Journal of Control & Instrumentation
Volume 3, Issue 2, August 2012, Pages

Design of an Automatic Device Controller Based on Temperature Sensing by Using Verilog HDL Targeting an FPGA

Mahboob Qaosar[1], Shamim Ahmad[2]
Department of Computer Science & Engineering, University of Rajshahi, Bangladesh

ABSTRACT

To get a single controller that is capable to produce some control signal depending on sensing temperature is getting popularities for industrial and home applications. In this paper uses an LM34 Precision Fahrenheit Temperature Sensor IC and an ADC0820 8-Bit, high-speed, µP-compatible flash A/D converter IC in order to design and implement the mentioned controller onto a SPARTAN™-II Field Programmable Gate Array (FPGA) by using Digilent™ Pegasus SPARTAN-II FPGA Board, the Verilog HDL has been used for hardware description of the proposed controller. The controller has been designed to measure the temperature as well as to display the temperature on a 7-segment display and then, depending on the pre-defined or configurable temperature values, the controller is capable of generating control signals that could be used to control any other electrical or electro-mechanical devices like Fans, Heaters and etc. This approach is different from existing Temperature based Device Controller ICs in a sense that, the new approach allows user to set different ranges of temperature by setting jumpers or combination of switches or buttons, at which level of temperature range the user would like to get a signal to control external device(s).

Keywords: *Controller, Sensor, SPARTAN-II, FPGA, Verilog HDL*

* ***Author for Correspondence*** E-mail: qaosar@yahoo.com, shamim_cst@yahoo.com,
Tel No: +8801713140107

1. INTRODUCTION

Automated systems utilize sensors, controllers, drives, actuators, and self-regulating mechanism to maintain intended process conditions and progress. Automated controlling systems [1] eliminate tedious human labor and show to be more effective; accurate and faster, resulting in increment of productivity of any system.

In this paper, an automatic controller [2] has been proposed which is capable to produce different control signals based temperature sensing requited to control temperature-dependant devices of any system, the control is capable to produce control signals based preset temperatures range. This proposal uses an LM34 [3] precision Fahrenheit temperature Sensor, an ADC0820 [4] high speed flash ADC, the controller have been implemented onto a SPARTAN™-II FPGA [5-8]. The developed controller can automatically initiate the ADC0820 to refresh the voltage readings found from LM34 temperature tensor, and then the convertor converts the read voltages readings into equivalent temperature value. The controller is capable of showing current temperature reading on a seven segment display and depending on the preset temperature range; it can produce external signals those can be used to turn on/off external devices, i.e. heater, cooler, electro-mechanic devices. The temperature range can be preset by using some predefined combination of buttons and switches.

Moreover the controller is capable to read data from different location, however, as a test case, for a single location by using a single LM34 temperature sensor has been implemented. But the developed system can be extended for sensing temperature from more than one location with a minor change in configuration of FPGA and with additional LM34 temperature sensors required as per number of locations along with the single ADC0820 High Speed ADC used earlier. From the design specifications, the whole controller system has been coded in Verilog HDL, compiled, simulated, synthesized and implemented using the Xilinx® ISE 7.1i VLSI development suite onto a Digilent™ Pegasus Spartan-II FPGA board [8].

2. DESIGN METHODOLOGY

The automatic device controller based on temperature sensing has been designed targeting the LM34 temperature sensor IC along with ADC0820 8-Bit, high-speed, µP-compatible A/D converter IC. As we know LM34 IC converts the temperature into equivalent voltage output [3] and ADC0820 converts the analog DC voltage input into its binary equivalent comparing with the reference voltage [4]. The ADC0820 is set to RD mode, since the controller only reads the converted binary data from it however, in order to refresh the readings; it is required to generate clock pulse to initiate the conversion process. So the major task of the developed temperature-sensed device controller is to initiate the temperature measurement by generating pulse with minimum delay, then overwriting the register with the measured value read from ADC0820, finally, showing the current read-temperature on a seven-segment display in decimal form and to take decision to generate control signals to control external devices, the whole controller system has been coded in Verilog HDL, compiled, simulated, synthesized and implemented using the Xilinx® ISE 7.1i VLSI Development Suite onto a Digilent™ Pegasus Spartan-II FPGA Board [8], therefore it is possible to manufacture the controller a digital IC at industrial scale..

3. DESIGN IDEAS

The automatic device controller based on temperature sensing has been designed to generate signals to control electrical or electro-mechanical devices and also to display the temperature on a seven-segment display The temperature based device controller has been designed basically as a finite-state machine (FSM) having eight net or vector net as inputs and four net or vector net as outputs.

The Xilinx® ISE 7.1i VLSI development suite has been used for the whole design process of the device controller and for its implementation onto the Digilent™ Pegusus Spartan-II FPGA board. The ADC0820 8-Bit µP-compatible A/D converter specifies that at RD mode (MODE = LOW), it requires a ‘HIGH’ reset pulse at RD pin and following

that it should be set zero for at least 2.55 μs (2.50 μs conversion time for RD mode +50 ηs access time) to end conversion and making the data available at the data line. As a result, it requires to generating a pulse whose pulse width should be greater than two times of 2.55 μs, that is, 5.10 μs. Hence, the frequency should be less than 1/5.10 ≅ 0.2 MHz. As we know, Pegasus Spartan-II FPGA board has on board 50 MHz oscillator [8,14], in order to use this oscillator for pulse generation, a frequency divider circuit module has to be designed and described in Verilog HDL too. Each time the controller fetches the ADC reading and stores data at the internal register of the controller, a set of predefined relevant operations will be performed.

In order to display the read-temperature in form of decimal digits on a 7-segment display which is built-in on Pegasus Spartan-II FPGA bard, the respective binary data requires to be converted into BCD. The necessary codes have been developed in Verilog HDL [9-13] for this purpose. The A2 expansion connector's output of SPARTAN-II FPGA board has been used as positive reference voltage which is 3.3 V according to the Digilent™ Pegasus Spartan-II FPGA board specification. The BCD conversion has been achieved by BCD additions of BCD equivalent values for the each particular bit position at 8-bit vector register which stores the ADC's reading. For this purpose, Table I has been constructed where it has been considered the positive reference voltage as 3.30 volt and negative reference voltage as 0 volt (GND) to feed to ADC0820.

Table I: *Table for Converting Binary to BCD When Full Scale Output is 330.0.*

MSB←Binary Reading From ADC0820→LSB								
Bit	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
Value	166.0	82.8	41.4	20.7	10.4	5.1	2.5	1.2

The BCD values displayed on the 7-segments display is sum of corresponding BCD of vector register values of Table I.

The corresponding BCD value of a vector register has been calculated in following way.

Step 1: Begin

Step 2: Set Total=0

Step 3: Repeat Step 3 for Index = 0 to 7

Step 4: Total = Total + Value [Index]; If Bit[Index] =1

Step 5: End

For example, if the binary value stored at vector register is 01001010 then the BCD sum that has been shown at seven-segment display is 0 + 82.8 + 0 + 0 + 10.4 + 0 + 2.5 + 0 = 95.7, and that is the temperature displayed at Fahrenheit scale.

In order to generate signals those control external devices, a digital comparator module has been designed, however, for testing purposes; those signals have been shown by on-board LEDs. In order to check the functionalities of comparator, six different signals for six different temperatures ranges have been generated and experimentally those

are used to drive six LEDs correspond to six external electrical or electro-mechanical devices. It has been found that the signals were changed in according with the changes in temperature as read from LM34 via ADC0820.

The controller has been designed to compare the temperature either with default preset value or with reconfigurable value prior to generate any signal. A switch of SPARTAN-II FPGA board has been used to select the mode between preset value and reconfigurable value. The reconfigurable value can be changed by combination of switches and buttons. For example, for the default mode of setting, the output external signals that are generated according to the temperature reading as follows:

Table II: *Output External Signals Generated According to the Temperature Reading.*

Condition (T = Temperature) °F	LED Status (1 = ON, 0 = OFF)
$T \geq 90$	110000
$90 > T \geq 80$	011000
$80 > T \geq 75$	001100
$75 > T \geq 70$	000110
$T < 70$	000011

The default setting is changeable during design process.

The LEDs output for reconfigurable mode is as follows:

Table III: *The LEDs Output.*

Condition (T = Temperature, U = Upper, L = Lower) °F	LED Status (1 = ON, 0 = OFF)
$T \geq U$	110000
$U > T \geq L$	001100
$T < L$	000011

It has been considered that the upper and lower boundary values for a new setting of temperature range can be changed by combination of switches and buttons.

The three signals have been utilized for this purpose. Let the signal are named as *flagbit*, *select* and *wr*. In the proposed system, 4 digit BCD value has been considered as upper and lower boundary value.

1. The *flagbit* represents the BCD value that could be set as any digit of upper and lower boundary value.
2. The MSB of *select* decides which boundary value should be updated (i.e. Upper boundary value/Lower boundary value) and other two bits decide which digit position should be updated.
3. The signal *wr* produces a pulse to initiate the corresponding registers to letch the value.

For example, let's consider once a time the upper value of boundary is 080.0 and lower boundary is 070.0. In order to replace the upper and lower value to 090.0 and 075.0 respectively, someone have to set following states of signals, switches and buttons.

Table IV: *Values of States of Signals, Switches and Buttons.*

Select	flagbit	Before wr pulse		wr	After wr pulse	
		UPPER	LOWER		UPPER	LOWER
110	1001	080.0	070.0	↑	090.0	070.0
001	0101	090.0	070.0	↑	090.0	075.0

In order to view upper and lower values of boundary at reconfigurable mode, two buttons of SPARTAN-II FPGA board have been utilized; one for viewing upper boundary value, and another for viewing lower boundary value. If any one of that button is pressed, the corresponding boundary values are then shown on the seven segment display. If no button is pressed, the seven segment display shows the current temperature.

The 7-segment display timing of Pegusus Spartan-II FPGA board has been shown in Figure 1.

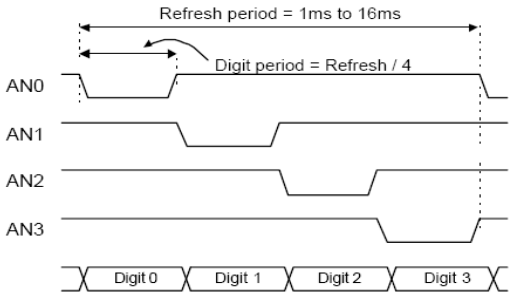

Fig. 1: *7-Segment Display Timing of Pegusus Spartan-II FPGA Board.*

According to the 7-segment display control specification; the design includes a module for frequency divider, counter, multiplexer and decoder. Therefore, in order to view temperature with decimal value on the seven segment display, a module has been designed that converts each BCD digit into corresponding 7-segment values to show the temperature on a 7-segment display.

4. STATE DIAGRAM

According to the design specifications of "Automatic Device Controller Based on Temperature Sensing", as described so far in previous section, the finite state machine (FSM) has been designed with the following eight states: *Start, S0, S1, S2, S3, S4, S5* and *S6*.

The entity of the "Device Controller" has *adcin, clock, sw, flagbit, select, wr, disp_upper*, and *disp_lower* as the main inputs and *signal, rd, segment*, and *an* as the main outputs. Besides these, there are some internal signals and registers: *t_reg, upper, lower, bcd, digit* and *q*.

The state diagram of the designed FSM has been shown in Figure 2. The FSM begins from *Start* state, and moves from one state to another according to changes in the input variable.

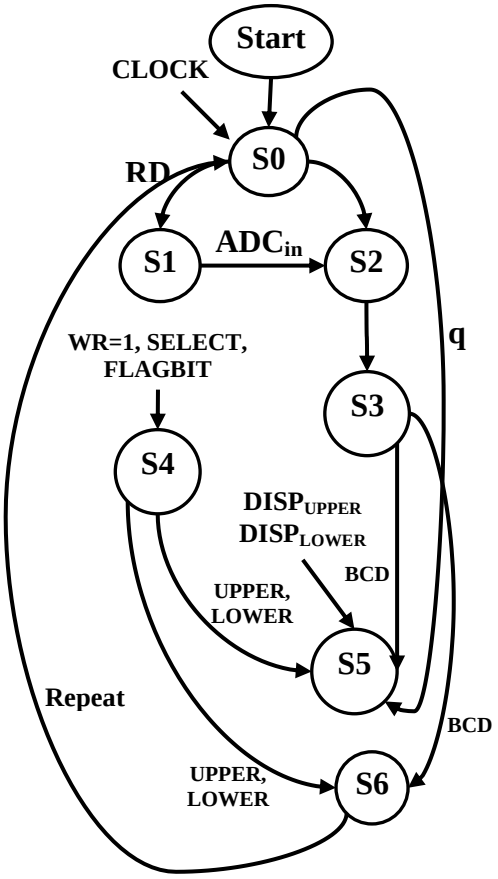

Fig. 2: *State Diagram of the Configurable Temperature Based Device Controller.*

At state *S0*, the *clock* input is passed to frequency divider module to divide input frequency and the output of frequency divider module has been used as output signal, *rd.* A counter has also been driven by the *clock* at this state, it's (the counter) output *q* has been used as the function of producing 7-segment display.

At state *S1*, the *rd* signal has been assigned as input signal to ADC0820 to its *rd* pin. The *rd* signal resets the ADC0820 by rising HIGH, and the ADC0820 starts A/D conversion and makes data available on its output ports, when *rd* signal falls to LOW from its HIGH state. The data that is available at the data output port of ADC0820 is named as *adcin*.

At state, *S2* each time when the *rd* is LOW, the data available at *adcin* is latched in vector register, *t_reg*. All the operations those are performed for comparison to produce external device control signals and for producing 7-segment display, are based on the value stored at *t_reg.*

In state *S3*, a combinational logic circuit is used to convert value stored at *t_reg* into *BCD*, the output of this state is a 16-bit 4-digit BCD, named *bcd*, which is equal to the temperature in decimal form. The combination of switches and buttons has been utilized at state *S4*, to set the configurable value, used for controlling signals controlling external devices.

In state *S5*, the signals for displaying current temperature or configurable upper or lower boundary value are generated. Two input signal $DISP_{UPPER,}$ $DISP_{LOWER}$ have been used to control output display.

Comparing of current temperature with default setting value or with configurable upper or lower boundary value and generation of signals to control external devices are performed at state *S6*.

The controller should continue the process of reading temperature from sensor, displaying temperature and generating signals to control external devices, until power is turned on or

someone does not presses reset button of the FPGA board.

The Verilog HDL code for the temperature based device controller has been designed according to the State Diagram of Figure 2. Each block of hierarchical model performs appropriate function to construct the system. The blocks of the hierarchical models are implemented using different modules. The module *top* has been designed, which has all the entities and contains instances of other modules. The other modules performs functions for, triggering ADC and reading binary data from it, BCD conversion of readings, customization of temperature boundary for controlling external device, display control, automatic device control based on temperature sensing.

5. SIMULATION WITH THE TEST BENCH

Behavioral Simulation has been performed to verify the design functions as expected. The simulation of the *stimulus* block has been generated to observe the design and verify the output with respect to time. The simulation of this test bench is shown in Figure 3.

The simulation result shows that the design is correct for implementation. The BCD signal changes according to change at ADC signal. Also the external signal changes its state according to the design specification.

Through the observations it has been seen that the test bench has given correct simulation according to the design, which in turn proved the Verilog HDL coding to be accurate and the behavioral simulation of the designed Automatic Device Controller based on Temperature Sensing has been confirmed.

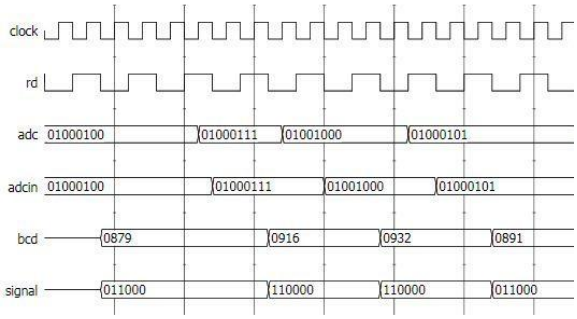

Fig. 3: *Test Bench Simulation of Configurable Temperature Based Automatic Device Controller.*

6. REGISTER TRANSFER LOGIC (RTL) SYNTHESIS

The Register Transfer Level (RTL) implies to the Verilog HDL code's description of how data is transformed as it is passed from register to register. The transforming of the data is performed by the combinational logic that exists between registers. Logic synthesis performs the RTL design into a large collection of lower-level constructs or standard cells. The resulting collection of standard cells, in addition to the required electrical connections among them, is called gate–level *netlist*. The RTL synthesis for the designed "Automatic Device Controller based on temperature Sensing" has been produced by

the XST from Xilinx® ISE 7.1i VLSI Design Suite.

Given below is the "device utilization summery" for the designed controller on Spartan-II FPGA.

Table V: *Design Summery for Automatic Device Controller Based on Temperature Sensing Selected Device: 2s50pq208-5.*

Logic Utilization	**Used**	**Available**	**Utilization**
Total Number Slice Registers:	51	1,536	3%
Number used as Flip Flops:	20		
Number used as Latches:	31		
Number of 4 input LUTs:	173	1,536	11%
Logic Distribution:			
Number of occupied Slices:	128	768	16%
Number of Slices containing only related logic:	128	128	100%
Total Number 4 input LUTs:	179	1,536	11%
Number of bonded IOBs:	38	140	27%
Number of GCLKs:	1	4	25%

7. RESULTS

As the design is implemented using the *Digilent™ Pegasus Spartan-II* FPGA board, all the pulse generator and counters have been designed according to the Pegasus Board's built-in 50 MHz clock. Since, to read converted value from the ADC0820 requires lower frequency; a frequency divider module has been designed. The frequency divider module has also been used fro visualizing temperature on the 7-segment display of *Digilent™ Pegasus Spartan-II FPGA* board.

The designed "Automatic Device Controller based on Temperature Sensing" has been simulated with an appropriate *test bench* according to the design specifications, to verify the design. In order to see the simulation properly, it was required to design the whole test bench for a much shorter duration and fewer entities than the actual design created for the controller. For this reason, the *test bench* has been created using a *stimulus* block with a modified version of Verilog HDL code and in this version; the frequency divider circuit has been eliminated. Here the changes in *BCD* output and *signal* output with respect to time and change at *adcin* has been observed.

The behavioral simulation of the designed "Automatic Device Controller based on Temperature Sensing" was found with desired

accurate result according to the design specifications. It has been seen from the simulated *test bench* result, as in Figure 3 that, the *BCD* output represents the temperature according to the design specification and *signal* outputs were also compared properly with the pre-set values as well as changed when required. The whole process continues and the outputs were changed with respect to *adcin* and *clock.*

The design has been configured onto a SPARTAN-II FPGA for experimental verification. The FPGA has been programmed to initiate the ADC0820 to read and refresh temperature reading and then receive their responses and make necessary decisions to produce signals to control devices as well as viewing current temperature. The output signal lines of the system have been assigned to six LEDs on the FPGA board for better understanding of changes occurred in them and it was found that they were changed properly with respect to temperature. The four digit 7-segment display of FPGA board shows the current temperature in Fahrenheit scale.

The "Automatic Device Controller based on Temperature Sensing" has been designed to compare a 8 bit value that represent temperature value between 0°F to 330°F and compared it with pre-assigned or configurable value, when the temperature exceeded that preset values, signals are generated to control external devices with respect to current temperature. The design has acceptable resolution in this regard; however resolution can be extended by using high resolution ADC or by reducing measurement range.

The response time of the proposed approach is considerable faster since the device is capable to sense and process the temperature changes with a minimum delay as found mathematically $1/(50M*(1/2^9)) \cong 10$ micro second.

Finally it can be summarized that the special features of the proposed "Automatic Device Controller based on Temperature Sensing" are as follows:

(a) The ranges of temperature boundaries are reconfigurable.
(b) Faster response time.
(c) Cost effective.
(d) Upgradable.

8. CONCLUSION

The designed "Automatic Device Controller based on Temperature Sensing" System has been coded in Verilog HDL, compiled, simulated, synthesized and implemented using the *Xilinx® ISE 7.1i VLSI Development Suite.* The controller has been implemented onto a SPARTAN-II FPGA using a *Digilent™ Pegasus Spartan-II* FPGA board. The behavioral simulation of the designed system confirmed its perfect functionality. Therefore, it can be concluded that this type of reconfigurable temperature controller is easily synthesizable with the ISE Development Suite

and the Spartan-II GPGA can be configured for such controller designs. Such compact and reconfigurable automatic controlling systems can be very helpful in different areas of research and industrial applications since it provided very high speed data communication in both directions with acceptable degree of accuracy and temperature in a remote place can be easily controlled. This type of controller can be used efficiently in electronic devices or in industries.

9. FUTURE SCOPE

The designed "Automatic Device Controller based on Temperature Sensing" can also be used to produce to control temperature of multiple places by using multiple LM34 and ADC0820 ICs, multiplexing them with the controller and with certain change in design specification. With some modification, the design can also be applied for produce digital volt-meter.

This work can be extended to control the levels of reservoir including water or oil tanks, where pump or supply line of the tank or reservoir will be automatically turned on or off until a predefined level is reached.

The design can be further expanded to utilize a Look-Up Table (LUT) with a list of different temperature ranges, pre-set values for the controller can be selected automatically form this LUT.

The received data can be re-utilized to re-initiate some other process or processes for perhaps controlling other systems. However, in order to meet these requirements, the whole Verilog HDL coding should be revised.

REFERENCES

1. Principles of Control Systems. Available at: *http://www.tpub.com/content/doe/h1013v2/css/h1013v2_110.htm*
2. Automatic Control System. Available at: *http://www.tpub.com/content/doe/h1013v2/css/h1013v2_112.htm*
3. National Semiconductor, *LM34 Precision Fahrenheit Temperature Sensors. Available at: http://cache.national.com/ds/LM/LM34.pdf*
4. Philips Semiconductor, *ADC0820 8-Bit, high-speed, µP-compatible A/D converter. Available at: http://www.datasheetcatalog.org/datasheet/philips/ADC0820CNED.pdf*
5. Xilinx®ISE 7 in-depth Tutorial. *http://www.xilinx.com/support/techsup/tutorials/tutorials7*.htm
6. Spartan™–II FPGA Datasheet. *http://www.xilinx.com/support/documentation/spartan-ii.htm#19645*
7. FPGA4FUN, *How JTAG works, http://www.fpga4fun.com/JTAG2.html*
8. Digilent™ Pegasus Board Reference Manual. Available at: *http://www.digilentinc.com/Products/Docu*

mentation.cfm?Navl=Products&Nav2=Do cumenation

9. Palnitkar S. *Verilog HDL*. Pearson Education. 2007.
10. Bhasker J. Verilog HDL Synthesis. Star Gallery Publishing. 2005.
11. Components and Port Maps. Available at: h*ttp://www.doulos.com/knowhow/vhdl_des igners_guide/components_and_port_maps*
12. RTL coding. Available at: *http://www.doulos.com/knowhow/vhdl_des ingers_guide/rtl_coding*
13. Wikipedia, *Application-specific integrated circuit",http://en.wikipedia.org/wiki/Appli cation-specific_integrated_circuit*
14. Jain V. K. *Introduction to Switching Theory and Digital Electronics*, Khanna Publishers, 1987.