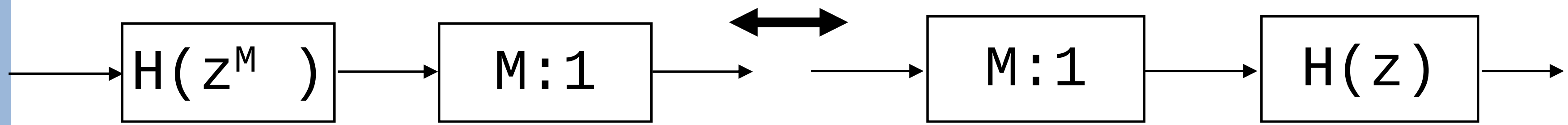
Multirate Filtering, Resampling Filters, Polyphase Filters

or how to make efficient FIR filters

THE NOBLE IDENTITY 1

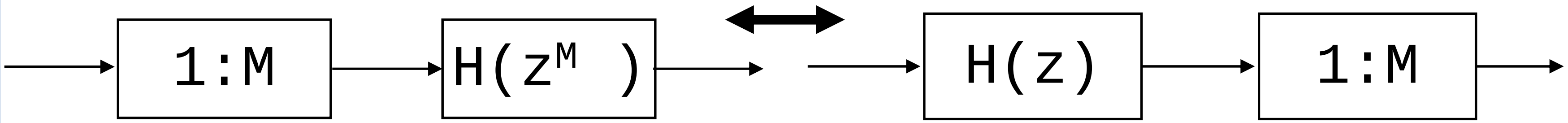
Efficient Implementation of Resampling filters



Rule 1: Filtering with M-unit delays followed by a M:1 downsampling is equivalent to M:1 downsampling followed by filtering with 1 unit delays.

THE NOBLE IDENTITY 2

Efficient Implementation of Resampling filters

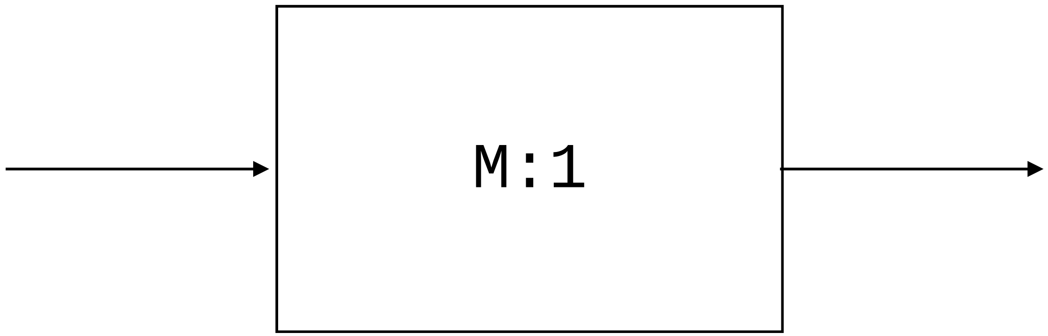


Rule 2: 1:M upsampling followed by filtering with M-unit delays is equivalent to filtering with 1 unit delays followed by 1:M upsampling.

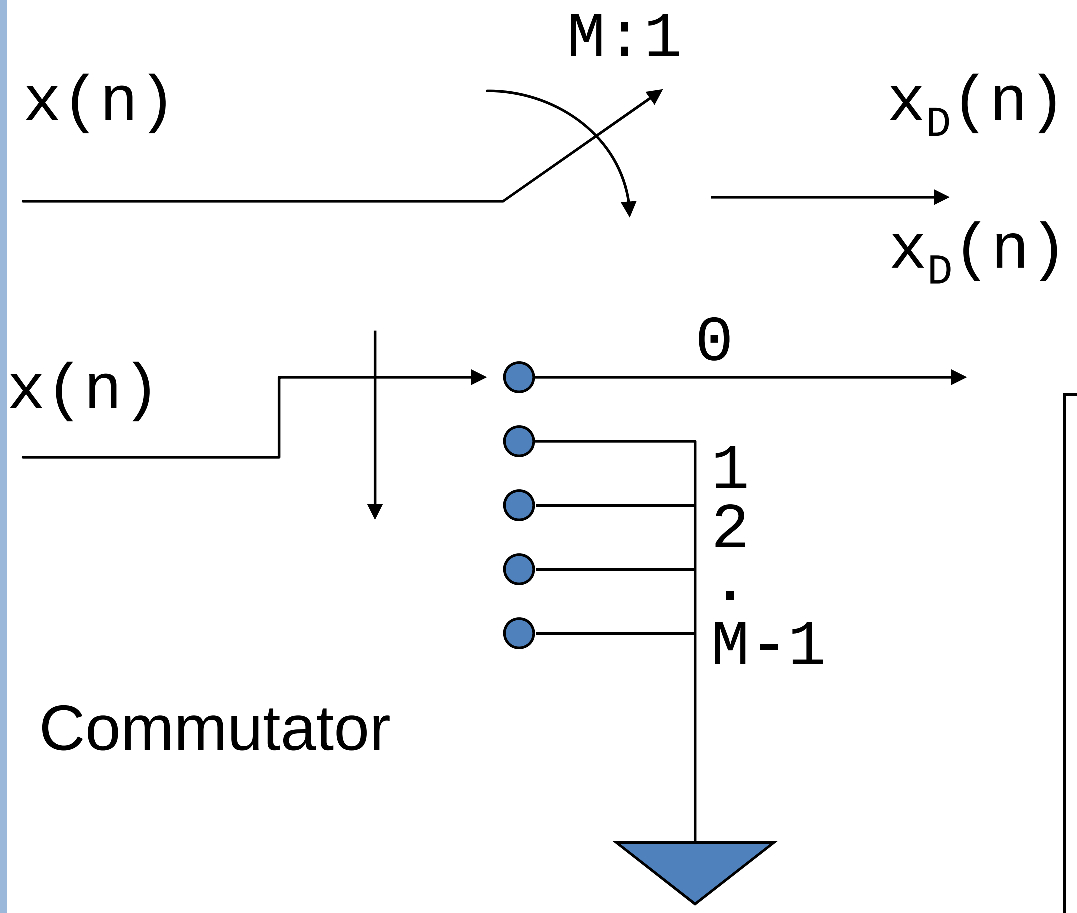
It is always more efficient to apply the filter at the lower sample rate

Downsampling

- Sometimes called *decimation*. This is poor nomenclature since decimation refers to removing 1 in 10.
- It is useful to think of the downsampling operation as the output of a commutator.



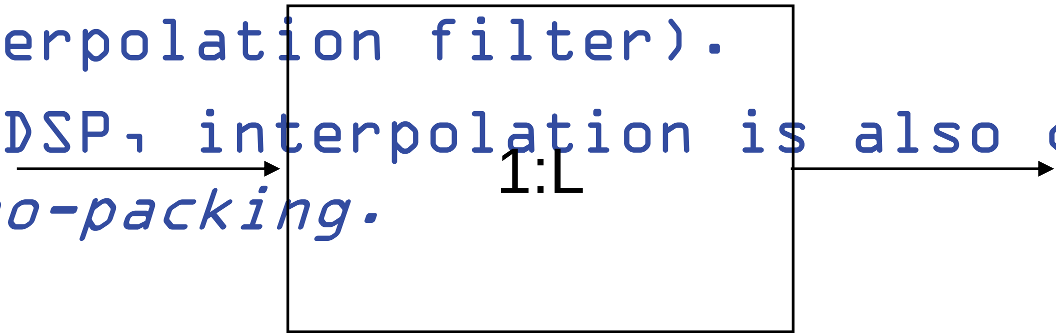
Downsampling



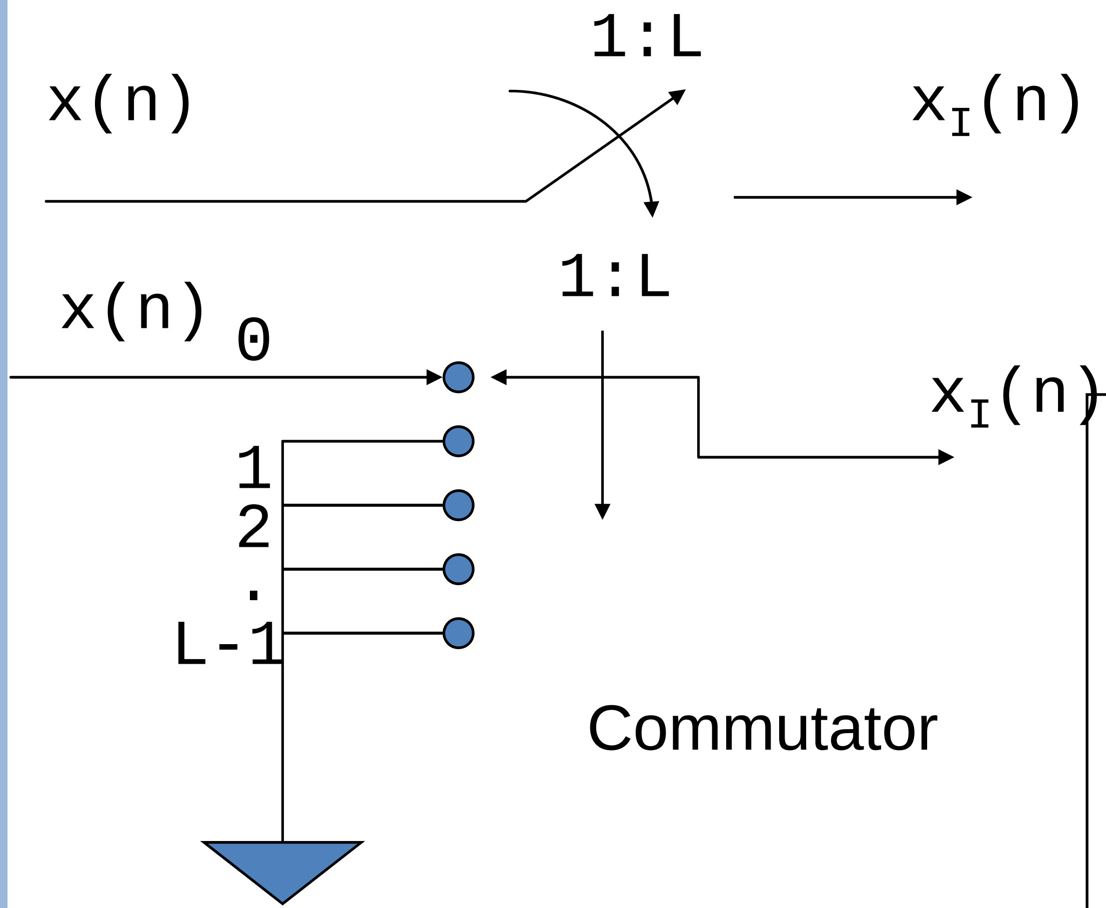
If x(n) has samples indexed from 0 to N-1, then $x_D(n)$ has samples indexed from 0 to (N/M)-1.

Upsampling

- Sometimes called *interpolation* although it is not interpolation in the conventional sense; it is just putting zeros between samples. If we wish to perform interpolation in the conventional sense, we must pass the DSP interpolated data through a low pass filter (anti-imaging or interpolation filter).
- In DSP, interpolation is also called *zero-packing*.

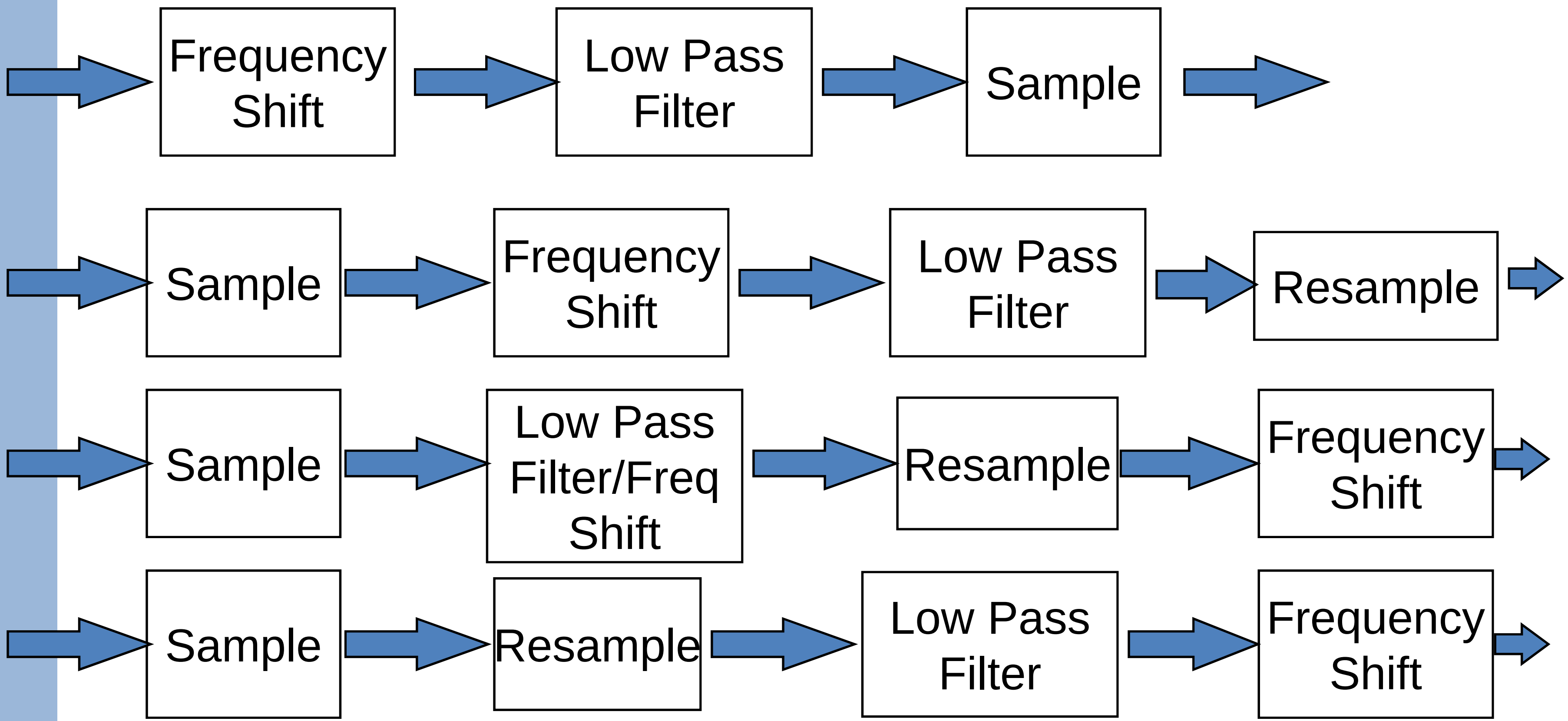


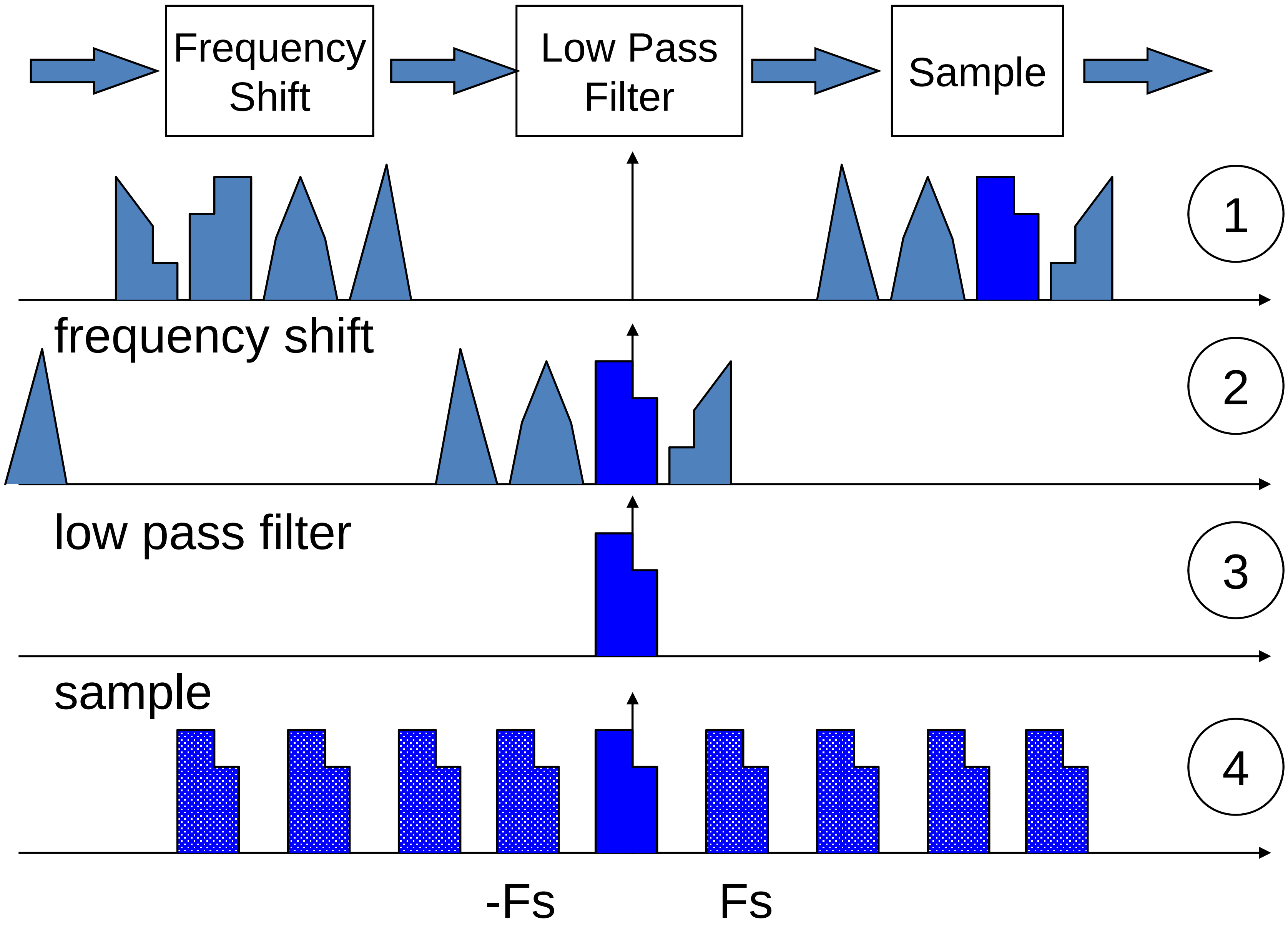
Upsampling



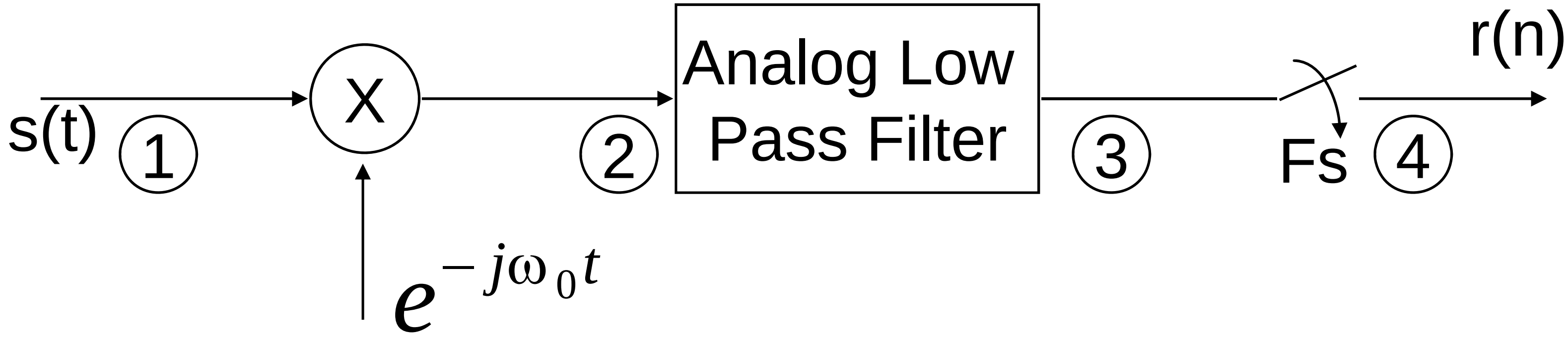
If $x(n)$ has samples indexed from 0 to *N-1*, then $x_I(n)$ has samples indexed from 0 to *LN-1*.

Digital Receiver Options

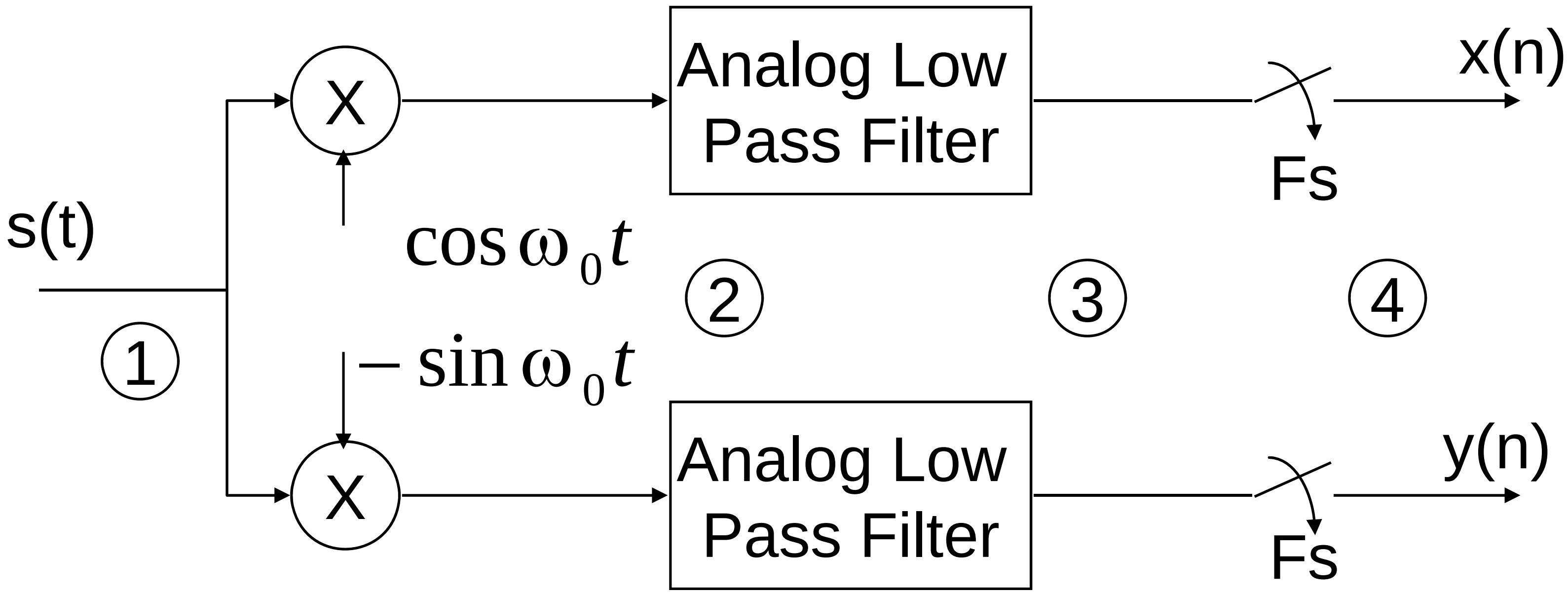



Frequency Shift
Low Pass Filter
Sample
1
frequency shift
2
low pass filter
3
sample
4
-Fs
Fs

Receiver Structure

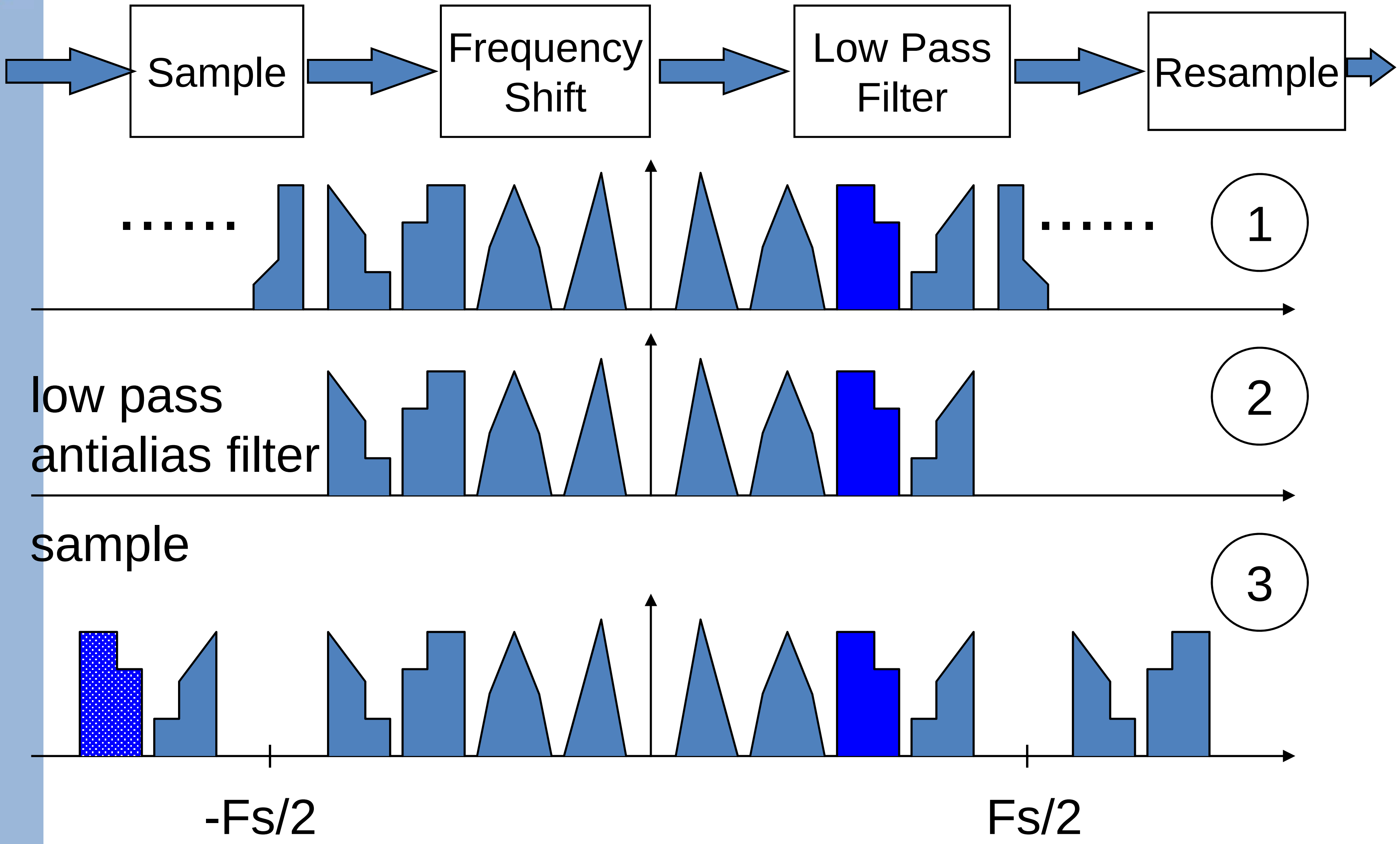


Which can be implemented by

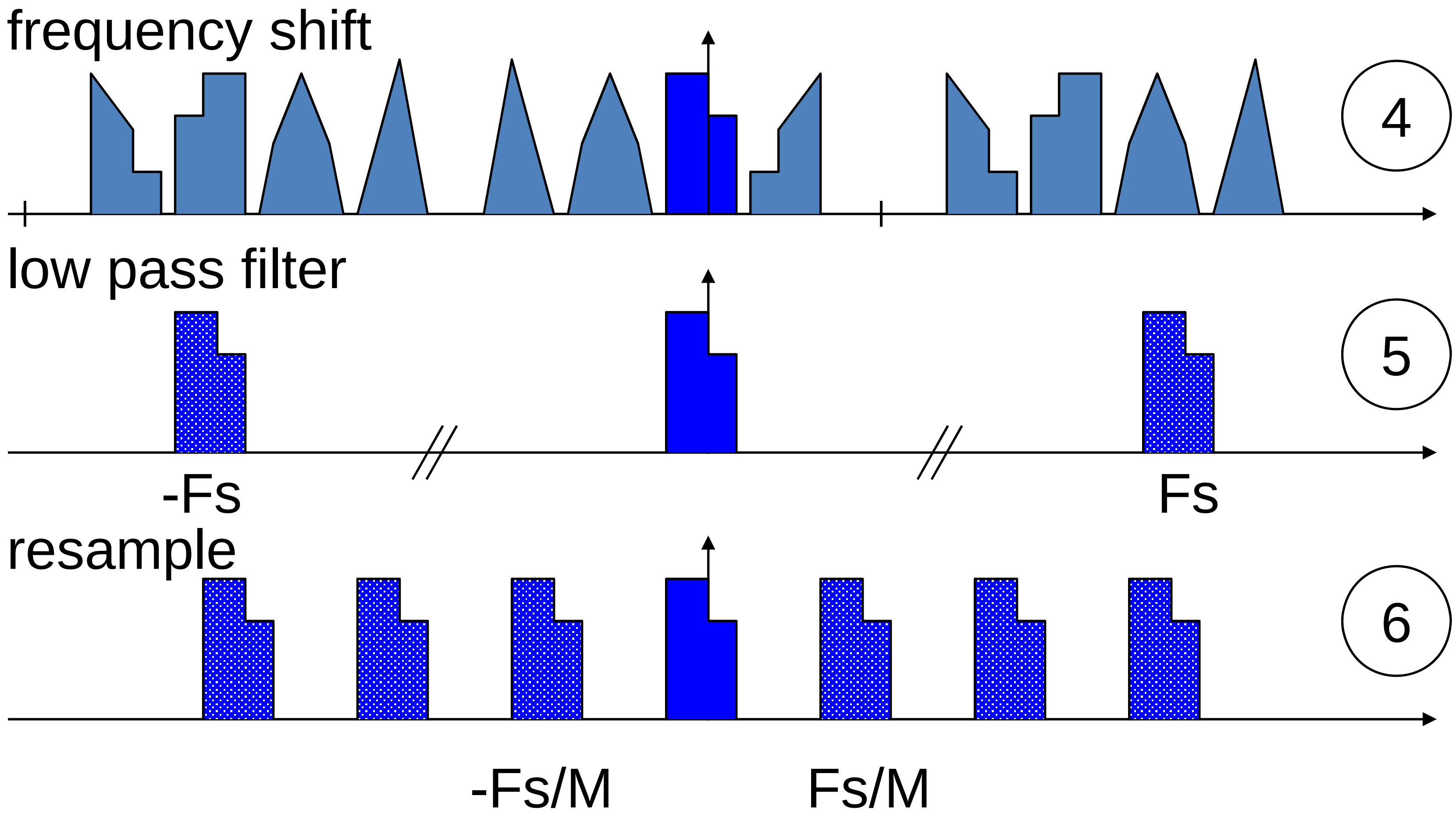


Comments

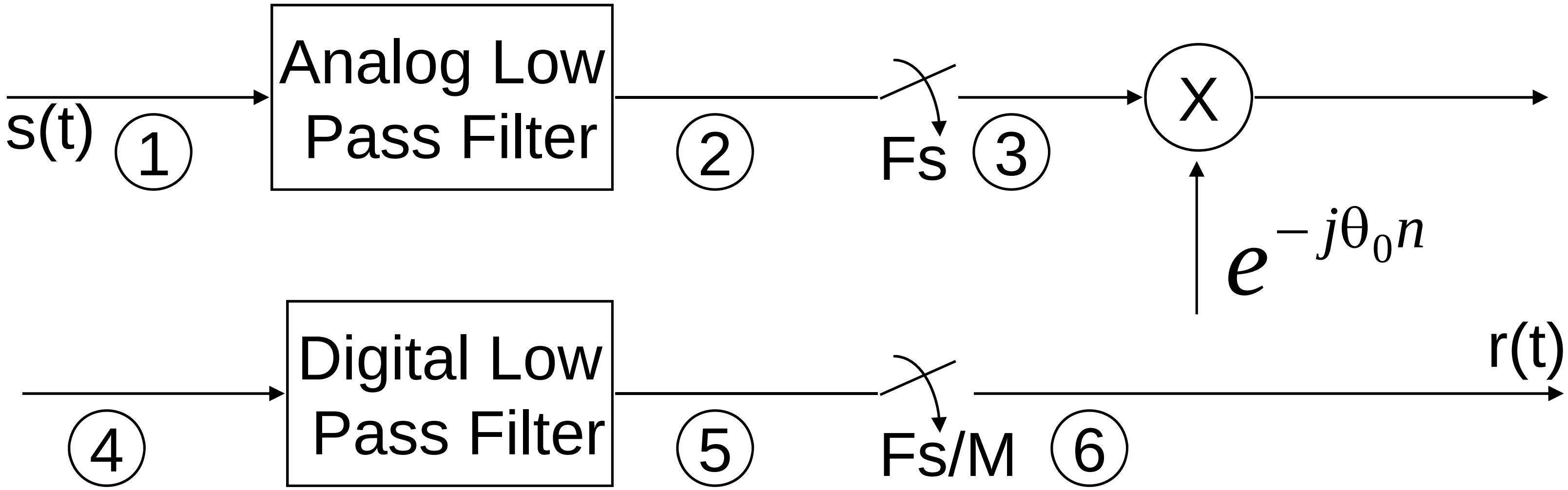
- Need to have
  - two perfectly matched analog mixers
  - two perfectly matched high order analog filters
  - two perfectly matched analog-to-digital converters
- Difficult to achieve and expensive
- Reduce expense by performing some of these operations in DSP

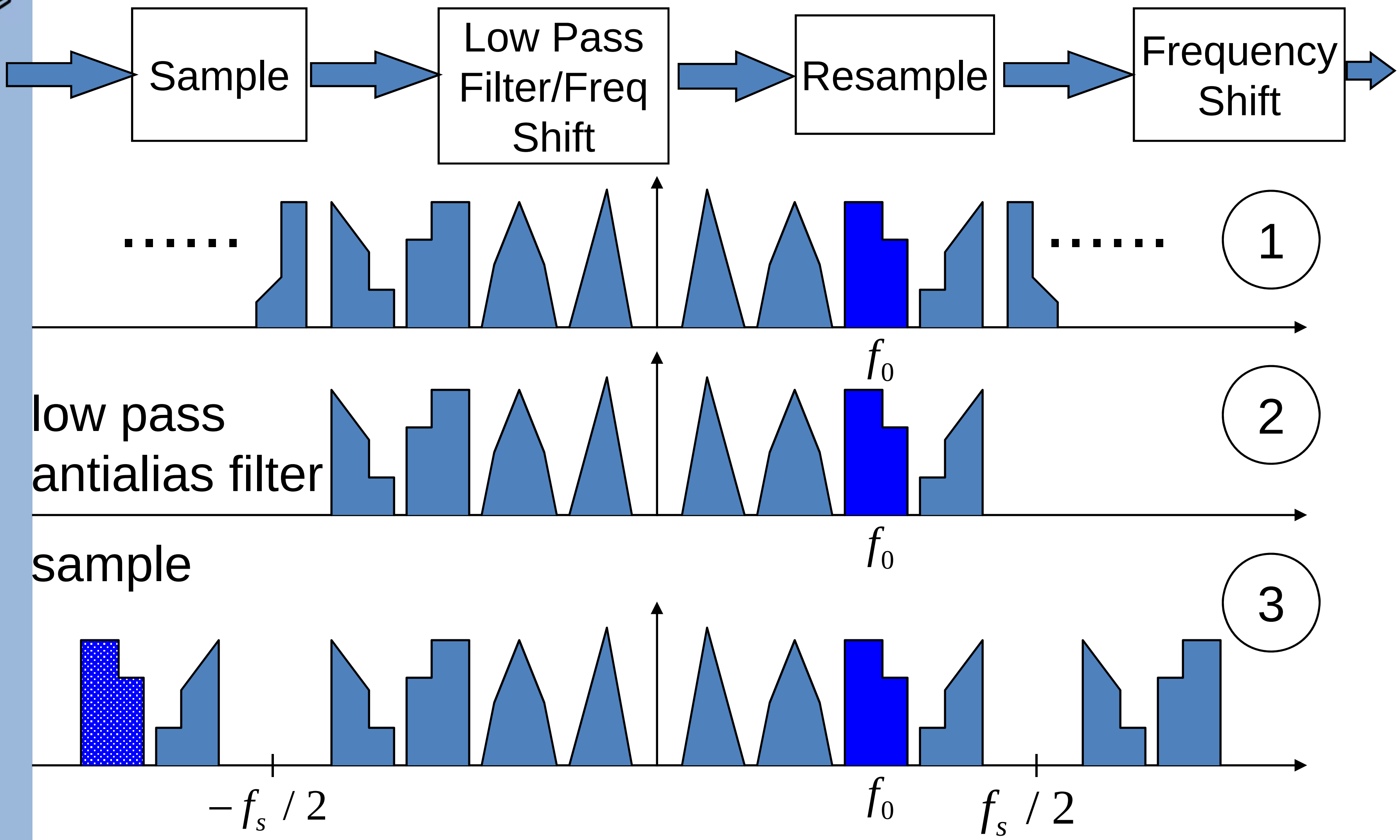


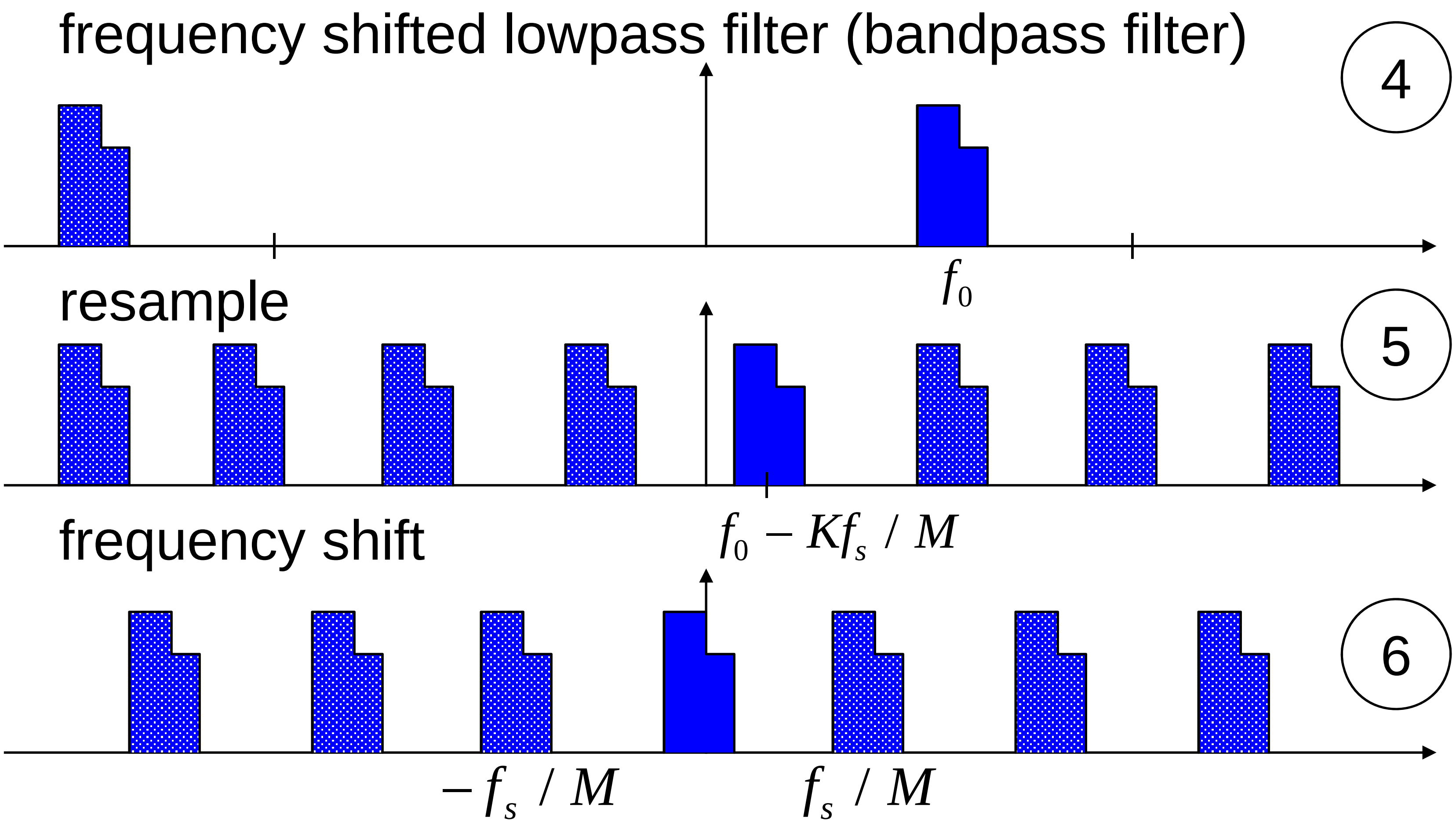
Sample
Frequency Shift
Low Pass Filter
Resample
1
2
3
low pass antialias filter
sample
-Fs/2
Fs/2


frequency shift
4
low pass filter
5
-Fs
Fs
resample
6
-Fs/M
Fs/M

Receiver Structure

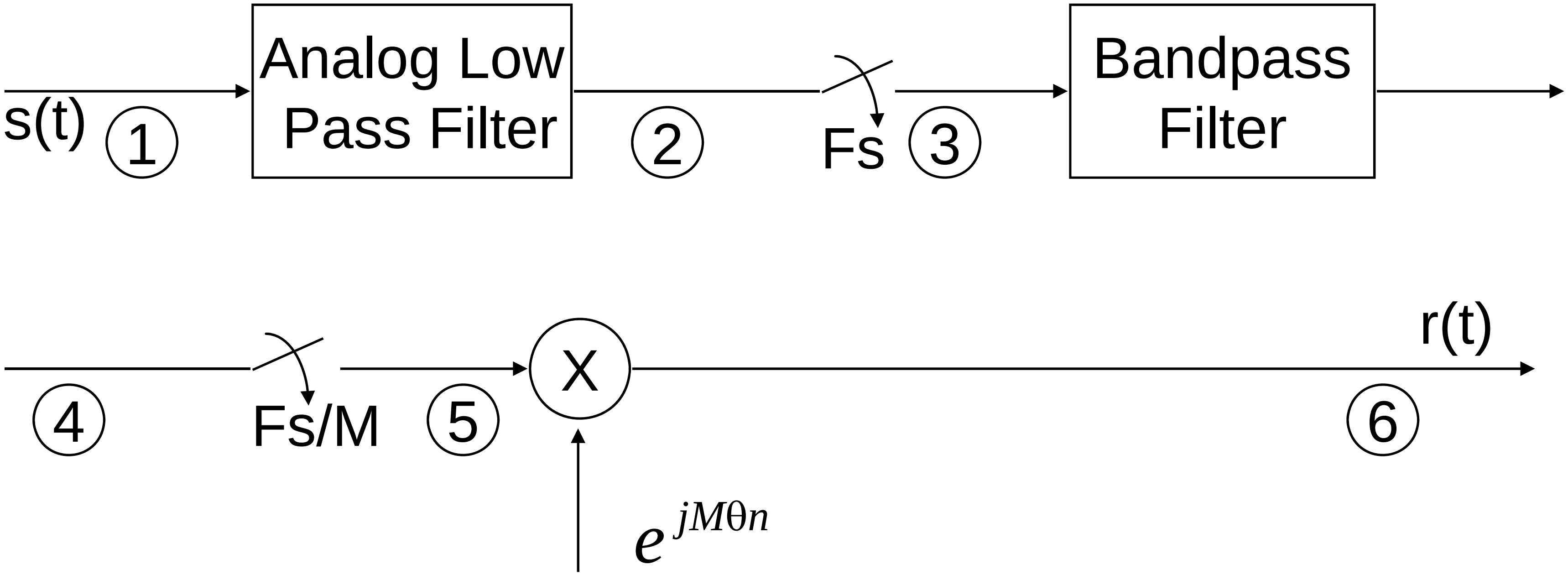



Sample
Low Pass Filter/Freq Shift
Resample
Frequency Shift
1
f_0
2
low pass antialias filter
f_0
sample
3
$-f_s/2$
f_0
$f_s/2$



f_s / M > two sided bandwidth

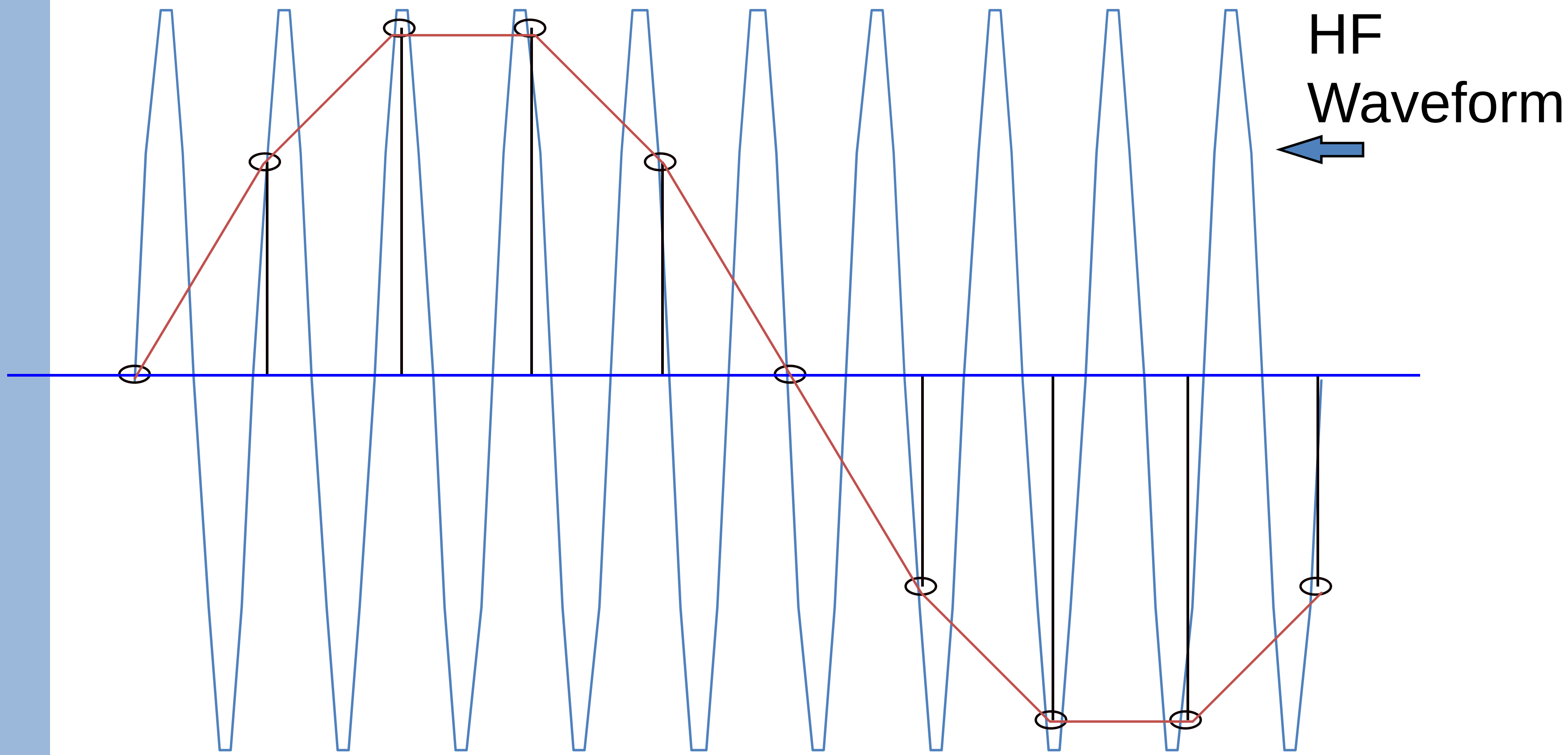
Receiver Structure



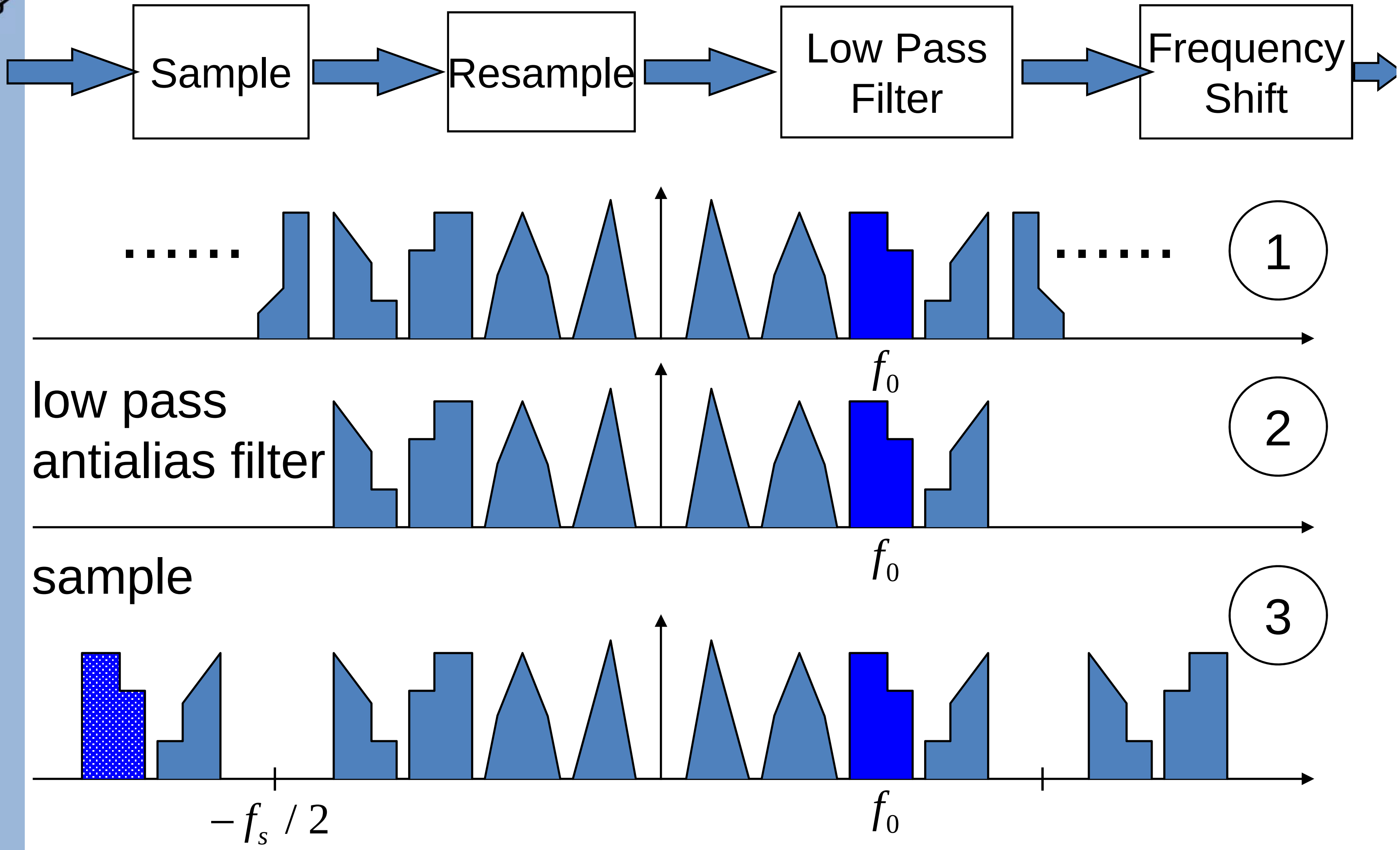
Comments

- No need to frequency shift signal to baseband before resampling.
- As long as the sampling frequency is greater than the two sided bandwidth, no information will be lost due to aliasing.
- This condition is sufficient even if the signal is passband although the resampling may appear to cause a frequency shift.
- This technique is exploited in sampling oscilloscopes which capture very high frequency signals with low sampling rates.

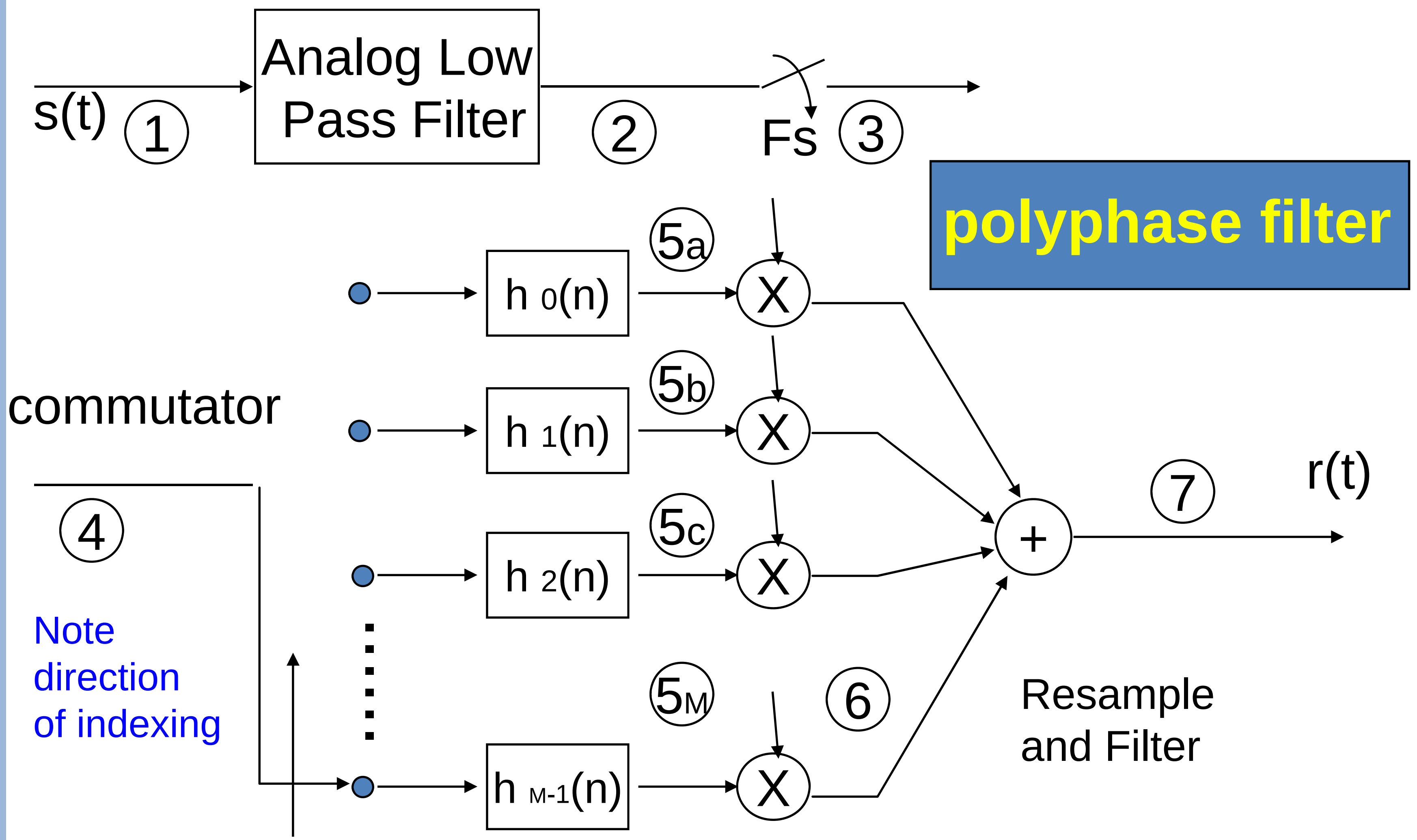
Sampling Oscilloscope Principle



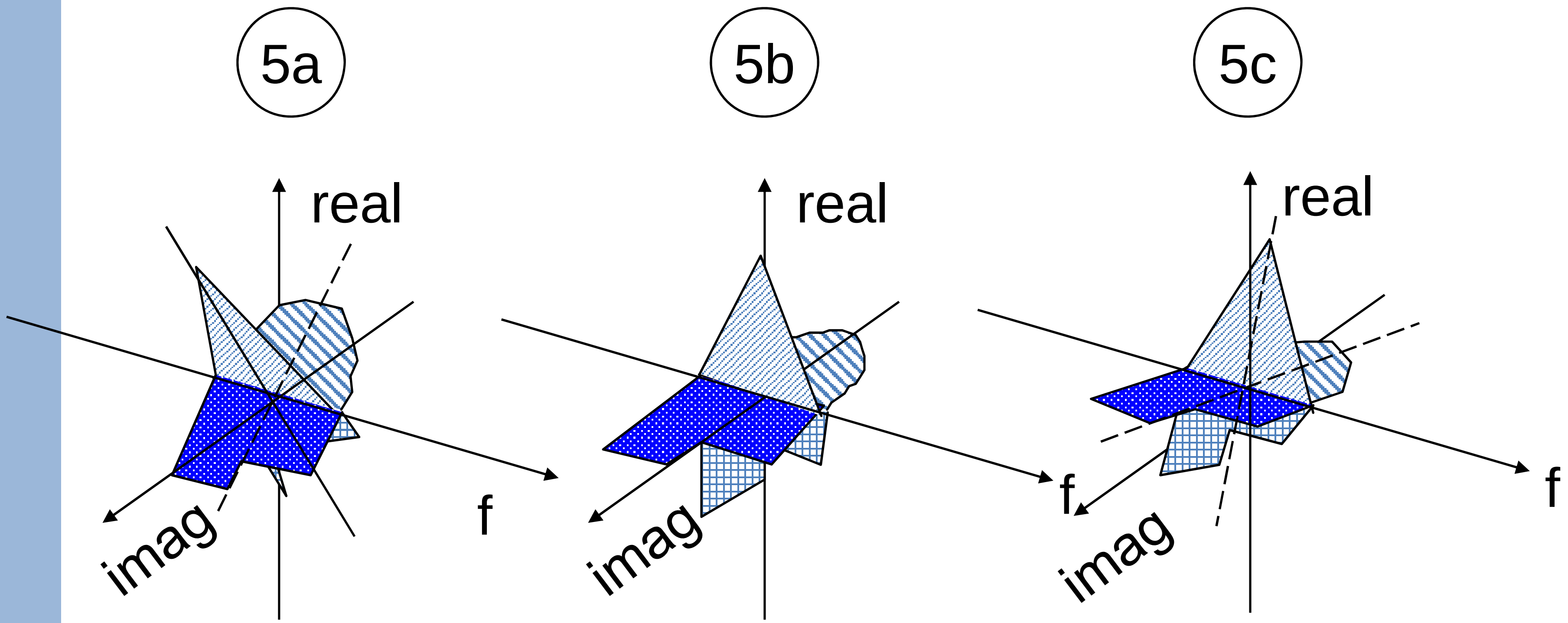
Sampled data is a time scaled version of HF waveform


Sample
Resample
Low Pass Filter
Frequency Shift
1
f_0
low pass antialias filter
2
f_0
sample
3
$-f_s/2$
f_0

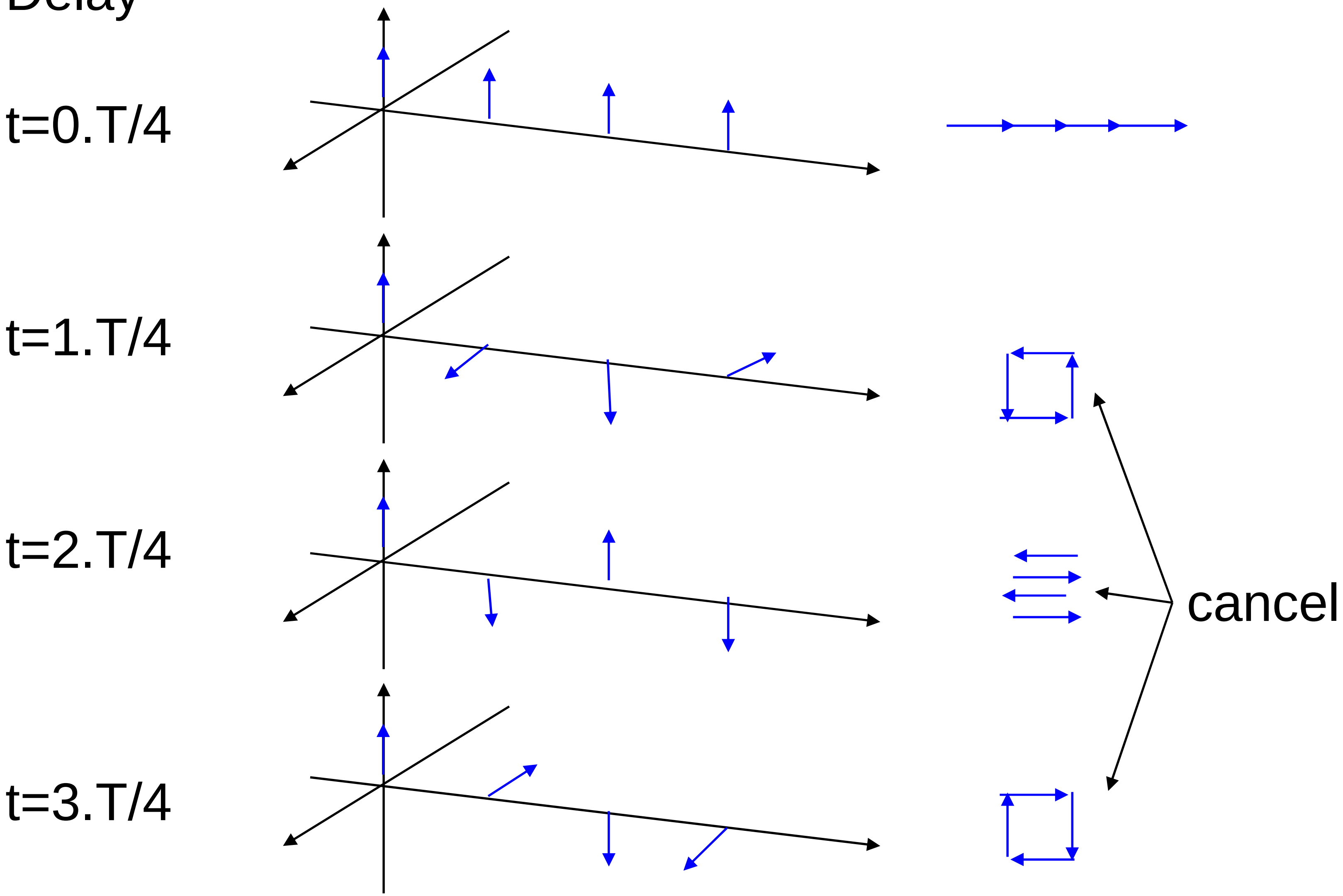
Receiver Structure



resample



Delay



Comments

- In the polyphase filter design we introduce deliberate aliasing by downsampling. Thus at the output of each filter, the desired signal is jumbled up with replicas of the other unwanted bands.
- Due to the phase delays in the separate paths through the filter, the signals in the unwanted bands will cancel at the summing node leaving only the desired signal.
- This structure is computationally more efficient than the previous bandpass filter design (by a factor of M).

Polyphase Filter Partition

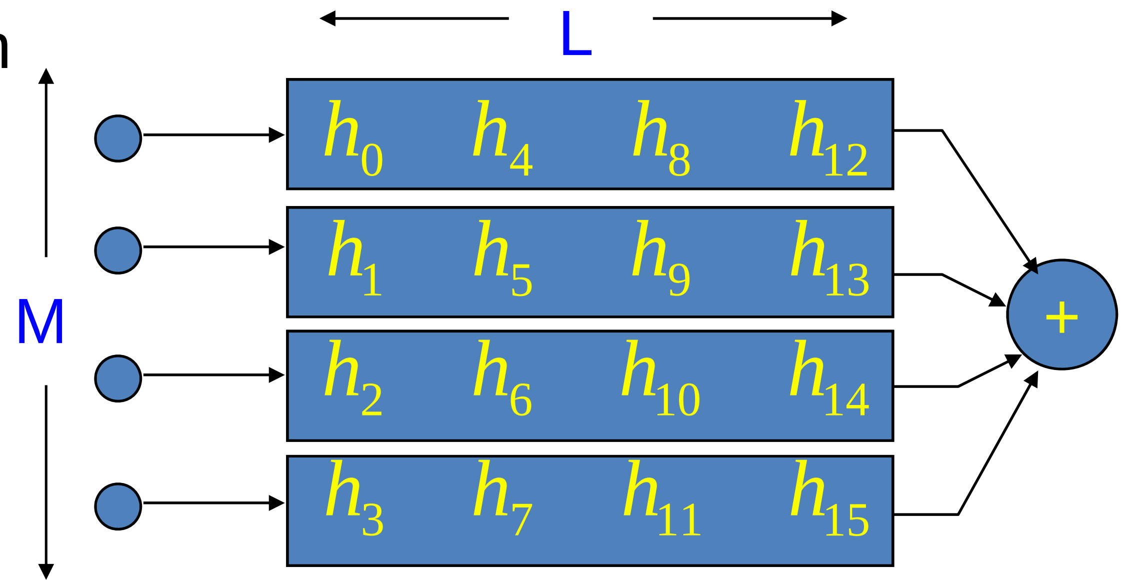
Let N = L*M

N = Filter Length

M = Resampling Rate

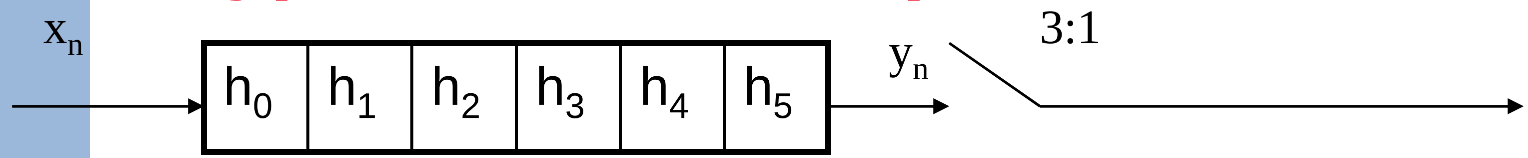
L = Subfilter Length

Place filter coefficients columnwise into an M by L matrix. Subfilters are the rows of the matrix.

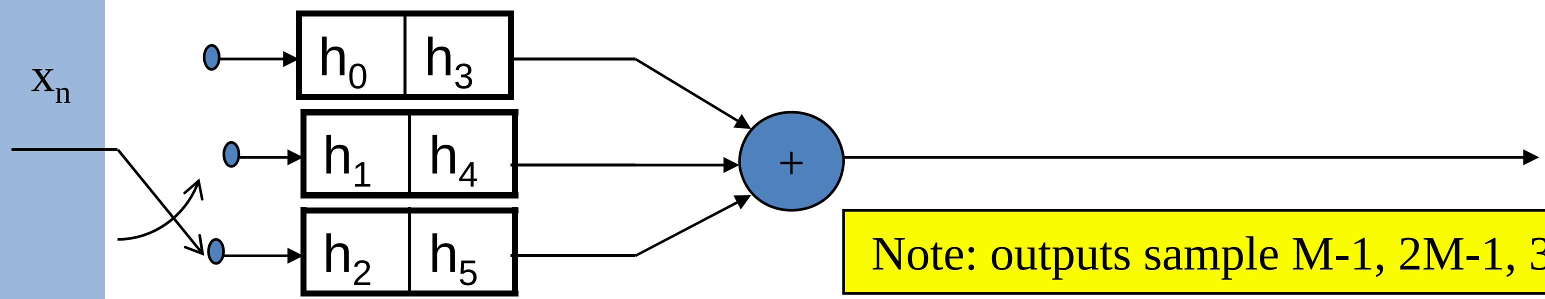


Note- can always zero pad to make N = L*M

Polyphase Downsampler in Detail

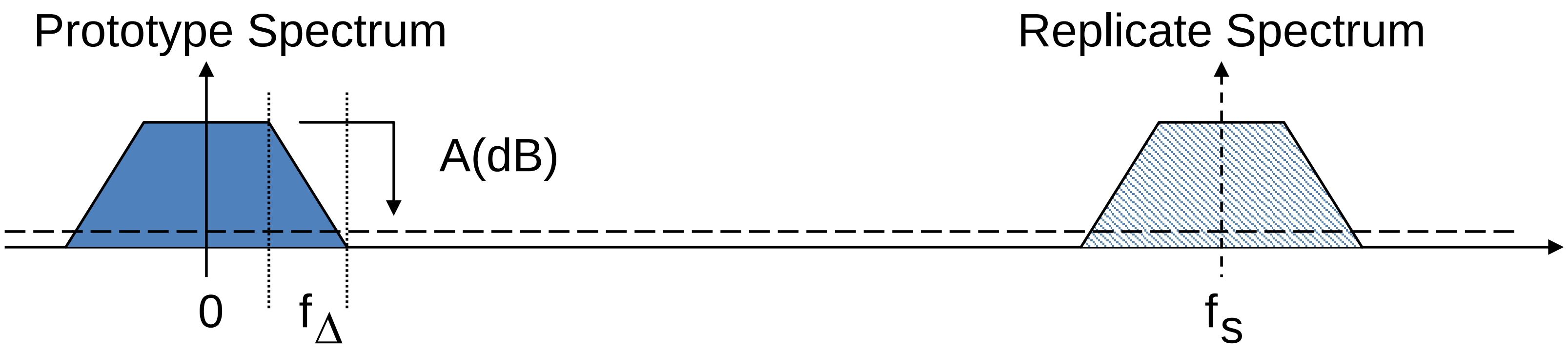


0: $y_0 = h_0x_0$
1: $y_1 = h_1x_0 + h_0x_1$
2: $\mathbf{y_2 = h_2x_0 + h_1x_1 + h_0x_2}$ → $\mathbf{y_2 = h_2x_0 + h_1x_1 + h_0x_2}$
3: $y_3 = h_3x_0 + h_2x_1 + h_1x_2 + h_0x_3$
4: $y_4 = h_4x_0 + h_3x_1 + h_2x_2 + h_1x_3 + h_0x_4$
5: $\mathbf{y_5 = h_5x_0 + h_4x_1 + h_3x_2 + h_2x_3 + h_1x_4 + h_1x_4}$ → $\mathbf{y_5 = h_5x_0 + h_4x_1 + h_3x_2 + h_2x_3 + h_1x_4 + h_1x_4}$



Note: outputs sample M-1, 2M-1, 3M-1 ….

Filter Design Equations



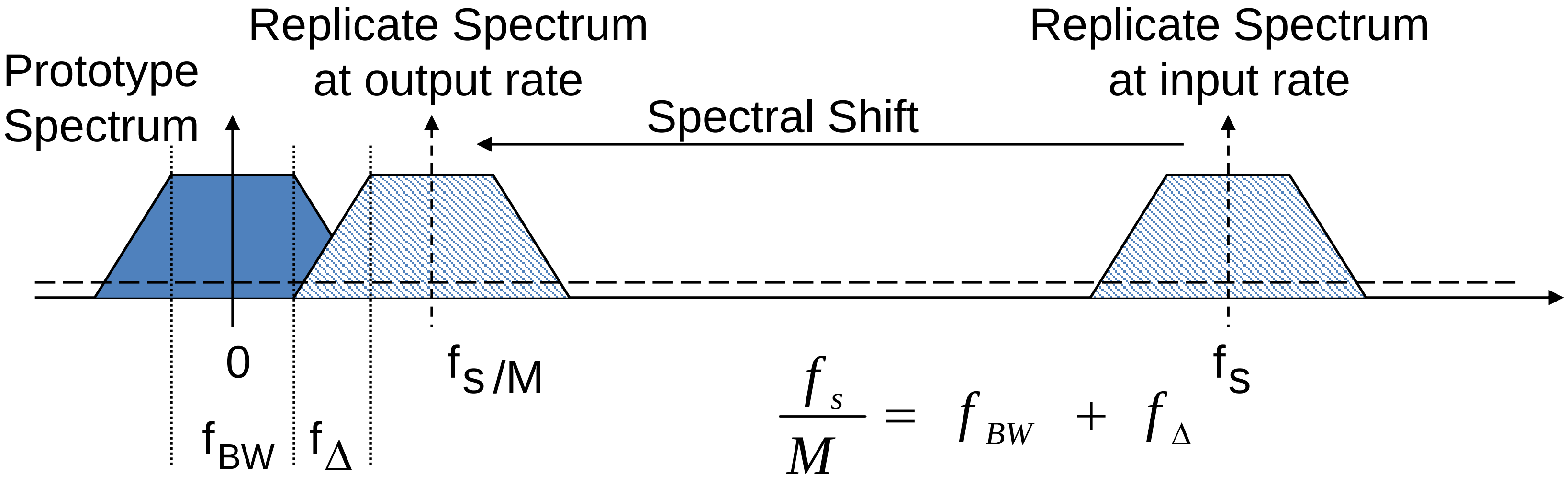
Example

$f_\Delta = 200$ Hz

$f_s = 20$ kHz

A (dB) = 80 dB

$$N = \frac{A(\text{dB})}{22}\frac{f_s}{f_\Delta} = \frac{80}{22}\frac{20{,}000}{200} = 364$$



$$f_\Delta = K(A)\frac{f_s}{N};\ K(A) = \frac{A(\mathrm{dB})}{22}$$

$$f_{BW} = a\frac{f_s}{M}$$

a = fractional bandwidth

$$\frac{f_s}{M} = \text{output sample rate}$$

$$\frac{f_s}{M} = f_{BW} + f_\Delta$$

$$\therefore\ \frac{f_s}{M} = a\frac{f_s}{M} + K(A)\frac{f_s}{N}$$

$$f_s\left[\frac{1-a}{M}\right] = K(A)\frac{f_s}{N}$$

$$N = M\frac{K(A)}{1-a}$$

$$\frac{N}{M} = \frac{K(A)}{1-a}$$

N/M determined by quality specifications of filter

N = filter size, number of ops per output point
M = resample ratio, number of input points per output point
N/M = Number of Ops per input point

Example

$$K(A) = 3.6 \quad (80\text{dB})$$

$$a = 0.6$$

$$\frac{N}{M} = \frac{K(A)}{1-a} = \frac{3.6}{0.4} = 9$$

Standard filter requires 364 Ops

Number of Ops/Input

Independent of bandwidth

Independent of sample rate

Independent of filter length

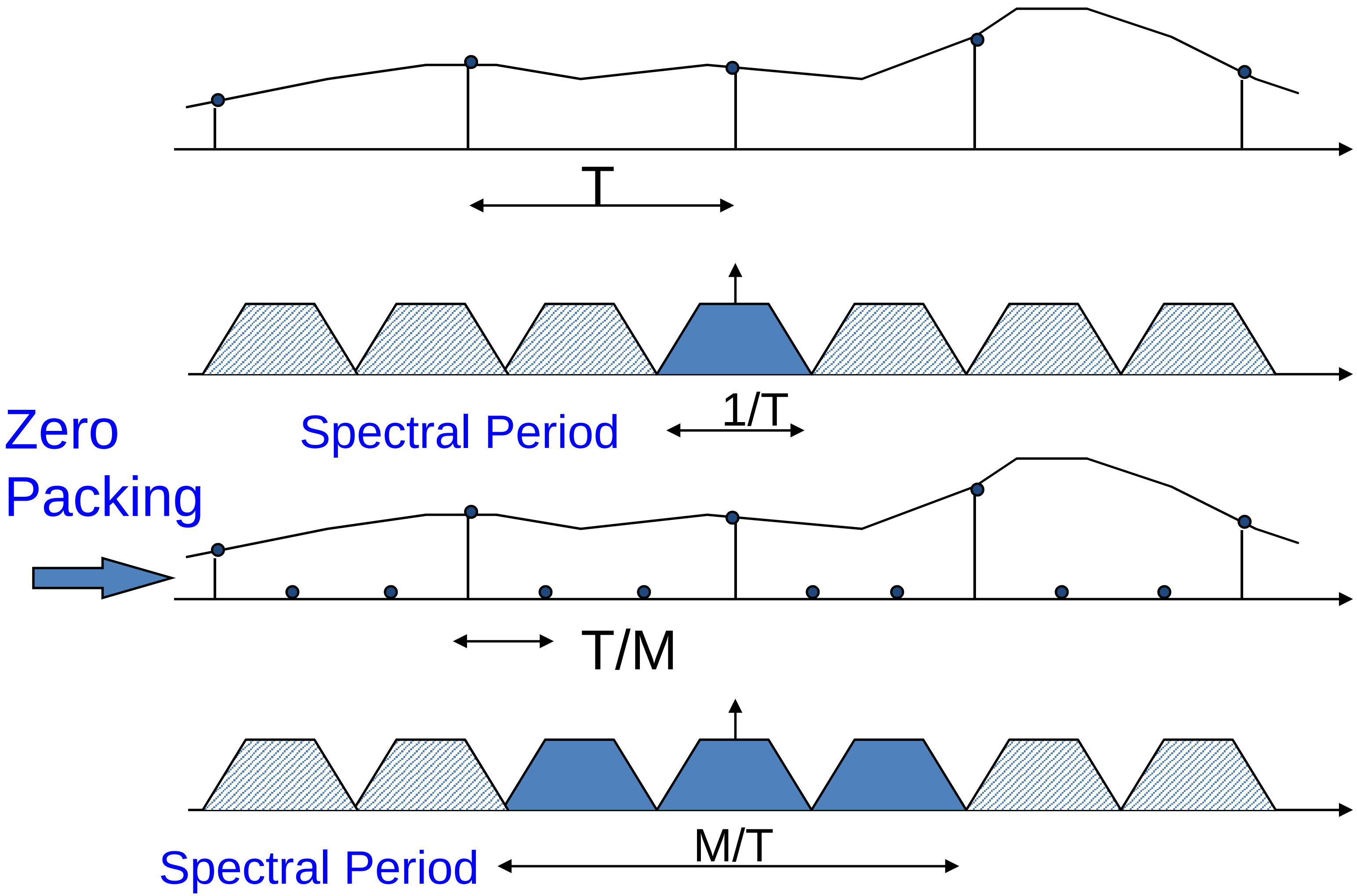
Dependent only on performance measures

fractional bandwidth (a) and out-of-band attenuation

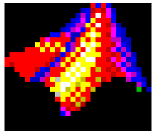
Comments

- This design formula lets the spectral replica overlap in the transition band so the attenuation never reaches the design value within the transition band.
- Need to use a filter design that has a small roll-off (say 3db per octave) in the stop band to ensure that the out of band noise does not stack up and prevent the filter from meeting specifications.

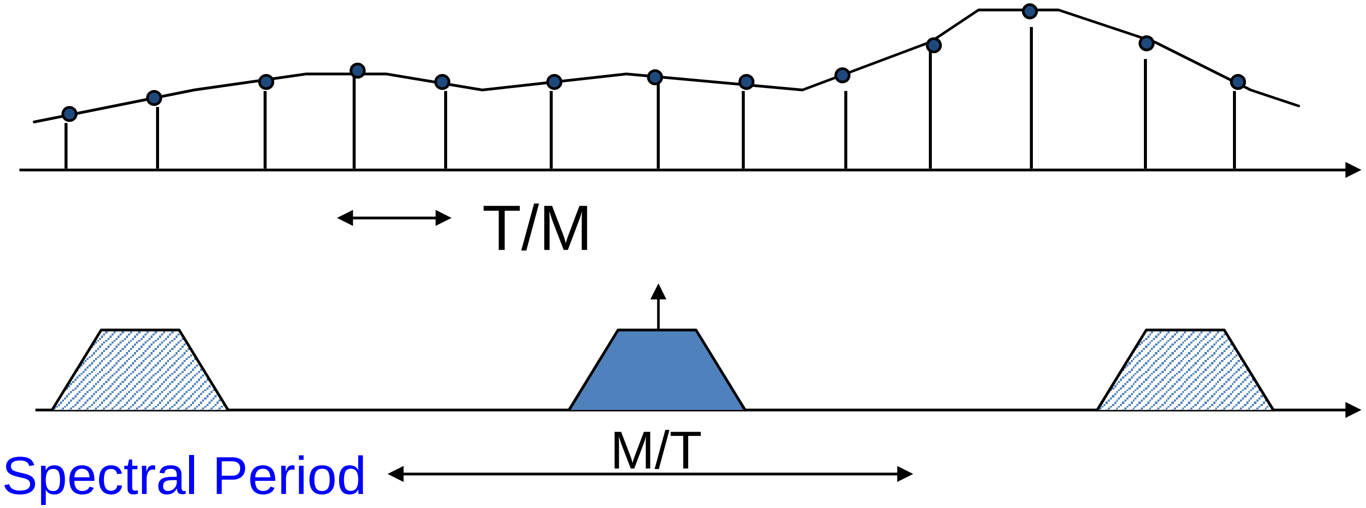
Upsampling



Upsampling

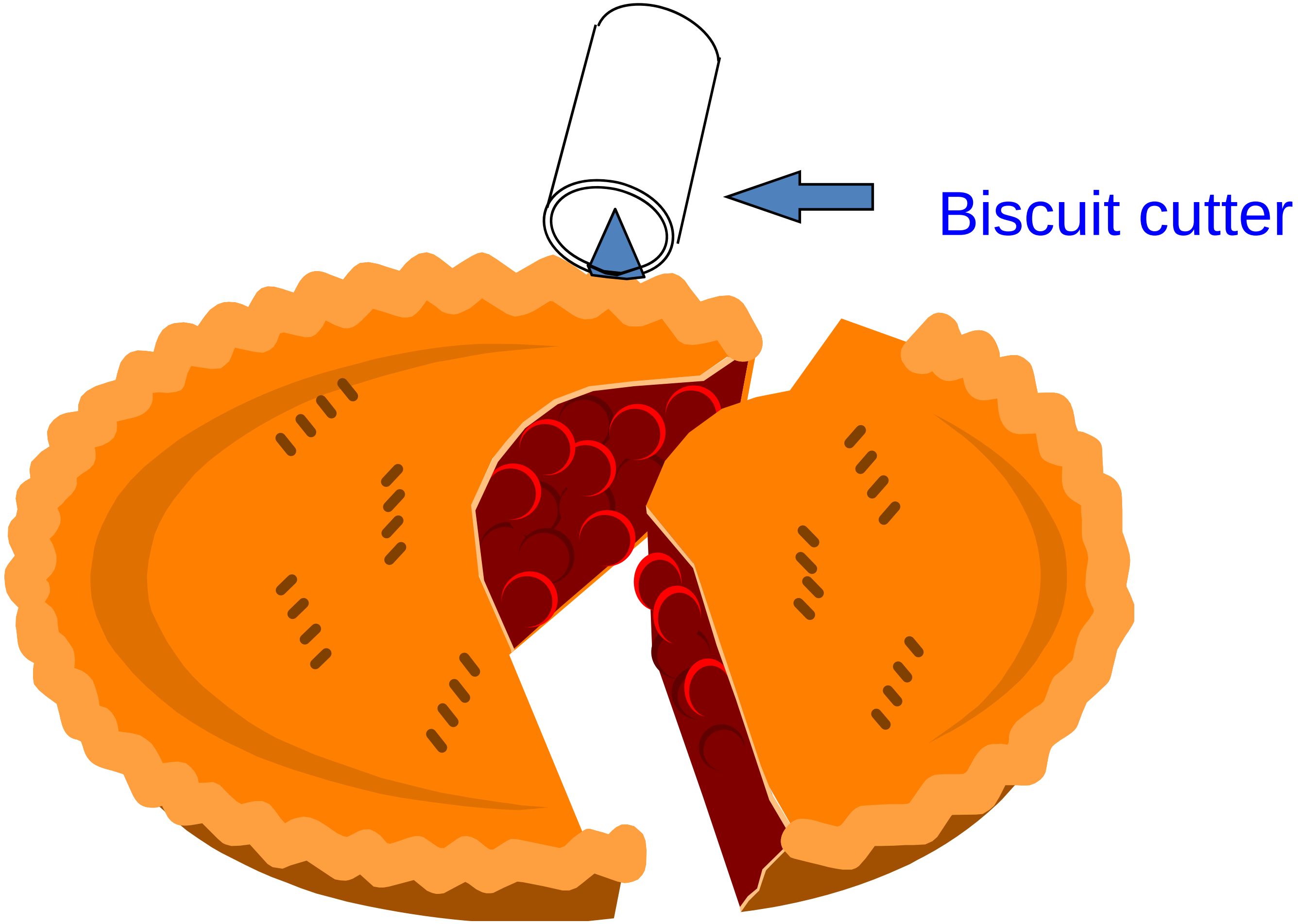


After low pass filtering to remove images
(interpolation, reconstruction, or anti-imaging filter)

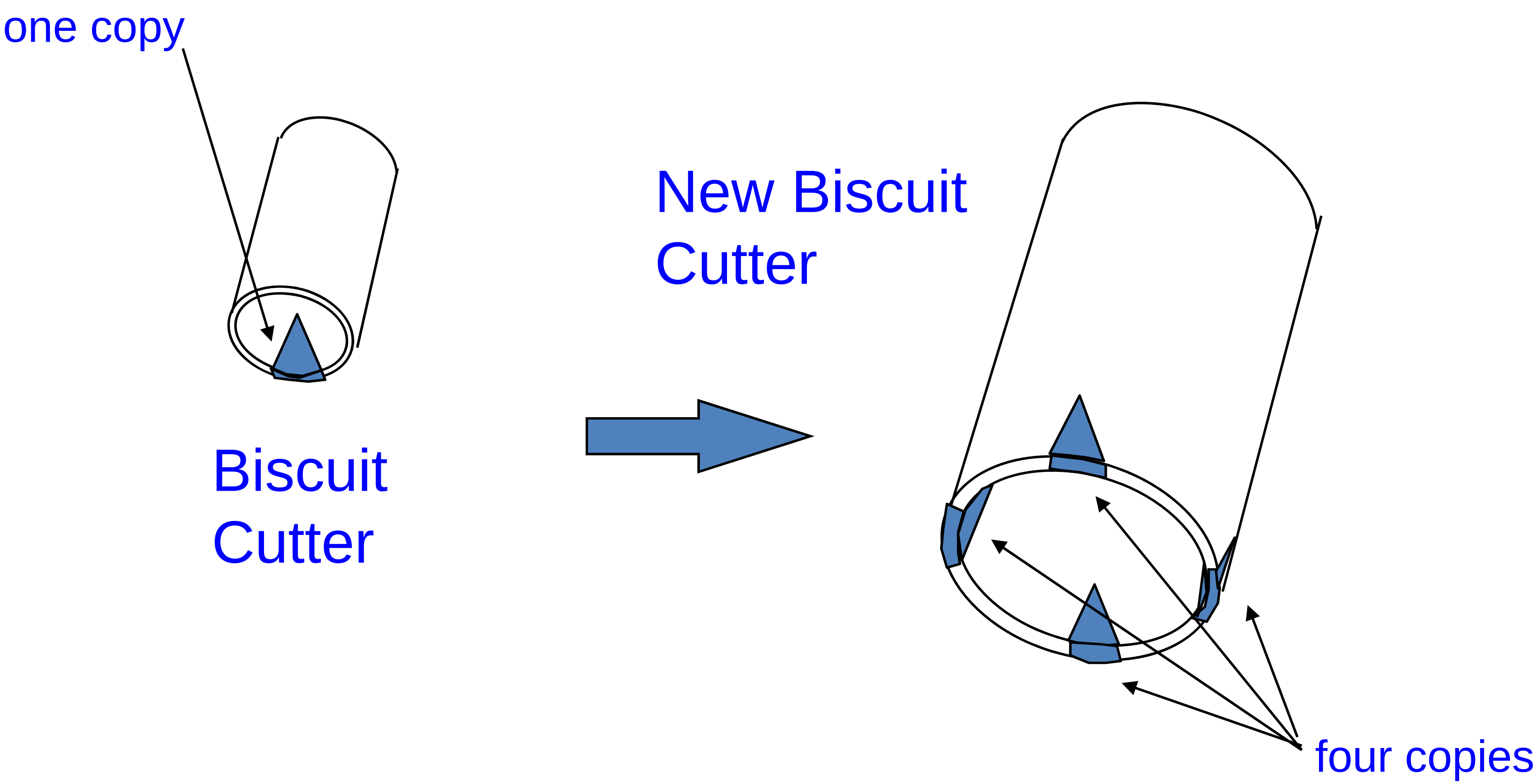


Interpolation(upsampling) = zero packing and low pass filtering

Cooking Class



Biscuit Cutter Model of Zero-Packing and DSP Filtering

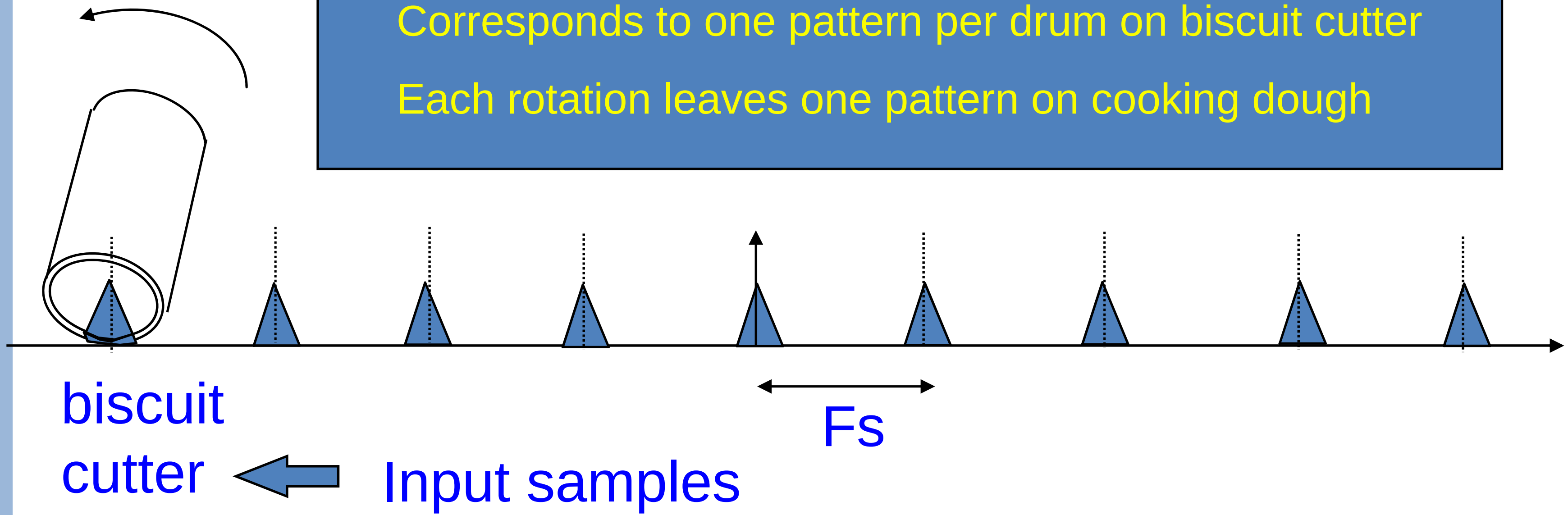


Input spectrum replicates every interval Fs =1/T

Period of spectrum with time samples every T seconds is Fs

Corresponds to one pattern per drum on biscuit cutter

Each rotation leaves one pattern on cooking dough

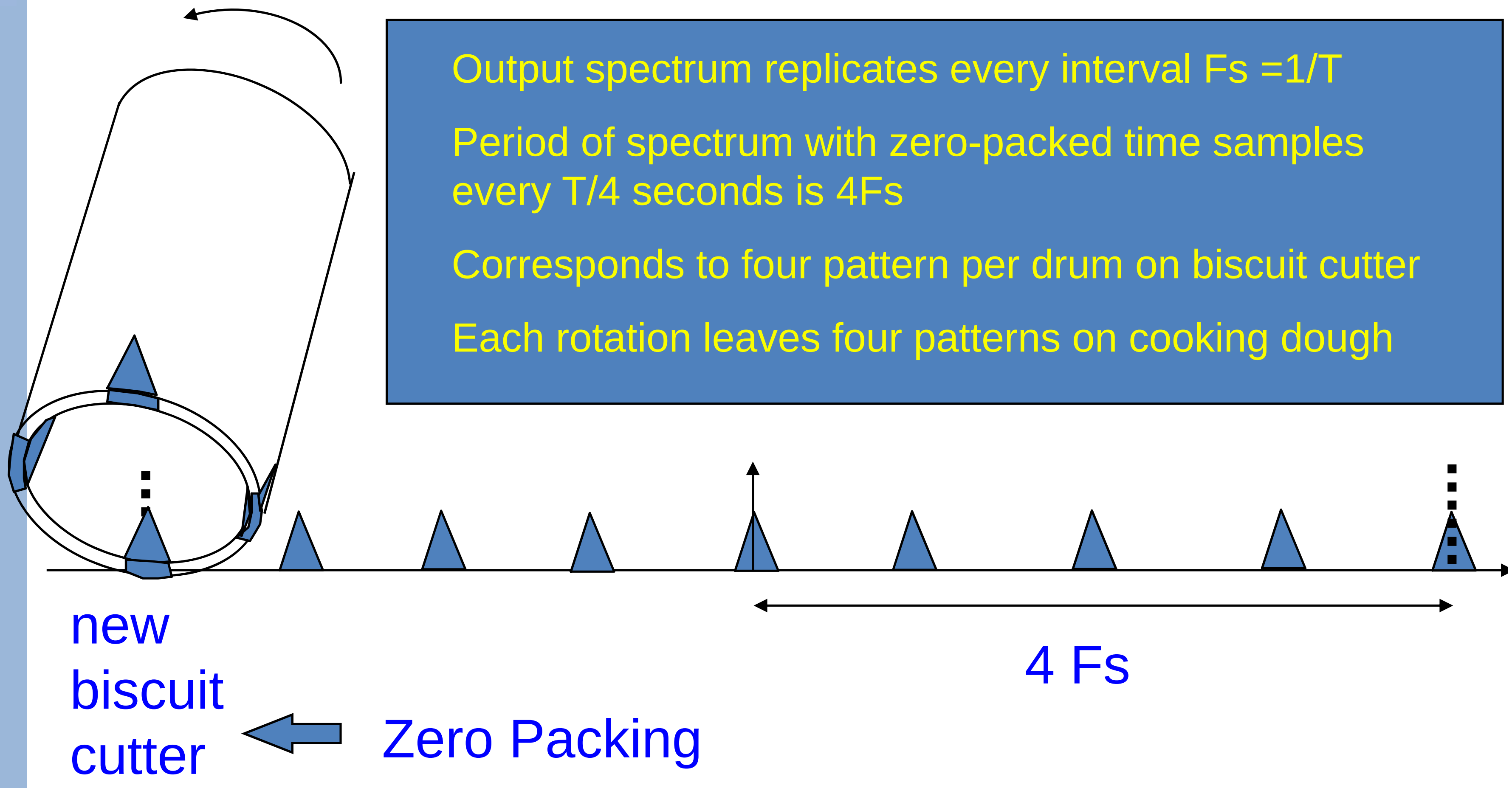


Output spectrum replicates every interval Fs =1/T

Period of spectrum with zero-packed time samples every T/4 seconds is 4Fs

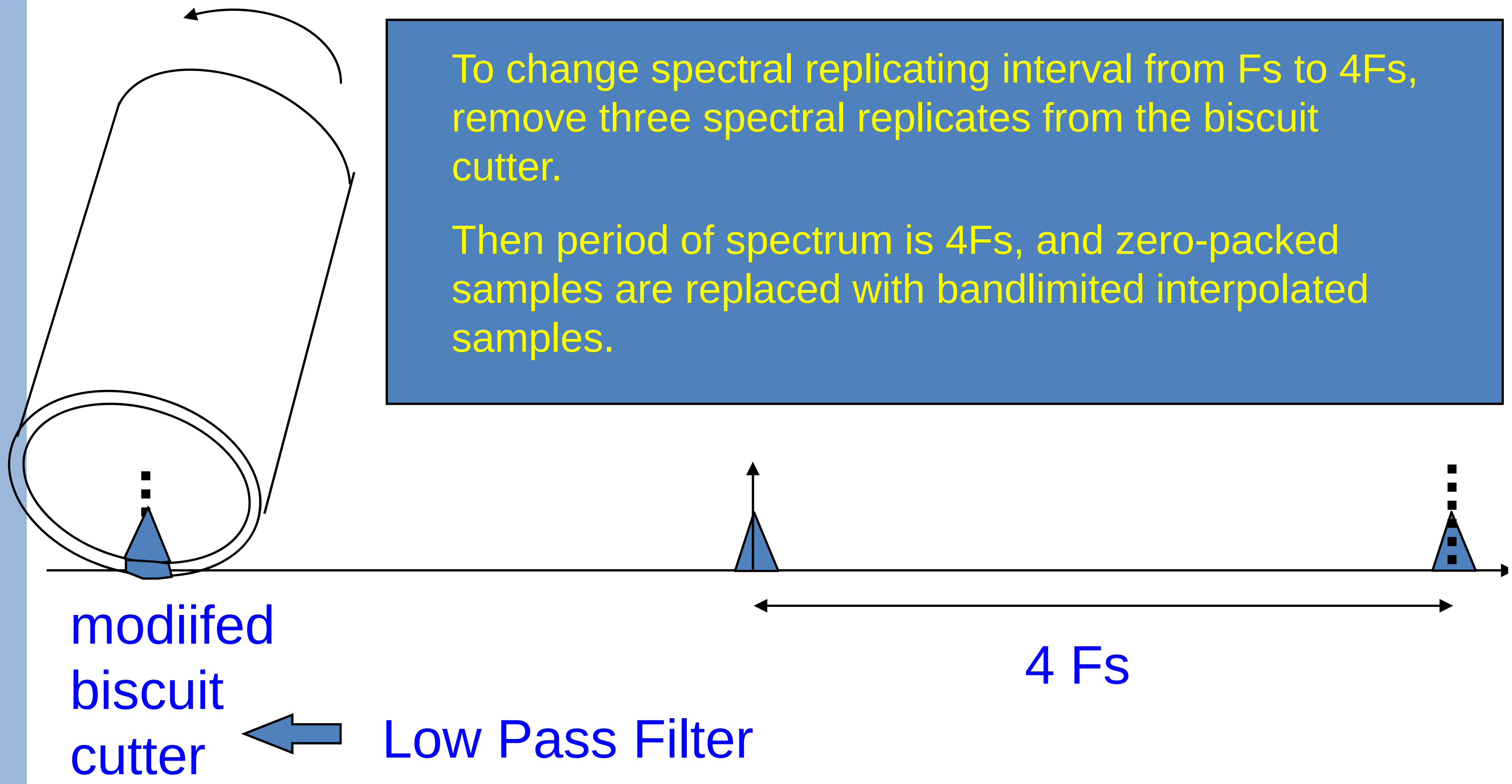
Corresponds to four pattern per drum on biscuit cutter

Each rotation leaves four patterns on cooking dough



To change spectral replicating interval from Fs to 4Fs, remove three spectral replicates from the biscuit cutter.

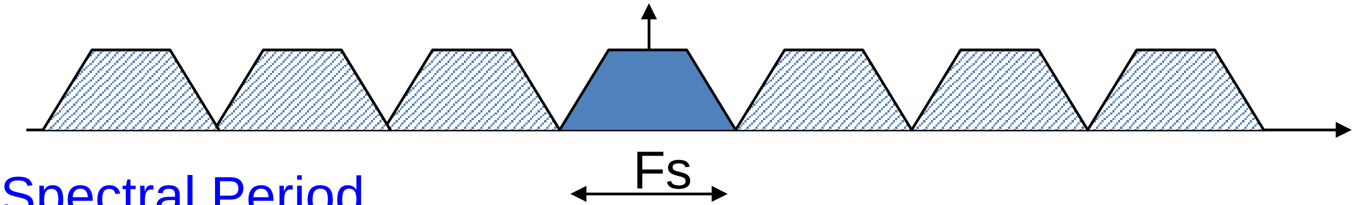
Then period of spectrum is 4Fs, and zero-packed samples are replaced with bandlimited interpolated samples.



DSP Approach

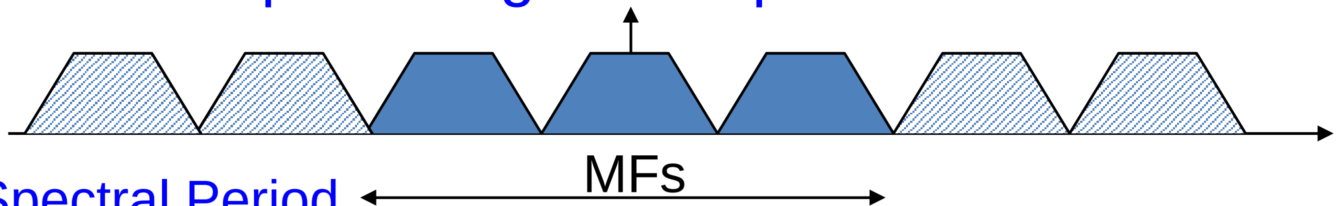
cirdemo

Spectrum: input to Upsampler



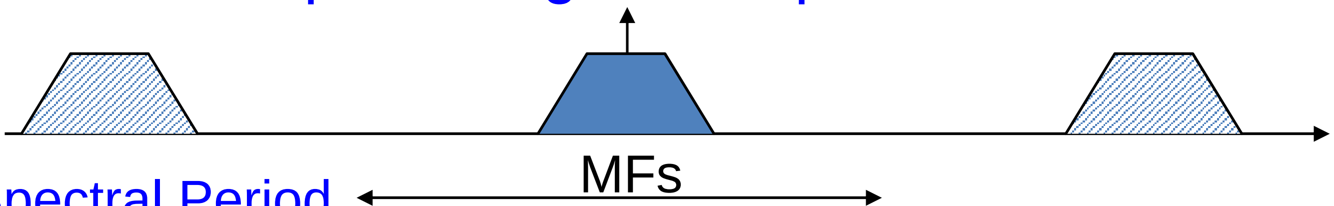
Spectral Period

Spectrum: Input to Digital Lowpass



Spectral Period

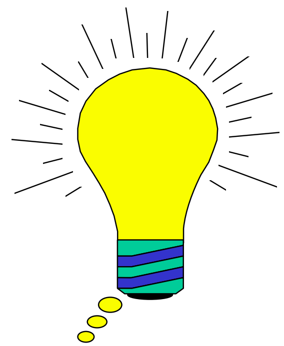
Spectrum: Output of Digital Lowpass



Spectral Period

Comments

- We have not used the Noble Identity which says that upsampling followed by filtering can be achieved by filtering followed by upsampling.
- Thus we should implement this upsampling with a polyphase structure to reduce the computational load by a factor of M.



Polyphase Upsampler

Let N = L*M

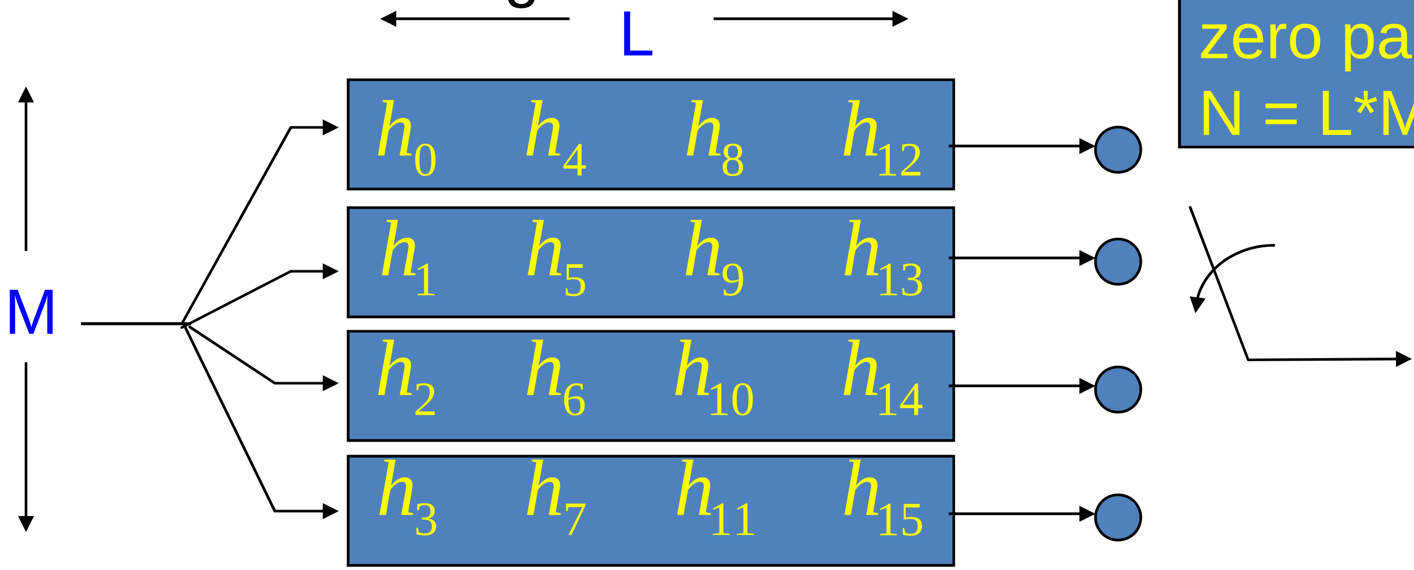
N = Filter Length

M = Resampling Rate

L = Subfilter Length

Place filter coefficients columnwise into an M by L matrix. Subfilters are the rows of the matrix.

Note - can always zero pad to make N = L*M



Another View