```
----------------------------------------------------------------------------------
-- Company:
-- Engineer: David Paquette
--
-- Create Date:    15:59:15 11/19/15
-- Design Name:
-- Module Name:
-- Project Name:
-- Target Device:
-- Tool versions:
-- Description:

-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity TemperatureControlMaster is
      Port ( clk_i : in std_logic;
          rst_i : in std_logic;
                adr_o : out std_logic_vector(31 downto 0);
          dat_i : in std_logic_vector(31 downto 0);
          dat_o : out std_logic_vector(31 downto 0);
          ack_i : in std_logic;
          cyc_o : out std_logic;
          stb_o : out std_logic;
          we_o  : out std_logic;
                   rx_out : out std_logic;
                   tx_in :  in std_logic;
                   incrementSetpointButton : in std_logic;
                   pwmOut : out std_logic;
                   decrememntSetpointButton : in std_logic
            );
end TemperatureControlMaster;

architecture Behavioral of TemperatureControlMaster is
      signal currentTemperature : integer range 0 to 100:=0;
      signal desiredTemperature : integer range 0 to 100:=32;
      signal fanSpeedPercent : integer range 0 to 100:=0;

      signal tx, rx, rx_sync, reset, reset_sync,tx_sig,onemsec_clk,pwm_clk :
            std_logic;

      signal eightBitBuffer : std_logic_vector(7 downto 0):=(others=>'0');
```

```
begin
            tx_sig <= tx_in;

            onemsec_clk_divider : entity work.clock_divider
                  generic map ( divisor => 100000 )
                  port map (
                        clk_in => clk_i,
                        reset => rst_i,
                        clk_out => onemsec_clk
                  );


      pwmFreqClock : entity work.clock_divider
                  generic map ( divisor => 2000 )
                  port map (
                        clk_in => clk_i,
                        reset => rst_i,
                        clk_out => pwm_clk
            );

      memoryWriter : entity work.MemoryWriter
            port map ( clk_i => clk_i, rst_i => rst_i ,
              adr_o => adr_o, dat_i => dat_i, dat_o => dat_o,
              ack_i => ack_i, cyc_o => cyc_o, stb_o => stb_o,
              we_o => we_o, currentTemperature=> currentTemperature,
              desiredTemperature=> desiredTemperature,
              fanSpeedPercent=> fanSpeedPercent
            );

      pidController : entity work.PIDController
            port map( samplingRateClock=>onemsec_clk,
                        reset=>rst_i,
                        setpoint=>desiredTemperature,
                        sensorFeedbackValue=>currentTemperature,
                        controlOutput =>fanSpeedPercent );

      temperatureSetPointControl : entity work.TemperatureSetpointControl
            port map(clk_i=>onemsec_clk,
                              rst_i=>rst_i,
                              incrementButton=>incrementSetpointButton,
                              decrementButton=>decrememntSetpointButton,
                              selectedTemperature=>desiredTemperature);

      temperatureSensor : entity work.TemperatureSensorInterface
            port map ( clk_i=>clk_i,
                                    rst_i=>rst_i,
                              temperatureCelcius=>currentTemperature);

      dcFanInterface: entity work.dcFanInterface
            port map(fanSpeed=>fanSpeedPercent,
                              --fanSpeed=>desiredTemperature,
                              pwmPinOut=>pwmOut,
                              clk_i=>pwm_clk);

      serialController : entity work.ValuesToSerial
      port map  (
```

```
                CLOCK       => clk_i,
                RESET       => reset,
                RX          => rx,
                TX          => tx,
            temperatureIn => eightBitBuffer+currentTemperature,
            fanSpeedIn => eightBitBuffer+fanSpeedPercent
            --fanSpeedIn => eightBitBuffer+desiredTemperature
    );

    process (clk_i, rst_i)
        begin
            if(rst_i='0') then
                reset <= '1'; -- the nexys4ddr is active low, so invert
                    reset to use with this serial lib
            elsif (clk_i'event and clk_i = '1') then
                reset <='0';
                rx_sync <= tx_sig; -- the perspective of the tx and rx
                    is reversed for the nexys
                rx   <= rx_sync;
                rx_out <= tx;
            end if;
    end process;

end Behavioral;
```