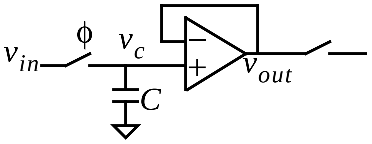
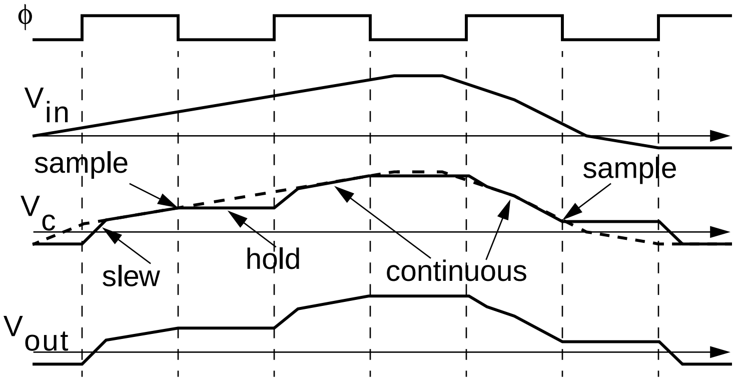
Sample And Hold



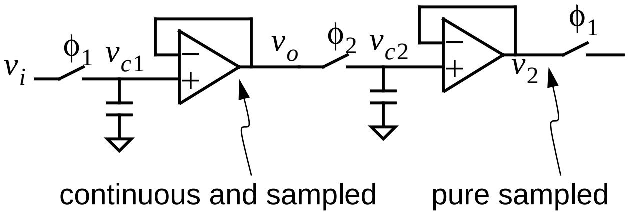
The sample is taken when ϕ is high, then V_c charges to V_{in}. When ϕ opens, V_c remains at the last sampled V_{in}, this is the hold phase. V_{out} follows V_c. The opamp is required so further stages can be driven without discharging C. Input resistance of MOS opamp is nearly infinite.

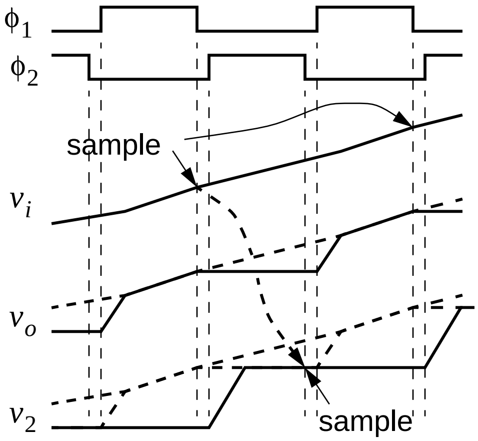


Right at the input, v_{in} may be a continuous analog signal.

v_c has continuous sections while ϕ is high and is held while ϕ is low. The held voltage is the voltage on v_{in} at the end of the ϕ high time, so that is the sample point.

The next stage will sample at the end of the “held” time and as a result, all contin-





Thus ϕ_1 samples and holds the value at the end of ϕ_1. Then ϕ_2 goes high and the constant value of v_o is sampled. The opamp goes through transients, etc, and at the end of ϕ_2 the sample is valid, and can be passed on to the

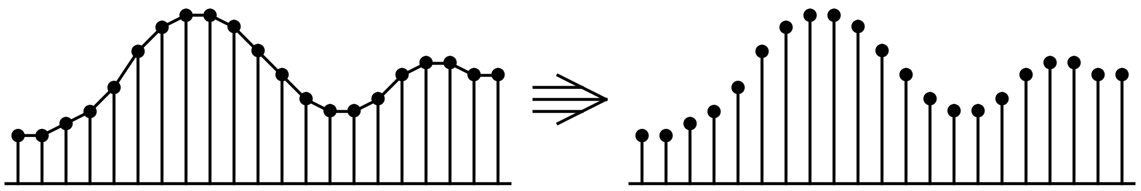
Thus the continuous analog input has been converted to a purely sampled output at v_2. Note that each stage has delay of half a clock period associated with it. In other words, when v_2 is valid, the voltage represents the input sampled half a clock period ago. Thus, we could say that $v_{2,n} = v_{i,n-1/2}$, where $v_{2,n}$ is v_2 at time now, and $v_{i,n-1/2}$ is v_i half a clock period in the past. The subscripts n, n-1/2, n-1 represent time now, half a clock period ago and 1 clock period ago.

Sampled Data, Replication, Sinx/x

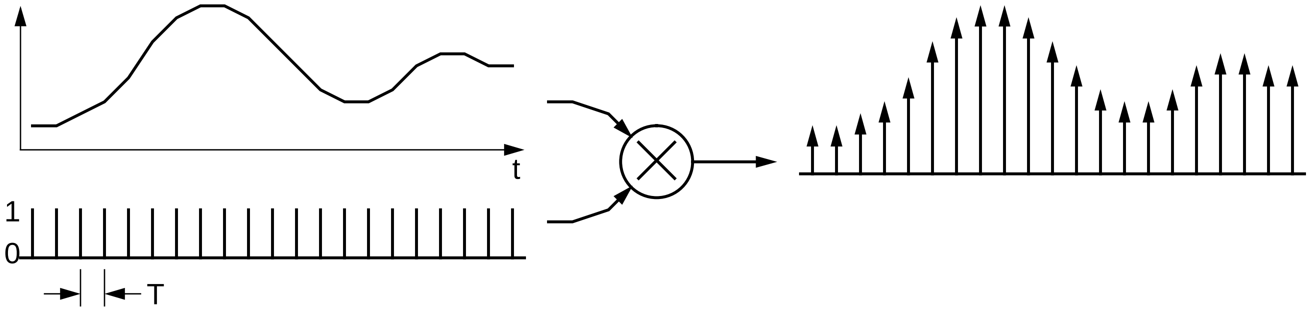
Conversion of a continuous signal into a sampled signal involves replication and aliasing. The conversion of a sampled signal into a continuous signal involves sinx/x.

Ideal Sampling

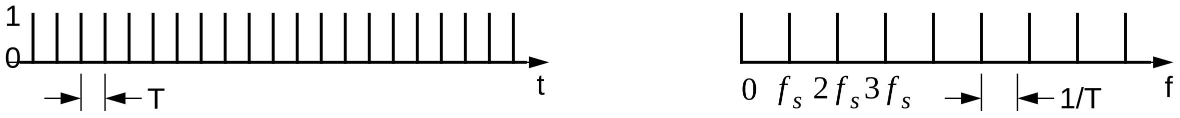
Consider the continuous signal being sampled



This is equivalent to multiplying the signal by an infinite succession of impulses of height 1 in the time domain.



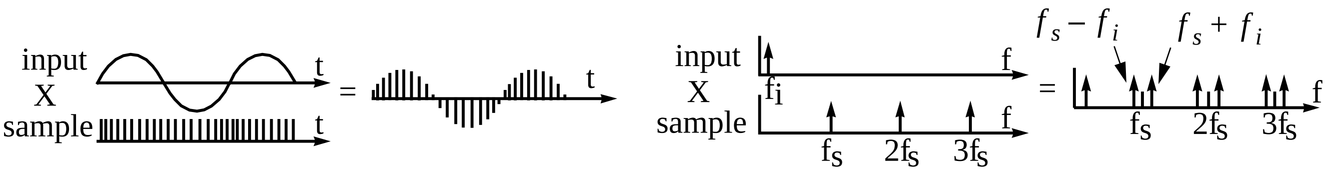
In the frequency domain, the series of pulses in the time domain is equivalent to an infinite series of pulses in the frequency domain separated by 1/T.



Thus, there is an impulse at DC, one at f_{clock} which above is labelled f_s for sampling frequency, one at $2f_s$, $3f_s$ etc. Each of these represents a sine wave at that particular frequency, either at DC or a multiple of the clock frequency. Multiplication in the time domain is equivalent to convolution in the frequency domain. Fortunately, convolution with a series of impulses is equivalent to multiplying the input spectrum by each of the impulses. Thus the input is multiplied by each multiple of the clock frequency producing sum and difference frequencies around each of the clock multiples. Note that multiplying two sine waves together is sometimes also referred to as mixing.

Replication

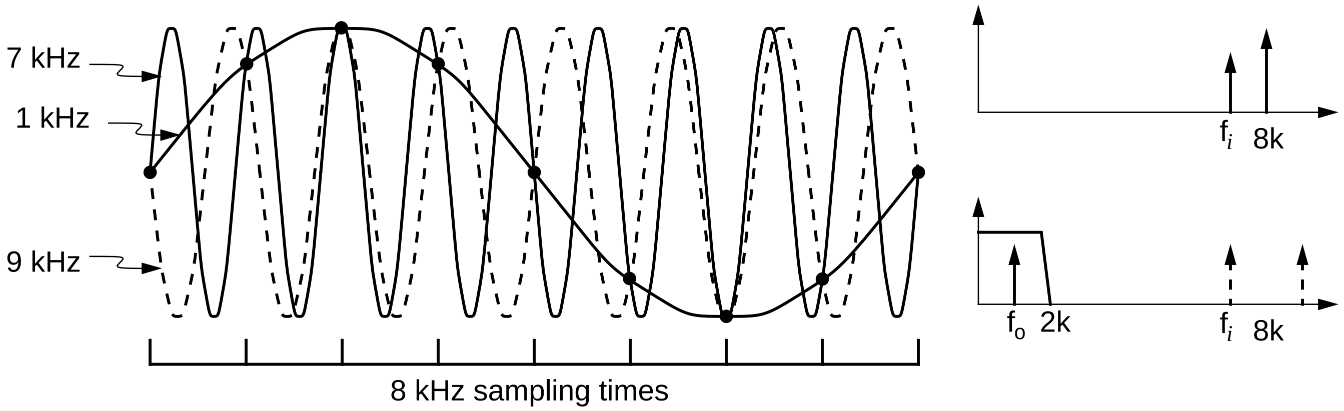
Pure sampling is equivalent to multiplying a continuous signal by a train of impulses in the time domain, separated by T. (Here, T is the period of the sampling frequency f_s.) The resulting spectrum in the frequency domain, is a set of impulses (frequency components) separated by 1/T, the clock frequency. The outputs occur at $nf_s \pm f_{in}$. This is called replication.



Thus an input signal, when sampled is replicated around multiples of the clock frequency (To infinity for ideal sampling). Note that at the output, one cannot tell which of the frequencies was the actual input frequency. For example, 1 kHz sampled by 8 kHz replicates to 1 kHz, 7 kHz, 9 kHz, 15 kHz, 17 kHz etc. An input signal at 15 kHz would have exactly the same output spectrum.

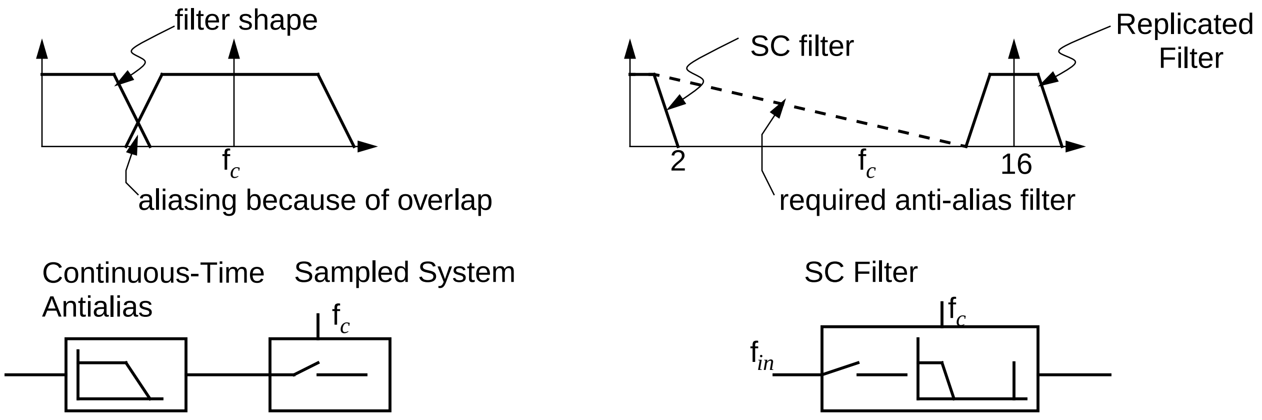
Aliasing

If a S.C. filter is built at baseband, frequencies input close to $n f_s$ will replicate into the baseband and will be indistinguishable from signals input at the baseband. These replicated signals are said to be aliased. As an example, if we have a system clocked at 8 kHz intending to pass 0 to 2 kHz, then for some component at 7 kHz would appear at the output as 1 kHz and would corrupt the desired input. This is an aliased component (alias meaning pretends to be, or assumes the name of, or an imposter)



Note that the points shown above could have come from the 7 kHz signal, the 9 kHz signal or the 1 kHz signal. There is no way of knowing. Note also that all of the amplitudes are the same. On the right is a sketch of an input at 7 kHz, sampling clock at 8 kHz. The result will be 1, 7, 9 etc. However there is a baseband filter so only 1 kHz makes it through. This is an aliased component as it has become a 1 kHz signal after starting life as a 7 kHz signal. This signal will now fall on top of any desired 1 kHz signals.

System View of Aliasing

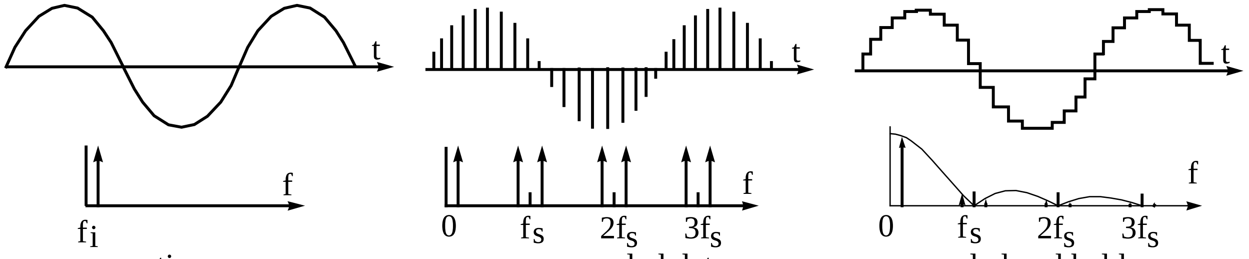


The systems view is shown on the left. The commonly quoted rule for this system is: - Filter shape should not extend past $f_c/2$, then there will be no aliasing. This is very important when trying to design a filter to cut something off at $f_{c/2}$, usually a continuous time filter.

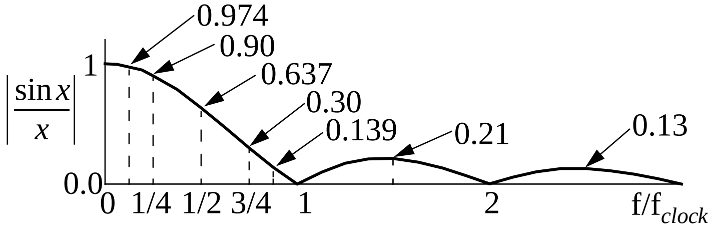
However, as shown on the right for a switched-capactor filter, the filter itself is sampled typically with a clock frequency quite a bit higher than the filter corner frequency. For example with a switched capacitor filter with cutoff at 2 kHz and $f_c = 16 \text{ kHz}$, any frequency in the range of 14 kHz to 18 kHz will alias into the filter passband, for example, a frequency of 15 kHz will be aliased to 1 kHz. Hence these frequencies must be removed. The required anti-alias filter must remove frequencies past 14 kHz. Note that 13 kHz at the input would be replicated back to 3 kHz, however, this would not make it through the filter, so is not a problem.

sin x /x

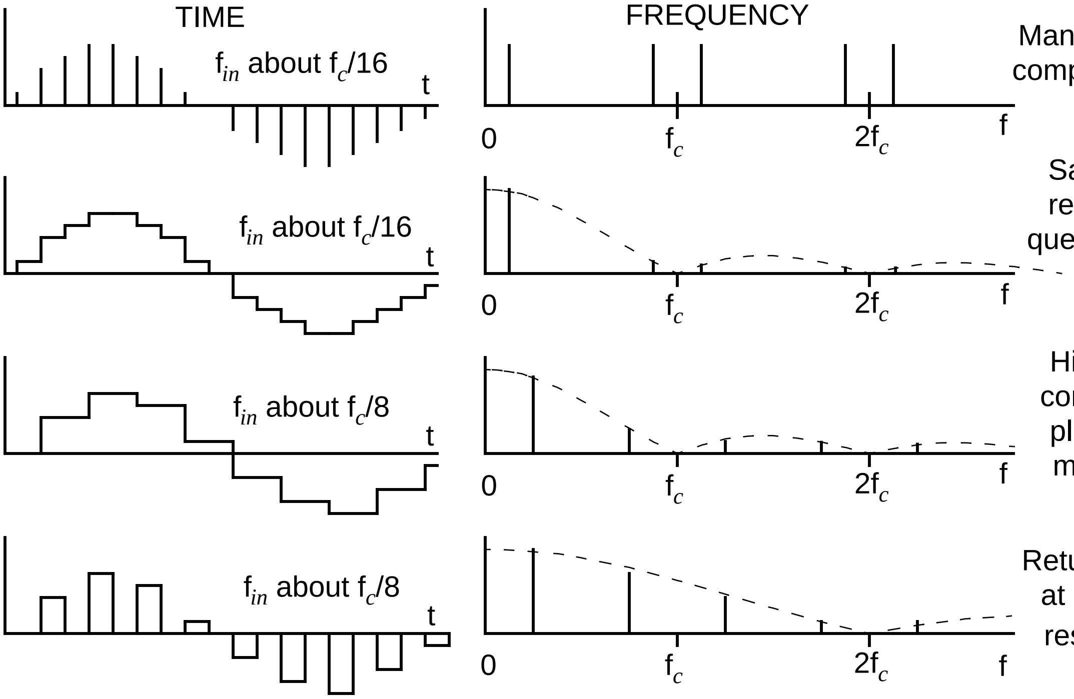
The sampled data world exists mathematically with ideal zero width samples. When viewed with a scope or spectrum analyzer, it sees the continuous world. In Switched-Capacitor world, this is a typically sampled and held. Pure sampled data when converted back to the continuous domain by being held for 100%. It can be shown that this results in the pure sampled data response being multiplied by $\frac{\sin x}{x}$, where $x = \pi\frac{f}{f_s}$. See examples below and on next page.



continuous pure sampled data sampled and held



This gives automatic lowpass filtering when we convert to the continuous domain
So, how does one demonstrate the effect of sinx/x? One way is to examine the output spectrum, and not that as the input frequency is closer to DC, the replicated components are closer to the clock frequency and hence they are attenuated more by sinx/x. In the time domain, this can be seen by noting that the steps in the output waveform represent higher frequency components. These steps get smaller as the input frequency gets closer to DC indicating more filtering.



Many high frequency components to infinity

Sample and hold reduces high frequency components by sinx/x

Higher input freq compared to sampling freq, result: more harmonics

Return to zero, null is at $2f_c$ instead of f_c, result, higher harmonics