Lab Class 5
14.10.2021

- **Driving the DACs and the ADCs hosted on the development board.**
- **Nyquist–Shannon sampling theorem made real.**

ADCs, DACs, drivers

ADCs

- Each one of the two ADCs placed on the board has a 14–bit resolution and works within a 2.5 V range centered about the common–mode value of 1.65 V (half of the power supply voltage). Consequently, V_{in} belongs to the range [0.4 V, 2.9 V] (actually, $(2.9 - 2.5 \cdot 2^{-14})$ V). The ADCs work in 2's complement.
- The number n_{ADC} corresponding to the input voltage V_{in} is given in 2's complement as a function of V_{in} by:
$$n_{\text{ADC}} = 2^{13} \cdot \frac{V_{\text{in}} - 1.65\ \text{V}}{1.25\ \text{V}},$$
with $n_{\text{ADC}} \in [-2^{13}, 2^{13} - 1]$.
- The negation of the MSB of n_{ADC} (an operation the corresponds to add, modulo 2^{14}, 2^{13}) produces a number n'_{ADC} that is linked to the input voltage V_{in} as follows:
$$n'_{\text{ADC}} = 2^{13} \cdot \frac{V_{\text{in}} - 0.4\ \text{V}}{1.25\ \text{V}} = 2^{14} \cdot \frac{V_{\text{in}} - 0.4\ \text{V}}{2.5\ \text{V}},$$
with $n'_{\text{ADC}} \in [0, 2^{14} - 1]$.

DACs

- Each one of the four DACs placed on the board (only two are used in the present lab class and throughout the course) has a 12–bit resolution and works within the full power range [0 V, 3.3 V] (actually, $3.3 \cdot (1 - 2^{-12})$ V). The DACs do not work with 2's complement, but, rather, in *unipolar* way.
- The converted voltage $V_{\rm out}$ corresponding to the number $n_{\rm DAC}$ is given by:

$$V_{\rm out} = 3.3 \text{ V} \cdot \frac{n_{\rm DAC}}{2^{12}} \, .$$

Drivers for the ADCs and the DACs

- The module "ADC_Driver" has a 28–bit output port Va_Vb that jointly returns $\{Va, Vb\}$, respectively for the two ADCs A and B.
- The module "DAC_Driver" has two 12–bit input ports Va, Vb, respectively for the two DACs A and B.

Module "nyquist.v"

- The module "nyquist.v" relays the number converted from the ADC A(B) into the DAC A(B) with an amplification of $3.3/2.5 = 1.32$. Here is why:
  - Let wb_Va, which is returned by the module "ADC_Driver", the 14–bit wire-bus that contains $n_{\rm ADC}$ for the ADC A. The number is fed to DAC A through

$$.Va(!wb\_Va[13], \, wb\_Va[12:2]) \, ,$$

    an operation that corresponds to dividing wb_Va by 4.
  - Consequently, provided that $V_{\rm in} \in [0.4 \text{ V}, 2.9 \text{ V}]$, one has:

$$V_{\rm out} = \frac{1}{4} \cdot \frac{3.3 \text{ V}}{2^{12}} \cdot \frac{2^{14}}{2.5 \text{ V}} \cdot (V_{\rm in} - 0.4V) = 1.32 \cdot (V_{\rm in} - 0.4V) \, .$$

Preliminary Operations

- **set the waveform generator so that the output voltage has an offset value of approximately 1.65 V and an amplitude of 1 V; check the result by using the oscilloscope**;
- connect the BNC cables **(avoid connecting the generator to the DAC output!)**;
- upload on the board via impact the executable file "nyquist.bit".

Problems

1. Nyquist frequency assessment.
   For each of the two positions of the leftmost switch (board pin "T9"), by suitably using the oscilloscope, determine:
   - the system's Nyquist frequency with a precision of at least 1 Hz;
   - the delay of both channels, A (ADC A → DAC A) and B (ADC B → DAC B);
   - the amplification of each channel.

   Record the two sets of three data each in the file `measurement.txt`. The file has to be sent along with the other files (in the same folder; see below).

2. Implementation of a delayer.
   Upon...
   (a) using the files of the project that generates "nyquist.bit",
   (b) using the negation of DAC_Driver's output *dacNumber* as the clock,

   carry out the following tasks:
   - modify the project so that DAC_A relays ADC_A as before, whereas DAC_B relays ADC_A with a delay of a single sampling period (beware: ADC_B is not used).

At the end, show the result to the lecturer.

Upon eliminating the unuseful files (only *.v*, *.ucf*, *.xise* are necessary), compress each working folder via
`tar czf labClass_5_<names>.tgz <Folder>`
and upload the resulting compressed file to the Moodle platform.

Additional problems

- By further modifying the project so that DAC_A relays ADC_A as before, implement the differentiator $h[n] = \delta[n] - \delta[n-1]$. Beware:
  - DAC_A has to relay ADC_A as before;
  - the differentiator has to be output on DAC_B as an additional option to the delayed relaying implemented above;
  - the output choice between differentiator and delayed relaying has to occur via a switch;
  - again, ADC_B is not used.

- Implementation of the improved differentiator $h[n] = \frac{1}{2}(3\delta[n] - 4\delta[n-1] + \delta[n-2])$ (similar guidelines as in the previous case have to be followed).

Upon eliminating the unuseful files (only *.v*, *.ucf*, *.xise* are necessary), compress each working folder via
`tar czf labClass_5_<names>_additional.tgz <Folder>`
and upload the resulting compressed file to the Moodle platform.

At the first favorable circumstance, show the result to the lecturer.