Exercise 4: Fan control and temperature sensor

G. Kemnitz[*], TU Clausthal, Institute of Computer Science

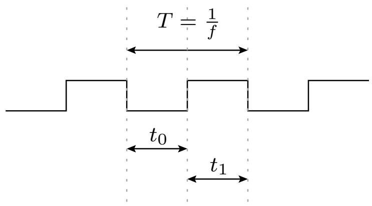
May 23, 2012

Abstract

A fan has to be controlled by a pulse width modulated (PWM) signal. From the PWM signal of a temperature sensor the temperature should be determined and displayed hexadecimal in grad Celsius at the 7-segment display.

1 Pulse width modulation

In a digital signal with constant frequency f the duty cycle is the ratio between the duration of the high value and the signal period $T = \frac{1}{f}$ (figure 1).



$$DC_0 = \frac{t_0}{T} = t_0 \cdot f$$

$$DC_1 = \frac{t_1}{T} = t_1 \cdot f$$

Figure 1: Definition of the duty cycle

A clock signal usually has the duty cycle $DC_1 = DC_0 = 50\%$. The duty cycle can vary continuously, so in PWM signals continuous values can be coded. First exercise is to design a circuit to control the speed of the fan at the corresponding expansion board on plug A1 via PWM. With a large duty cycle DC_1 the fan should move fast and with a small duty cycle it should move slow. (figure 2). The value of the duty cycle should be adjustable via the switches sw1 to sw3 in eight steps.

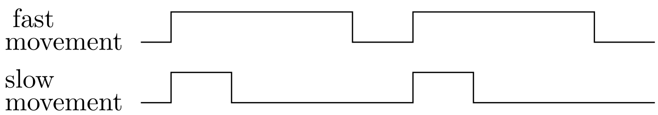


Figure 2: PWM signal for fan control

2 Temperature sensor

On the expansion board is also a temperature sensor. It is behind the fan and o type SMT160. This sensor summits the temperature PWM coded. According to the data sheet the relation between temperature and pulse width is:

$$DC_0 = 0,32 + 0,0047 \cdot T$$

*Tel. 05323/727116

(T – temperature in °C). The temperature to be calculated is in dependence of the duty cycle:

$$T = \frac{DC_0 - 0,32}{0,0047}$$

To calculate the duty cycle DC_0 the PWM output signal of the temperature sensor must be sampled with a high enough frequency over a longer period of time (figure 3).

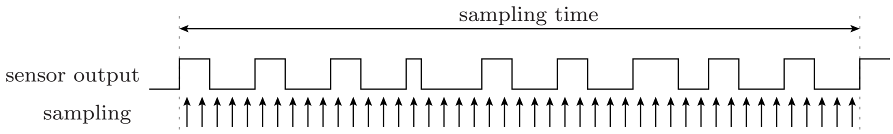


Figure 3: Sampling of the PWM output signal of the temperature sensor

During this the number of »low values« are counted and the result divided by the number of all sampling values. In the figure 45 values are sampled and 27 times the sampling value is zero. Inserted into the formula the duty cycle is

$$DC_0 = \frac{n_0}{N} = \frac{27}{45} = 0,6$$

and the temperature:

$$T = \frac{0,6 - 0,32}{0,0047} \approx 60°C$$

To calculate the temperature by hardware the division should be avoided by using a power of two for the divider. For a hardware divider costs at least a state machine, a subtracter, a block shifter etc. (see [**?**, p. 177]). The division by $N = 2^k$ is a right shift by k bits. The division by 0,0047 can be substituted by a multiplication with the reciprocal value. The simplified formula to calculate the temperature in degree Celsius is:

$$T \approx \mathrm{srl}\,(213 \cdot n_0,\, k) - 68$$

(srl(a, b) – right shift of a by b bits).

3 Test and modification of the example design

Download the files a4_pwm.xise, a4_pwm.vhd, and a4_pwm.ucf from the web page and copy them in the working folder in your home directory. Open the project in ISE. Confirm that the add-on board with the fan and the temperature sensor is plugged to A1 of the FPGA board and ready for work:

- 12 V power supply voltage (red LED beside the power connector is on).
- Ein/Aus-switch on the expansion board »on« (ein), (red LED over the switch is on).
- Other both switches on the expansion board to position P1.1 and P1.0. respectively.

The following exercises has to be solved:

1. Translate the project, program the board and test the circuit. With switch SW1 the fan can be turned on and off. Try to control fan speed manually.
2. Change the fan control so that eight different fan speeds can be set by the switches SW1 to SW3.

3. Plug the logic analyzer to pin ≫DB0≪ of the expansion board plugged to A2 and display the output of the temperature sensor. The required xml file with trigger conditions etc. should be created by yourself. Estimate for the recorded signal the duty cycle DC_0 and calculate the temperature from it. Turn on the heating resistor by SW0, to increase temperature slowly. Wait until the sensor has warned-up noticeable. Record the sensor signal again with the logic analyzer an determine the temperature again.

4. Design in VHDL the functional description of a circuit,

   - which samples the input with a frequency of 400 KHz and with $N = 2^{16}$ sampling points,
   - counts the number of low levels,
   - calculates the temperature for it and
   - displays the temperature values at the 7-segment display with two hexadecimal digits.

   Of the four 7-segment digits on the board the most left should be dark, the second and the third should display the two hexadecimal digits and the most right the character ≫h≪.

Note that arithmetic operations are only defined for the data types signed and unsigned from package ≫numeric_std≪ and for integer and natural. To add two operands of the type std_logic_vector, both operands first have to be converted by same named functions to ≫signed≪ or ≫unsigned≪ and the result back to ≫std_logic_vector. The addition of two bit vectors a and b as unsigned numbers has to be described by:

```
y <= std_logic_vector(unsigned(a) + unsigned(b));
```

4 Additional exercise

On the expansion board is also a RGB-LED. This LED can be also controlled by PWM signals to display different colors. Design a circuit, that allow to adjust color values for red, green and blue by for different duty cycles via two switches each. As the result it should be possible to select between 2^6 different colors. The pin assignment to the RGB LED can be found in the ucf-file as commends.

5 Check list for the compliance test

- Eight adjustable fan speeds.
- Two manually calculated temperature values for waveforms recorded by the logic analyzer, one without and the second with switched on heating resistor (keep waveform data and calculation way).
- Display of plausible temperature values on the 7 segment display. The values must increase if the heating resistor is switched on and decrease if the cooling fan is switched on.

References