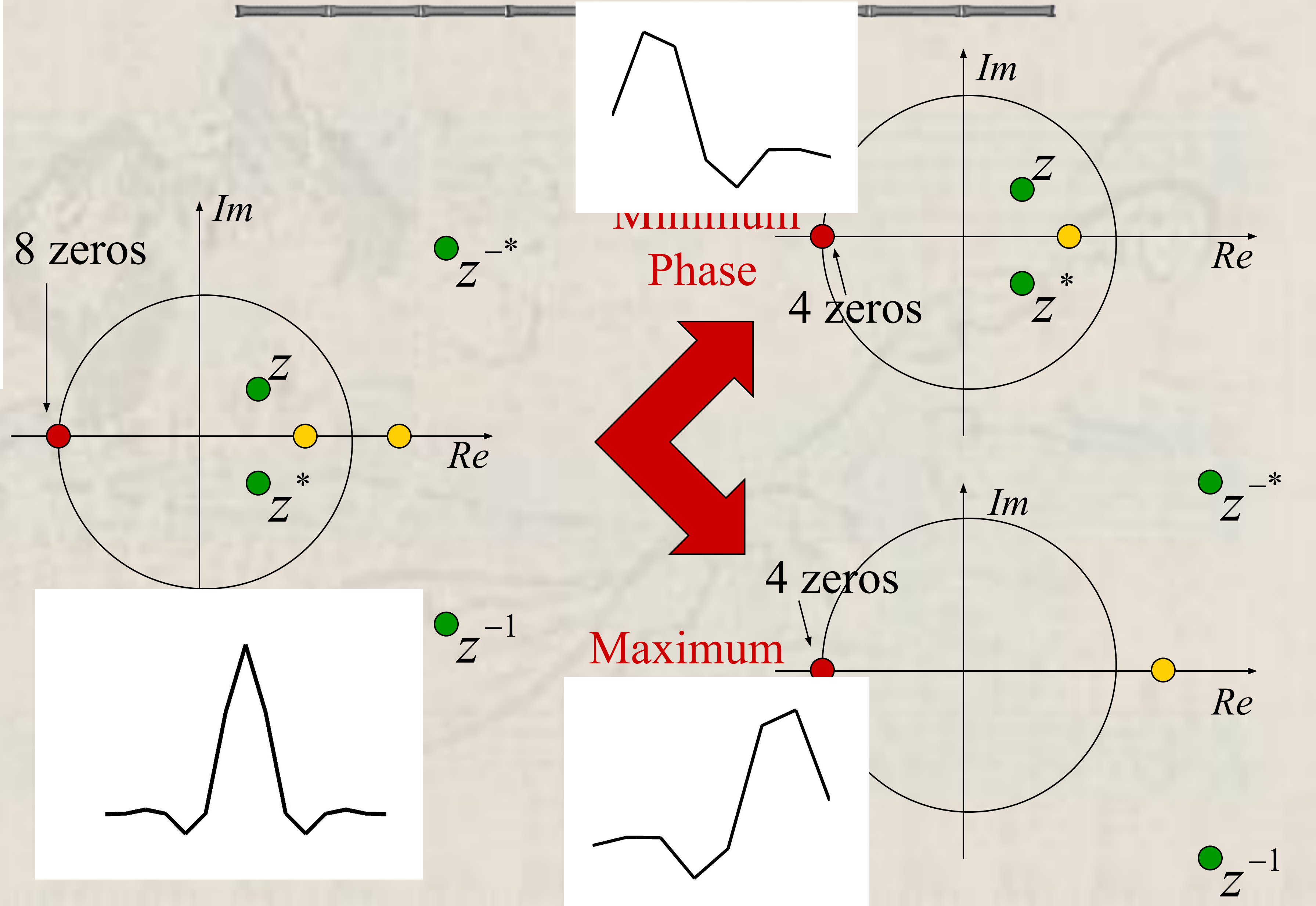
Zeros of Half-band Filter

- Real-coefficient
  - *z* and *z** must stay together
- Orthogonality
  - $f_i[n] = h_i[-n]$
  - $F_i(z) = H_i(z^{-1})$
  - *z* and *z^-1* must be separated
- Symmetry
  - $h_i[n] = \pm h_i[L-1-n]$
  - $H_i(z) = \pm z^{-(L-1)} H_i(z^{-1})$
  - *z* and *z^-1* must stay together

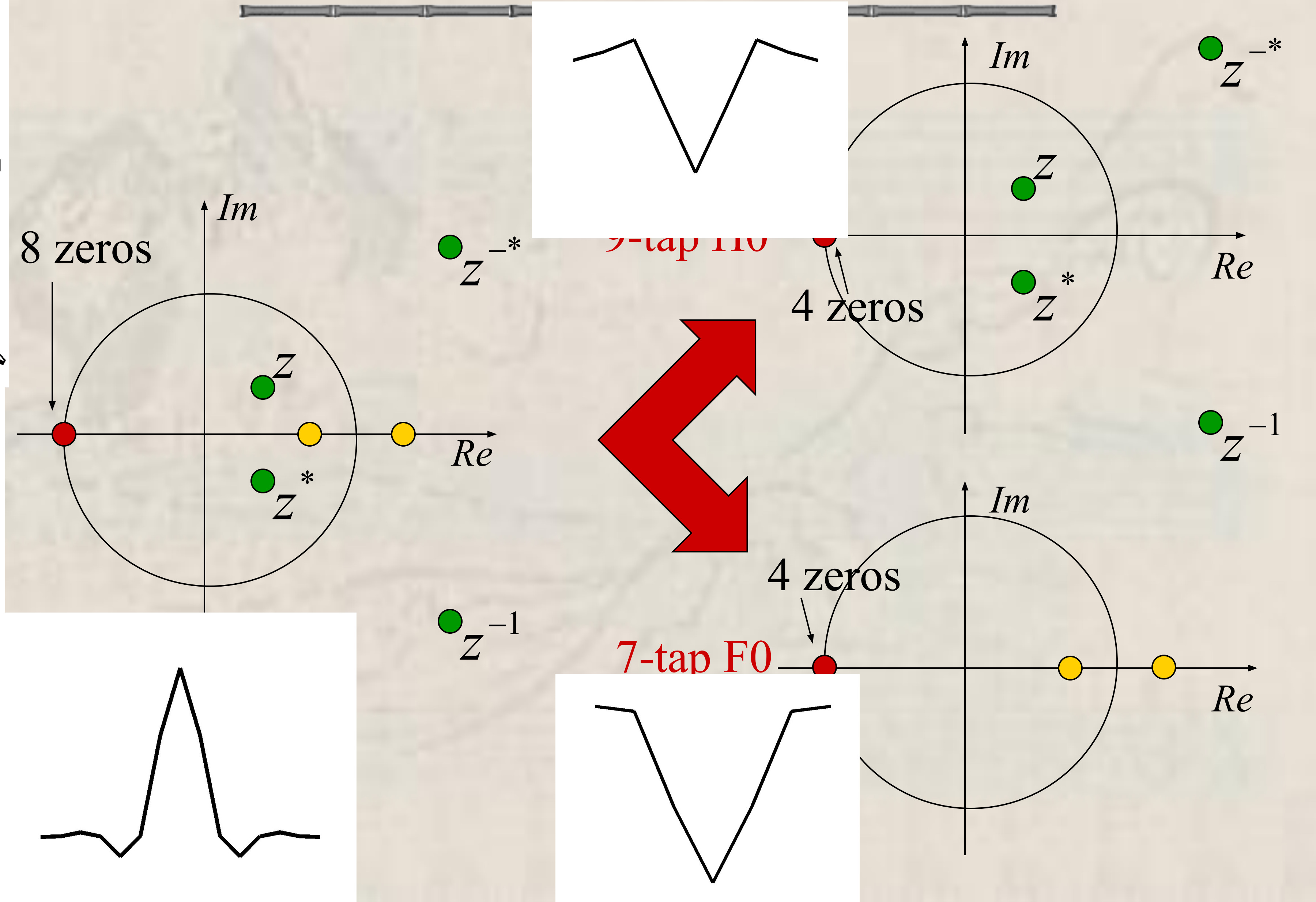
$$P_0(z) = \prod_n (1 - z_n z^{-1}); \qquad \{z_n\} = roots$$

$$H_0(z) = \prod_{k \in H} (1 - z_k z^{-1}) \qquad F_0(z) = \prod_{l \in F} (1 - z_l z^{-1})$$

Spectral Factorization: Orthogonal



Spectral Factorization: Symmetry

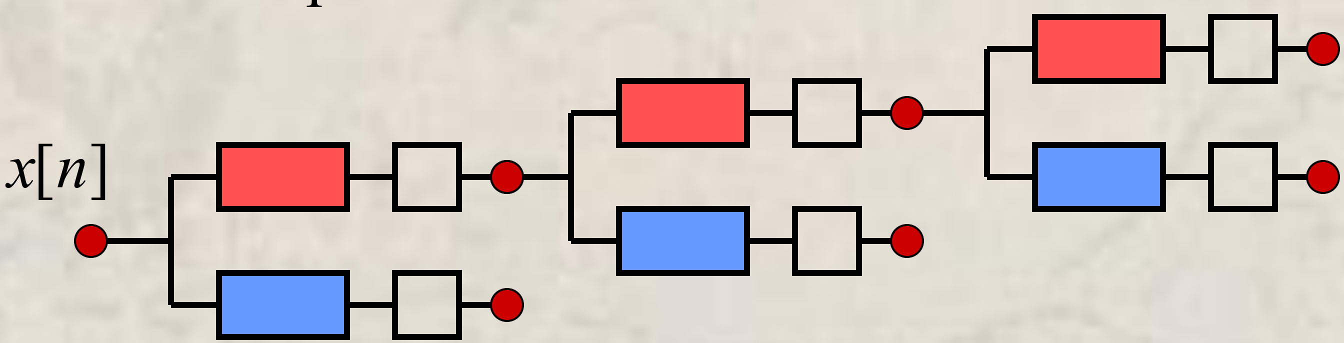


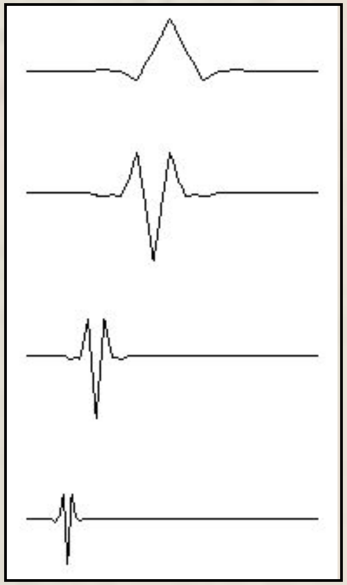
History: Wavelets

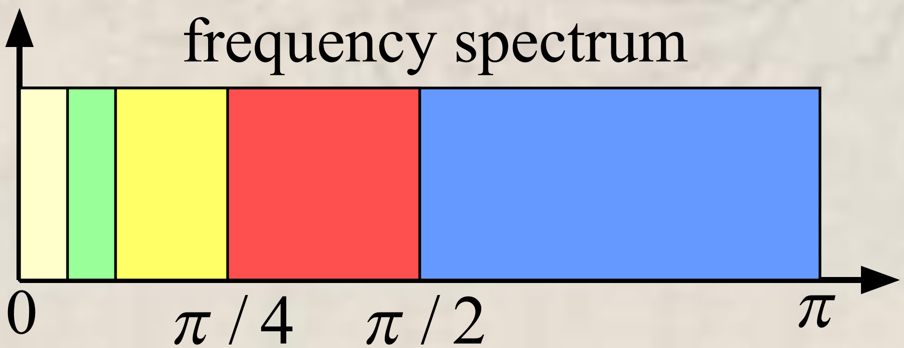
- Early wavelets: for geophysics, seismic, oil-prospecting applications, [Morlet-Grossman-Meyer 1980-1984]
- Compact-support wavelets with smoothness and regularity, [Daubechies 1988]
- Linkage to filter banks and multi-resolution representation, fast discrete wavelet transform (DWT), [Mallat 1989]
- Even faster and more efficient implementations: lattice structure for filter banks, [Vaidyanathan-Hoang 1988]; lifting scheme, [Sweldens 1995]

From Filter Bank to Wavelet

- [Daubechies 1988], [Mallat 1989]
- Constructed as iterated filter bank
- Discrete Wavelet Transform (DWT): iterate FB on the lowpass subband

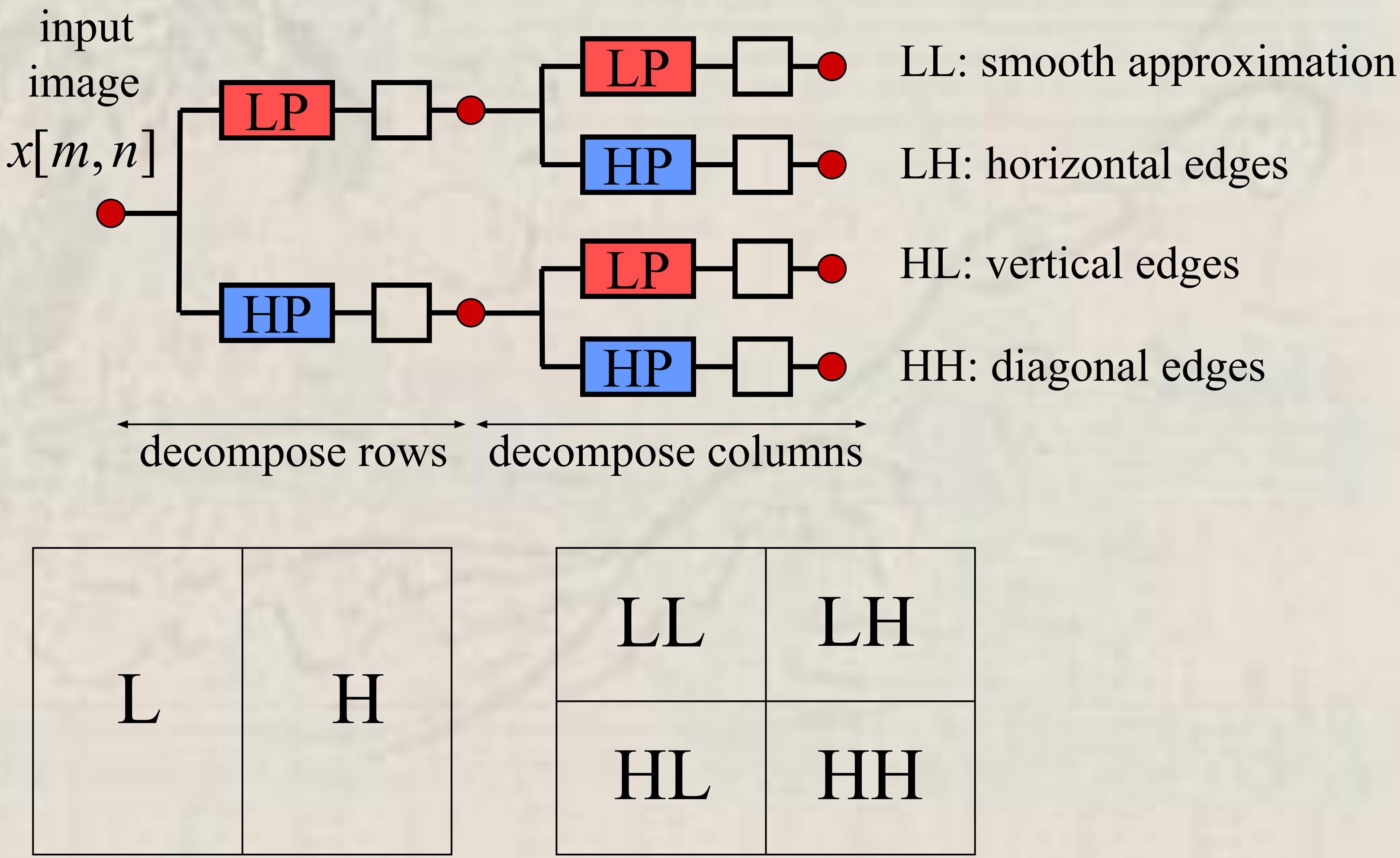




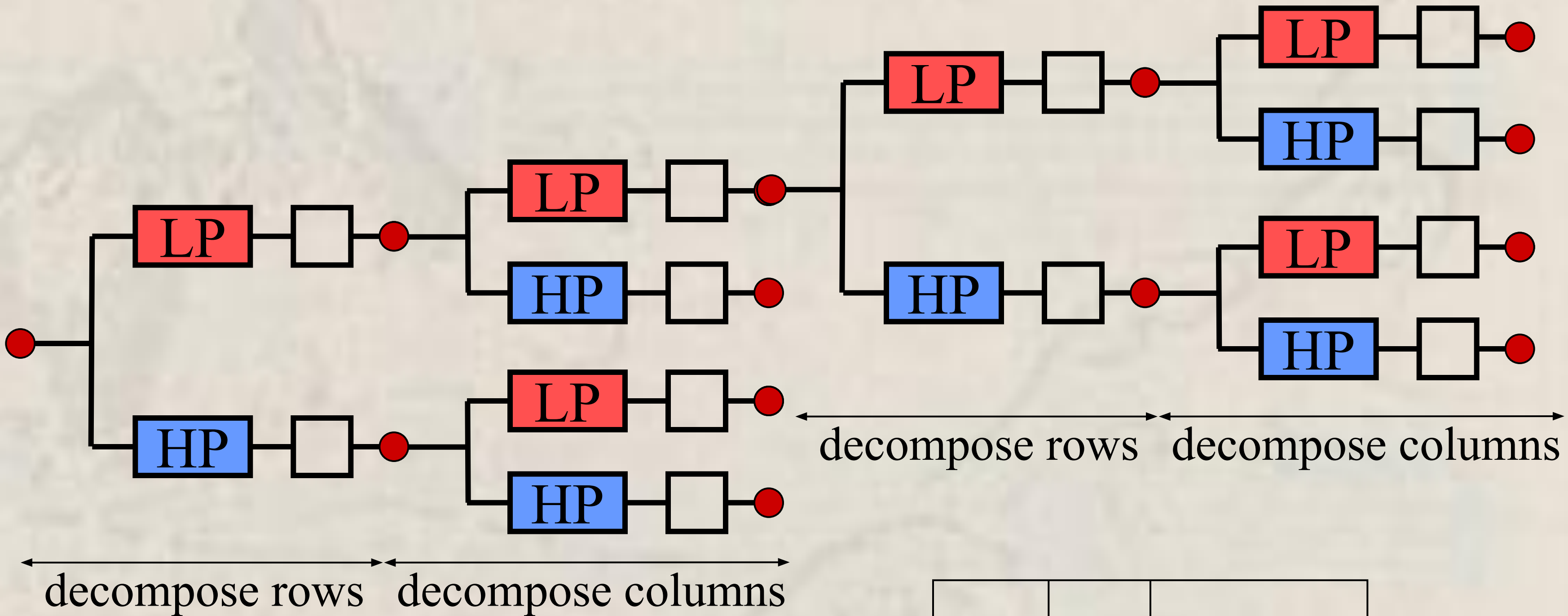


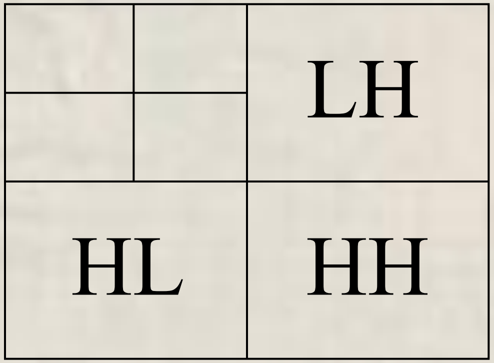
$$x(t) = \sum_{k=-\infty}^{\infty} \sum_{i=0}^{\infty} c_{i,k} \Psi(2^i t - k)$$

1-Level 2D DWT

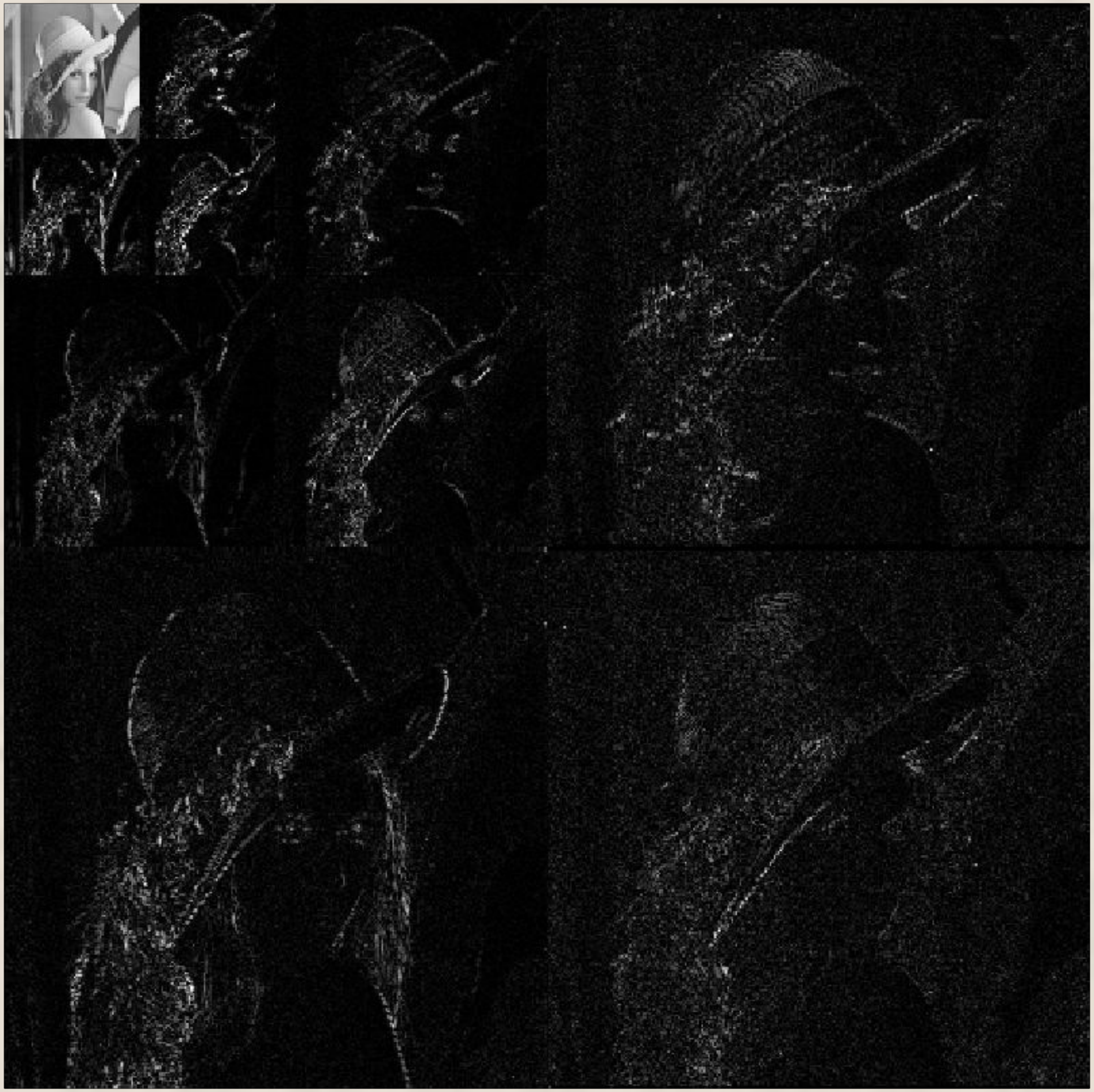


2-Level 2D DWT



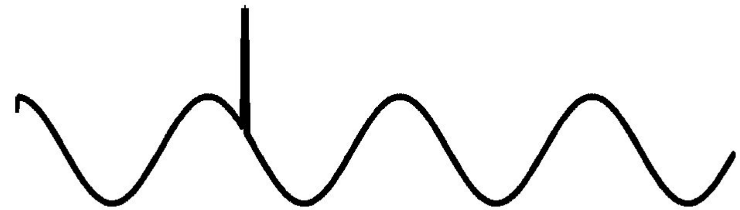


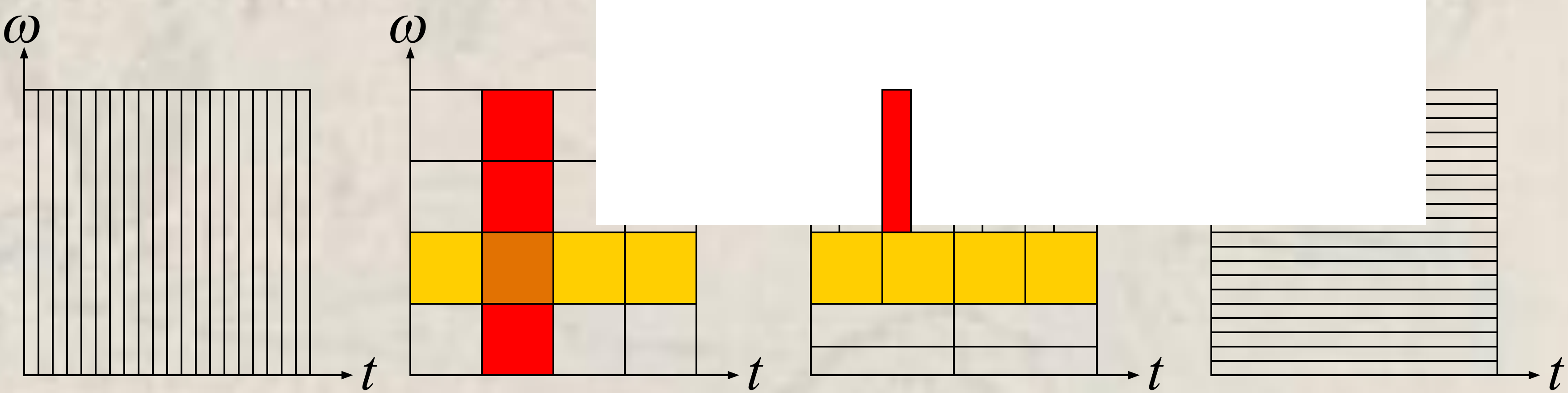
2D DWT



Time-Freq

$x[n] = \cos(a\pi n) + \delta[n]$





best time localization

STFT uniform tiling

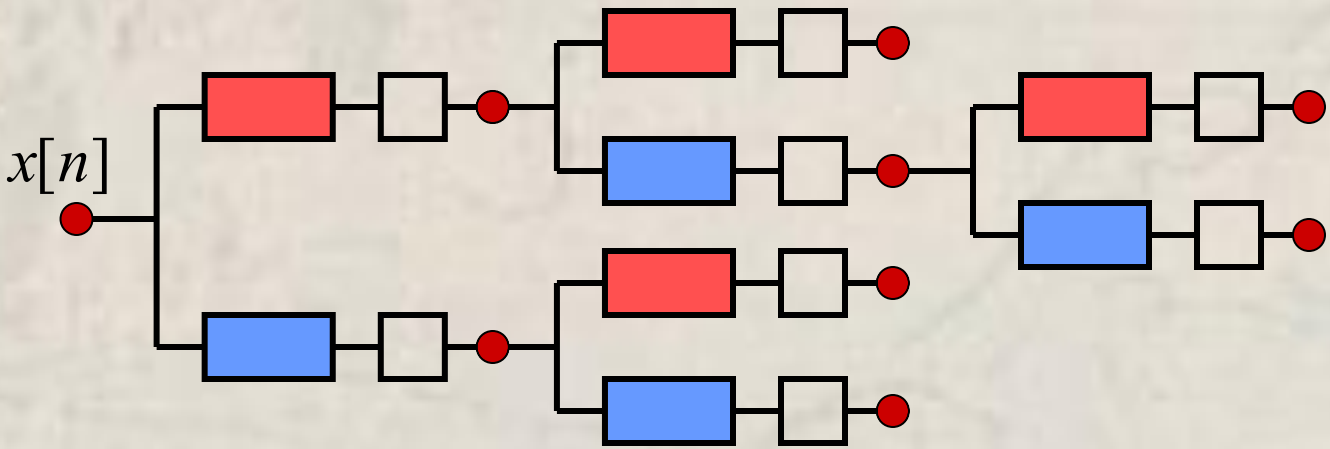
wavelet dyadic tiling

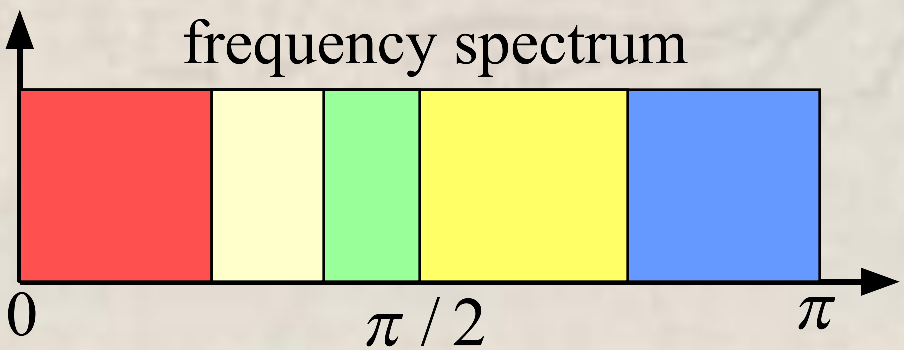
best frequency localization

- Heisenberg's Uncertainty Principle: bound on T-F product
- Wavelets provide flexibility and good time-frequency trade-off

Wavelet Packet

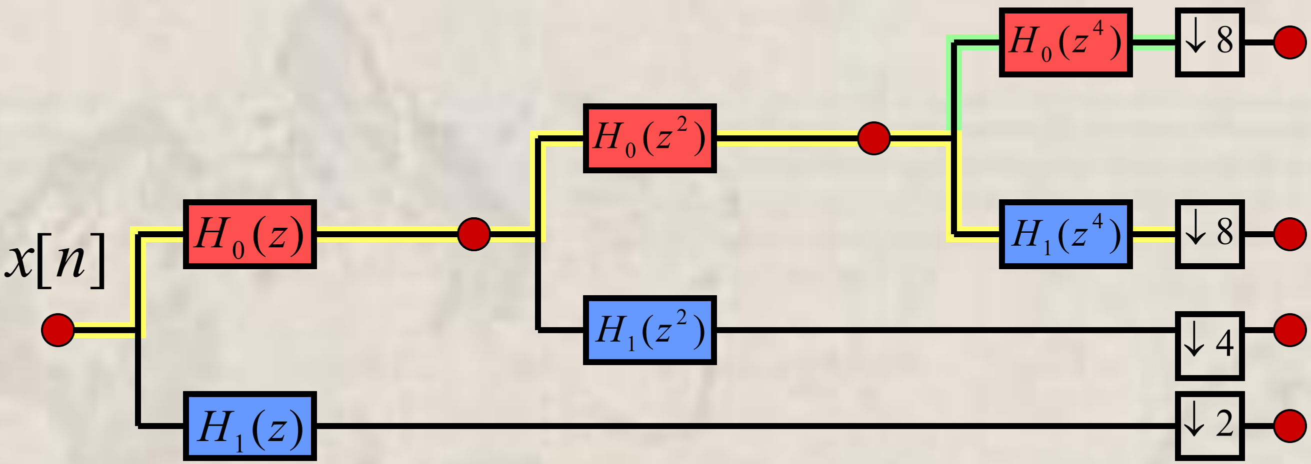
- Iterate adaptively according to the signal
- Arbitrary tiling of the time-frequency plain





Question: can we iterate any FB to construct wavelets and wavelet packets?

Scaling and Wavelet Function



Discrete Basis

$$\begin{cases} H_0^L(z) = \prod_{k=1}^{L} H_0\left(z^{2^{k-1}}\right) \\ H_1^i(z) = \left[\prod_{k=1}^{i-1} H_0\left(z^{2^{k-1}}\right)\right] H_1\left(z^{2^{i-1}}\right) \end{cases}$$

product filters

Continuous-time Basis

scaling function

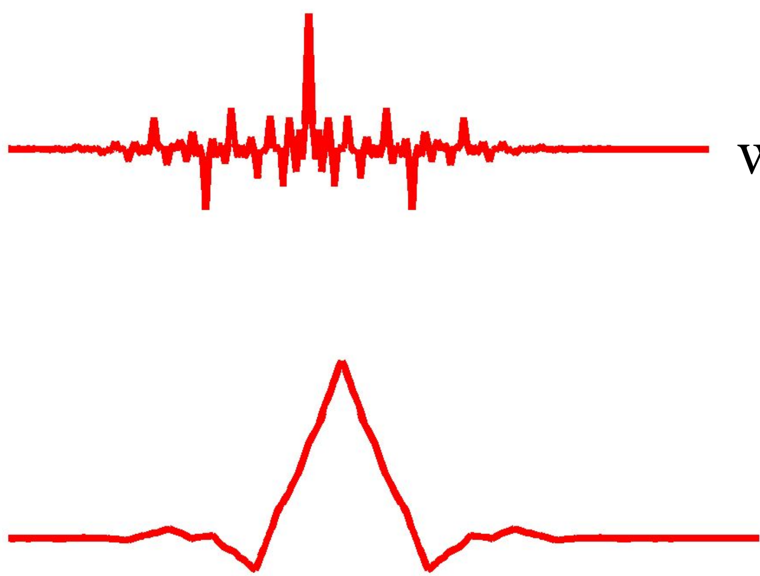
$$\begin{cases} \Phi(\omega) = \prod_{k=1}^{\infty} \frac{1}{\sqrt{2}} H_0\left(\frac{\omega}{2^k}\right) \quad \Leftrightarrow \quad \varphi(\mathrm{t}) = \sqrt{2}\sum_n h_0[n]\varphi(2t-n) \\ \Psi(\omega) = \frac{1}{\sqrt{2}} H_1\left(\frac{\omega}{2}\right) \prod_{k=2}^{\infty} \frac{1}{\sqrt{2}} H_0\left(\frac{\omega}{2^k}\right) \quad \Leftrightarrow \quad \psi(\mathrm{t}) = \sqrt{2}\sum_n h_1[n]\varphi(2t-n) \end{cases}$$

wavelet function

Convergence & Smoothness

- Not all FB yields nice product filters
- Two fundamental questions
  - Will the infinite product converge?
  - Will the infinite product converge to a smooth function?
- Necessary condition for convergence: at least a zero at $\omega = \pi$

many zeros at $\omega = \pi$



with 2 zeros at $\omega = \pi$

with 4 zeros at $\omega = \pi$

Regularity & Vanishing Moments

- In an orthogonal filter bank, the scaling filter has K vanishing moments (or is K-regular) if and only if
  - Scaling filter has K zeros at $\omega = \pi$
  - $\sum_n n^k h_1[n] = 0, \quad k = 0,1,\ldots,K-1$
  - All polynomial sequences up to degree (K-1) can be expressed as a linear combination of integer-shifted scaling filters [Daubechies]

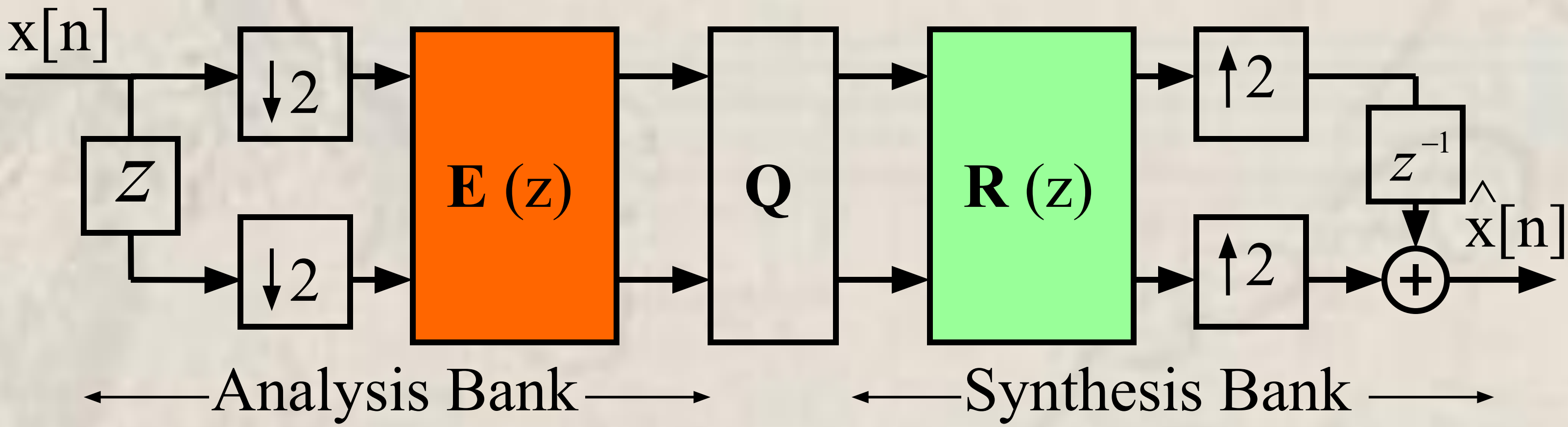
- Design Procedure: max-flat half-band spectral factorization

enforce the half-band condition here

$$P_0(z) = \left(1 + z^{-1}\right)^{2K} Q(z)$$

maximize the number of vanishing moments

Polyphase Representation



$R(z), E(z)$: 2 x 2 polynomial matrices

Perfect Reconstruction: $R(z)E(z) = z^{-l} I$

FIR filters: $|E(z)|, |R(z)|$ = monomial

$h_0[n] = [\ a \quad b \quad c \quad d\ ]$

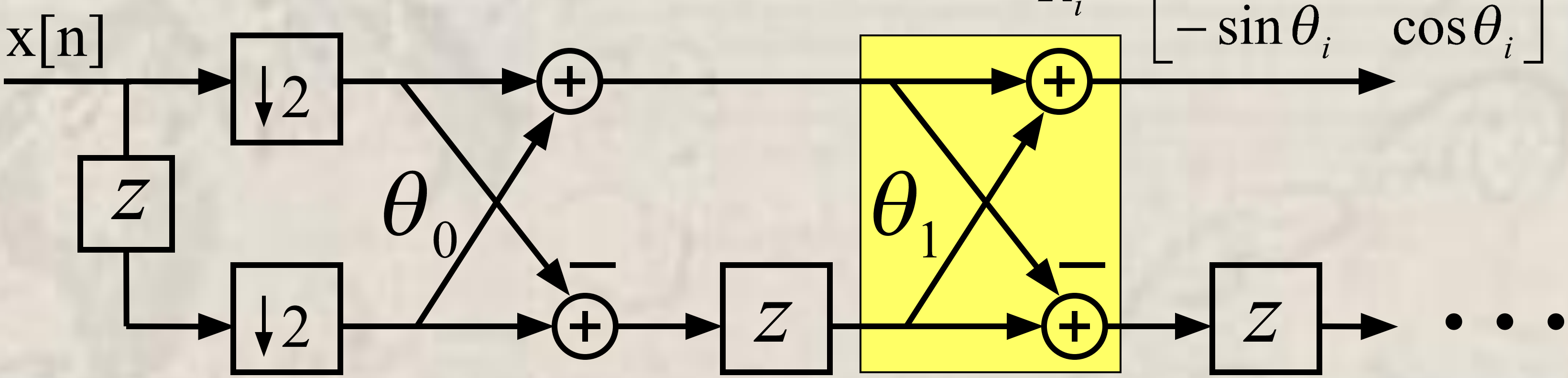
$h_1[n] = [\ d \quad -c \quad b \quad -a\ ]$

$$\mathbf{E}(z) = \begin{bmatrix} a+cz & b+dz \\ d-bz & -c-az \end{bmatrix}$$

Lattice Structure

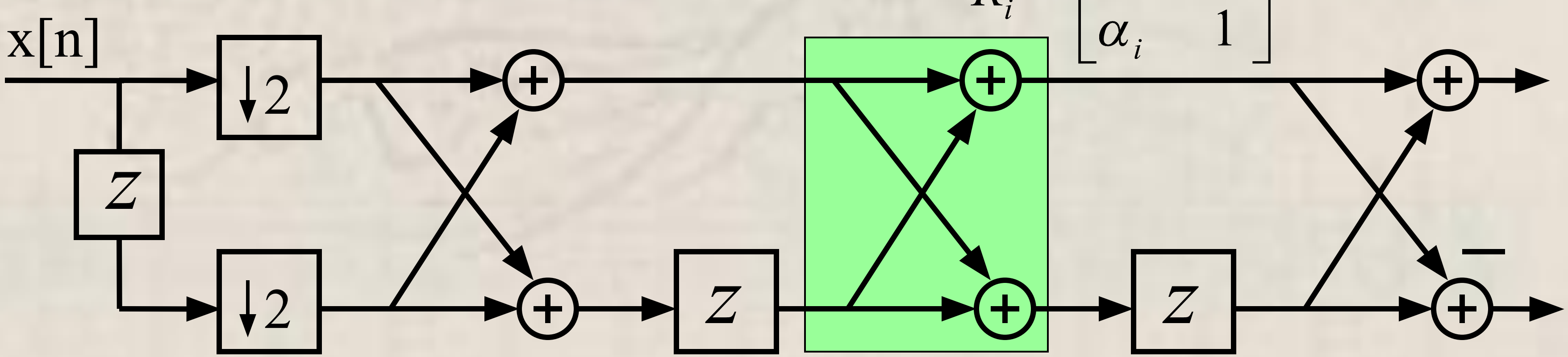
- Orthogonal Lattice

$$R_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix}$$



- Linear-Phase Lattice

$$R_i = \begin{bmatrix} 1 & \alpha_i \\ \alpha_i & 1 \end{bmatrix}$$



FB Design from Lattice Structure

- Set of free parameters $\{\theta_i\}$ or $\{\alpha_\mathrm{i}\}$
- Modular construction, well-conditioned, nice built-in properties
- Complete characterization: lattice covers all possible solutions
- Unconstrained optimization

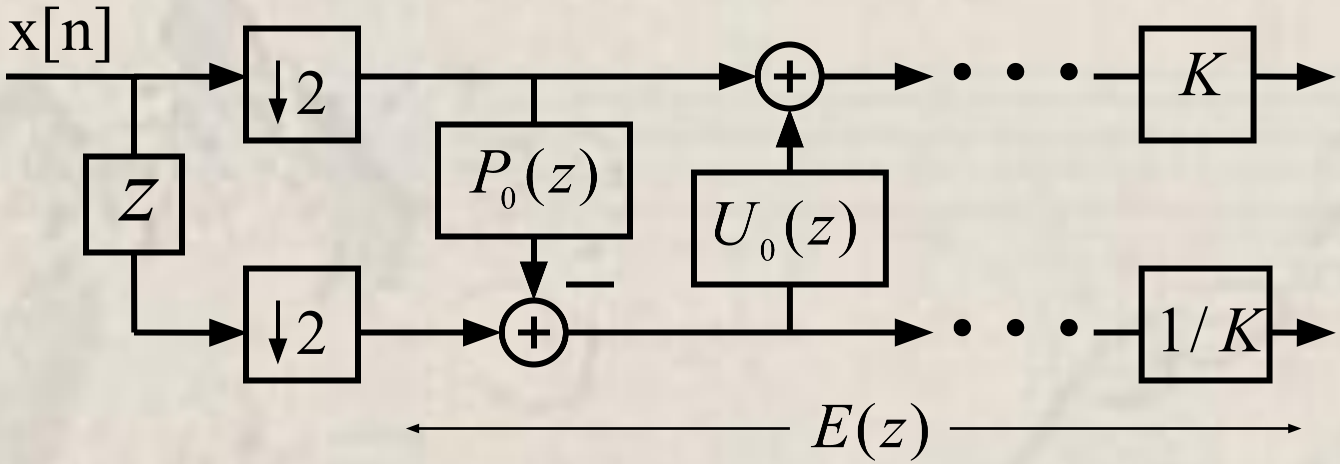
$$\min_{\{\theta_i\}} f\{\theta_i\}$$

stopband attenuation

regularity

combination

Lifting Scheme



$$E(z)=\begin{bmatrix}K & 0\\ 0 & 1/K\end{bmatrix}\prod_i\begin{bmatrix}1 & U_i(z)\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ -P_i(z) & 1\end{bmatrix}$$

Example:

$$E(z)=\begin{bmatrix}1 & \frac{1}{4}(1+z)\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ -\frac{1}{2}(1+z^{-1}) & 1\end{bmatrix}$$

$$h_0[n]=\left[-\frac{1}{8},\frac{1}{4},\frac{3}{4},\frac{1}{4},-\frac{1}{8}\right]$$

$$h_1[n]=\left[-\frac{1}{2},1,-\frac{1}{2}\right]$$

Wavelets: Summary

- Closed-form construction

$$\text{Mother Wavelet}: \Psi(t) \Rightarrow \begin{cases} \text{Time} - \text{Scaled}: & \Psi(2t) \\ \text{Translation}: & \Psi(t\text{-}k) \\ \text{Time} - \text{Scaled} + \text{Translation}: & \Psi(2t\text{-}k) \end{cases}$$

- Fundamental properties - Advantages
  - non-redundant orthonormal bases, perfect reconstruction
  - compact support is possible
  - basis functions with varying lengths with zoom capability
  - smooth approximation
  - fast *O(n)* algorithms
- Connection to other constructions
  - filter bank and sub-band coding in signal compression
  - multi-resolution in computer vision
  - multi-grid methods in numerical analysis
  - successive refinement in graphics