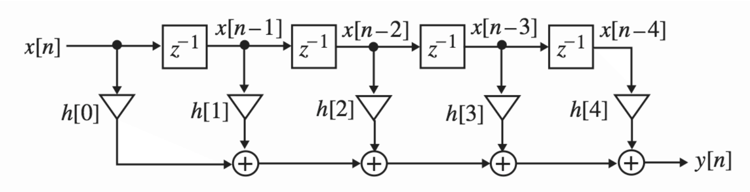
# Digital Signal Processing
# Basic FIR Realization Structures

D. Richard Brown III

# FIR Direct Form




This is called "direct form" because it is a direct implementation of the convolution operation. The number of delays is equal to the order of the filter, hence this structure is **canonic**.

It should be clear that

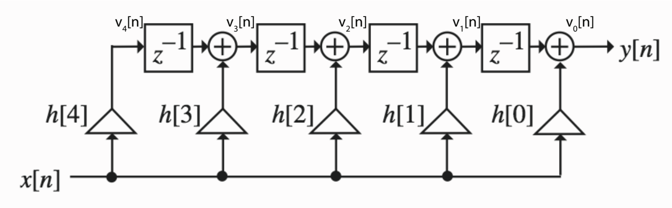
$$y[n] = h[0]x[n] + h[1]x[n-1] + \cdots + h[N]x[n-N] = \sum_{k=0}^{N} h[k]x[k-n]$$

# Pseudocode Implementation of FIR Direct Form

```
<shift input buffer to the right,
  x[n-N] = x[n-N+1]
  x[n-N] = x[n-N+2]
  ...
  x[n-1] = x[n]
>

<read in new input x[n]>

<zero out y[n]>

<multiply and accumulate,
  y[n] += h[0]*x[n]
  y[n] += h[1]*x[n-1]
  ...
  y[n] += h[N]*x[n-N]
>
```

# FIR Direct Form Transposed

If we perform this transpose operation on a direct form FIR filter, we get the "direct form transposed" FIR filter structure:




Also canonic. Note that

$$
\begin{aligned}
v_4[n] &= h[4]x[n] \\
v_3[n] &= v_4[n-1] + h[3]x[n] = h[4]x[n-1] + h[3]x[n] \\
v_2[n] &= v_3[n-1] + h[2]x[n] = h[4]x[n-2] + h[3]x[n-1] + h[2]x[n] \\
v_1[n] &= v_2[n-1] + h[1]x[n] = h[4]x[n-3] + h[3]x[n-2] + h[2]x[n-1] + h[1]x[n] \\
v_0[n] &= v_1[n-1] + h[0]x[n] = h[4]x[n-4] + h[3]x[n-3] + h[2]x[n-2] + h[1]x[n-1] + h[0]x[n] \\
y[n] &= v_0[n]
\end{aligned}
$$

which is identical to the standard (non-transposed) form.

# Pseudocode Implementation of FIR DF Transposed

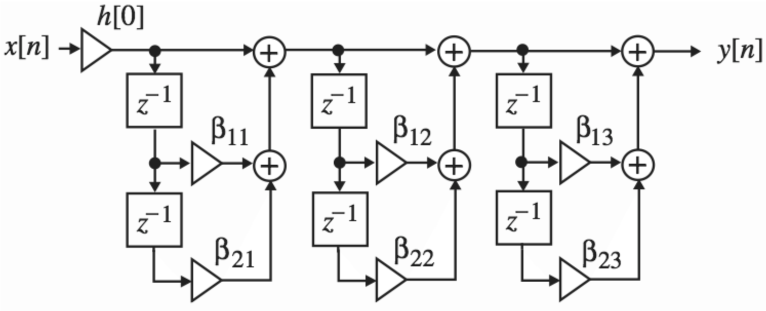
```
<read in new input x[n]>

<compute new output,
  v[0] = h[0]*x[n]+v[1]
  y[n] = v[0]
>

<update v-buffer,
  v[1] = h[1]*x[n]+v[2]
  ...
  v[N-1] = h[N-1]*x[n]+v[N]
  v[N] = h[N]*x[n]
>
```

# FIR Cascaded

Idea: Factor transfer function into product of smaller transfer functions.

Example:

$$\begin{aligned} H(z) &= h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} + h[4]z^{-4} + h[5]z^{-5} + h[6]z^{-6} \\ &= h[0](1+\beta_{1,1}z^{-1}+\beta_{2,1}z^{-2})(1+\beta_{1,2}z^{-1}+\beta_{2,2}z^{-2})(1+\beta_{1,3}z^{-1}+\beta_{2,3}z^{-2}) \end{aligned}$$

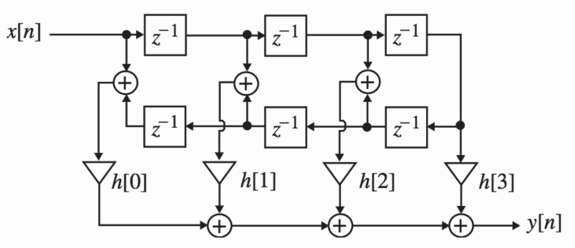


# Efficient Structures for Linear Phase FIR Filters

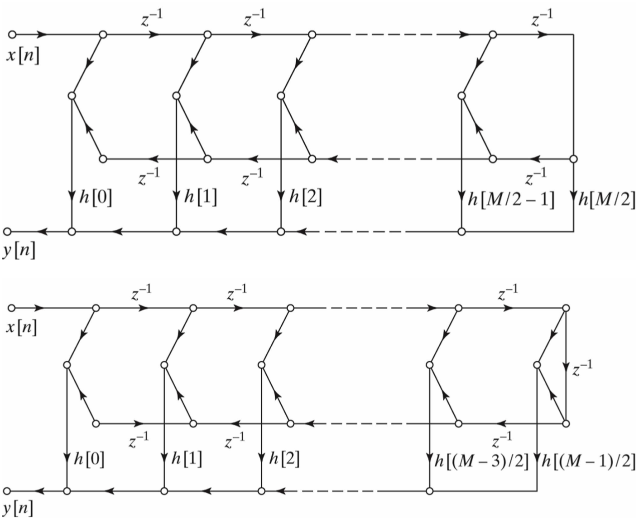
Recall the various types of linear phase FIR filters, each having either symmetric or anti-symmetric impulse response. For example (Type I):

$$\begin{aligned} h[n] =& h[0]\delta[n] + h[1]\delta[n-1] + h[2]\delta[n-2] \\ &+ h[3]\delta[n-3] + h[2]\delta[n-4] + h[1]\delta[n-5] + h[0]\delta[n-6] \\ =& h[0](\delta[n] + \delta[n-6]) + h[1](\delta[n-1] + \delta[n-5]) \\ &+ h[2](\delta[n-2] + \delta[n-4]) + h[3]\delta[n-3] \end{aligned}$$

This symmetry can be exploited to reduce the number of multipliers.



# Efficient Structures for Linear Phase FIR Filters




These structures can also be implemented in cascade form.