FIR filters

ELG6163

Miodrag Bolic

Outline

- FIR filters
  - Structures
  - Polyphase FIR filters
  - Parallel polyphase FIR
  - Decimated FIR
- Implementations of FIR filters

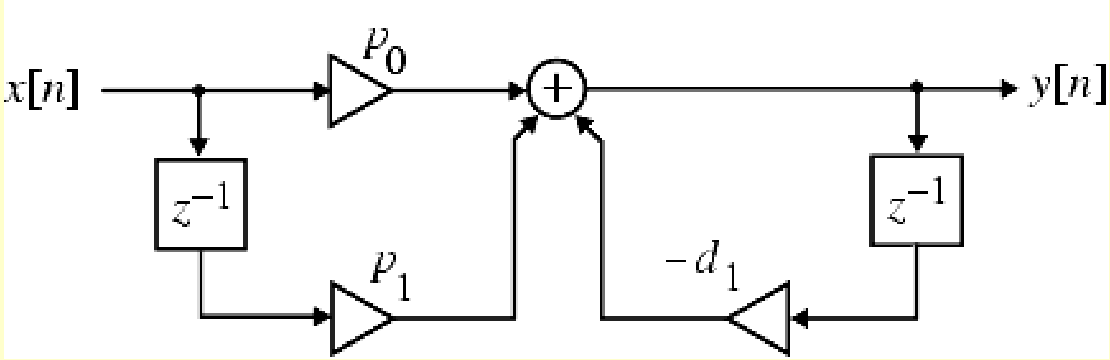
Canonic and Noncanonic Structures

- A digital filter structure is said to be ***canonic*** if the number of delays in the block diagram representation is equal to the order of the transfer function
- Otherwise, it is a ***noncanonic*** structure

Canonic and Noncanonic Structures

- The structure shown below is noncanonic as it employs two delays to realize a first-order difference equation

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



Basic 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

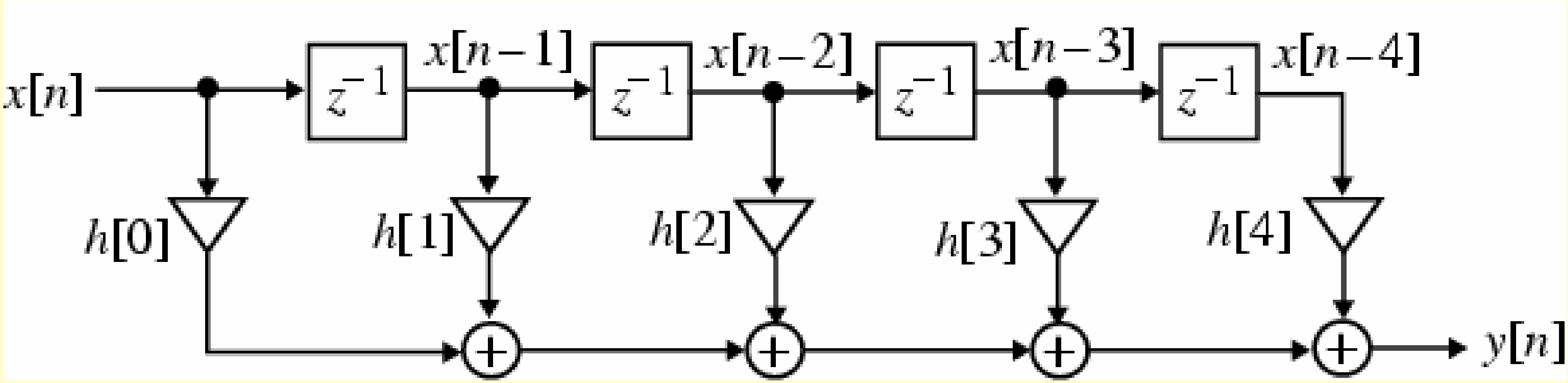
$$y[n] = \sum_{k=0}^{N} h[k] x[n-k]$$

Direct Form FIR Digital Filter Structures

- 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***

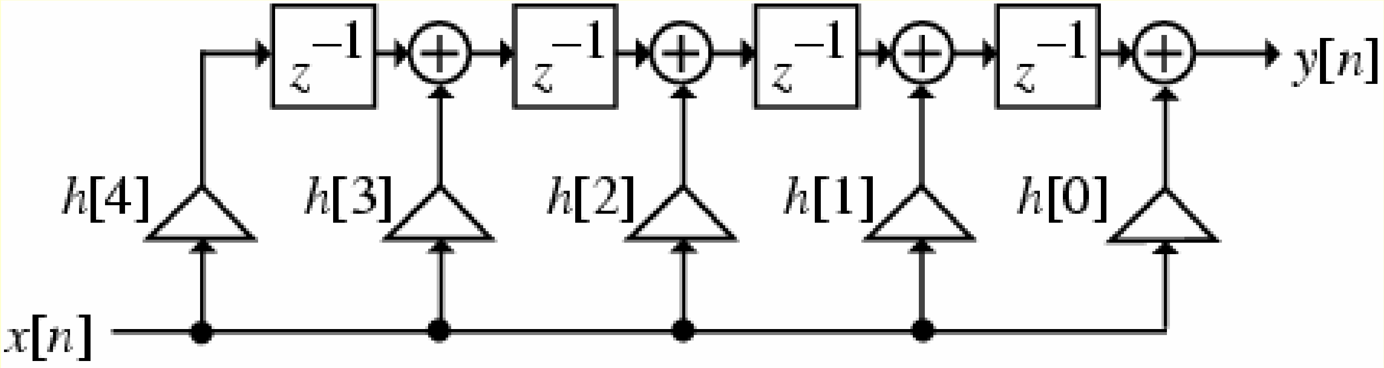
Direct Form FIR Digital Filter Structures

- A direct form realization of an FIR filter can be readily developed from the convolution sum description as indicated below for $N = 4$



Direct Form FIR Digital Filter Structures

- The transpose of the direct form structure shown earlier is indicated below



- Both direct form structures are canonic with respect to delays

Cascade Form FIR Digital Filter Structures

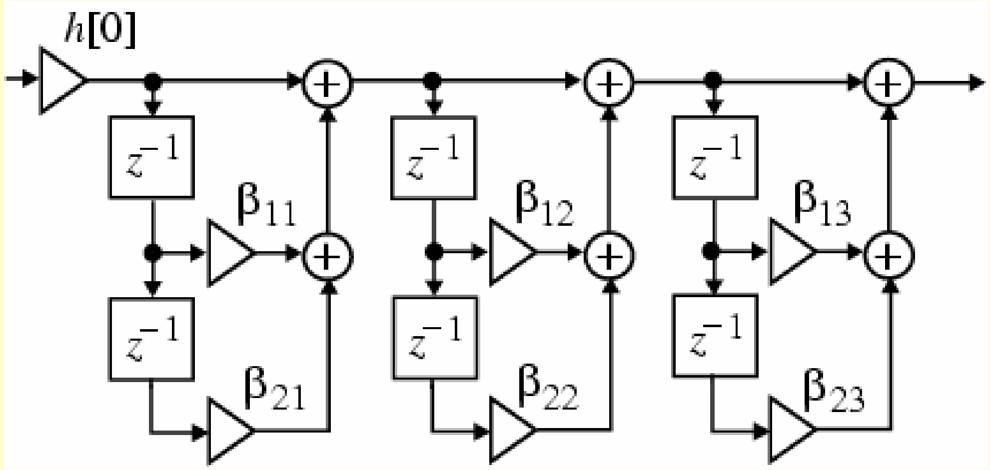
- A higher-order FIR transfer function can also be realized as a cascade of second-order FIR sections and possibly a first-order section
- To this end we express $H(z)$ as

$$H(z) = h[0]\prod_{k=1}^{K}(1+\beta_{1k}z^{-1}+\beta_{2k}z^{-2})$$

where $K = \frac{N}{2}$ if N is even, and $K = \frac{N+1}{2}$ if N is odd, with $\beta_{2K} = 0$

Cascade Form FIR Digital Filter Structures

- 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

Linear-Phase FIR Structures

- The symmetry (or antisymmetry) property of a linear-phase FIR filter can be exploited to reduce the number of multipliers into almost half of that in the direct form implementations
- Consider a length-7 Type 1 FIR transfer function with a symmetric impulse response:

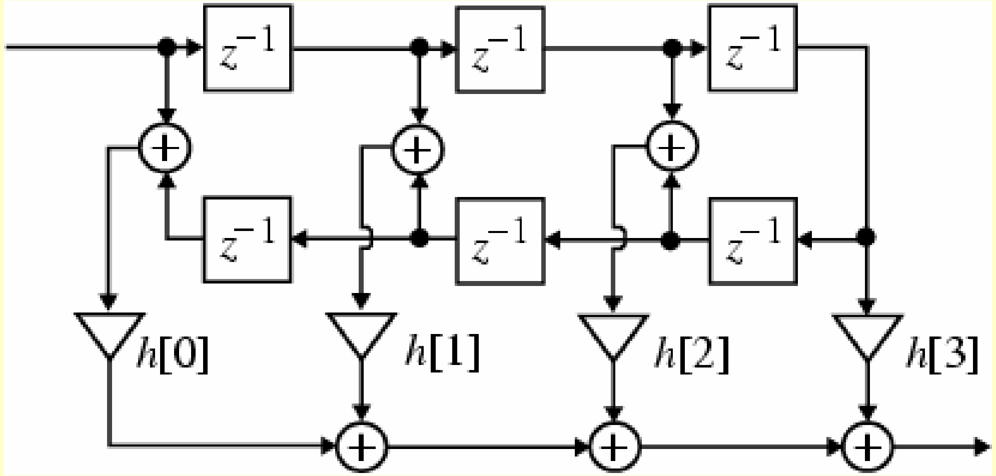
$$H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3}$$
$$+ h[2]z^{-4} + h[1]z^{-5} + h[0]z^{-6}$$

Linear-Phase FIR Structures

- Rewriting $H(z)$ in the form

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

we obtain the realization shown below



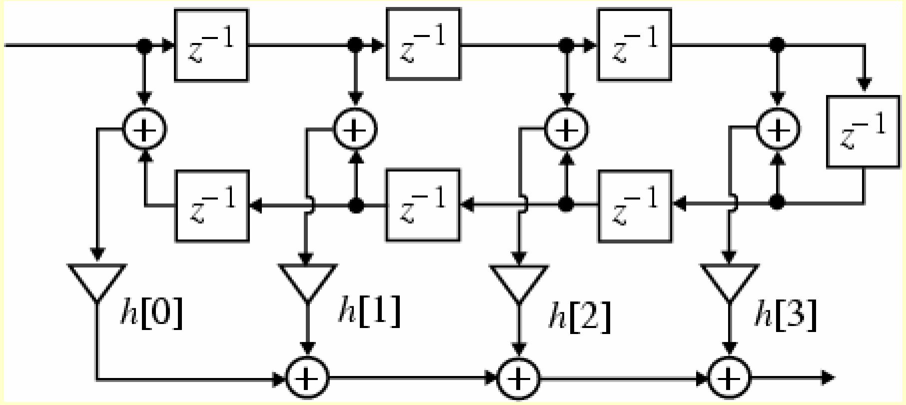
Linear-Phase FIR Structures

- A similar decomposition can be applied to a Type 2 FIR transfer function
- For example, a length-8 Type 2 FIR transfer function can be expressed as

$$H(z) = h[0](1 + z^{-7}) + h[1](z^{-1} + z^{-6})$$
$$+ h[2](z^{-2} + z^{-5}) + h[3](z^{-3} + z^{-4})$$

- The corresponding realization is shown on the next slide

Linear-Phase FIR Structures



- Note: The Type 1 linear-phase structure for a length-7 FIR filter requires 4 multipliers, whereas a direct form realization requires 7 multipliers

Linear-Phase FIR Structures

- Note: The Type 2 linear-phase structure for a length-8 FIR filter requires 4 multipliers, whereas a direct form realization requires 8 multipliers
- Similar savings occurs in the realization of Type 3 and Type 4 linear-phase FIR filters with antisymmetric impulse responses

Polyphase FIR Structures

- The polyphase decomposition of $H(z)$ leads to a parallel form structure
- To illustrate this approach, consider a causal FIR transfer function $H(z)$ with $N = 8$:

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

$$+ h[5]z^{-5} + h[6]z^{-6} + h[7]z^{-7} + h[8]z^{-8}$$

Polyphase FIR Structures

- $H(z)$ 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:

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

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

Polyphase FIR Structures

- By using the notation

$$E_0(z) = h[0] + h[2]z^{-1} + h[4]z^{-2} + h[6]z^{-3} + h[8]z^{-4}$$

$$E_1(z) = h[1] + h[3]z^{-1} + h[5]z^{-2} + h[7]z^{-3}$$

we can express $H(z)$ as

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

Polyphase FIR Structures

- In a similar manner, by grouping the terms in the original expression for $H(z)$, we can re-express it in the form

$$H(z) = E_0(z^3) + z^{-1}E_1(z^3) + z^{-2}E_2(z^3)$$

where now

$$E_0(z) = h[0] + h[3]z^{-1} + h[6]z^{-2}$$

$$E_1(z) = h[1] + h[4]z^{-1} + h[7]z^{-2}$$

$$E_2(z) = h[2] + h[5]z^{-1} + h[8]z^{-2}$$

Polyphase FIR Structures

- The decomposition of $H(z)$ in the form

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

or

$$H(z) = E_0(z^3) + z^{-1} E_1(z^3) + z^{-2} E_2(z^3)$$

is more commonly known as the ***polyphase decomposition***

Polyphase FIR Structures

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

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

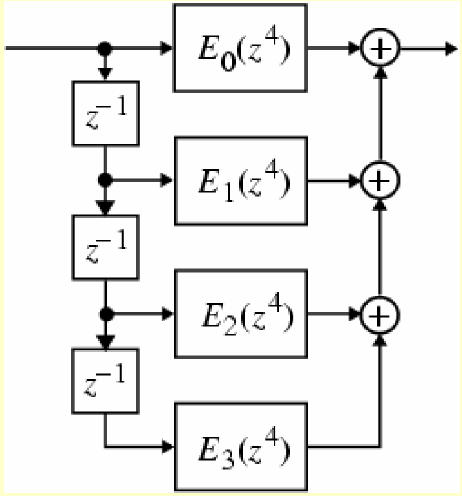
where

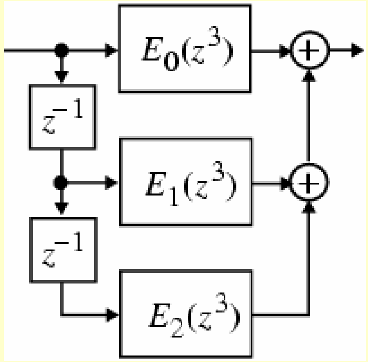
$$E_m(z) = \sum_{n=0}^{\lfloor (N+1)/L \rfloor} h[Ln+m] z^{-m}$$

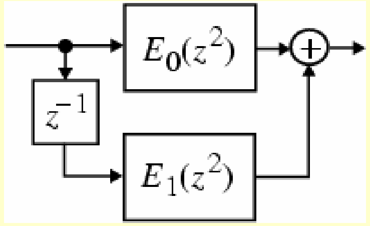
with $h[n]=0$ for $n > N$

Polyphase FIR Structures

- Figures below show the ***4-branch, 3-branch, and 2-branch polyphase realization*** of a transfer function $H(z)$







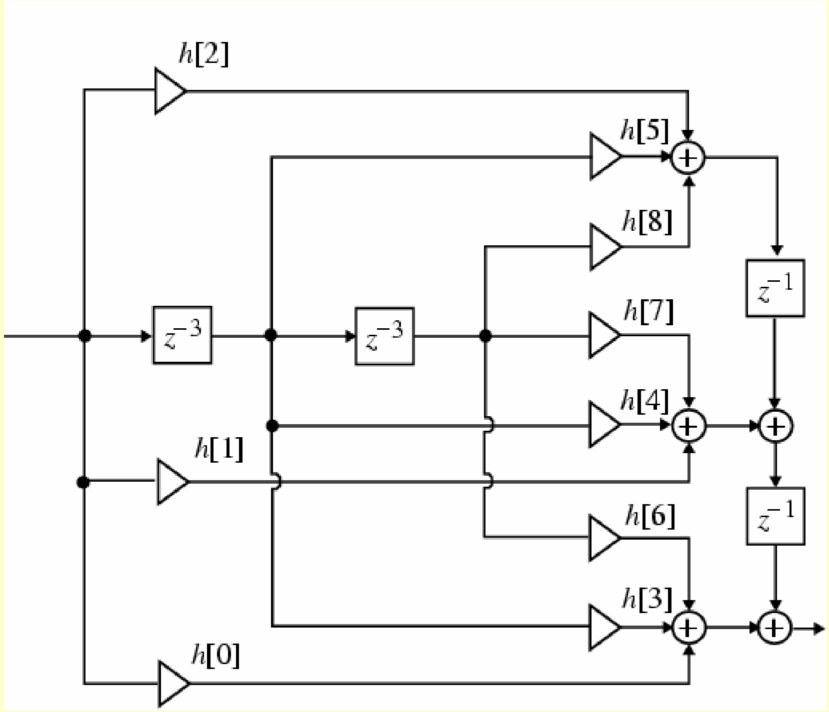
- Note: The expression for the polyphase components $E_m(z)$ are different in each case

Polyphase FIR Structures

- The subfilters $E_m(z^L)$ in the polyphase realization of an FIR transfer function are also FIR filters and can be realized using any methods described so far
- However, to obtain a canonic realization of the overall structure, the delays in all subfilters must be shared

Polyphase FIR Structures

- Figure below shows a canonic realization of a length-9 FIR transfer function obtained using delay sharing



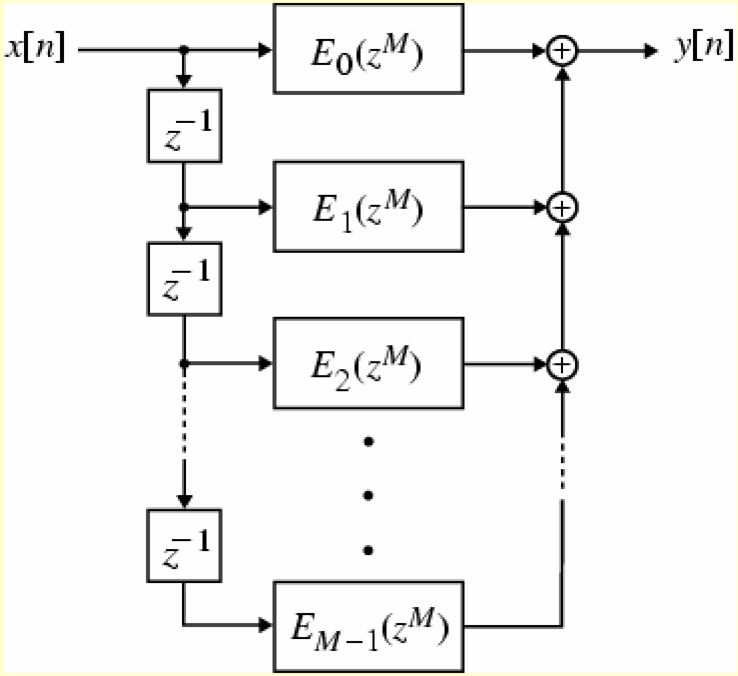
FIR Filter Structures Based on Polyphase Decomposition

- We shall demonstrate later that a parallel realization of an FIR transfer function $H(z)$ based on the polyphase decomposition can often result in computationally efficient multirate structures
- Consider the M-branch Type I polyphase decomposition of $H(z)$:

$$H(z) = \sum_{k=0}^{M-1} z^{-k} E_k(z^M)$$

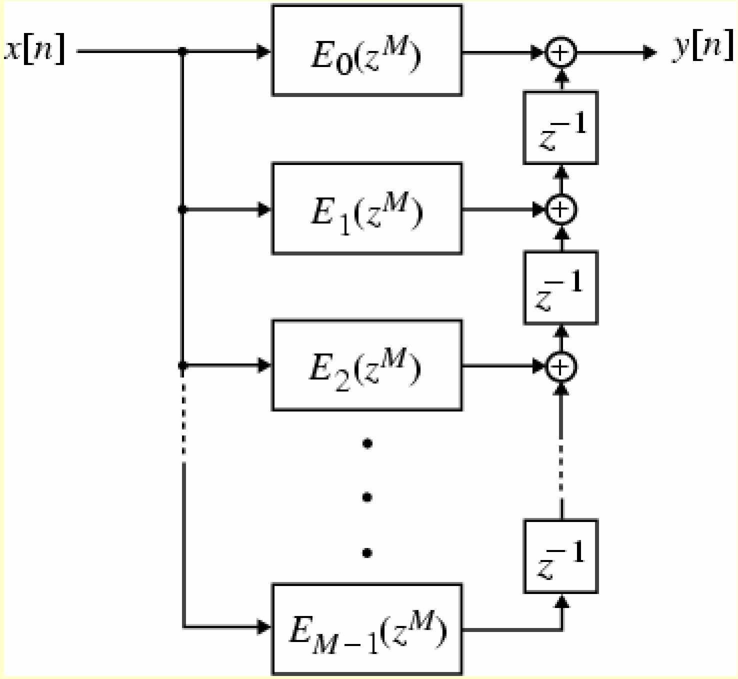
FIR Filter Structures Based on Polyphase Decomposition

- A direct realization of $H(z)$ based on the Type I polyphase decomposition is shown below



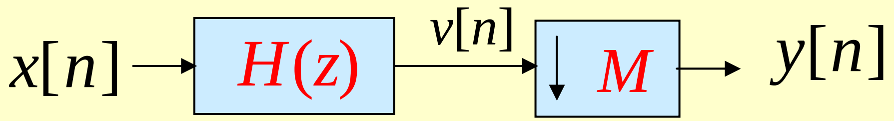
FIR Filter Structures Based on Polyphase Decomposition

- The transpose of the Type I polyphase FIR filter structure is indicated below



Computationally Efficient Decimators

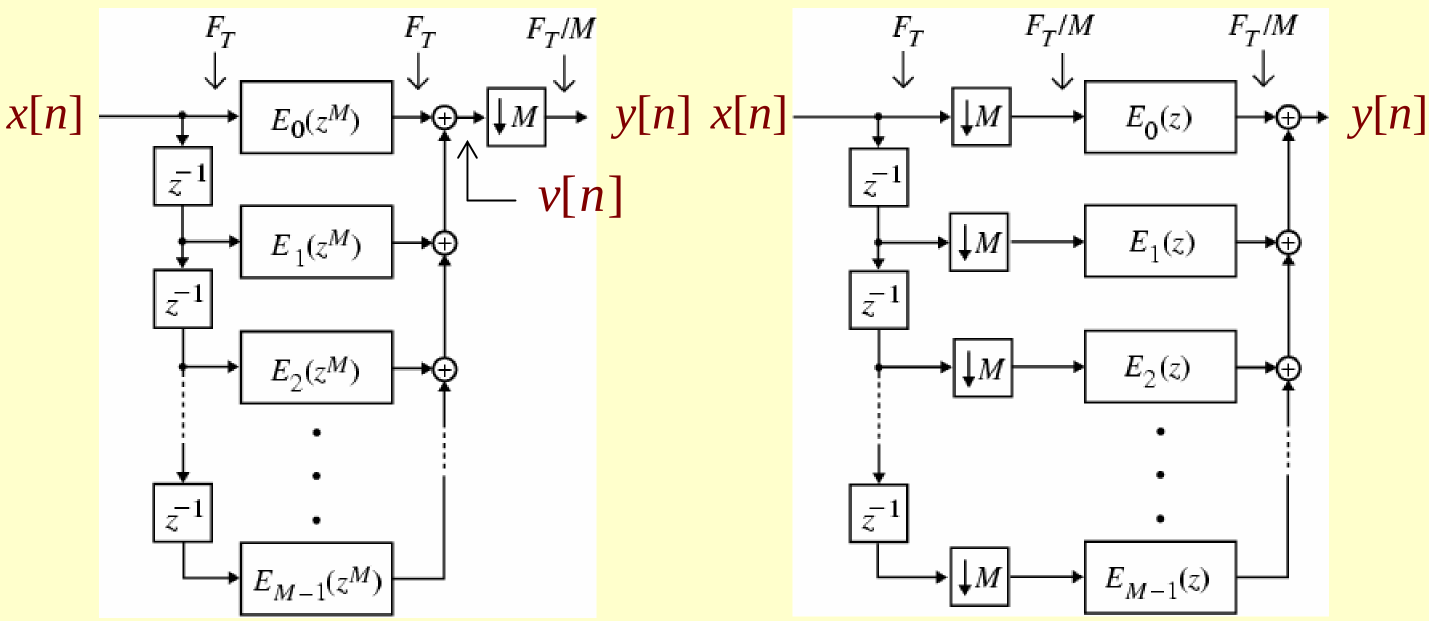
- Consider first the single-stage factor-of-M decimator structure shown below



- We realize the lowpass filter $H(z)$ using the Type I polyphase structure as shown on the next slide

Computationally Efficient Decimators

- Using the cascade equivalence #1 we arrive at the computationally efficient decimator structure shown below on the right



Decimator structure based on Type I polyphase decomposition

Computationally Efficient Decimators

- To illustrate the computational efficiency of the modified decimator structure, assume $H(z)$ to be a length-N structure and the input sampling period to be $T = 1$
- Now the decimator output $y[n]$ in the original structure is obtained by down-sampling the filter output $v[n]$ by a factor of M

Computationally Efficient Decimators

- It is thus necessary to compute $v[n]$ at

$$n = \ldots, -2M, -M, 0, M, 2M, \ldots$$

- Computational requirements are therefore N multiplications and $(N-1)$ additions per output sample being computed
- However, as n increases, stored signals in the delay registers change

Computationally Efficient Decimators

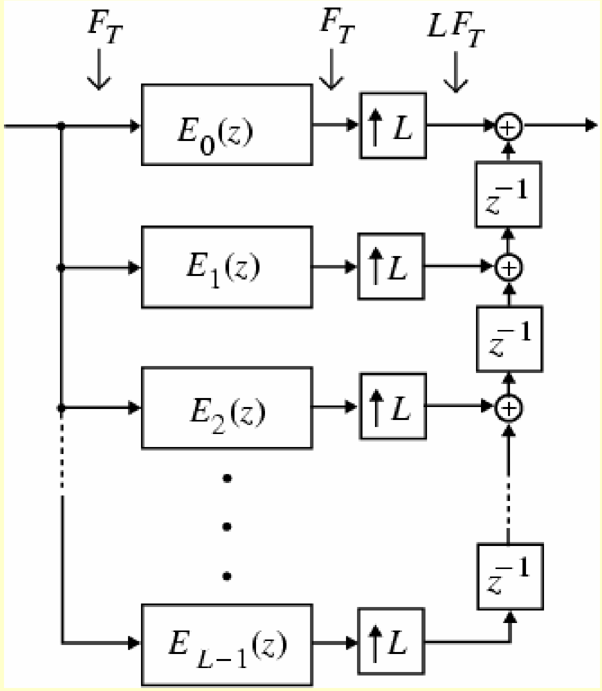
- Hence, all computations need to be completed in one sampling period, and for the following $(M-1)$ sampling periods the arithmetic units remain idle
- The modified decimator structure also requires N multiplications and $(N-1)$ additions per output sample being computed

Computationally Efficient Decimators and Interpolators

- However, here the arithmetic units are operative at all instants of the output sampling period which is $1/M$ times that of the input sampling period
- Similar savings are also obtained in the case of the interpolator structure developed using the polyphase decomposition

Computationally Efficient Interpolators

- Figures below show the computationally efficient interpolator structures



Interpolator based on
Type I polyphase decomposition

Computationally Efficient Decimators and Interpolators

- More efficient interpolator and decimator structures can be realized by exploiting the symmetry of filter coefficients in the case of linear-phase filters $H(z)$
- Consider for example the realization of a factor-of-3 ($M = 3$) decimator using a length-12 Type 1 linear-phase FIR lowpass filter

Computationally Efficient Decimators and Interpolators

- The corresponding transfer function is

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

$$+ h[5]z^{-6} + h[4]z^{-7} + h[3]z^{-8} + h[2]z^{-9} + h[1]z^{-10} + h[0]z^{-11}$$

- A conventional polyphase decomposition of $H(z)$ yields the following subfilters:

$$E_0(z) = h[0] + h[3]z^{-1} + h[5]z^{-2} + h[2]z^{-3}$$

$$E_1(z) = h[1] + h[4]z^{-1} + h[4]z^{-2} + h[1]z^{-3}$$

$$E_2(z) = h[2] + h[5]z^{-1} + h[3]z^{-2} + h[0]z^{-3}$$

Computationally Efficient Decimators and Interpolators

- Note that $E_1(z)$ still has a symmetric impulse response, whereas $E_0(z)$ is the mirror image of $E_2(z)$
- These relations can be made use of in developing a computationally efficient realization using only 6 multipliers and 11 two-input adders as shown on the next slide

Computationally Efficient Decimators and Interpolators

- Factor-of-3 decimator with a linear-phase decimation filter