Center of Sums based Defuzzifier Unit VLSI Architecture

Asim Majeed Murshid

Department of Computer Science , College of Science , University of Kirkuk , Kirkuk , Iraq

Asim.majeed2001@gmail.com

Abstract

Defuzzification is a process of getting a crisp value from the fuzzy data. 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 sums has been used, because of its computationally efficient nature. The Center of Sums (COS) is faster than many defuzzification method and the method is not restricted to symmetric membership function is simple and is being generally used in comparison to more complex Center of gravity (also called Center of area or Centroid method) defuzzification method. The proposed architecture has been modeled in VHDL and implemented in XILINX and Spartan field programmable gate arrays (FPGA). The proposed architecture is more efficient in area (the area of implementation of VLSI Architecture) and the speed of operation in comparison to a more complex architecture used for the Center of gravity. The functional analysis has revealed that the proposed architecture is implementing COS based defuzzifier efficiently and accurately.

Keywords: Defuzzification; Fuzzy processor; Center of Sums method; Low power; VLSI design.

Introduction

The idea of fuzzy logic was proposed by Lotfi A. Zadeh of the University of California at Berkeley in a 1965 paper. He introduced the concept of "linguistic variables", in his paper in 1973. Other research followed, with the first industrial application, a cement kiln built in Denmark, coming on line in 1975 [1, 2]. The fuzzy logic finds applications nowadays in almost all fields of engineering and science, including control system, consumer electronics, data processing, expert systems, computer vision, and signal processing and so on [3, 4].

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" [5 , 6]. Further, there is no need of advanced mathematics or theory to develop or implement a fuzzy logic system [7, 8]. 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 [9,10].

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 [11,12].

H. Peyravi et al. [13] have proposed reconfigurable inference engine for the analog fuzzy logic controller, based on Mamdani inference technique. An adaptive memory efficient fuzzy processor is designed by J.M. Jou et al. [14]. A bit scalable architecture of fuzzy processors is proposed, this architecture is suited for automatic synthesis of digital fuzzy controller by R. d 'Amore [15]. N. E. Evmorfopoulong et al. [16] have proposed different architecture for the fuzzy inference processor. A significant improvement is reducing power and reducing redundancy has been obtained in this structure. The inference engine performance is an important issue which needs to be addressed.

Many researchers have done a work on the defuzzification processor, Roberto d'Amore et al [13] 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 COS technique initially. An architecture has been designed and developed for the model and the functional analysis has been performed using Very High Description Language (VHDL) and implemented in Spartan-3E 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 proving that the architecture developed is calculating the defuzzified values correctly and efficiently.

Defuzzification Process

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". 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 [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 Sums (COS).

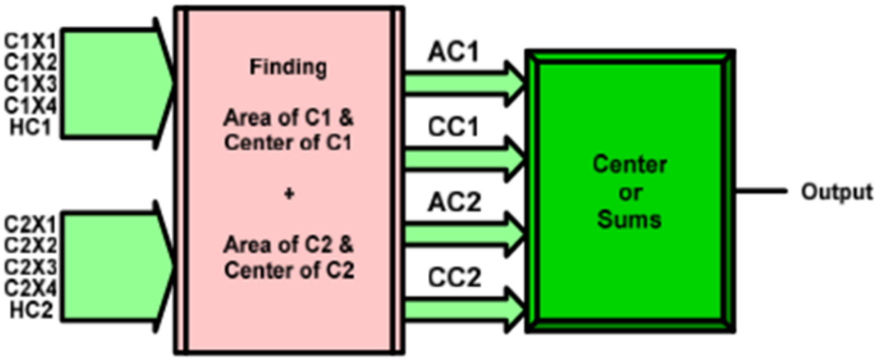


Figure 1: Block diagram of a defuzzifier.

$$\mathbf{Z^*} = \frac{\int \bar{Z} \sum_{k=1}^{n} \mu c(z) dz}{\int \sum_{k=1}^{n} \mu c(z) dz} \qquad \dots(1)$$

Where $\int$ denotes an algebraic integration, $\sum$ denotes the algebraic sum, $\bar{Z}$ is the centroid of each symmetric membership function and μc(z) the area of application.

In this work, we have used a model to study and realize the defuzzification architecture. In this model, two rules are being fired and Mamdani implication has been used for inference [17]. 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].

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

where $\mu y(y)$ is first operation of fuzzification is consisting minimum of $\mu y(y)$ from (1 to r)

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

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

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

In Model 3, 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 Sums (COS) technique, as given in equation 1.

Calculation of defuzzified values

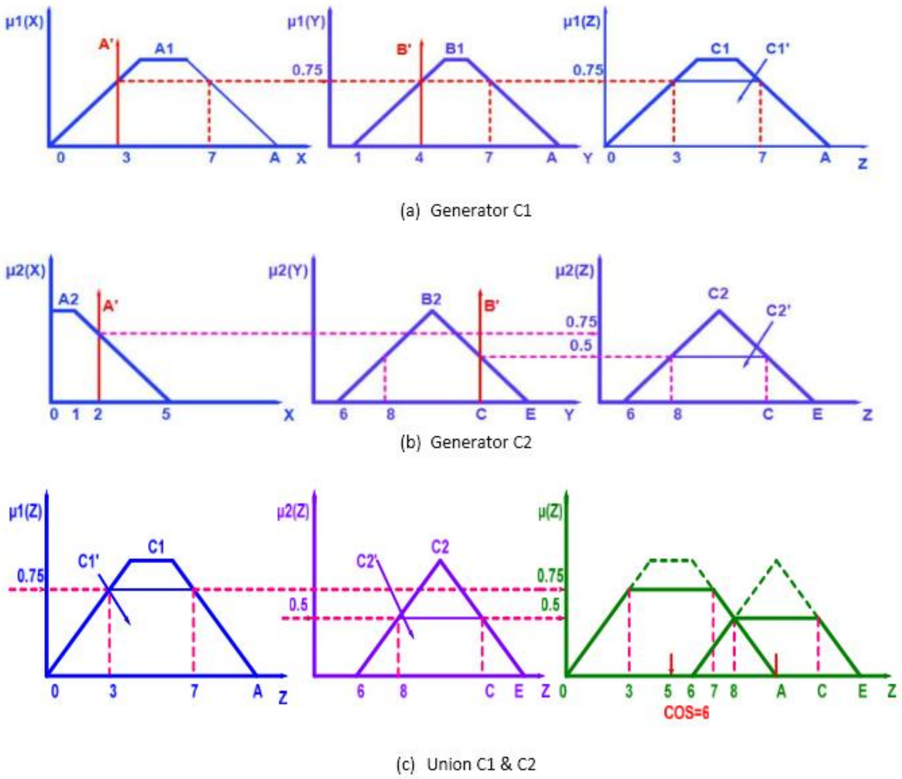
The defuzzified or crisp value for MODEL 3 can be calculated first manually and then using the proposed architecture. It is important that the two values must match; otherwise the proposed architecture is not well designed. In this study, Center of Sums (COS) is being used to calculate the defuzzified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

The calculations go as following

$$Z* = \frac{\int \bar{Z} \sum_{k=1}^{n} \mu c(z) dz}{\int \sum_{k=1}^{n} \mu c(z) dz} = \frac{\mathrm{CC1*AC1+CC2*AC2}}{\mathrm{AC1+AC2}} = \frac{\mathrm{TC1+TC2}}{\mathrm{AC1+AC2}} \quad \dots.(1)$$

Where CC1 and CC2 are the Center of trapezoid MFs C1 and C2 respectively, AC1 and AC2 are the Area of trapezoid MFs C1 and C2 respectively, also TC1 and TC2 are the Multiply of Area to Center of trapezoid MFs C1 and C2 respectively.

Model (3):



(a) Generator C1

(b) Generator C2

(c) Union C1 & C2

Figure 2: Fuzzification, Inference and Defuzzification using Center of Sums for Model 3

Hardware realization of the defuzzification process

In section III, the defuzzified values have been manually calculated using the Center of Sums, 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 COS technique has been designed and simulated. The proposed architecture has been modeled in Very High Description Language (VHDL) and has been implemented in field programmable gate array (FPGA). The defuzzified output in the COS method is given by the following equation 1. Since equation 1 carries summation in both the numerator and denominator, so two architectures have been developed for numerator and denominator separately and have been combined together. The output of the Center of Sums defuzzifier is given below as.

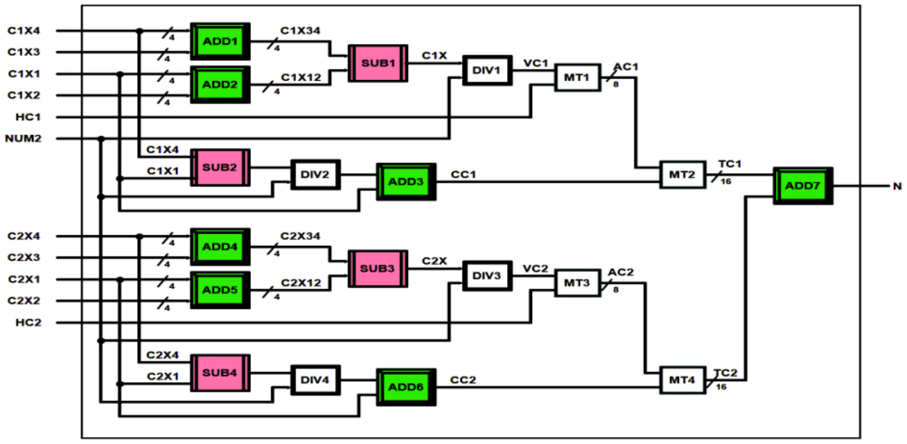


Figure 5: Architecture for numerator of COS defuzzifier

$$Z^* = \frac{CC1*AC1+CC2*AC2}{AC1+AC2} \quad ....(1)$$

Where CC1 and CC2 Center of C1 and C2 respectively also AC1 and AC2 Area of C1 and C2 respectively

$$CC1 = \frac{C1X4 - C1X1}{2} + C1X1, \quad CC2 = \frac{C2X4 - C2X1}{2} + C2X1$$

and

AC1 = HC1 * VC1, AC2 = HC2 * VC2

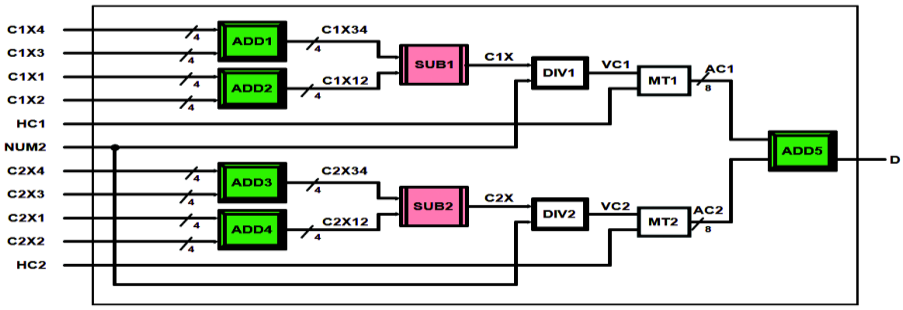


Figure 6: Architecture for denominator of COS defuzzifier

Figure 5 and Figure 6 are the architecture realization of the numerators and denominators of equation (2'). The two architectures are merged into a single architecture as shown in Figure 7. The Very High Description Language (VHDL) modeling of the proposed architectures have been performed. The functional analysis is shown in Figure 8. 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 6H, which is same as obtained manually for Figure 3. Similarly, for T-State second and third, the outputs of the defuzzifier is 7H and 6H respectively, which is same as calculated manually. This shows the proposed architecture is realizing the defuzzifier action efficiently and accurately.

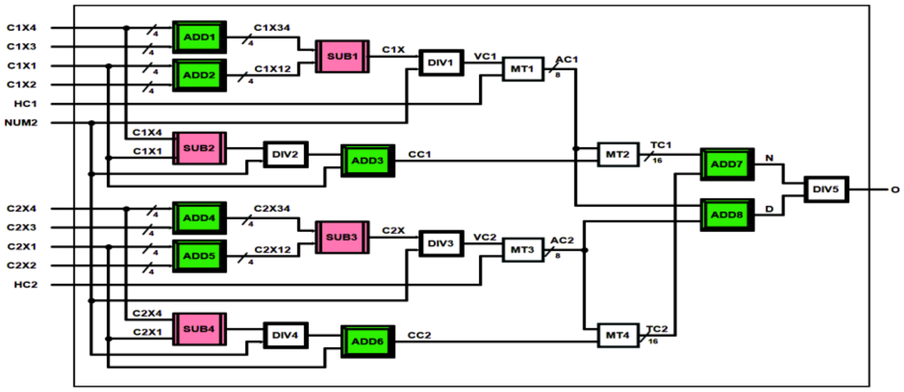


Figure 7: Proposed Architecture of the defuzzifier

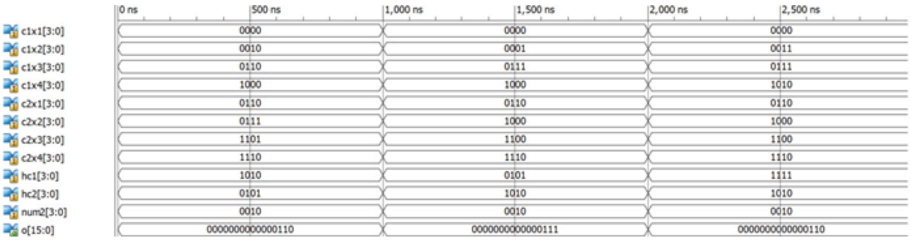


Figure 8: Timing diagram of the proposed defuzzifier

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 1, schematic and netlist generated of the proposed defuzzifier shown in Figure 9 and 10 respectively. The implementation results shown in Table 1 show that there is further scope for improvement in the proposed structures by incorporeity more parallelism in the architecture.

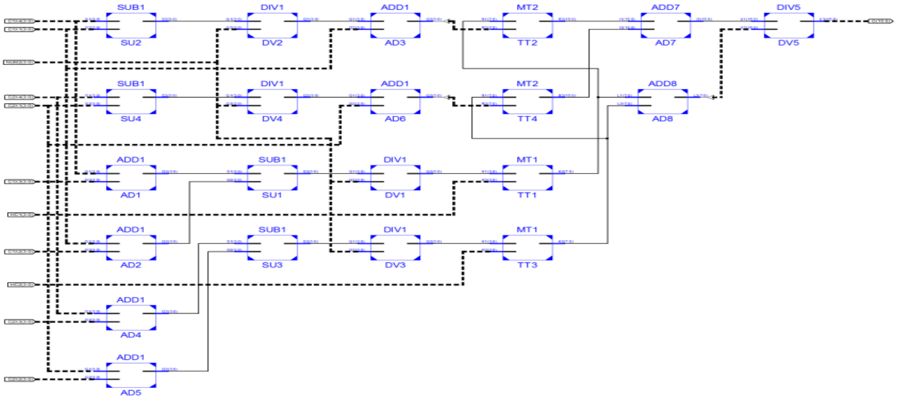


Figure 9: Schematic of the proposed defuzzifier

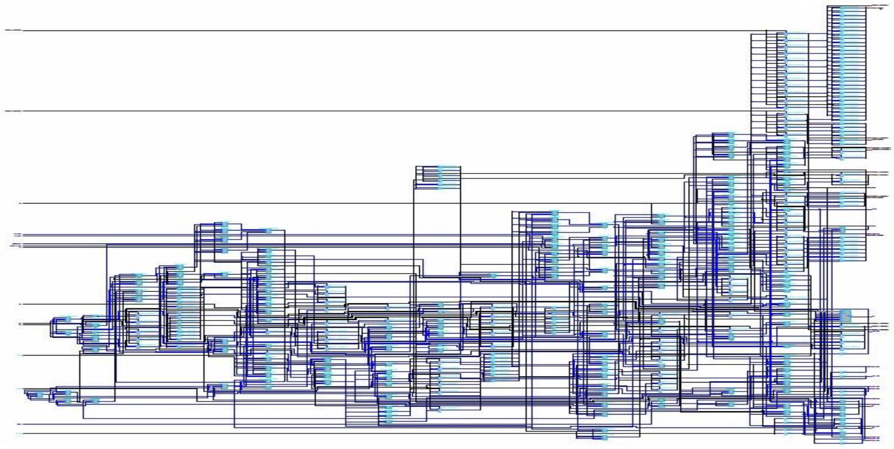


Figure 10: Netlist generated by the synthesis tools.

TABLE 1: FPGA Implementation results

Device Utilization Summary (FPGA: Spartan 3E XC3S100E)		
Logic Utilization	**Used**	**Available**
Number of 4 input LUTs	1,202	1,920
Number of occupied Slices	609	960
Number of Slices containing only related logic	609	609
Number of Slices containing unrelated logic	0	609
Total Number of 4 input LUTs	1,202	1,920
Number of bonded IOBs	60	66
Number of MULT 18X18SIOs	4	4
Average Fanout of Non-Clock Nets	3.21	

Conclusion

A Very Large Scale Integration (VLSI) Architecture of a defuzzifier is proposed. The proposed architecture is based on Center of Sums. The COS 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 FPGA implementation of the proposed architecture has been done. It has been seen that the proposed architecture realizes COS based defuzzifier efficiently, as there is a complete match between the results obtained manually and through the architecture.

APPENDIX

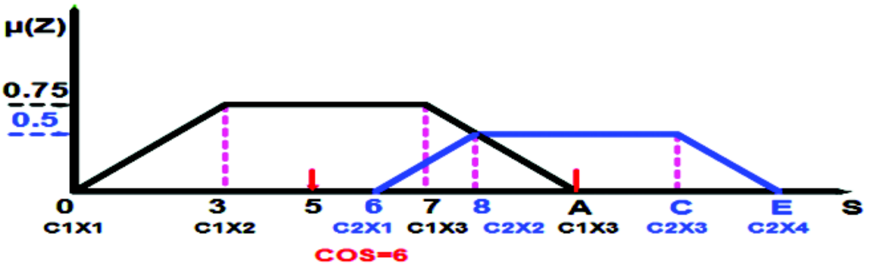


Figure 3:Defuzzification for MODEL 3

$$Z^* = \frac{\frac{CC1*hC1*[(C1X4-C1X1)+(C1X3-C1X2)]}{2}+\frac{CC2*hC2*[(C2X4-C2X1)+(C2X3-C2X2)]}{2}}{\frac{hC1*[(C1X4-C1X1)+(C1X3-C1X2)]}{2}+\frac{hC2*[(C2X4-C2X1)+(C2X3-C2X2)]}{2}}$$

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 = 0.75 = F H.

C1X1 = 0, C1X2 = 3, C1X3=7, C1X4=10

Center of C1 = CC1 = $\frac{C1X4 - C1X1}{2}$ + C1X1 = $\frac{10 - 0}{2}$ + 0 = 5.

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

C1X34 = C1X3 + C1X4 = 10 + 7 = 17, C1X12 = C1X1+ C1X2 = 0 + 3 = 3

C1X = C1X34 - C1X12 = 17 - 3 = 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 = 15 * 7 = 105.

Product of Area of C1 (AC1) to Center of C1 (CC1) = TC1.

TC1 = CC1 * AC1 = 5 * 105 = 525.

height of C2 = HC2 = 0.5 = A H.

C2X1 = 6, C2X2 = 8, C2X3=12, C1X4=14

Center of C2 = CC2 = $\frac{C2X4 - C2X1}{2}$ + C2X1 = = $\frac{14 - 6}{2}$ + 6 = 10.

Area of trapezoid = 0.5 × height × (L1 + L2)

Hence, L1 is longer than L2 in trapezoid C1 or C2.

Where L1 = (C2X4 – C2X1) and L2 = (C2X3 – C2X2) for trapezoid C2, therefore;

(L1 + L2) = [(C2X4 + C2X3) – (C2X1 + C2X2)] = [C2X34 – C2X12]

Meaning (L1 + L2) = C2X

C2X34 = C2X3 + C2X4 = 12 + 14 = 26, C2X12 = C2X1+ C2X2 = 6 + 8 = 14

C2X = C2X34 – C2X12 = 26 - 14 = 12.

VC2 = $\frac{C2X}{2}$ = $\frac{12}{2}$ = 6.

Area of trapezoid C2 = height × $\frac{(L1+L2)}{2}$ = HC2 × $\frac{C2X}{2}$ = HC2 × VC2

AC2 = HC2 * VC2 = 10 * 6 = 60.

Product of Area of C2 (AC2) to Center of C2 (CC2) = TC2.

TC2 = CC2 * AC2 = 10 * 60 = 600.

Numerator = N = TC1 + TC2 = 525 + 600 = 1125.

Denominator = D = AC1 + AC2 = 105 + 60 = 165.

O = $\frac{N}{D}$ = $\frac{1125}{165}$ = 6.

So the manually calculated defuzzified value is 6H. Another model of type Figure 2 has been designed and tested. The final defuzzification diagram of that model is given in Figure 4. Again COS is used to calculate the defuzzified value. The calculations have shown that the final crisp value as 7H.

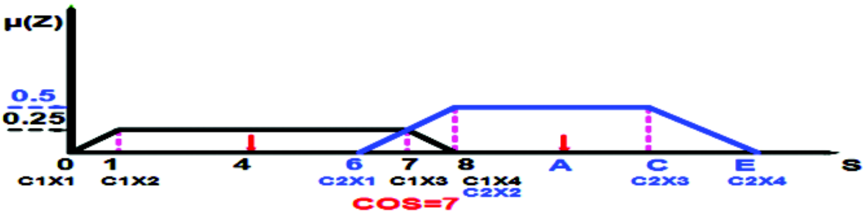


Figure 4: Defuzzification for MODEL 2

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.

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
C1X34 = C1X3 + C1X4 = 8 + 7 = 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.
Product of Area of C1 (AC1) to Center of C1 (CC1) = TC1.
TC1 = CC1 * AC1 = 4 * 35 = 140.
height of C2 = HC2 = 0.5 = A H.
C2X1 = 6, C2X2 = 8, C2X3=12, C1X4=14
Center of C2 = CC2 = $\frac{C2X4 - C2X1}{2}$ + C2X1 = = $\frac{14-6}{2}$ + 6 = 10.
Area of trapezoid = 0.5 × height × (L1 + L2)

Hence, L1 is longer than L2 in trapezoid C1 or C2.
Where L1 = (C2X4 – C2X1) and L2 = (C2X3 – C2X2) for trapezoid C2, therefore;
(L1 + L2) = [(C2X4 + C2X3) – (C2X1 + C2X2)] = [C2X34 – C2X12]
Meaning (L1 + L2) = C2X
C2X34 = C2X3 + C2X4 = 12 + 14 = 26, C2X12 = C2X1+ C2X2 = 6 + 8 = 14
C2X = C2X34 – C2X12 = 26 - 14 = 12.
VC2 = $\frac{C2X}{2} = \frac{12}{2}$ = 6.
Area of trapezoid C2 = height × $\frac{(L1+L2)}{2}$ = HC2 × $\frac{C2X}{2}$ = HC2 × VC2
AC2 = HC2 * VC2 = 10 * 6 = 60.
Product of Area of C2 (AC2) to Center of C2 (CC2) = TC2.
TC2 = CC2 * AC2 = 10 * 60 = 600.
Numerator = N = TC1 + TC2 = 140 + 600 = 740.
Denominator = D = AC1 + AC2 = 35 + 60 = 95.
O = $\frac{N}{D} = \frac{740}{95}$ = 7.
The same procedure has been used for other defuzzified model, not shown in a figure. The defuzzified value obtained is 6H.

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] A. M. Murshid, S. A. Loan, S. A. Abbasi, and A. M. Alamoud, “A Novel Multi Membership Function Based VLSI Architecture of a fuzzy inference processor," *Int. Journal of Fuzzy Systems*, Vol. 16, No. 4, pp. 468-482, Dec. 2014.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] S. N. Sivanandam, S. Sumathi, and S. N. Deepa, "Introduction to Fuzzy Logic using MATLAB", by Springer, 2007.

مركز الجموع اعتمادا على وحدة فك الضبابية بمعمارية دوائر متكاملة ذات كثافة عالية جداً

عاصم مجيد مرشد

قسم علوم الحاسوب ، كلية العلوم ، جامعة كركوك ، كركوك ، العراق

الملخص

فك الضبابية هو عملية الحصول على قيمة واضحة من البيانات الغامضة. لأن البيانات الغامضة لا يمكن استخدامها مباشرة للتطبيقات في الوقت الحقيقي، وبالتالي فإنه لا بد من تحويلها إلى قيمة واضحة في تصميم وحدة فك الضبابية. في هذا العمل تم استخدم طريقة مركز الجموع لفك الضبابية ، بسبب الطبيعة الفعالة حسابيا. و طريقة مركز الجموع من فك الضبابية هي أسرع من بقية طرق فك الضبابية وهي ليست مقيدة على عضوات الدوال المتناظرة فقط وهي طريقة بسيطة ويتم استخدامه بشكل أوسع بالمقارنة إلى طريقة مركز الثقل لفك الضبابية (تسمى أيضا طريقة مركز المساحة أو الطريقة المركزية) الأكثر تعقيدا من طريقة مركز الجموع لفك الضبابية. تم كتابة وتنفيذ كودات البرنامج بطريقة تنفيذ الدوائر الالكترونية عالية الدقة المعروفة (VHDL) وتنفيذها على (XILINX) والجهاز المستخدم لهذا الغرض (Spartan) في مجموعة بوابة حقل برنامج (FPGA). المعمارية المقترحة هو أكثر كفاءة في المساحة (المساحة المستخدمة لتنفيذ المعمارية المقترحة) وسرعة العملية بالمقارنة مع معمارية أكثر تعقيداً المستخدمة في طريقة مركز الثقل. وقد كشف التحليل الوظيفي أن المعمارية المقترحة بتنفيذ COS لفك الضبابية ذات كفاءة ودقة.

الكلمات الدالة: فك الضبابية ؛ عمليات الضبابية ؛ طريقة مركز الجموع ؛قدرة واطئة؛ تصميم VLSI