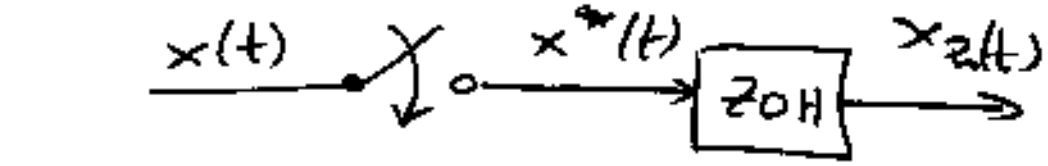
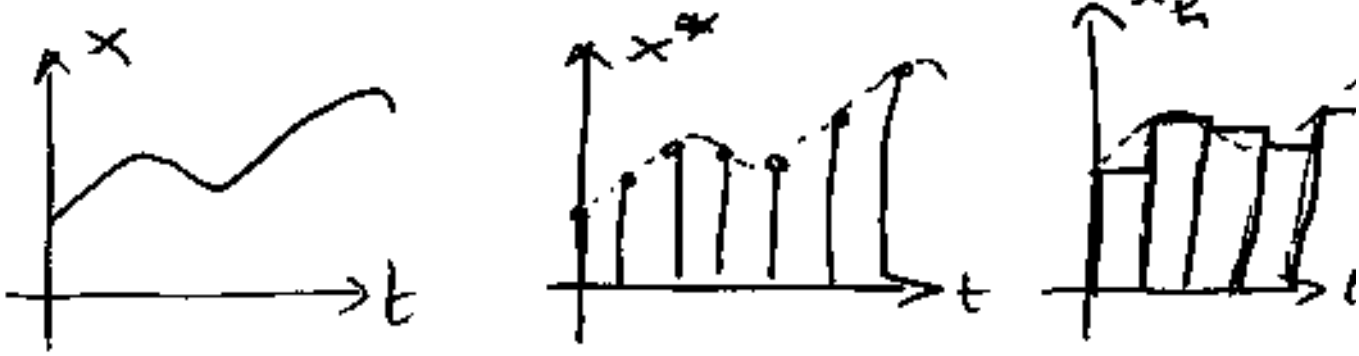
Discrete Systems:

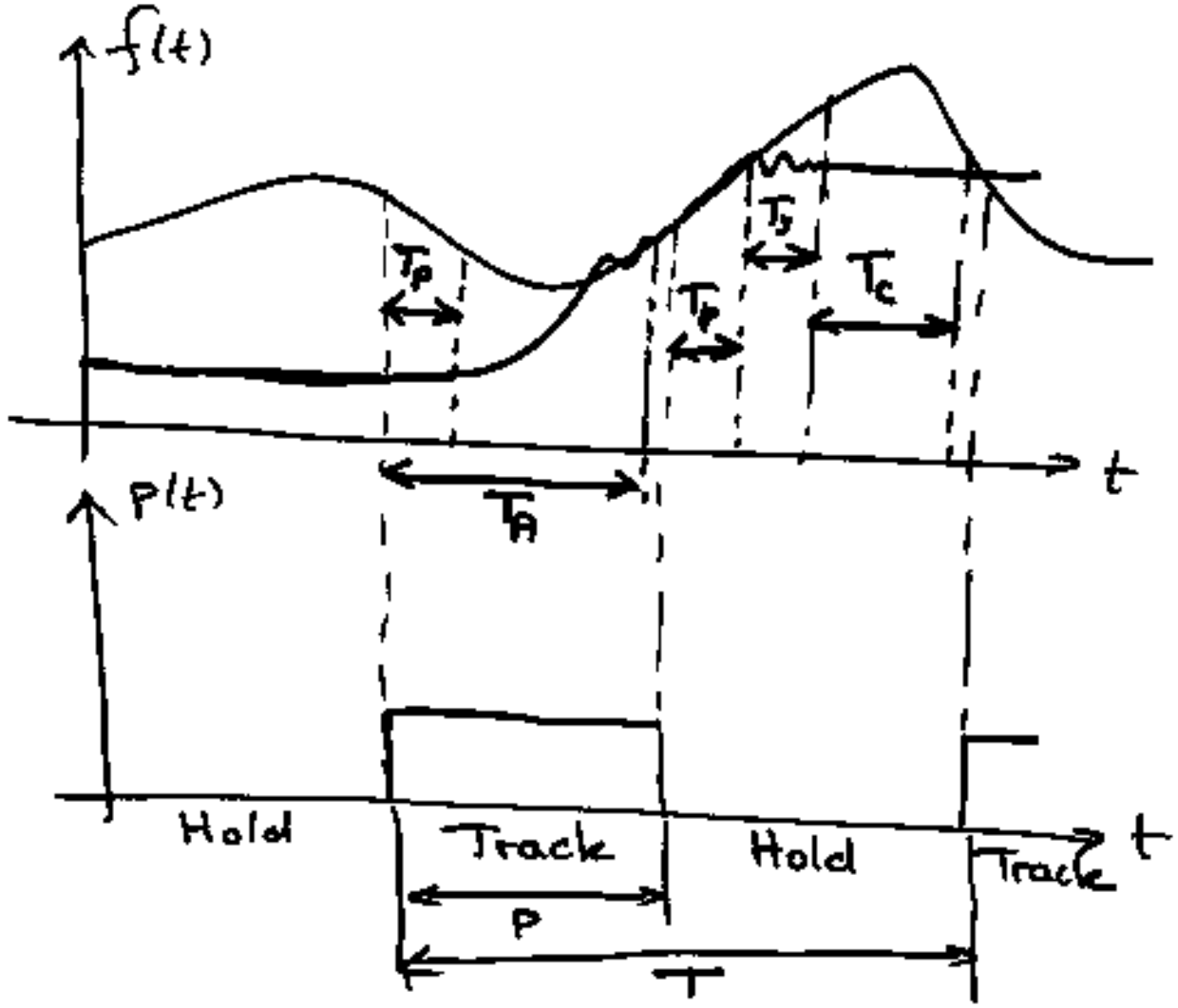
Signals are often digitalized, because digital signals are cheaper to transmit and process than analog signals.





The separation between the (ideal) sampler and the zero-order hold is, however, only a mathematical construct. In reality, these are inseparable.

Sample & Hold Circuit :



P := pulse time

T := sampling time (period)

T_p := Aperture time

T_A := Acquisition time

T_s := Settling time

T_c := Conversion time

When the trigger signal goes high, tracking should begin. However, it takes some time, the aperture time T_p, before the circuit even realizes that it should track. It takes quite a bit longer until the new value of the signal $f(t)$ is acquired. Obviously, the circuit won't work correctly, unless

$$p \geq T_A .$$

Once the trigger goes low, the circuit should hold the current value of $f(t)$. Again, some time (T_p) passes, before the circuit starts to react. Then follows a transient phase, called the settling time, until the signal has stabilized

to its new value. Only then, the A/D conversion can begin, which also consumes some time (see later). Obviously, this won't work, unless:

$$\overline{T} \geq T_A + T_P + T_S + T_C$$

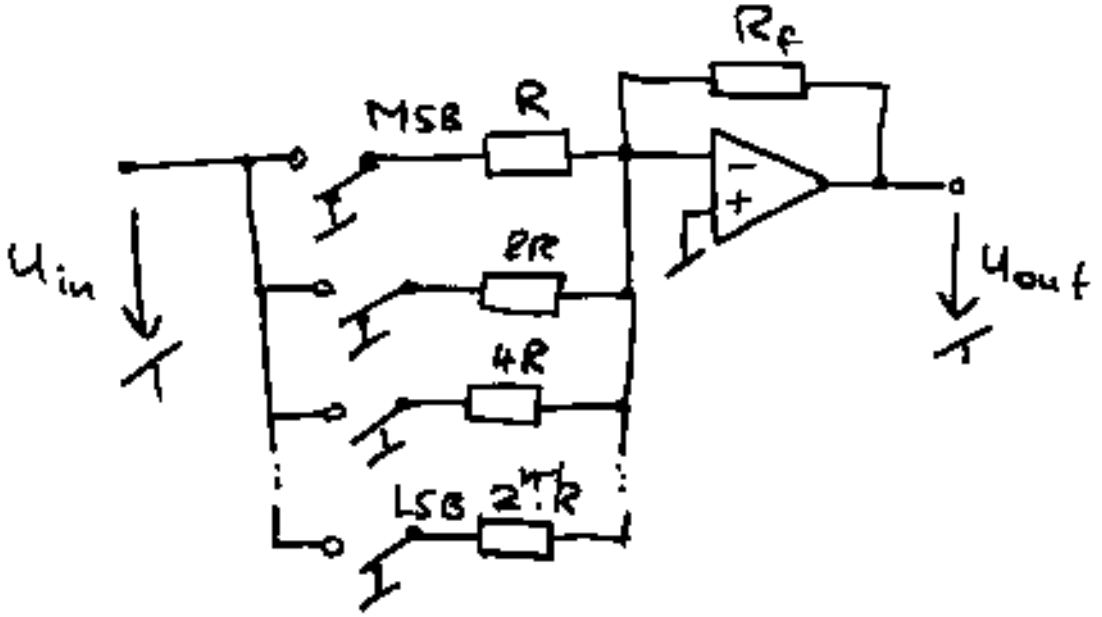
In reality, the value of $\overline{T}$ may need to be larger, because some computation needs to be done with the digitized signal, and obviously:

$$\overline{T} \geq T_{comp}$$

where T_{comp} is the time needed for one cycle of computation.

Digital / Analog Converters:

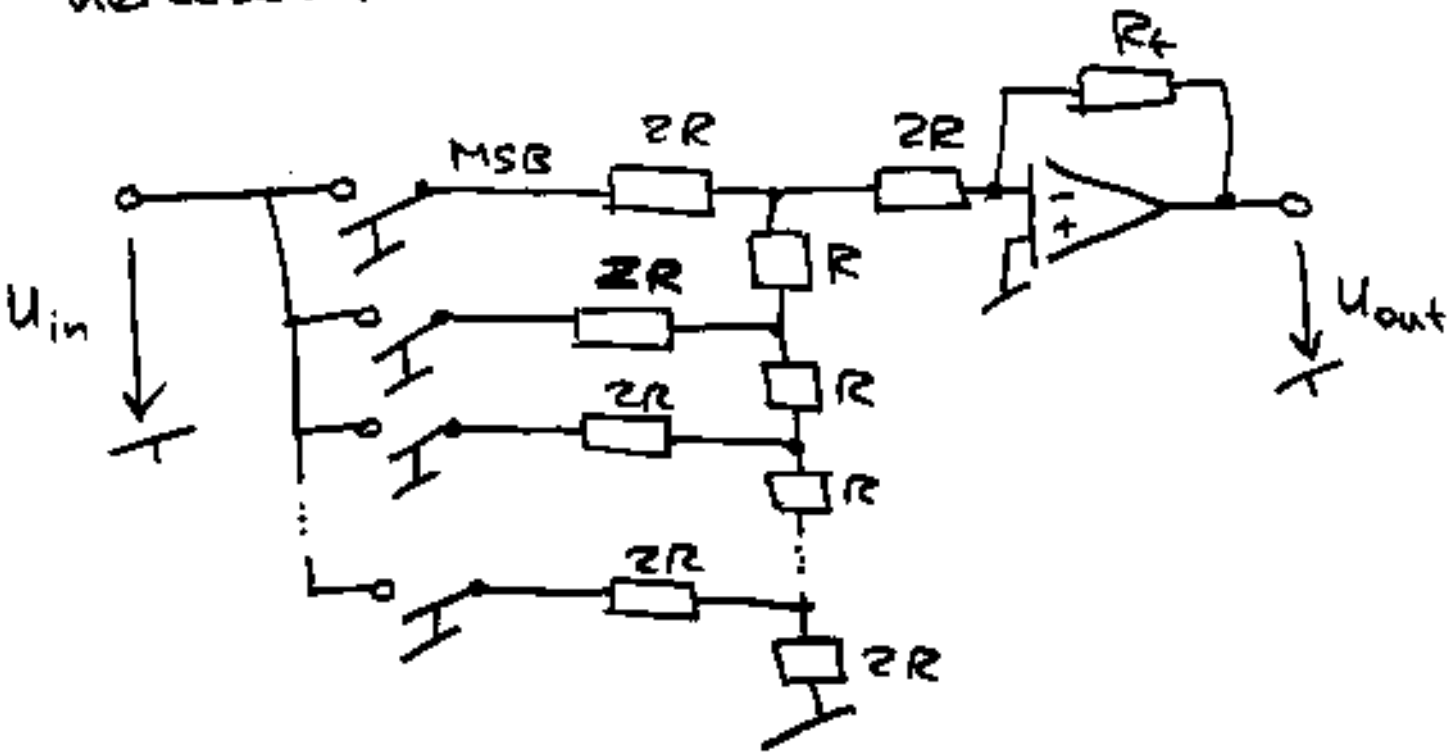
A straightforward A/D Converter circuit can be developed using the opamps introduced in ECE 220:



to obtain an A/D converter with n bits. The most significant bit (MSB) is at the top, the least significant bit (LSB) is at the bottom. Open switches represent Ø and closed switches represent 1.

However, it is technologically difficult and expensive to make resistors on a single chip that are vastly different.

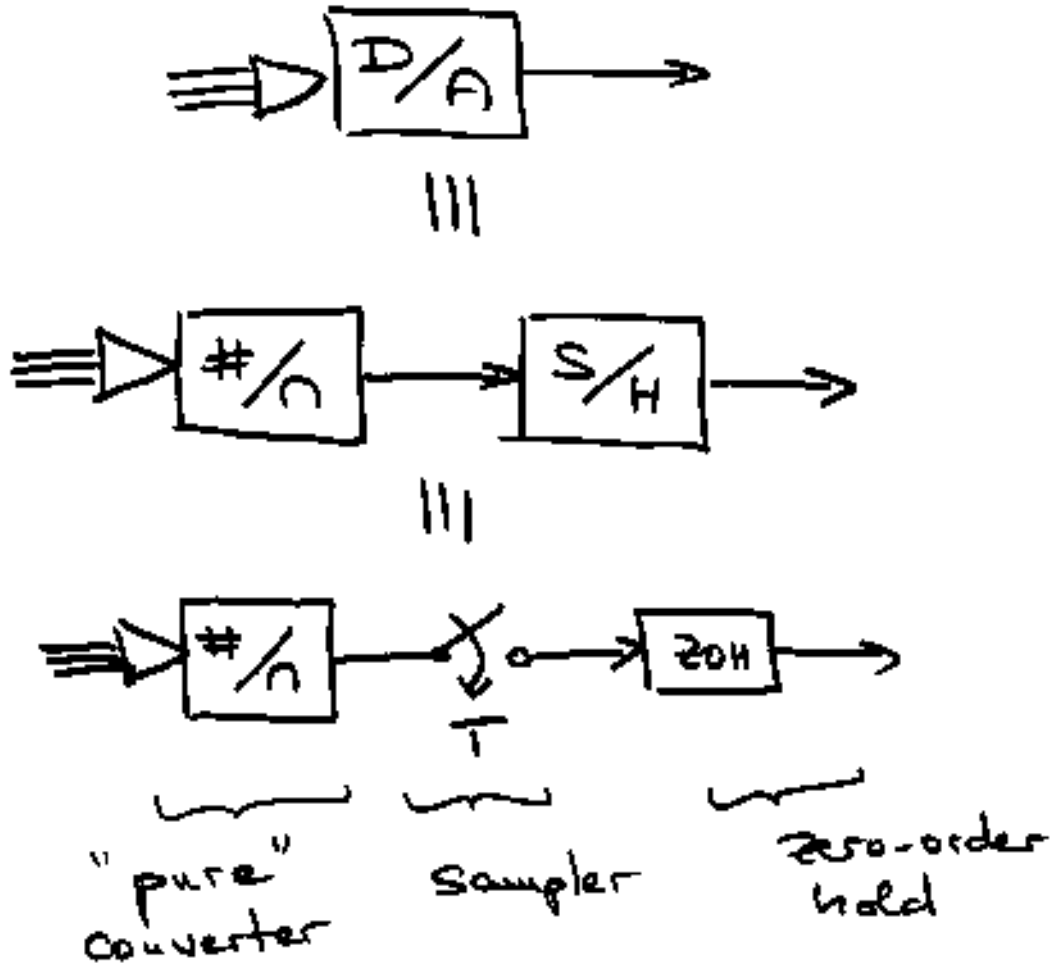
Therefore, a better solution is to use the following "ladder network."



This is easy to manufacture, because only two types of resistors (R and 2R) are needed. Why this works, is not as easy to see.

D/A converters are cheap and fast. Adding a few more bits just calls for some more resistors, i.e., the effort grows linearly, and the speed is not affected at all. The "switches" are usually built as MOS transistors.

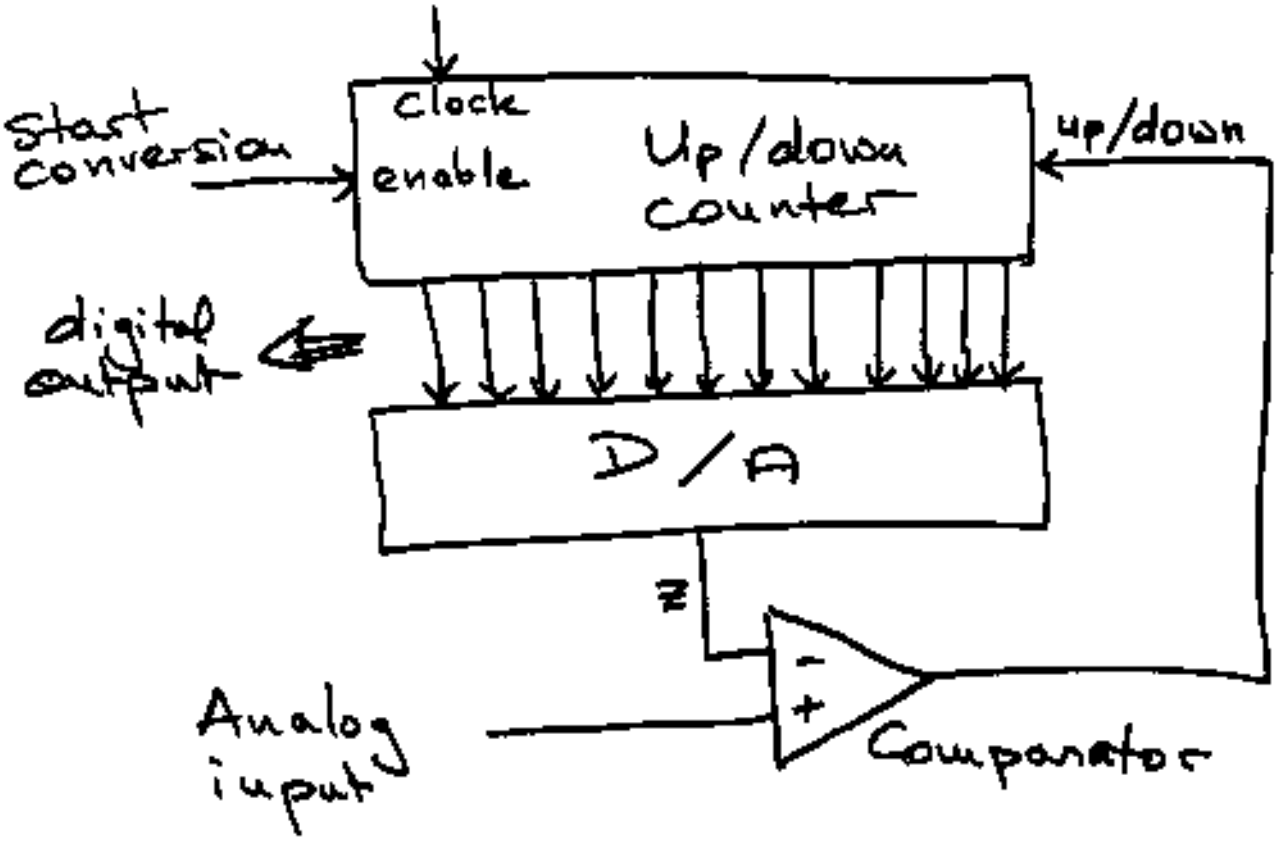
Digital signals are always clocked, i.e., the switch positions change only at fixed time instants, which are multiples of the sampling rate T. Thus, it makes sense to decompose the signal mathematically as follows:



A S/H - circuit may be physically added to the A/D- circuit as well, in order to avoid spikes.

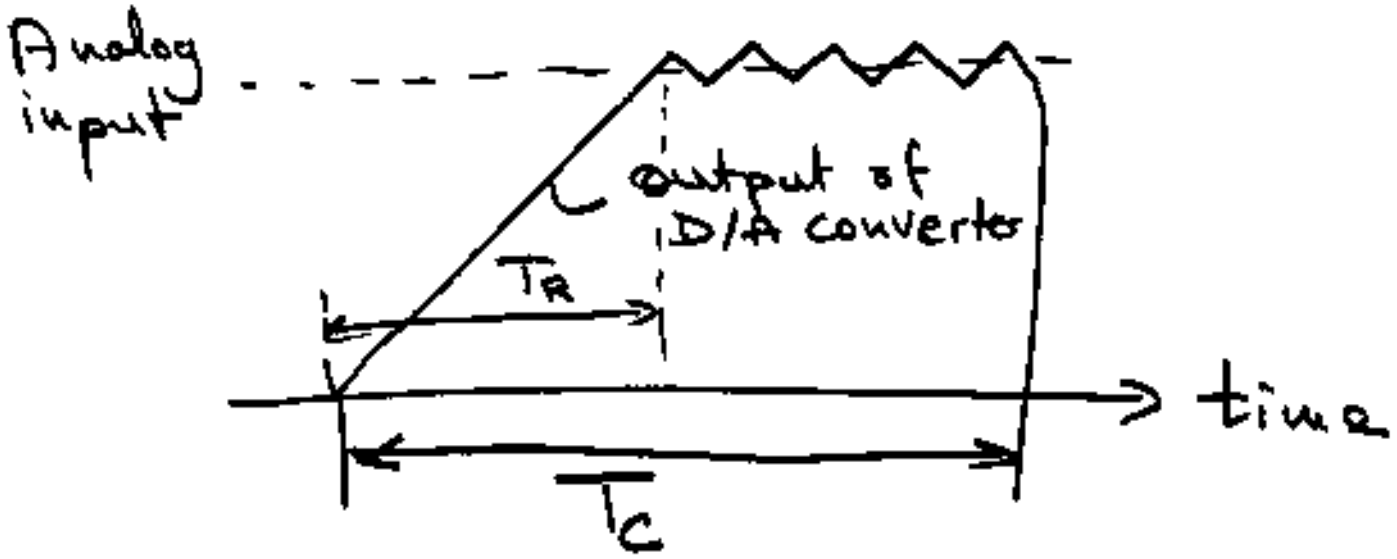
Analog / Digital Converters:

An A/D - converter can e.g. be built as follows:



At the beginning of the conversion, the up/down counter is Enabled. Once enabled, it will count up as long as the u/d signal is high, and it will count down otherwise.

As long as the output of the D/A converter is lower than the analog input, the output of the open-gain opamp is high, and the counter will count up, making the D/A output to grow linearly. As soon as the output of the D/A converter reaches the level of the analog input, the u/d signal goes low.



Obviously this won't work unless the conversion time, T_C, is larger than the ramping time, T_R:

$$T_C \geq T_R .$$

T_R grows exponentially with the number of bits $\Rightarrow$ A/D converters are either <u>slow</u> or <u>expensive</u>.

Commercial A/D converters don't use up/down counters, but some other form of clocked (sequential) logic, but the principle is the same.

A better algorithm works as follows:

(a) Set the MSB to 1, all other bits to $\emptyset$. This gives $1/2$ of full scale (FS) at the output of the D/A converter, z.

(b) If $z > u_{in}$, set MSB back to $\emptyset$, and try next bit $\Rightarrow 1/4$ of FS, etc.

(c) If $z < u_{in}$, leave MSB on, and set the next bit also $\Rightarrow 3/4$ of FS, etc.

(d) Continue until LSB is reached.

With this algorithm, the time needed (# of clocks) grows linearly rather than exponentially in the # of bits.

During the conversion time, the analog input should preferably stay constant, i.e., it makes sense to add a S/H circuit at the input (physically):