Polyphase Decomposition

The Decomposition

- Consider an arbitrary sequence $\{x[n]\}$ with a z-transform $X(z)$ given by

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

- We can rewrite $X(z)$ as

$$X(z) = \sum_{k=0}^{M-1} z^{-k} X_k(z^M)$$

where

$$X_k(z) = \sum_{n=-\infty}^{\infty} x_k[n] z^{-n} = \sum_{n=-\infty}^{\infty} x[Mn+k] z^{-n}$$

$$0 \le k \le M-1$$

Polyphase Decomposition

- The subsequences $\{x_k[n]\}$ are called the **polyphase components** of the parent sequence $\{x[n]\}$
- The functions $X_k(z)$, given by the z-transforms of $\{x_k[n]\}$, are called the **polyphase components** of $X(z)$

Polyphase Decomposition

- The relation between the subsequences $\{x_k[n]\}$ and the original sequence $\{x[n]\}$ are given by

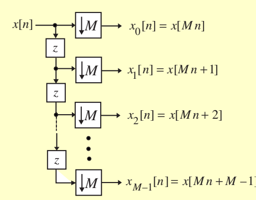
$$x_k[n] = x[Mn+k], \quad 0 \le k \le M-1$$

- In matrix form we can write

$$X(z) = \begin{bmatrix} 1 & z^{-1} & \cdots & z^{-(M-1)} \end{bmatrix} \begin{bmatrix} X_0(z^M) \\ X_1(z^M) \\ \vdots \\ X_{M-1}(z^M) \end{bmatrix}$$

Polyphase Decomposition

- A multirate structural interpretation of the polyphase decomposition is given below



Polyphase Decomposition

- The polyphase decomposition of an FIR transfer function can be carried out by inspection
- For example, consider a length-9 FIR transfer function:

$$H(z) = \sum_{n=0}^{8} h[n] z^{-n}$$

Polyphase Decomposition

- Its 4-branch polyphase decomposition is given by

$$H(z) = E_0(z^4) + z^{-1}E_1(z^4) + z^{-2}E_2(z^4) + z^{-3}E_3(z^4)$$

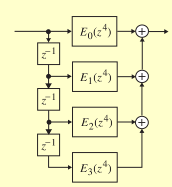
where

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

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

$$E_2(z) = h[2] + h[6]z^{-1}$$

$$E_3(z) = h[3] + h[7]z^{-1}$$



Polyphase Decomposition

- The polyphase decomposition of an IIR transfer function $H(z) = P(z)/D(z)$ is not that straight forward
- One way to arrive at an M-branch polyphase decomposition of $H(z)$ is to express it in the form $P'(z)/D'(z^M)$ by multiplying $P(z)$ and $D(z)$ with an appropriately chosen polynomial and then apply an M-branch polyphase decomposition to $P'(z)$

Polyphase Decomposition

- Example - Consider
$$H(z) = \frac{1-2z^{-1}}{1+3z^{-1}}$$
- To obtain a 2-band polyphase decomposition we rewrite $H(z)$ as
$$H(z) = \frac{(1-2z^{-1})(1-3z^{-1})}{(1+3z^{-1})(1-3z^{-1})} = \frac{1-5z^{-1}+6z^{-2}}{1-9z^{-2}} = \frac{1+6z^{-2}}{1-9z^{-2}} + \frac{-5z^{-1}}{1-9z^{-2}}$$
- Therefore,
$$H(z) = E_0(z^2) + z^{-1}E_1(z^2)$$

where
$$E_0(z) = \frac{1+6z^{-1}}{1-9z^{-1}}, \quad E_1(z) = \frac{-5}{1-9z^{-1}}$$

Polyphase Decomposition

- Note: The above approach increases the overall order and complexity of $H(z)$
- However, when used in certain multirate structures, the approach may result in a more computationally efficient structure
- An alternative more attractive approach is discussed in the following example

Polyphase Decomposition

- Example - Consider the transfer function of a 5-th order Butterworth lowpass filter with a 3-dB cutoff frequency at 0.5π:
$$H(z) = \frac{0.0527864(1+z^{-1})^5}{1+0.633436854z^{-1}+0.0557281z^{-2}}$$
- It is easy to show that $H(z)$ can be expressed as
$$H(z) = \frac{1}{2}\left[\left(\frac{0.105573+z^{-2}}{1+0.105573z^{-2}}\right) + z^{-1}\left(\frac{0.52786+z^{-2}}{1+0.52786z^{-2}}\right)\right]$$

Polyphase Decomposition

- Therefore $H(z)$ can be expressed as
$$H(z) = E_0(z^2) + z^{-1}E_1(z^2)$$

where
$$E_0(z) = \frac{1}{2}\left(\frac{0.105573+z^{-1}}{1+0.105573z^{-1}}\right)$$
$$E_1(z) = \frac{1}{2}\left(\frac{0.52786+z^{-1}}{1+0.52786z^{-1}}\right)$$

Polyphase Decomposition

- Note: In the above polyphase decomposition, branch transfer functions $E_i(z)$ are stable allpass functions
- Moreover, the decomposition has not increased the order of the overall transfer function $H(z)$

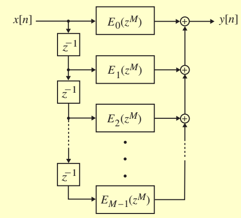
FIR Filter Structures Based on Polyphase Decomposition

- 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 $H(z)$:

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

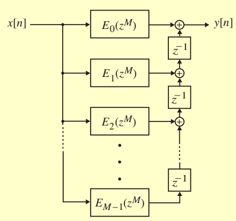
FIR Filter Structures Based on Polyphase Decomposition

- 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



FIR Filter Structures Based on Polyphase Decomposition

- An alternative representation of the transpose structure shown on the previous slide is obtained using the notation

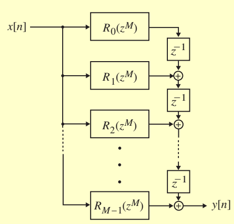
$$R_\ell(z^M) = E_{M-1-\ell}(z^M), \qquad 0 \le \ell \le M-1$$

- Substituting the above notation in the Type I polyphase decomposition we arrive at the Type II polyphase decomposition:

$$H(z) = \sum_{\ell=0}^{M-1} z^{-(M-1-\ell)} R_\ell(z^M)$$

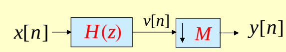
FIR Filter Structures Based on Polyphase Decomposition

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



Computationally Efficient Decimators

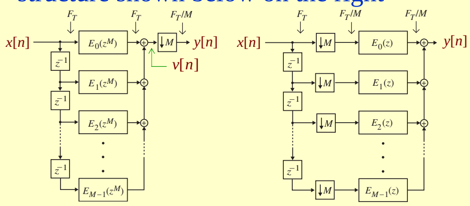
- Consider first the single-stage factor-of-M decimator structure shown below



- 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

19

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

20

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

21

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

22

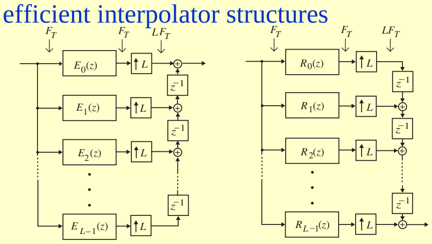
Computationally Efficient Decimators and Interpolators

- 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

23

Computationally Efficient Interpolators

- Figures below show the computationally efficient interpolator structures



Interpolator based on Type I polyphase decomposition

Interpolator based on Type II polyphase decomposition

24

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

25

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:

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

26

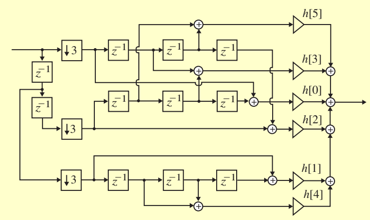
Computationally Efficient Decimators and Interpolators

- 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

27

Computationally Efficient Decimators and Interpolators

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



28

Rational Sampling Rate Converter

- The complexity of the design of the fractional sampling rate converter depends on the ratio of the sampling rates between the input and the output digital signals
- For example, in digital audio applications, the three different sampling frequencies employed are 44.1 kHz, 32 kHz, and 48 kHz

29

Rational Sampling Rate Converter

- As a consequence there are three different values for the sampling rate conversion factor:
- 2:3 (or 3:2), 147:160 (or 160:147), and 320:441 (or 441:320)
- Likewise, in digital video applications, the sampling rates of composite video signals are 14.3181818 MHz and 17.734475 MHz

30

Rational Sampling Rate Converter

- The sampling rates of the digital component video signal are 13.5 MHz and 6.75 MHz for the luminance and the color-difference signals, respectively, for the NTSC and PAL systems
- Here, again there are different sampling rate conversion factors

31

Rational Sampling Rate Converter

- We outline next the implementation of a computationally efficient FIR fractional rate converter with a rational conversion factor L/M, where L and M are mutually prime
- Now two mutually prime integers L and M satisfy the relation

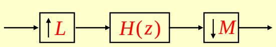
$$\mu M - \lambda L = 1$$

where μ and λ are unique distinct positive integers

32

Rational Sampling Rate Converter

- The general structure for a rational sampling rate converter shown below



can be made computationally efficient by making use of one of the structures based on the polyphase decompositions

33

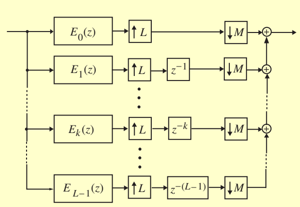
Rational Sampling Rate Converter

- Without any loss of generality, assume $L < M$
- The structure for the case $L > M$ can be derived by applying the transpose operation
- To develop the structure for the case $L < M$, we first replace the cascade of the factor-of-L up-sampler and the filter $H(z)$ with its equivalent Type I polyphase decomposition-based realization

34

Rational Sampling Rate Converter

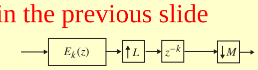
with the factor-of-M down-sampler moved to all L branches as shown below



35

Rational Sampling Rate Converter

- Consider the k-th branch of the structure shown in the previous slide



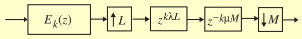
- Using the identity $\mu M - \lambda L = 1$ we can write

$$z^{-k} = z^{-k(\mu M - \lambda L)} = z^{-k\mu M} - z^{k\lambda L}$$

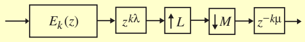
- Hence, we can replace the block of k delays with a block of $k\mu M$ unit delays and a block of $k\lambda L$ unit advances as shown next

36

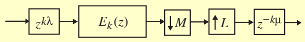
Rational Sampling Rate Converter



- This branch can be further redrawn by invoking the noble identites as shown below



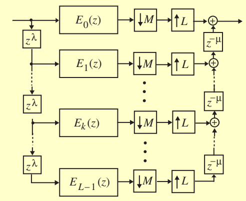
- We next interchange the positions of the up-sampler and the down-sampler



37

Rational Sampling Rate Converter

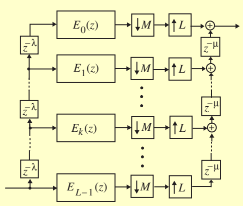
- As a result, the general rational sampling rate converter structure in Slide 36 can be redrawn as indicated below



38

Rational Sampling Rate Converter

- Its equivalent realizable form is as shown below



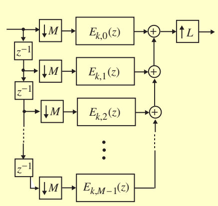
39

Rational Sampling Rate Converter

- Next, the cascade of the polyphase section followed by the down-sampler can be replaced with a computationally efficient realization based on a Type I polyphase decomposition as indicated in the next slide

40

Rational Sampling Rate Converter

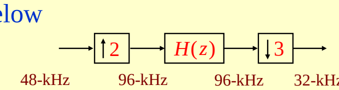


- Finally by combining all k branches we arrive at a computationally efficient rational sampling rate converter

41

Rational Sampling Rate Converter

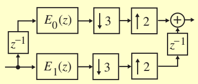
- Example – The basic form of a rational sampling rate converter with an interpolation factor 2/3 needed in the conversion of a digital audio signal of 48-kHz rate to one of 32-kHz rate is shown below



42

Rational Sampling Rate Converter

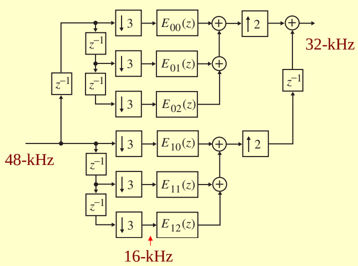
- For this design we have $L = 2$ and $M = 3$
- The identity $3\mu - 2\lambda = 1$ is thus satisfied with $\mu = \lambda = 1$
- Hence, the general structure of this converter is as indicated below



Rational Sampling Rate Converter

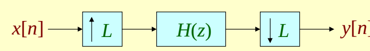
- By realizing the sub-filters $E_0(z)$ and $E_1(z)$ in Type I polyphase forms and then applying the cascade equivalence we arrive at the final computationally efficient structure shown in the next slide
- Here, all filters operate at the 16-kHz rate

Rational Sampling Rate Converter



A Useful Identity

- The cascade multirate structure shown below appears in a number of applications



- Equivalent time-invariant digital filter obtained by expressing $H(z)$ in its L-term Type I polyphase form $\sum_{k=0}^{L-1} z^{-k} E_k(z^L)$ is shown below