ADC/DAC Basics

by Erol Seke

For the course "**Communications**"

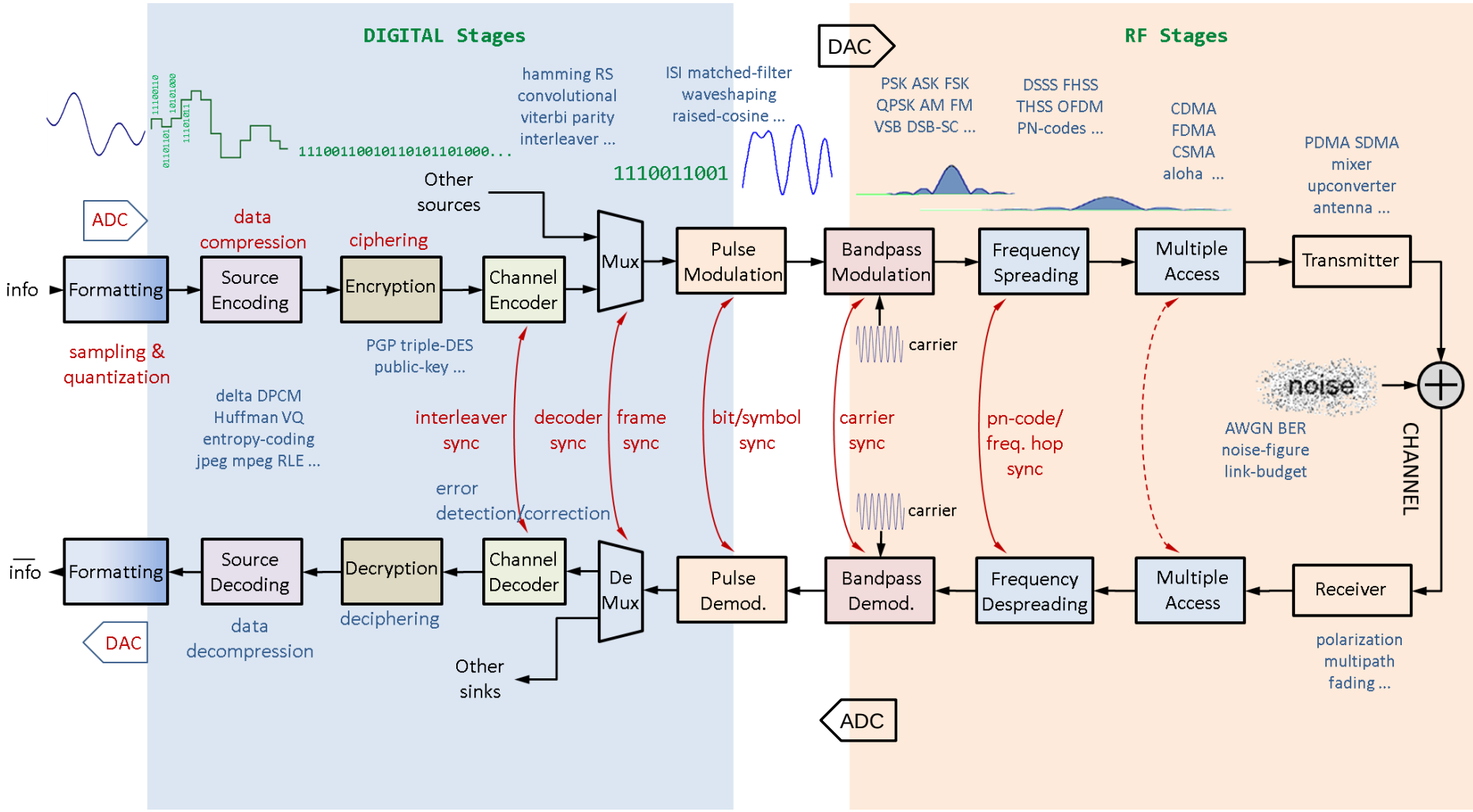
ESKİŞEHİR OSMANGAZİ UNIVERSITY

Where do we have them in Comm. Sys.?

1. When we want to do processing digitally (ADC).
    a) Analog message signal, digital transceiver.
    b) When processing hard to do in analog like data compression, encryption, channel coding, FFT.
    c) We want to make it easy to upgrade.
2. Signal needs to be analog (DAC).
    a) Some simple processes are challenging to do in digital, like HF modulation.
    b) Bandwidth requirements
    c) The message signal is analog, like voice signal.
3. It is generally easier to design in digital theoretically but challenging to implement.

«Requirements vs Cost» dictates the general design/implementation.

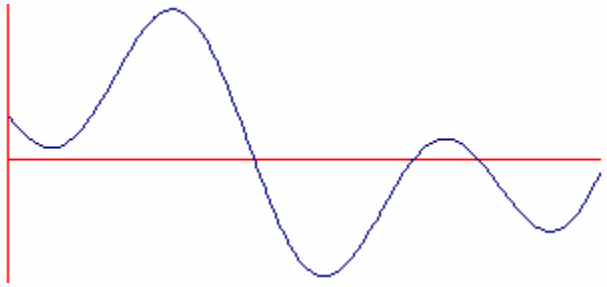
General Communication System



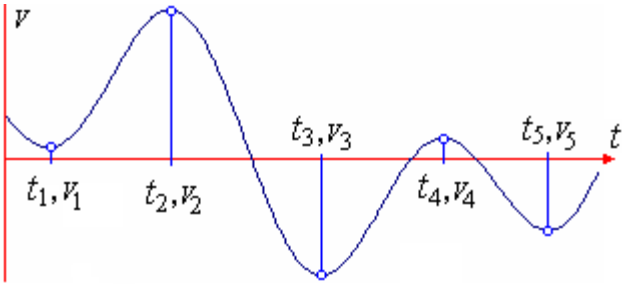
What does an ADC do?

Problem : We have a signal at the hand which we would like to replicate at another location, but we are not allowed to transmit it.

Is it possible? : Someone looks at the signal and tells about the shape of it to someone at the other end and tries to make him construct the signal. ☺



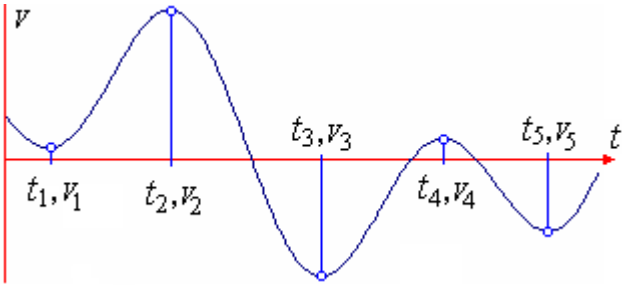
Example : Tell some values at some critical points. How about min-max points?



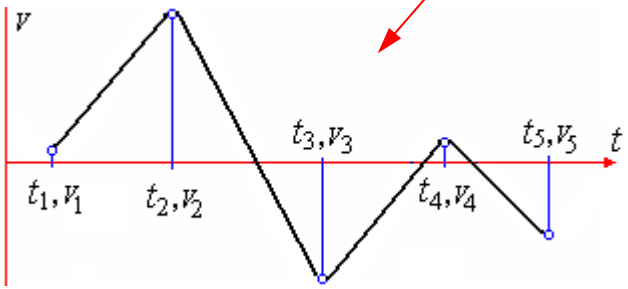
…and transmit time and value

$$(t_1,V_1)(t_2,V_2)(t_3,V_3)(t_4,V_4)(t_5,V_5)\ldots$$

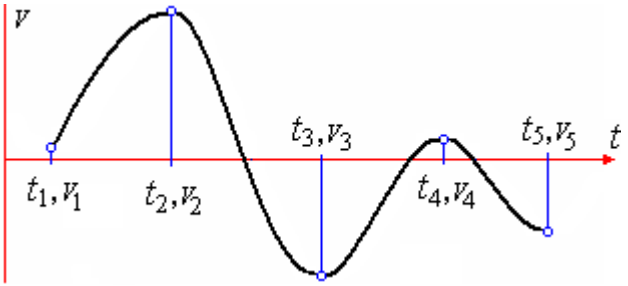
…and reconstruct the signal from its irregular samples.



reconstruct from samples



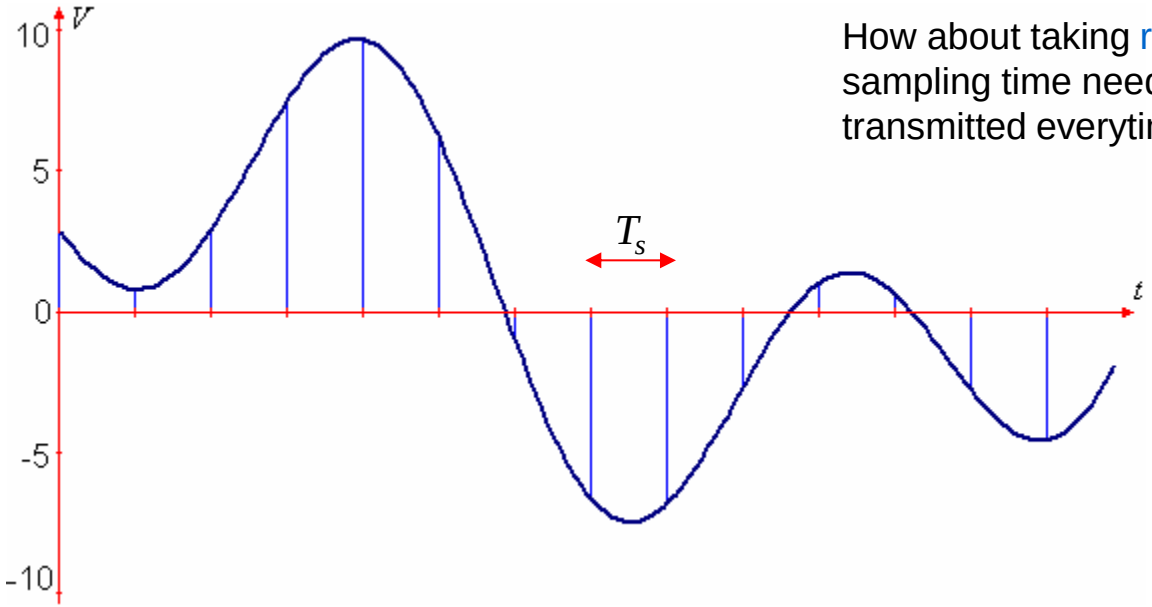
linear interpolation

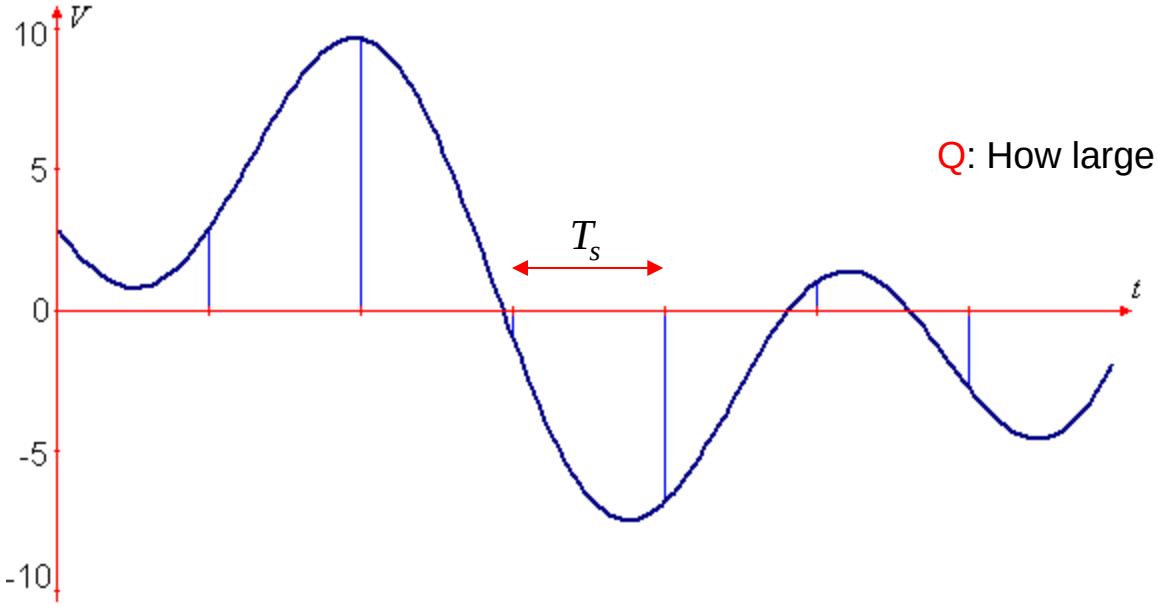


Higher order curve fit techniques

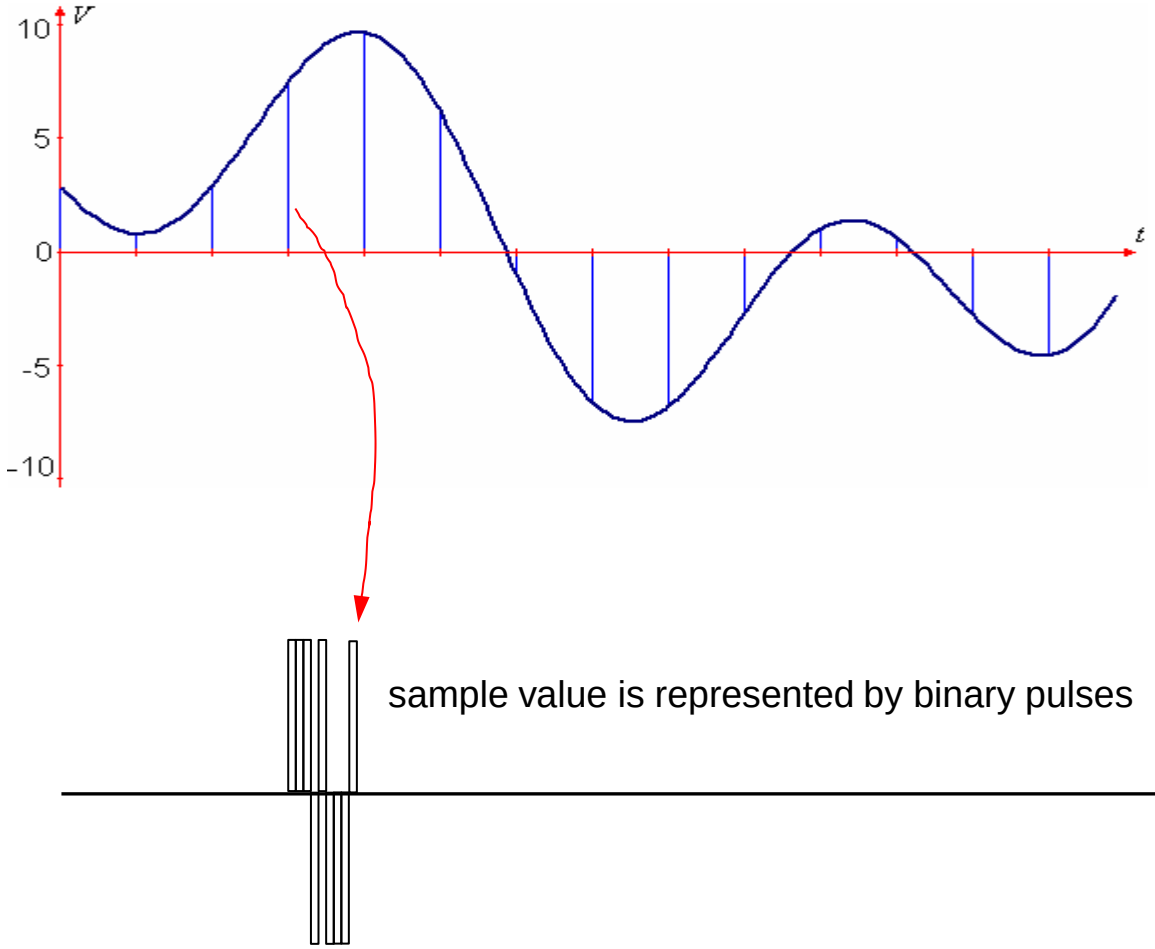
Any difficulty ? ☹

How about taking regular samples in which sampling time need not be measured and transmitted everytime a sample is taken?





Q: How large the intervals can be?

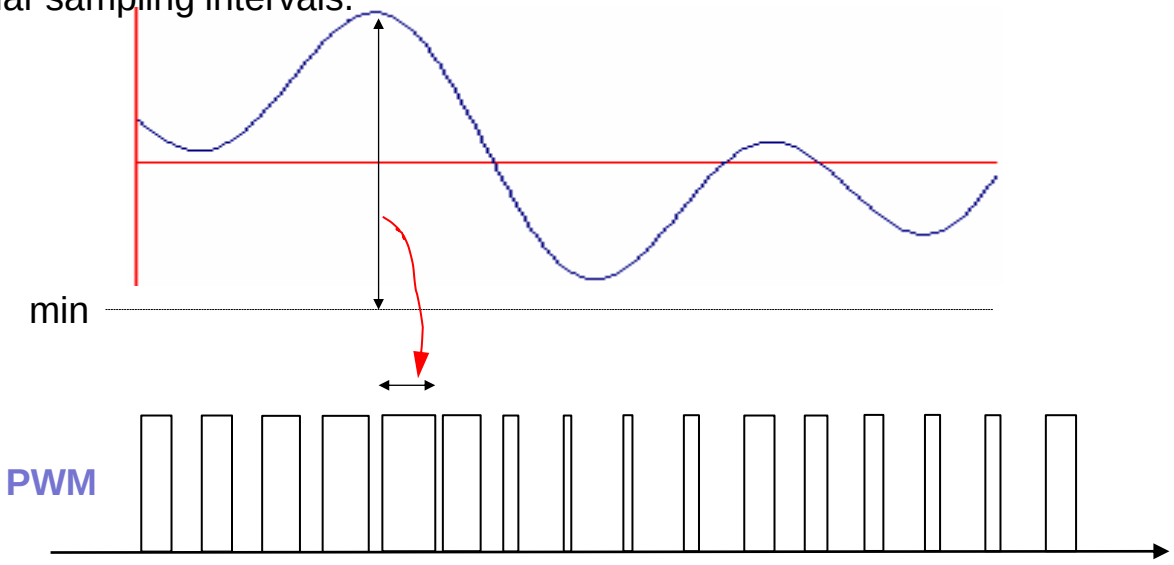


PCM

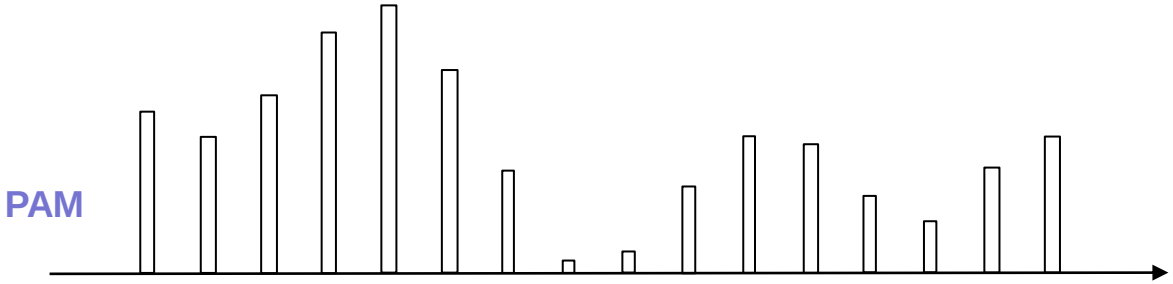
Most known binary pulse code system is the “straight binary”

Some Other Pulsed Representations

Let us assume that we have an analog baseband signal that we want to represent using pulses with regular sampling intervals.



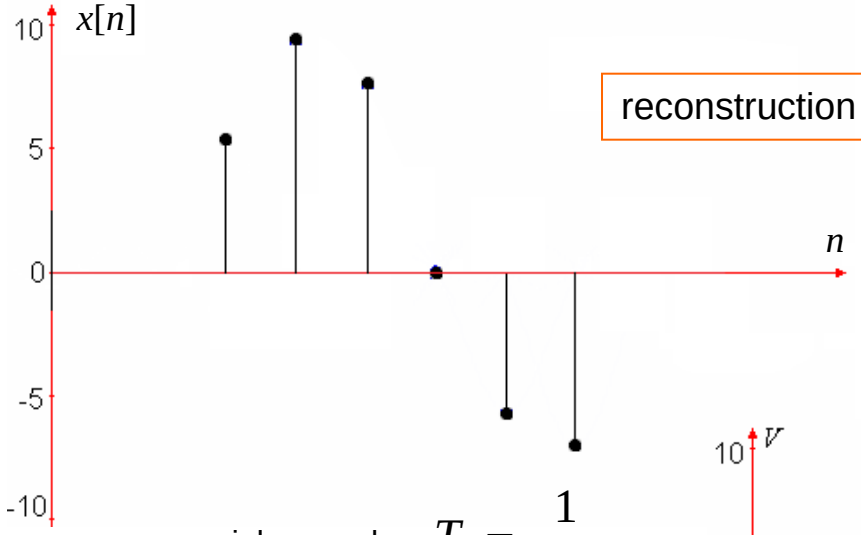
pulse widths represent the corresponding sample value



pulse heights represent the corresponding sample value

Nyquist's sampling criterion : A baseband signal with the highest frequency component at f_m can be reconstructed from its samples if the interval between samples is less than $1/2f_m$

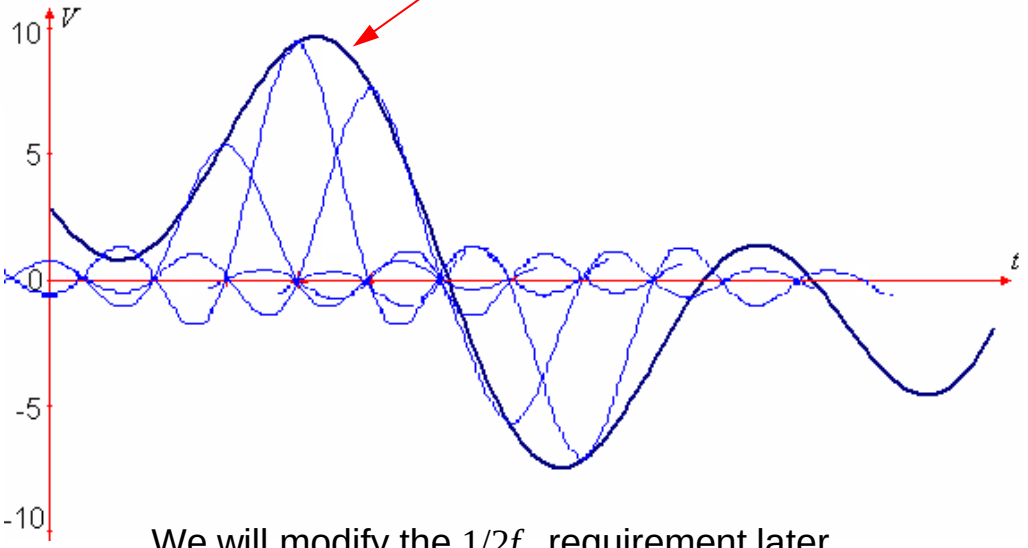
That is; sampling frequency must be higher than twice the highest frequency of the signal



reconstruction formula $x(t) = \sum_{n=-\infty}^{n=\infty} x(nT_s)\,\mathrm{sinc}(2f_m(t-nT_s))$

The sum of all **sinc**s

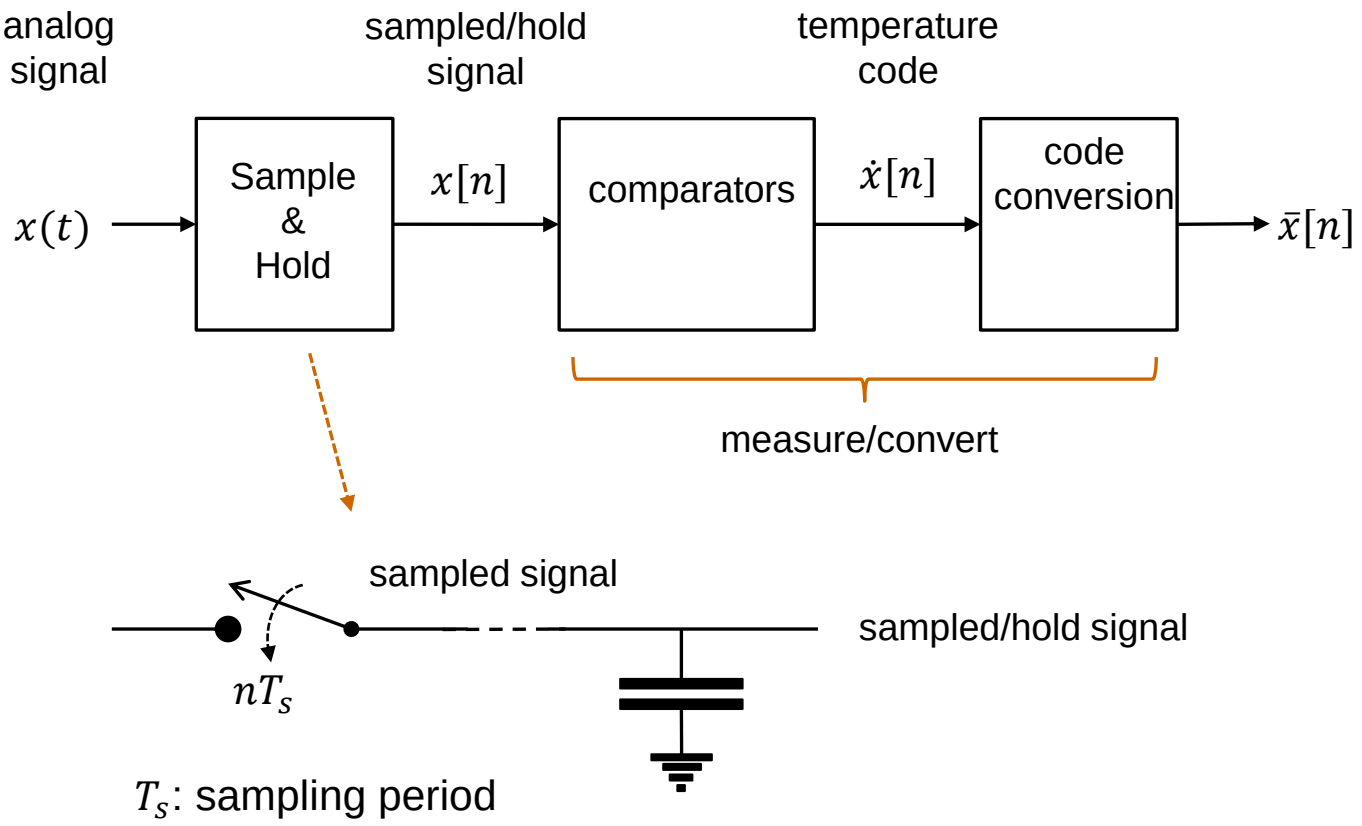
a special case when $T_s = \dfrac{1}{2f_m}$



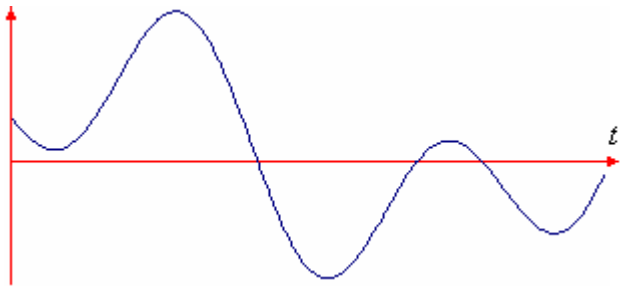
truncated sinc function

We will modify the $1/2f_m$ requirement later.

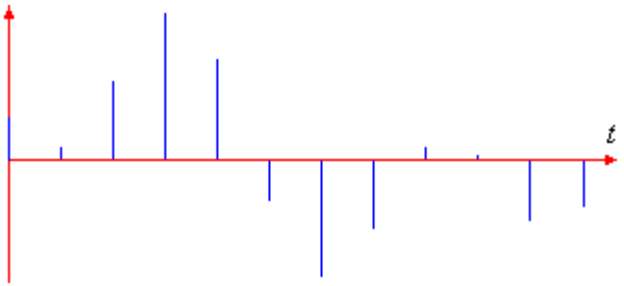
Basic ADC Model



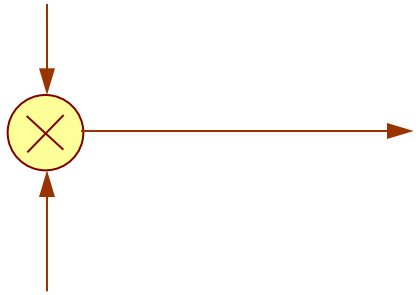
Original signal

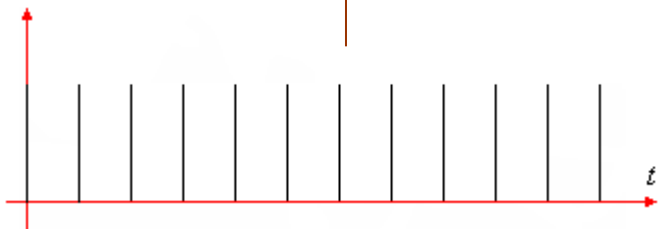


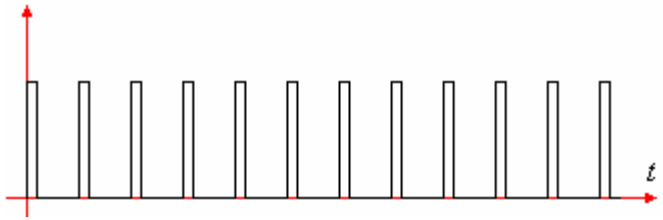
Sampled signal



Sampling signal
(impulse train)

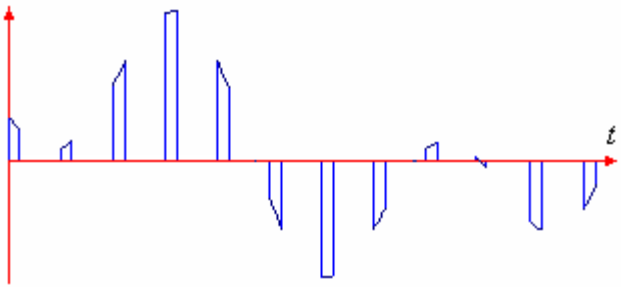




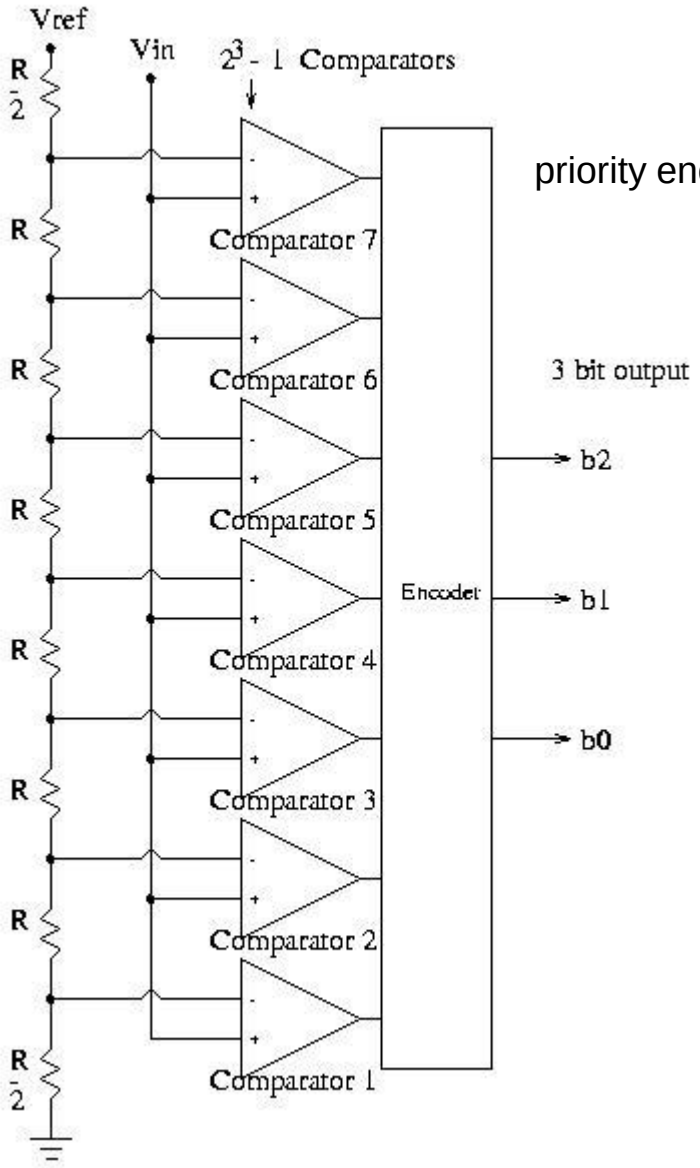


more realistic sampling signal

the signal values when the measurement is initiated and ended may be different



Parallel Analog-To-Digital Conversion (measure/convert)



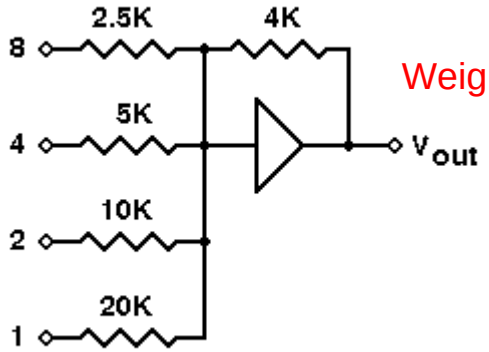
resistor bank
(voltage dividers)

priority encoder

3 bit output

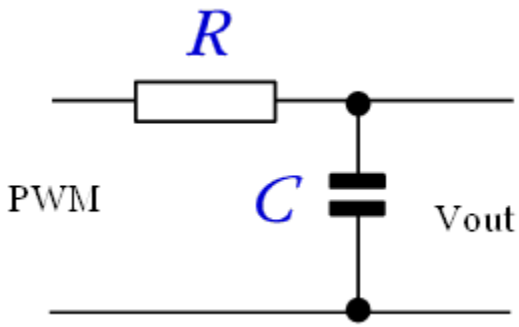
Q: How many comparators are needed for 14-bits conversion?

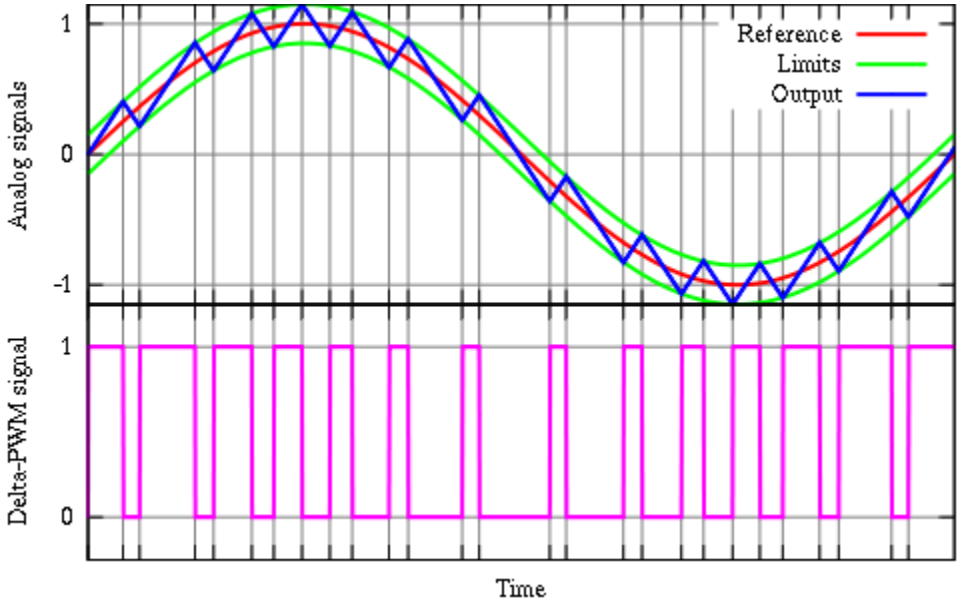
Digital-to-Analog Conversion



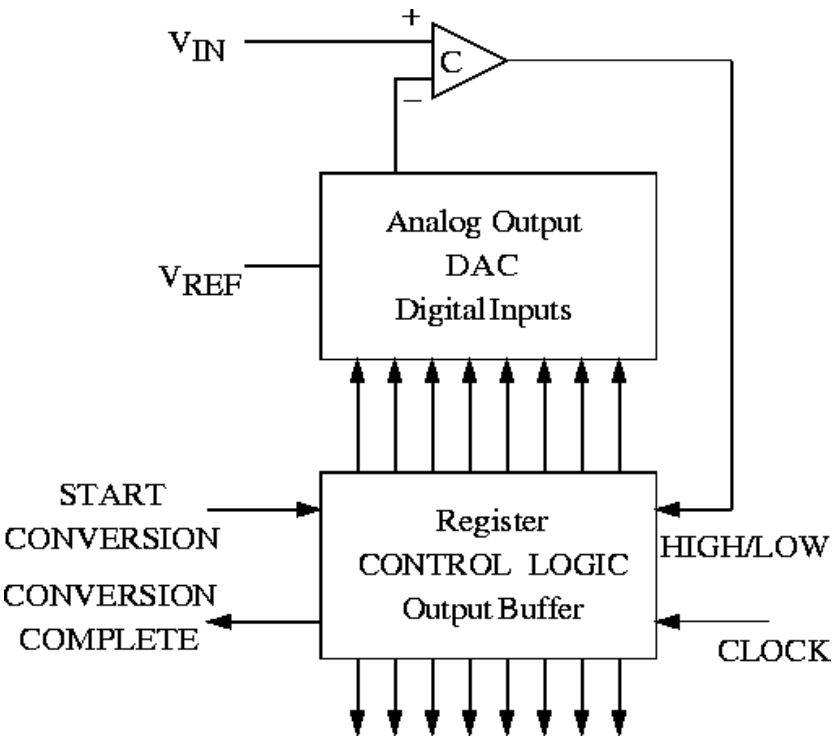
Weighted adder method

PWM method

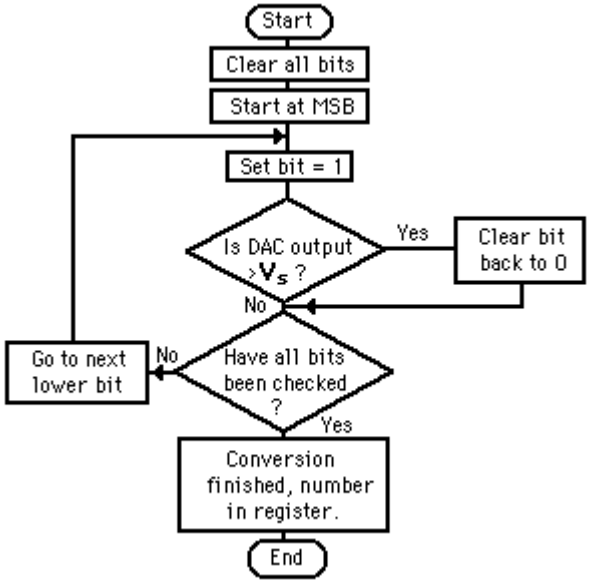




Analog-To-Digital Conversion via Successive Approximation



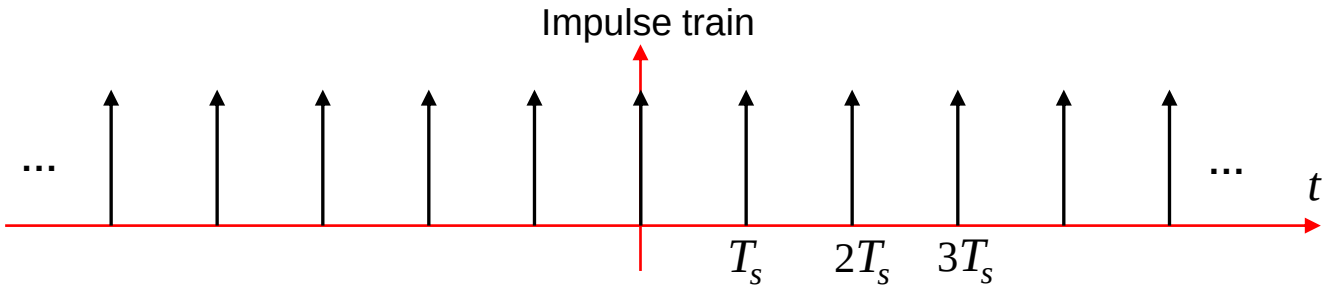
algorithm

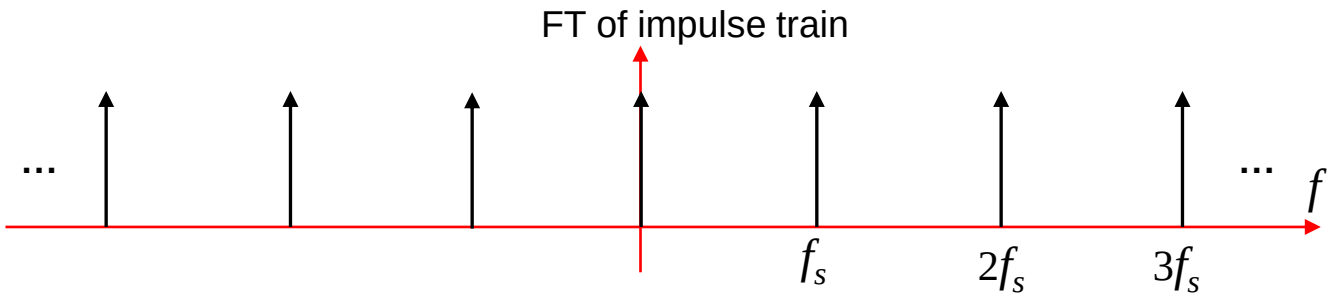


1. Set msb to 1
    1. convert to analog V
    2. if V>Vin revert msb
2. Set msb-1 to 1
    1. convert to analog V
    2. …
3. repeat this until after lsb.

Since DAC is cheaper, SA-ADC is common for low speed conversions.
(at least b clocks required for a conversion)

Spectral Considerations



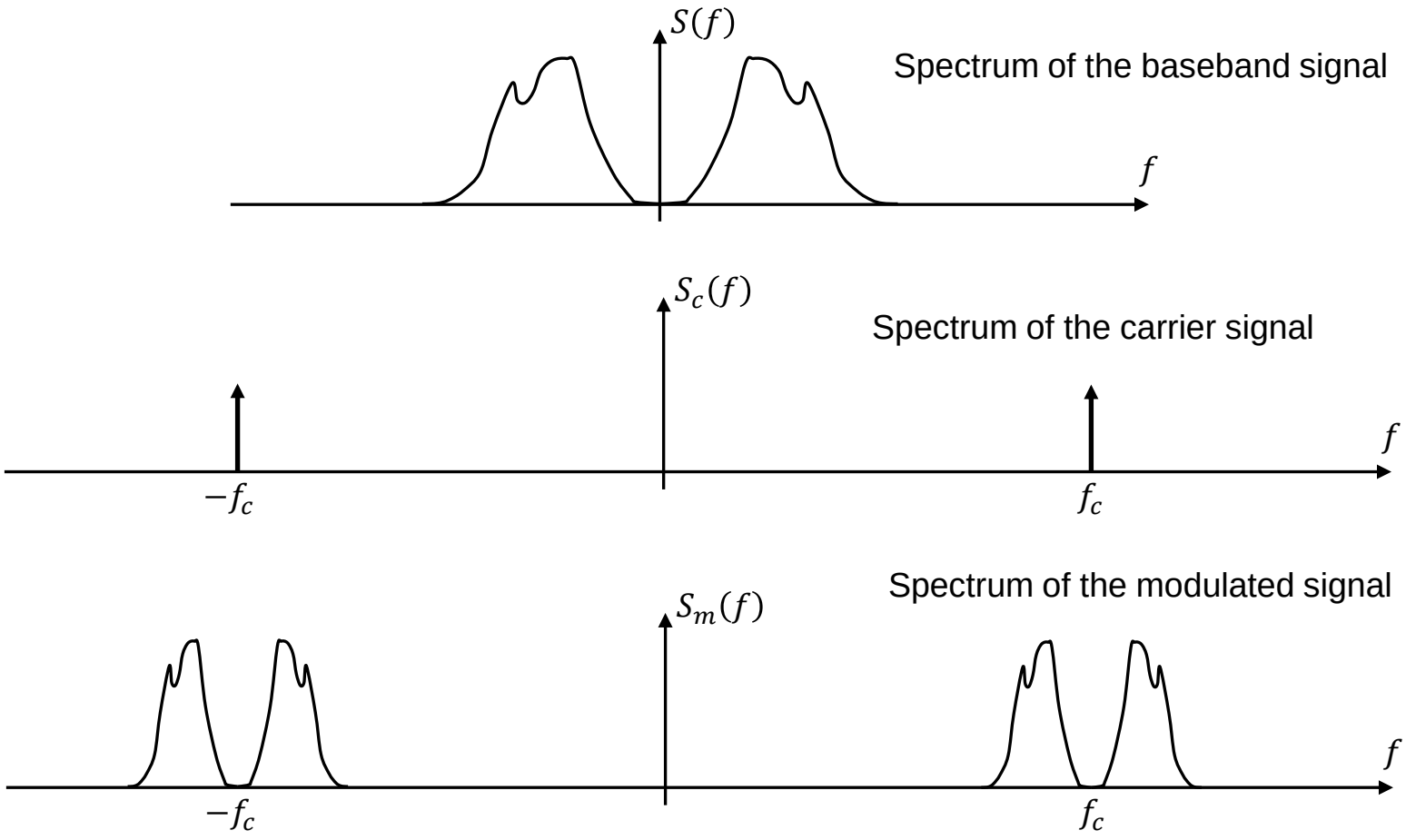


$$f_s = 1/T_s$$

inversely proportional : remember $\mathcal{F}\{x(at)\} = \frac{1}{|a|} X\left(\frac{\omega}{a}\right)$

Hmw: Define impulse train and prove that FT of that is also an impulse train in freq. domain.

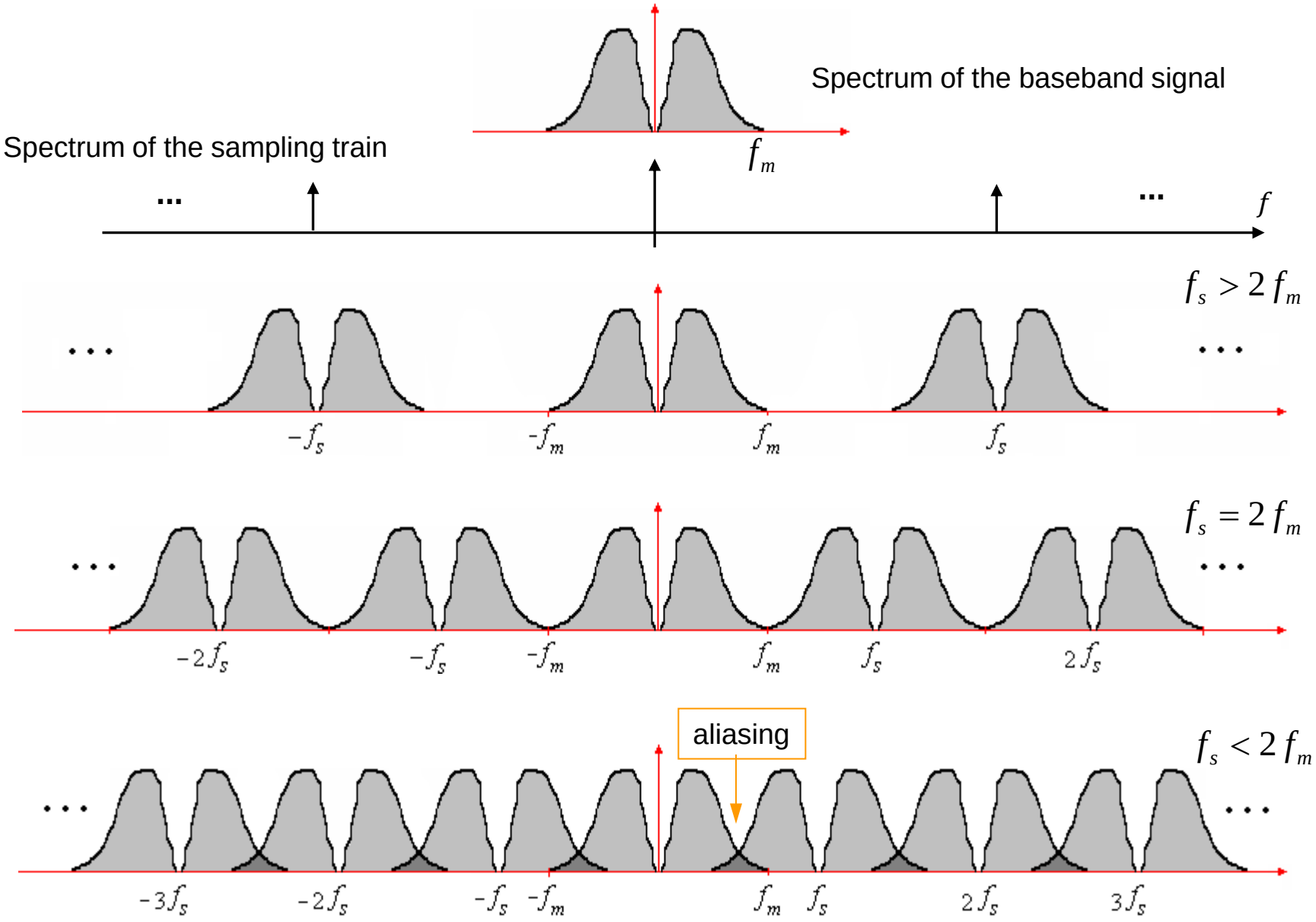
Remember the Modulation Property of FT



Instead of two impulses in Fourier domain, now we have infinite number of impulses.

(next page)

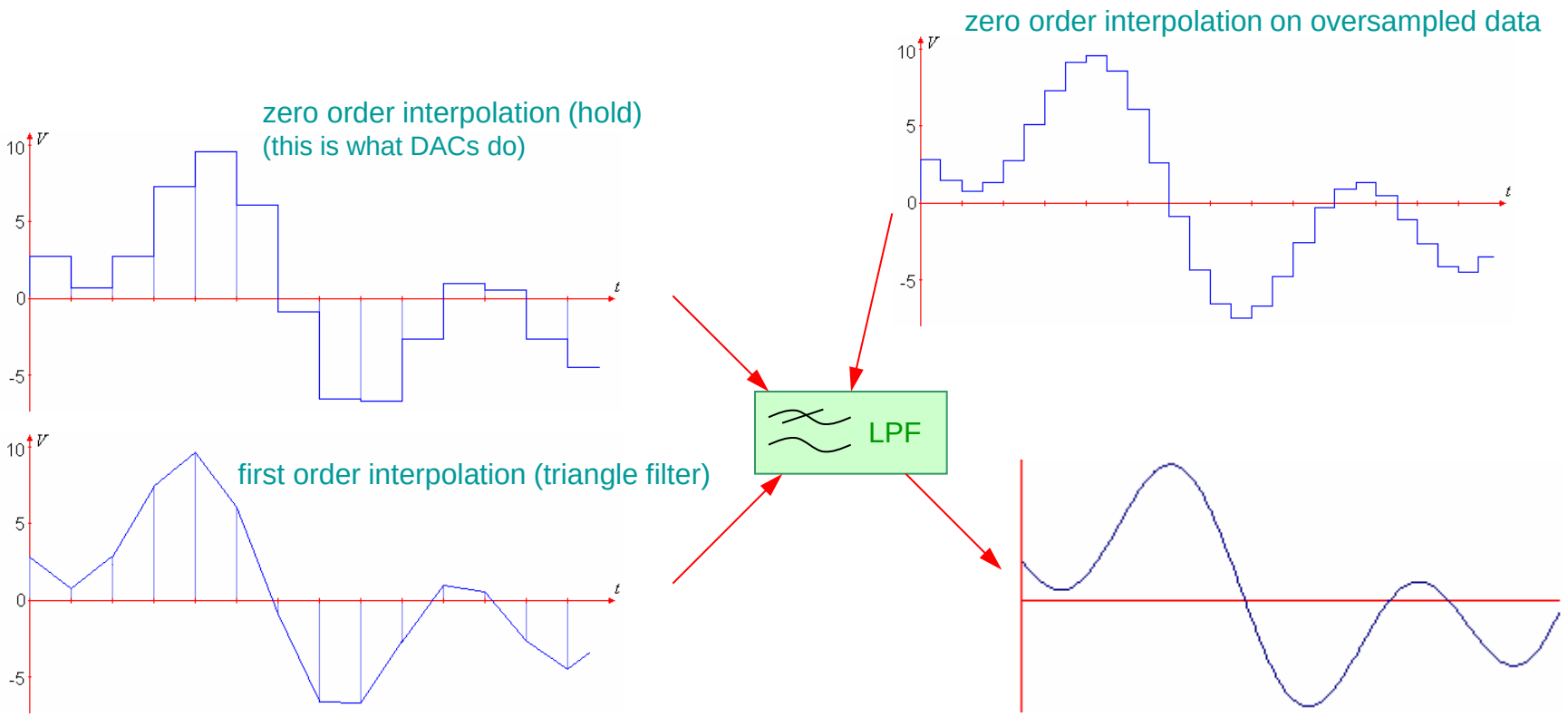
Where does the requirement $f_s > 2f_m$ come from?



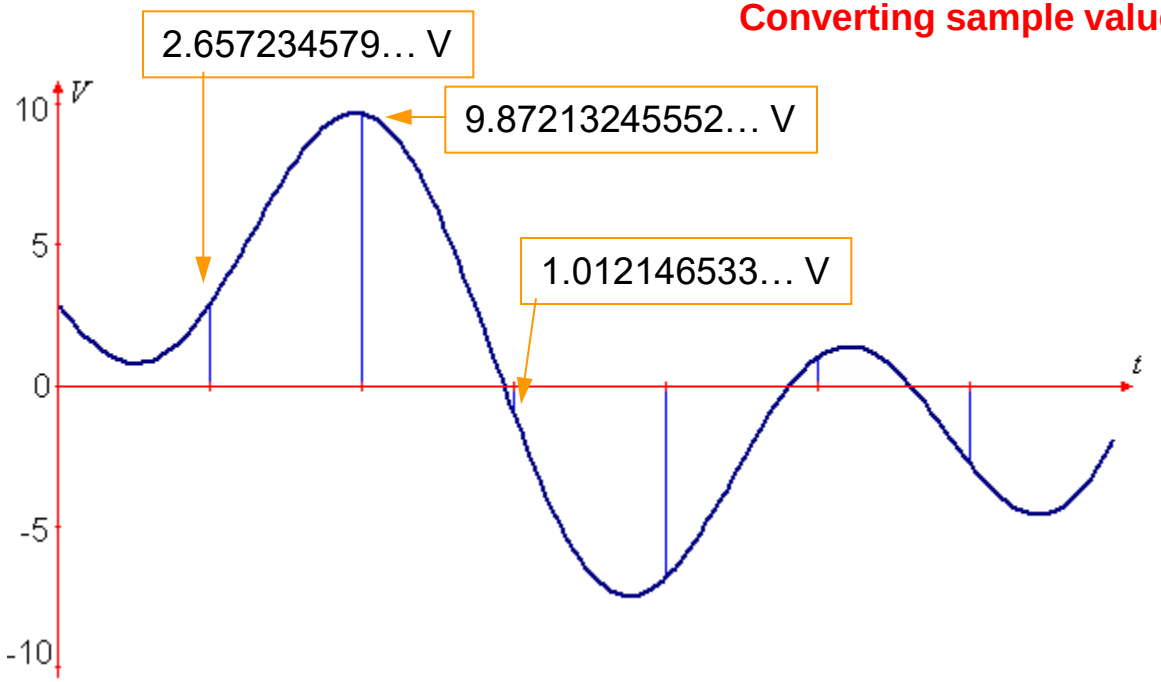
Ways to reconstruct the baseband signal from its samples

The actual problem is to “fill in the blanks” between the sample values and make it look like the original signal as good as possible. The ideal way is to use an ideal LPF which passes no components beyond f_m and disturbs none of the lower components. Ideal filter (Π), however, has infinite time duration (sinc). Truncated sinc is still impractical in most cases.

Filling the gaps some other way and passing the signal through an ordinary LPF is practical but that requires higher sampling rates than Nyquist rate (oversampling).



Converting sample values to digits



Transmit the sequence of the sample values as 2.657234579… V 9.87213245552… V …?

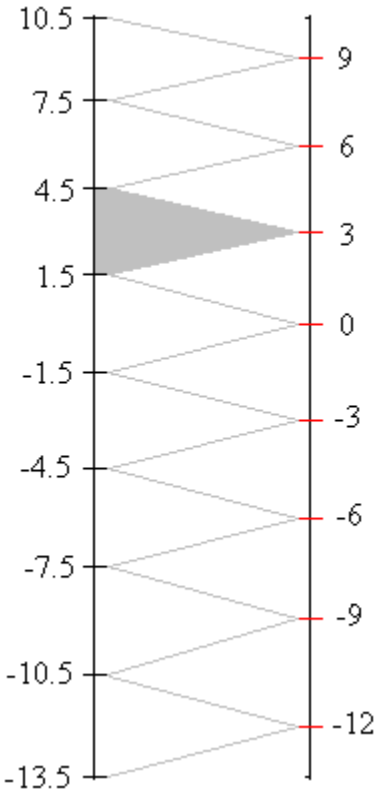
Is it getting costlier than transmitting the original signal ?

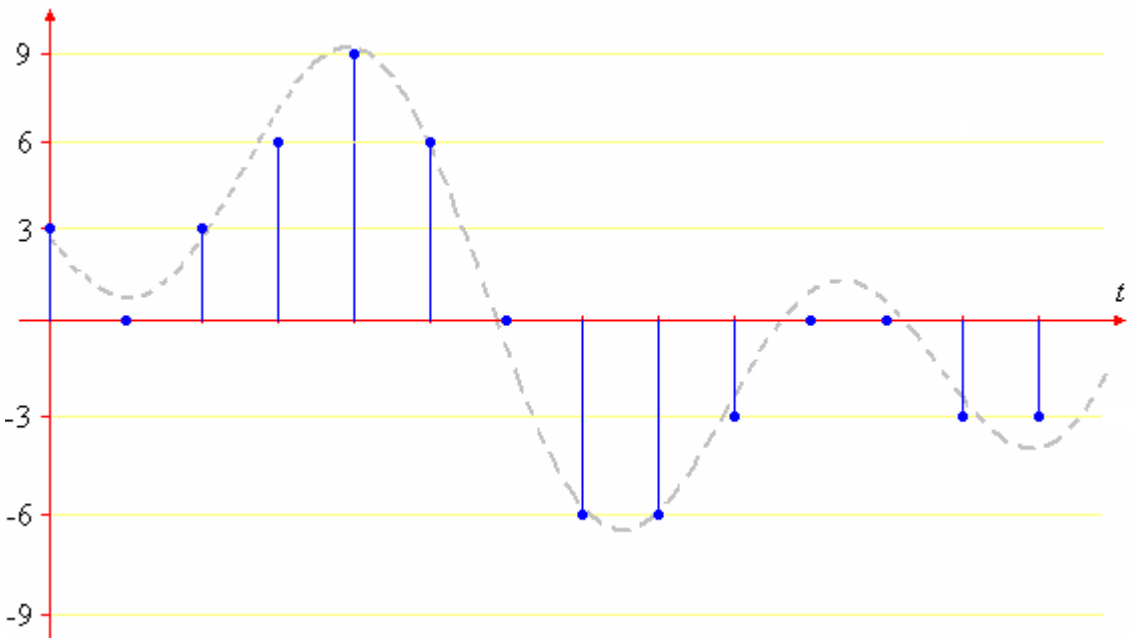
It looks like we have to give up some precision… and possibly transmit 2.6… V 9.8… V etc.

The truncation done is called **quantization**. (Nyquist's rule is **not** involved here)

Let us divide (-13.5,+10.5) range into 3 V sub-ranges and map all values in a sub-range to a single value

…and instead of transmitting 1.71323426148… V let us transmit 3 as shown





obviously there are some differences what the sample values are and what they are going to be after the truncation/quantization. Because of this differences, the original value can never be recovered.

The transmitted sequence would be 3, 0, 3, 6, 9, 6, 0, -6, -6, -3, 0, 0, -3, -3 for the above example

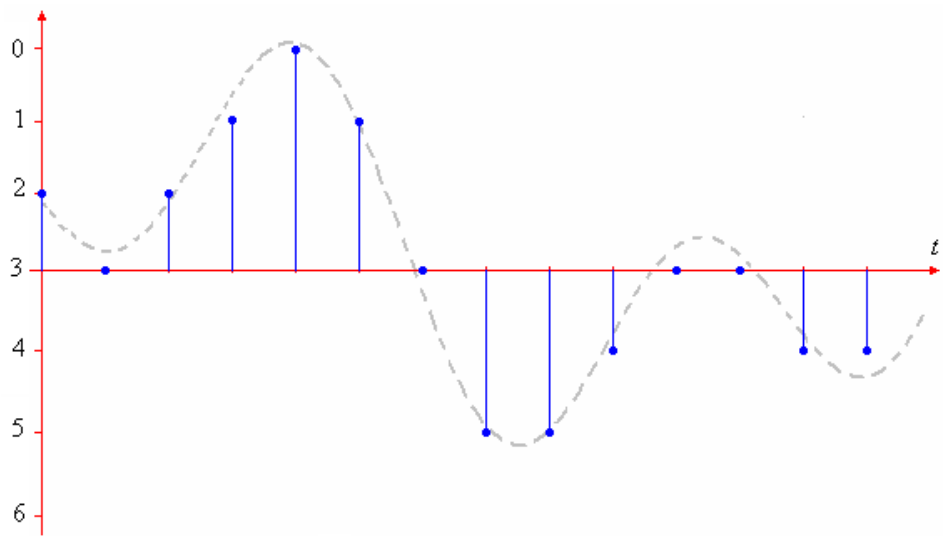
In order to transmit the value of 9 in binary we would need 4 bits,
however for -9 5-bits are required (one for the sign).
For generality, we would say that all values in the set {-9, -6, -3, 0, 3, 6, 9} can be transmitted by using 5 bits each. This is a terrible waste of bits.

Instead, we can assign an integer for each possible value so the set {0, 1, 2, 3, 4, 5, 6} would represent all the possible values. Instead of transmitting the actual values, we transmit the corresponding integer which is merely 3 bits.

$$\begin{bmatrix} 9 & \to & 0 \\ 6 & \to & 1 \\ 3 & \to & 2 \\ 0 & \to & 3 \\ -3 & \to & 4 \\ -6 & \to & 5 \\ -9 & \to & 6 \end{bmatrix}$$

So the sequence is 2, 3, 2, 1, 0, 1, 3, 5, 5, 4, 3, 3, 4, 4 using the mapping

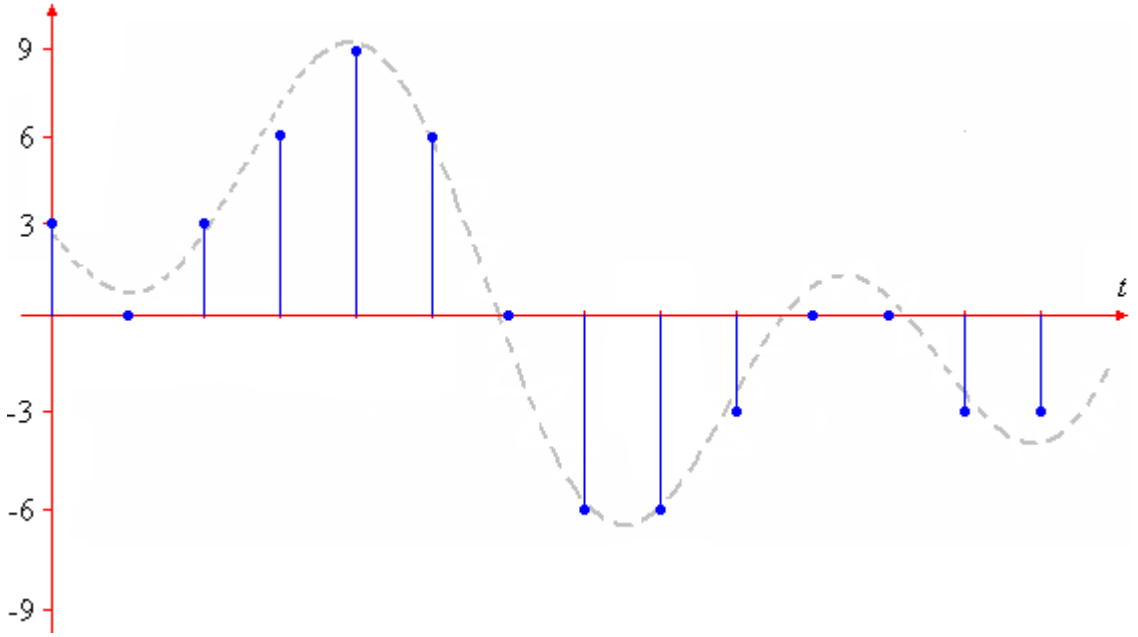
provided that the receiver knows the inverse of the map.



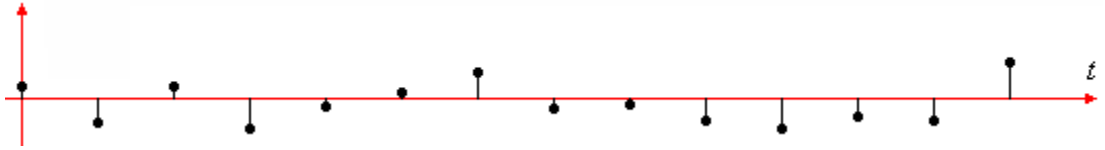
It is obvious that the number of quantization levels is usually chosen to be a *power of 2* so that bits are used most efficiently.

Once the receiver gets the integer number, it first converts it to its original scale which might be just another integer or a real number, but the truncated portion is lost and cannot be recovered.

Samples with some error



Quantization noise



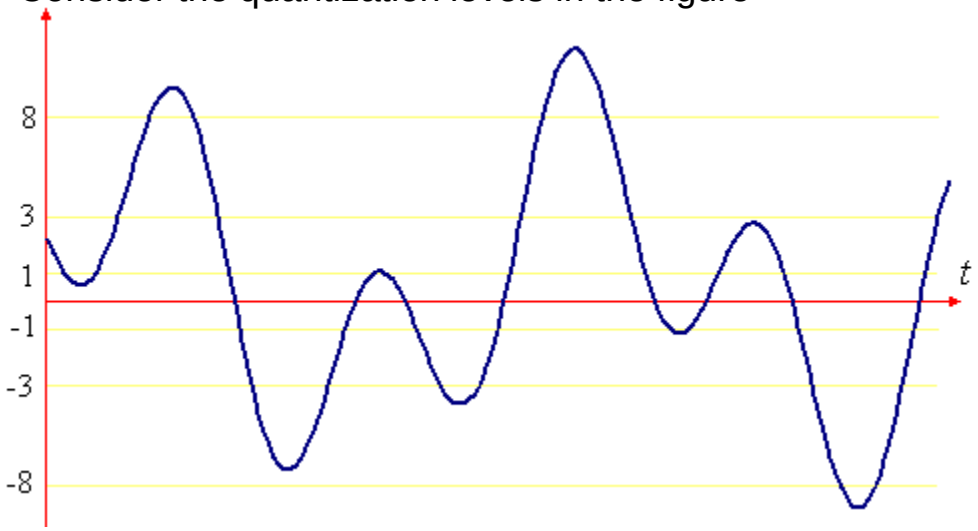
More quantization levels mean less quantization noise

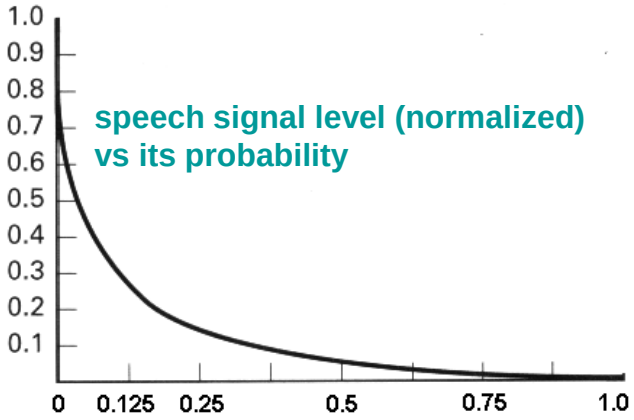
One more bit reduces quantization error to half

Uniform quantization levels for voice signal samples are not quite efficient, because

1. Voice signal statistics say that the signal roams around zero, most of the time. Uniform quantization would waste many quantization levels for high values although the signal rarely assumes such values.
2. One could not tell the difference between two high levels by just hearing them, nor it is necessary to distinguish them if we are just trying to understand what is being spoken (phone system, for example).
3. We need more quantization levels in lower levels in order to tell apart low signal samples. We wouldn't understand what is being whispered, otherwise. (now that's a big problem in a phone conversation)

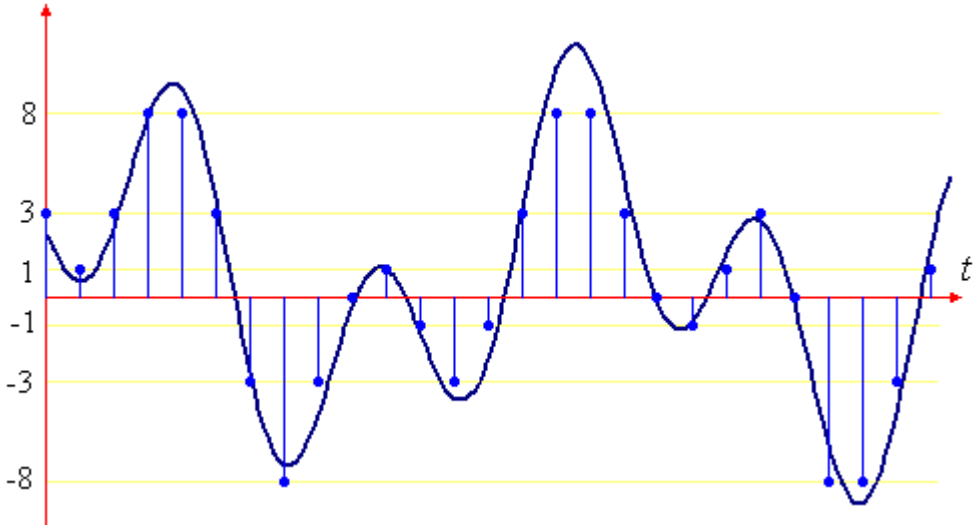
Consider the quantization levels in the figure





We have same number of levels but low voltage levels are more accurately measured than the high levels.

Such a schema would allow low levels to be measured more accurately.
But this increases the quantization errors made in the measurement of higher level samples.

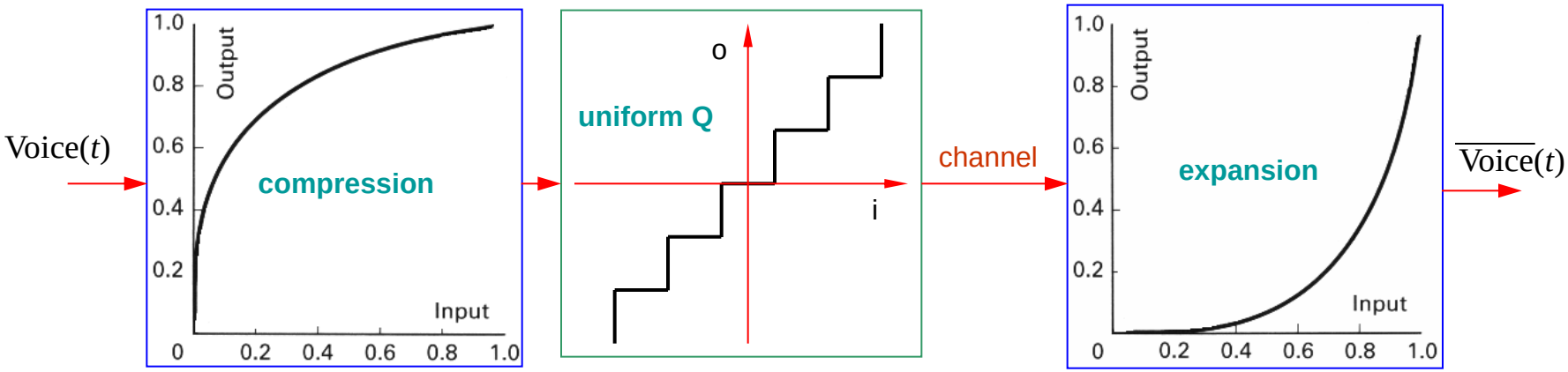


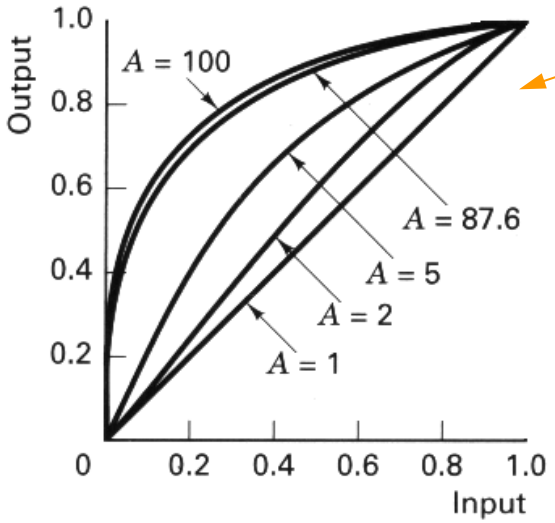
Since the signal is mostly around (and close to) zero, on the average we reduced the total error.

Voice sample statistics can be analyzed and quantization levels can be selected accordingly so that minimum amount of total error is achieved.

But, it is difficult (or meaningless) to manufacture Analog-to-Digital Converter chips (which will generate binary values corresponding analog voltages) with such non-uniform levels

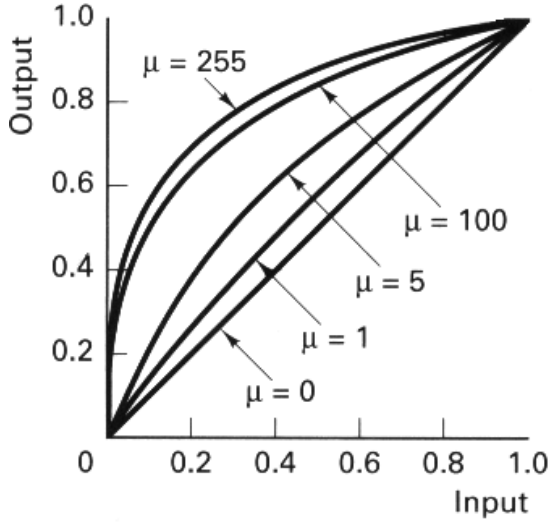
Instead of assigning non-uniform quantization levels, it is practical to pass the signal through a nonlinear device/amplifier which amplifies lower voltage levels more than higher voltage levels so that a uniform quantization afterwards shall have the same effect of non-uniform quantization





companding standard used in North America (μ-Law)

$$y = y_{\max}\frac{\ln(1+\mu(|x|/x_{\max}))}{\ln(1+\mu)}\operatorname{sgn}(x)$$

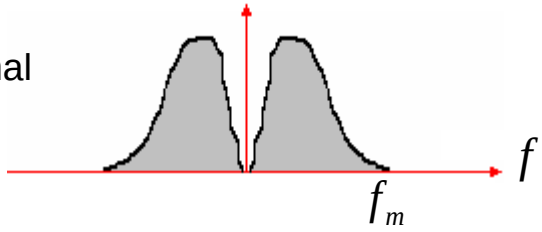


companding standard used in Europe (A-Law)

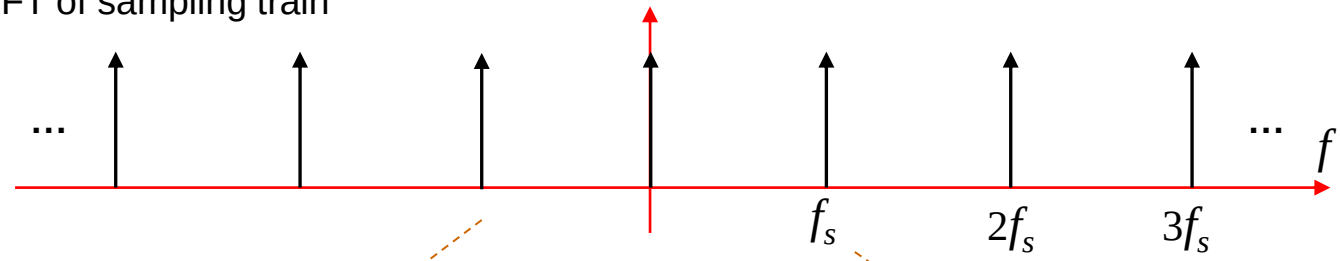
$$y = \begin{cases} y_{\max}\dfrac{A(|x|/x_{\max})}{1+\ln A}\operatorname{sgn}(x) & , 0 < \dfrac{|x|}{x_{\max}} < \dfrac{1}{A} \\ y_{\max}\dfrac{1+\ln(A(|x|/x_{\max}))}{1+\ln A}\operatorname{sgn}(x) & , \dfrac{1}{A} < \dfrac{|x|}{x_{\max}} < 1 \end{cases}$$

Remember the appropriately sampled baseband signal?

Spectrum of the baseband signal

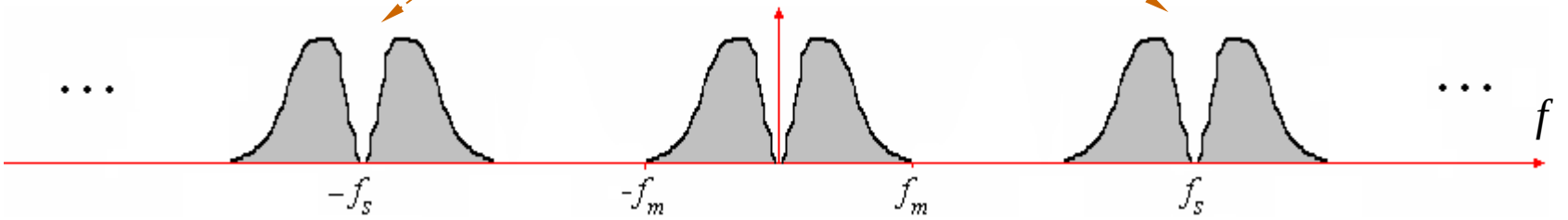


FT of sampling train



Spectrum of the sampled signal

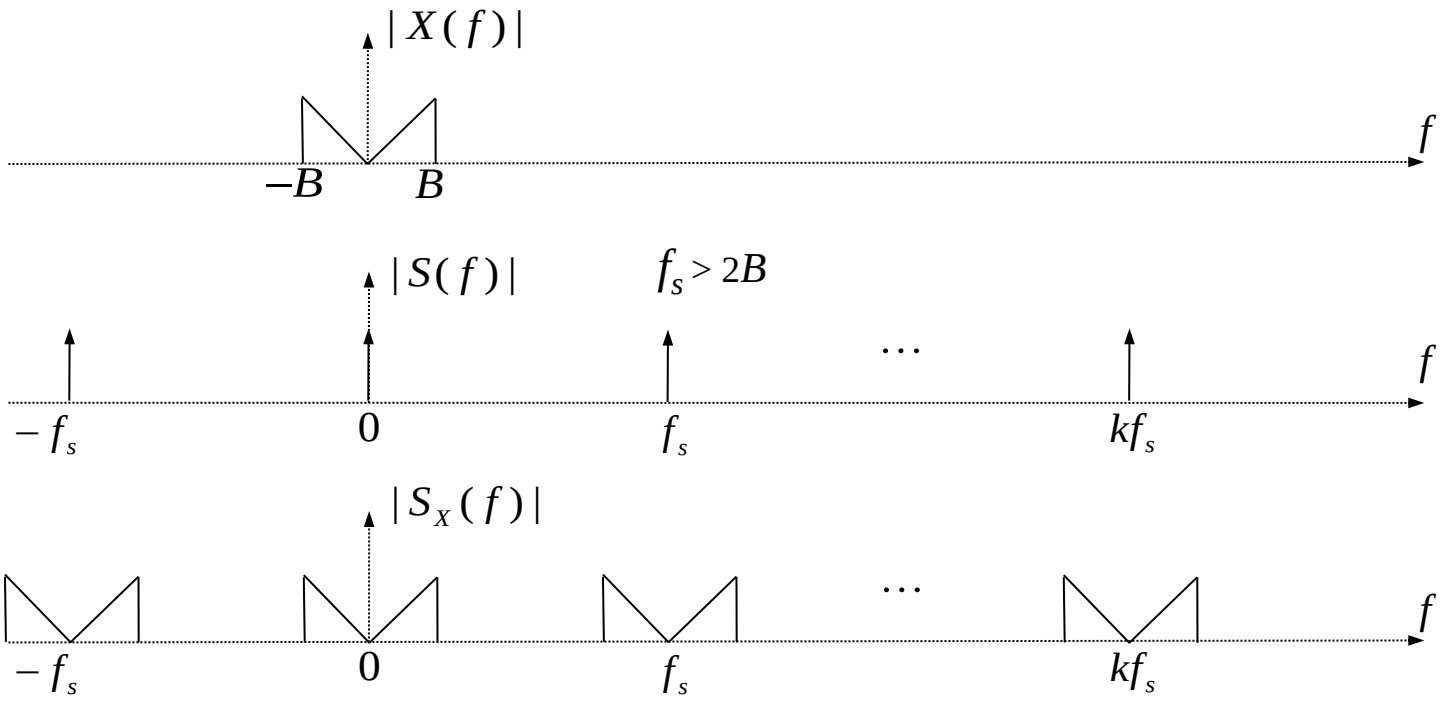
$f_s > 2\,f_m$



we would use a LPF in order to extract this and obtain original baseband signal

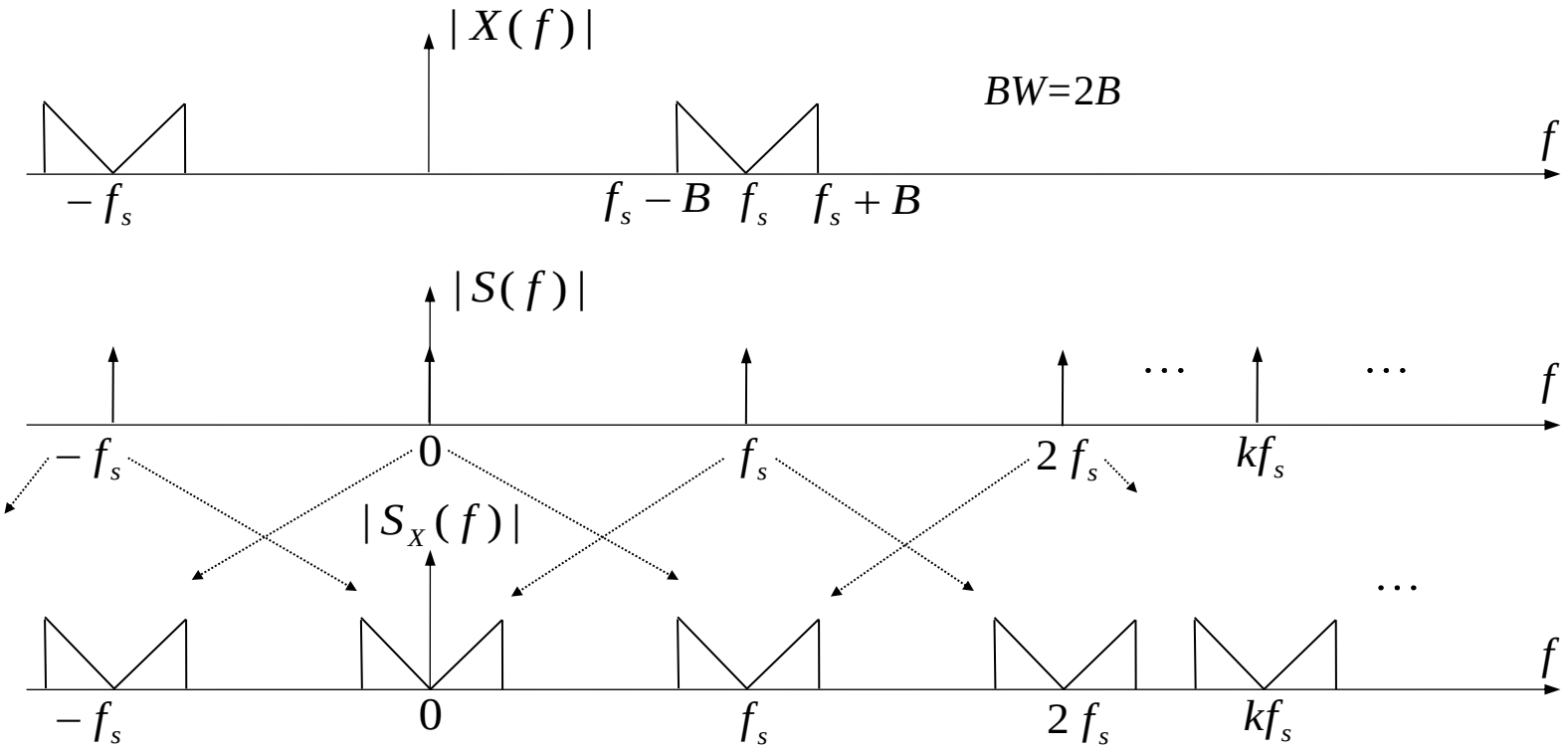
what is we use a BPF and extract this part?

Frequency Up Conversion



One can conveniently extract one of the HF bands with an suitable BPF

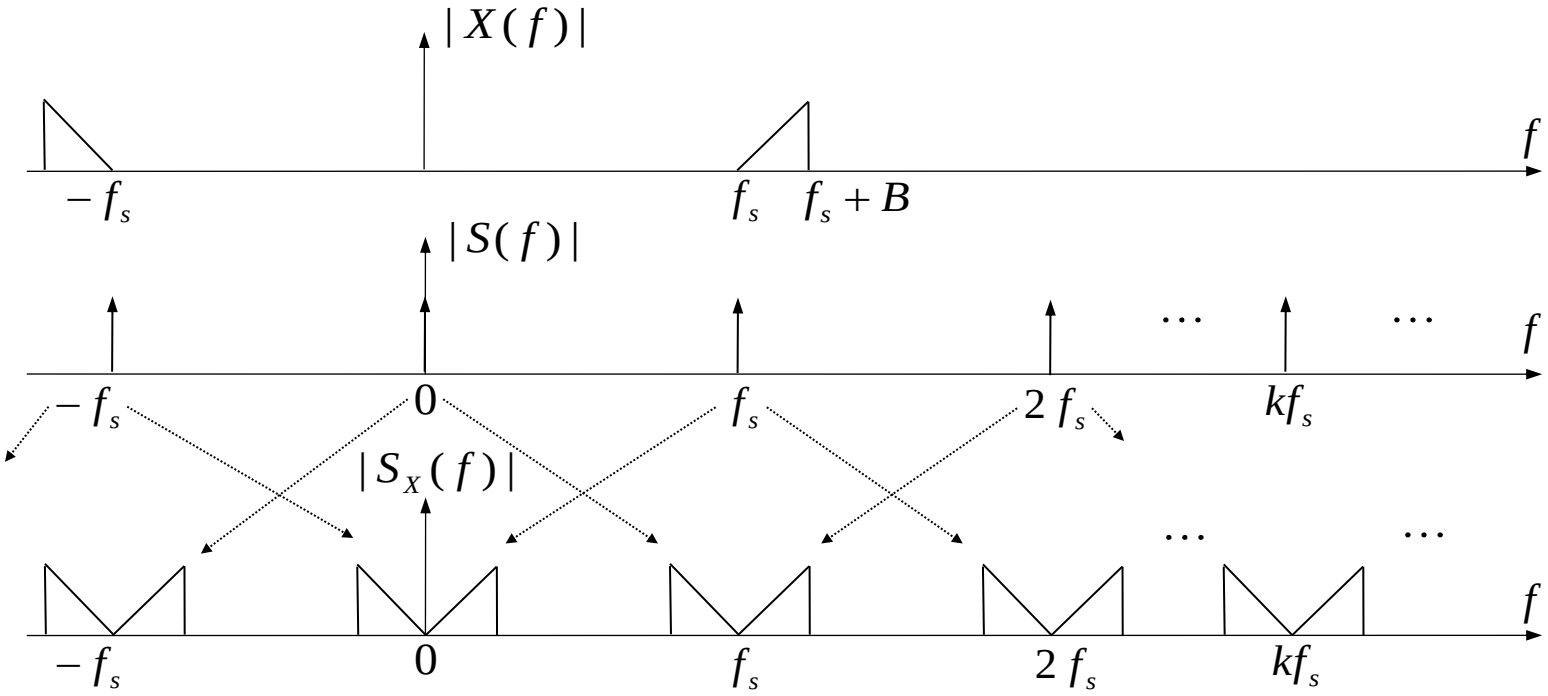
Frequency Down Conversion



Refinement on Nyquist's sampling criterion : A signal with the bandwidth of BW Hz can be reconstructed from its samples if it is sampled at a rate greater than 2BW s/s

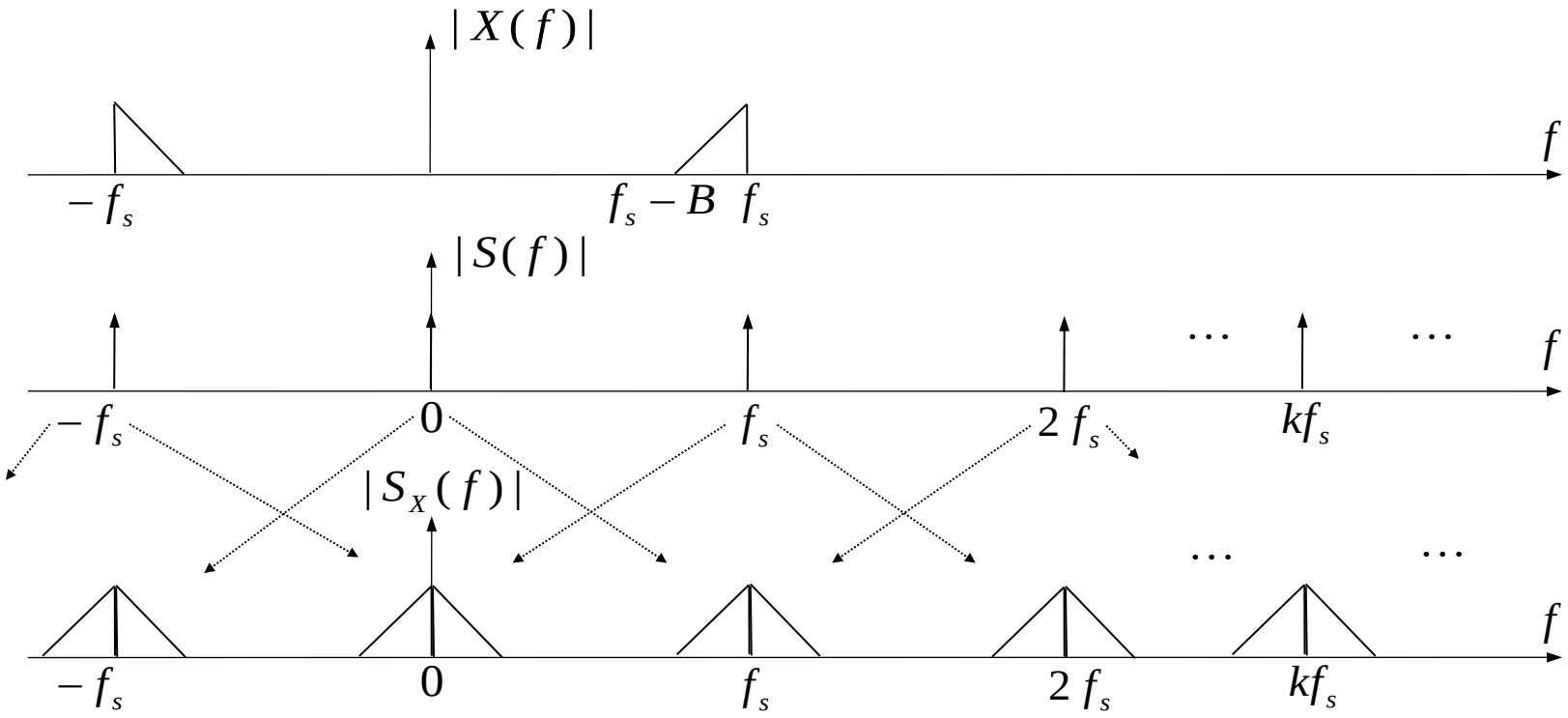
$$f_s > 2BW$$

Down Conversion of a Single Side Band



corollary: A SSB signal can be both demodulated and digitized at the same time

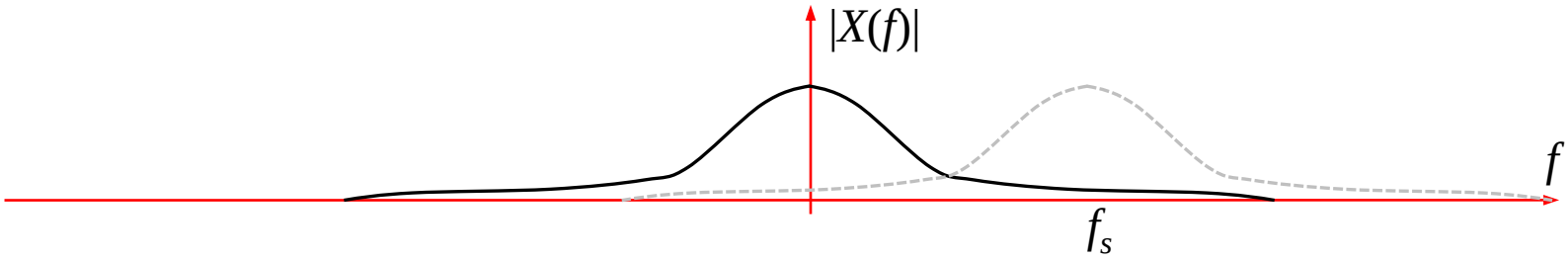
Spectral Inversion



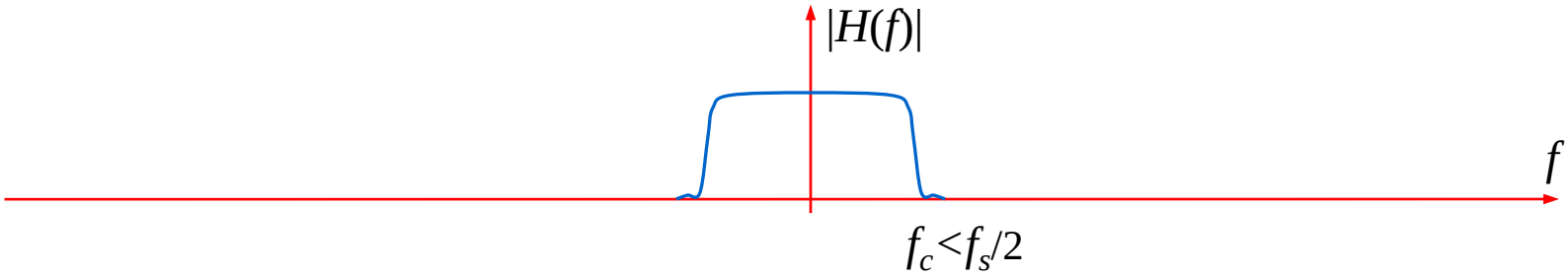
corollary: We need to be careful about whether it is USB or LSB

If the BW of the Signal is Wider than the Sampler can Handle

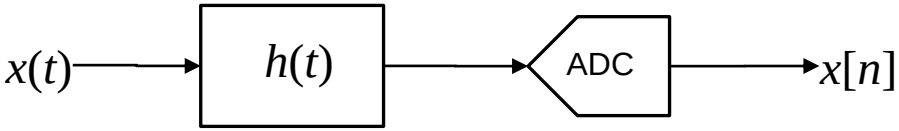
(which is the case, most of the time)



One wishes to limit the bandwidth in order to prevent aliasing

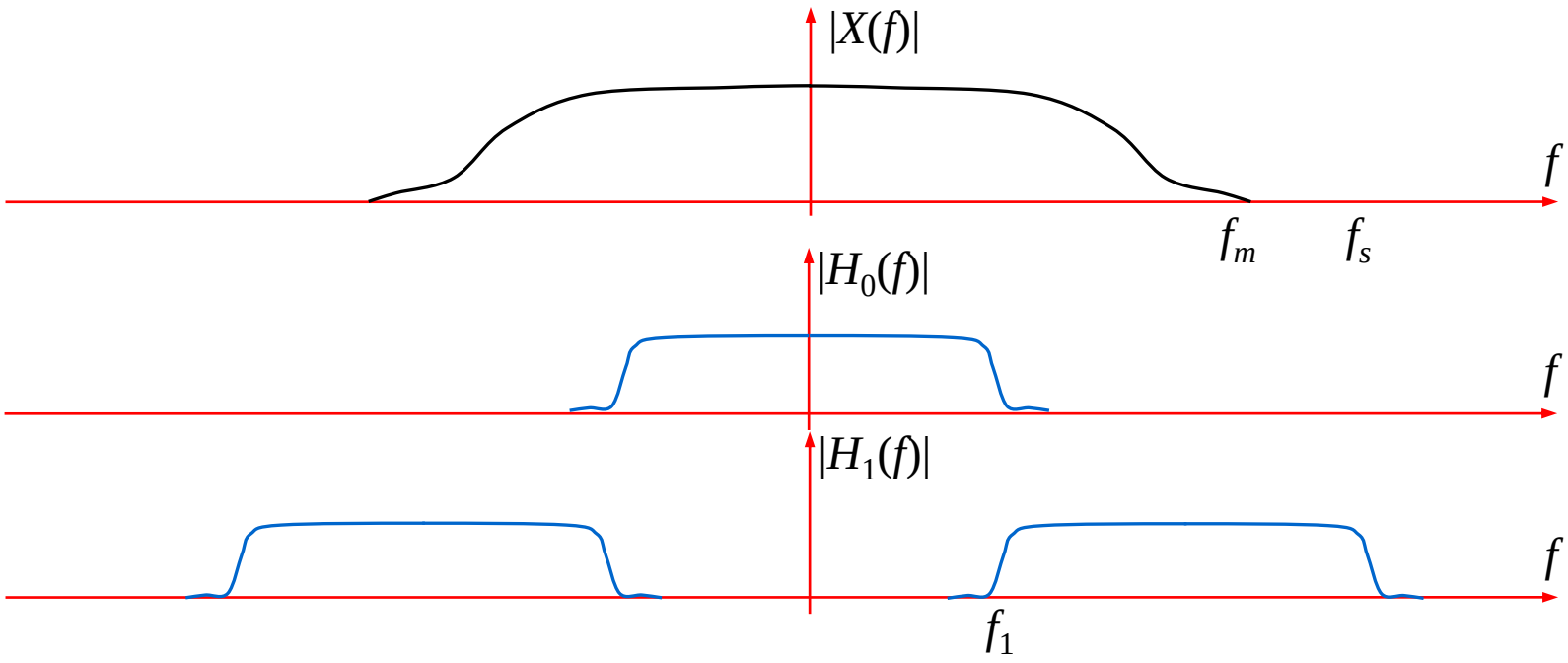


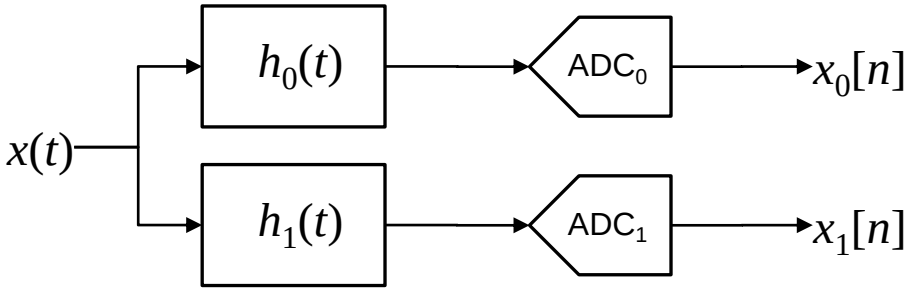
Antialiasing Filter



Not something we actually want, but something we are forced to have, in order to protect the sampled portion

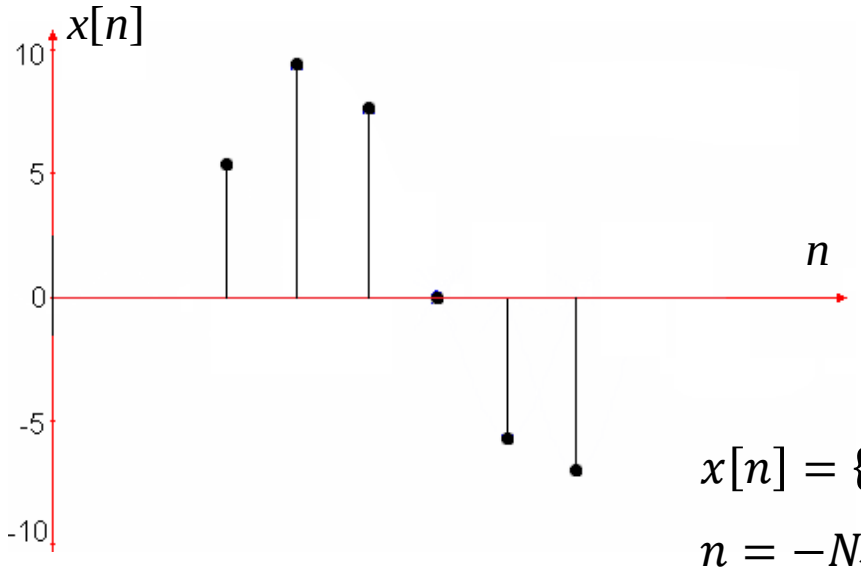
Another Problem Solver





Hmw: Let f_m=100 kHz, and maximum sample rate of ADCs are 100 ksps. Determine filters.

We have the samples, so what?



$$x[n] = \{\cdots, 5,9,7,0,-6,-8,\cdots\}$$

$$n = -N_{min} \cdots -1,0,1,2,\cdots, N_{max}$$

or $$n = 0,1,2,\cdots, N-1$$

Now, we can do;

1. Digital signal procesing on the data, including
    a. data compression (source coding)
    b. error correction (channel coding)
2. Modulations / signal shaping in digital domain

We are not going to delve into digital signal processing in this course, but only summarize how some things are done in digital

Discrete Fourier Transform (DFT)

DFT is the discrete counterpart of FT.
Although they are distinct transforms, their properties are verrry similar.

DFT is defined as

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi kn/N}$$

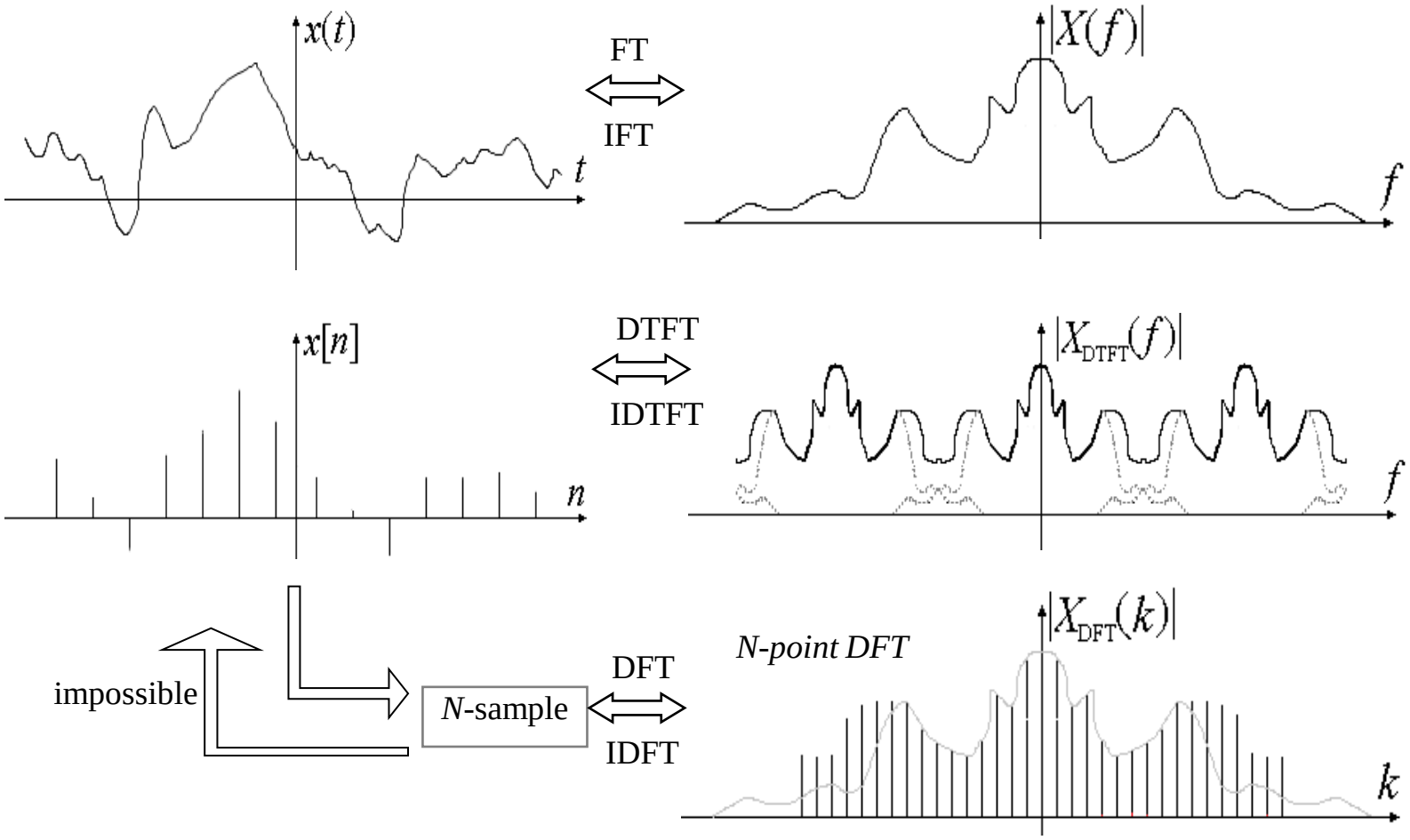
compare them with the continuous FT

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt$$

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df$$

where $x[\cdot]$, $X[\cdot]$, $x(\cdot)$ and $X(\cdot)$ are complex valued

Relations

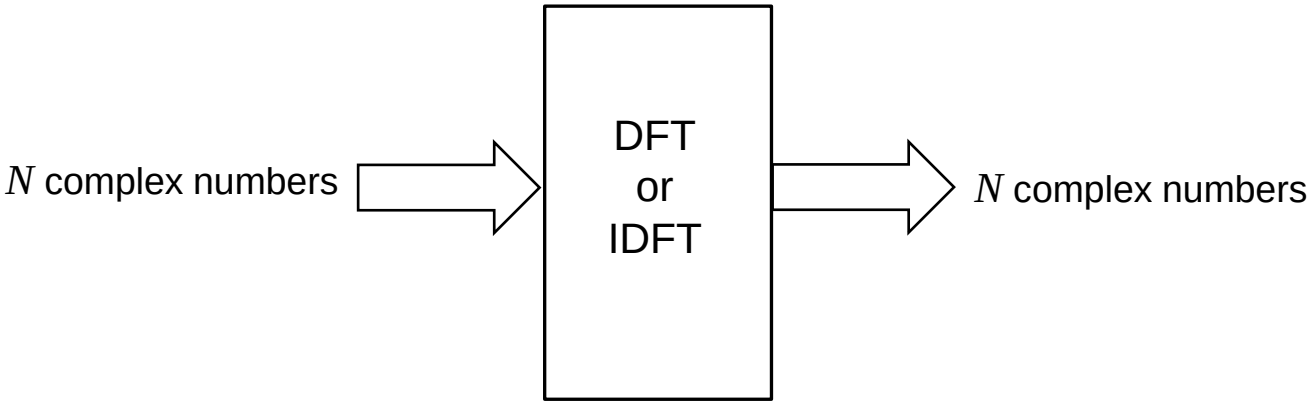


N samples can not represent entire waveform

DFT assumes that the sampled signal is periodic with period $T = NT_s = N/f_s$

In that case DFT gives the Fourier Series coefficients (assumption: $x(t)$ is bandlimited)

Calculation of DFT



Think of DFT as one of several orthogonal linear transforms.
That is, the input data set is represented by the same number of orthogonal functions and the output data set is the coefficients (weights) of these functions whose weighted sum forms the original continuous signal (whose samples are the input data)

DFT and IDFT can be calculated using their definitions.
However, this is almost never the case in practice.

For all real-time signal processing, DFT/IDFT is calculated using fast calculation methods in which symmetry properties of FT are used

Fast Fourier Transform (FFT)

FFT is not a different transform, it is identical to DFT.
The only difference is that, FFT uses symmetry properties of the transform kernel ($e^{-j2\pi kn/N}$) in order to reduce the number of arithmetic operations (additions, multiplications).

$$X_k = \sum_{n=0}^{N-1} x_n W_N^{nk}$$

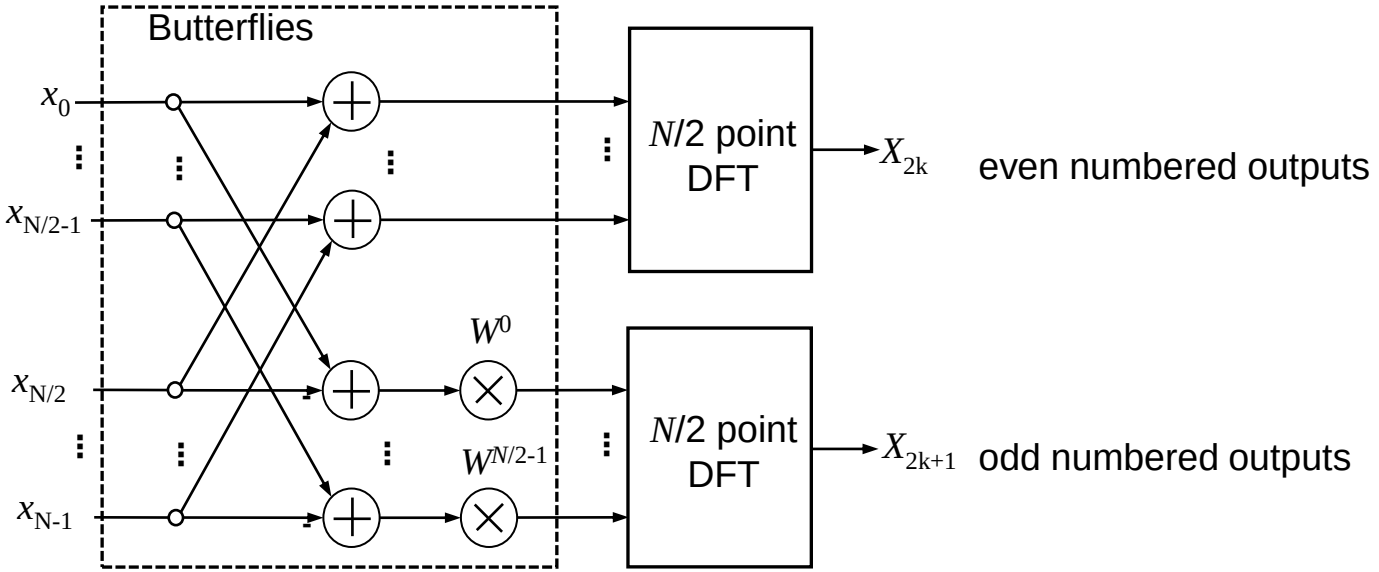
where

$$W_N^{nk} = e^{-j2\pi kn/N}$$

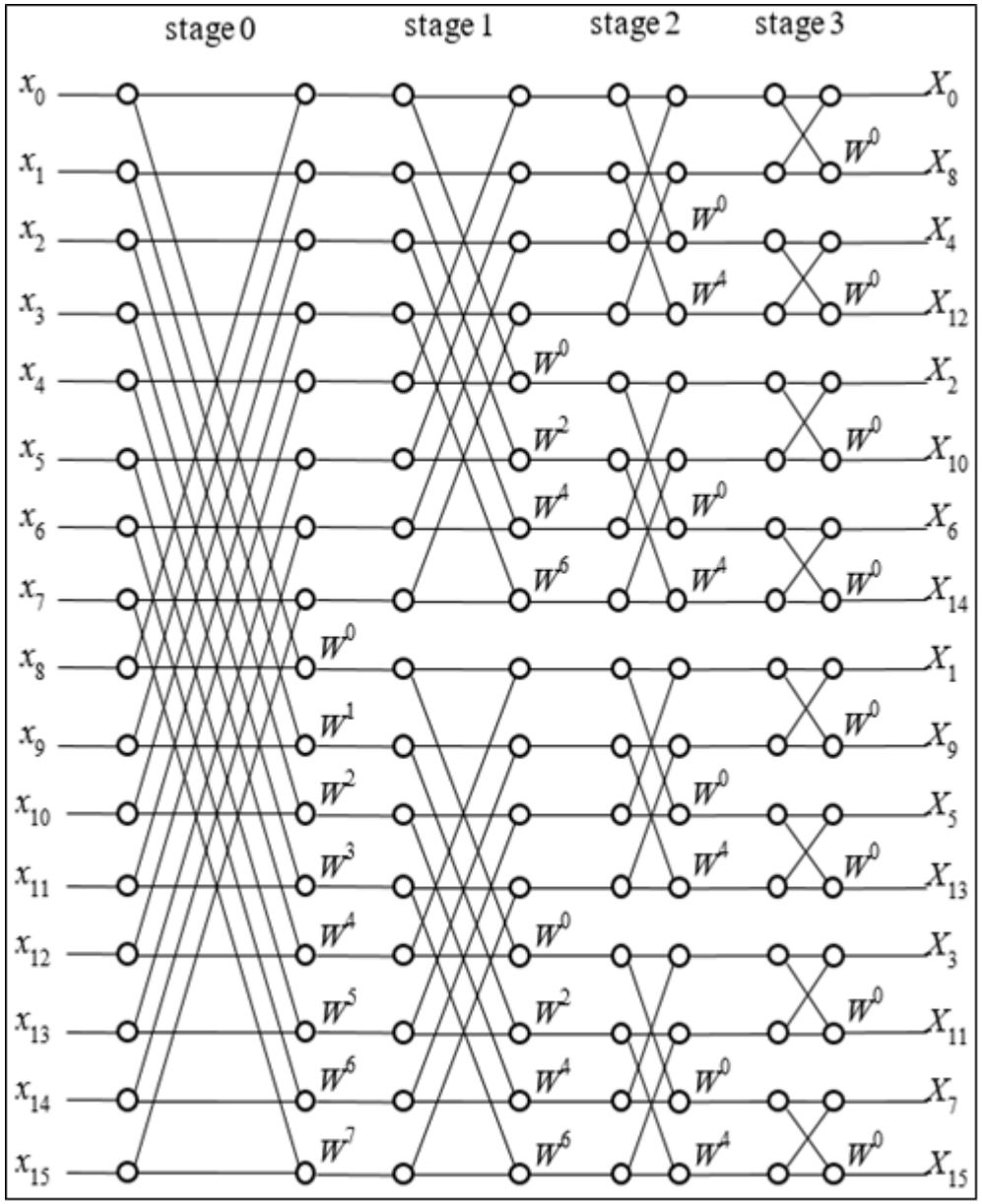
odd and even numbered outputs are calculated separately

$$X_{2k} = \sum_{n=0}^{\frac{N}{2}-1} \left[x_n + x_{n+N/2}\right] W_{N/2}^{nk}$$

$$X_{2k+1} = \sum_{n=0}^{\frac{N}{2}-1} W_N^n \left[x_n + x_{n+N/2}\right] W_{N/2}^{nk}$$



N=16 point, Radix-2, Decimation in Frequency FFT



FFT reduces the complexity of DFT from

$$N^2$$

to

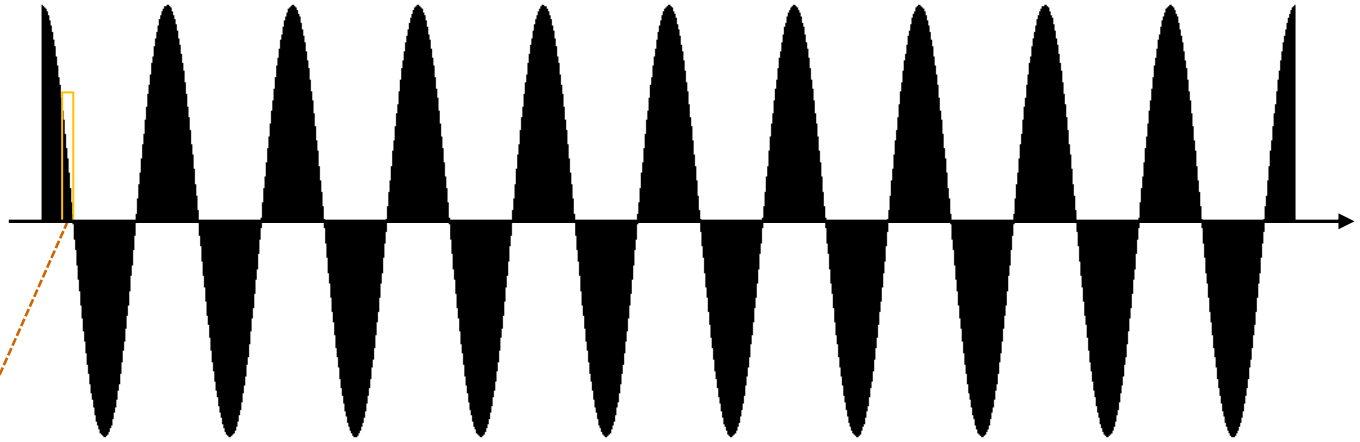
$$N/2\ log_2(N)$$

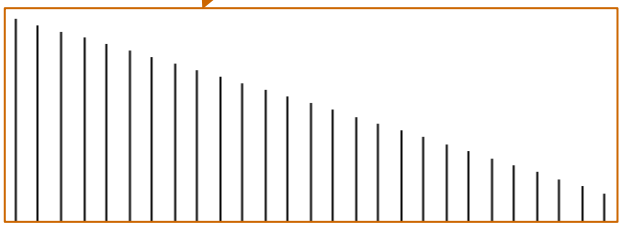
(5120 instead of 1048576 for N=1024. Good savings)

There is much more in sampling / DFT than we summarized here.

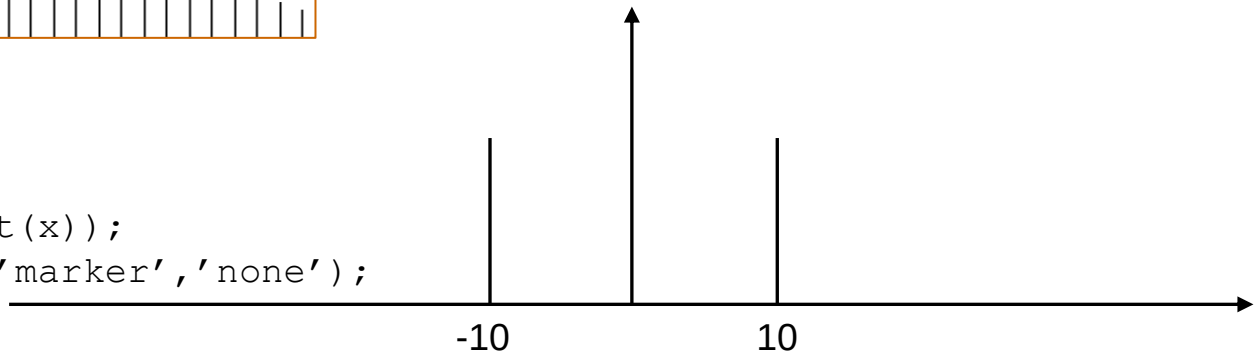
DFT of a Sampled Sinusoidal

```
t=0:3599;
x=cosd(t);
stem(x,'marker','none');
```



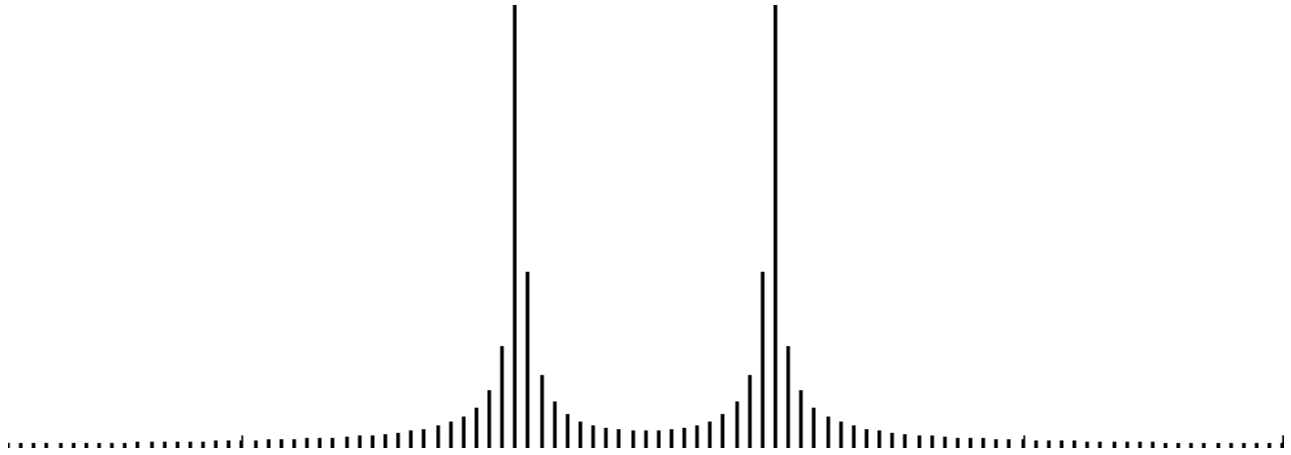


```
X=fftshift(fft(x));
stem(real(X),'marker','none');
```



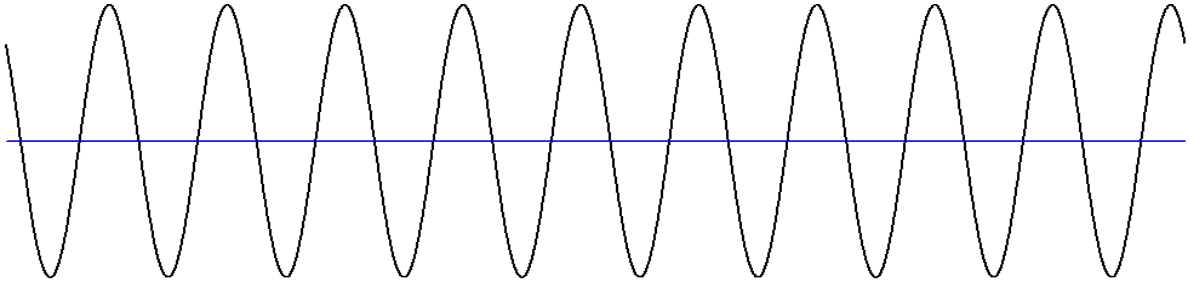
What if we do not have integer number of sinusoidal periods in data?

```
X=fftshift(fft(x(1:3500)));
stem(abs(X),'marker','none');
```

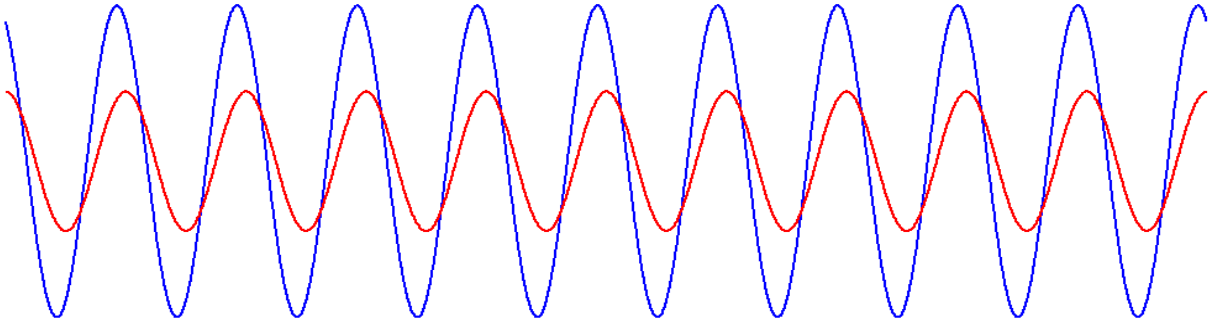


It works in reverse too ☺

```
s=zeros(1,3600); s(11)=1+1i; s(3591)=1-1i; y=ifft(s);
plot(real(y),'black'); hold on; plot(imag(y),'blue');
```



```
s=zeros(1,3600); s(11)=1+1i; s(3591)=1; y=ifft(s);
plot(real(y),'blue'); hold on; plot(imag(y),'red');
```



So, we can create samples of a sinusoidal from complex numbers representing it

Summary

1. What happens when we sample a continuous signal
2. Requirements for reconstructing the sampled signal
3. Frequency UpConversion / DownConversion while sampling / reconstruction
4. Relations between FS, FT, DFT, FFT
5. Q: What do we do with DFT in communication?

END