# THE IMPLEMENTATION OF FUZZY FLIP-FLOPS AS MEMORY MODULES

Hamdi Atmaca H. Serhan Yavuz

Osmangazi University
Electrical and Electronics Engineering Department
Tel : +90-222-239 28 40
e-mail : hatmaca@ogu.edu.tr

**Abstract**

In this paper, a fuzzy flip-flop concept is introduced according to the fuzzy logic operations. Characteristic equations of the flip-flops have been derived by extending binary logic flip-flop concept and making some orientations depending on the fuzzy logic rules. JK flip-flop is a common type of flip-flops and other types may be produced by modifying the equations of the JK flip-flop. Under these concepts, the characteristic equations of all types of flip-flops have been derived by applying the same method used for the JK flip-flop. Computer simulations of each type are made using ideal OP-AMPs.

## 1. Introduction

A flip-flop is a circuit that uses logic gates and feedback to produce a bistable device that can memorize single bits of information. It is the basic building block for registers and memory. In binary logic, the most common type of electronic memory device is the "flip-flop". There are four common types of flip-flops so called JK, RS, D and T type flip-flops. An array of *k* flip-flops can serve to register or remember a *k* bit binary word.

Similar concept is valid for fuzzy flip-flop [1]. A fuzzy flip-flop can be viewed as an extension of the binary flip-flop using fuzzy gates instead of binary gates. After the representation of fuzzy logic concept, a great deal of research has been directed towards realization of fuzzy computers. Because, a fuzzy computer operation is similar to the human thinking. A few types of fuzzy processors [2,3,4,5,6] which perform fuzzy operations can be realized and tested. However, in order to realize multi-stage fuzzy inference, (i.e. fuzzy computer,) fuzzy memory modules are indispensable. For an ordinary computer, a binary flip-flop is used as a fundamental element of memory modules. In the case of a fuzzy computer, fuzzy flip-flops are used to construct the memory modules of the device [7,8].

## 2. The Concept of Fuzzy Logic

Fuzzy logic is based on concept of fuzzy sets. Therefore, the value of any variable must be in the range [0, 1]. So, the truth value of a formula can assume any value in the interval [0, 1]. Basic properties of fuzzy logic operations are given by the following expressions [9, 10]:

(1) $T(S) = T(A)$ if $S = A$.
(2) $T(S) = 1 - T(A)$ if $S = \bar{A}$.
(3) $T(S) = \max[T(S_1), T(S_2)]$ if $S_1 = S_1 + S_2$.
(4) $T(S) = \min[T(S_1), T(S_2)]$ if $S_1 = S_1 S_2$.

where T(S) denotes the truth value of a formula S.

Another point here to remember is that two-valued logic is a special case of fuzzy logic; all the rules shown above are applicable for two-valued, binary logic.

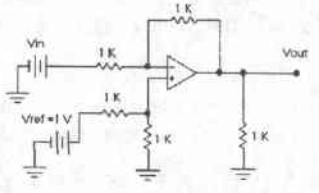
## 3. Fuzzy Algebra

In fuzzy algebra, the concepts of fuzzy sets are used. Fuzzy variables are the main elements of fuzzy algebra. So, "fuzzy variable" is replaced by the term "membership grade" of a fuzzy variable in a set [11].

In Boolean algebra, the specific conditions which are $A \cap \bar{A} = 0$ and $A \cup \bar{A} = 1$ always hold true; however, in fuzzy logic, none of these rules always holds true, since in fuzzy logic, the corresponding operations will be $A \cap \bar{A} = \min[A, (1-A)]$ and $A \cup \bar{A} = \max[A, (1-A)]$.

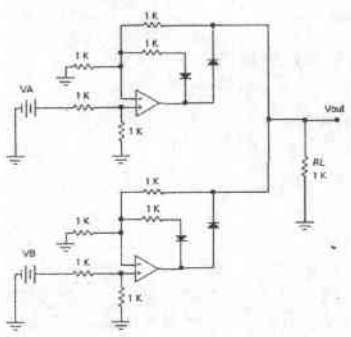
## 4. Fuzzy Logic Gates

As in binary logic, the basic circuits for fuzzy logic are the standard AND, OR, and INVERTER gates. More complex devices can be constructed by different compositions of these gates. Using the three basic gates, any sophisticated circuit can be achieved. Assuming ideal op-amps, the hardware implementations of fuzzy AND, fuzzy OR, and fuzzy INVERTER gates are proposed in Figures 1, 2, and 3.

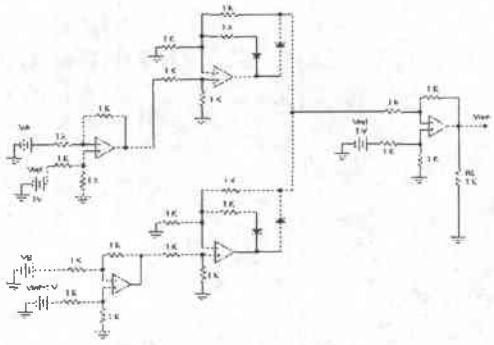



$V_{out} = 1 - V_{in}$

**Figure 1.** Fuzzy Logic Inverter Circuit



$V_{out} = \max[V_A, V_B]$.
**Figure 2.** Fuzzy Logic OR Circuit



$V_{out} = \min[V_A, V_B]$.
**Figure 3.** Fuzzy Logic AND Circuit

## 5. Fuzzy Flip-Flops

A flip-flop is the most common type of electronic memory device. It is the basic building block for registers where binary information is stored. The so-called JK flip-flop is a common type in binary logic implementations; other flip-flop types may be produced by modifying a JK flip-flop.

A similar concept can be applied for fuzzy logic [12]. A fuzzy flip-flop can viewed as an extension of the binary flip-flop with fuzzy gates.

Four types of fuzzy flip-flops are discussed in this paper deriving equations for JK, RS, D and T type flip-flops. The JK flip-flop is the basic type of them and the equations for other types can be derived similarly from the equation of the JK flip-flop.

### 5.1. JK Flip-Flop

**Table 1.** Truth Table for binary logic JK Flip-Flop.

| J(t) | K(t) | Q(t) | Q(t+1) |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

From minterm expression for the truth table shown above in Table 1, the state equation is given as follows:

$$Q(t+1) = J\overline{Q} + \overline{K}Q \qquad [5.1]$$

Maxterm Expression for the next state is similarly shown below:

$$Q(t+1) = (J+Q)(\overline{K}+\overline{Q}) \qquad [5.2]$$

As shown above, the binary JK flip-flop can be implemented either as the sum of products; or as the product of sums. Both of these equations in binary logic gives the same result. This can be proved by using De Morgan's Law in the following manner:

$$(A\cap B)\cup(\bar{A}\cap C)\cup(B\cap C)=(A\cap B)\cup(\bar{A}\cap C) \qquad [5.3]$$

$$(A\cup B)\cap(\bar{A}\cup C)\cap(B\cup C)=(A\cup B)\cap(\bar{A}\cup C) \qquad [5.4]$$

But these two equations ([5.1] and [5.2]) can be fuzzified in two different manners. The first one is called as *reset type* while the second type is called as the *set type* fuzzy JK Flip-Flop. The reset type equation implies the maximum of minimum terms and the set type equation implies the minimum of maximum terms.

Equation [5.1] can be fuzzified as

$$Q_R(t+1) = \{J \wedge (1- Q)\} \vee \{(1-K) \wedge Q\} \qquad [5.5]$$

Similarly, equation [5.2] can also be fuzzified as

$$Q_S(t+1) = \{JVQ\} \wedge \{(1-K) \vee (1-Q)\} \qquad [5.6]$$

It is the fact that $Q_R(t+1) = Q_S(t+1)$ does not always hold true according to fuzzy logic operations. Some cases can be investigated to give the difference between $Q_R(t+1)$ and $Q_S(t+1)$:

*Case 1)* If $J<Q<0.5<K$

$Q_R(t+1) = \{J \wedge (1- Q)\} \vee \{(1-K) \wedge Q\}$
$Q_R(t+1) = J \vee \{(1-K) \wedge Q\}$
$Q_R(t+1) = \{J \vee (1-K)\} \wedge \{J \vee Q\}$
$Q_R(t+1) = \{J \vee (1-K)\} \wedge Q$

$Q_S(t+1) = \{JVQ\} \wedge \{(1-K) \vee (1-Q)\}$
$Q_S(t+1) = Q \wedge \{(1-Q)\}$
$Q_S(t+1) = Q$

Here, $\max(J,\{1-K\})$ may be greater than Q or less than Q. If it is less than Q, $Q_R(t+1)=\max(J,\{1-K\})$ which is less than Q. If it is greater than Q, $Q_R(t+1)=Q$, $Q_S(t+1)=Q$. Then $Q_R(t+1) \le Q_S(t+1)$ is valid.

*Case 2)* If $J<0.5<Q<K$

$Q_R(t+1) = \{J \wedge (1- Q)\} \vee \{(1-K) \wedge Q\}$
$Q_R(t+1) = \{J \wedge (1- Q)\} \vee \{(1-K)\}$
$Q_R(t+1) = \{J \vee (1-K)\} \wedge \{(1-K) \vee (1-Q)\}$

$Q_S(t+1) = \{JVQ\} \wedge \{(1-K) \vee (1-Q)\}$

$Q_S(t+1) = Q \wedge \{(1-Q)\}$
$Q_S(t+1) = (1-Q)$
Similarly, J V (1-K) may be less than {1-Q}, so $Q_R(t+1) \leq Q_S(t+1)$ is valid.

In the most of the cases the equality $Q_R(t+1) = Q_S(t+1)$ can be caught; but the two cases written above prove that $Q_R(t+1)$ is not always equal to $Q_S(t+1)$. Therefore, it should be given that $Q_R(t+1) \leq Q_S(t+1)$.

However, the values of Q(t+1)s of both type are equal to each other at the bounds of J's and K's. That is when J=K is valid, $Q_R(t+1)$ is always equal to $Q_S(t+1)$ as shown below:

*Case 3)* If J=K<Q
$Q_R(t+1) = \{J \wedge (1-Q)\} V \{(1-K) \wedge Q\}$
$Q_R(t+1) = \{J \wedge (1-Q)\} V \{(1-J) \wedge Q\}$
Here $Q > J \geq J \wedge (1-Q)$, so $(1-J) \wedge Q > J \wedge (1-Q)$

$Q_S(t+1) = \{JVQ\} \wedge \{(1-K) V (1-Q)\}$
$Q_S(t+1) = Q \wedge \{(1-J) V (1-Q)\}$
$Q_S(t+1) = \{Q \wedge (1-J)\} V \{Q \wedge (1-Q)\}$
Similarly $Q \wedge (1-J) \geq Q \wedge (1-Q)$,
Then,

$$Q_R(t+1) = Q_S(t+1)$$

*Case 4)* If J=K=Q
$Q_R(t+1) = \{J \wedge (1-Q)\} V \{(1-K) \wedge Q\}$
$Q_R(t+1) = \{J \wedge (1-J)\} V \{(1-J) \wedge J\}$
$Q_R(t+1) = J \wedge (1-J)$

$Q_S(t+1) = \{JVQ\} \wedge \{(1-K) V (1-Q)\}$
$Q_S(t+1) = J \wedge \{(1-J) V (1-J)\}$
$Q_S(t+1) = J \wedge (1-J)$

It is obvious that $Q_R(t+1) = Q_S(t+1)$.

*Case 5)* If J=K>Q
$Q_R(t+1) = \{J \wedge (1-Q)\} V \{(1-K) \wedge Q\}$
$Q_R(t+1) = \{J \wedge (1-Q)\} V \{(1-J) \wedge Q\}$
Since $J > Q \geq (1-J) \wedge Q$,
$Q_R(t+1) = J \wedge (1-Q)$

$Q_S(t+1) = \{JVQ\} \wedge \{(1-K) V (1-Q)\}$
$Q_S(t+1) = J \wedge \{(1-J) V (1-Q)\}$
$Q_S(t+1) = \{J \wedge (1-J)\} V \{J \wedge (1-Q)\}$
Similarly $J \wedge (1-Q) \geq J \wedge (1-J)$,
Then,

$$Q_R(t+1) = Q_S(t+1).$$

As a conclusion, if the condition J=K is valid, $Q_R(t+1)$ is always equal to $Q_S(t+1)$.

It is proved that $Q_R(t+1) \leq Q_S(t+1)$ holds always true. Therefore, a single equation for characteristics of both reset and set type flip-flops can be derived as follows:

$$Q(t+1) = \{J V (1-K)\} \wedge (J V Q) \wedge \{(1-K) V (1-Q)\} \quad [5.7]$$

The last equation shown above is the fundamental equation of a JK flip-flop which has characteristics of both *set* and *reset-types*.

The realization of a JK flip-flop using the fuzzy AND, OR, and NOT gates is given in Figure 4.

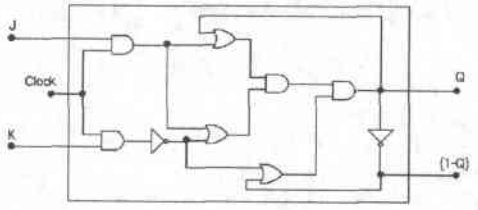



**Figure 4.** Implementation of Fuzzy JK Flip-Flop

### 5.2. RS Flip-Flop

**Table 2.** Truth table for RS Flip-Flop

| R(t) | S(t) | Q(t) | Q(t+1) |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | ? |
| 1 | 1 | 1 | ? |

The minterm expression for the next state output of the truth table shown in Table 2 is given below:

$$Q(t+1) = S + R\overline{Q} \quad [5.8]$$

Similarly maxterm expression for Q(t+1) is shown below:

$$Q(t+1) = (S+Q)(\overline{R} + \overline{Q}) \quad [5.9]$$

The fuzzified equations are derived as

$$Q_R(t+1) = \{S\} V \{(1-R) \wedge Q\} \quad [5.10]$$

$$Q_S(t+1) = \{S V Q\} \wedge \{(1-R) V (1-Q)\} \quad [5.11]$$

In a similar manner, $Q_R(t+1)$ is not always equal to $Q_S(t+1)$. For example:

*Case 1)* If S=R<Q
$Q_R(t+1) = S V \{(1-R) \wedge Q\}$
$Q_R(t+1) = \{S V (1-S)\} \wedge \{S V Q\}$
$Q_R(t+1) = \{S V (1-S)\} \wedge Q$

$Q_S(t+1) = \{SVQ\} \wedge \{(1-R) V (1-Q)\}$
$Q_S(t+1) = Q \wedge \{(1-S) V (1-Q)\}$
If S=R<Q<0.5, Then, $Q_R(t+1)=Q$ $Q_S(t+1)=Q$.
If 0.5<S<Q, Then, $Q_R(t+1)=S$ $Q_S(t+1)=1-S$. So,

$$Q_R(t+1) \neq Q_S(t+1)$$

*Case 2)* If S=R=Q

$Q_R(t+1) = S \vee \{(1-R) \wedge Q\}$
$Q_R(t+1) = S \vee \{(1-S) \wedge S\}$
$Q_R(t+1) = S$

$Q_S(t+1) = \{SVQ\} \wedge \{(1-R) \vee (1-Q)\}$
$Q_S(t+1) = S \wedge (1-S)$
If S>0.5, Then, $Q_S(t+1)=(1-S)$, $Q_R(t+1)=S$.

$$Q_R(t+1) \neq Q_S(t+1)$$

*Case 3)* If S=R>Q

$Q_R(t+1) = S \vee \{(1-R) \wedge Q\}$
$Q_R(t+1) = S \vee \{(1-S) \wedge Q\}$
$Q_R(t+1) = \{S \vee (1-S)\} \wedge \{S \vee Q\}$
$Q_R(t+1) = \{S \vee (1-S)\} \wedge S$
$Q_R(t+1) = S \vee \{S \wedge (1-S)\}$
$Q_R(t+1) = S$

$Q_S(t+1) = \{SVQ\} \wedge \{(1-R) \vee (1-Q)\}$
$Q_S(t+1) = S \wedge \{(1-S) \vee (1-Q)\}$
If S<0.5, Then, $Q_S(t+1)= S \wedge (1-Q)=S$.
If S<0.5 and Q<0.5, Then, $Q_S(t+1)=1-Q$ and $Q_R(t+1) = S$. So,

$$Q_R(t+1) \neq Q_S(t+1)$$

A general case can be found to determine the characteristic equation of RS flip-flop given in the following equation;

$$Q(t+1) = \{S \vee (1-R)\} \wedge (S \vee Q) \wedge \{(1-R) \vee (1-Q)\} \quad [5.12]$$

The hardware implementation of this fuzzy RS flip-flop using the fuzzy AND, OR, and NOT gates is shown in Figure 5.

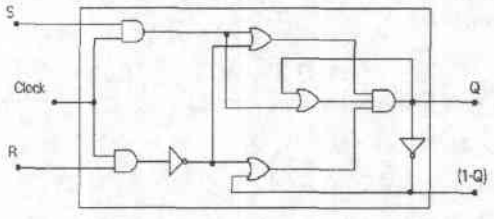



**Figure 5.** Implementation of Fuzzy RS Flip-Flop

### 5.2. *D Flip-Flop*

**Table 3.** Truth Table for D Flip-Flop

| D(t) | Q(t) | Q(t+1) |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Minterm expression for this flip-flop is given below from Table 3.

$$Q(t+1) = D\overline{Q} + DQ \quad [5.13]$$

A similar maxterm expression is shown as follows:

$$Q(t+1) = (D + Q)(D + \overline{Q}) \quad [5.14]$$

The characteristic equation of D flip-flop is found by first extending the maxterm expression using De Morgan's Theorem, then this equation is fuzzified as

$$Q(t+1) = \{D \vee Q\} \wedge \{D \vee (1-Q)\} \wedge D \quad [5.15]$$

The realization of D flip-flop using the fuzzy AND, OR, and NOT gates is given in Figure 6.

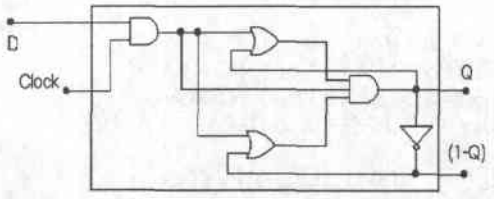



**Figure 6.** Implementation of Fuzzy D Flip-Flop

### 5.3. T Flip-Flop

**Table 4.** Truth Table for T Flip-Flop

| T(t) | Q(t) | Q(t+1) |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Minterm expression is given below:

$$Q(t+1) = T\overline{Q} + \overline{T}Q \quad [5.16]$$

Maxterm expression is shown as follows:

$$Q(t+1) = (T + Q)(\overline{T} + \overline{Q}) \quad [5.17]$$

And the characteristic equation of T flip-flop is given by the following expression:

$$Q(t+1) = \{T \vee Q\} \wedge \{(1-T) \vee (1-Q)\} \wedge \{(1-T) \wedge T\} \quad [5.18]$$

The hardware implementation of this T flip-flop using the fuzzy AND, OR, and NOT gates is given in Figure 7.

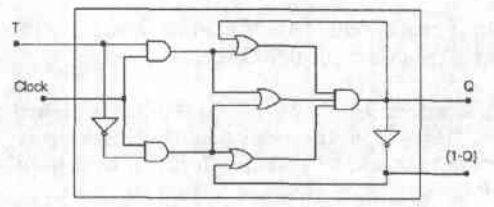



**Figure 7.** Implementation of Fuzzy T Flip-Flop

## 6. Conclusion

A fuzzy flip-flop concept has been proposed in this paper. Flip-flop is the basic memory device, so, fuzzy flip-flops may play an important role for implementation of fuzzy computers. Since a flip-flop can memorize a single bit of information, the memory modules of the fuzzy computer may include various types of flip-flops. In this study, the characteristic equations of fuzzy flip-flops are determined according to the rules of fuzzy logic. The whole study contains all possible cases of the flip-flop equations in theory and it compares the properties of crisp logic and fuzzy logic. It also gives an alternative approach for hardware implementations of fuzzy AND, OR, and INVERTER gates. The hardware implementations of these flip-flops can be made using FET types of operational amplifiers.

## References

[1] K. Hirota and K. Ozawa, "The Concept of Fuzzy Flip-Flop", IEEE Trans. Systems, Man and Cybernetics Vol.19, No:5, pp.980-997, Oct. 1989.

[2] T.Yamakawa and T.Miki, "The Current Mode Fuzzy Logic Integrated Circuits Fabricated by the Standard CMOS Process", IEEE Trans. Comp. Vol.C35, pp.161-167, Feb. 1986.

[3] T.Yamakawa, "High Speed Fuzzy Controller Hardware System", Proc 2nd Fuzzy Systems Symp, Tokyo, Japan, pp.122-130, 1986.

[4] M. Togai and H.Watanabe, "A VLSI Implementation of Fuzzy Inference Engine Toward and Expert System on a Chip", Inform. Sci. Vol. 38, pp.147-163, 1986.

[5] P.N. Marinos, "Fuzzy Logic and its Application to Switching Systems", IEEE Trans. Comput., Vol C.18, No:4, pp. 343-348, April 1969.

[6] K. Shimizu, M.Osumi, and F. Imae, "Digital Fuzzy Processor FP-5000", Proc. 2nd Int. Conf. Fuzzy Logic, Neural Networks, Iizuka, Japan, pp.539-542, July 1992.

[7] T. Yamakawa, T.Inoue, F.Ueno and Y. Shiran, "Implementations of Fuzzy Logic Hardware Systems: Three Fundamental Arithmetic Circuits", Trans. Inst. Electronic Commun. Eng., Japan, Vol. J63C, No:10, pp. 720-721, 10. Oct. 1980.

[8] T. Miki, H. Matsumoto, K. Ohto and T. Yamakawa, "Silicon Implementation for a Novel High Speed Fuzzy Inference Engine: Mega-FLIPS Analog Fuzzy Processor", J. Intell. Fuzzy Syst. Vol. No:1, pp. 27-42, 1993.

[9] Dubois, D. and H. Prade, "Fuzzy Sets and Systems: Theory and Applications", Academic Press, New York, 1980.

[10] Gupta, M. and J.Qi, "Theory of T Norms and Fuzzy Inference Methods", Fuzzy Sets and Systems, Vol.40, pp. 431-450, 1991.

[11] Klir, G. and T. Folger, "Fuzzy Sets, Uncertainity and Information", Prentice Hall, Englewood Clifts, N.J., 1988.

[12] Yager, R.R. "On the Implication Operator in Fuzzy Logic", Inform Sci, Vol.31, pp. 141-164, 1983.