Design Procedure

1. Design a half-band Type I linear-phase filter P_0 with center of symmetry

$$P_0(z) - P_0(-z) = 2z^{-L}, \quad L \text{ is an odd integer}$$

2. Factorize $P_0(z)$ into $H_0(z)$ and $G_0(z)$

3. Define $H_1(z) = G_0(-z)$

$$G_1(z) = -H_0(-z)$$

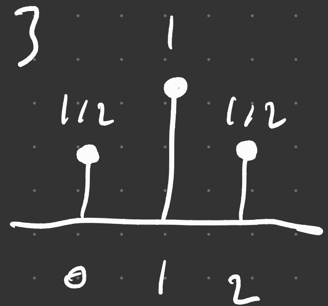
Q: Given $P_0(z)$, how do we distribute its zeros across $H_0(z)$ and $G_0(z)$?

Ex: What's the simplest half-band filter?

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

$$= \frac{1}{2}\left(1+2z^{-1}+z^{-2}\right) = \frac{1}{2} + z^{-1} + \frac{1}{2}z^{-2}$$

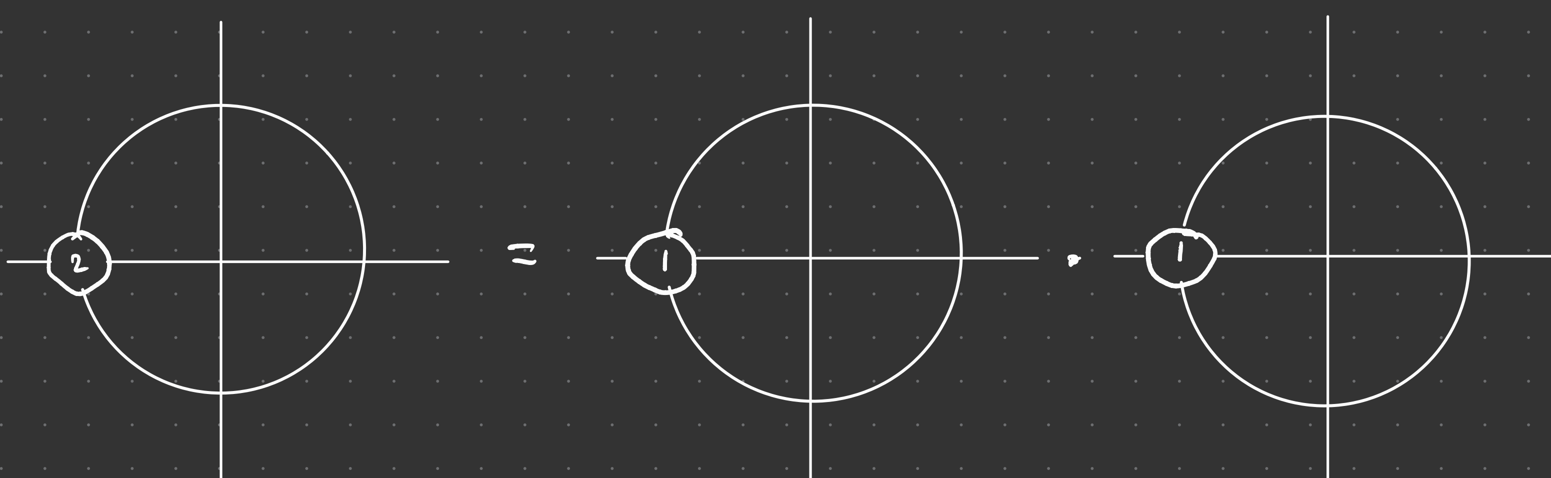
$p_0[n]$



2nd-order interp. filter

Type I linear-phase

$$P_0(z) = G_0(z) \cdot H_0(z)$$



$\frac{1}{2}(1+z^{-1})^2$ $\qquad$ $\frac{1}{\sqrt{2}}(1+z^{-1})$ $\qquad$ $\frac{1}{\sqrt{2}}(1+z^{-1})$

Obs: • $G_0(z) = H_0(z)$

• G_0 & H_0 are of the same order ($= 1$)

• $G_0(z) = z^{-1} H_0(z^{-1})$

$\longrightarrow$ orthogonal filter bank

$\longrightarrow$ orthogonal wavelets

• $H_1(z) = G_0(-z)$

$H_1(e^{j\omega}) = G_0\left(e^{j(\omega - \pi)}\right)$ shift in frequency

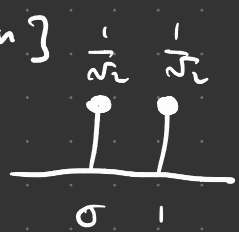
$h_1[n] = e^{j\pi n} g_0[n]$ modulation in time

$= (-1)^n g_0[n]$ flip odd coeffs

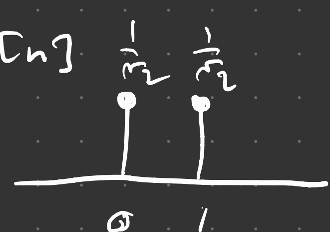
• $G_1(z) = -H_0(-z)$

$g_1[n] = (-1)^{n+1} h_0[n]$ flip even coeffs

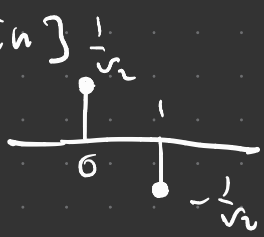
$h_0[n]$



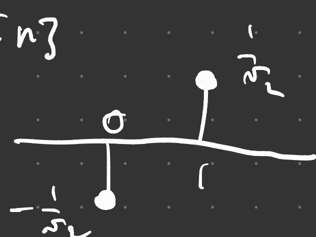
$g_0[n]$



$h_1[n]$



$g_1[n]$



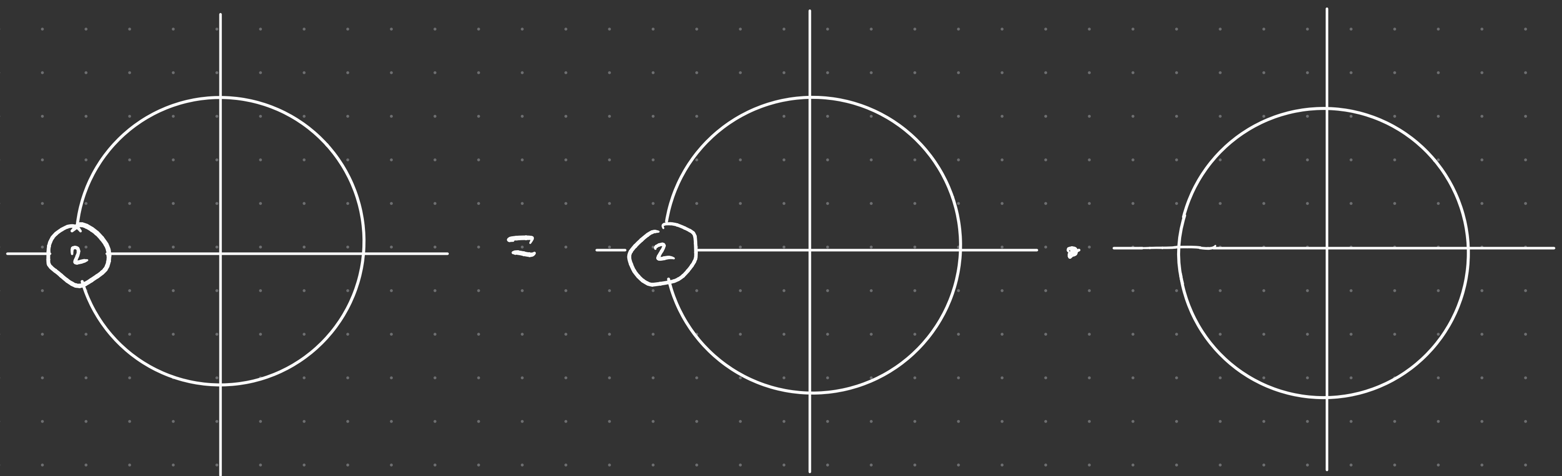
Haar wavelet filters

db(1)

Daubechies

Q: Are there other possible factorizations?

$$P_0(z) = G_0(z) \cdot H_0(z)$$



$\frac{1}{2}(1+z^{-1})^2$ $\qquad$ $\frac{1}{2}(1+z^{-1})^2$ $\qquad$ 1

or flip these

Q: What is the next possible order?

A: 6th-order (because of halfband cond.)

(not 4)

Obs: General rule is to add 4 to get the next valid order.

6th-order:

Remark: There are many different possible design choices.

Q: How do we get an orthogonal filter bank/wavelets?

A: $G_0(z) = z^{-3} H_0(z^{-1})$

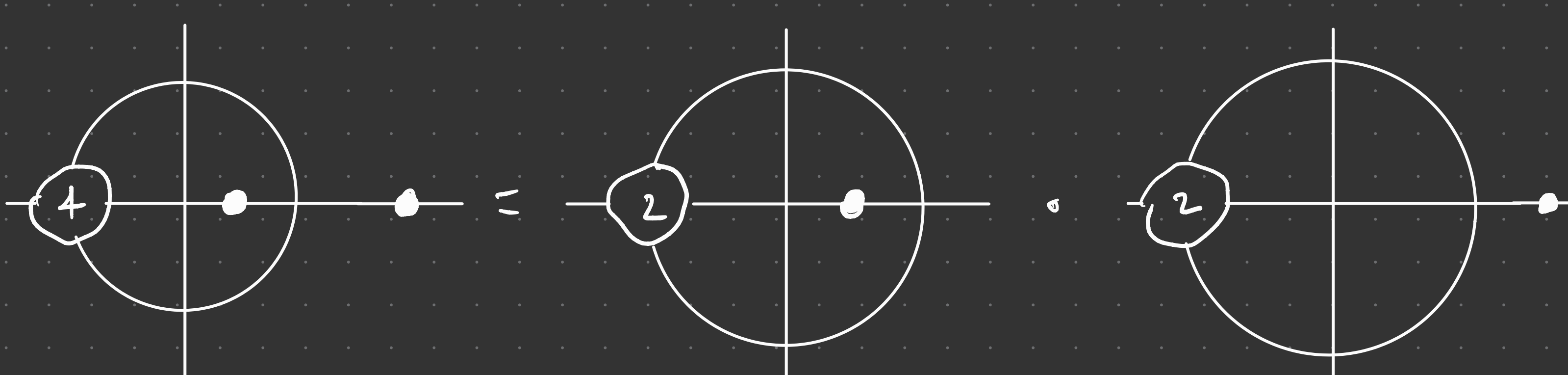
General: Orthogonality is guaranteed when

$$G_0(z) = z^{-L} H_0(z^{-1})$$ flipped zeros

Obs: G_0 and H_0 have the same order.

Ex: $\boxed{\text{db(2) wavelet filters}}$

$P_0(z) \quad = \quad G_0(z) \quad \cdot \quad H_0(z)$



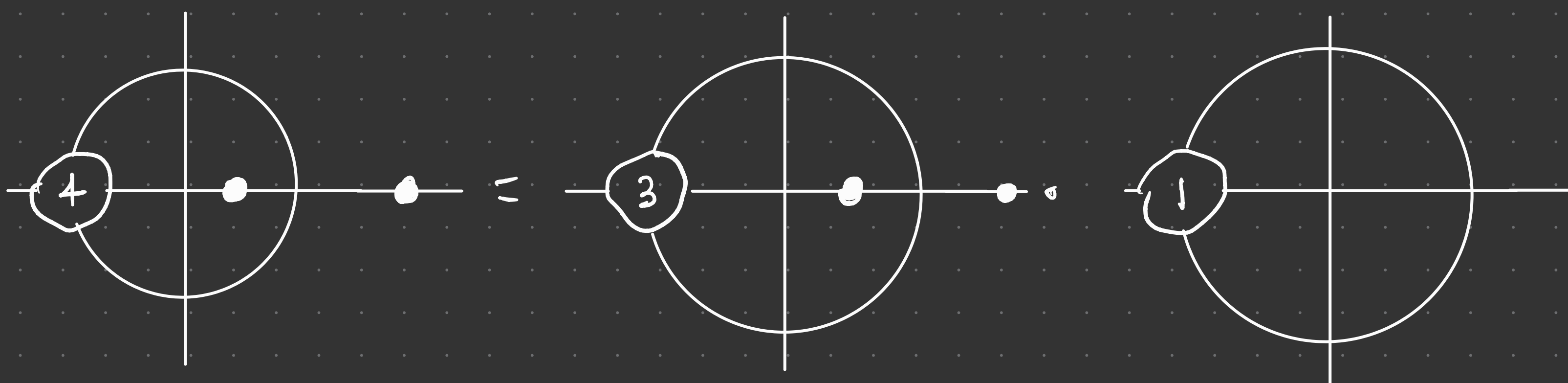
Type I linear-phase — not linear-phase — not linear-phase

Remark: Orthogonal filter banks cannot be linear-phase except for the Haar.

Note: In the example above, even though F_0 & H_0 are not linear-phase, the product filter $P_0(z) = G_0(z)\, H_0(z)$ is linear-phase.

Ex: $\boxed{\text{bior}(3,1)}$

$P_0(z) \quad = \quad G_0(z) \quad \cdot \quad H_0(z)$

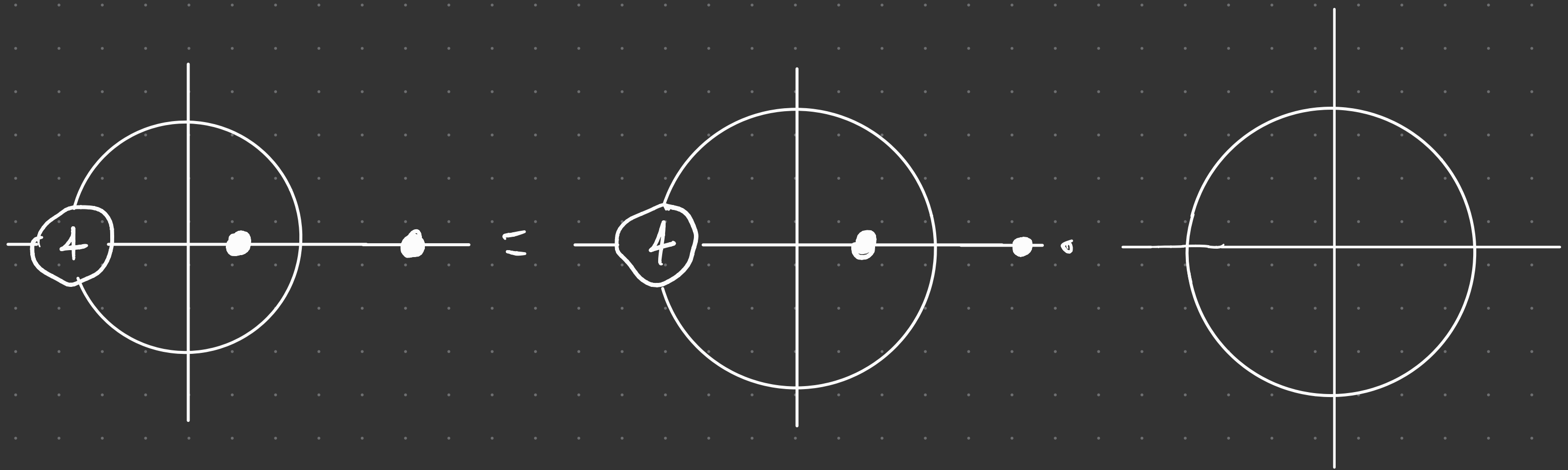


Type I linear-phase   Type II linear-phase   Type II linear-phase

Remark: This filter bank is called biorthogonal.

Ex:

$$P_0(z) = G_0(z) \cdot H_0(z)$$



or flip these

Remark: This kind of system is not very useful. The general rule of thumb is to balance the zeros as much as possible.

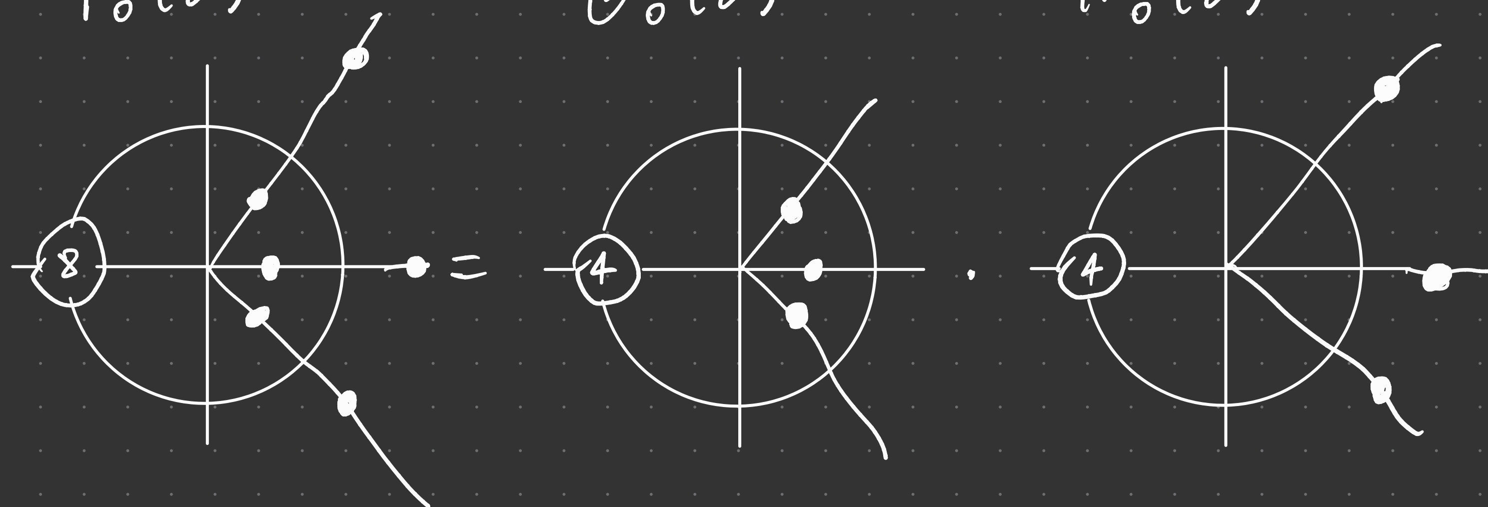
14th - order

To guarantee orthogonality, we want

$$G_0(z) = z^{-7} H_0(z^{-1})$$

Ex: $\boxed{db(4)}$

$P_0(z) \quad = \quad G_0(z) \quad \cdot \quad H_0(z)$



Type I linear-phase

not linear-phase
min-phase

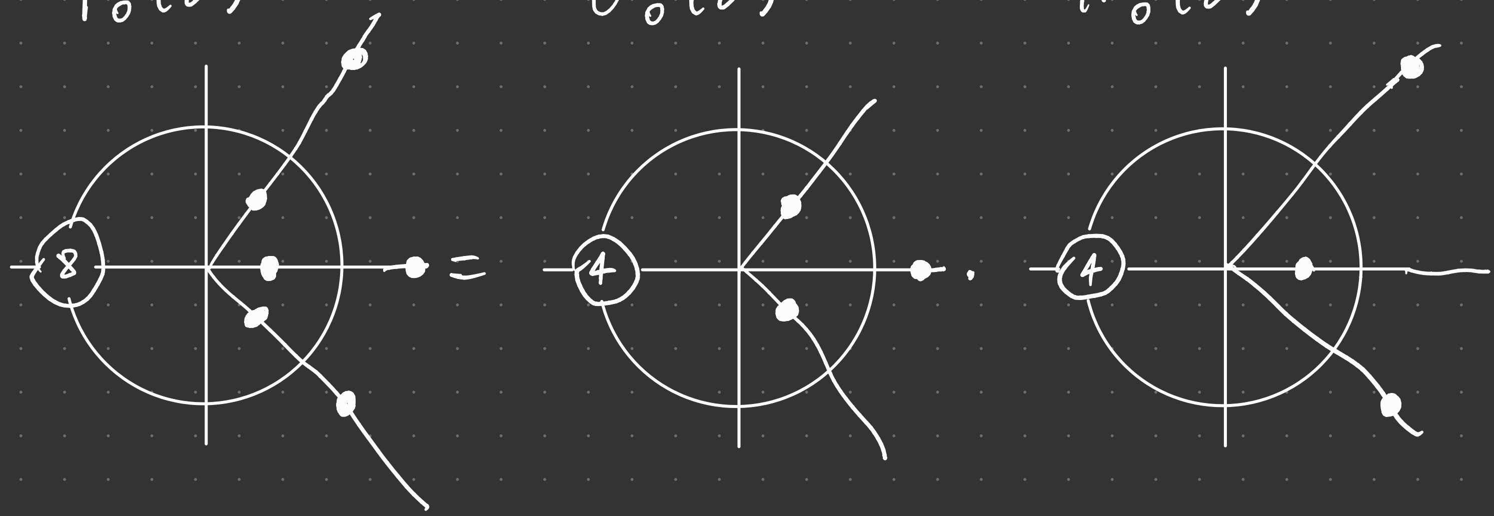
not linear-phase
max-phase

Obs: This factorization produces an orthogonal filter bank/wavelet system.

Obs: This factorization is <u>not</u> unique.

Ex: $\boxed{\text{sym}(4)}$

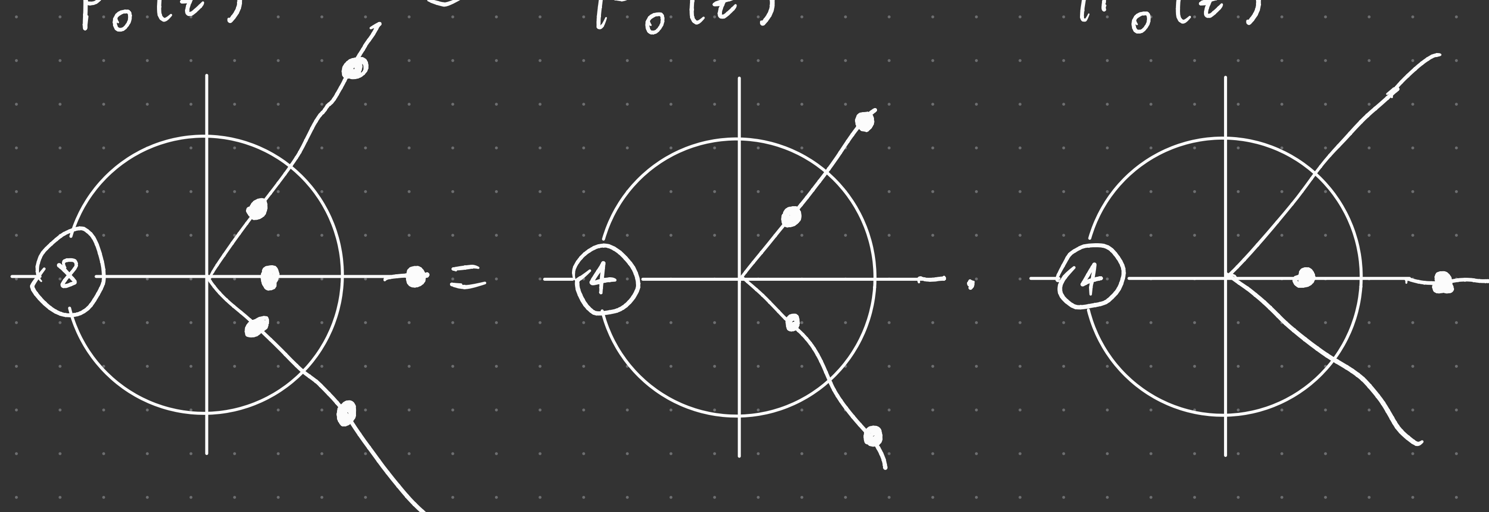
$$P_0(z) = G_0(z) \cdot H_0(z)$$



Obs: This factorization produces an orthogonal filter bank/wavelet system.

Ex: $\boxed{\text{bior}(4,4)}$ JPEG2000

$$P_0(z) = F_0(z) \cdot H_0(z)$$



Type I linear-phase   Type I linear-phase   Type I linear-phase

Obs: This factorization produces a biorthogonal FB.

Remark: More zeros @ π, the "smoother" the resulting filters are.

- Smoother wavelets
- Better signal-approximation properties.

Goal: People try to design filters with the maximum # of zeros @ π.

- Max-flat filters.

At this point we are "experts" at the design procedure for

- Perfect-Reconstruction
- FIR
- Two-Channel

filter banks with linear-phase product filters.

- Designing (bi)orthogonal wavelets