Detailed SYLLABUS of the course

LABORATORY OF ADVANCED ELECTRONICS

Department of Physics, University of Trento, a.y. 2024–2025

Lecturers: **LEONARDO RICCI**, **ALESSIO PERINELLI**

Important remark: changes on the present document are possible without any preliminary announcement.
(last updated on November 26, 2024)

LECTURES

(23/09/2024) 1. **Introduction to the course. Signals and systems. Sequences.**

- Signal processing:
  - analog signal processing;
  - digital signal processing (DSP).
- Noise (a very short review):
  - Johnson noise;
  - shot noise.

- Signals:
  - a formal definition of signals (pictures included!);
  - continuous signals;
  - sequences (discrete–time signals).
- Sequences:
  - graphical representation;
  - remarkable sequences: impulse–sequence $\delta[n]$, step–sequence $u[n]$;
  - delayed sequences, relation between $\delta[n]$ and $u[n]$;
  - representation of a generic sequence by means of delayed impulse–sequences;
  - periodic sequences;
  - energy of a sequence.

- Systems:
  - a formal definition of systems.

(27/09/2024) 2. **LTI systems. Basic properties of LTI systems. Difference equations.**

- Linear, time–invariant (LTI) systems:
  - action on a sequence;
  - impulse–response (a.k.a. transfer function) $h[n]$, and convolution;
  - a comment on the symmetry between input $x[n]$ and impulse–response

$h[n]$: their roles can be swapped!

- Properties of an LTI system:
  - reality;
  - causality;
  - (noncausality);
  - (marginal stability;)
  - *bound–input, bound–output* ("BIBO") stability,
    - definition,
    - necessary and sufficient condition for BIBO stability;
  - examples ($h[n] = \delta[n]$, $h[n] = u[n]$, $h[n] = a^n u[n]$).

- Difference equations:
  - similarity with the continuous case: a difference equation characterizes an LTI system and it is typically a recursive equation;
  - example,
    - $h[n]$ in the case of $y[n] = x[n] + x[n-1]$ (causal and noncausal solution),
    - Fibonacci sequence (hint at a solution via matrices; left as a homework).

(30/09/2024) 3. **z–transform and its inversion.**

- z–transform:
  - a discussion on the importance of changing space, in analogy with continuous systems, to solve difference equations;
  - definition and region of convergence ("ROC");
  - remarkable examples,
    - $\delta[n]$,
    - $u[n]$,
    - $-u[-n-1]$.
- Basic properties of z–transform:
  - linearity (important: beware of the intersecting ROCs!);
  - time–shift (important: beware of new ROC!);
  - convolution theorem (important: beware of the intersecting ROCs!).
- Complementary topics: graphical representation of systems:
  - linear combinations of systems;
  - cascade systems, and invertibility of two systems (proof via z–transform).

- Summary of residue calculus:
  - $\frac{1}{2\pi i}\oint_{\Gamma\ um\ z_o}(z - z_o)^n dz = \delta_{n,\ -1}$;
  - $G(z)$ has an n^{th}–order pole in $z_o \Longrightarrow$
    $\frac{1}{2\pi i}\oint_{\Gamma\ um\ z_o} G(z)dz = \frac{1}{(n-1)!}\frac{d^{n-1}}{dz^{n-1}}[G(z)(z - z_o)^n]_{z=z_o}$.
- Inversion of the z–transform:
  - derivation of the inversion expression;
  - examples,
    * $X(z) = 1$ with ROC $= \mathbb{C}$,
    * $X(z) = \frac{z}{z-1}$ with ROC $= \{z \,|\, |z| < 1\}$,
    * $X(z) = \frac{z}{z-1}$ with ROC $= \{z \,|\, |z| > 1\}$.

(07/10/2024) 4. **Basic properties of LTI systems as seen in the z–domain. Inversion of the z–transform: examples. Introduction to Nyquist–Shannon sampling theorem.**

- Basic properties of systems as seen in the z–domain:
  - reality,
    * reality $\Longleftrightarrow \overline{X}(\overline{z}) = X(z)$,
    * examples ($\delta[n]$, $a^n u[n]$, $-a^n u[-n-1]$),
    * existence of complex-conjugate zeroes and poles;
  - causality,
    * beware: causal $\neq$ $\overline{\text{noncausal}}$ ($k \cdot \delta[n] \ldots$ is both!),
    * causality, namely $x[n] = 0$ for $n < 0 \Longleftrightarrow \infty \subset$ ROC (implication $\Leftarrow$ via Taylor-Laurent series),
    * noncausality, namely $x[n] = 0$ for $n > 0 \Longleftrightarrow 0 \in$ ROC (implication $\Leftarrow$ via Taylor-Laurent series),
    * causuality & noncausality, namely $x[n] = 0$ for $n \neq 0$, i.e. $x[n] = k \cdot \delta[n]$ with $k \in \mathbb{C} \Longleftrightarrow$ $0 \in$ ROC, $\infty \subset$ ROC $\Longleftrightarrow X(z)$ uniform on $\mathbb{C}$ (the statement corresponds to the Liouville theorem: a bounded holomorphic function whose ROC coincides with $\mathbb{C}$ is uniform),
    * examples ($\delta[n]$, $a^n u[n]$, $-a^n u[-n-1]$);
  - BIBO stability,
    * BIBO stability $\Longleftrightarrow \Gamma_1 \subset$ ROC,
    * examples ($\delta[n]$, $a^n u[n]$, $-a^n u[-n-1]$).

- Remarkable examples concerning the inversion of the z–transform:
  - Fibonacci sequence;

- Poisson process and distribution.

- Towards Nyquist–Shannon sampling theorem:
  - sampling,
    - generation of a sequence $x[n]$ starting from a continuous signal $x_a(t)$:
      $x[n] = x_a(nT)$,
    - basic issue: how much information is lost in the sampling process, or, better, *how good a continuous signal can be reconstructed from a sequence*?
  - important definitions,
    - sampling time/period T,
    - sampling frequency/rate $f_s = \frac{1}{T}$,
    - sampling frequency $\omega_s = \frac{2\pi}{T}$,
    - Nyquist frequency $f_{\mathrm{Ny}} = \frac{1}{2T} = \frac{f_s}{2}$, $\omega_{\mathrm{Ny}} = \frac{\pi}{T}$,
    - Nyquist band $\left(-\frac{\pi}{T}, \frac{\pi}{T}\right)$.

(11/10/2024) 5. **Nyquist–Shannon sampling theorem.**

- Nyquist–Shannon precursor relation, reconstruction:
  - starting conditions:
    1a) analog, continuous signal $x_a(t) \in L^2$ and
    1b) related sampled sequence $x[n]$ being BIBO stable;
  - *"Nyquist–Shannon precursor relation"* between the $\mathcal{F}$–transform $\tilde{X}(\omega)$ of $x_a(t)$ and the z–transform $X(z)$ of $x[n]$:
    $T \cdot X\left(e^{-i\omega T}\right) = \sum_{\forall k} \tilde{X}(\omega + \frac{2\pi}{T}k)$
    (where both terms are periodic);

  - useful definitions I. (*ex post*), II., III. (*ex ante*),
    I. *"folding"*, $\tilde{X}_{folding}(\omega) \equiv \sum_{\forall k} \tilde{X}(\omega + \frac{2\pi}{T}k)$,
    II. $\frac{\pi}{T}$–window, $W(\omega) \equiv [\theta(\omega + \frac{\pi}{T}) - \theta(\omega - \frac{\pi}{T})]$,
    III. *"reconstruction"*, $x_{rec}(t) \equiv \sum_{\forall k} x[k] \operatorname{sinc}\left[(t - kT)\frac{\pi}{T}\right]$;
  - reconstruction,
    - $\mathcal{F}^{-1}\left[W(\omega) \cdot \tilde{X}_{folding}(\omega)\right] = x_{rec}(t)$,
    - remarkably, $x_{rec}[n] = x_a(nT) = x[n]$.

- Nyquist–Shannon sampling theorem:
  - final condition (the crucial one!):
    2) the analog, continuous signal $x_a(t)$ is $\frac{\pi}{T}$–BL (band–limited);

- Nyquist–Shannon sampling theorem in the frequency domain), $\tilde{X}(\omega) = W(\omega) \cdot \tilde{X}_{folding}(\omega)$;
- Nyquist–Shannon sampling theorem in the time domain, $x_a(t) = x_{rec}(t)$.

- Aliasing in the case of a non–$\frac{\pi}{T}$–BL signal $x_a(t)$:
  - Aliasing:
    if condition 2) is not met, i.e. $x_a(t)$ is not $\frac{\pi}{T}$–BL, then
    $\tilde{X}(\omega) \neq W(\omega) \cdot \tilde{X}_{folding}(\omega)$, and
    $x_{rec}(t) \neq x_a(t)$, i.e. $x_{rec}(t)$ is not equal to $x_a(t)$ but it is *something else* ($\sim$ latin *alias*).

- Practical statement of Nyquist–Shannon sampling theorem: if f_0 is the maximum frequency occurring in a signal, use a sampling frequency f_s such that $f_s > 2f_0$.
- Remarkable example: aliasing in the case of sinusoidal and cosinusoidal signals.

(14/10/2024) 6. **Digital simulation of an analog system.**

- Theorem concerning the digital simulation of an analog system ("simulation theorem"):
  - general discussion;
  - proof assuming...
    * 1a) Fourier-transformable input signals and transfer function, i.e. $x(t)$, $G(t) \in L^2$,
    * 1b) BIBO-stable simulator $h[n]$,
    * 2) $\frac{\pi}{T}$–BL input signals;
  - expression of the ideal transfer function $h[n]$ of the simulator;
  - reality: $G(t)$ is real $\Rightarrow h[n]$ is real.
- Implementation issues:
  - stability of the ideal transfer function $h[n]$ is to be assessed case by case;
  - non–causality issue for the ideal transfer function $h[n]$ (example: first-order low-pass filter);
  - difficulty – in the general case – of calculating $h[n]$ (example: first-order low-pass filter);

  - a remarkable example: digital differentiator (ideal solution $\propto$ (1 $-$

$\delta[n])\frac{(-1)^n}{n}$).

- "Backward interpretation of the simulation theorem":
  - statement: any digital system characterized by an impulse sequence $v[n]$—and the related z-transform $V(z)$—is a perfect simulator of an analog system whose impulse response's Fourier transform $\tilde{V}(\omega)$ is given by $V(e^{-i\omega T})$.
- Approximated simulation via "backward interpretation of the simulation theorem":
  - an approach to overcome the implementation issues: requiring the simulator's output $y'[n]$ to approximate the sampled analog output $y[n]$, $y'[n] \cong y[n]$, rather than imposing $h[n] = g[n]$.
    So, because it is mostly impossible to find $H(z)$, and thus $h[n]$, such that it exactly simulates a given $\tilde{G}(\omega)$,
    find an implementable $V(z)$, and thus $v[n]$, such that its Fourier–transform equivalent function $V(e^{-i\omega T})$ suitably approximates $\tilde{G}(\omega)$;
  - (approximation of an ideal system $h[n]$, $H(z)$ through a real one $v[n]$, $V(z)$ via, ex. gr., minimization of the Tchebycheff error or the root–mean–square error;)
  - a remarkable example: digital differentiator,
    * solution $\delta[n] - \delta[n-1]$,
    * solution via a non recursive (FIR) filter based on delays up to 2 periods.

(21/10/2024) 7. **Simulation via bilinear transform.**

- Bilinear transform:
  - an issue: how to express ω, or s, in terms of z by relying on $z = e^{-i\omega T}$ and exploiting the approximation approach to simulation provided by the backward interpretation of the simulation theorem;
  - bilinear transform statement $\omega \to \frac{2i}{T}\frac{z-1}{z+1}$, $s \to \frac{2}{T}\frac{z-1}{z+1}$
    $\Longrightarrow V(z) = \tilde{H}\left(\omega = \frac{2i}{T}\frac{z-1}{z+1}\right)$; $V(z) = \tilde{H}_{\text{Laplace}}\left(s = \frac{2}{T}\frac{z-1}{z+1}\right)$;
  - desirable frequency behaviour for $T \ll bandOfInterest^{-1}$.
- Simulation via bilinear transform of an LTI system characterized by a rational transfer function:
  - basic properties,
    * reality $\Big(h(t)$ is real $\Longleftrightarrow \tilde{H}^*(\omega) = \tilde{H}(-\omega)\Big)$,
    * causality (to be imposed),
    * BIBO–stability, $h(t)$ is BIBO–stable $\Longleftrightarrow$ stable system with rational transfer function $\tilde{H}(s)$,

    - $\cdot\ s \to \frac{2}{T}\frac{z-1}{z+1} \quad \Longleftrightarrow \quad z \to \frac{sT/2+1}{sT/2-1}$,
    - $\cdot \implies \quad \mathrm{Re}\, s < 0 \leftrightarrow |z| < 1$;
  - in the case of systems characterized by rational transfer functions $\tilde{H}(\omega) = N(\omega)/D(\omega)$, with degree$[N(\omega)] <$ degree$[D(\omega)]$, superiority of bilinear transform with respect to other transforms (for example the one in which $\tan(\omega T/2)$ is replaced by $\sin(\omega T)$), due to a desirable behaviour in the neighbourhood of the Nyquist frequency.
- Example:
  - low-pass filter simulator via bilinear transform;
  - implementation via a difference equation;
  - frequency response.

(21/10/2024) 7. **Simulation via bilinear transform.**

- Bilinear transform:
  - an issue: how to express ω, or s, in terms of z by relying on $z = e^{-i\omega T}$ and exploiting the approximation approach to simulation provided by the backward interpretation of the simulation theorem;
  - bilinear transform statement $\omega \to \frac{2i}{T}\frac{z-1}{z+1}$, $s \to \frac{2}{T}\frac{z-1}{z+1}$ $\implies V(z) = \tilde{H}\left(\omega = \frac{2i}{T}\frac{z-1}{z+1}\right)$; $V(z) = \tilde{H}_{\mathrm{Laplace}}\left(s = \frac{2}{T}\frac{z-1}{z+1}\right)$;
  - desirable frequency behaviour for $T \ll bandOfInterest^{-1}$.
- Simulation via bilinear transform of an LTI system characterized by a rational transfer function:
  - basic properties,
    - $*$ reality $\Big(h(t)$ is real $\Longleftrightarrow \tilde{H}^*(\omega) = \tilde{H}(-\omega)\Big)$,
    - $*$ causality (to be imposed),
    - $*$ BIBO–stability, $h(t)$ is BIBO–stable $\Longleftrightarrow$ stable system with rational transfer function $\tilde{H}(s)$,
      - $\cdot\ s \to \frac{2}{T}\frac{z-1}{z+1} \quad \Longleftrightarrow \quad z \to \frac{sT/2+1}{sT/2-1}$,
      - $\cdot \implies \quad \mathrm{Re}\, s < 0 \leftrightarrow |z| < 1$;
  - in the case of systems characterized by rational transfer functions $\tilde{H}(\omega) = N(\omega)/D(\omega)$, with degree$[N(\omega)] <$ degree$[D(\omega)]$, superiority of bilinear transform with respect to other transforms (for example the one in which $\tan(\omega T/2)$ is replaced by $\sin(\omega T)$), due to a desirable behaviour in the neighbourhood of the Nyquist frequency.
- Example:

- first-order low-pass filter simulator via bilinear transform;
- implementation via a difference equation;
- frequency response.

(28/10/2024) 8 **Filter design.**

- A recap on designing a first-order low-pass filter via bilinear transform:
  - derivation of $V(z)$ via bilinear transform;
  - basic properties,
    * reality,
    * causality (to be imposed),
    * BIBO–stability;
  - desirable behaviour in the neighbourhood of the Nyquist frequency;
  - implementation via a difference equation;
  - transfer function $\tilde{V}(\omega)$ via "backward interpretation of the simulation theorem".
- Structure of a filter implementing a rational function in the z–domain:
  - effect of zeroes and poles on $\tilde{V}(\omega)$;
  - example: design of a bandpass filter at $f = f_{Nyquist}/4 = f_{sampling}/8$:
    * positioning of zeroes and poles, by taking into account reality, causality, BIBO–stability,
    * final expression for $V(z)$,
    * a discussion on the frequency response $\tilde{V}(\omega)$,
    * inversion of z–transform $V(z)$ and practical implementation;
  - example: design of a notch filter at $f = f_{Nyquist}/2 = f_{sampling}/4$:
    * positioning of zeroes and poles, by taking into account reality, causality, BIBO–stability,
    * final expression for $V(z)$,
    * frequency response $\tilde{V}(\omega)$,
    * inversion of z–transform $V(z)$ and practical implementation.

(04/11/2024) 9. **Discrete–Fourier–Transform (DFT).**

- Discrete–Fourier–Transform (DFT):
  - assumption of the periodicity of $x_a(t)$, with period NT, $N \in \mathbb{N}^+$ and calculation of the coefficients of the Fourier series;
  - assumption of the $\frac{\pi}{T}$–band–limitedness of $x_a(t)$;

  - (assumption of N being even);
  - derivation of DFT and relation with the Fourier series coefficients;
  - inversion, periodicity, Parseval's theorem.
- Matrix representation of DFT:
  - notation,
    * $f_n \equiv f[n]$,
    * *twiddle factor* $W_N \equiv e^{2\pi i/N}$;
  - matrix representation;
  - dependence $O(N^2)$ of the number of operations required for the DFT on the dimension N of the input string.
- Example: DFT of $\cos(2\pi f t)$ with $f = 2 \cdot \frac{1}{NT}$ in the case $N = 8$.
- A short mention on DFT (and FFT) of a nonperiodic function and windowing.

(11/11/2024) 10. **Fast–Fourier–Transform (FFT).**

- A summary of last lecture.

- *Radix–2 decimation–in–time* Fast–Fourier–Transform (FFT) algorithm:
  - short history: Gauss 1805, Cooley and Tukey 1965;
  - Danielson–Lanczos lemma (*decimation*, i.e. split);
  - *butterfly* diagram (basic computational unit);
  - element ordering via *bit swapping*.
- Dependence $O(N \log N)$ of the number of operations required for the FFT on the dimension N of the input string.

- Examples of FFT in the case $N = 8$:
  - FFT of $\cos(2\pi f t)$ with $f = 2 \cdot \frac{1}{NT}$;
  - generic sequence $\{f_0, f_1, \ldots, f_7\}$, and equivalence with the DFT;
  - triangular wave $\{0, 1, 2, 3, 4, 3, 2, 1\}$;

  - ...proposed as a homework,
    * FFT of $\sin(2\pi f t)$ with $f = 2 \cdot \frac{1}{NT}$,
    * FFT of a constant,
    * FFT of $\delta[n]$,
    * FFT of $\delta[n-3]$,
    * FFT of $\delta[n-4]$.

(18/11/2024) 11. **An introduction to information theory: Shannon's source coding theorem for symbol codes.**

- The basic issue: a definition of information, i.e., how it can be measured:
  - Shannon's approach (1948) with uncertainty;
  - example of a sport match.

- Convex functions and Jensen's inequality:
  - convex functions and strictly convex functions;
  - Jensen's inequality,
    - proof of Jensen's inequality,
    - corollary: equality in the case of a strictly convex function.
- Kullback–Leibler divergence and Gibbs' inequality:
  - relative entropy, aka Kullback–Leibler divergence between two probability distributions;
  - Gibbs' inequality.

- Ensambles (Khinchin's finite schemes; "Experiment") as triplets made of. . . :
  - random variable;
  - alphabet of symbols (or vocabulary of words), each corresponding to a realization of the r.v.;
  - probability distribution of the r.v.

- Binary symbol codes and compression issue:
  - definitions related to symbol codes,
    - (binary) symbol code, codewords, length of codewords,
    - uniquely decodable symbol code and prefix code,
a prefix code is uniquely decodable; the contrary is not true (counterexample {1, 101}),
a prefix code is (generally) easy to decode,
    - examples;
  - expected length $L(C, X)$ of a symbol code C that encodes an ensamble X;
  - compression issue: given an ensamble X, generation of a symbol code C that. . .
    - (compulsory) is uniquely decodable,
    - (desirable) is easy to decode (so, possibly, a prefix code),
    - (compulsory) minimizes the expected length $L(C, X)$;

- Kraft inequality in the case of unique decodeability,
  - expression and proof,
  - definition of a complete symbol code,

- Shannon's source coding theorem for symbol codes.

(25/11/2024) 12. **An introduction to information theory: Shannon entropy of an ensamble; compression.**

- A summary of last lecture on Shannon's source coding theorem:
  - (uniquely decodable, and possibly prefix) binary symbol codes and compression;
  - expected length $\ell(C, X)$ of a symbol code C encoding an ensamble X;
  - compression issue: given an ensamble X, generate a symbol code C that...
    - is uniquely decodable, possibly a prefix one, and
    - minimizes the expected length $\ell(C, X)$;
  - statement of Shannon's source coding theorem (part I): $\ell(C, X) \geqslant H(X)$.
- Shannon entropy of an ensamble, and a mention of uniqueness theorem and of noisy channel coding theorem:
  - from "information" to "entropy" (a uniform distribution maximizes $H(X)\ldots$);
  - a mention of,
    - Shannon's source coding theorem, part II, namely a uniquely decodable code exists such that $\ell(C, X) < H(X) + 1$,
    - Shannon-Khinchin axioms and uniqueness theorem,
    - noisy channel coding theorem.

- Huffman *lossless* coding algorithm:
  - algorithm;
  - example;
  - properties,
    - Huffman algorithm generates prefix symbol codes (provable by construction),
    - Huffman is complete (provable by construction),
    - Huffman coding is optimal (statement only),
    - (a drawback:) necessity of knowing $p(x)$ in advance;

  - compression improvement via "syllables", and example with $p(a) = 1/4$, $p(b) = 3/4$.
- Lempel–Ziv *lossless* compression algorithm (LZ77):
  - algorithm;
  - example.
- A mention of lossy compression algorithms.

LABORATORY CLASSES

(04/10/2024) L0. **Software setup session.** (2 hours)

- Setting up a Linux environment.
- Walkthrough to set up Vivado.
- Compilation and device programming tests.

(08/10/2024) L1. **Introduction to Verilog programming on a FPGA device. Counters and frequency dividers.**

- A short overview on FPGA devices and on HDL programming languages.
- Introductory problem: design of a 1 Hz counter with 8–LED array display, relying on standard analog and digital circuitry and an 8 Hz clock source.
- Basic hardware circuits:
  - frequency divider;
  - counter.
- Verilog programming language:
  - module architecture;
  - template example: development of an 8–bit, 1 Hz counter with an 8–LED array display, working from 0 to 255;
  - basic modules implementing basic hardware circuits:
    * frequency divider;
    * counter.
- Problems:
  - development of an 8–bit, 1 Hz counter with an 8–LED array display, working from 0 to 9;
  - development of an 8–bit, 10 Hz counter with an 8–LED array display, working from 0 to 9.
  - development of a 10 Hz counter, working from 0 to 99, with a 2–digits BCD coding and a 2 x 4–LED array display;
- Additional problems:
  - development of a 10 Hz down counter.

(09/10/2024) L2. **Multiplexers and demultiplexers. Synchronous counters.**

- Solutions to the problems assigned in the previous lab class.
- Hardware and software architectures:
  - combinatorial (are made of *gates*, and depend on *states*) and sequential circuits (are made of *flip-flops [and gates]*, and depend on *clocks [and states]*);
  - synchronous and asynchronous circuits;
  - example: asynchronous counter and synchronous version by means of a finite-state machine.
- Basic hardware circuits:
  - multiplexer;
  - demultiplexer.
- Verilog programming language:
  - combinatorial (*assign*) and sequential (*always*) Verilog modules;
- Problems:
  - development of a stopwatch from 0 to 99.99 s (1/100 s resolution), with 2–digits BCD coding and a 2 x 4–LED array display;
  - development of a syncronous counter based on a *"master clock"*.

(10/10/2024) L3. **Toggle flip-flops. Monostable multivibrators.**

- Solutions to the problems assigned in the previous lab class.
- Risetime issue when clocking a flip-flop.
- Verilog programming language:
  - basic modules implementing basic hardware circuits:
    - multiplexer (synchronous and asynchronous version);
    - synchronous counter with set to a preset value, and reset.
  - basic modules implementing basic hardware circuits:
    - synchronous toggle flip-flop [*];
    - synchronous monostable multivibrator [*].

  [*] to be developed within the problems.

- Problems:
  - implementation of a synchronous module **toggle flip-flop** and, through this, of a *push light switch* to switch on/off an LED; observation of the bouncing effect in a pushbutton;
  - implementation of a synchronous module **monostable multivibrator** and, through this, of a timer to switch on an LED for a 1 second;
  - implementation, by means of a monostable multivibrator and a toggle flip-flop, of an improved *push light switch* to switch on/off an LED; solution to the bouncing effect in a pushbutton.

(18/10/2024) L4. **Finite-state machines.**

- Solutions to the problems assigned in the previous lab class.

- Finite-state machines.

- Basic hardware circuits:
  - data latches;
  - shift registers.
- Verilog programming language:
  - peripheral device drivers (ex.gr.: OLED driver with display of the lap mode).

- Problems:
  - implementation of a synchronous 5-state finite-state machine with two pulse control;
  - final implementation of a stopwatch with 1/100 s resolution, start/stop, lap/reset function, and OLED display.

(23/10/2024) L5. **Nyquist–Shannon sampling theorem made real. Waveform generation.**

- Using the *2's complement* representation with ADC/DACs.

- Verilog programming language:
  - driver of the ADCs placed on the development board;
  - driver of the DACs placed on the development board.

- Problems:

  - evaluation of the Nyquist frequency and the transfer function gain (voltage-to-number-to-voltage) of a ADC-DAC feedthrough system;
  - implementation of a delayer;
  - implementation of sawtooth waveform generators.

(30/10/2024) L6. **Implementation of a harmonic oscillator.**

- Theoretical and experimental aspects linked to the development of a harmonic oscillator:
  - general discussion on the difficulty of implementing an oscillator;
  - from the differential equation of a forced oscillator to the z–transform of the simulator response function $V(z)$;
  - derivation of the difference equation;
  - setting of the boundary conditions for the *cosine* operation;
  - dependency of the working frequency $f_0 = \omega_0/(2\pi)$ on the parameter k, provided that $\omega_0 T \ll 1$:
    $f_0 = f_s/(\pi 2^{\frac{k}{2}+1})$, with $f_s = 1/T$.
- Practical demonstration of FFT windowing.
- Problems:
  - implementation and characterization of a harmonic oscillator.

(06/11/2024) L7. **Digital simulation of analog filters.**

- A summary on Bode diagrams (and on frequency roll-off in filters).
- Theory problem: design of a first-order low-pass filter via bilinear transform:
  - from the Fourier transform of the real system's response function to the z–transform of the simulator's response function $V(z)$;
  - difference equation and block diagram of the simulator;
  - implementation by using a pole placed at $1 - 2^{-k}$;
  - dependency of the cutoff frequency $f_{3\,\mathrm{dB}} = (2\pi\tau)^{-1}$ on the parameter k, provided that $\omega_0 T \ll 1$;
  - frequency behaviour via "backward interpretation of the simulation theorem".
- Problems:

- implementation of a first-order low-pass filter;
- experimental check of the theoretical dependence of the cutoff frequency on the parameter k;
- assessment of the transfer function (to be displayed via Bode–diagrams) for a specific cutoff frequency.

(13/11/2024) L8. **Simulation of a final exam.**

- Simulation of a final exam.