IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 4, Aug-Sept, 2013
ISSN: 2320 - 8791
www.ijreat.org

Development of FPGA based Closed Loop Speed Control of the Motor

P. K. Gaikwad
Department of Electronics,
Willingdon College, Sangli, India

***Abstract*— The prototype developed in this research work is useful to control the motor speed; immediately when there is desire of changing the speed. It also sustains the motor speed to the pre-set level even there is increase in the load coupled with the motor shaft. The fast processing for maintaining the motor-speed was possible to obtain due to the deployment of Field Programmable Gate Array (FPGA). The system developed here was tested on the Digital Storage Oscilloscope (DSO), by varying the pre-set value of motor speed input given to the Soft Intellectual Proprietary (IP) core implemented in the Xilinx FPGA Virtex-5 device. The FPGA produces 8 bit signal, which was further given to the Digital to Analogue Converter (DAC), whose output magnitude controls the speed of the motor. Such motor speed controlling prototype is applicable in Programmable Logic Control (PLC) based systems, which majorly uses the motors and requires fast motor speed control.**

***Keywords*— DAC, FPGA, Motor-Speed, PLC, Preset**

I. Introduction

The present paper describes the Field Programmable Gate Array (FPGA) based speed control of an electric motor. The speed of the motor was noted in real-time using a tachometer, which was constructed with the opaque and transparent lines on a disk coupled to the motor. An opto-coupler device was attached with a moving disk to collect square wave pulses; generated due to turning on and off of the photo transistor; housed in the opto-coupler. These pulses were input to the Soft Intellectual Proprietary (IP) Core implemented in the FPGA. The prototype developed here allows the user to set the motor-speed. Immediate to inputting the expected speed related values, the FPGA provides digital data out in such a way that, the motor speed gets sustained immediately to the used defined speed. An R2R ladder type Digital to Analogue Converter (DAC) was constructed to control the motor speed as per the magnitude of analogue output voltage produced by DAC. FPGA takes care of generating the digital input for the DAC, according to the user defined speed. The system takes the status of the speed of the motor and controls. For this reason it can be said as a Closed Loop Control System. The speed control of a motor is applicable in industrial applications; for example the systems based on Programmable Logic Controller, PLC.

As given in [1], the peed control of DC motors is possible by controlling Flux, Armature circuit resistance or applied voltage. In the present work the speed control of the motor is achieved by providing voltage to the motor. A set of slide switches was used to adjust the user defined speed. The Soft IP Core provides necessary digital data to the DAC input; consequently the analogue voltage controls the motor speed. To develop this system the Xilinx FPGA Virtex-5 board; developed by Digilent Inc. was deployed. Very High Speed Integrated Circuit Hardware Description Language (VHDL) source code was developed and implemented in Xilinx Integrated Software Environment (ISE), version 14.1. The Synthesis of the core was viewed at Register Transfer Level (RTL). A User Constraints File (.ucf) was written to interconnect the VHDL entity ports with input/output pads of the FPGA. The final result of the system was tested using a Digital Storage Oscilloscope (DSO) to show variations in output voltage of DAC according to the user speed; set by slide switches available on the Genesys board of FPGA. Hence the system can be classified as Closed Loop System.

II. Software Design Flow

The Fig. 1 shows a flow diagram to develop a VHDL source code and implement the same for motor control system. The Genesys board has several clock sources available, including a 3.3V 100MHz crystal oscillator, a socket for a user-supplied half-size DIP oscillator, and two high-speed and highly stable differential clock sources produced by an IDT 5V9885 programmable clock generator [2]. The on board clock source provided on the FPGA board was initially divided to obtain 100Hz clock. This clock division was done by deploying two clock divider VHDL modules provided in [3]. The 100Hz clock triggers a counter and update it according to the present status obtained from tachometer; pertaining to the speed of the motor. If the present speed is less than user desired speed, may be due to some increase in the load placed on the motor' shaft, then FPGA core increases the motor speed by providing binary data out in such a way that the DAC output voltage increases and the motor speed sustains with the user (inputting) speed.

The following lines of VHDL code show the instantiation of two clock divider entities and development of motor speed-control module.

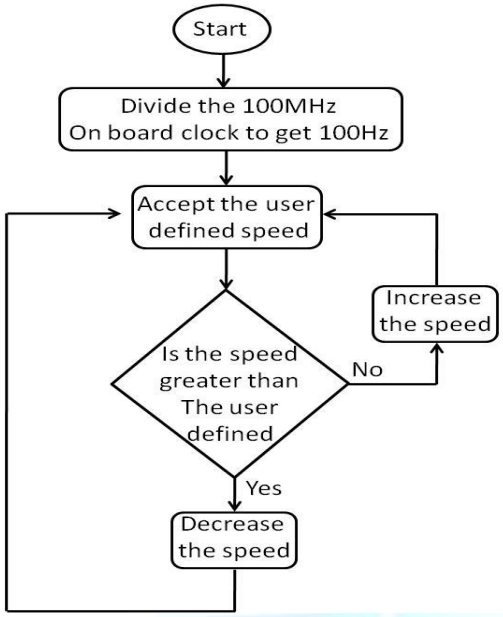

Figure 1. Software Development Flow to Sustain the Motor Speed with the user defined Speed.

```
entity pwmPLCmotor is
    PORT ( clock: IN STD_LOGIC;
           reset: IN STD_LOGIC;
           motor_pulse: IN STD_LOGIC;
           motor_speed_set: integer range 0 to 255;
           motor_out: INOUT integer range 0 to 255
          );
    end pwmPLCmotor;
    architecture Behavioral of pwmPLCmotor is
    component clkDivMain is
       Port ( clkSys100MHz: in     STD_LOGIC;
              clkOut100KHz  : out    STD_LOGIC;
              rst           : in     STD_LOGIC);
    end component;
    component clkDivSecondary is
       Port ( clkInt100KHz : in     STD_LOGIC;
              clkOut100Hz  : out    STD_LOGIC;
            early100Hz     : out STD_LOGIC;
            rst            : in     STD_LOGIC);
    end component;
    signal clk100KHz_int: std_logic;
    signal clk100Hz_int: std_logic;
    signal motor_pulse_counter: integer range 0 to 255;
    signal motor_speed_status: integer range 0 to 255;
    signal motor_out_sig : integer range 0 to 255;
    begin
    F100Mz_100KHz: component clkDivMain
          port map( clkSys100MHz => clock,
                    clkOut100KHz => clk100KHz_int,
                                rst => reset);
    F100KHz_100Hz: component clkDivSecondary
          port map( clkInt100KHz => clk100KHz_int,
                    clkOut100Hz => clk100Hz_int,
                                rst => reset);
    MOTOR_PULSES: process(reset,motor_pulse)
    begin
    if (reset='1') then
          motor_pulse_counter<=0;
          motor_out_sig<=motor_speed_set;
          elsif (motor_pulse='1') then
             if rising_edge(clk100Hz_int) then
                motor_out_sig<= motor_out_sig + 1;
                if (motor_out_sig > motor_speed_set) then
                     motor_out_sig<= motor_out_sig - 1;
                else
                     motor_out_sig<= motor_out_sig + 1;
                end if;
           end if;
    end if;
    end process;
    motor_out<=motor_out_sig;
    end Behavioral;
```

The VHDL code shows that the internal signal defined in the architectural part of the entity, 'motor_out_sig', keeps on update as on decrease or increase in the motor speed. The update process takes at every positive event on the clock signal of 100Hz; named as 'clk100Hz_int'. At the beginning stage of the process execution, i.e. when the process is reset, it accepts the user defined speed associated value present at the input port named as 'motor_speed_set' and store it in an internal signal 'motor_out_sig '. This signal is directly connected to the output port of the entity 'pwmPLCmotor',

which is further given to the input of the DAC; constructed by R2R ladder type of DAC.

III. RTL Synthesis View of the Soft IP Core

Modern FPGAs embed many pre-defined/fabricated IP components, such as arithmetic function units, embedded memories, embedded processors, and embedded system buses. These pre-defined building blocks can be modelled precisely ahead of time for each FPGA platform and, to a large extent, confine the design space [4]. The synthesis is a process of converting a source code into an implementation level, so as to hardwire it into the reconfigurable devices like FPGA or custom Application Specific Integrated Circuits (ASICs).

The Fig. 2 illustrates Register Transfer Level (RTL) view of the soft IP core developed for motor speed control. It consists of two entities for clock division process.

The Fig. 2 shows four components developed by VHDL source code. The component 'clkDivMain' input is connected to the input clock source of 100MHz. The output from this entity is given to the another module ' clkDivSecondary'. This further divides the 100KHz signal; produced by 'clkDivMain' and gives 100Hz clock signal. It is used to update the counter module shown as 'COUNTER:1' in the Fig.2. The fourth component whose instance name is 'GREATER:1' is selected from the FPGA component library due to the comparator code written the process of top level VHDL module developed in this research work.

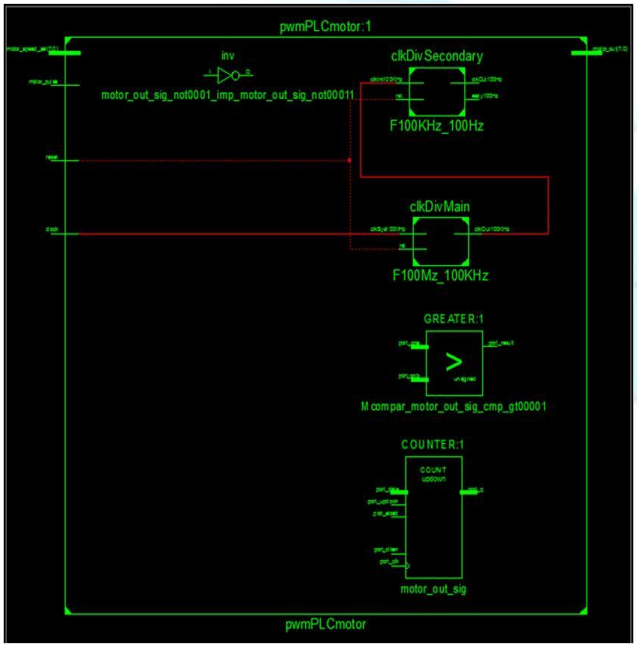

Figure 2. RTL Synthesis View of the Soft IP Core developed to motor speed control.

IV. Hardwiring Soft IP Core into the FPGA and Testing of the System

As the part of Xilinx ISE design flow, after synthesizing the VHDL code, a user constrains file (.ucf) was scripted in the necessary format. The Genesys FPGA board developed by Digilent Inc. provides a set of 8 slide switches. From these switches itself, the motor-speed data was input to the FPGA. The board also has four 12 pin Pmod connectors as shown in Fig 3. Output pads of the FPGA, associated to the top level entity were routed to one of the Pmod connectors. The details of these connectors and their FPGA pin connections are given in [2]. The entire project code was further taken to the process of Implementation. The bit-stream (.bit file) of the implemented project was then generated. Using Adept Software [5], developed by Digilent Inc. to download the bit-stream and many more other functions, the bit file was downloaded in the FPGA.

The Table I shows various speed related inputs, in response to which the FPGA produced 8 bit binary output. For that, the DAC produced various output voltages to control speed of the motor.

TABLE I. INPUT SPEED DATA CAUSING THE DAC OUTPUT VOLTAGE CHANGE

Input Port Data at 'motor_speed_set(7:0)'	DAC Output Voltage (V)
"11000000"	0.780
"11000011"	0.940
"11001111"	1.920
"11011111"	2.280

The Fig.3 shows a photograph taken at the time of actual testing of the system.

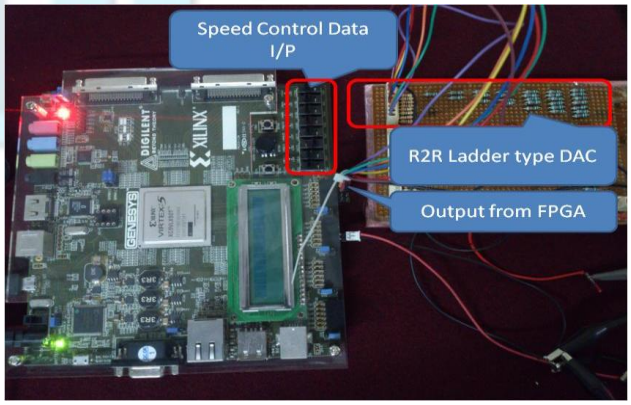

Figure 3. A prototype developed to test the DAC output for Motor Speed Control.

The Fig. 3 shows 8 input switches to provide motor-speed input (to be maintained by FPGA). The output of FPGA core was taken at Pmod connector (JA) and given to the input of R2R ladder DAC. The motor-speed was controlled by DAC output voltage. The output of DAC was observed on DSO.

The Fig. 4 illustrates one of the observations seen on the oscilloscope channel for an input data associated with "11011111". It shows that an output produced by DAC was 2.28 V; the same readings are also mentioned in Table I.

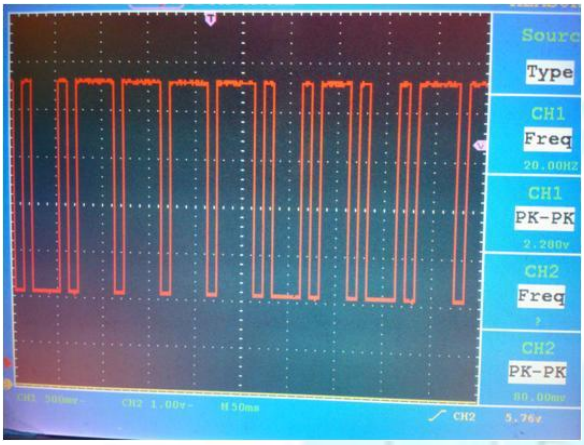

Figure 4. DAC Output waveforms observed on DSO for input "11011111".

The Fig. 4 show a square wave with peak to peak amplitude of the order of 2.28 V; stimulated to the channel number 1(CH1) of the oscilloscope. The amplitude produced from DAC causes to change the speed of the motor.

v. Conclusion

The present paper shows a potential of Hardware Description Language (HDL) utilized here for a system to control a closed loop processing. It monitors the status of the speed of the motor and sustains, regardless of increase in the load given to the motor. On the other hand immediately it accepts the user defined speed setting in run-time and controls the speed of the motor accordingly. A Xilinx FPGA Virtex-5 was deployed to implement the software IP Core developed in VHDL programming language to control the system.

References

[1] Viewed 10 August 2013 from http://www.uotechnology.edu.iq/dep-eee/lectures/2nd/Electrical%20machines%201/DC%20MACHINES%20part4.pdf.

[2] Support Document, "Genesys Board Refference Manual", Digilent Inc., 1300 Henley Court | Pullman, WA 99163, February 28, 2012, Doc: 502-138.

[3] Support Document, "Xilinx ISE demo project for the PmodAD2 and a PmodDA1", Digilent, Inc., Pullman, WA 99163-0428, November 2011, DSD-0000321.

[4] Jason Cong, Stephen Neuendorffer, Kees Vissers and Zhiru Zhang, "High-Level Synthesis for FPGAs: From Prototyping to Deployment", viewed 11 August 2013 from http://www.ecs.umass.edu/ece/labs/vlsicad/ece667/reading/hls-survey.pdf.

[5] Viewed 12 April 2011 from http://digilentinc.com/Products/Detail.cfm?NavPath=2,66,828&Prod=ADEPT2.