Center of Largest Area Defuzzifier Vnit VLSI Architecture

Asim Majeed Murshid , Ahmed Chalak Shakir , Sarmed Nihad Mohammed
Department of Computer Science , College of Science , University of Kirkuk , Kirkuk , Iraq
Asim.majeed2001@gmail.com

Abstract

Defuzzification unit combines fuzzy variables and fuzzy decisions to form a corresponding real (crisp or non-fuzzy) signal which can be used then to actuate some processes. Since the fuzzy data cannot be used directly for the real time applications, therefore it has to be converted into a crisp value. In designing the defuzzification unit in this work, the center of largest area has been used, because of its computationally efficient nature. The Center of Largest Area (COLA) of defuzzification is simple and is being generally used in comparison with more complex center of sums defuzzification method. The proposed architecture has been modeled in Very High Description Language (VHDL) and implemented in XILINX and Spartan field programmable gate arrays (FPGA).The proposed architecture is more efficient in area and the speed of operation in comparison with more complex architecture used for the Center of sums. The functional analysis has revealed that the proposed architecture is implementing COLA based defuzzifier efficiently and accurately.

Keywords: Defuzzification; Fuzzy processor; Center of largest area; VLSI design.

Introduction

The modern concept of fuzzy sets was introduced by Lotfi Zadeh in his work "Fuzzy Sets" which described the mathematics of fuzzy set theory nearly three decades ago. Although a relatively new theory, fuzzy logic has been used in many engineering applications because being considered as a simplistic solution available for the specific problems. In 1920, Lukasiewicz extended the conventional bi-valued logic of Aristotle and proposed a new tri-valued logic in his paper entitled "On three-valued logic"[1-2]. He added a new truth value to the truth logic 0 and false logic 1 and can be best translated as "possible," with a numerical values assigned between True and False [3-5].

The advantages of fuzzy logic over the traditional solutions is that it allows computers to be able to reason more like humans, respond effectively to complex inputs to deal with notations like "too hot", "too cold", "too high" or "just right". Further, there is no need of advanced mathematics or theory to develop or implement a fuzzy logic system. The fuzzy data obtained from the fuzzification process is not suitable for the real time applications and have to be converted into crisp form. The conversion of data from fuzzy form to crisp form is known as the defuzzification, also called as "rounding off". The defuzzification has the capability to reduce a fuzzy quantity to a crisp single valued quantity. It reduces the collection of membership function values into a single quantity [6-9].

Related works

The various fuzzy systems are realized by different researcher for different applications. The original digital realization of fuzzy inference processor was performed by Toga and Watanabe [10-11].

H. Peyravi et al. [12] have proposed reconfigurable inference engine for the analog fuzzy logic controller, based on Mamdani inference technique. J.M. Jou et al. [13], R. d 'Amore [14] and N. E. Evmorfopoulong et al. [15] have proposed different architecture for the fuzzy inference processor. A significant improvement is reducing power and reducing redundancy has been obtained in these structures. The inference engine performance is an important issue which needs to be addressed.

Many researchers have done a work on the defuzzification fuzzy processors. Roberto d'Amore et al [12] has developed a two input one output bit scalable architecture for fuzzy processors. In this work, a model has been developed from which the defuzzified value has been obtained manually using the COLA technique initially. An architecture has been designed and developed for the model and the functional analysis has been performed using VHDL and implemented in Vertex-4 field programmable gate array (FPGA). In this work, three models have been designed to test the functionality of the proposed fuzzy processor. It has been observed that a complete match exists between the manual calculation of defuzzified values and the values calculated by using the proposed architecture. This shows that the architecture developed is calculating the defuzzified values correctly and efficiently.

Defuzzification process

The conversion of data from fuzzy form to crisp form is known as the defuzzification, also called as "rounding off". It reduces the collection of membership function values into a single quantity. The different defuzzification methods used in literature are, (1) Max membership principle. 2. Centroid method. 3. Weighted average method. 4. Mean-max membership. 5. Centre of sums. 6. Center of largest area. 7. First of maxima or last of maxima. These methods have their own applications, advantages and disadvantages [1-4].

The researchers have developed various architectures of defuzzifiers depending on what application the fuzzy processor or the fuzzy controller is being designed for. Figure 1 shows the block diagram of a defuzzifier circuit. C1X1, C1X2, C1X3, C1X4, HC1, C2X1, C2X2, C2X3, C2X4, HC2 are the fuzzy inputs to the defuzzifier, which comprises of elements and their associated membership functions. The defuzzifier block is having an architecture intact based the defuzzification techniques as mentioned above to extract the defuzzified or the crisp value. The crisp output value only can be used to control various processes or mechanisms. The defuzzification technique used in this work is the Center of largest area.

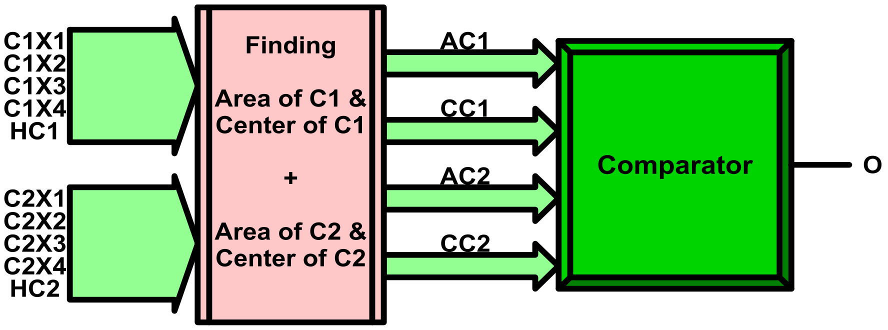


Figure 1: Block diagram of a defuzzifier

This method uses the overall output or union of all individual output fuzzy sets, also is the most accurate method among all. The researchers have generally designing the Very Large Scale Integration (VLSI) architecture of the COLA defuzzifier; however, in this work an efficient COLA based defuzzifier is designed and implemented. Equation 1 shows the model for the COLA based defuzzification [1-4].

$$Z^* = \frac{\int \mu c(z) z dz}{\int \mu c(z) dz} \qquad (1)$$

In this work, we have used a model to study and realize the defuzzification architecture as shown in Figure 2. In this model two rules are being fired and Mamdani implication has been used for inference. We have used aggression of rules, as the overall consequent is being obtained from the two individual consequent in each set. In each set, the minimum of two antecedent membership values is propagated to the consequent as “AND” connective is used between the two antecedents in the rule. The propagated membership value from operations on the antecedents truncates the membership function for the consequent for that rule.

The truncated membership functions from each rule are aggregated according to the following equation used for conjunctive system of rules [3].

$$\mu y(y) = min\ (\mu y^1\ (y), \mu y^2\ (y), \ldots\ldots\ldots, \mu y^{\ r}\ (y))\ for\ y \in Y \qquad (2)$$

Each rule comprises of two antecedents (A1, B1 and A2, B2) and one consequent (C1, C2) and is represented as

$$IF(X\ is\ A1)\ AND\ (Y\ is\ B1)\ then\ (Z\ isC1) \qquad (3)$$

$$IF(X\ is\ A2)\ AND\ (Y\ is\ B2)\ then\ (Z\ isC2) \qquad (4)$$

In MODEL 1, as shown in Figure 2, FA1 and FB1 are the first and second fuzzy antecedents of the first rule, respectively, and C1 refers to the fuzzy consequent of that fuzzy rule. Since; the antecedents are connected by logical "AND", therefore, the minimum membership value of the antecedents propagate through to the consequent and truncates the membership function for the consequent of each rule. Similarly, FA2 and FB2 are the first and second fuzzy antecedents of the second rule, respectively, and C2 refers to the fuzzy consequent of the second fuzzy rule. This process is the inference and it follows Mamdani’s implication method, which is the most common in practice and in the literature. The union of two consequents C1 and C2 is shown in row 3 of Figure 2. From this union the defuzzified value is being obtained by using the Center of Largest Area (COLA) technique, as given in equation 1.

Calculation of defuzzified values

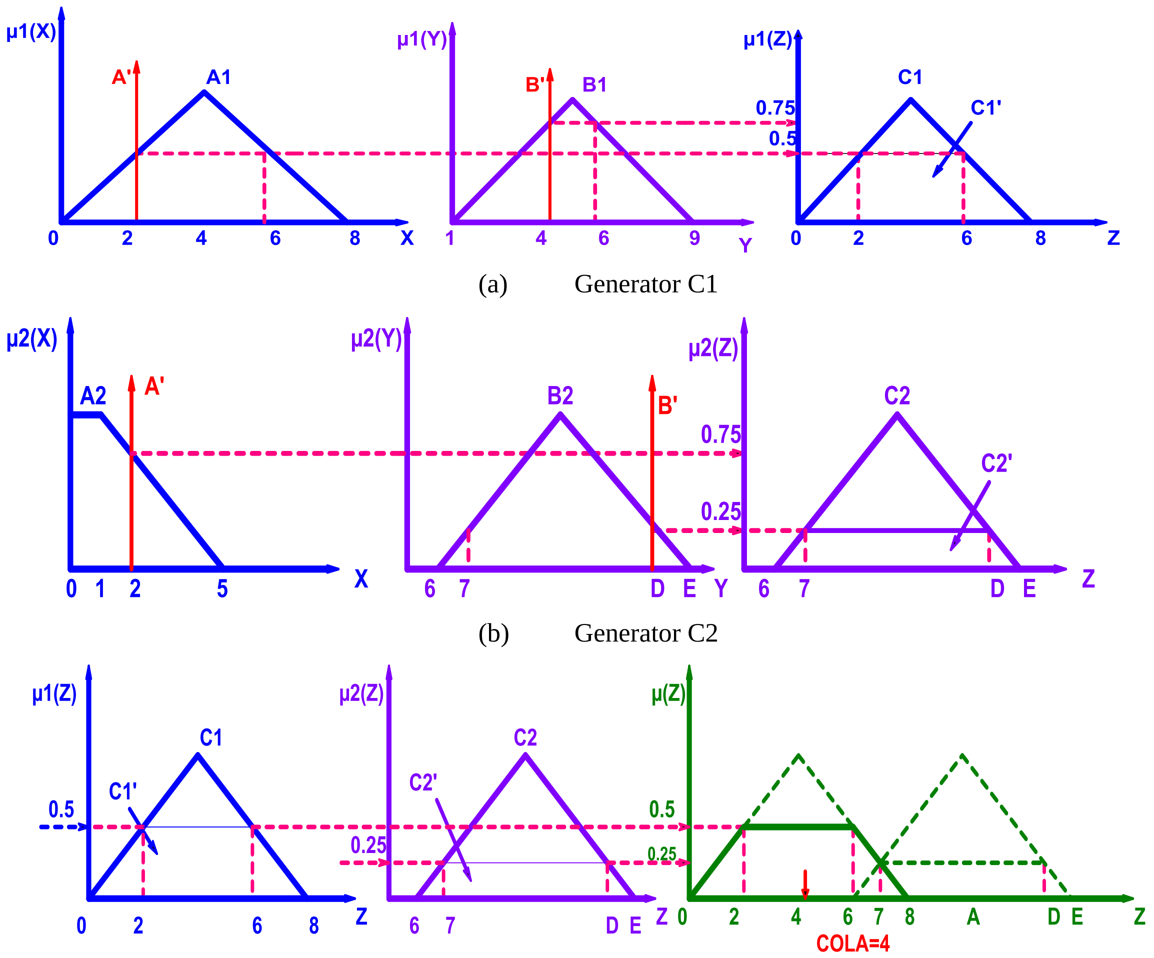
The defuzzified or crisp value for MODEL 1 can be calculated first manually and then using the proposed architecture. It is important that the two values must match; otherwise the proposed architecture is not well designed. In this study, Center of Largest Area (COLA) is being used to calculate the defuzzified value. Figure 3 again shows the defuzzification for the MODEL 1 as given by Figure 2. The application of equation 1 to Figure 3 will generate the defuzzified value of model 1. The defuzzified value obtained out of COLA defuzzifier (Y_{COLA}) is again given by equation (1')

The calculations go as follows

$$Y_{COLA} = Z^* = \frac{CR1 * AR1 + CR2 * AR2}{AR1 + AR2 +} \qquad (1')$$

Where CR1 and CR2 are the Center of region R1 and R2 respectively, AR1 and AR2 are the Area of region R1 and R2 respectively.

Model (1):



(a) Generator C1

(b) Generator C2

(c) Adding C1 & C2

Figure 2:Fuzzification, Inference and Defuzzification using Center of Largest Area for model 1

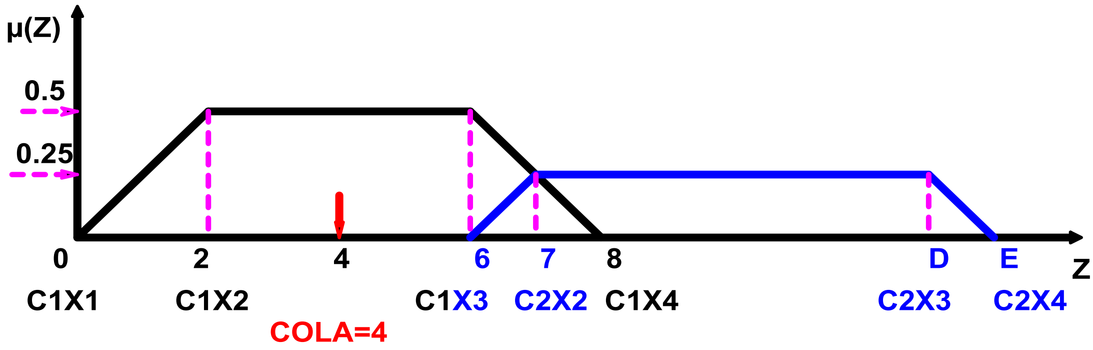


Figure 3:Defuzzification for MODEL 1

In the center of largest area method we first determine the areas of the two individual convex fuzzy output sets. After selecting Largest area we find the center of largest area.
Area of trapezoid = 0.5 × height × (L1 + L2)
Hence, L1 is longer than L2 in trapezoid C1 or C2.
Where L1 = (C1X4 – C1X1) and L2 = (C1X3 – C1X2) for trapezoid C1, therefore;
(L1 + L2) = [(C1X4 + C1X3) – (C1X1 + C1X2)] = [C1X34 – C1X12]
Meaning (L1 + L2) = C1X
Where C1X1, C1X2, C1X3, C1X4 and C2X1, C2X2, C2X3, C2X4 are the representation point of trapezoid MFs C1 and C2 respectively, also HC1 and HC2 are the Height value of trapezoid MFs C1 and C2 respectively.

Height of C1 (HC1) = 0.5 = A H.
C1X1 = 0,C1X2 = 2, C1X3=6,C1X4=8
Center of C1 = CC1 = $\frac{C1X4 - C1X1}{2}$ + C1X1 = $\frac{8-0}{2}$ + 0 = 4.
C1X34 = C1X3 + C1X4 = 6 + 8 = 14,C1X12 = C1X1+ C1X2 = 0 + 2 = 2
C1X = C1X34 - C1X12 = 14 - 2 = 12
VC1 = $\frac{C1X}{2}$ = $\frac{12}{2}$ = 6.
Area of trapezoid C1 = height × $\frac{(L1+L2)}{2}$ = HC1 × $\frac{C1X}{2}$ = HC1 × VC1
AC1 = HC1 * VC1 = 10 * 6 = 60.
height of C2 = HC2 = 0.25 = 5H.
C2X1 = 6,C2X2 = 7,C2X3=13,C1X4=14

Center of C2 = CC2 = $\frac{C1X4 - C1X1}{2}$ + C1X1 = = $\frac{14-6}{2}$ + 6 = 10.
C2X34 = C2X3 + C2X4 = 13 + 14 = 27,C2X12 = C2X1+ C1X2 = 6 + 7 = 13
C2X = C2X34 – C2X12 = 27 - 13 = 14.
VC2 = $\frac{C2X}{2}$ = $\frac{14}{2}$ = 7.
AC2 = HC2 * VC2 = 5 * 7 = 35.
AC1 > AC2 = 60 > 35
∴ C1 is Largest Area
∴ Center of Largest Area is Center of C1 (CC1). = 4.
tested. The final defuzzification diagram of that model is given in Figure 4. Again COLA is used to calculate the defuzzified value. The calculations have shown that the final crisp value as 10H.

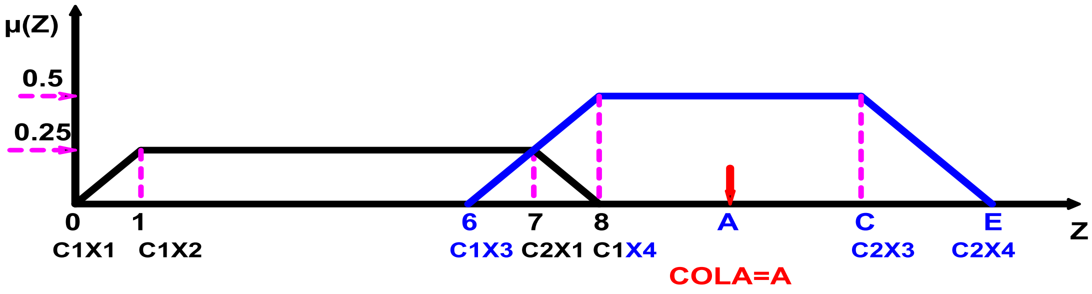


Figure 4: Defuzzification for MODEL 2

Height of C1 = HC1 = 0.25 = 5 H.
C1X1 = 0,C1X2 = 1,C1X3=7,C1X4=8
Center of C1 = CC1 = $\frac{C1X4 - C1X1}{2}$ + C1X1 = = $\frac{8-0}{2}$ + 0 = 4.
C1X34 = C1X3 + C1X4 = 7 + 8 = 15,C1X12 = C1X1+ C1X2 = 0 + 1 = 1
C1X = C1X34 - C1X12 = 15 - 1 = 14
VC1 = $\frac{C1X}{2}$ = $\frac{14}{2}$ = 7.
Area of trapezoid C1 = height × $\frac{(L1+L2)}{2}$ = HC1 × $\frac{C1X}{2}$ = HC1 × VC1
AC1 = HC1 * VC1 = 5 * 7 = 35.
Height of C2 = HC2 = 0.5 = AH.
C2X1 = 6,C2X2 = 8,C2X3=12, C1X4=14
Center of C2 = CC2 = $\frac{C1X4 - C1X1}{2}$ + C1X1 = = $\frac{14-6}{2}$ + 6 = 10.
C2X34 = C2X3 + C2X4 = 12 + 14 = 26,C2X12 = C2X1+ C1X2 = 6 + 8 = 14
C2X = C2X34 – C2X12 = 26 - 14 = 12.
VC2 = $\frac{C2X}{2}$ = $\frac{12}{2}$ = 6.
AC2 = HC2 * VC2 = 10 * 6 = 60.
AC2 > AC1 = 60 > 35

∴ C2 is Largest Area
∴ Center of Largest Area is Center of C2 (CC2). = 10.

The same procedure has been used for other defuzzified model, not shown in a figure. The defuzzified value obtained is 5H, all results of models 1, 2 and 3 as shown in table 1.

Hardware realization of the defuzzification process

In section III, the defuzzified values have been manually calculated using the Center of Largest Area, as shown in Figure 3 and Figure 4. However, manual calculation is not sufficient, a real hardware is needed which will automatically calculate the defuzzified values. Therefore, in this section, a novel architecture of a defuzzifier based on C.O.L.A. technique has been designed and simulated. The proposed architecture has been modeled in VHDL and has been implemented in field programmable gate array (FPGA). The defuzzified output in the C.O.L.A. method is given by the following equation

$$\mathbf{Z^*} = \frac{\int \mu c(z) z dz}{\int \mu c(z) dz} \qquad (1)$$

TABLE 1: Calculation of Defuzzification parameters

parameters	Model (1)	Model (2)	Model (3)
C1X1	0	0	0
C1X2	2	1	3
C1X3	6	7	7
C1X4	8	8	10
Height of C1	10	5	15
Center of C1	4	4	5
Area of C1	60	35	105
C2X1	6	6	6
C2X2	7	8	8
C2X3	13	12	12
C2X4	14	14	14
Height of C2	5	10	10
Center of C2	10	10	10
Area of C2	35	60	60
Largest Area	C1	C2	C1
Center of Largest Area	4	10	5

Figure 5 are the architecture realization of the proposed architecture of equation (1). The VHDL modeling of the proposed architectures have been performed. The functional analysis is shown in Figure 6. It is clear from the functional analysis that there is a clear cut match between the results obtained manually and results generated by the architecture. In the first T-state of the functional analysis, the output of the defuzzifier 'O' is 4H, which is same as obtained manually for Figure 3. Similarly, for T-State second and third, the outputs of the defuzzifier is AH and 5H respectively, which is same as calculated manually. This shows the proposed architecture is realizing the defuzzifier action efficiently and accurately.

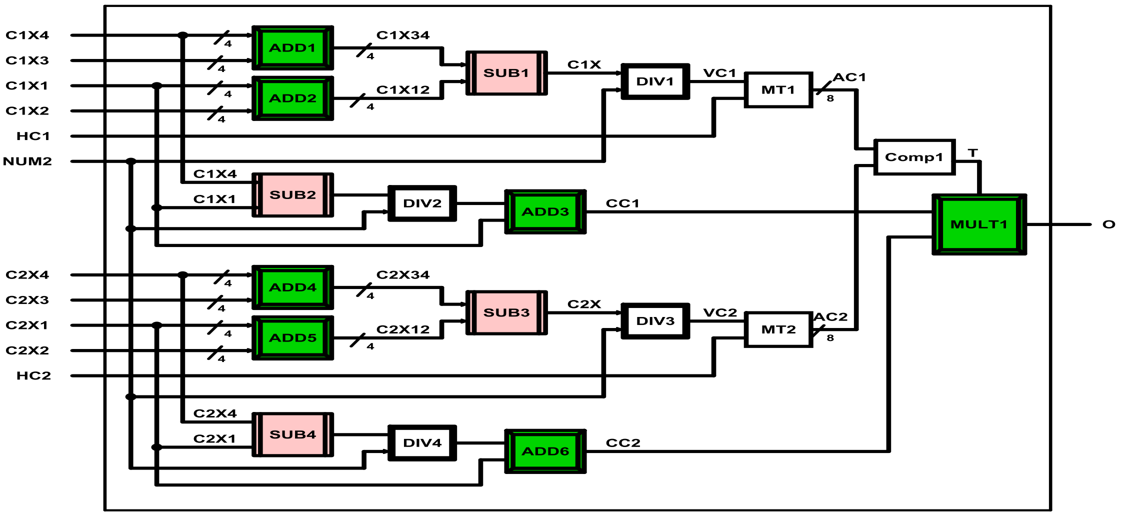


Figure 5: Proposed Architecture of the defuzzifier

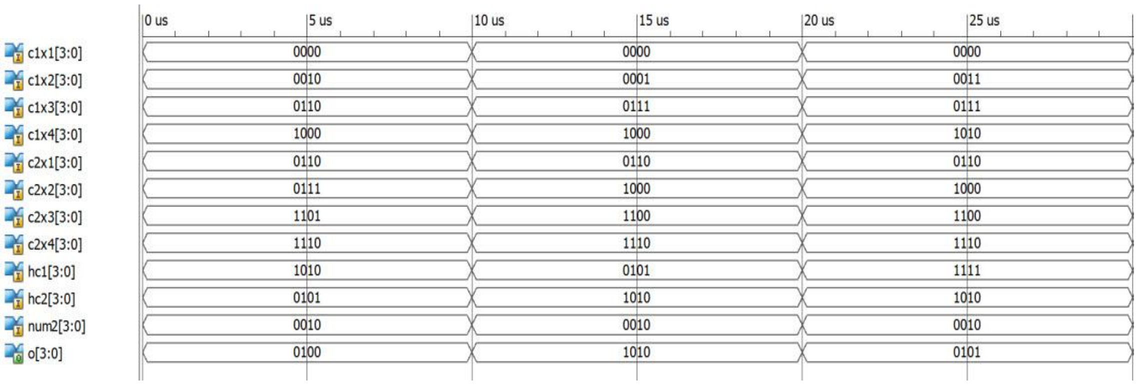


Figure 6: Timing diagram of the proposed defuzzifier

Fpga implementation of the defuzzifier

A Spartan 3E XC3S100E FPGA platform from XILINX has been used to implement the proposed architecture. The FPGA has around 1920 4-input look up tables (LUT) and 66 bonded input/output buffers (IOB). The FPGA logic resource used in an implementation are shown in Table 2, schematic and netlist generated of the proposed defuzzifier shown in Figure 7 and 8 respectively. The implementation results shown in Table 2 show that there is further scope for improvement in the proposed structures by incorporeity more parallelism in the architecture.

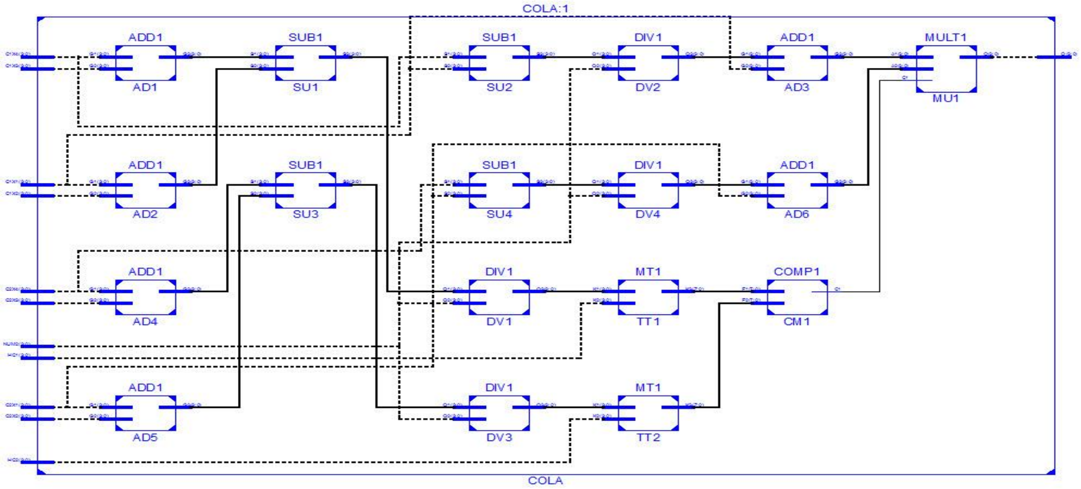


Figure 7: Schematic of the proposed defuzzifier

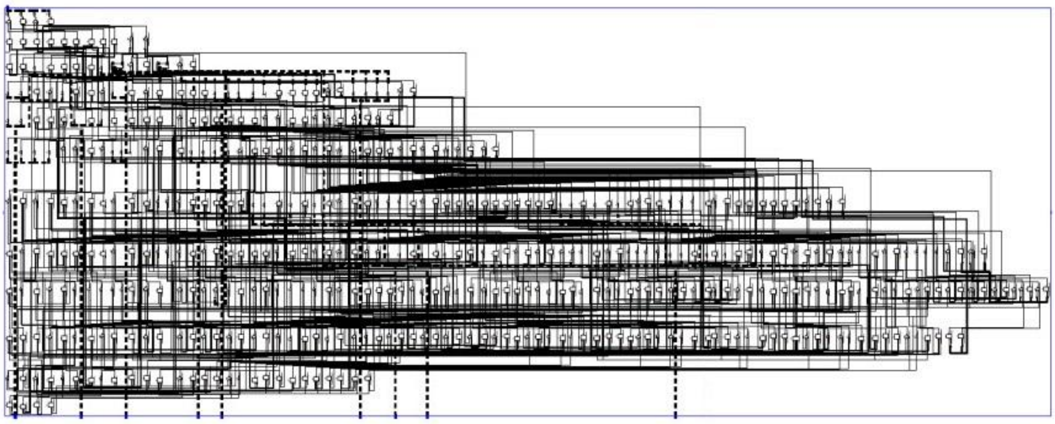


Figure 8: Netlist generated by the synthesis tools.

TABLE 2: FPGA Implementation results

Device Utilization Summary (FPGA: Spartan 3E XC3S100E)		
Logic Utilization	**Used**	**Available**
Number of 4 input LUTs	281	1,920
Number of occupied Slices	145	960
Number of Slices containing only related logic	145	145
Number of Slices containing unrelated logic	0	145
Total Number of 4 input LUTs	281	1,920
Number of bonded IOBs	48	66
Number of MULT 18X18SIOs	2	4
Average Fanout of Non-Clock Nets	2.98	

Conclusion

The proposed architecture is based on Center of Largest Area. The COLA is the most important and simplest of the defuzzification methods. The defuzzified values have been initially manually calculated and finally have been obtained by the proposed architecture. The VHDL modeling and Spartan 3E FPGA implementation of the proposed architecture has been done. It has been seen that the proposed architecture realizes COLA based defuzzifier efficiently, as there is a complete match between the results obtained manually and through the architecture.

References

[1] L. A. Zadeh, “Fuzzy logic”, *IEEE Computer*, vol.1 no. 4, pp. 83–93, 1988.

[2] T. J. Ross, “fuzzy logic with engineering applications," *by John Wiley Sons inc.*, 2005.

[3] E. H. Mamdani, “Applications of fuzzy algorithm for simple dynamic plant”, *Proc. Inst. Elect. Eng.*, Vol. 121, pp. 1585-1588, 1974.

[4] C. C. Lee, “Fuzzy logic in control systems: fuzzy logic controller-part I”, *IEEE Trans. Syst., Man. Cybern.*, Vol. 20, no. 2, pp. 404-418, 1990.

[5] A. M. Murshid, S. A. Loan, S. A. Abbasi, and A. M. Alamoud, “VLSI Architecture of Fuzzy Logic Hardware Implementation: a Review," *Int. Journal of Fuzzy Systems*, Vol. 13, No. 2, pp. 74-88, June 2011.

[6] A. M. Murshid, S. A. Loan, S. A. Abbasi, and A. M. Alamoud, “A ovel VLSI architecture for a fuzzy inference processor using Triangular-shaped membership function," *Int.Journal of Fuzzy Systems*, Vol. 14, No. 3, pp. 345-360, Sep. 2012

[7] S. A. Loan, A. M. Murshid, S. A. Abbasi, and A. M. Alamoud, “A novel VLSI architecture for a fuzzy inference processor using Gaussian-shaped membership function," *Journal of Intelligent* and fuzzy systems, vol.23, pp. 1-16, 2012.

[8] Sajad A. Loan, Asim M. Murshid and Faisal Bashir, “A Novel VLSI Architecture of a Defuzzifier Unit for a Fuzzy Inference Processor’, in proceedings of IEEE ICEASE 2013.

[9] Sajad A. Loan, Asim M. Murshid, Ahmed C. Shakir, Abdul Rahman Alamoud and Shuja A. Abbasi, “A Novel VLSI Architecture of a Weighted Average Method based Defuzzifier Unit’, in proceedings of International MultiConference of Engineers and Computer Scientiists, Hong Kong, 12-14 March, 2014.

[10] H. Watanabe, W. D. Dettloff, and K. E. Yount, “A VLSI fuzzy logic Controller with reconfigurable, cascade architecture”, *IEEE Journal of Solid-State Circuits*, Vol. 25, pp. 376-381, 1990.

[11] M. Togai and H. Watanabe, “Expert system on a chip: an engine for real-Time approximate reasoning”, *IEEE Expert Mag.*, Vol. 1, pp. 55- 62, 1986.

[12] H. Peyravi, A. Khoei, and K. Hadidi, “Design of an analog CMOS fuzzy logic controller chip”, *Fuzzy Sets and Systems 132*, PP. 254- 260, 2002.

[13] J. M. Jou and P. Y. Chen”, “An adaptive fuzzy logic controller: its VLSI architecture and applications”, *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, Vol. 8, pp. 52-60, 2000.

[14] R. d’Amore, O. Saotome, and K. H. Kientiz, “A Two – Input, One – Output Bit – Scalable Architecture for Fuzzy Processors,” *IEEE Computer Society press*, vol. 18, pp. 56-64, July 2001.

[15] N. E. Evmorfopoulos, and J. N. Avaritsiots, “An adaptive digital fuzzy architecture for application-specific integrated circuits”, *Active and Passive Elec. Comp.*, Vol. 25, pp. 289-306, 2002.

معمارية الدوائر المتكاملة الفائقة المدى لوحدة إعادة التنضيب

عاصم مجيد مرشد ، أحمد جالاك شاكر ، سرمد نهاد محمد

قسم علوم الحاسبات ، كلية العلوم ، جامعة كركوك ، كركوك – العراق

Asim.majeed2001@gmail.com

الملخص

وحدة فك الضبابية يجمع بين المتغيرات الغامضة والقرارات الغامضة لتشكيل الإشارة الحقيقية والتي يمكن استخدامها بعد ذلك لتحفيز بعض العمليات. بما إن البيانات الغامضة لا يمكن استخدامها مباشرة في تطبيقات الزمن الحقيقي، وعليه فإنه لا بد من تحويلها إلى قيمة واضحة. فبهذا العمل تم تصميم وحدة فك الضبابية، وبسبب خواصه الفعالة حسابيا فقد استخدم مركز اكبر مساحة في التصميم. مركز الـ COLA من فك الضبابية بسيطة ويتم استخدامه بشكل عام بالمقارنة مع طرق أكثر تعقيدا من طريقة مركز الجموع. تم غرار الهيكل المقترح في VHDL و تنفيذها في XILINX والميدان المتقشف لمصفوفات البوابات القابلة للبرمجة (FPGA). الهيكل المقترح هو أكثر كفاءة في المنطقة وسرعة العملية بالمقارنة مع بنية أكثر تعقيدا تستخدم لمركز الجموع. وقد تبين من التحليل الوظيفي أن الهيكل المقترح بتنفيذ COLA لفك الضبابية لديها كفاءة ودقة.

الكلمات المفتاحية: فك الضبابية ؛ عمليات الضبابية ؛ مركز اكبر مساحة ؛ تصميم VLSI.