Design and FPGA Implementation of Takagi- Sugeno Fuzzy Controller Based on LUTs

Rasha Ilham Majeed
rasha_elham@yahoo.com

Abstract

In this paper, an approach for designing Fuzzy Controller based on Takagi-Sugeno inference engine with high computational speeds in architecture is proposed. The work focuses on advantages and disadvantages of Takagi-Sugeno control as compared with Mamdani's', in addition to focus on how computational complexity of the inference engine can be reduced and the speed of computation can be increased. Fuzzy Controller is implemented on FPGA using Look-Up Table (LUT). Whereas, each LUT is represented by Block RAMs in FPGA besides using number of arithmetic units in the design. To interface the design with users, a GUI program is designed using Visual Basic. Using JTAG port, the GUI's data can be stored in Block RAMs. Later, a designed Air Conditioning application is implemented and the practical results (in FPGA), theoretical results (computed by hand) and Matlab results are compared.

Keywords: LUT, FPGA, JTAG port, Takagi-Sugeno Fuzzy Controller, GUI.

التصميم والبناء باستخدام FPGA لمسيطر مضبب نوع تاكاي- سوكينو باعتماد جدول البحث

رشا إلهام مجيد
كلية هندسة الالكترونيات / قسم هندسة الحاسبات والمعلوماتية / جامعة الموصل

الملخص

في هذا البحث , تم تصميم مسيطر مضبب يتألف من آلة استنتاج نوع تاكاي- سوكينو ويعمل بسرعة حساب عالية . خلال العمل, تم التركيز على مميزات السيطرة المضببة من نوع تاكاي- سوكينو بالمقارنة مع السيطرة المضببة من نوع مامداني بالإضافة إلى التقليل من تعقيد معمارية التصميم للحصول على سرعة حساب عالية. حيث أنّ كل جدول بحث يُمثل (LUT)باستخدام مبدأ جدول البحث بالإضافة إلى استخدام عدة وحدات حسابية في التصميم. ولتمكين تعامل FPGAبقالب ذاكرة في رقاقة (فضلا عن إرسال بيانات GUIالمستخدم مع التصميم , تم تصميم واجهات المستخدم الرسومية . لتنفيذ واختبار JTAG باستخدام منفذ FPGAالمسيطر المضبب إلى قوالب الذاكرة في رقاقة المسيطر المصمم, تم تطبيق نظام التكييف الحراري في التصميم ومقارنة النتائج العملية مع النتائج المحسوبة نظريا فضلا عن النتائج المحسوبة باستخدام بيئة Matlab.

Received : 19 – 2 – 2009 **Excepted : 9 – 2 - 2010**

1- Introduction

Fuzzy Logic, which was developed in early 1960s, provided a tool to deal with uncertainty and human reasoning [1]. The main characteristic of a fuzzy logic is the capability to express the knowledge in a linguistic way. Fuzzy Logic empowers the digital designer with the ability to use non-linear controllers for their applications. Fuzzy Logic is used to control systems, whose mathematical model is unknown or time varying, or that requires human experience [1][2]. Fuzzy systems are also used for controlling high speed systems, and for processing of large volumes of data where enormous amount of computational speeds are required. High computational speeds may be achieved when these systems are realized on a digital hardware [2]. Computational speed of a digital hardware can be increased either by having parallelism or reducing computational complexity or both. Since a digital system works on a reference clock, the computational speed will also depend on the speed grade of the digital system[3]. Fuzzy Controller contains two main types: Mamdani and Takagi-Sugeno Fuzzy Controller [4]. Despite Mamdani Fuzzy Controllers are more famous and used than Takagi-Sugeno Type, but its digital hardware design is long and complex [2]. Whereas, Mamdani design consists of several parts of computing fuzzy outputs. These prosperities of Mamdani control are not working well in several types of applications [4]. On the other hand, Takagi-Sugeno design has many advantages as working well with linear techniques (as PID), having guaranteed continuity of the output surface, and the systems are computationally efficient [2][4]. To find the properties of Takagi-Sugeno design in implementation and application, an approach to design an architecture of Takagi-Sugeno Fuzzy controller is introduced and implemented on an FPGA kit. Then, a Sugeno GUI program to interface the design with the application of user is designed. The application is represented by an Air Conditioning System. The implementation of that application in Takagi-Sugeno is accomplished and its practical and theoretical outputs, are compared with the values obtained by Matlab. This paper is organized as follows: besides this introducing section, section 2 presents the design of a Takagi-Sugeno Fuzzy inference engine for an FPGA implementation. In section 3, the architecture of such controller is described. The design of Sugeno GUI is given in section 4. Section 5 contains the utilization of RAM blocks of FPGA in storing files. A Fuzzy Controller example on air conditioning system is designed in section 6. The implementation steps and results are illustrated in section 7. Finally, this paper is concluded in section 8.

2- Design of Takagi-Sugeno Fuzzy Inference Engine for an FPGA Implementation

Fuzzification is the process of converting the input crisp values into a set of membership values in the interval {0, 1} in the corresponding fuzzy sets. There are numerous different types of membership functions [5]. A finite collection of triangular fuzzy subsets are considered, which are unimodal, convex, and normal. The triangular and trapezoidal membership functions (triangular membership function being a special case of the trapezoidal function) are the most famous functions [2]. Sugeno or Takagi-Sugeno Inference Engine is introduced in 1985; it is similar to the Mamdani method in many respects. The first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same [2]. The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant. A typical rule in a Sugeno fuzzy model has the form

If Input 1 = *x* and Input 2 = *y* Then Output is *z* = *ax* + *by* + *c*

where a, b, and c are the constants of linear equation. For a zero-order Sugeno model, the output level z is a constant *(a=b =0)*. The output level z_i of each rule is weighted by the firing strength w_i of the rule. For example, for an AND rule with Input 1 = *x* and Input 2= *y*, the firing strength is

$$w_i = AndMethod\ (F_1(x), F_2(y))$$

where $F_{1,2}$ *(.)* are the membership functions for Inputs 1 and 2 as shown in Figure (1). The final output of the system is the weighted average of all rule outputs, computed as the easiest way to visualize first-order Sugeno systems is to think of each rule as defining the location of a moving singleton as given by

$$\text{Final Output} = \frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i} \qquad \text{------ (1)}$$

That is, the singleton output spikes can move around in a linear fashion in the output space, depending on what the input is.

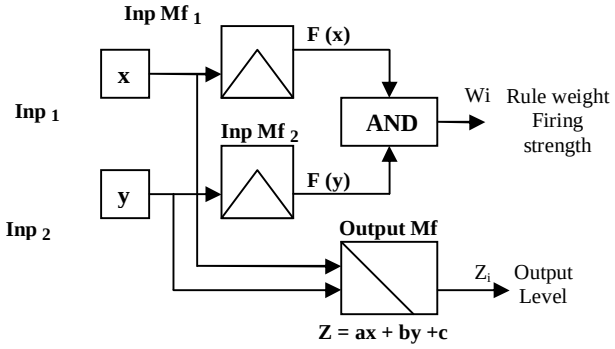


Figure (1): Sugeno Fuzzy Inference Engine
Inp_1 , Inp_2: $Input_1$ and $Input_2$ respectively
Inp MF_1,Inp MF_2, Output MF: Memberships of $Input_1$, $Input_2$, and Output respectively

3- Takagi-Sugeno Fuzzy Controller Architecture

The designed fuzzy inference engine consists of two inputs and one output. Takagi- Sugeno fuzzy controller consists of two main stages: fuzzification and inference engine. Fuzzification stage is designed using LUT1 and LUT2 components as shown in Figure (2), which are for input1 and input2 respectively. The inputs are taken from locations 200_h and 201_h in RAM2, respectively as 8-bit values to serve as addresses to the LUTs. The membership functions (MFS1, MFS2) for the first input stored in LUT1. The membership function (MFS3, MFS4) for second input is stored in LUT2. The data in the LUTs are the membership values. Each of LUT1 and LUT2 consists of two Block RAMs which are RAM1 and RAM2. RAM1 stores the values of MFS1, and MFS2 of the input, and RAM2 stores the values of L1 and L2, which are the locations of the input in its membership functions. The address of each RAM

depends on the value of the input. Figure (2) shows the block diagram of hardware implementation. Both inputs are processed in parallel. I.e., the outputs of LUT1 and LUT2 are available at the same clock pulse. The outputs of LUT1 and LUT2 are collected to a 20-bit data consisting of two possible membership values (8-bit each) and the two fuzzy sets (3-bit each). Inference Engine has two components. The Minimum block consists of four sub_blocks which are used to find the minimum degree which is the weight. For example, the first one is designed to find the minimum between MFS1 and MFS4 and so on. The second one is termed as 'LUT3' block. It represents four identical Block RAMs. They store the outputs of Sugeno linear equations which are computed using GUI program and, stored using JTAG port. L1, L2, L3, and L4 of fuzzification after merging serve as address to those four Block RAMs (merging is done in VHDL code of the architecture.

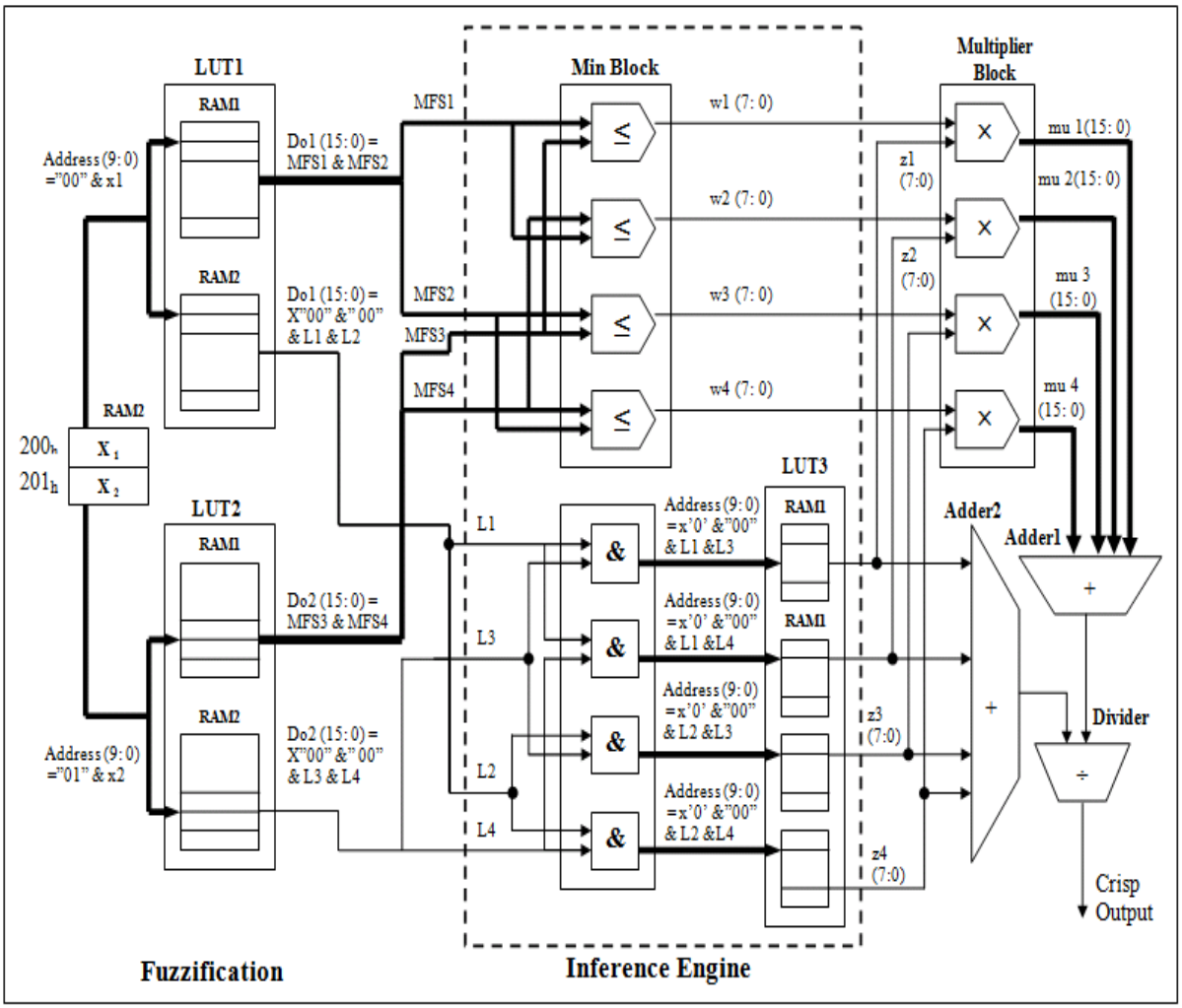


Figure (2):Takagi - Sugeno Fuzzy Controller Architecture

For example, 10- bit address of RAM_1 in LUT3= "10" & "00" & L1(5:3) & L3(2:0), whereas the total number of address bits equals 10 and the first two bits "10" represent the part of RAM1 which linear equations are stored in). LUT3 and Minimum block are implemented in parallel (at the same clock)..The next block is the multiplier block which is designed to multiply the weights (the outputs of minimum block) with the z_i (the outputs of LUT3) [6], then adding the outputs of multiplier block together using Adder1 block and adding the outputs of LUT3 together using Adder2 block in the same clock. Later, the first adding results a redivided by the second one using Divider block to find the crisp output of the controller [7]. Figure (3) shows the block diagram of the Takagi-Sugeno fuzzy controller

4- Design of Sugeno GUI

To interface Sugeno fuzzy controller design with the application of user, GUI program is designed using Visual Basic. This program is designed by three sequential forms as shown in Figure (4). The first form is used to enter the values of points of each membership functions as decimal values ranged from 0 to 255, then computing the values of degree1(MFS1 in Figure (2)) and degree2 (MFS2 in Figure (2)) for input1 depending on the equations shown in Form a of Figure (4). Computations of degree1, degree2, L1, and L2 for 256 values starting at input1=0 to input1=255 are done using a loop inside the program. The computed values are saved in File1 and File2 as hexa values. Then, the second form which is similar to the first one as shown in Form b of Figure (4) only it is used for input2. Later, the third form which is used to edit the Sugeno rules as shown in Form c of Figure (4). After completing the editing operation, all the edited rules would be saved at File2 as hexa values. Later, entering the crisp values input1 and input2 to save them in the last two locations of File2.

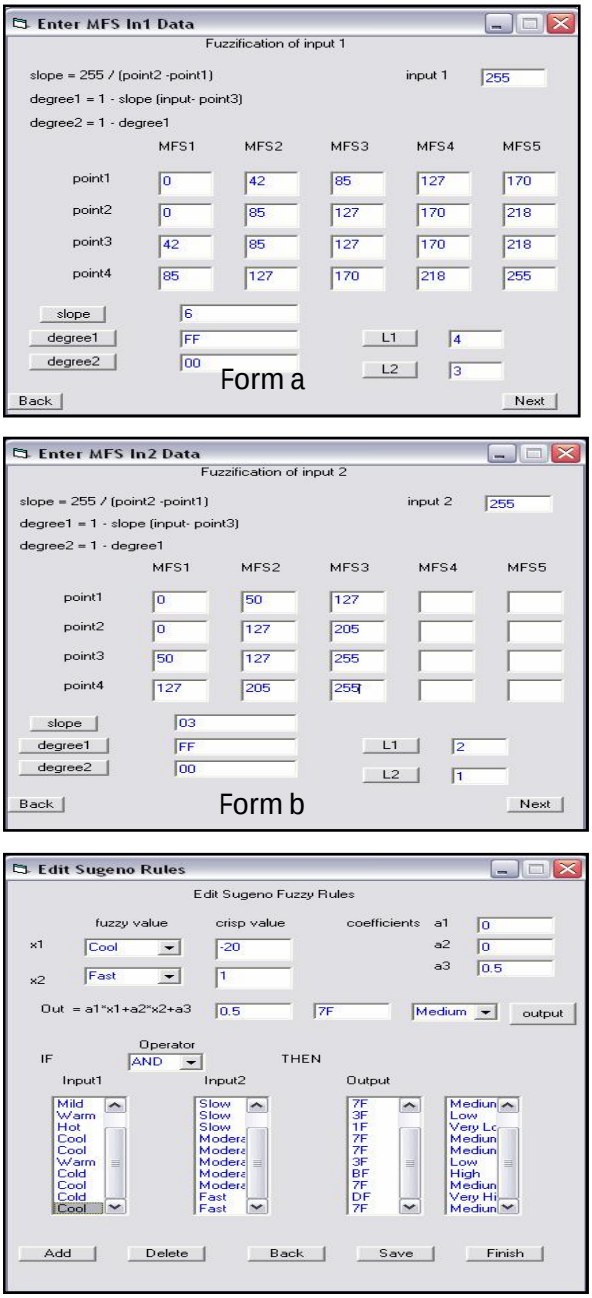


Figure (3): Block diagram of Takagi-Sugeno fuzzy controller

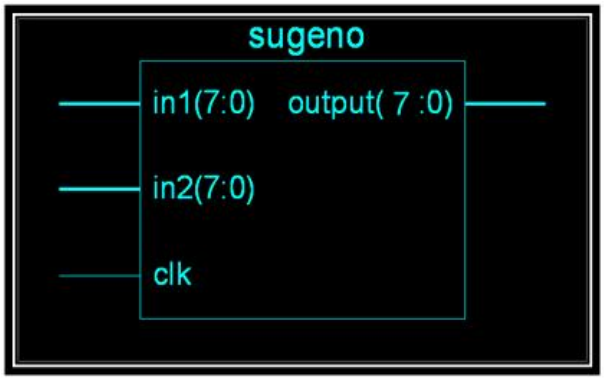


Figure (4): GUI of Takagi-Sugeno Controller

5- RAM Blocks

After saving the data from GUI program in File1 and File2, JTAG port are used to download these files into Block RAMs of FPGA [8], whereas Block RAM1 and Block RAM2 are used to store File1 and File2, respectively [9]. Each RAM consists of two main buses, Address and data bus [10]. Address bus is taken as 10-bit and data bus as 16-bit depending on type of Block RAM in FPGA. Whereas, RAMB16_S18 is used in the design. That means 1K × 16+ 2 Parity bits, (1K) its size is represented by 1024 locations, (address = 10 bits) , 16 its data bus is represented by 16 bits, and 2 is parity bits [11]. As shown in Figure (5), RAM1 is divided into three parts. The first one is for LUT1, contains 256 locations $(0\text{-}FF)_h$ to store the membership functions of input1

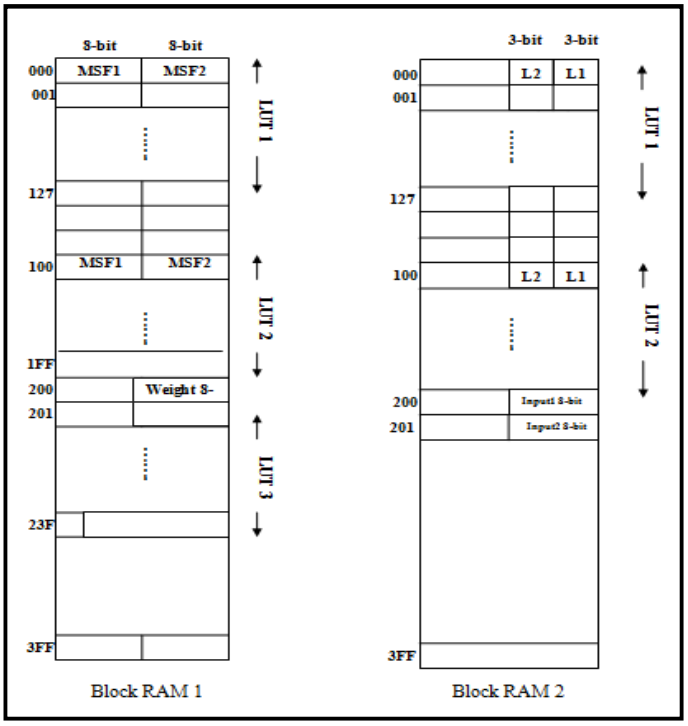


Figure (5): Configuration RAM 1 and RAM 2

The second one is for LUT2, contains 256 locations $(100\text{-}1FF_h)$ to store MFS of input2. Both of MFSs in inputs represents as 8-bit. The last part stores the data of Sugeno linear equations which is represented by 8-bit and other bits are don't care (equal zeros). RAM2 is divided into two parts which are similar in locations and in parallel with the first two parts in Block RAM1 except in data

because they are used to store data of labels for two inputs which are represented by 3-bit for each one. As shown in Figure (5), the values of input1 and input2 are stored in address (200_h) and (201_h) of RAM2, respectively.

6- A Fuzzy Controller Example

Fuzzy Controller Air Conditioning System is considered in this paper as an example. There are two control input variables to adjust the temperature of an air conditioning (AC) room[12]: the temperature error T_E, which is defined as the difference between the current temperature and the target set temperature as given by

$T_E = T_{set\text{-}point} - T_{preent}$ ----- **(2)**

and dT_E/dt, the rate of the temperature error change with time as given by

$dT_E/dt = [T_{En} - T_{En-1}]/[t_n - t_{n-1}]$ ------ **(3)**

where T_E = Temperature Error, t= time, n= integer.

The membership function for T_E consists of five fuzzy logic ranges that can be defined using the linguistic terms as *Cold*, *Cool*, *Warm*, *Mild*, and *Hot*. The graphical representation of the membership function of T_E is depicted in Figure (6). The values of the x-axis of the membership function have been scaled as an 8-bit hexa value representing the actual reading from the temperature sensor. Similarly, the y-axis for the grade of the membership function starts from 00h (false) to FFh (completely true).

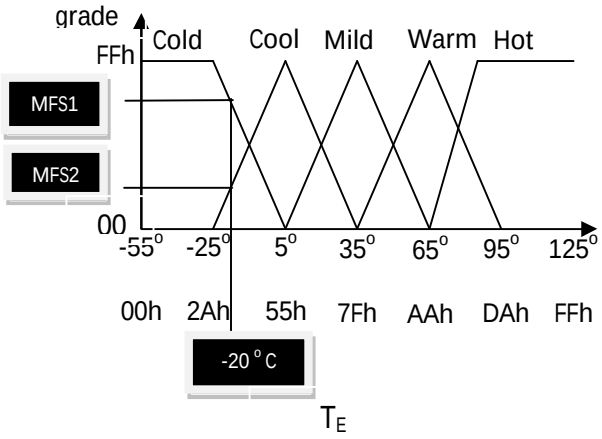


Figure (6): Membership function of Temperature Error (T_E)

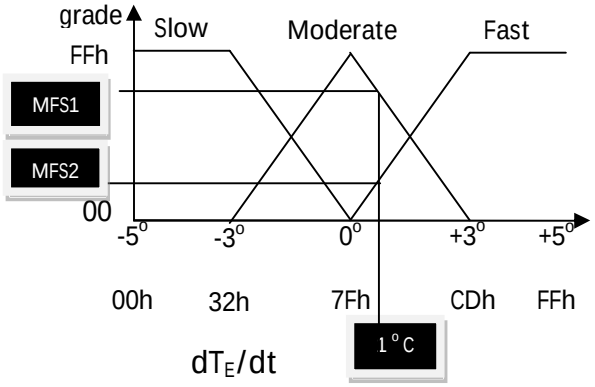


Figure (7): Membership function of Rate of Temperature Change (dT_E/dt)

The second input variable to the temperature fuzzy controller is the rate of change of temperature with respect to time. The calculation results in the rate of temperature change per time interval as defined by

$$\text{Rate} = \frac{\text{Present Temp – previous. Temp}}{\text{Time between Temp}} \quad \text{----- (4)}$$

The purpose of implementing the rate of temperature change in fuzzy control is to determine the velocity at which the process should react. Since time between temperature readings is triggered by every clock cycle in this application, the rate of temperature change can be simplified to become the difference between the current and previous temperature reading. As usual, the rate of temperature change takes three linguistic terms *Slow*, *Moderate*, and *Fast* as its membership function. The graphical representation of this membership function is shown in Figure (7).

The output membership functions of the fuzzy temperature control are singletons; five different linguistic terms of the singleton output *Very Low*, *Low*, *Medium*, *High*, and *Very High* are used to describe the heated control variations. Figure (8) shows the graphical representation of the singletons of these five variables. Table (1) shows the Takagi-Sugeno fuzzy rules of the Air Conditioning Systems.

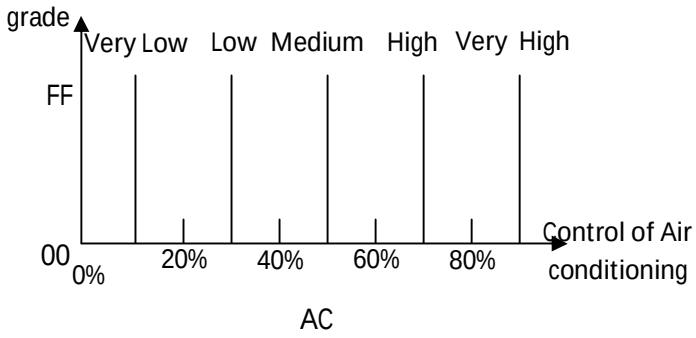


Figure (8): Membership function of Air Conditioning Control (AC)

7- Implementation Steps and Results

The implementation steps are:
1- Entering Membership points for each input using Visual Basic as shown in Figure (4).
2- Finding the Membership degrees for each input (Input1= -20^{o}C, Input2=1^{o}C are chosen to implement the Air Conditioning System [12]) are shown in Figures (6) and (7) respectively, and then storing the inputs values and its degrees in hexa files as shown in Figure (4). Table (2) shows computing the values of degree1 and degree2 theoretically.
3- Storing the Takagi-Sugeno Rules in File1. hex where as a1, a2=0 and c =constant represents points of Singleton Functions.
4- At the same form, the inputs are entered and stored in File2.hex to convert them to locations (200) and (201) in Block RAM2.
5- Convert the data of File1 and File2 to Block RAM1 and Block RAM2, respectively in FPGA using JTAG Port.
6- The inputs stored in Block RAM2 represent the address for Block RAM1 and Block RAM2 to find the degrees and labels for each input. This step deserves one clock pulse.
7- Compare the degrees for input1 and input2 to find the minimum degrees using Min Block and at the same clock the output values are found using LUT3 Block. This step spends one clock pulse because these blocks are operated in parallel. Table (3) shows the results of this step theoretically.
8- Then, Multiply Wi ×zi as shown in Table (4). This step spends one Clock pulse (Parallel). We use four parallel multipliers to find the values of this component.
9- Find the results of Adder1 and Adder2. This step spends one clock pulse (both of them are in parallel). Table (5) shows the theoretical results for both of them.
10- Divide $\sum \mathbf{mu_i} / \sum \mathbf{z_i}$. Figure (9) shows the practical final results.

Theoretical Output = B7 $_{h}$ Practical Output = B4 $_{h}$

Table (1): Takagi-Sugeno Rules

IF T_E AND dT_E/dt THEN				
Rule	**T_E**	**dT_E/dt**	**Ac Singleton**	**Ac Sugeno a1=0,a2=0,C= value**
1	Cold	Fast	Very High	90% - DF_H
2	**Cold**	Slow	High	70% - BF_H
3	Cool	Moderate	Medium	50% - $7F_H$
4	Mild	Slow	Medium	50% - $7F_H$
5	Warm	Slow	Low	30% - $3F_H$
6	Warm	Moderate	Low	30% - $3F_H$
7	Hot	Slow	Very low	10% - $1F_H$
8	Hot	Fast	Very low	10% - $1F_H$

Table (2): Degrees and Labels for Input1 and Input2

	Input ° C	**Input hex**	**MFS1 hex**	**MFS2 hex**	**L1**	**L2**
X1	**-20**	**3F**	**1E**	**E1**	**1**	**0**
X2	**1**	**81**	**06**	**F9**	**2**	**1**

Table (3): Results of Min Blocks and LUT3 blocks.

Min block	**LUT 3**
MFS1 min MFS3 W1=06	"10" & "00" L3 & L1= 211 , z1=(211)=7F
MFS1 min MFS4 W2=1E	"10" & "00" L4 & L1= 209 , z2 = (209)=7F
MFS2 min MFS3 W3=06	"10" & "00" L3 & L2= 210 , z3 =(210)=DF
MFS2 min MFS4 W4=E1	"10" & "00" L4 & L2= 208 , z4 = (208)=BF

Table (4): Results of Multiply Block

	Multiplier ($W_i \times z_i$)
1	mu_1= W1× z1 = 2FA
2	mu_2=W2 × z2 =EE2
3	mu_3=W3 × z3 =53A
4	mu_4=W4×z4 =A7DF

Table (5): Results of Adder1 and Adder2

Adder1 ($\sum mu_i$)	**Adder2 ($\sum z_i$)**
$BEE5_h$	$10B_h$

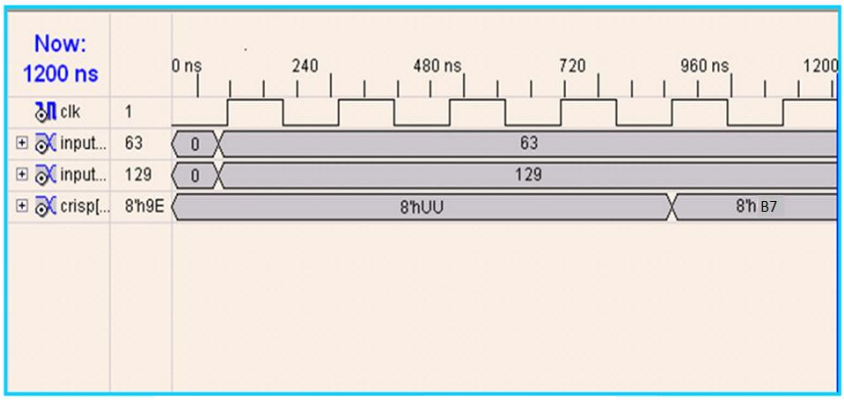


Figure (9): Practical result

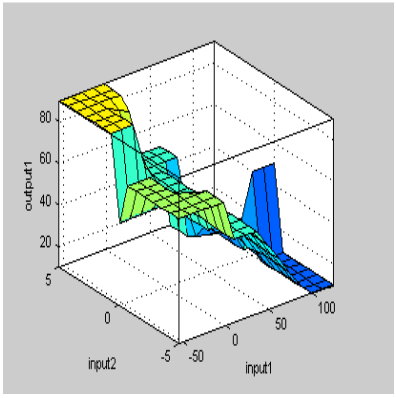


Figure (10): Matlab Surface for UOD of the Air Conditioning System

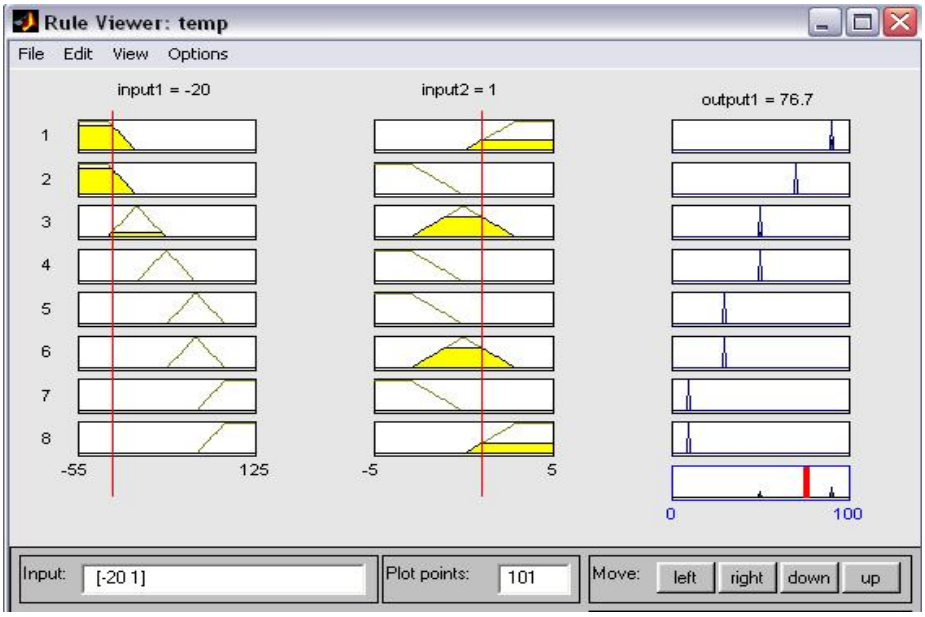


Figure (11): Matlab Result of Air Conditioning Application

8- Conclusions

In this paper, certain properties of Takagi-Sugeno Fuzzy Controller design have been concluded. From Table (7), we notice that the consumed area (number of slices) of implementing the design in Spartan 3E kit is 13% of total number of slices. That means the consumed area is very small with fast computations as result of reducing the computation and complexity of the design and using LUTs in Fuzzification stage instead of arithmetic units.

Table (7): The practical properties of designed controller

Number of Slices	635 out of 4656	13%
Number of IOs	29	
Number of BRAMs	8 out of 20	40%
Number of GCLKs	1 out of 24	4%

As comparing the results of Takagi-Sugeno design with Mamdani design' results, it can be concluded that the consumed area of Takagi-Sugeno design is less than Mamdani's as result of using additional units in Mamdani's system to find Fuzzy outputs as in Ref. [13]. Using LUTs in Takagi-Sugeno design in parallel have also reduce the complexity and computation of the design in addition to the fast execution of the system. Unfortunately, the use of LUTs may lead to some reduction in accuracy of implementation [14].

From this paper, It may also be concluded that each type of Fuzzy controller has special uses. Whereas, Takagi- Sugeno is working well with linear techniques (e.g., PID control), with optimization and adaptive techniques, and mathematical analysis, Mamdani on the other hand is well suitable for non linear control system.

To test the designed controller, Sugeno Fuzzy Controller is tested as a Proportional Fuzzy Logic Controller (PFLC) for a closed loop system. To compare the Sugeno operation with its Mamdani , the same controlled system (plant) has been taken from Ref. [3]. To implement that system, a closed loop system using LabView has been implemented and interfacing Sugeno FLC (on FPGA) with LabView System has been done via DAQ 6251 as shown in Figure (12). The implemented system has been divided into two parts: the first part is for supplying Step Function = 5 volt and finding the error and change of error of the system. The second represents the plant of the system. After implementing, comparing the characteristics of the system response in Figure (24) of Ref. [3] with the obtained system response in this paper shown in Figure (13) is done. From comparing, the characteristics of the system response in Ref. [3] with its characteristics in this paper, we notice that Mamdani Fuzzy Control is better than Sugeno Fuzzy Control in controlling on control system. Whereas, the value of overshoot in system response is higher than its value in Mamdani control. Table (8) shows the characteristics (Max. Overshoot, rise time, Settling time , Overshoot time) of that system based on Mamdani Fuzzy Controller and Sugeno Fuzzy Controller.

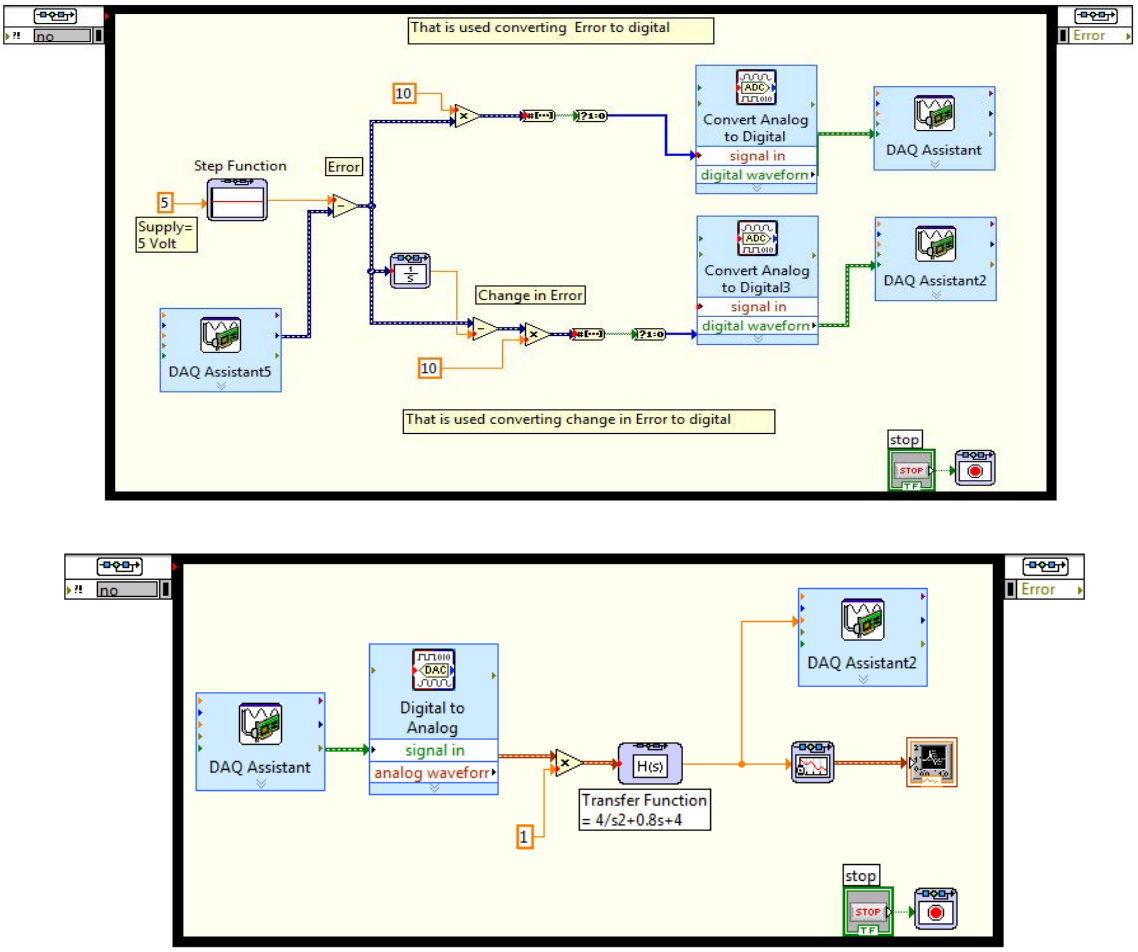
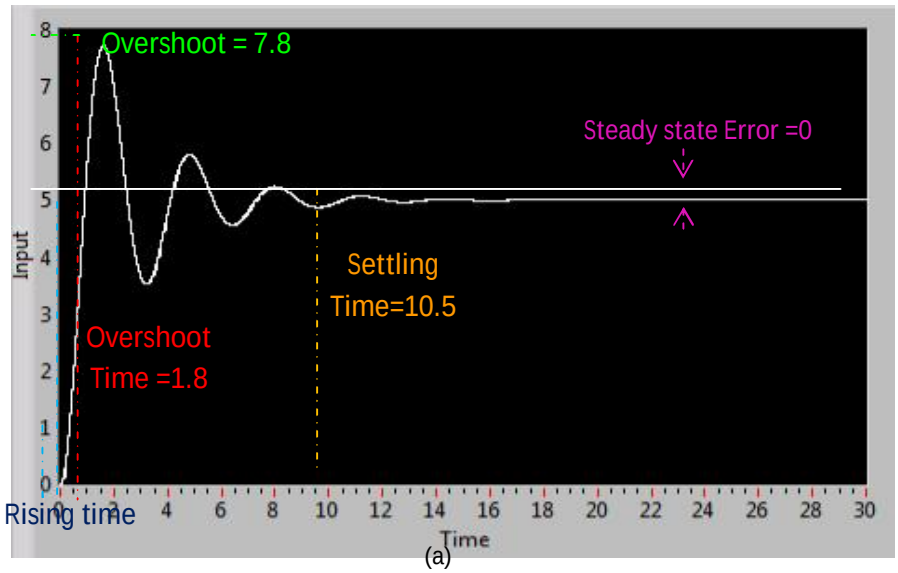


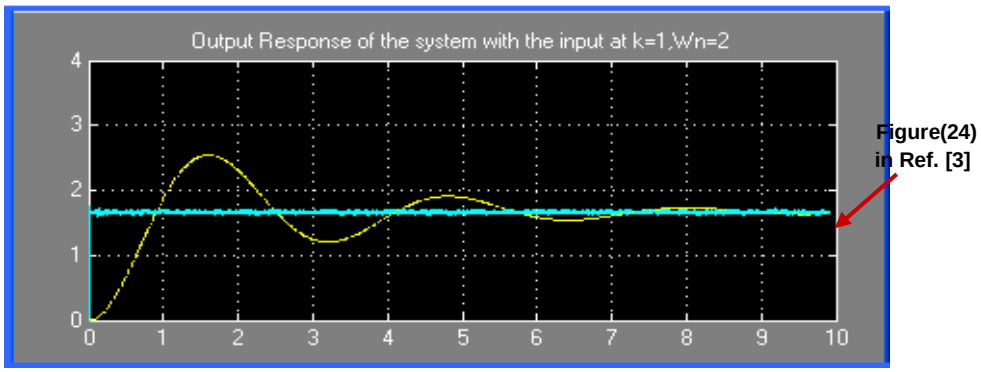
Figure (12): Block diagram of the Sugeno system using LabView

Table (8): Characteristics of System Response

System Characteristics	Overshoot	Overshoot Time (sec)	Settling Time (sec)	Rising Time
Theoretically (Sugeno)	5.876 When Vin= 5 V	1.8	10.2	0.64
(Sugeno)	7.8	1.9	10.5	0.6
Mamdani From Ref. [14]	2.5 When Vin=1.625	1.6	9.5	0.6



(a)



(b)

Figure (13): System Response at Wn=2, K=1, ζ=0.2

References

1. Dick, S.; Gaudet, V.; Huiqing Bai **"Bit-serial arithmetic: A novel approach to fuzzy hardware implementation"** Fuzzy Information Processing Society, 2008. NAFIPS 2008 Annual Meeting of the North American Volume, Issue, 19-22 May 2008 Page(s): 1 – 6

2. Jan Jantzen, **"Design of Fuzzy Controllers"**. Technical University of Denmark, Department of Automation, Bldg 326, DK-2800 Lyngby, Denmark. Tech. report no 98-E 864 (design), 19 Aug 1998.

3. Majed R., Mahmood B., **"DESIGN GENERAL FUZZY CONTROLLER IMPLEMENTING IN VHDL AND SYNTHESIZED USING FPGA"** 7th Int. ISC'2009, Loughborough, UK, pp. 262-268, June 1-3, 2009.

4. Poorani S. , Urmila Priya T.V.S., Kumar K.U. , Renganarayanan S. **"FPGA BASED FUZZY LOGIC CONTROLLER FOR ELECTRIC VEHICLE"**. Journal of the Institution of Engineers Singapore. Vol 45 Issue 5, 2005.

5. Yen and R. Langari, **"Fuzzy Logic Intelligence, Control, and Information"**, Singapore, Pearson Education, 2003.

6. " Multiplier v8.0"; www.xilinx.com; DS305 January 18, 2006. ©2006 Xilinx, Inc.

7. " Pipelined Divider V3.0 "; www.xilinx.com; DS305 April 28, 2005. Produced specification 1-800-255-7778. ©2004 Xilinx, Inc.
8. " JTAG Loader- Quick Guide-11". ©2004 Xilinx, Inc. Http://www.support.Xilinx.com/xlnx/xweb/xil-lx-home. jsp
9. Bridgeford B. And Cammon J. **"SVF and XSVF File Formats for Xilinx Devices".** XAPP503 (v1.0) April 17, 2002 . ©2002 Xilinx, Inc.
10. Chaplin K. **"Reconfiguring Block RAMs- Part1".** Http://www. Xilinx.com/xlnx/xweb/xil-lx-print friendly.jsp. Publishing Date: 07/17/2003. ©2003 Xilinx,Inc.
11. Chaplin K. **"Reconfiguring Block RAMs- Part2"** Http://www. Xilinx.com/xlnx/xweb/xil-lx-print friendly.jsp. Publishing Date: 07/17/2003. ©2003 Xilinx, Inc.
12. Philip T. Vuong, Asad M. Madnni, and Jim B. Vuong, **"VHDL Implementation For a Fuzzy Logic Controller".** Automatic Congress, 2006, WAC'06, World. Publication date: 24-26 July 2006. ISBN: 1-889336-33-9. Budapest
13. Masmoudi N., Hachicha M. , and Kamoun L.,**"Hardware design of programmable Fuzzy Controller on FPGA".** PP. 1675-1679; ISBN: 0-7803-5406-0, ©1999 IEEE.
14. S. Singh and K. S. Rattan, **"Implementation of a fuzzy logic controller on an FPGA using VHDL"** 22nd Int. Conf. of the North American Fuzzy Information Processing Society, NAFIPS 2003, pp. 110 - 115, July 24-26, 2003.

The work was carried out at the college of Engg. University of Mosul