

Versuch 9 Protokoll, Gruppe 10

1.

Licht:

Konstanten festlegen für die gefragten Spannungswerte von 10, 50 und 200lux. Fallunterscheidung (if) je nach gemessenem Lichtwert wird die LED auf 0%, 25%, 50% gedimmt oder mit 100% eingeschaltet

Temperatur:

Da Sensordaten unsigned sind, zunächst überprüfen welcher Wert größer ist, dann die Differenz bilden. Liegt die Differenz in einem zuvor definierten Toleranzbereich von 4 Grad, werden Lüfter und Peltier ausgeschaltet. Sonst wird entsprechend gekühlt oder geheizt.

2.

Asha Regelung:

--! Standardbibliothek benutzen

library IEEE;

--! Logikelemente verwenden

use IEEE.STD_LOGIC_1164.ALL;

--! Numerisches Rechnen ermöglichen

use IEEE.NUMERIC_STD.ALL;

--! @brief ASHA-Modul - Regelung

--! @details Dieses Modul enthaelt die Regelung

entity AshaRegelung is

Port (

Clock : in std_logic;

--! Taktsignal

Reset : in std_logic;

--! Resetsignal

EnClockLight : in std_logic;

--! Enable-

Signal fuer die Lichtregelung

EnClockTemp : in std_logic;

--! Enable-Signal

fuer die Temperaturregelung

SensordataLight : in std_logic_vector(11 downto 0);

--! Aktuelle Lichtwerte

SensordataTempIn : in std_logic_vector(11 downto 0);

--! Aktuelle

Innentemperatur

SensordataTempOut : in std_logic_vector(11 downto 0);

--! Aktuelle

AuÄentemperatur

PWM1FanInsideValueControl : out std_logic_vector(7 downto 0);

--! PWM-Wert

innerere Luefter

PWM2FanOutsideValueControl : out std_logic_vector(7 downto 0);

--! PWM-Wert aeusserer

Luefter

PWM3LightValueControl : out std_logic_vector(7 downto 0);

--! PWM-Wert Licht

PWM4PeltierValueControl : out std_logic_vector(7 downto 0);

--! PWM-Wert Peltier

PeltierDirectionControl : out std_logic;

--! Pelier

Richtung heizen (=1)/kuehlen(=0)

ControlLightDiffOut : out unsigned(12 downto 0);

--! Aktuelle

Regeldifferenz Licht

```
ControlTempDiffOut : out unsigned(12 downto 0)    --! Aktuelle Regeldifferenz Temperatur
);
end AshaRegelung;
```

architecture Behavioral of AshaRegelung is

```
constant lux10 : unsigned(11 downto 0) := "111111011010"; -- 4058
constant lux50 : unsigned(11 downto 0) := "111101001000"; -- 3912
constant lux200 : unsigned(11 downto 0) := "110100100100"; -- 3364
constant tempDifference : unsigned(11 downto 0) := "000000010000"; -- 16
constant tempTolerance : unsigned(11 downto 0) := "000000001000"; -- 8
begin
```

-- Versuch 9: Realisierung der Lichtsteuerung

lightControl: process (Clock)

begin

```
    if rising_edge(Clock) then
        -- TODO
        if unsigned(SensordataLight) > lux10 then
            -- 255 -> 100%
            PWM3LightValueControl <= "11111111";
        elsif unsigned(SensordataLight) > lux50 then
            -- 128 -> 50%
            PWM3LightValueControl <= "10000000";
        elsif unsigned(SensordataLight) > lux200 then
            -- 64 -> 25%
            PWM3LightValueControl <= "01000000";
        else
            -- 0 -> 0%
            PWM3LightValueControl <= "00000000";
        end if;
    end if;
```

end process lightControl;

-- Versuch 9: Realisierung der Temperatursteuering

-- Ziel: Innen zwei Grad waermer als draussen

tempControl: process (EnClockTemp)

begin

-- TODO

if rising_edge(EnClockTemp) then

if unsigned(SensordataTempIn) >= unsigned(SensordataTempOut) then

if (unsigned(SensordataTempIn) - unsigned(SensordataTempOut)) <=

(tempDifference + tempTolerance) --Differenz < 3 Grad

and (unsigned(SensordataTempIn) - unsigned(SensordataTempOut)) >= (tempDifference - tempTolerance) then --Differenz > 1 Grad

PeltierDirectionControl <= '0';

```

        PWM1FanInsideValueControl <= b"00000000";
        PWM2FanOutsideValueControl <= b"00000000";
        PWM4PeltierValueControl <= b"00000000";
    elsif (unsigned(SensordataTempIn) - unsigned(SensordataTempOut)) >
(tempDifference + tempTolerance) then --> 3 Grad kühlen
        PeltierDirectionControl <= '0';
        PWM1FanInsideValueControl <= b"11111111";
        PWM2FanOutsideValueControl <= b"11111111";
        PWM4PeltierValueControl <= b"11111111";
    elsif (unsigned(SensordataTempIn) - unsigned(SensordataTempOut)) <
(tempDifference - tempTolerance) then --< 1 Grad heizen
        PeltierDirectionControl <= '1';
        PWM1FanInsideValueControl <= b"11111111";
        PWM2FanOutsideValueControl <= b"11111111";
        PWM4PeltierValueControl <= b"11111111";
    end if;

else

        PeltierDirectionControl <= '1';
        PWM1FanInsideValueControl <= b"11111111";
        PWM2FanOutsideValueControl <= b"11111111";
        PWM4PeltierValueControl <= b"11111111";
    end if;
end if;
end process tempControl;

```

-- Versuch 9: Ansteuerung der 7-Seg-Anzeige

SevenSegOutput: process (Clock)

begin

if rising_edge(Clock) then

-- Output für Siebensegmentanzeige

ControlLightDiffOut(11 downto 0) <= unsigned(SensordataLight);

ControlLightDiffOut(12) <= '0';

-- Fallunterscheidung da unsigned

if unsigned(SensordataTempIn) >= unsigned(SensordataTempOut) then

ControlTempDiffOut(11 downto 0) <= unsigned(SensordataTempIn) -

unsigned(SensordataTempOut); -- wenn Innen wärmer ist

ControlTempDiffOut(12) <= '0';

else

ControlTempDiffOut(11 downto 0) <= unsigned(SensordataTempOut) -

unsigned(SensordataTempIn); -- wenn Außen wärmer ist

ControlTempDiffOut(12) <= '1';

end if;

end if;

end process SevenSegOutput;

end Behavioral;