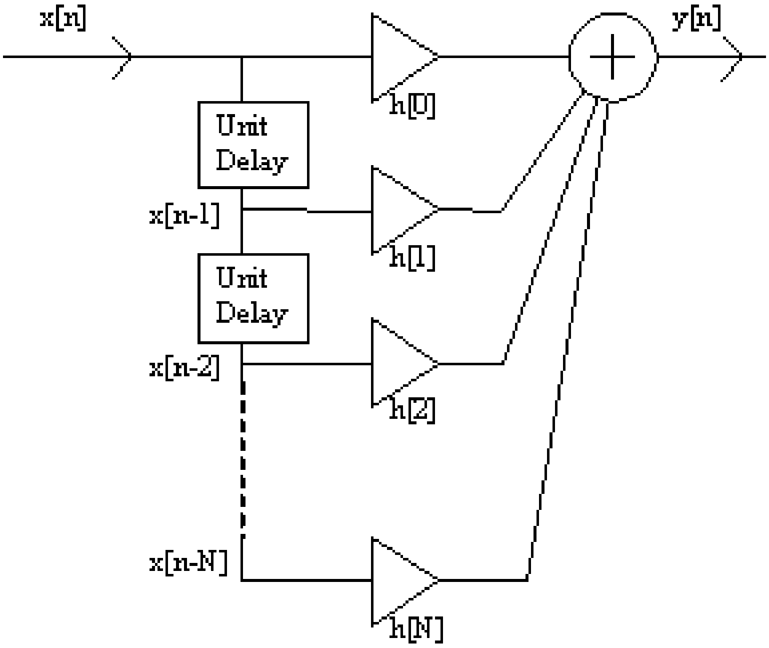
Solutions for: Finite Impulse Response Filters Tutorial

1. Assume that the input to the filter shown below is a single pulse at time zero, i.e. $x[0] = 1$ and all other x-values are zero. Work out where the "1" is located in the above diagram in each clock cycle and from this confirm that the output sequence is:

$$h[0], h[1], h[2], ..., h[N], 0, 0, 0, ...$$



Solution This can be illustrated with the use of a table. Assume that N=3, to help limit the size of the table:

n	$x[n]$	$x[n-1]$	$x[n-2]$	$x[n-3]$	$y[n]$
-2	0	0	0	0	0
-1	0	0	0	0	0
0	1	0	0	0	$h[0]$
1	0	1	0	0	$h[1]$
2	0	0	1	0	$h[2]$
3	0	0	0	1	$h[3]$
4	0	0	0	0	0
5	0	0	0	0	0

2. How do you find $H(f) = \sum_{k=0}^{N} h[k]e^{-j2\pi fkT}$ from $H(z) = \sum_{k=0}^{N} h[k]z^{-k}$ **Solution** Comparing the formulas we can see that it is a simple substitution: $z \leftarrow e^{j2\pi fT}$ Or, making use of the normalised radian frequency $\Omega = 2\pi f/f_s = 2\pi fT$:

$$z \leftarrow e^{j\pi\Omega}$$

i.e. evaluate aboveH(z) on the unit circle:

$$|e^{j\pi\Omega}|^2 = |cos(\pi\Omega) + j.sin(\pi\Omega)|^2 = cos^2(\pi\Omega) + sin^2(\pi\Omega) = 1$$

3. You are required to design a low-pass digital filter with a sampling frequency of 48kHz and a cut-off frequency at 8kHz. The theoretical pulse response of an ideal low-pass filter which exactly meets this specification is

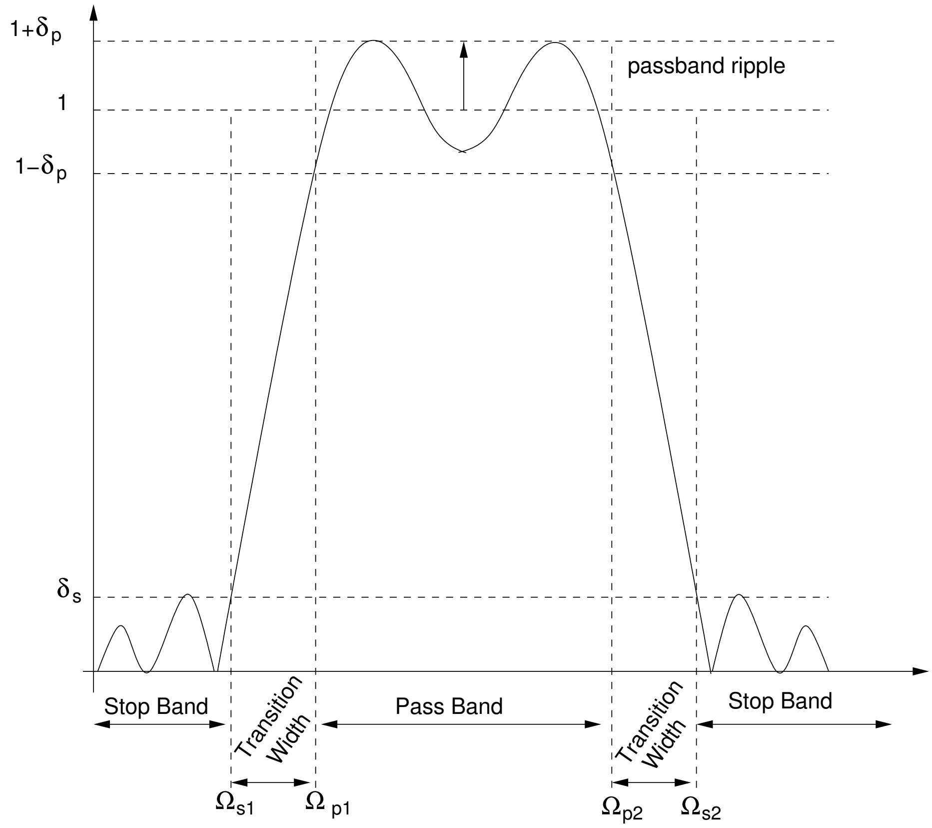
$$h[k] = \frac{1}{3}\text{sinc}\left(\frac{k}{3}\right).$$

(a) Explain why it is necessary to truncate and delay this response in order to have a filter that can be implemented in practice. **Solution** The theoretical response runs from $k = -\infty$ to $k = +\infty$. In order to make it finite it has to be truncated (usually symmetrically about $k = 0$). However the filter will still be non-causal, i.e. looking into the future, for $k > 0$. Therefore the entire filter is shifted, substituting $k - M = k$ for all values of k where M is the maximum non-causal k value so that, afterwards, all $k \leq 0$.

(b) Outline the consequences of the truncation on the amplitude response of the filter. **Solution** Truncation has 3 effects on the amplitude response:

- It creates a pass-band ripple
- It creates a stop-band ripple
- It also creates a finite width transition band, where the transition between the passband and the stopband is associated with a finite number of frequencies. This transition band is sometimes a problem because it prevents the elimination of frequencies below or above a certain point whilst passing frequencies below or above another point. The distance between these two points should ideally be zero, but this is not possible in practice because of the need to truncate the amplitude response of the filter.

These effects can be seen in the figure below.



(c) Outline the consequences of the delay on the amplitude response of the filter. **Solution** Making the filter causal has no effect on the amplitude response.

(d) State the effects on the amplitude response of applying a window to the truncated coefficients **Solution** The windowing helps to reduce the size of the ripples, providing a flatter frequency response in both the pass band and the stop band. However it also increases the width of the transition band.

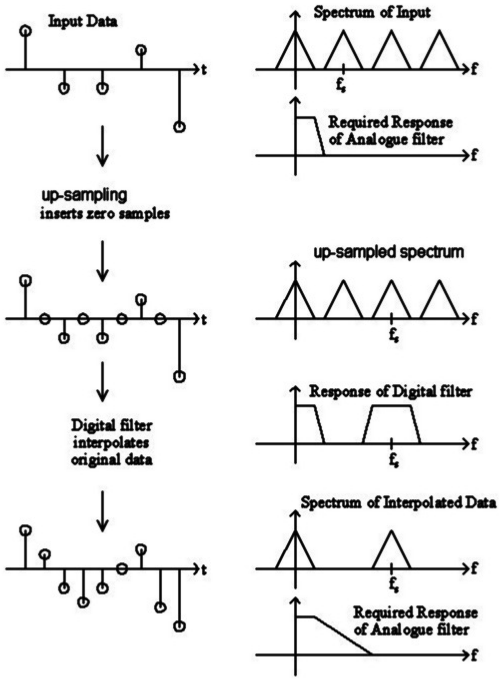
(e) Outline the potential impact on the amplitude response of representing the filter coefficients by finite-precision binary numbers. **Solution** The finite precision binary numbers are inevitably required by the computer to represent filter variables that may require greater precision than is available on a computer. An extreme case might be the representation of a variable using only an 8 bit binary number, however greater number of bits can often be used these days. Despite the availability of higher precision cheaper devices, the limiting effects of finite precision arithmetic can still affect the response of a filter, where the finite precision used to represent filter variables may make the filter frequency specification deviate from the intended design. The filter may even become unstable, depending on the overall effect of the finite precision representation.

(f) If the filter is to be implemented using a DSP chip which is capable of carrying out one MAC every 20ns, explain why the implementation can only allow a maximum of 1041 MACs per output sample.
Solution From the initial filter specification we can see that the sampling frequency is 48kHz.

- Time between output samples = $1/48000$ seconds
- Time to carry out one MAC = $20ns$
- Number of MACs possible in this time = $1041.666..$
- You cannot utilise a fraction of a MAC, so the maximum is 1041.

4. An FIR digital filter is required for an oversampled audio DAC. The audio was originally sampled at 44.1 kHz, the DAC is using 16-times oversampling and the processor implementing the filter takes 20 ns to carry out one multiply-and-add operation.

(a) Draw a diagram illustrating the process of oversampling in both time and frequency domains and use it to explain how oversampling works and why it is beneficial in this application. **Solution**



- Inserting zeros shifts the sampling frequency but does not change the frequency response other than the sampling frequency
- Digital filter removes spectral copies
- Allows the use of a lower-order analogue filter after the DAC

(b) Calculate the number of multiply-and-add operations the processor can use to work out the numerical value of each output sample of the digital filter and from this deduce the maximum filter order that can be implemented. Explain your logic clearly. **Solution**

- time between output samples = $\frac{1}{16\times 44100\text{Hz}} = 1.41723356...\mu$s.
- number of MACs in this time $\leq \frac{1.41723356...\mu\text{s}}{20\text{ns}} = 70.86...$ i.e. 70 MACs.
  Round down fraction of a MAC because fractional MAC is meaningless.
- Maximum filter length $= 70 \times 16 = 1120$ (multiply by zero data uses no MACs)
- Maximum filter order = 1119