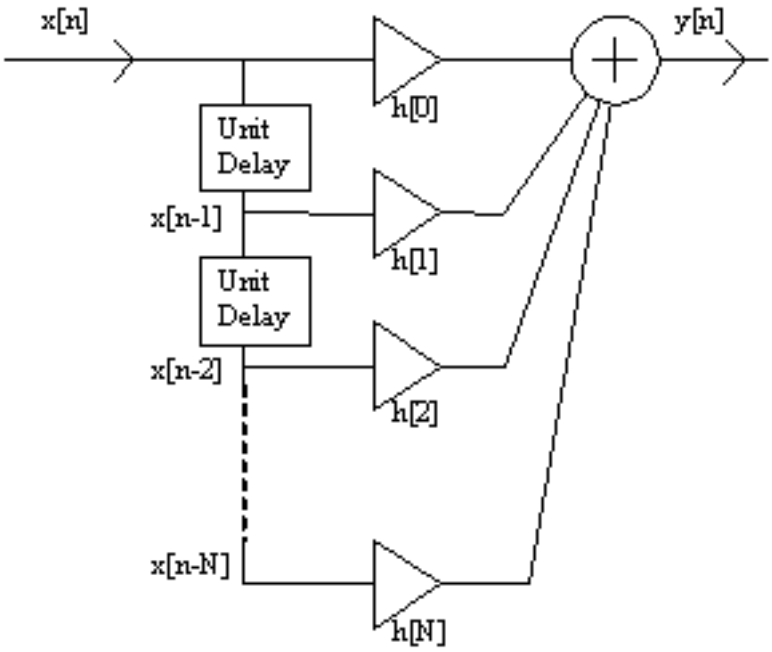
Tutorial Questions:
Finite Impulse Response Filters

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:

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



2. How do you find $H(f) = \sum_{k=0}^{N} h[k]e^{-j2\pi fkT}$ from $H(z) = \sum_{k=0}^{N} h[k]z^{-k}$

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.

   (b) Outline the consequences of the truncation on the amplitude response of the filter.

   (c) Outline the consequences of the delay on the amplitude response of the filter.

   (d) State the effects on the amplitude response of applying a window to the truncated coefficients

   (e) Outline the potential impact on the amplitude response of representing the filter coefficients by finite-precision binary numbers.

   (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.

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.

   (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.