A New Hardware Architecture for Fuzzy Logic System Acceleration

Aumalhuda Gani Abood
Electrical Engineering Department
University of Basrah
Basrah, Iraq
aumalhuda_comp@ yahoo.com

Dr. Mohammed A. Jodha
Computer Engineering Department
University of Basrah
Basrah, Iraq
m.a.al-ebadi@ieee.org

Dr. Majid A. Alwan
Computer Engineering Department
. University of Basrah
Basrah, Iraq
altimimee@yahoo.com

Abstract *In this work, a new architecture is designed for fuzzy logic system. The proposed architecture is implemented on field programed gate array (FPGA). The hardware designed fuzzy systemimproves the excution speed with very high speed up factor using low cost availble kits such as FPGA. The implementation of the proposed architecture uses very low amount of logic elements and logic array blocks as proven when implementing the proposed architucture on FPGA.*

Index Terms— FPGA, Fuzzy, Architecture, Hardware Accelerator.

I. INTRODUCTION

Today one of the successful technologies used for sophisticated control system development is fuzzy logic systems and fuzzy controllers [1]. Fuzzy controllers are more robust as compared to conventional controllers. It capable to cover a wider range of conditions in a given operating environment due to its computational efficiency. It is also more tolerate to noise and different types of disturbance.

In problem solving, different conclusions can be drawn from ambiguous and imprecise vague information. That is including many parameters of real world environment [2].

High speed real time constraints may be imposed on the implementation to realize the system. So that, software solution for a computer generated program may be not suitable for these cases [3]. On the other side, FPGA (Field Programmable Gate Array) which become recently a popular general purpose technology that can be dedicated for creating any digital system, especially that need high speed processing [2].

FPGA integrates a large amount of components requires digital logic. These elements are mapped in one chip [4]. Flexibility and reprogramming nature of FPGA make it is more efficient than Application Specific Integrated Circuits in case of design modification after first creation[2][4].

Depending on these features and FPGA facilities fuzzy logic system can be implemented more efficiently using the FPGA chips programed with VHDL language [5][6]. If the system is managed to use the facilities of parallelism and good FPGA architecture, the system performance and speed can be more enhanced [5].

In this work, anew hardware architecture is designed for general two inputs fuzzy logic system and implemented using FPGA chip (Altera cyclone III). The new architecture gives high speed execution with very high speed up factor.

Beyond this introductory section, the paper consists of five sections; section II describes fuzzy inference system. FPGA is described in section III. The proposed implementation is depicted in section IV. Section V shows the simulation results. Section VI concludes the acquired results.

II. FUZZY INFERENCE SYSTEM

Fuzzy logic system is generally consisting of three main parts: fuzzifier, rule base, and defuzzifier as depicted in Figure 1.

Part 1: Fuzzifier

This part responsible for converting crisp input in to a form can be handled by fuzzy rule. This is done using membership function, crisp input is fuzzified into a membership value in range [0,1] for each fuzzy set. This value represents the membership degree of the input for the specified

set. Exponential, trapezoidal, and triangular shaped functions are typically used as membership functions [6][7].

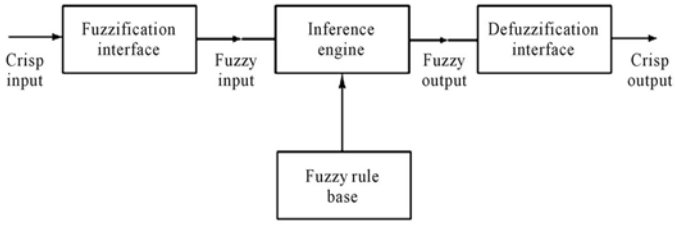

Fig. 1: Fuzzy Inference System

Part 2: Fuzzy Inference

Linguistic statements (so called as rules) form the knowledge base of the fuzzy system. Rules define the relation of fuzzy output in term of fuzzy input variables (output of fuzzifier stage). This stage can be considered as a mapping stage between fuzzy input domain to fuzzy set output domain [7].

Part 3: Defuzzifier

The output of the fuzzy inference (rules) is also fuzzy set and not a crisp value. This output need to be converted to crisp output to be used by the given plant. Defuzzifier stage performs this task using defuzzification methods. Center of gravity method average is the most famous way used for defuzzification [2][7]. The mathematical form of center of gravity is given in Equation 1.

$$u = \frac{\sum_{i=1}^{k} u_i * \mu(u_i)}{\sum_{i=1}^{k} \mu(u_i)} \quad \ldots (1)$$

Where:

u_i: is the i-th members of the output vector.
$\mu(u_i)$: is the multiplying coefficients of the output membership function.

III. FPGA

Field Programmable Gate Array FPGA is an integrated circuit(IC) consisting of an array of logic cells and interconnected blocks (programmable switches) as shown in Figure 2. Designer can program FPGA to act as a dedicated integrated circuit performs a desired operation. Large computation are broken into multiple logic element pieces and mapped into physical logic blocks in the array.

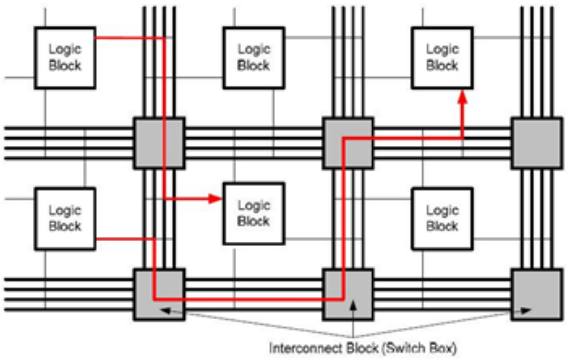

Fig. 2: FPGA Architecture

The interconnection is configured to route signal between logic blocks appropriately [3][8].
Normally FPGA consist of:

1. Programmable blocks: responsible for implementing required logic functions.
2. Programmable routing: connect the logic functions.
3. I/O blocks: represent external chip interface, routing interconnect I/O blocks to logic blocks.

Based on the background of computer architecture and digital logic, it is known that any function can be formed as a Boolean equation. And any Boolean equation, in its turn, can be expressed in a truth table format. In more specific expression; each hardware logic element can implement a truth table in control logic block (LUT). LUT is constructed as M-bit array of memory (SRAM) and M-input to 1-output multiplexer. This general Lookup table structure gives FPGA the generality in implementation for any digital logic [8].

IV. THE PROPOSED ARCHITECTURE FOR FUZZY INFERENCE SYSTEM

The proposed architecture implements fuzzy system with two input variables (error e and change of error ce) each of them in the universe of discourse [-1, 1]. The used membership function is chosen as triangular shape which is used for both inputs, each input has five membership functions: nb(negative big), ns(negative small),

z(zero), ps(positive small), pb(positive big), as shown in Figure 3.

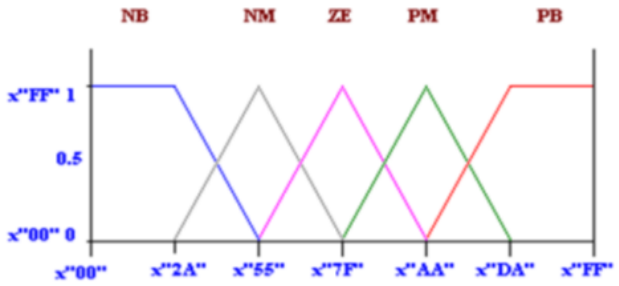

Fig. 3: The Input Membership Function

Real value input in range [-1,1] is mapped to hexadecimal numbers in the range ["00","FF"], such that "00H" represent -1 and "FFH" represent 1. Mapping process is done by Equation 2:

$$Y = \frac{2(xH)}{FF} - 1 \qquad \ldots (2)$$

Where
Y: represent the real value.
x: represent equivalent value.

Each inputs uses a triangular membership which is define by (point1, point2, point3) values as shown in Figure 4 and describe in Equation 3.

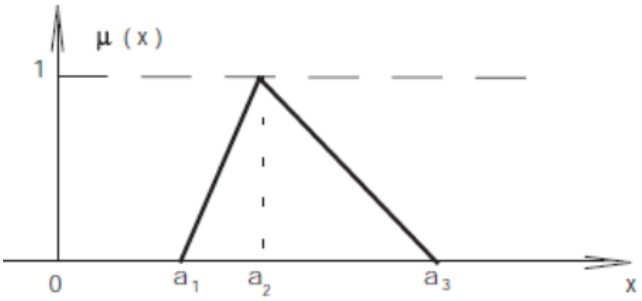

Fig. 4: The Triangular Membership

$$f(x) = \begin{cases} 0 & ,x < a_1 \\ (x - a_1)/(a_2 - a_1) & , a_1 \le x \le a_2 \\ (a_3 - x)/(a_3 - a_2) & , a_2 \le x \le a_3 \\ 0 & ,x > a_3 \end{cases} \ldots (3)$$

For the proposed design, the hardware architecture is shown in Figure 5, which has input1 represent error, input2 represent change of error coming from the plant, clock refer to the external clock signal which entered for all parts in design and reset represent the reset signal which reset the design when has value (1) but make the design work when has value(0).

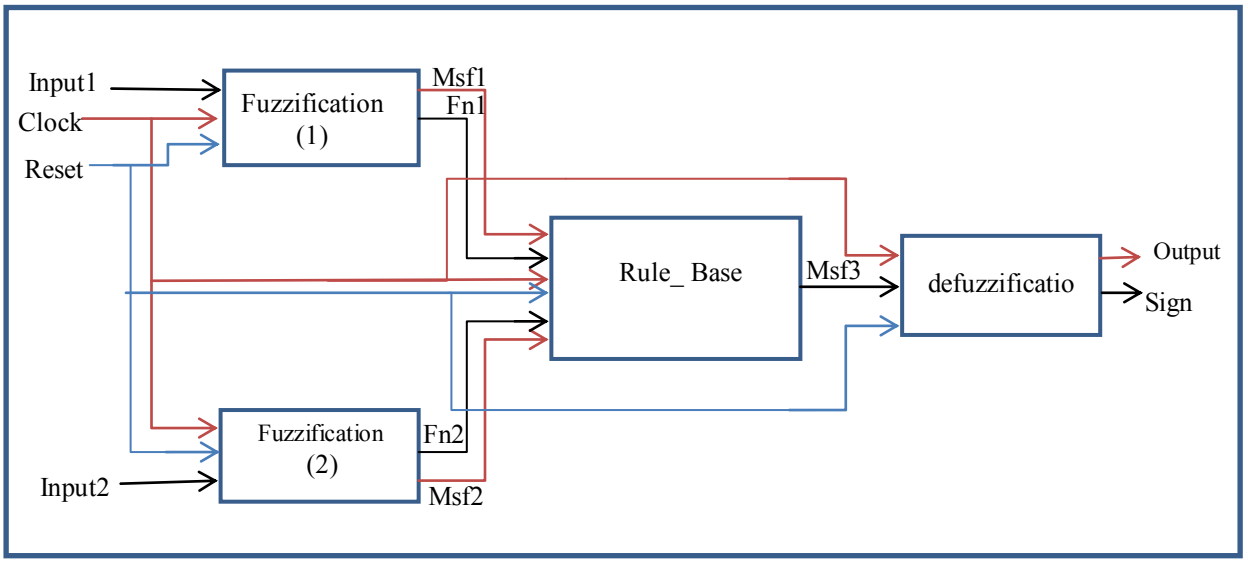

Fig. 5: Hardware Architecture

Fuzzification process has two arrays each one with five elements, fn and msf. The first array fn is represented as a binary vector (0, 1) indicates to which set (nb, ns, z, ps, pb) the input is lie, while the second array ,msf, represent the value of membership function belongs to this input. For example if input1 corresponds to 50H then the first array value is 01100. And the second array [00 1E E4 00 00] as shown in Figure 6.

	nb	ns	ze	ps	pb
fn	0	1	1	0	0

	nb	ns	ze	ps	pb
msf	00	1E	E4	00	00

Fig. 6: Fuzzification Arrays Example

For the proposed design, using two inputs, the fuzzification part is implemented twice: one for each input. For Input 1 (error), fuzzification process producing fn1 and msf1 and for input2 (change of error) it producing fn2 and msf2.

The procedure of the fuzzification process can be illustrated in the following algorithm:

For i=1 to n do
If input<a_1 then fn(i)=0;
Else fn(i)=1;
If ((input >a_1) and(input <a_2)) then
Msf(i)=(input-a_1)/($a_2 - a_1$);
else
If ((input >a_2) and(input <a_3)) then
Msf(i)=(a_3 − input)/($a_3 - a_2$);
End if ;
End if;
End loop;

Where:
n: number of membership of each input.
fn: array of member input.
msf: array of membership function value belongs to this input.

The designed logic circuit used to implement the i^{th} membership function of the fuzzification takes the architecture shown in Figure 7.

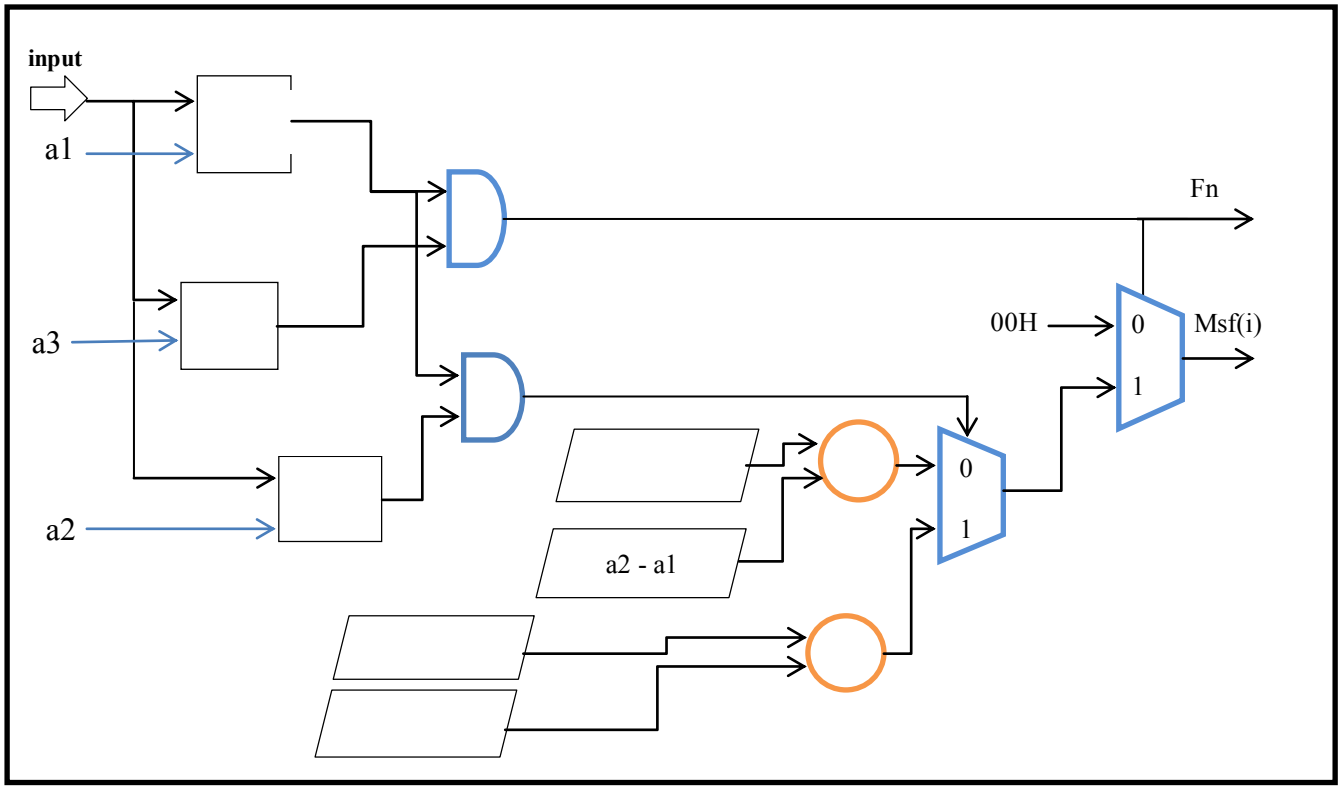

Fig. 7: Fuzzification Logic Circuit

After determine the degree of membership function in fuzzification part, if-then rules are generated for the system as shown in
Table I. The 25 rules formulated using Mac Vicar that are implemented using AND operator to represent the minimum degree of membership between the two input in the rule condition part.

Table I: Rule Base

x1 / x2	NB	NS	ZE	PS	PB
NB	NB	NB	NB	NS	ZE
NS	NB	NS	NS	ZE	PS
ZE	NB	NS	ZE	PS	PB
PS	NS	ZE	PS	PS	PB
PB	ZE	PS	PB	PB	PB

If there are more than one rules resulting in the same output, OR operator, represented by maximum, is used to generate the final output for this stage. The rules are represented as:

If x1 is nb AND x2 is nb then out is nb
If x1 is nb AND x2 is ns then out is nb

.
.
.

If x1 is pb AND x2 is pb then out is pb

Rule base are implemented in FPGA for the proposed design using two stages: firing stage shown in Figure 8 and combining stage. Firing stage consists of 25 rule divided into five part (equal to fuzzy sets) each with five rules. The combining stage aimed to obtain the maximum output of firing rules. This stage consists 20 combining elements (the logic circuit of each one shown in Figure 9) arranged in three levels: 10 in first level with input output signals shown in Table II, 5 in the second level with input output signals shown in Table III, and 5 in the third level with input output signals shown in Table IV.
The crisp output is computed by the deffuzzification process using Sugeno weighted average (centroid) method as in Equation 1. This can be obtained from multiplying each rule evaluation output with corresponding singleton value, summing these values together and dividing by the summation of rule evaluation output. In this architecture the mapping of the output extract positive and negative value of the output by using the variable signal (0, 1). The mapped output value is obtained from dividing the hexadecimal output value by FFH. In the proposed hardware architecture, sign is represented as one bit output signal (sign) (0 for +ve value), (1 for –ve value) of the output. The proposed architecture of the defuzzification section of the system is shown in Figure 10.

Table II. First Level Combining Elements

Combining Element no.	Input1	Input2	output
1	m0(0)	m0(1)	v0(0)
2	m0(2)	m0(3)	v0(1)
3	m1(0)	m1(1)	v1(0)
4	m1(2)	m1(3)	v1(1)
5	m2(0)	m2(1)	v2(0)
6	m2(2)	m2(3)	v2(1)
7	m3(0)	m3(1)	v3(0)
8	m3(2)	m3 (3)	v3(1)
9	m4(0)	m4(1)	v4(0)
10	m4(2)	m4(3)	v4(1)

Table III. Second Level Combining Elements

Combining Element no.	Input1	Input2	output
1	v0(0)	m0(4)	v0(2)
2	v1(0)	m1(4)	v1(2)
3	v2(0)	m2(4)	v2(2)
4	v3(0)	m3(4)	v3(2)
5	v4(0)	m4(4)	v4(2)

Table IV. Third Level Combining Elements

Combining Element no.	Input1	Input2	output
1	v0(1)	v0(2)	v0(3)
2	v1(1)	v1(2)	v1(3)
3	v2(1)	v2(2)	v2(3)
4	v3(1)	v3(2)	v3(3)
5	v4(1)	v4(2)	v4(3)

V. RESULTS

The proposed architecture of fuzzy logic system is simulated in Quarts II operate on Altera Cyclone III (EP3C16F484C8). The overall architecture is constructed from collecting the circuits of fuzzification, rule based, and defuzzification. The register transfer level RTL of the fuzzy logic system is shown in Figure 11.

For comparison purpose, a software version is run using Matlab software matlab2012b running on Lenovo laptop with Intel core i5-4200U of 1.6 GHz (4 cpus) and 4GB RAM. To verify the designed FPGA system functionality, the output is compared for Matlab and FPGA. Some such results are shown in Table V. When the proposed architecture is simulate in FPGA the test bench waveform is shown in Figure 12. The operational frequency is 160MH, 8 clock cycles are required to perform the overall operations, and therefore the execution time is 50 ns. The simulation report obtained from running is shown in Table VI.

Table V: Verification Result

Input1	Input2	Mapped input1 FPGA	Mapped input2 FPGA	FPGA output		Mapped FPGA output	Matlab output
				sign	Value in Hex		
-0.7490	-0.9294	20H	59H	0	AA	0.6666	0.6666
0.2550	-0.1215	A0H	70H	1	17	-0.0902	-0.1001
-0.4480	0.0431	40H	85H	0	6F	0.4353	0.4341
0.7803	0.7568	E3H	E0H	1	AA	-0.6666	-0.6667

As compared to related work using Cyclone FPGA to implement similar fuzzy system of Q. Khanh [9], the proposed architecture gives less logical elements usage with less execution time as shown in Table VII.

Table VI. Simulation Report

Logic Item	Used	Total	Ratio
Total logic elements	1,510	15,408	10%
Total combinational functions	1,502	15,408	10%
Total pins	40	347	12%
Dedicated logic registers	622	15,405	4%
Total LABs	107	963	11%

Table VII. Related Work Comparison

Feature	Proposed system	Q. Khanh
Logic Elements	1,510	6,264
Execution Time	50 ns	80 ns

Some implementation results are shown in Figure 13 (LAB logic elements) and Figure 14 (LAB-wide signals). About 107 LAB are use from the cyclone III board. Since LAB consists of 16 LEs, the detailed statistic gives 77 LAB used the sixteenth LE of each, 21 LABs used the ninth LE, two LABs are used by using the 2nd LE of them, two other LABs are used with the 15th LEs, other five LAB are used with single LE of each. LAB wide signals refers to the control signals used in the proposed design When software counterpart program is run in Matlab the required time is (42.84 us).

The speedup factor is generally used as a speed measure to compare the speed gain of hardware (FPGA) implementation as compared to software. The speedup factor is given in Equation 4[8].

$$speedup\ factor=\frac{the\ execution\ time\ of\ the\ software}{the\ execution\ time\ of\ the\ hardware} \quad \ldots(4)$$

Speedup factor=42.84us/50 ns=856.8

The computed speedup factor of the proposed design is (856.8).

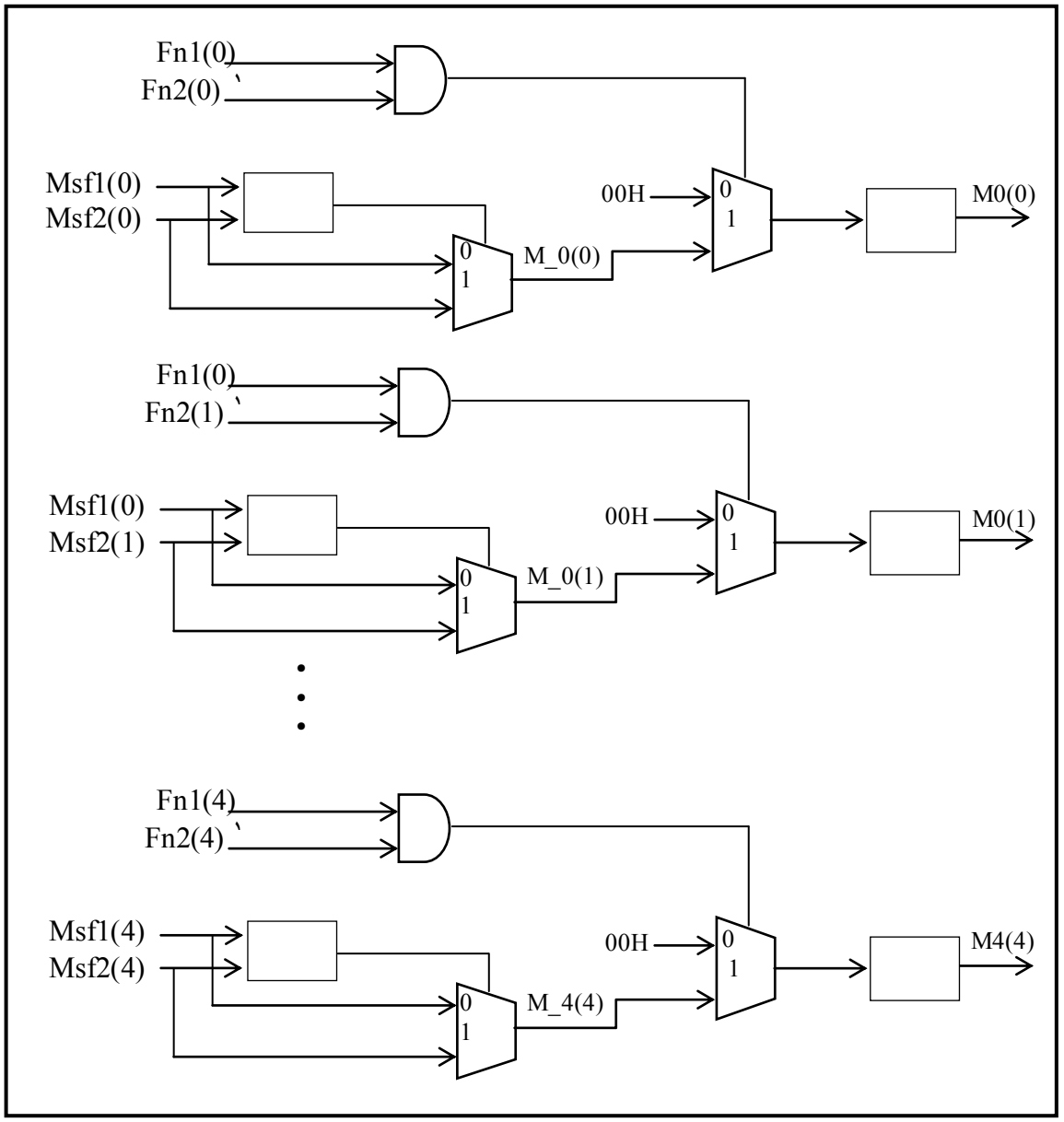

Fig.8: Rule Based Logic Circuit/ Firing Stage

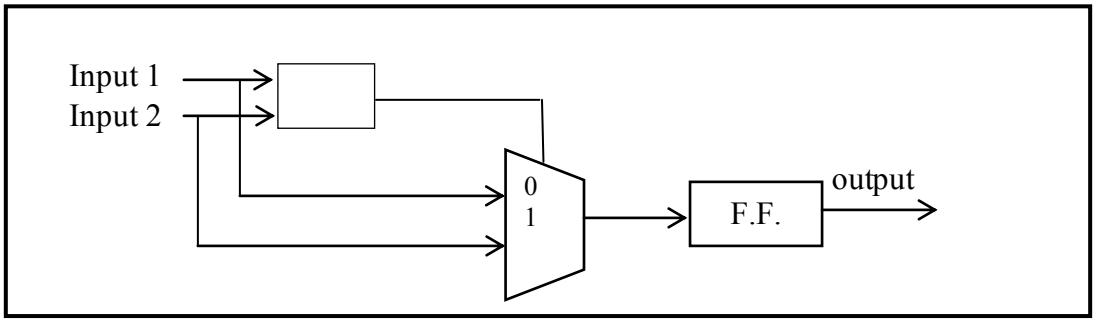

Fig.9: Rule Based Logic Circuit/ Combining Element

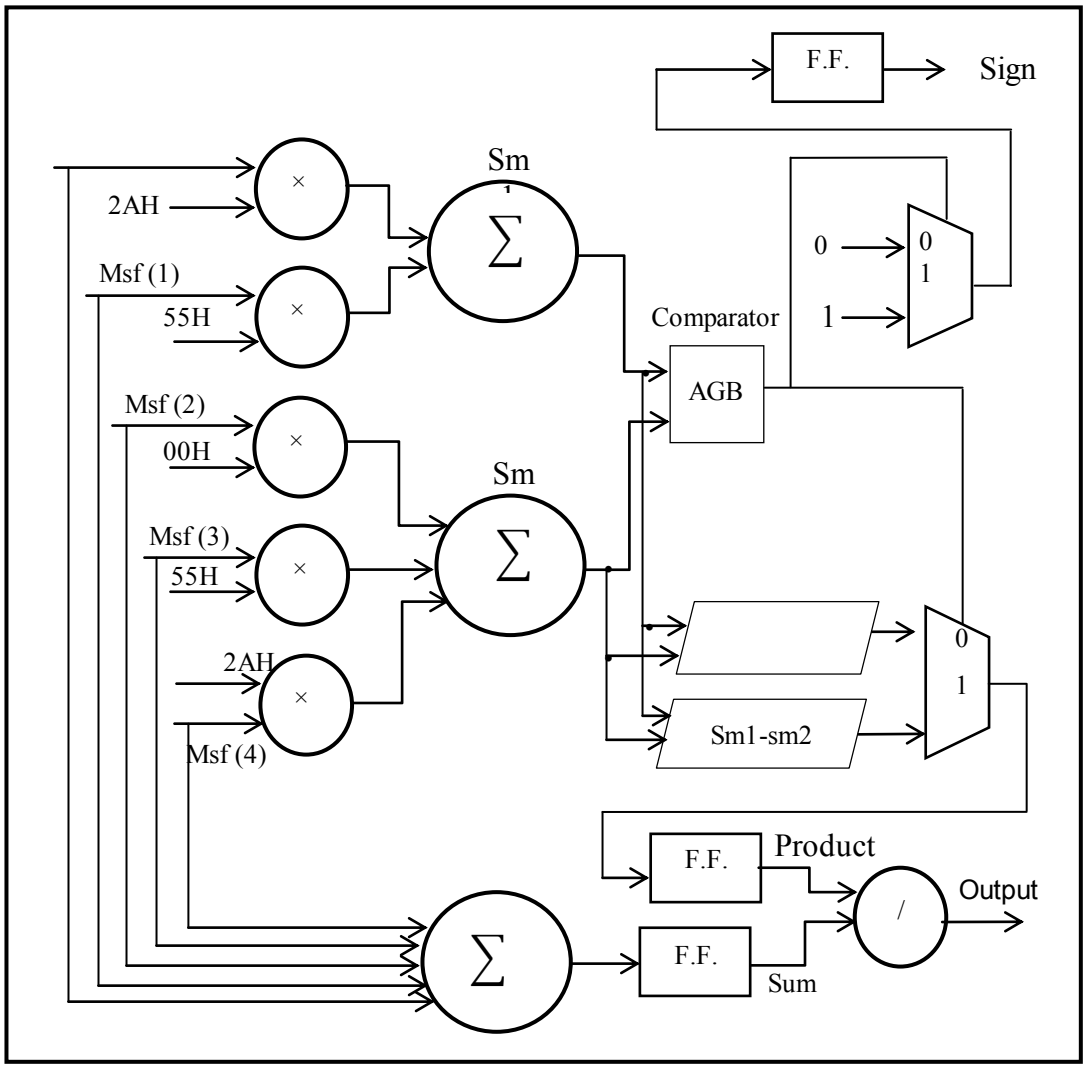

Fig.10: Defuzification Architecture

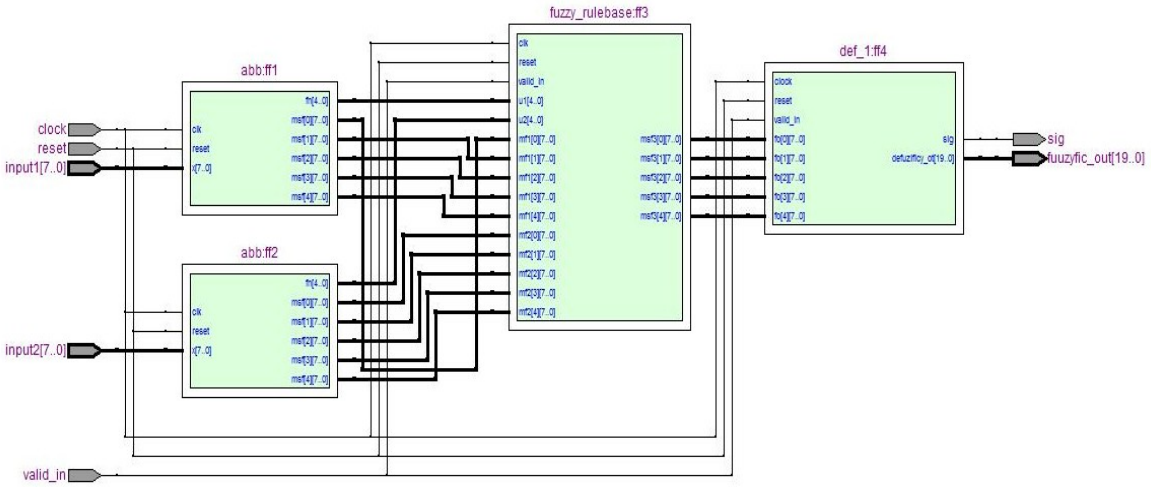

Fig. 11: RTL Viewer

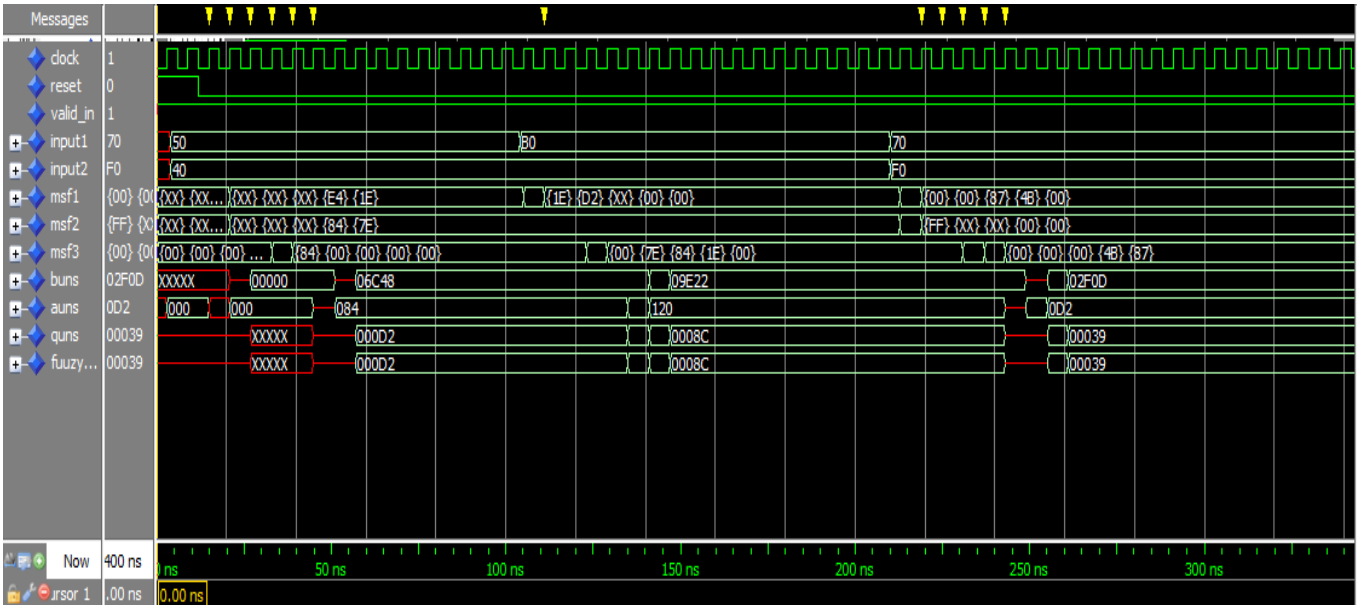

Fig. 12: Test Bench Waveform

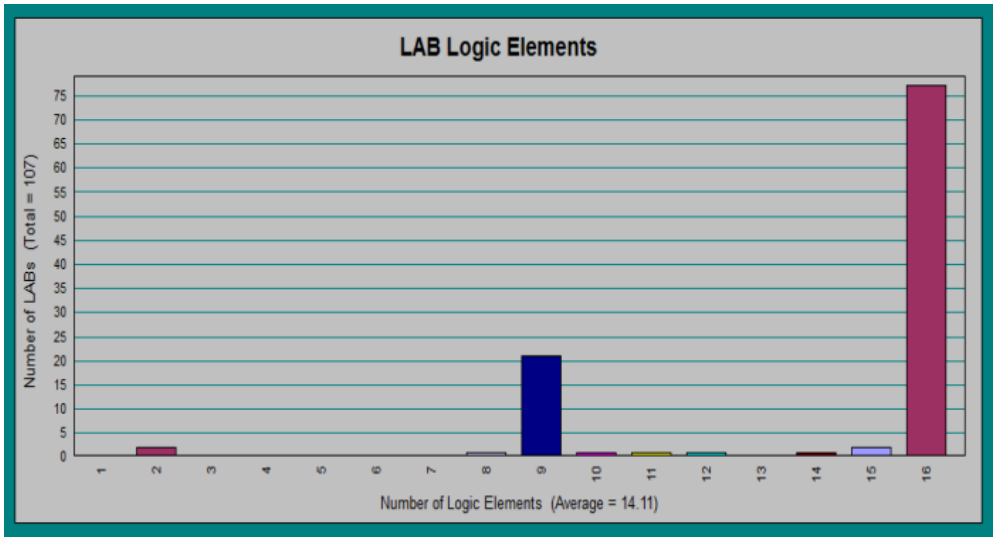

Fig. 13: LAB Logic Elements

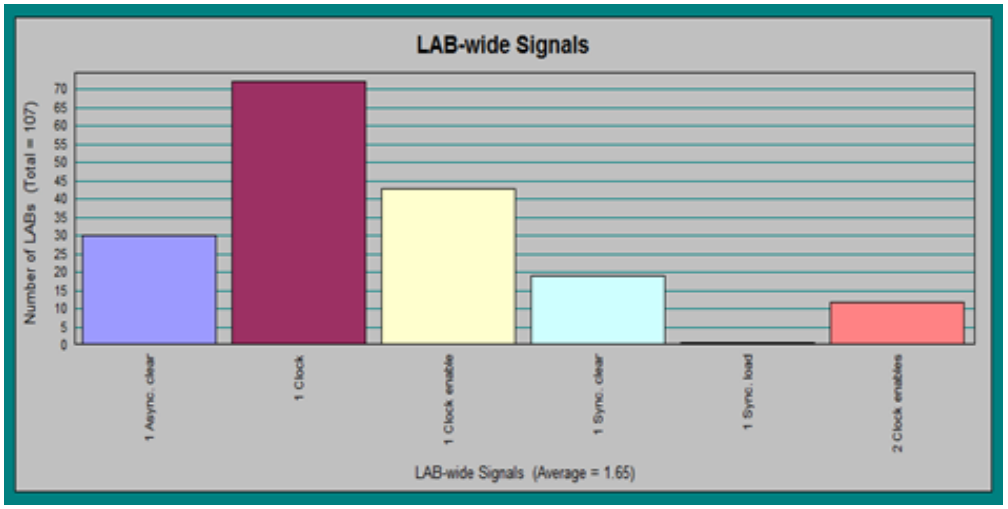

Fig. 14: LAB-Wide Signals

VI. CONCLUSIONS

The proposed architecture gives good result that is verified by software calculations in Matlab as in Table 2. The designed FPGA based fuzzy logic system makes fuzzy system calculation very fast with speed up factor 856.8 (FPGA fuzzy logic system about 856.8 time faster than software one). The designed architecture is implemented on low cost, low power Cyclone III Altera FPGA with high speed and less logic elements. The FPGA chip usage is no more 10 percent of the logic elements.

REFERENCES

[1] Z. Obaid, A. Sulaiman and M. . Hamidon, " FPGA-based Implementation of Digital Logic Design using Altera DE2 Board", IJCSNS International Journal of Computer Science and Network Security,Vol. 9, No. 8, pp. 186-194, 2009.

[2] Kasim M. Al-Aubidy, "FPGA Implementation of Fuzzy Inference System for Real Time Applications", Proceedings of the Fourth IASTED International Conference on Advances in Computer Science and Technology , pp. 406–411, 2008.

[3] N. Sulaiman, Z. Obaid, M. Marhaban, and M. Hamidon, " FPGA-Based Fuzzy Logic: Design and Applications – a Review", IACSIT International Journal of Engineering and Technology, Vol. 1, No. 5, pp. 491-503, 2009.

[4] Tukaram R. Kumbhar, Sunil S. Nirmale, and R. R. Mudholkar, "FPGA Implementation of Fuzzy Logic Controller for Tempreture Control", International Journal of Computer Applications, Vol. 62, No. 20, pp. 19-23, 2013.

[5] Virendra K. Verma, sarika sapre," A Methadology for Designing Fuzzy Logic Controller using VHDL ", International Journal of Engineering Trends and Technology (IJETT) Vol. 24, No. 4, pp. 217-222, 2015.

[6] Narendra M. Pathade, Prof. Sania M. Ansari, "Speed Control of DC Motor for Robotic Application Using Fuzzy Controller on FPGA Platform", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4, , No. 12, pp. 410-414, 2014.

[7] G. Kamalesh, T., Reddy " fpga implementation of high speed pi like fuzzy Control System for Indeustrial Automation Applications", International Journal of Modern Engineering Research(IJMER), Vol. 3, No. 3, pp. 1147-1453,2013.

[8] M. Joudah, "FPGA-Based Parallel Shortest Path Searching Processor for OSPF routing Protocol", PhD Thesis in Electrical Engineering, University of Basrah, 2013.

[9] Q. Khanh, "Development of FPGA based Control Architecture for PMSM Drivs", PhD Thesis in Information Technology, Faculty of Engineering and Information Technology, University of Technology, Sydny, 2015.