Problem 1 (40 min, 4 pt.)

For the digital circuit described in VHDL, it is requested:

1. Complete sensitivity lists so that simulations and logical synthesis are correct and consistent.
2. Draw the hardware resulting from the logical synthesis of the P3 process. How many flip-flop will be needed?
3. Draw the state diagram of the state machine described in processes P1 and P2.

```
library IEEE;
use IEEE.std_logic_1164.all;

entity CIRCUITO is
  port (
    Clk    : in  std_logic;
    Reset  : in  std_logic;
    Valid  : in  std_logic;
    IsOpen : out std_logic);
end CIRCUITO;

architecture BEHAVIORAL of CIRCUITO is
  type     tStates is (s0, ok1, ok2, err);
  signal   CurrentState : tStates;
  signal   NextState    : tStates;
  constant cTopTimer    : natural := 18;
  signal   Timer1       : natural range 0 to cTopTimer;
  signal   EnaT         : std_logic;
begin

  P1: process(                                )
  begin
    if Reset = '1' then
      CurrentState <= s0;
    elsif Clk'event and Clk = '1' then
      CurrentState <= NextState;
    end if;
  end process P1;

  P2: process(                                )
  begin
    EnaT <= '1';
    IsOpen <= '0';
    case CurrentState is
      when s0 =>
        if Valid = '1' AND Timer1 = cTopTimer then
          NextState <= ok1;
        else
          NextState <= s0;
        end if;
```

```
      when ok1 =>
        if Valid = '0' then
          NextState <= s0;
        elsif Timer1 = cTopTimer then
          NextState <= ok2;
        else
          NextState <= ok1;
        end if;
      when ok2 =>
        IsOpen <= '1';
        if Valid = '0' then
          NextState <= err;
        elsif Timer1 = cTopTimer then
          NextState <= s0;
        else
          NextState <= ok2;
        end if;
      when err =>
        EnaT <= '0';
        if Valid = '1' then
          NextState <= s0;
        else
          NextState <= err;
        end if;
    end case;
  end process P2;

  P3: process(                              )
  begin
    if Reset = '1' then
      Timer1 <= 0;
    elsif Clk'event and Clk = '1' then
      if EnaT = '1' then
        if Timer1 = cTopTimer then
          Timer1 <= 0;
        else
          Timer1 <= Timer1 + 1;
        end if;
      end if;
    end if;
  end process P3;
end BEHAVIORAL;
```

Problem 2 (50 min, 6 pt.)

We want to design a digital circuit that processes a signal from an infrared sensor. This signal has a frequency of 100 Hz and must be multiplied by a sine reference (also 100 Hz) and then filtered to keep the continuous component of the product. The system has a 100 kHz clock (CLK) to perform operations. The circuit entity is:

```
entity LOCK-IN is
  port (
    Clk         : in  std_logic;
    Reset       : in  std_logic;
    ADC_ready   : in  std_logic;
    ADC_in      : in  unsigned(13 downto 0);
    Ref_in      : in  unsigned(13 downto 0);
    Data_out    : out signed(__ downto 0) );
end LOCK-IN;
```

The sampling rate is 1kHz, so we get a sample every 1 ms. The signal has 14 bits and comes from a digital to analog converter (ADC) that retrieves the signal in natural binary. However, this signal must be converted to an integer value (two's complement) because the ADC centers the signal on the medium value of the full scale (the value 0 of the signal is the value 8192 of the ADC). The ADC requires 200 µs to retrieve a value; when the data is converted, the input ADC_ready is activated during a single clock cycle.

Once the input signal has been converted to two's complement, it is multiplied by the sine reference (also 14-bit input, always available) and an 8-stage FIR filter is applied to the product. The coefficients of this filter have 6 bits and are configurable, so there is no optimization of the final result. They can be accesses as a constant array called aC(0 to 7) and already declared in a package.

For all questions, reason your answers.

1. Indicate how many CLK clock cycles are available between sample and sample of the signal.
2. Indicate how many CLK clock cycles are available for data processing once the ADC data has been acquired.
3. Write the concurrent statements or the combinational process that converts the data from the ADC (natural binary) to two's complement.
4. Indicate the number of bits the final result will have, if we do not want to truncate any intermediate operation.
5. We want to make a parallel architecture for the processing (truncating the result of each intermediate operation to 24 bits). Assume that the input is captured with the signal ADC_ready, and the output is captured in the next clock cycle.
   a) Describe the sequential process that records input data to the FIR filter.
   b) Describe the FIR filter using concurrent statements or a combinational process.
   c) Describe the sequential process that records the output of the FIR filter. Generate the required Load signal.
6. If only one multiplier block were available, draw the data path of the serial architecture to use. How many clock cycles are needed? Are there enough, according to your reply to question 2?