Departament de Disseny i Programació de Sistemes Electrònics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# Embedded Systems
# Final exam. June 17, 2019

Duration: 2 hours. Last revision day: June 26

## 1 Embedded systems

1. In the context of of the subject *Embedded Systems* try to link: ubiquitous computing, invisible computing, pervasive computing, ambient intelligence, embedded systems, cyber-physical systems, Internet of things, industry 4.0 and real-time systems.

2. Comment on the following sentence: *Cyber-physical systems must be dependable.*

3. Define *reliability*, *repairability* and *availability*.

## 2 ADC: changing the analog reference

During the project course we have used different platforms to detect a DTMF signaling. When working with Arduino we have taken the eight most significant bits of the 10-bit ADC of the AVR microcontroller with a range that goes from ground to 5 V. When working with DE0-Nano we have taken the eight most significant bits of the 12-bit ADC, external to the FPGA, with a range that goes from ground to 3.3 V.

When testing one of the DTMF signal test with FPGA we used $th_{FPGA} = 4 \times 10^6$ (i.e. what during the project we called the threshold, a level with which to compare the power of each one of the DFT components computed by the Goertzel algorithm).

1. Find out the value of the threshold, $th_{AVR}$, that should be used with the Arduino to obtain the same results with exactly the same DTMF signal test. Consider that in all platforms we use the same DTMF signal test, with a peak-to-peak amplitude of $3\,V_{pp}$.

## 3 DTMF project course

You want to design your own DTMF signaling (taking windows of 25 ms) increasing the eight frequencies used by the standard to the maximum allowed by your chosen platform. This way, instead of 4_low_freq x 4_high_freq =16 DTMF signals when using 8 different frequencies, you will have 4_low_freq x 5_high_freq =20 DTMF signals when using 9 different frequencies, 5_low_freq x 5_high_freq =25 DTMF signals when using 10 different frequencies, and so on. You are in the stage of choosing your development platform.

1. Based on the following results, obtained during this DTMF project course, make your decision:. Arduino/ATmega328P: 45% CPU usage. DE0-Nano/Altera Cyclone IV EP4CE22F17C6N FPGA: 1245/22300 logic elements; 16/132 multipliers. RasPi B+: 20 ms computation time; consider zero reading time from the system serial buffer. RasPi 3 B+: 4 ms computation time; consider zero reading time from the system serial buffer.

## 4 Software implications

1. Design a notch filter with the direct form I and II. Compare the two designs considering hardware cost and quantization noise.

2. Give an example of underflow when using floating-type arithmetic.

3. Describe a good design practice when you have to deal with a filter of order five.

## 5 FPGA and VHDL: synchronizing clock enable with data

A signal called `clk_en_in` is high the first rising edge of `clk` after `data_in` is ready and is low the next rising edge. We want to design an entity in order to update `data_out` from `data_in` every two `clk_en_in`. In addition, a signal called `clk_en_out` must be high the first rising edge of `clk` after `data_out` is ready and must be low the next rising edge. The following `VHDL` code tries to do that.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity se_exam_2017 is
  port (clk       : in  std_logic;--100kHz
        clk_en_in : in  std_logic;-- 10kHz
        data_in   : in  std_logic;
        clk_en_out: out std_logic;
        data_out  : out std_logic
        );
end entity;

architecture arch_1 of se_exam_2017 is
  signal n : unsigned(7 downto 0):= to_unsigned(0,8);
begin
  process(clk)
  begin
    if rising_edge(clk) then
      if clk_en_in = '1' then
        if n = 1 then
          clk_en_out <='1';
          data_out <= data_in;
          n <= to_unsigned(0,8);
        else
          clk_en_out <='0';
          n <= n+1;
        end if;
      end if;
    end if;
  end process;
end architecture;
```

1. Unfortunately, `clk_en_out` is not well generated. Draw the actual digital waveform (`clk`, `clk_en_in` and `clk_en_out`).

2. Modify the code and draw the right digital waveform.