2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Flexible Decision Support System Using Dynamic Partial Reconfiguration Technology

Janos L. Grantner and Chinh K. Nguyen

***Abstract*—Intelligent decision support systems may be required to adapt to changes in their operating environment. FPGA-based hardware accelerators can be used to design embedded systems that are reprogrammable and reconfigurable. Partial reconfiguration allows that some regions of the FPGA be dynamically reconfigured while other tasks are still running in the remainder of the device. A hardware accelerator was developed for a fuzzy automaton-based decision support system in the context of testing eye-hand coordination skills for handicapped children. A Zynq-7000 FPGA platform was used for this research. Matlab simulations and a test bench environment were designed to verify the results of the hardware simulations and implementation.**

I. Introduction

THE core module of an intelligent decision support system can often be defined as an automaton, or state machine. If the decision support process is required to work with imprecise or uncertain information then a fuzzy automata model can be devised to deal with that environment.

In some applications, the state transition graph that describes the algorithm for decision-making should be modified under real-time conditions in order to cope with changes in its environment. It may mean that the current state machine should be replaced with a revised or partially new one while retaining some, or all of the existing functions and system inputs and outputs, respectively.

Platform FPGAs (FPGAs along with one or more processor cores on the same chip) provide a powerful computing platform to design hardware accelerators with the capability of dynamic partial reconfiguration [9] [10]. In one hand, the processor section of these System-on-Chip (SoC) devices can be used to implement some standard computer interface protocol or some human-machine interface (HMI) protocol and also to interact with the FPGA section. On the other hand, the FPGA section can implement the customized automata functions and utilize the dynamic partial reconfiguration capability. The Zynq-7000 FPGA by Xilinx [1] is one of the currently available devices to build a hardware accelerator for a flexible decision support system.

In order to investigate hardware architectures and algorithms for the implementation of such systems the results of a former research project on the testing of the skills of children with eye-hand coordination problems [2][3] was chosen as a starting point. In that research a customized HFB FSM [4] fuzzy automata model was used to implement the state transitions of the decision algorithm on the grounds of changes in the values of two fuzzy inputs. For this current research, an additional decision algorithm (also in the form of a state transition graph) was devised. A new hardware accelerator was developed such that it would switch from one decision algorithm to the other one by evaluating an input condition.

This paper is organized as follows: Section II provides a short description of the former research on eye-hand coordination assessment decision support system along with the added decision algorithm. Section III covers the key properties of the customized HFB FSM automaton. Section IV reports on the results of the dynamic partial reconfiguration implementation of the automaton to support the two decision algorithms. Simulation results are presented in Sections V. Conclusions and ideas for further research are given in Section VI.

II. Testing Skill Levels of Eye-Hand Coordination

The results of this former research project [2] were used as the point of departure to develop a hardware accelerator for the implementation of a flexible decision support system. In the eye-hand coordination testing research, the subjects were tasked to use a stylus of a haptic robot to travel from the beginning to the end of a labyrinth. Two parameters were measured by the attached computer workstation: the time the subject needed to move the stylus along the full length of the labyrinth and the accuracy of the stylus movement in terms of the number of occasions the stylus crossed the walls of the labyrinth. Based upon the performance of the subject in a particular test setup and an experimental knowledge base, an occupational therapy expert made a decision on whether the performance of the subject qualified him, or her to be presented with a more difficult test, or the next test should be a lower-level one, or the testing process should be stopped at that point. The objective of the research was to develop a decision support system in order to partially automate the decision making and also by using an agreed upon knowledge base make the decisions less subjective.

The block diagram of the Intelligent Decision Support System (IDSS) [3] is shown in Fig. 1. The age group was set by the operator. In addition, the operator decided whether or not the test process should be halted after a particular test. In the research reported in [2] only one age group was considered.

Janos L. Grantner is a Professor in the Department of Electrical and Computer Engineering at Western Michigan University, Kalamazoo, MI 49008, USA (phone: 269-276-3154; fax: 269-276-3151; e-mail: janos.grantner@wmich.edu).

Chinh K. Nguyen is a master's student in computer engineering in the same department (email: kychinh.nguyen@wmich.edu).

978-1-4799-2072-3/14/$31.00 ©2014 IEEE

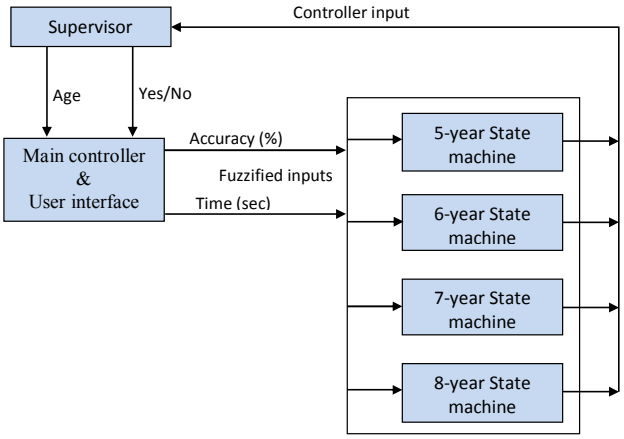

Figure 1. Block Diagram of the IDSS

The decision making process was modeled by using a fuzzy automaton. Being in a particular state determined the level of the test to be given to the subject. The two measured inputs were fuzzified.

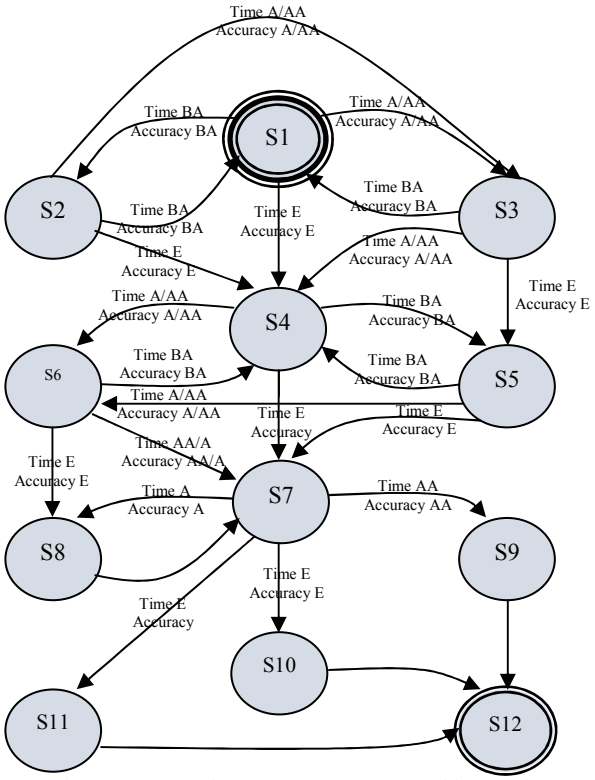

Figure 2. 12-state fuzzy automaton state transition graph

The conditions for the state transitions from the present state to the next one were devised on the grounds of the fuzzified outcomes of the current test. This approach led to a simplified instance of the HFB FSM fuzzy automaton model [4] with crisp states only and without inference operations. Inference operations were not included because making an explicit, qualitative judgment on the skills of the subjects was not part of the research objectives. Using a fuzzy automaton to support the decision making process was justified by the fact that the interpretation of the test results had inherent uncertainties.

The original 12-state automaton state transition graph that was developed for testing the eye-hand coordination skills of the subjects [2] is given in Fig. 2. The membership functions over the universal spaces for Accuracy and Time, respectively [2], are given in Fig. 3. Fig. 3 also includes the definition of the non-overlapping Boolean subintervals for the B algorithm that maps fuzzy variable input changes to changes in a set of Boolean variables that are used to trigger the crisp state transitions of the HFB FSM automata [4].

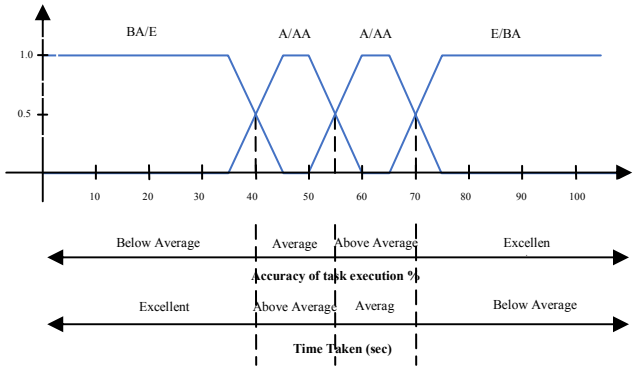

Figure 3. Universal sets for the Accuracy and Time fuzzy inputs

The state transition conditions for the 12-state fuzzy automaton are listed in Table I.

TABLE I
CONDITIONS FOR STATE TRANSITIONS (12-STATE FUZZY AUTOMATON)

Current State	Next State	Transition Condition
1	2	Time BA and/or Accuracy BA
1	3	Time A/AA and/or Accuracy A/AA
1	4	Time E and Accuracy E
2	1	Time BA and/or Accuracy BA
2	3	Time A/AA and/or Accuracy A/AA
2	4	Time E and Accuracy E
3	1	Time BA and/or Accuracy BA
3	4	Time A/AA and/or Accuracy A/AA
3	5	Time E and Accuracy E
4	5	Time BA and/or Accuracy BA
4	6	Time A/AA and/or Accuracy A/AA
4	7	Time E and Accuracy E
5	4	Time BA and/or Accuracy BA
5	6	Time A/AA and/or Accuracy A/AA
5	7	Time E and Accuracy E
6	4	Time BA and/or Accuracy BA
6	7	Time A/AA and/or Accuracy A/AA
6	8	Time E and Accuracy E
7	8	Time BA and/or Accuracy BA
7	9	Time A and/or Accuracy A
7	10	Time AA and/or Accuracy AA
7	11	Time E and Accuracy E
8	7	Time X and Accuracy X
9,10,11	12	Stop
	BA	**Below Average**
	A	**Average**
	AA	**Above Average**
	E	**Excellent**
	X	**Don't care**
	/	**Or**
	A and/or B	**A and B, A or B**

TABLE II
CONDITIONS FOR STATE TRANSITIONS (15-STATE FUZZY AUTOMATON)

Current State	Next State	Transition Condition
1	2	Time BA and/or Accuracy BA
1	3	Time A/AA and/or Accuracy A/AA
1	4	Time E and Accuracy E
2	1	Time BA and/or Accuracy BA
2	4	Time A/AA and/or Accuracy A/AA
2	5	Time E and Accuracy E
3	2	Time BA and/or Accuracy BA
3	4	Time A/AA and/or Accuracy A/AA
3	5	Time E and Accuracy E
4	3	Time BA and/or Accuracy BA
4	5	Time A/AA and/or Accuracy A/AA
4	6	Time E and Accuracy E
5	4	Time BA and/or Accuracy BA
5	6	Time A/AA and/or Accuracy A/AA
5	7	Time E and Accuracy E
6	4	Time BA and/or Accuracy BA
6	7	Time A/AA and/or Accuracy A/AA
6	8	Time E and Accuracy E
7	8	Time A and/or Accuracy A
7	9	Time AA and/or Accuracy AA
7	10	Time E and Accuracy E
8	7	Time BA and/or Accuracy BA
8	9	Time A/AA and/or Accuracy A/AA
8	10	Time E and Accuracy E
9	8	Time BA and/or Accuracy BA
9	11	Time A/AA and/or Accuracy A/AA
9	12	Time E and Accuracy E
10	9	Time BA and/or Accuracy BA
10	11	Time A/AA and/or Accuracy A/AA
10	12	Time E and Accuracy E
11	12	Time A/AA and/or Accuracy A/AA
11	13	Time E and Accuracy E
12	11	Time BA and/or Accuracy BA
12	13	Time A/AA and/or Accuracy A/AA
12	14	Time E and Accuracy E
13	10	Time BA and/or Accuracy BA
13	14	Time A/AA and/or Accuracy A/AA
13	15	Time E and Accuracy E
14	15	Stop
	BA	**Below Average**
	A	**Average**
	AA	**Above Average**
	E	**Excellent**
	X	**Don't care**
	/	**Or**
	A and/or B	**A and B, A or B**

The objective of the current research was to develop a hardware accelerator for a flexible version of the IDSS that can dynamically switch from one fuzzy automaton to the other one in a very short time by checking the status of an added input signal. This capability is not crucial for a system like testing the eye-hand coordination skill levels but it is in the case of controlling autonomous robots operating in an uncertain, real-time environment.

III. SIMPLIFIED HFB FSM FUZZY AUTOMATON

The extended Hybrid-Fuzzy-Boolean Finite State Machine (HFB FSM) was introduced in [4]. Its block diagram is shown in Fig. 5. X_F, W_B and X_A stand for fuzzy, two-valued (Boolean), and analog inputs with associated threshold values, respectively. X_T is the result of the comparison of the analog inputs with the associated threshold values. Z_F, Z_C and U_B stand for fuzzy, defuzzified fuzzy, and two-valued (Boolean) outputs, respectively. R* is the composite linguistic model, and ◦ is the operator of composition. A fuzzy state is made up of a set of crisp states, the HFB FSM stays simultaneously in each of them, to a certain degree that is represented by a state membership function. There is a dominant crisp state in this state set for which the degree of the state membership function is 1.

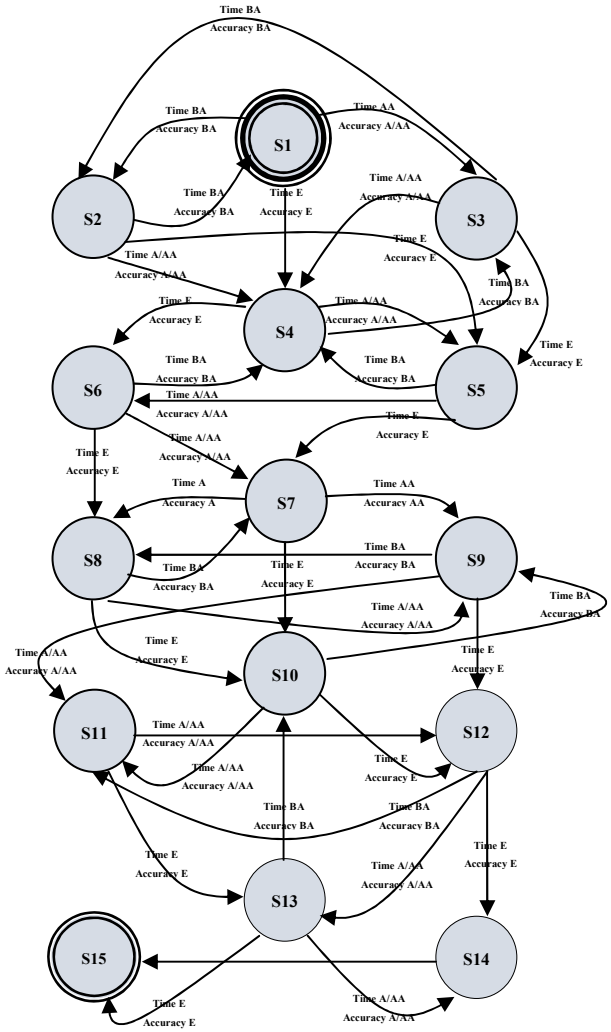

Figure 4. 15-state fuzzy automaton state transition graph

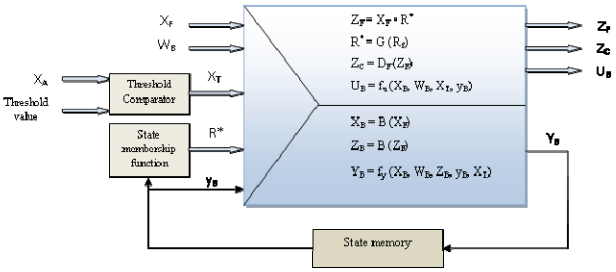

Figure 5. Block diagram of the HFB FSM fuzzy automaton

For this application a radically simplified version of the HFB FSM was sufficient enough. From the various input options, only two fuzzy inputs were retained. In general, for inference calculations an aggregated linguistic model is created dynamically. In this aggregated linguistic model, the state membership function values along with the states' linguistic models determine the inferred fuzzy output(s). Since there was no need for inference in this particular IDSS, each fuzzy state was reduced to its dominant crisp state, i.e., the state membership functions were dropped. Finally, out of the possible output options only a Boolean output vector was implemented. It displayed a "one-hot" code to identify the new present state of the fuzzy automaton for the Main

Controller.

One of the most critical theoretical problems for devising a fuzzy automata model is defining an algorithm for state transitions between the dominant states. With respect to the fuzzy inputs, the algorithm should provide a method to track the changes in the values of the input linguistic variables. A heuristic Fuzzy-to-Boolean algorithm (B Algorithm) was introduced in [5]. It maps the overlapping intervals of the linguistic values of a fuzzy variable to a suitable set of non-overlapping subintervals in the universe of discourse. The Mean of Maxima (MoM) defuzzification method is used to track the changes in the position of the center point of each fuzzy input value received. When the MoM value of the fuzzy input received falls into a specific non-overlapping subinterval the associated Boolean variable is set to 1 and the other variables making up the rest of the Boolean vector reset to 0. Hence, the changes in the value of a fuzzy input are mapped to changes in the contents of the corresponding Boolean vector. On these grounds two-valued logic design methods can be used to implement the state transition algorithm of the fuzzy automata.

It is critical for the performance of the hardware accelerator to devise a fast implementation of the B Algorithm. The approach is briefly characterized in Section IV.

IV. Hardware Accelerator for Flexible Decision Support System

The continuing advancement of Field Programmable Gate Array (FPGA) devices featuring reconfigurable logic functions, low power consumption, high performance of operation and large data storage capacity offers an attractive approach to develop and prototype hardware accelerators for fuzzy logic systems. In this research, a Xilinx Zynq-7000 FPGA-based Evaluation Board (Zedboard) [6] was chosen as the hardware platform. The Zynq-7000 FPGA supports dynamic partial reconfiguration. Partial Reconfiguration (PR) provides even more flexibility than that of offered by a generic FPGA. It allows the modification of an actively operating FPGA design by loading a partial configuration file, usually a partial bit stream file. After a full bit file has configured the FPGA, partial bit files can be downloaded in order to modify reconfigurable regions in the FPGA. It happens without interrupting the integrity of the applications running on those sections of the device that are not beixng reconfigured.

The main blocks of the hardware accelerator that are considered for dynamic partial reconfiguration are depicted in Fig. 6.

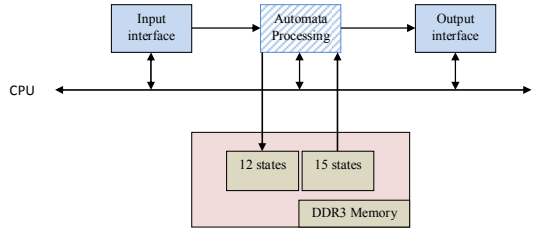

Figure 6. Partial reconfiguration design

The Automata State Transition module (either 12-state, or 15-state) was identified as a good target for partial reconfiguration. The reconfigurable partition refers to the physical location on the FPGA selected for partial reconfiguration while the remainder of the design and logic are referred to as static logic. The Automata State Transition module is a dynamic logic that is defined by its hardware interfaces and ports and is mapped onto the reconfigurable partition. A reconfigurable module in conjunction with the static logic is referred to as a configuration. The two dynamic configurations in this design are called the twelve states and fifteen states configuration, respectively. A configuration describes a complete FPGA design and produces a full bit stream for the reconfigurable module along with the static logic and a partial bit stream for only the reconfigurable module. All full bit streams and partial bit stream are stored in the DDR Memory during the boot process. These bit streams can be configured into the Programmable Logic section of the Zynq-7000 chip by the Xilinx development software.

The efficient hardware implementation of the Mean of Maxima (MoM) algorithm for evaluating the conditions for a possible state transition of the automaton was of particular interest. The approach taken is illustrated in Fig. 7

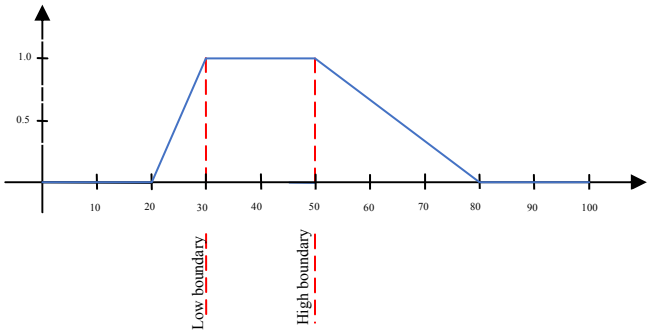

Figure 7. MoM algorithm

The hardware was tasked to find the low and the high boundaries of the plateau of the fuzzy input value and then calculate the position of the element at the middle point with respect to the discrete set of elements of the universal space. A parallel solution was implemented as shown in Fig. 8. The number of 4-bit comparators is equal to the number of elements in the universal space (100 for this research). Only one clock period was needed to find the two boundaries.

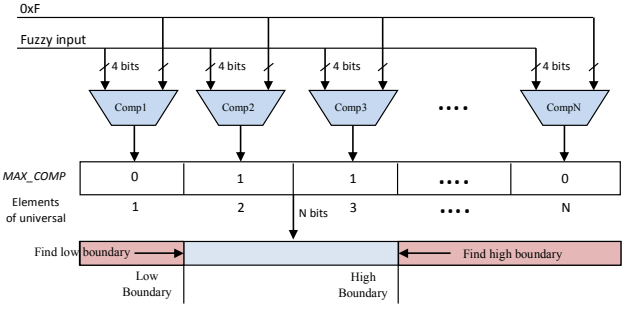

Figure 8. MoM comparator module

Table III below shows the sizes of the full and partial configuration bit streams for the 12-state automaton and the 15-state automaton, respectively. The configuration times are also given. The reconfiguration time depends on the size of configuration bit stream and the clock frequency of the processor and of the peripheral, respectively. For this project, the processor clock rate was 66.666 MHz and the processor configuration access port (PCAP) clock rate was 33.33 MHz.

TABLE III
BIT STREAM SIZES AND CONFIGURATION TIMES

Zynq board	Full Bitstream	Partial Bitstream
Bitstream size 12-states	4,045,564 Bytes	30,108 Bytes
Bitstream size 15-states	4,045,590 Bytes	30,211 Bytes
Configuration time 12 states with Standalone library	31104us	233us
Configuration time 15 states with Standalone library	31105us	233us

V. SIMULATION RESULTS

For verification, purposes both real-time hardware simulations using ISIM [7] and ChipScope [7] and functional simulations using Matlab were carried out. In addition, the implementation bit streams were downloaded to the Zynq-7000 FPGA on the Zedboard and the correct operations with respect to a number of fuzzy input sets were checked out. The same fuzzy input sets were used in all simulation sessions. The system worked according to its specifications.

Some hardware real-time (post-route) simulation waveforms are shown in Fig. 9. In order to make the monitoring easier, the signals were placed into five groups referred to as Fuzzified input, MoM Time, MoM Accuracy, and State Transition. A ChipScope simulation session is given in Fig. 10.

Matlab simulations are also carried out to double-check the results delivered by the hardware accelerator. Figs. 11-14 show the outcomes of the simulations for a sequence of fuzzy data sets, both for the 12-state and the 15-state fuzzy automata. The Matlab simulation results matched the hardware simulation results and the results obtained using the Zedboard as a prototype.

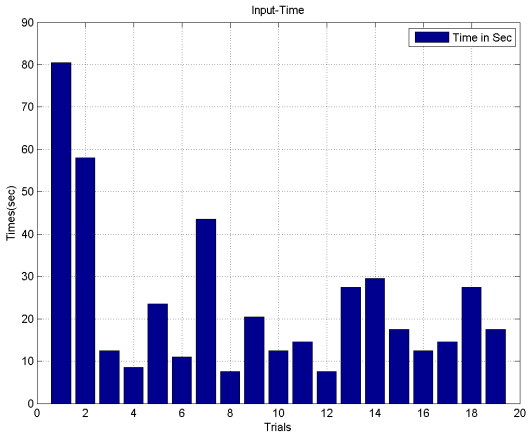

Figure 11. Defuzzified fuzzy input Time

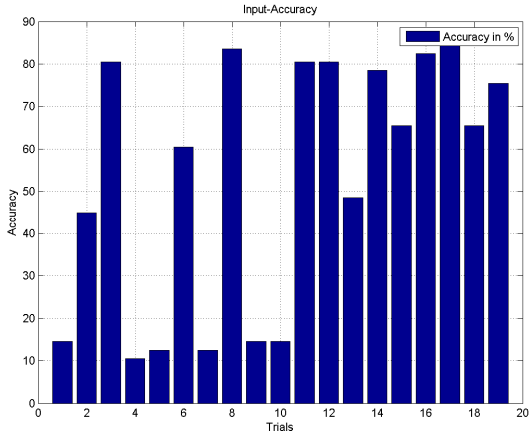

Figure 12. Defuzzified fuzzy input Accuracy

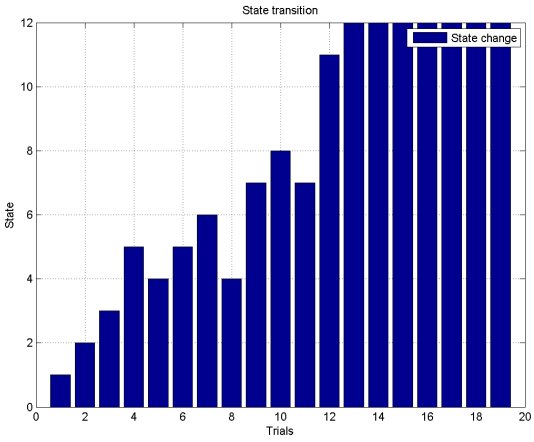

Figure 13. State transitions of the 12-state fuzzy automaton

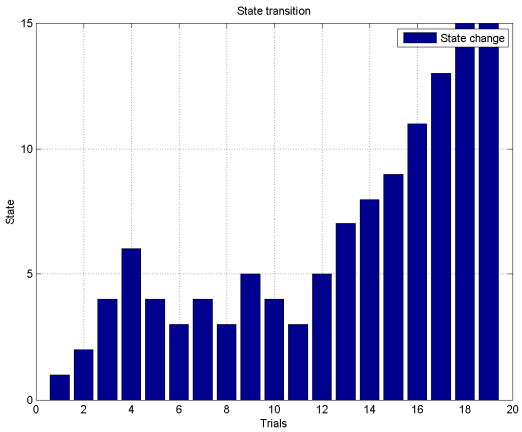

Figure 14. State transitions of the 15-state fuzzy automaton

VI. CONCLUSIONS AND FURTHER RESEARCH

The results of the hardware implementation of the key component of a flexible decision support system (namely, a fuzzy automaton) were presented. By using the Partial Reconfiguration function on the Zynq-7000 FPGA it was demonstrated that dynamic, real-time reconfiguration of a fuzzy automation is feasible. This feature represents a very attractive property to implement virtual fuzzy automata [8] for supervisory controllers of complex systems. This architecture

allows the supervisory controller to modify the fuzzy automaton in order to model a particular state cluster in the supervisory controller's state flow, as needed. Another advantageous property of the design is that it was controlled by embedded software running on the ARM-Core of the Zynq-7000.

In future work, the input data sets for the flexible decision support system will be received and the results will be sent from/to remote devices through a Gigabit Ethernet interface. In addition, reconfigurable inference and defuzzification engines will also be developed.

REFERENCES

[1] Xilinx, Zynq-7000 All Programmable SoC, Xilinx, March 2013.

[2] Janos L. Grantner, R. Gottipati, N. Pernalete, S. Edwards, G. A. Fodor, Intelligent Decision Support System for Eye-Hand Coordination Assessment, 2005 IEEE International Conference on Fuzzy Systems, May 22-25, 2005 Reno, Nevada, Proceedings on CD

[3] Janos L. Grantner, George A. Fodor, Ramakrishna Gottipati, Norali Pernalete and Sandra Edwards, Intelligent Decision Support System, IFSA 2007 World Congress, Cancun, Mexico, 18-21 June 2007, in Advances in Soft Computing, vol. 41, Analysis and Design of Intelligent Systems, Patricia Melin, Oscar Castillo, Eduardo Gomez Ramirez, Janusz Kacprzyk and Witold Pedrycz (Eds.), ISBN: 78-3-540-72431-5, pp 549-557.

[4] J. L. Grantner, G. A. Fodor, Fuzzy Automaton for Intelligent Hybrid Control Systems, 2002 World Congress on Computational Intelligence, WCCI'02/FUZZ-IEEE'02, May 12-17, 2002, Hilton Hawaiian Village Hotel, Honolulu, HI, in the CD Proceedings, ISBN: 0-7803-7281-6 J.L.

[5] Grantner, G. Fodor, D. Driankov, Hybrid Fuzzy-Boolean Automata for Ontological Controllers, 1998 World Congress on Computational Intelligence, FUZZ-IEEE'98, Anchorage, Alaska, May 4-9, 1998, Proceedings, Vol. I, pp. 400-404.

[6] Xilinx, Zedboard Schematic, Xilinx, Jan 2013.

[7] Xilinx, Zynq-7000 All Programmable SoC Software Developers Guide, Xilinx, Jun 2013.

[8] J. L. Grantner, G. A. Fodor, D. Driankov, The Virtual Fuzzy State Machine Approach - A Domain-Independent Fault Detection and Recovery Method for Object-based Control Systems, 18th International Conference of the North American Fuzzy Information Processing Society - NAFIPS'99, June 10-12, 1999, New York, NY, Proceedings, pp. 158-162.

[9] D. Kim, "An Implementation of Fuzzy Logic Controller on the Reconfigurable FPGA System," *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS,* p. 1, June 2000.

[10] C. Kohn, Partial Reconfiguration of a Hardware Accelerator on Zynq-7000 All Programmable SoC Devices, 2013.

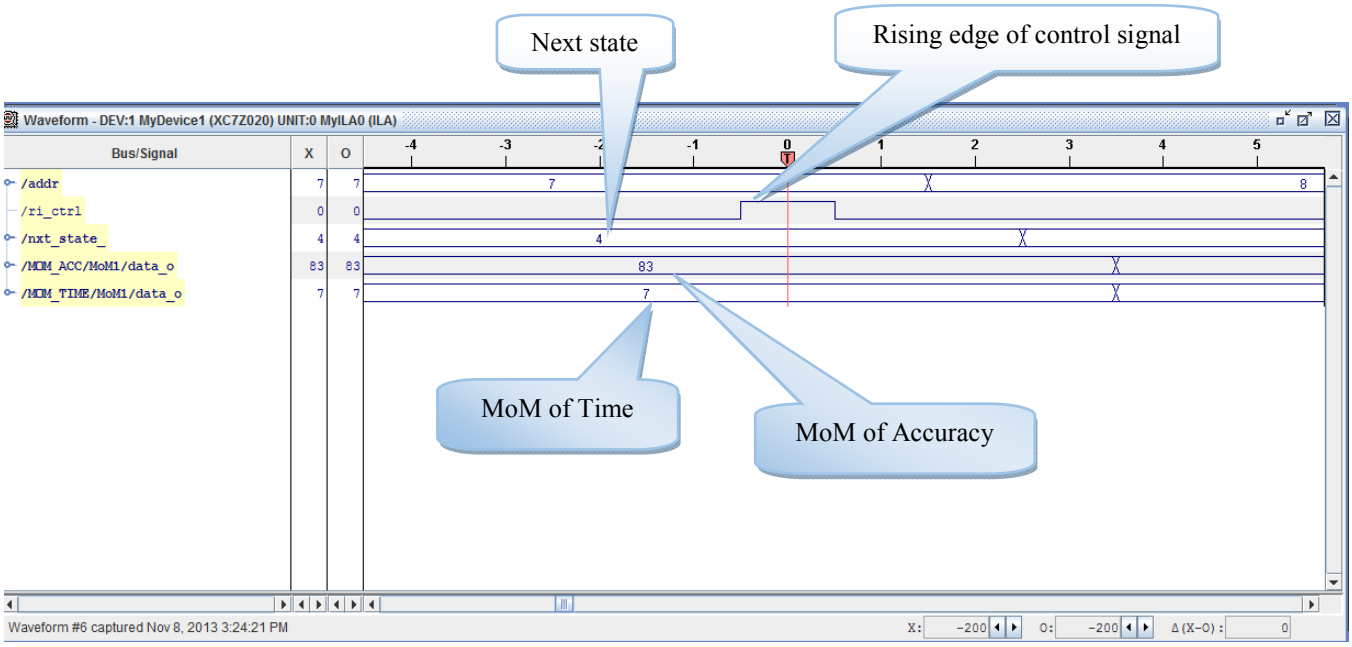

Figure 9. Real-time simulation using ISIM

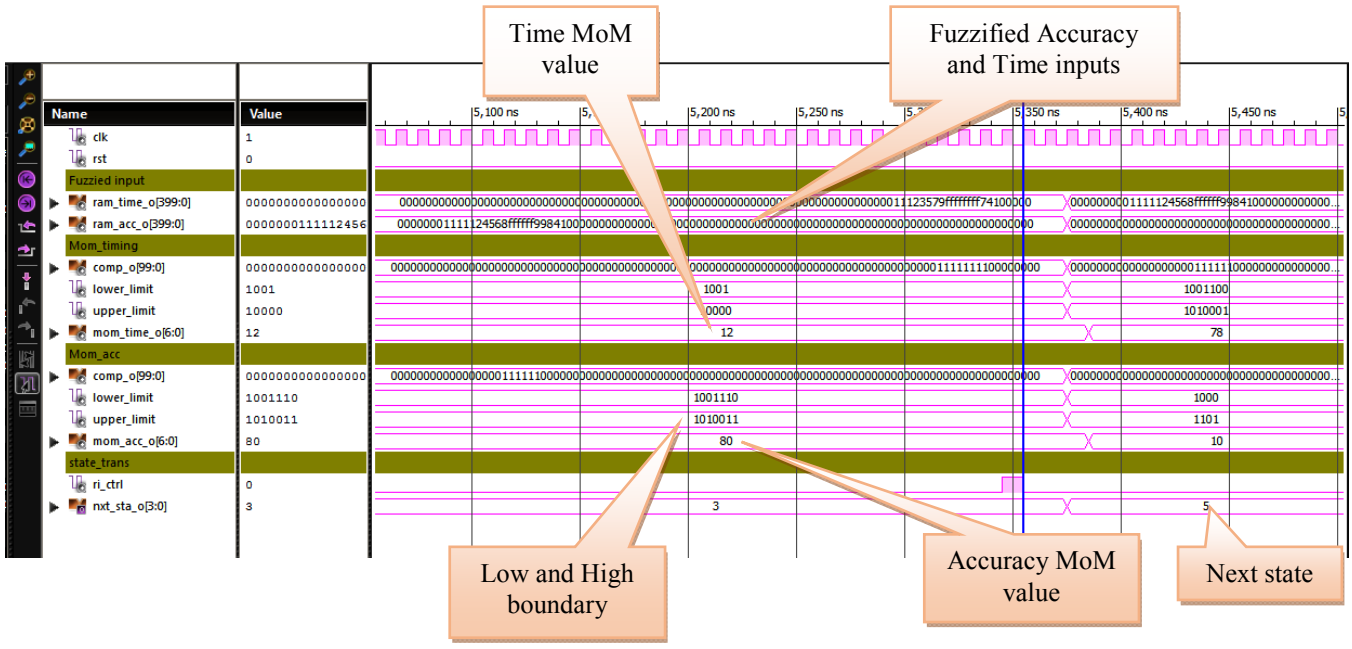

Figure 10. Real-time simulation using ChipScope