Defence Science Journal, Vol 45, No 4, October 1995, pp 325-332


# Fuzzy Logic and VLSI Testing

M.V. Atre and D. Krishna Kumar

*Advanced Numerical Research & Analysis Group, Hyderabad-500 258*

**ABSTRACT**

A new application of Fuzzy logic (FL), in the context of test vector generation in VLSI testing is presented. Fuzzification of the threshold value simulation (TVS) approach and setting up of mathematical concepts are carried out in terms of a hierarchy of membership functions. The test-vectors are found by optimising a suitable membership function. The Fuzzy model besides giving a different mathematical basis, also helps in defining new and better optimising functions, thus proving its utility. The concepts outlined in this paper, though demonstrated on toy model of a circuit consisting of only AND gates, can easily be extended to circuits with other logic gates.

## 1. INTRODUCTION

The advances in VLSI technology have brought forward integrated circuits of extremely high complexities. Both, the hardware and the software have become increasingly complex with the aim of designing and manufacturing smaller chips with higher gatecounts.

One of the key issues, as important as the design and manufacture of chips, is VLSI testing where the aim is that of developing sophisticated testing software and methodologies towards ensuring high reliability of chips. This involves widely different areas, such as efficient testability analysis, fault modelling, fault simulators, test vector generation packages, automatic test equipment, etc. These topics have been discussed in literature in detail[1,2]. This paper concentrates on some aspects of test vector generation (TVG).

The main aim of TVG is to find a set of inputs (called test vectors) to a given circuit so as to detect as many manufacturing faults as possible. A test-vector gives different outputs for a faulty and a fault-free circuit. This involves: (i) Deciding the levels at which faults need to be detected; e.g. functional, gate-level, transistor-level, PLA level, etc., with gate-level fault-modelling being the most practical and useful. (ii) Modelling the faults, with logical single stuck-at faults being the most studied; (iii) Fault-simulators for finding all the faults detected by a given test-vector; single, deductive, concurrent, etc; and (iv) A TVG package which selectively chooses a test-vector from the space of all possible input vectors to a circuit. (A circuit requiring m-bit string input has $2^m$ possible different inputs which is usually a very large number).

Many different techniques of TVG have been investigated and implemented[3,4]. This paper concentrates on a directed search method developed by Cheng and Agrawal[5] for combinational circuits. The threshold characteristics of the various gates in the circuit are modified such that, for any input, the output is a continuous number between 0 and 1. The outputs are calculated for both fault-free and faulty cases (some stuck-at fault in the circuit is assumed) and the difference $\Delta$ is used to define a cost function which represents the distance of the input from a test-vector and thus helps in a directed search. The entire analysis is done via simulation and the method is called threshold value simulation (TVS) of TVG. The important difference between this method and other methods is that TVS deals with circuit inputs and outputs which are not integers but continuous real values between 0 and 1. The entire analysis is carried out in terms of the

Received 25 January 1994, revised 12 August 1994

continuous variable, $\Delta$ which indicates how close or far the input is from a test-vector.

Fuzzy sets (FS) is precisely that branch of mathematics which deals with the analysis of qualitative concepts, such as close, far, tall, short, etc. Many excellent texts exist on the subject[6-7], including a collection of fundamental papers by L.A. Zadeh[8] (the founder of FS). In this paper, it is suggested that the 'closeness' of an input from a test-vector can be reformulated in the language of FS, as also the analysis of any circuit with logical components (AND, NAND,etc.) in terms of Fuzzy logic (FL). Thus FS and FL are proposed as the ideal mathematical frameworks for describing the TVS method of TVG.

A brief description of the TVS method of TVG, along with an example, is given in Sec. 2. In Sec. 3, a brief outline of the areas in TVS where Fuzzy concepts can be applied is given. The complete mathematical formulation of the TVS method in terms of FL and FS, along with the general description of the optimisation problem is then given in Sec. 4. The implementation of the ideas of FL and FS are carried out in Sec. 5 for a toy model to show the utility of the method. Discussion and conclusions are presented in Sec. 6.

## 2. TVS & TVG

The TVS method of TVG has been presented in detail elsewhere[5]. Here the basic ideas are briefly presented and illustrated using a simple example.

### 2.1 Brief Description of the TVS

(a) A threshold value (TV) function is given for each gate whereby the output is calculated for each gate, given the input. The output is a continuous variable taking values between 0 and 1, i.e. it is not a binary variable. An example of the TV function for an AND gate is given in Fig. 1. The TV functions for other gates can be found elsewhere[5].

(b) For an input $x_g$, the output $y_g$ is obtained as follows: Define a mean input $x_g^{mean}$ to the gate as

$$x_g^{mean} = \frac{1}{n} \sum_{\text{all input lines}} x_g, \quad 0 \leq x_g \leq 1 \tag{1}$$

i.e., $x_g^{mean}$ is the sum of all the inputs from different lines divided by the number of input lines. Then the output is

$$y_g(x_g) = T_g(x_g^{mean}), \quad 0 \leq y_g \leq 1 \tag{2}$$

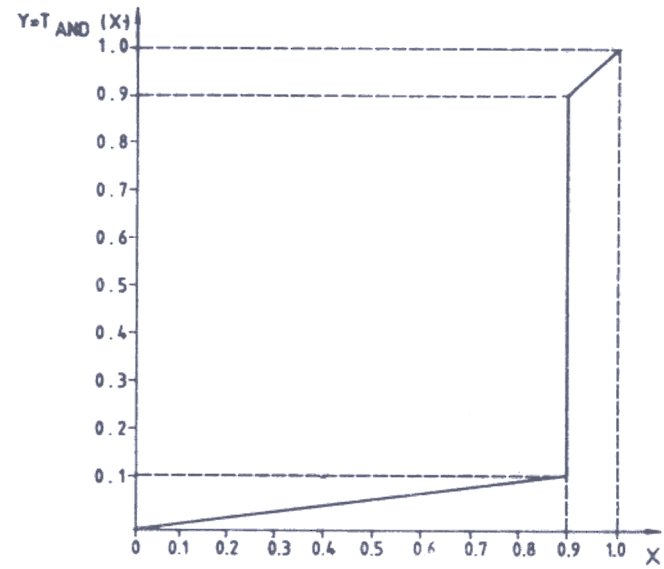


**Figure 1. Threshold value function for an AND gate.**

Where $T_g$ is the TV function for the gate $g$. The threshold function $T_g$ will vary from gate to gate.

Thus nonbinary values as gate outputs are obtained, which ultimately yield nonbinary primary outputs, thus helping in defining cost functions suitable for optimisation.

(c) Proceeding from primary inputs, the outputs of all gates at level 1 are obtained, which then serve as inputs of gates at next level, and so on, till we arrive at the primary output of the circuit, $y_c$. At each level, Eqn (2) is used to proceed to the next level. Essentially,

$$y_c(x_c) = T_g, T_g, T_g \ldots . T_g[(x_{pg}^{mean})] \tag{3}$$

is the primary output of the circuit for primary input $x_c$, *where* $X_c$ is the set of all primary inputs to circuit $C$, $x_{p,g}^{mean}$ is the mean primary input to gate $g$ as calculated from Eqn (1) and denotes the composition law.

(d) Now the entire analysis is redone by assuming the existence of a fault in the circuit, say some logical stuck-at fault $f$ on some line connecting two gates. This yields another output $y_c^f$ which may or may not be different from $y_c$ (of the fault-free circuit). Define

$$\Delta^f\{y_c(x_c)\} = |y_c^f(x_c) - y_c(x_c)| \tag{4}$$

where $\Delta^f$ is the difference between the faulty and the fault-free outputs and $y_c$ is a function of $x_c$. This is interpreted as a measure of how well $x_c$ (as input test-vector) can distinguish between a fault-free and faulty circuits.

For a circuit with many primary outputs, the minimum of the entire set of $\Delta_i^f$ (for various outputs) is chosen, i.e.

$$\Delta^f = \min_i \{\Delta_i^f\} \tag{5}$$

(e) A cost function $C^f(\Delta^f)$ is defined for the circuit as

$$C^f\{y_c(x_c)\} = 1/\Delta^f(y_c(x_c)) \tag{6}$$

According to Cheng and Agrawal[5], if $C^f < C_{th}$ (where $C_{th}$ is a suitably defined threshold cost), the input $x_c$ to the circuit is said to be a test-vector for finding fault $f$.

$C_{th}$ is determined as follows. If the difference $\Delta^f$ between $y_c$ and $y_c^f$ is large enough to be unambiguously detected by the threshold function $T_g$, then $x_c$ is suitable to detect the assumed fault in the circuit, i.e. $x_c$ is a test-vector. Hence there is a need for a threshold $\Delta_{th}^f$ such that $\Delta^f > \Delta_{th}$, where $\Delta_{th}$ is obtained for the TV function. $C_{th}$ is then determined from $\Delta_{th}$ by Eqn (6).

(f) For any two inputs $x_c^1$ and $x_c^2$ which are not test-vectors (i.e., $C^f(x_c^1)$ and $C^f(x_c^2)$ both $> C_{th}$), $x_c^1$ is said to be better than $x_c^2$ if $C^f(x_c^1) < C^f(x_c^2)$. This helps in setting up a directed search around $x_c^1$ and not $x_c^2$.

*Example*

The above ideas are illustrated for a simple three AND gates-circuit shown in Fig. 2. The lines are labelled by alphabets A to G.

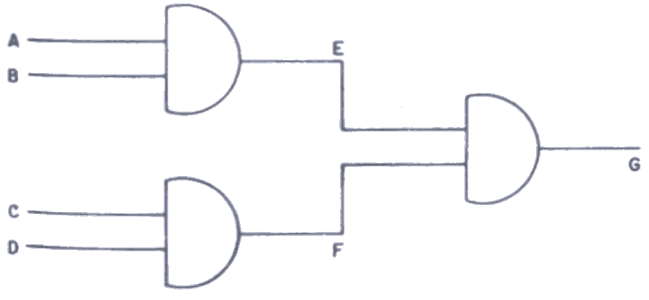


**Figure 2. Simple circuit made up of three AND gates.**

For the primary input $x_c$ = (ABCD) = (1010), the primary output is easily found to be $y_c = 0.0062$, by using TV function.

Introducing a single fault in the circuit as $f = E$ *s-a-0* (i.e., fault $f$ in line $E$ is logically stuck-at-0), the output can be calculated to be $y_c^f = 0.0031$.

Hence $\Delta^f = |0.0031 - 0.0062| = 0.0031$.

Also from Fig. 1, $\Delta_{th}$ is seen to be (0.9 – 0.1), i.e., $\Delta_{th} = 0.8$.

Since $\Delta^f = 0.0031$ is not greater than $\Delta_{th} = 0.8$, or in other words $C^f = 1/0.0031$ is not less than $C_{th} = 1/0.8 = 1.25$, it can be concluded that $x_c$ = (1010) is not a test-vector to catch the fault $E$ *s-a-0*.

## 3. ROLE OF FUZZY SETS AND FUZZY LOGIC IN THRESHOLD VALUE SIMULATION

Concepts and techniques in FS and FL can be found in literature[6,7].

### 3.1 Relevance of FS and FL in TVS Model

As has been described in Sec. 2, the use of continuous, and not binary, variables leads to a natural application of the ideas of FS and FL to the TVS method to generate test-vectors.

Fuzzification of TVS can be done as follows:

(a) The value of each line can be interpreted as a membership function $\mu^g$ for that line to have value 1, i.e., the closeness of the line to be near 1. Hence

$$\mu_A^g(x_g) = x_g \tag{7}$$

where A is the FS of the property of the line having values 'close' to 1.

(b) Defing a logic function $L_g$ for $g$ which maps inputs to outputs in terms of membership functions:

$$L_g : \{\mu_A(x_g)\} \rightarrow \mu_A(y_g), \tag{8}$$

where $y_g$ is the gate output.

(c) At any intermediate level, the FL rules are used to determine the output for any gate. For example, if $\{\mu_A(x_1) \ldots \mu_A(x_N)\}$ are the inputs for an $N$-input AND gate, then the output is [6]

$$\mu(y_g) = \min\{\mu_A(x_1) \quad \mu_A(x_N)\} \tag{9}$$

Thus the outputs of gates are calculated using FL and not TVS model. Rules for other gates can also be used[6] similarly.

(d) The output at the final level, i.e., the primary output, is the membership function $\mu^c$ for the entire circuit in terms of the primary input $x_c$.

(e) The outputs for the faulty and the fault-free cases, calculated using FL then define $\Delta^f\{y_c(x_c)\}$ as in Eqn (4) which, in turn helps define a new membership function $\mu_B^f(y_c(x_c))$, where $B$ is the FS of outputs having the property of 'distinguishing' the two cases, and detecting fault $f$.

(f) An intermediate distance function $d^f(x_c)$ in the space of all primary inputs $x_c$ is defined in terms of $\Delta^f(y_c(x_c))$ to measure how close $x_c$ is to a test-vector.

(g) Another membership function $\mu_D^f(x_c)$ is defined on $x_c$ in terms of $d^f(X_c)$, where $D$ is the FS of inputs 'close' to the test-vector.

(h) Optimisation of $\mu_D^f$ leads to a directed search for a test-vector.

The following points need to be noted:

(a) TV function (Fig. 1) has to be used at the first level to get outputs of gates from primary inputs. This can be readily seen in the case of an n-input AND gate: If any one input is 0, Eqn. (9) gives the output as 0, and for a circuit of AND gates the primary output will always be 0. But once we get non zero values, then only Eqn. (9) is used.

(b) A hierarchy of membership functions have been defined

$$\mu_A^g \rightarrow \mu_A^C \rightarrow \Delta^f = \mu_B^f \rightarrow d^f \rightarrow \mu_D^f$$

where A is the FS of outputs close to 1, B is the FS of outputs 'distinguishing' faulty and fault-free cases, and $D$ is the Fuzzy set of inputs 'close' to the test-vector.

(c) Optimisation of $\mu_D^f$ is basically an optimisation of a Fuzzy variable.

(d) Dependence of $d^f(x_c)$ on $\mu_B^f$ leads to certain restrictions on $d(x_c)$, but it is general enough to allow defining a variety of functions.

A rigorous description of all the above ideas will be presented in the next section.

## 4. THEORETICAL FORMULATION

### 4.1 Notation

Define the following:

| | |
|---|---|
| $g$ | an elemental logic gate in the circuit |
| $C$ | circuit comprising logic gates |
| $x_g$ | input to $g$ |
| $x_g^n$ | input to gate $g$ at level $n$ |
| $X_g$ | set of all inputs to $g$ |
| $y_g$ | output from $g$ |
| $T_g$ | threshold value function of $g$, i.e. $y_g = T_g(x_g)$ |
| $x_c$ | primary input to circuit $C$ |
| $x_{p,g}$ | primary input to gate $g$ |
| $X_c$ | set of all primary inputs to $C$ |
| $y_c$ | primary output from $C$ |
| $Y_c$ | set of all possible primary outputs |
| $f$ | single stuck-at-fault in $C$ |
| $F$ | set of all single stuck-at-faults in $C$ |
| $\tilde{R}$ | $(r: 0 \leq r \leq 1)$ |
| $R^+$ | $(r: 0 \leq r \leq \infty)$ |
| $\tilde{P}(X_c)$ | set of FS defined on $X_c$ |
| $\mu_A$ | membership function for $A \in \tilde{P}(X_c)$. |

### 4.2 Fuzzification for the TVS Model

(a) Let $x_c \in X_c$ be some primary input to the $C$. Then $x_c$ can be broken up into a number of primary inputs to various gates at level 1. Hence $x_0$ can be effectively written as a sum of primary inputs to the gates.

$$x_c = \{x_{p,g}\}$$

(b) For each gate $g$ at level define the mean input as $x_g^{mean,1}$ from Eqn (1).

Now $T_g$ is the TV function which gives an output for $g$ from its mean input;

$$y_g^1 = T_g(x_g^{mean,1}), y_g \in \tilde{R}. \quad (11)$$

Thus the outputs to all gates at level 1 are obtained which, in turn, act as inputs to gates at level 2,

$$\{y_g^1\} = \{x_g^2\}. \quad (12)$$

The outputs for succeeding levels of gates are calculated using FL as described below. (The

reason for treating level 1 separately has been discussed in Sec. 3).

(c) Let $A \in \tilde{P}(x_g^n)$ be a FS on $X_g^n$ (the set of inputs to the $n$ th level gates). A is defined as the set of $x_g^n$ having values close to 1. Let $\mu_A^g(x_g^n)$ be the membership function for input $x_g^n$ of $g$ to have value close to 1. For simplicity, the membership function is assumed to have the same value as the input itself, i.e.,

$$\mu_A^g(x_g^n) \equiv x_g^n. \tag{13}$$

Thus, for the full set of inputs

$$(x_g^n) = \{\mu_A^g(x^n)\} \tag{14}$$

gives the corresponding membership functions. A very important task of identifying the values on each line as the membership function itself has been carried out here.

(d) Let $L_g$ be the FL rule for gate g which yields the output membership function, given the input membership functions, i.e.,

$$L_g^n: \{\mu_A(x_g^n)\} \rightarrow (y^n), \tag{15}$$

where $L_g$ is defined in Eqn (8) and $n$ stands for the $n$ th level. Hence the output of gate at level $n$ is found, given the inputs at level $n$, provided $L_g^n$ is known. For example, if $g$ is an $m$-input AND gate, then

$$y^n = \mu_A^g(y_g^n) = \min\{\mu_A^g(x_g^n)_1, \mu_A^g(x_g^n)_2, \quad \mu_A^g(x_g^n)_m\} \tag{16}$$

where the subscripts $1 \ldots m$ stand for the $m$ inputs. Hence instead of using the TV function to calculate outputs at each stage, the FL rule (Eqn (16)), which defines $L_g^n$ for an AND gate, is employed.

(e) Thus the sequence

$$\rightarrow y_g^n \rightarrow x_g^{n+1} \rightarrow \mu_A^g(x_g^{n+1}) \rightarrow y_g^{n+1} \rightarrow x_g^{n+2} \rightarrow. \tag{17}$$

is obtained until the last level $N$ is reached, whereby the output is identified as the primary output of the entire circuit

$$\rightarrow x_g^N \rightarrow \mu_A^g(x_g^N) \rightarrow L_g \rightarrow y_g^N. \tag{18}$$

In short, if $x^m$ and $y^m$ denote the inputs and the outputs at the $m$ th level and $[L_g]^m$ denotes the rule $L_g$ implemented $m$ times, then, symbolically,

$$\begin{aligned} y_c \equiv y^N &= (L_g^N)[x^N] = L_g^N[y^{N-1}] = L_g^N[L_g^{N-1}]\,[x^{N-1}] \\ &= L_g^N \cdot L_g^{N-1} \ldots L_g^2[x^2] \\ &= [L_g]^{N-1} \cdot T_g[x_{pg}{}^{mean}] \end{aligned} \tag{19}$$

Compare Eqn (19) with Eqn (3) where $y_c$ is obtained by applying $T_g$ repeatedly. Hence,

$$y_c = Z[x_c]$$

where

$$Z = [L_g]^{N-1} \quad T_g$$

denotes the operator for obtaining the primary output from the primary input, i.e.,

$$Z: X_c \rightarrow Y_c = \{y_c\}$$

where $y_c$ takes values between 0 and 1.

Since $y_c$ depends on $x_c$, it is written as $y_c$ ($x_c$

(f) The process of finding the output $y_c$ for $C$ can be repeated after assuming that a particular fault $f \in F$ exists in the circuit (e.g., a particular line having a logical stuck-at fault).

Hence the function $Z$ is modified to a function $Z^f$ such that

$$Z^f: F \times X_c \rightarrow Y_c$$

i.e., the operator transforming $x_c$ into $y_c$ depends on the chosen fault $f \in F$. This redefines the primary output as $y^f$ where

$$y^f = Z^f(f, x_c)$$

Thus the output of the circuit, for a given fault is calculated using FL.

(g) Define a new variable

$$\Delta^f(x_c) = |y_c^f(x_c) - y_c(x_c)|$$

where

$$\Delta^f: x_c \rightarrow \tilde{R}$$

This is the 'measure' of the Fuzzy analysis to distinguish between the outputs corresponding to faulty and fault-free circuits and thus detecting the fault f. This enables the definition of another FS, $B \in P(X_c)$. Here $B$ is the FS 'capable' of detecting $f$ and is described by the membership function $\mu_B^f(x_c)$, where

$$\mu_B^f(x_c) = \Delta^f(y_c(x_c)) \tag{27}$$

Hence the notion of 'ability' or 'measure' to detect $f$ has also been fuzzified.

(h) It is desirable to define a function

$$d^f: X_c \rightarrow R^+ \tag{28}$$

where $d^f(x_c)$ is a measure of the distance of $x_c$ to a test-vector. If $x_c$ is a test-vector then $d^f(x_c)=0$, and if $x_c$ is definitely not a test-vector then $d^f(x_c)$ is $\infty$.

It is obvious that $d^f$ should be a function of $\Delta^f$, i.e.,

$$d^f = d^f(\Delta^f). \tag{29}$$

(i) *Requirement*: $d^f$ is a monotonic decreasing function of $\Delta^f$.

*Justification*: $\Delta^f$ is a measure of the outputs to be different for faulty and faultless circuits. If $\Delta^f = 0$, then both outputs are identical and $x_c$ is definitely not a test-vector.

As $\Delta^f$ increases, the capability to distinguish the outputs increases, which should result in $x_c$ coming closer to a test-vector, i.e., in $d^f$ decreasing. As $\Delta^f$ becomes 1, $x_c$ becomes a test-vector and $d^f$ becomes zero. Thus as $\Delta^f$ increases from 0 to 1, $d^f$ should decrease from $\infty$ to 0 continuously.

If $d^f$ is not monotonic, then an increase in $\Delta^f$ will result in an increase in $d^f$, which is meaningless.

(j) A new membership function $\mu_D^f(x_c)$, where

$$\mu_D^f: X_c \rightarrow \tilde{R} \tag{30}$$

can be defined in terms of df. Here $D \in \tilde{P}(X_c)$ is a FS of inputs close to a test-vector for detecting fault $f$. $\mu_D^f$ is defined as[9-11].

$$\mu_D^f = 1/(1+d^f). \tag{31}$$

(k) Optimisation of the Fuzzy membership function $\mu_D^f$ will lead to a test-vector. Also, comparison of $\mu_D^f$ for two primary inputs $x_c^1$ *and* $x_c^2$ will tell which of the two inputs is better, thus leading to a directed search.

### 4.3 Some Candidates for $d^f$, Threshold, Test-vectors

It has been argued in the previous section that $d^f$ is a monotonically decreasing function of $\Delta^f$. The boundary conditions are:

Worst case: $\Delta^f = 0 \rightarrow d^f \rightarrow \infty \rightarrow \mu_D^f = 0$,
Best case: $\Delta^f = 1 \rightarrow d^f = 0 \rightarrow \mu_D^f = 1$. (32)

Some of the likely candidates for $d^f$ are

(i) $[1/\Delta^f]-1$
(ii) $[1/e\exp(1/\Delta^f)]-1$
(iii) $[-\ln\Delta^f/e]-1$
(iv) $\tan\pi/2(1-\Delta^f/2)-1$
(v) $(-\ln\Delta^f)/\Delta^f$
(vi) $\text{cosech}\,\Delta^f - 0.8509$ (33)

The definition $d^f = 1/\Delta^f - 1$ can be seen to correspond to the cost function defined by Cheng and Agrawal[5].

The monotonic property of $d^f$ allows many more definitions in terms of $\Delta^f$, as given by the expressions given in (33).

Though Cheng and Agrawal[5] define a cost function $d^f$ (which is essentially equivalent to $1/(\Delta^f-1)$) and then minimise it, the function optimised in this paper is the Fuzzy function $\mu_D^f$, which has a more natural interpretation in the language of FS.

It is also possible to define $d_{th}$ (the threshold for $d^f$) from the $\Delta_{th}$, in turn leading $D_{D,th}$, thus enabling the determining of a test-vector. In fact, $x_c$ is a test-vector if

$$\begin{aligned} &\Delta^f(x_c) > \Delta_{th} \\ &\rightarrow d^f(x_c) < d_{th} \\ &\rightarrow \mu_D^f(x_c) > \mu_{D,th} \end{aligned} \tag{34}$$

## 5. IMPLEMENTATION FOR A SIMPLE CIRCUIT

### 5.1 Sample Calculation

We demonstrate the calculation of $\Delta^f$ for the circuit given in Fig. 2.

Let $x_c$ = (1010). Then lines $E$ and $F$ attain values 0.056 each as obtained from TV function. At the second

level, min (0.056, 0.056) = 0.056 is calculated. Hence $y_c = 0.056$.

On the other hand, if there is a fault $f = E$ *s-a*-0, then line $E$ has value 0 and line $F$ has value 0.056. Hence $y_c^f = \min(0, 0.056) = 0$. Thus $\Delta^f = 0.056$.

It can be seen that, after the first level, there is no need to refer to TV function but minima of a set of numbers need be computed, which is faster.

Since $\Delta_{th} = 0.8$, and $\Delta_f < \Delta_{th}$, it is concluded that (1010) is not a test-vector for the fault $E$ *s-a*-0.

### 5.2 Complete Calculations, Test-vectors

Since the circuit in Fig. 2 is small, explicit calculations were carried out for all the input vectors. (Not all primary inputs gave different answers because of symmetry. Only the relevant ones are mentioned.)

Table 1 gives the $\Delta^f$ as computed by the TVS method of Cheng and Agrawal (briefly denoted by $\Delta_{TVS}$) and also by the FL method (denoted by $\Delta_{FL}$) for the single stuck-at-fault $E$ *s-a*-0. The cost function $C_{TVS} = 1/\Delta_{TVS}$ is also given. Obviously (1111) is a test-vector.

**Table 1.** $\Delta^f$ **for different inputs as obtained using TVS and FL methods.** $C_{TVS}$ **is the cost function** $= 1/\Delta_{TVS}$

| Vector | $\Delta_{TVS}$ | $C_{TVS}$ | $\Delta_{FL}$ |
|---|---|---|---|
| 1000 | 0.0031 | 320 | 0 |
| 1100 | 0.0056 | 178 | 0 |
| 1010 | 0.0031 | 320 | 0.056 |
| 1110 | 0.0549 | 18.1 | 0.056 |
| 1111 | 0.944 | 1.05 | 1 |

The $d^f$ for six different functions mentioned in Sec. 4.3 is calculated, along with the membership function $\mu_D^f$, as also $d_{th}$ and $\mu_{D,th}$. All these results are presented in Table 2. (The superscript $f$ is dropped for brevity.) It can be obviously seen that all of them choose (1111) as the test-vector to detect $E$ *s-a*-0.

It is interesting to note that although all the different functions for $d^f$ yield numerically similar values for $\mu_{D,th}$ (between 0.726 and 0.818) and $d_{th}$ (0.223 to 0.378), the $\mu_D^f$ values vary over a very wide margin (0.258 to $0.478\times10^{-7}$).

If the quality of a definition for $d^f$ depends on how far the value of $\mu_D^f$ for an input which is not a test-vector, is from $\mu_{D,th}$, then it is reasonable to conclude that $d^f = (1/e)\exp(1/\Delta^f) - 1$ does a much better discrimination between a 'good' vector and a 'bad' vector than other functions. In fact, the choice of functions as per the 'goodness' dictated by FL may be made in the following order: $(1/e)\, e^{1/\Delta} - 1$, $-1/\Delta \ln \Delta - 1$, $\tan\pi/2(1-\Delta/2)-1$, cosech $\Delta$-0.8509, $1/\Delta - 1$ (corresponding to Cheng and Agrawal), $-\ln(\Delta/e)-1$.

**Table 2.** $d^f$, $\mu_D$, $\mu_{D,th}$, $d_{th}$ **for six different cases (the superscript** $f$ **is dropped for brevity)**

| Vectors | $\Delta$ | $d=(1/\Delta)-1$ | $\mu=1/(1+d)$ | |
|---|---|---|---|---|
| 1000 | 0 | $\infty$ | 0 | |
| 1100 | 0 | $\infty$ | 0 | |
| 1010 | 0.056 | 16.86 | 0.056 | $d_{th}=0.25$ |
| 1110 | 0.056 | 16.86 | 0.056 | $\mu_{D,th}=0.8$ |
| 1111 | 1 | 0 | 1 | |

| Vectors | $\Delta$ | $d=1/e\exp(1/\Delta)-1$ | $\mu=1/(1+d)$ | |
|---|---|---|---|---|
| 1000 | 0 | $\infty$ | 0 | |
| 1100 | 0 | $\infty$ | 0 | |
| 1010 | 0.056 | $2.09\times10^{7}$ | $0.478\times10^{-7}$ | $d_{th}=0.284$ |
| 1110 | 0.056 | $2.09\times10^{7}$ | $0.478\times10^{-7}$ | $\mu_{D,th}=0.779$ |
| 1111 | 1 | 0 | 1 | |

| Vectors | $\Delta$ | $d=[-\ln(\Delta/e)]-1$ | $\mu=1/(1+d)$ | |
|---|---|---|---|---|
| 1000 | 0 | $\infty$ | 0 | |
| 1100 | 0 | $\infty$ | 0 | |
| 1010 | 0.056 | 2.88 | 0.258 | $d_{th}=0.223$ |
| 1110 | 0.056 | 2.88 | 0.258 | $\mu_{D,th}=0.818$ |
| 1111 | 1 | 0 | 1 | |

| Vectors | $\Delta$ | $d=\tan\pi/2(1-\Delta/2)-1$ | $\mu=1/(1+d)$ | |
|---|---|---|---|---|
| | 0 | $\infty$ | 0 | |
| | 0 | $\infty$ | 0 | |
| | 0.056 | 21.82 | 0.044 | $d_{th}=0.378$ |
| | 0.056 | 21.82 | 0.044 | $\mu_{D,th}=0.726$ |
| | 1 | 0 | | |

| Vectors | $\Delta$ | $d=-1/\Delta\ln\Delta$ | $\mu=1/(1+d)$ | |
|---|---|---|---|---|
| | 0 | $\infty$ | 0 | |
| | 0 | $\infty$ | 0 | |
| | 0.056 | 51.43 | 0.019 | $d_{th}=0.279$ |
| | 0.056 | 51.43 | 0.019 | $\mu_{D,th}=0.782$ |
| | | 0 | 1 | |

| Vectors | $\Delta$ | $d=$Cosech$\Delta$-.8509 | $\mu=1/(1+d)$ | |
|---|---|---|---|---|
| 1000 | 0 | $\infty$ | 0 | |
| 1100 | 0 | $\infty$ | 0 | |
| 1010 | 0.056 | 16.99 | 0.055 | $d_{th}=0.278$ |
| 1110 | 0.056 | 16.99 | 0.055 | $\mu_{D,th}=0.784$ |
| 1111 | | | | |

## 6. DISCUSSION AND CONCLUSIONS

A new mathematical framework for the TVS of TVG has been presented by invoking the concepts of FS and

FL. Fuzzification of various issues and introduction of a hierarchy of membership functions defined for different FS over the universal set of input vectors were carried out. The search for a test-vector was retranslated into an optimisation problem of a Fuzzy function.

The paper also presents a new application of FS and FL, especially in the field of VLSI testing and TVG. The nonbinary variables in TVS model provided an ideal platform for the implementation of the idea of FS and FL.

Using FL, some interesting functions were defined which were found to possess some particular properties. This, in turn, led to the finding of many more functions as choices for the cost or optimising functions. Of course, different cost functions can be defined ad hoc anyway in the TVS model, but the analysis carried out in this paper gives a different perspective and basis for the various concepts introduced.

Elementary calculations of a toy model have shown that some cost functions may be better than others in choosing a vector 'closer' to a test-vector, and hence may reduce the number of possibilities that may have to be tried out in the directed search algorithm.

The determination of $\Delta^f$ requires the calculations of the primary outputs. In the TVS model, the threshold function has to be invoked repeatedly and the mean input has to be calculated every time for each gate, thus resulting in a large number of computations. In the FL approach, only the minimum of a set of numbers has to be found, which may result in considerable saving of computer time.

Though the concepts in this paper have been illustrated for a circuit with AND gates, it can be easily extended to a circuit with other logic gates (The output membership function for various logic gates in terms of the input membership functions can be found in literature[6]).

Implementation of the ideas and method presented in this paper for large realistic circuits will be taken up next.

## ACKNOWLEDGEMENT

The authors thank the Director, ANURAG, Dr K. Neelakantan for his support and encouragement in carrying out this work. The authors thank also the referees for their valuable comments.

## REFERENCES

Miczo, A. Digital logic testing and simulation. John Wiley, Singapore, 1987.

2. Agrawal, V.D. & Seth, S.C. Test generation for VLSI chips – Tutorial, IEEE Catalog No. EH0277-4, 1988.

3. Cheng K.T. and Agrawal, V.D. Unified methods of VLSI simulation and test generation. Kluwer Academic Publishers, Boston, 1989.

4. Lombardi, F. & Sami, M. (Eds). Testing and diagnosis of VLSI and ULSI, Nato ASI Series E, Vol. 151. Kluwer Academic Publishers, Amsterdam, 1988.

5. Cheng, K.T. & Agrawal, V.D. A simulation based directed search method for test generation. Proc. Int. Conf. Comput. Design (ICCD'87), Port Chester, New York. 1987.

6. Klir, G.J. & Folger, T.A. Fuzzy sets, uncertainty and information. Prentice Hall, India, Delhi, 1991.

7. Dubois, D. & Prade, H. Fuzzy sets and systems: theory and applications. Academic Press, New York, 1980.

8. Yager, R.R.; Ovchinnikov, S.; Tong, R.M. & Nguyen, H.T. Fuzzy sets and applications. *In* Selected papers by L.A. Zadeh. John Wiley, New York, 1987.

9. Zimmermann, H.J. *In* Analysis of fuzzy information: Vol. III; Applications in engineering and science, Edited by J.C. Bezdek. CRC Press, Florida, 1987.

10. Gupta, M.M. & Sanchez, E. (Eds). Fuzzy information and decision processes. North Holland, New York, 1982.

Kaleva, D. & Seikkala, S. On Fuzzy metric spaces. Fuzzy sets and systems. Vol.12, 1984, 215-29.