Safety Fuzzy Logic Controller of 1oo2 Architecture for FPGA Implementation

Bsiss Mohammed and Amami Benaissa,

Laboratory of computer science, systems and telecommunications (LIST)
Faculty of Science and Technology of Tangier,
University Abdelmalek Essaadi Tangier Morocco
90 000 BP 416 Tangier Morocco

Abstract
The motivation behind the implementation of a safety fuzzy logic controller (FLC) in VHDL was born not only out of the collective need for functional safety to be included in the hardware implementation of an inexpensive generic safety fuzzy logic controller for industrial and commercial applications, but also because the use of simple architecture for FLCs in industrial settings, such as the automotive sector, has led to serious accidents.
In recent years, more advanced FPGAs have seen greater use in embedded systems design as; they make the design easier and are more reliable as they can quickly provide a complex digital system prototype leading ultimately to a faster production of boards. Certainly there are now more advantages in using FPGAs, not only because the number of gates and features has increased and allowing a system-on-chip (SoC) on a single device to be finally realized, but also leading FPGA vendors are offering easy-to-use development tools that accelerate time-to-market and allow not only increased design productivity but also a reduction in the cost of development.
Nevertheless, these embedded blocks must be configured, verified, validated and properly connected to the rest of the system; however, this is safe only if the implemented design in FPGA is safe.
The challenge is how to ensure that the contents of the FPGA are functioning properly.
This paper presents an experimental simulation of a functional safety fuzzy logic controller with 1oo2 [1] architecture.

Key words:
Safety fuzzy logic controller, XooY architecture, safety norm 61508, VHDL, field programmable gate array.

1. Introduction

The internal structure of FPGAs is composed of arrays of configurable logic blocks (CLBs) along with interconnection channels and input/output blocks (IOBs); however, it remains a difficult task to control and observe the internal nodes of the FPGAs due to their complex design. Notice that the Very Large-Scale Integration VLSI technology and the rapid developments in packaging technology have greatly increased the density of circuits used in FPGAs; which means the testing of systems is more complicated.
Different FPGA testing strategies can be found in the literature.
The first strategy discussed in [1] is based on creating several application circuits and testing them with test vectors developed specifically for each circuit. The second strategy is based on testing the internal structure and reconfigurability of an FPGA and is called the Multi-Configuration Strategy (MCS) [2]. The third strategy [3] is based on the concept of Built-In Self-Test (BIST).It offers the possibility to simultaneously test several FPGAs during fabrication, by using internal BIST circuits. Generally the three strategies are used by unprogrammed FPGAs to detect the following failures: bridging fault, stuck-at fault, interconnect defect, CLB defect and LUTs defect; however, it is impossible to detect delay faults caused by interconnection.
But how can we detect this failure using programmed FPGAs? Or, in other words, how can we be sure that the generated code for the simple fuzzy logic controller architecture [4], [5] is correctly operated on the device? Under these circumstances, a simple-structured for FLC-system without redundancy does not provide reliability and the safety is only partial. Particularly with regard to safety-related systems, model structures are necessary in order to allow safe operation in case of system failure.
This security must meet the requirements defined in security Norm IEC61508 [6].

2. Safety fuzzy logic controller

The convention XooY characterizes the architecture of a system (sensor, actor, controller, processor module, input/output module…etc.) for which it is necessary that X (on account of the system having X channels) channels

[1] 1oo2 is read 1 out of 2. It describes the number of available channels for safety.

Manuscript received April 5, 2011
Manuscript revised April 20, 2011

operate correctly so that the safety function can be executed.
The following architectures are mentioned in literatures [7] and [8]:

1oo1, one-out-of-one
1oo2, one-out-of -two
1oo3, one-out-of- three
1oo4, one-out-of-four
2oo2, two-out-of- two
2oo3, two-out of -three and
2oo4, two-out- of- four

The most used architecture for safety systems is the one-out-of-two architecture (1oo2), which means that in these particular cases, the system must operate at least, so that the security function can react when an error occurs, and so brings the system to a condition of safety. This system may represent different components such as a sensor, a processor and in other cases a fuzzy logic controller.
The reason for this architecture being chosen is the value of probability of failure on demand (PFD). Here we must mention the concept of the safety integrity level "SIL" in order for the term "PFD" to be understood. The safety Norm 61508 defines the SIL as a "discrete level (one out of a possible four)" for specifying the safety integrity requirements of the safety functions to be allocated to the E/E/PE safety-related systems, where safety integrity level 4 has the highest level of safety integrity, and safety integrity level 1 the lowest [CEI IEC 61508-4 First edition 1998-12].
The four levels are defined based on the required risk-reduction factor (RRF) or probability of failure on demand (PFD) and are defined in IEC 61508 [10] as follows:

Table 1: Safety Integrity Levels

Safety integrity Level (SIL)	***Probability of failure on demand (PFD)***	Risk reduction factor (RRF)
1	10^{-1}-10^{-2}	10-100
2	10^{-2}-10^{-3}	100-1000
3	10^{-3}-10^{-4}	1000-10000
4	10^{-4}-10^{-5}	10000-1000000

The security and safety parameters of operation in 1oo2 architecture are preferred over those in the 2oo2 system, was privileged the disponibility, because the mathematical formula of the PDFavg for 1oo2 architecture has the form A^2 (the proof test coverage factor and the common cause factor beta are excluded) [9].

$$PDF_{avg} = 1/3x\ (\lambda_{Du}xT)^2$$

And the mathematical formula of the PDFavg by 2oo2 architecture has the form 2*A (the proof test coverage factor and the common cause factor beta are excluded) [9].

$$PDF_{avg} = \lambda_{Du}xT$$

Figure 1 shows a basic model for a safety fuzzy logic controller with redundancy architecture (1oo2).

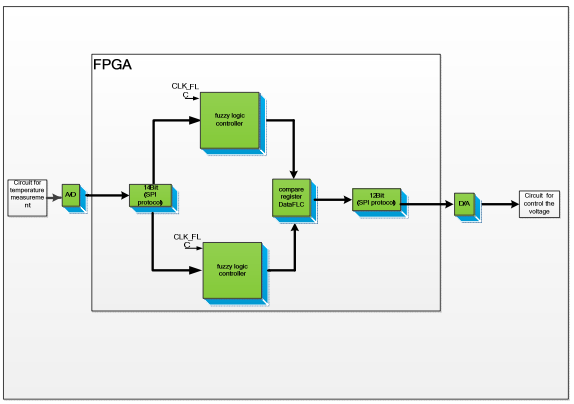

Figure 1 a basic model for safety fuzzy logic controller with redundancy architecture

As shown Figure 1 the safety fuzzy logic controller for controlling the output voltage depends on the temperature input and consists of the following components:

- Redundancy fuzzy logic controller architecture.
- Compare register DataFLC.
- 14-bit analog-to-digital converter LTC6912-1 from Linear Technology controlled by the FPGA (SPI)[11]
- 12-bit digital-to-analog converter LTC2624 from Linear Technology controlled by the FPGA[12]

2.1 Fuzzy logic controller one-out-of-one architecture

The simple fuzzy logic controller has one-out-of-one architecture (1oo1) consisting of the fuzzification process, the rule evaluation process and the defuzzification process. The details of this process can be found in [10].

2.1.1 Fuzzification

There are two variables for controlling the input in order to adjust the temperature:
The error of temperature Te and the derivation dTe/dt.
Figure 2 below shows the membership degrees of error of temperature Te with 4 fuzzy sets. It is partitioned into 4 zones known as: cold (-0 ° C to +20 ° C), cool (10 ° C to

+30 ° C), mild (20 ° C to +40 ° C) and hot (30 ° C to 120 ° C).

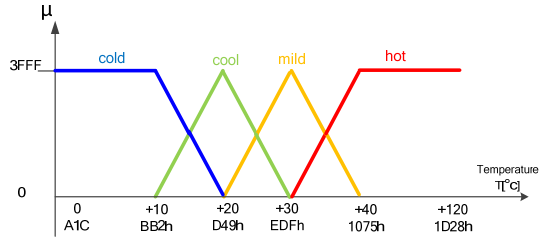

Figure 2 The representation of membership degrees of temperature with four fuzzy sets

Using a 14-bit computer resolution from [11], a membership degree μ = 1 is equal to 3FFFh or 16383 in decimal.

The second input variable for FLC is the rate of change of temperature dTe/dt.
The membership functions of the derivation dTe/dt takes three linguistic terms (slow, moderate and fast). The graphic representation of membership functions is shown in **Figure 3** .

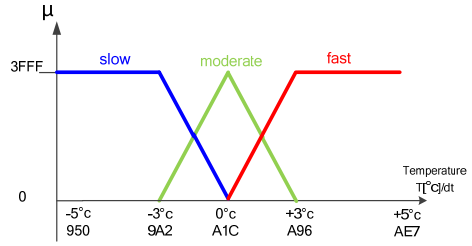

Figure 3 The representation of membership degrees of the rate of temperature change

As described in [4], two points and two slopes can be used to define the structure of a trapezoidal membership function.
The following pseudo code illustrates how this process is implemented in VHDL:

2.1.2 Rule inference engine

In this step, each input value is applied to its membership function in order to determine the value of the fuzzy input. In this application there are two inputs, one with four membership functions and the other with three, which makes seven degrees of membership functions to be calculated.
A value of specific inputs can cross two membership functions to create the degree of membership function for the corresponding input, so the share of several of them will be zero. As described above, two points and two slopes are used to define the structure of a trapezoidal membership function.

```
For i = 1 to n do begin
if input value < membership[i].point1 then ?i] = 0,
for i = 1 to n do begin
if DataIn < membership[i].point1 then
u[i] = 0,
else if DataIn < membership[i].point2 then
u[i] = tan g1 * (DataIn – p2) + 3FF
else
u[i] = 3FF – (DataIn – p2) * tan g2
endif;
endloop;
```

After that, the degrees of membership function are determined in a fuzzification step, the next: step is to use linguistic rules to decide what action should be taken in response to a set of data. The Mandani min-max technique [9] is used to calculate the numerical results of linguistic rules based on the input values of the system.
Before we commence calculating how many rules might be needed for the system, we must define the output membership functions.
The output membership functions of the fuzzy voltage control are 3 singletons. The linguistic terms of the singleton output (slow, moderate and fast) are used. The membership degree in each of the 3 classes can be declared in VHDL as a 12-bit [12] constant value as follows:

Figure 4 The representation of membership degrees of the output voltage with 3 fuzzy sets

The FLC has two inputs, one with four linguistic terms and the other with three and an output with three linguistic terms. This makes a total of 4*3*3 different rules that may be used to describe the strategy of total control.

2.1.3 Defuzzification

After the rules for each output have been established, the next step is to combine them into a single output value that can be used to control the output.
The center of gravity [10] will be used to obtain the release of the final system. In this application, the variables are used as a singleton output membership function. Defuzzification takes the weighted average of all fuzzy outputs. Each fuzzy output is multiplied by the corresponding singleton, and then the sum of these products divided by the sum of all fuzzy outputs for the final result.
The design of the fuzzy logic controller of 1oo1 architecture is shown in Figure 5

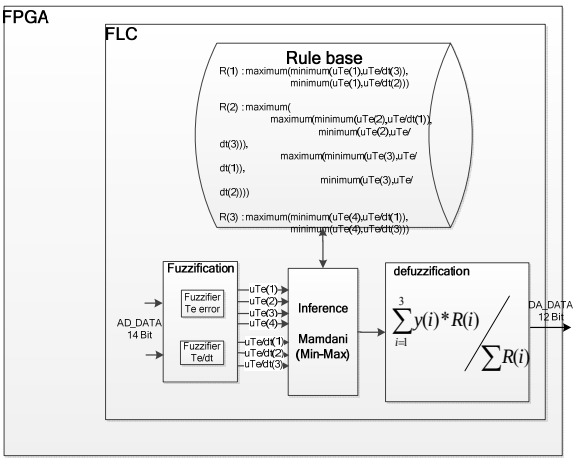

Figure 5 The design of the fuzzy logic controller of 1oo1 architecture

3. Redundancy fuzzy logic controller with the DataFLC compares register

Internally redundant architecture of FLC is realized by comparing data from the first fuzzy logic controller with that of the second FLC using the DataFLC compare register. The comparison takes place during each clock-cycle. The comparison of the faulty data is avoided by testing the system at each start with the TestFLC module. It consists of the DataTest register, ResultDataTest register, test data and test patterns.
The redundancy FLC with the TestFLC module allows not only the detection of several types of failures as bridging fault, stuck-at fault, interconnect defect, CLB defect and LUTs defect but also places the system in a safe state.
The design of the safety logic controller architecture is shown in **Figure 6**

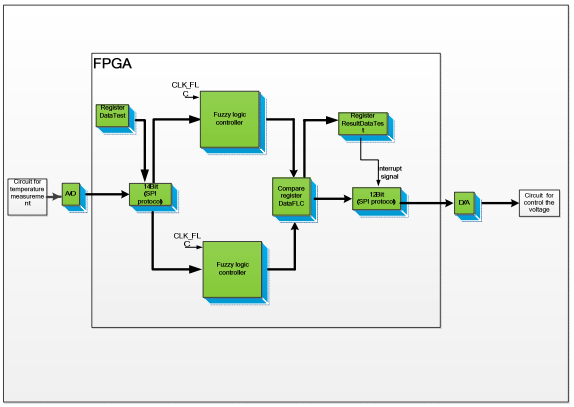

Figure 6 The design of safety fuzzy logic controller of 1oo2 architectures

4. Functional and Timing Simulations

After the synthesis process, meaning after converting the VHDL code into gate-level schematics, a test bench for redundancy FLC was designed using the Isim tools [13] from Xilinx. This allows the timing and the correct functionality of the system to be verified. Notice that the simulations are synthesisable. The system consists of many hardware components as shown in **Figure 7**

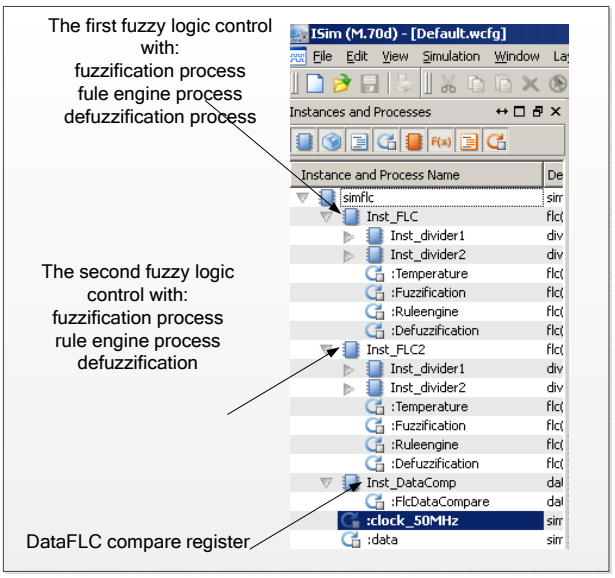

Figure 7 The hardware component for Safety FLC

A set of stimuli as inputs is fed into the test bench with the propagation delay taken into account.
A synthesisable simulation result (timing diagram) is seeing in the **Figure 8**.

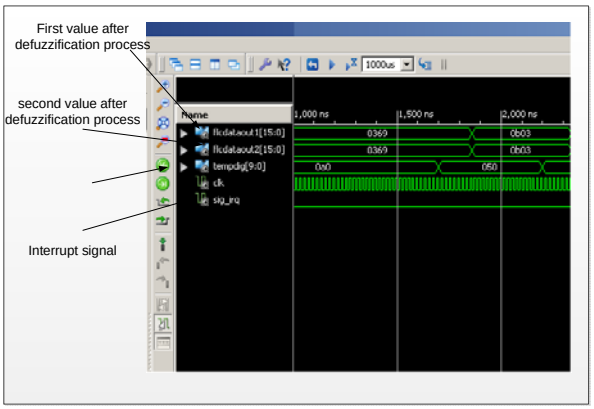

Figure 8 Waveform of functional simulation of the Safety FLC

The waveform in **Figure 8** shows the values of the temperature inputs and the corresponding output in hex form at the various instances determined by the stimuli in the test bench.

As seen in **Figure 8** the result of the first FLC is exactly the same as the result of the second. If the interrupt signal (sig_irg) switches to high, it means there is a discrepancy in the results and the system goes into the safe state. When the system goes into the safe state, it means that the output is switched off.

5. Conclusion

A safety fuzzy logic controller for controlling the output voltage depends on the temperature having been designed to industrial standards. The design has been realized in VHDL using Xilinx12.3.

The safety FLC operates at a frequency of 50 MHz and can very not only react very quickly to the output whenever an error occurs, it can also put the systems into a safe state.

The simulation with ISim 12.3 from Xilinx demonstrated complete, functionality while meeting all the initial system requirements.

6. Abbreviation

VHDL : Very High Speed integrated Circuit
FPGA : Field programmable Gate Array
SoC : system-on-chip
XooY : X-out-of-Y
CBL : configuration logic blocks
IOB : input/output block
VLSI : Very Large-Scale Integration
MCS : Multi Configuration Strategy
BIST : Built-In Self-Test
FLC : fuzzy logic controller
PDF : probability of failure on demand
SIL : safety integrity level
T : proof test period
E/E/PE : Electrical/Electronic/programmable Electronic
PDF_{avg} : Average probability of failure on demand
λ_{Du} : Rate of Dangerous undetected failures

References

[1] C.Jordan W.P Marnane,"Incoming inspection of FPGAs", Proc. European Test Conf pp371-377, 1993
[2] M.B Tahoori, E J. Mc.Cluskey, M.Renovell, P. Faure "A Multi-Configuration Strategy for Application Dependent Testing of FPGAs Proc. VLSI Test.,2004"
[3] C.Stroud, P Chen, S.Konala, and M.Abramovici, "using ILA testing for BIST in FPGAs" in proceedings of IEEE VLSI Test Symposium, 1995 pp.259-265
[4] Philip T.Vsuong,Asad M.Madni and Jim B. Vuong "VHDL Implementation for a Fuzzy Logic Controller", Bei technologies, inc.13100 telfair avenue, Sylmar, California 91342 USA,2006
[5] Zeyad assi Obaid, Nasri Sulaiman and M. N. Hamidon "FPGA-based Implementation of Digital Logic using Altera DE2 Board" in IJCSNS International Journal of Computer Science and Nezwork Security, VOL.9 No.8,July 2009
[6] A Summary of the IEC 61508 Standard for Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems , version 2.0, january2, 2006 http://www.exida.com/articles/iec61508_overview.pdf
[7] Josef Börcsök "Elektronische Sicherheitssysteme Hardwarekonzepte, Modelle und Berechnung" 2überarbeitete Auflage ISBN 978-3-7785-4021-3
[8] ISA- The instrumentation Systems, and Automation Society "Safety Instrumented Functions (SIF)-Safety Integrity Level (SIL) Evaluation Techniques Part 2: Determining the SIL of a SIF via Simplified Equations" ISA-TR84.00.02-2001-Part2
[9] Bonne Hoekstra "Reliability with respect to Safety Instrumented Systems" Yokogawa Global Safety Solutions Center Apeldoorn, The Netherlands October, 2004 published in Epigram Spring 2002,pp6
[10] Fuzzy sets. Information and Control. 1965;8:338-353
[11] Datasheet Analog to digital converter with 14-bit from Linear technology LTC6912 "Dual programmable gain Amplifiers with Serial Digital Interface". http://cds.linear.com/docs/Datasheet/6912fa.pdf
[12] Datasheet Digital to analog converter with 12 bit from lineartechnologyLTC2624. http://cds.linear.com/docs/Datasheet/2604fd.pdf
[13] http://www.xilinx.com/support/documentation/sw_manuals/xilinx11/plugin_ism.pdf

Bsiss Mohammed received the B.S. and M.S. degrees in Electrical Engineering from Kassel University Germany in 2006. He worked as a software developer in Mannheim Germany for Blood glucose meter. He does currently PhD at the Faculty of Science and Technology Tangier, Morocco. His interest is in the area of the implementation of safety controller in FPGA.

Amami Benaissa received PhD from Paris 6 University in 1992, Professor of Electrical Engineering at the Faculty of Science and Technology, Tangier, Morocco. Currently, Director of Laboratory of Computer science, Systems and Telecommunications