DOI: 10.5604/20830157.1121364

THE CONTROL OF FAN SPEED USING FPGA BOARDS

Mayada Daboul, Ziad Nouman
Brno University of technology, Faculty of electrical engineering and communication, Department of Electrical Power Engineering

Abstract. *This article describes the implementation of a DC fan controller using a field-programmable gate array (FPGA). Hardware description language (VHDL) is used to design and implement the processor of this controller. The processor is composed of a memory block that has a function of generation a Look up Table (LUT). Data saved in the memory block are used to generate a triangular signal. A carried signal can be added. This signal can be generated from digital temperature sensors. A comparator compares between the triangular signal and the carried signal to generate a Pulse-Width Modulation (PWM) output that controls the fan speed. The fan speed depends on the digital temperature sensor output. When the output signal of the sensor equals the output of the triangular wave, the fan speed is maximal and the temperature is high. This design requires a FPGA board and software ISE 14.4.*

Keywords: fan speed control, FPGA, PWM, VHDL, digital temperature sensor

STEROWANIE PRĘDKOŚCIĄ WENTYLATORA PRZY UŻYCIU BEZPOŚREDNIO PROGRAMOWALNEJ MACIERZY BRAMEK (FPGA)

Streszczenie. *Artykuł przedstawia implementację sterownika DC wentylatora używając bezpośrednio programowalnej macierzy bramek - FPGA (ang. Field programmable gate array). Język VHDL (ang. Very High Speed Integrated Circuits Hardware Description Language) użyto do projektowania i implementacji procesora tego kontrolera. Procesor składa się z bloku pamięci, mającej funkcję generowania struktury nazywanej tablicą (ang. Look up Table – LUT). Dane zapamiętane w bloku pamięci używane są do generowania przebiegów trójkątnych. Komparator porównuje sygnał użyteczny, ten wygenerowany przez cyfrowy czujnik temperatury, z sygnałem trójkątnym, aby sygnał wyjściowy modulowany szerokością impulsu (ang. PWM) regulował prędkość wentylatora. Prędkość wentylatora zależy od sygnału wyjściowego z cyfrowego czujnika temperatury. Kiedy sygnał wyjściowy czujnika jest równy fali trójkątnej, wtedy prędkość wentylatora jest maksymalna a temperatura jest wysoka. Takie projektowanie wymaga FPGA i software ISE 14.4.*

Słowa kluczowe: sterowanie prędkością wentylatora, FPGA, PWM, VHDL, cyfrowy czujnik temperatury

Introduction

The cooling fans are widely used in computers and electrical equipment, because amount of heat is dissipated, and it must be in normal limits. Increasing of heat over the normal limit causes damage in components of electrical devices. The amount of heat differs from device to another. For example, in AC inverters, much of heat is generated by the switching transistors, in notebooks the heat is generated by the processors.

A heat sinks and heat pipes are used to remove the heat, but sometimes they are insufficient and the designers recommend improving cooling circuit using fans. A simple integrated circuit in the control system can add to drive a fan. The controlled fan generates airflow around the chassis and the heat generating components. There are many different types of fans and ways of controlling them [1].

PWM control is used for controlling the fan speed. The analog devices IC are used for generating PWM signals with low frequency. In this way, the voltage applied to the fan is always either zero or full scale. Figure 1 shows a control circuit with PWM signal for controlling the fan speed. The advantage of this circuit is that it is simple.

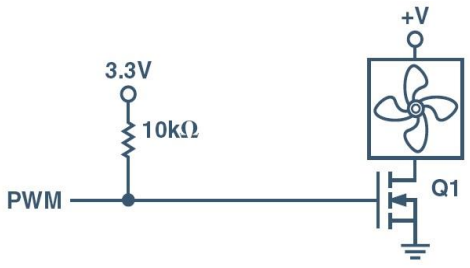

Fig. 1. Control circuit for fan speed

The PWM signal drives the fan by the drive FET. The disadvantage of this circuit is that the fan is on or off, and the change between these two modes causes an electric shock. This shock can make damage for components of the circuit.

Sometimes, analog devices are used for generating PWM signals. This method increases the cost of cooling circuit. In PC_s boards, a programmable device generates PWM signal. In FPGA boards, the same cooling circuit is used, and it has the same disadvantages.

This article proposes new software that generates PWM signal for driving the fan speed. The software is composed of block memory for generating a triangular wave, data input coming from the digital temperature sensor and comparator for comparing between signals.

1. Field-programmable gate arrays overview

Field-programmable gate arrays (FPGAs) are programmable digital logic chips that combine the best parts of Application Specific Integrated Circuits (ASICs) and processor-based systems.

FPGAs is distinguished from Application Specific Integrated Circuits (ASICs) that FPGAs can be reprogrammed to desired applications or functionality requirements after manufacturing while ASICs are manufactured for specific design tasks.

FPGAs are used by engineers in the design of specialized integrated circuits (ICs) that can later be produced hard-wired in large quantities for distribution to computer manufacturers.

FPGAs consist mainly of configurable logic blocks (CLBs) which are connected via programmable interconnect. Moreover, FPGAs include function logic blocks such as multipliers and memory resources like embedded block RAM.

- The basic logic units of an FPGA are the configurable logic blocks (CLBs) that are sometimes referred to as logic cells. These CLBs consist basically (as is shown in Fig. 2) of two basic components.
- Lookup Tables (LUTs): are used to implement function generators in CLBs, and it can be considered as a memory that includes the truth tables about implementation of all combinatorial logic (ANDs, ORs, NANDs, XORs, and so on). A truth table is a predefined list of outputs for every combination of inputs.
- Flip-flops: are binary shift registers used for synchronization logic and saving logical states among clock cycles within an FPGA circuit. Flip-flop latches TRUE or FALSE (1 or 0 values) on its input at every clock edge, holds that value constant until the next clock edge.

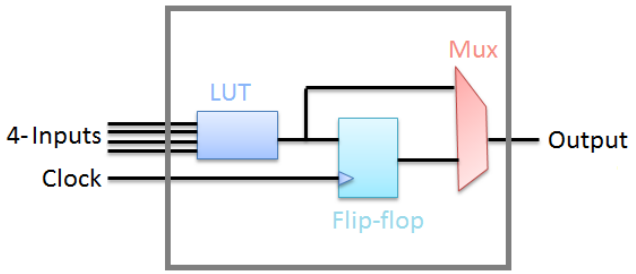

Fig. 2. Logic-cell in FPGA

The fundamental principle of the work FPGA chip can be described as a series of successive steps:

1) definition digital computing tasks in software using development tools:

Recently, the rules of FPGA programming are changed by the rise of high-level synthesis (HLS) design tools (e.g. NI Lab VIEW system design software) that delivers new technologies to convert graphical block diagrams into digital hardware circuitry.

Hardware Description Languages (HDLs) such as VHDL are evolved into the primary languages to design the algorithms running on the FPGA chip.

For a hybrid syntax, an external I/O ports are used to give signals connected to internal signals that are wired to the functions housed the algorithms.

Verification of the logic created by an FPGA programmer is important to write test benches in HDL to exercise the FPGA design. The test bench and FPGA code are run in a simulation environment which models the hardware timing behavior of the FPGA chip and displays all of the input and output signals to the designer for test validation.

2) Compilation the tasks down to a configuration file:

After the completion of the FPGA design using HDL and verification of it, there is a need to feed it into a compilation tool.

It takes the text-based logic and through several complex steps synthesizes down the HDL into a configuration file that contains information on how the components should be wired together.

However, it is important to appreciate how much is actually happening when block diagrams are compiled down to execute in chip.

In short, FPGAs have a great number of features such as unprecedented logic density increases, embedded processors and clocking; wherefore FPGAs are a better proposition for almost any type of design. Hardware-timed speed and reliability are provided using FPGAs.

2. PWM control

Figure 3 shows the signals that can be used for generating the PWM signal.

The duty cycle of PWM signal is given in the equation (1):

$$D = \tau / T , \tag{1}$$

where: $0 \le D \le 1$

The average value of PWM signal that control the drive is given in the equation (2):

$$V_{out} = D \times V_{in} . \tag{2}$$

The duty cycle value changes depending on the carried signal. If the value of carried signal (data from sensor) equals the value of triangular signal, the duty cycle is an approximate, and the equation (2) will be:

$$V_{out} \approx V_{in} . \tag{3}$$

Then the fan speed is maximal.

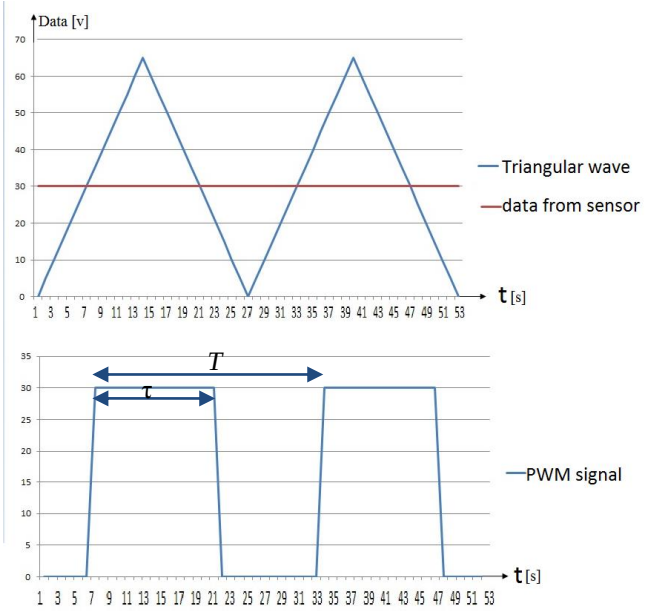

Fig. 3. Triangular signal, data from sensor, and PWM signal

2.1. Triangular Wave Generator

The triangular wave generator is a block memory that contains 103 samples. The values of these samples are between 0 and 255 with a step 5.

Table I shows some samples of the block memory.

Table I. Some samples of the block memory

Number of sample	Value in Decimal	Value in Binary
1	0	00000000
2	5	00000101
3	10	00001010
51	250	11111010
52	255	11111111
53	250	11111010
101	10	00001010
102	5	00000101
103	0	00000000

Table I shows that the sample are composed of two parts, the first part is the rising slope and the second part is the falling slope. The sample frequency is 50 MHz. the period of one sample is given as:

$$T_{sample} = 1 / f_{clk} , \tag{4}$$

$$T_{sample} = 1 / 50 = 20 \text{ nS} .$$

The period of triangular wave can be deduced by:

$$T_{triangularwave} = N \times T_{sample} , \tag{5}$$

where: N is the number of samples in the block memory.

$$T_{triangularwave} = 103 \times 20 \text{ ns} = 2.06 \text{ µs} .$$

The frequency of triangular wave is given by equation (6):

$$f_{triangularwave} = 1 / T_{triangularwave} , \tag{6}$$

$$f_{triangularwave} = 1 / 2.06 \approx 485 \text{ kHz} .$$

The block memory also contains a processor for controlling the output of sampling. The processor is a counter and the numerical value of increases with the rising clock. After the number (103) of samples, the triangular wave is formed.

From this result, the generator gives a triangular wave with frequency 485 kHz. Figure 3 shown this wave.

VHDL is used for building the block memory [3].

2.2. Comparator

The PWM signal is the output of the comparator [2], this comparator compares the data coming from the block memory and the data coming from digital temperature sensor. The fan speed depends on the duty cycle of PWM signal. The duty cycle increases, if the temperature increases and the fan speed will be maximal, when the duty cycle equals one.

Figure 3 shows the PWM signal. If the data coming from digital temperature sensor is greater than the data coming from the block memory by the counter, the PWM output will be high, and if it is less than data of triangular signal formed in block memory, the PWM output will low.

In addition to block memory and comparator, the design is composed of digital clock manager (DCM). DCMs provide advanced clocking capabilities for FPGA application [4]. Figure 4 shows the scheme of proposed design.

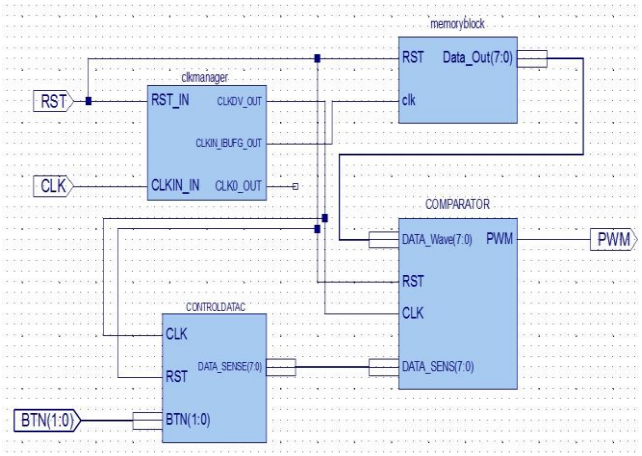

Fig. 4. The scheme of proposed desig

The block CONTROLDATAC is added for simulation the out of digital temperature sensor. This block generates 8 bits. Two push buttons can change the output of this block, according to the change of temperature, from 0bx00000000 to 0bx1111111. The output data present the carried signal that is compared with the triangular wave. VHDL is used for generating this block.

In figure 3 is shown that block CONTROLDATAC uses a clock signal coming from DCM. The frequency of this clock depends on the CLKDV_OUT. The value of CLKDV_OUT is given in equation (7):

$$CLKDV\_OUT = {f_{clk}}/{divider} \tag{7}$$

where: f_{clk} is 50 MHz, divider equals 10 in this design.
From equation (7), CLKDV_OUT can be deduced:

$$CLKDV\_OUT = {50}/{10} = 5 \text{ MHz}$$

The clock signal is used for increment the counter using buttons. RST signal is used for reset of the system to the beginning, where a push button is used to do this function.

3. Testing and Monitoring Signals of Design

The signals of design are tested using ChipScope Pro program. This program allows analyzing and monitoring the signals in the system [5].

First, the timing constraints code is built for configuration and using peripheries on FPGA board [6]. Figure 5 shows the architecture of this code.

```
# PlanAhead Generated physical constraints
NET "btn[1]" LOC = U15;
NET "btn[0]" LOC = T15;
NET "clk" LOC = E12;
NET "PWM" LOC = R20;
NET "rst" LOC = T14;
```

Fig. 5. Architecture of timing constraints code

Second, the LogiCore IP ChipScope Pro is added to the design. This IP core is a virtual input/output that can monitor and drive internal FPGA signals in real time.

A bit stream file is generated for download the design onto FPGA board.

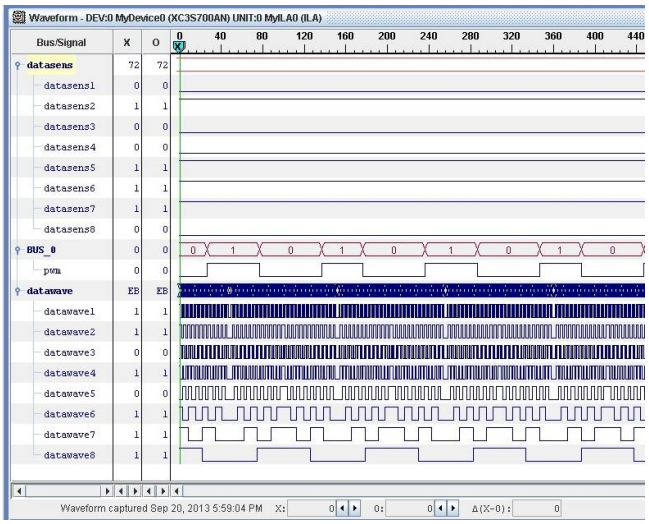

Fig. 6. Analysis of signals based ChipScope Pro

Figure 6 shows 3 buses of signals:

- First bus represents data coming from digital temperature sensor. This bus is composed of 8 bit.
- Second bus represents the samples of block memory. This bus is also composed of 8bits for generating the triangular wave.
- Third bus represents the PWM signal. This signal is the result of comparison between the data of previous buses.

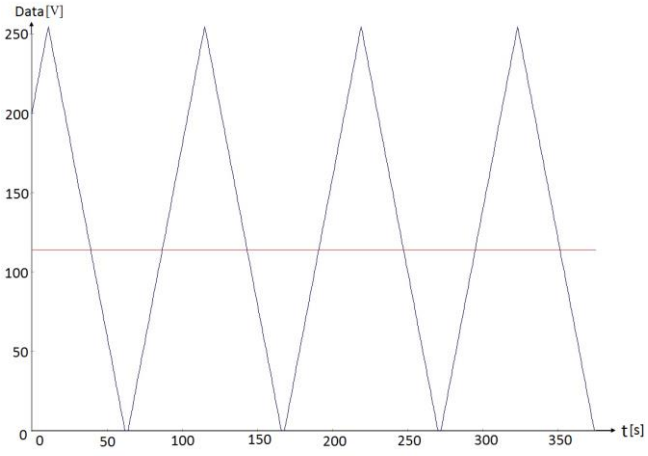

Fig. 7. Triangular wave with carried signal

Figure 7 shows the triangular wave and the carried signal. Triangular wave is the output of block memory, the full scale of this wave is 255 (maximam value is 255).

Carried signal represents one value of the present temperature. This value is variable according to the temperature of components in FPGA board.

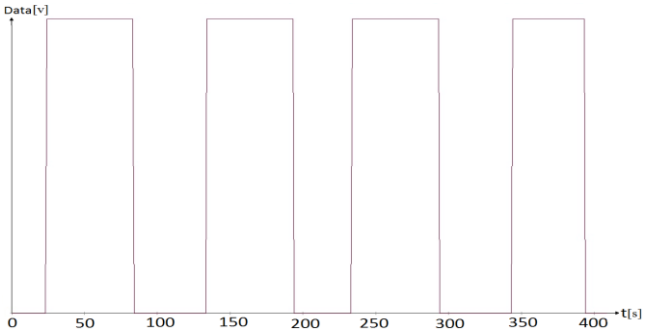

Fig. 8. PWM generator with variable duty cycle

Figure 8 illustrates the PWM signal that controls the drive of fan motor.

The value of PWM signal is one, if the carried signal is greater than triangular signal. The period of one cycle is calculated from equation (5), where it was 2.06 µs.

The duty cycle of PWM signal can be calculated depending on changing of digital output temperature sensor. If τ represents the value of data coming from digital temperature sensor, and T represents the full scale of data coming from block memory, the present duty cycle D can be calculated from equation (1). Figure 5 shown that τ is 0hx88, and the full scale of triangular wave is 0hxFF, then the duty cycle is 53.3 %.

The value of PWM signal can be calculated from equation (2). The voltage of bank in Spartan 3AN is 2.5 V, and the output voltage of PWM signal is Dx2.5 V. The output voltage changes from 0 to 100%x2.5 V, depending on the changing of duty cycle.

Figure (9) shows the PCB Circuit that can control the fan speed in FPGA board.

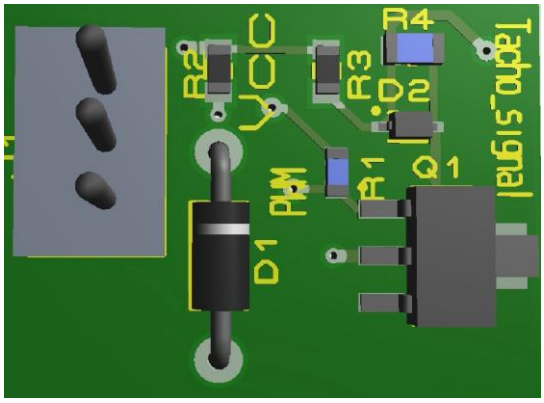

Fig. 9. PCB circuit of fan speed control in FPGA board

FET transistor is used as drive for controlling the fan speed. The gate threshold voltage is typical 2.4 V. PWM signal connects to the gate transistor directly, and it can't switch the transistor with output voltage between 0 and 2.5 V. Adding a pull up resistor with voltage 2.3, PWM signal with minimum output voltage will be able to switch the gate transistor, and control the fan speed.

If the fan has a tachometric output, the tacho signal pin can be connected to the FPGA pin for closed-loop speed control.

4. Conclusion

This article proposed a new method for controlling the fan speed in FPGA boards. This method is composed of two parts: the first part is software written using VHDL. Block memory generates the triangular wave. Carried signal is the data output of digital temperature sensor. Comparator compares between the triangular wave and the carried signal. PWM generator is the result of comparison.

The second part is hardware that represents the drive of the fan motor. The drive is FET transistor with gate threshold voltage 2.4 V. Pull-up resistor is used to implement the logic level for switching the transistor. A Tacho_signal pin is available, if the fan has a tachometric output.

ChipScope pro software is used for monitoring and analysis the data signals between the blocks of design. ISE 14.4 software and Spartan 3AN board are used for test the design. The design will be a part of FPGA board for improving the cooling system. This design can also be used in PC boards and all boards that need a cooling system. FPGAs are a better proposition for almost any type of design.

Acknowledgment

This work was supported by the European Regional Development Fund under project No. CZ.1.05/2.1.00/01.0014 and by the faculty project FEKT-S-11-14 "Utilization of new technologies in the power electronics".

References

[1] Burke M.: Why and how to control fan speed for cooling electronic equipment. Analog dialog 38-02, February 2004.
[2] KRÁL J.: Solved Examples in VHDL FPGA for beginners (czech language): Praha: BEN-Technical literature, 2010.
[3] PONG P.: FPGA Prototyping by VHDL examples Xilinx SPARTAN-3 VERSION. Hoboken, New Jersey,USA: John Wiley & Sons,Inc, 2008..
[4] XILINX. Spartan-3 Generation FPGA User Guide: Extended Spatan-3E, and Spartan-3 FPGA Families .UG331, V1.8. June 13, 2011.
[5] XILINX. Spartan-3A/3AN FPGA Starter Kit Board User Guide UG334.V14.3.October 16, 2012.
[6] XILINX.Spartan-3A/3AN FPGA Starter Kit Board User Guide UG334. V1.1. June 19, 2008.

Ing. Mayada Daboul
e-mail: xdabou00@stud.feec.vutbr.cz

Mayada Daboul (1986) is Ph.D. student at Faculty of electrical engineering and communication of Brno University of Technology.
She is researching at field of the central control systems for power substation.

Ing. Ziad Nouman
e-mail: xnouma00@stud.feec.vutbr.cz

Ziad Nouman (1980) is Ph.D. student at Faculty of electrical engineering and communication of Brno University of Technology.
He is researching at field of the usage the field programmable gate arrays in industrial automation.

otrzymano/received: 08.07.2014 *przyjęto do druku/accepted: 21.07.2014*