Fuzzy-Based CMOS Circuit Partitioning in Built-In Current Testing

Wang-Dauh Tseng and Kuochen Wang, *Member, IEEE*

***Abstract*—We propose a *fuzzy-based* approach which provides a soft threshold to determine the module size for CMOS circuit partitioning in *built-in current testing* (BICT). Experimental results show that our design approach indeed provides a feasible way to exploit the design space of BICT partitioning in comparison with other approaches with a fixed threshold, and a better module size can thus be determined to reflect a change of circuit properties.**

***Index Terms*—CMOS, cost, partitioning, performance, test.**

I. Introduction

UNLIKE THE built-in circuit testing (BICT) partitioning approach which tries to increase the gap between the normal and abnormal I_{DDQ} values, the current signatures approach [1] tracks all measurements and observes I_{DDQ} distribution to screen defect dies. One limitation of current signatures is that a high volume of test results makes it infeasible to on-chip testing. In BICT, the module size is limited by the quiescent current value in a fault-free circuit. Most of the BICT partitioning approaches [2], [3] use a fixed threshold, either binary or a predetermined value, to determine the boundary between normal and abnormal I_{DDQ}. Based on the fixed threshold, the maximum size of a module is decided. However, these methods are rigid since I_{DDQ} testing requires the measurement of an analog quantity rather than a digital signal in the case of voltage testing [2]. In addition, many factors such as temperature and pressure affect the measurement of I_{DDQ} and will make the current drifting. Thus, it is hard to obtain a threshold exactly. We propose a fuzzy-based approach to decide the module size in BICT partitioning. A mathematical framework is presented to deal with the real-world imprecision and uncertainty of I_{DDQ} measurement.

II. Preliminaries

By partitioning a circuit into modules and using a separate built-in-circuit sensor (BICS) for each module, it not only facilitates on-line self-testing, but also makes I_{DDQ} testing feasible to large circuits. The outputs of the BICS of a module are observed by applying proper input vectors to detect defects in that particular module. If there are defects that lie at boundaries of modules, it can be modeled as the neighboring module faults. Such a scheme can also be modified for external testing simply by removing the BICS's and adding test pins for current monitoring. Logic partitioning is a technique to divide a large circuit into a collection of smaller modules according to some criteria [4]. It is useful for synthesis, layout, and packaging due to the following reasons:

1) to reduce the cut size as well as to minimize the delay in critical paths for synthesis;
2) to minimize both delay and routing area for layout;
3) to limit the numbers of logic components;
4) external terminals for packaging [4], [5].

However, we must take care of the partitioning impact on the functional operation of the circuit. For example, in preserving the functional mode of the circuit after partitioning, certain sub-circuits have to be assigned to one module. As to the presence of separate power/ground lines that may affect circuit performance, we have to make sure that certain amount of power/ground nets needs to be attached to a module depending on its power assumption/size and noise concern, which are directly linked to the module size. There are similar issues for clocks that feed all modules. However, clocks have other issues such as phase delay and skew, which degrade circuit performance, and are directly linked to the module size as well.

We use five parameters:

1) *resolution* (*R*);
2) *noise immunity* (*N*);
3) *area overhead* (*A*);
4) *performance* (*P*);
5) *testing time* (*T*)

which can be characterized by the module size, and they are the input variables of our fuzzy system. Large module size may cause the leakage current over the threshold and invalidate the I_{DDQ} testing. Thus, the module size has to be limited to keep a reasonable *resolution* to insure the correctness of testing. In another point-of-view, as the module size increases, the total number of BICS's needed in a circuit decreases and, thus, the *area overhead* reduces. In order to minimize the area overhead, the module size is suggested as large as possible. Large module size results in large capacitance. Large capacitance requires long charge time and the circuit can tolerate high noise pulses. That is, as the module size increases, the *noise immunity* increases. On the contrary, large capacitance slows down system clock and thus reduces circuit *performance*. To obtain a high-speed circuit, a small module size is expected. *Testing time*, on the other hand, is proportional to the circuit capacitance since the measurement of I_{DDQ} in each test clock needs to wait until all switching transients are quiescent. Thus, small module size is required to lower down the capacitance and to speed up the I_{DDQ} measurement.

III. Design Approach

Fig. 1 shows the block diagram of our fuzzy system. Initially, the *mapping function* maps an initial module size

Manuscript received May 18, 1997; revised September 23, 1997 and March 14, 1998. This work was supported in part by the National Science Council, Taiwan, R.O.C., under Grant NSC84-2215-E-009-055.

The authors are with the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, 300 Taiwan, R.O.C. (e-mail: kwang@cis.nctu.edu.tw).

Publisher Item Identifier S 1063-8210(99)00704-0.

1063–8210/99$10.00 © 1999 IEEE

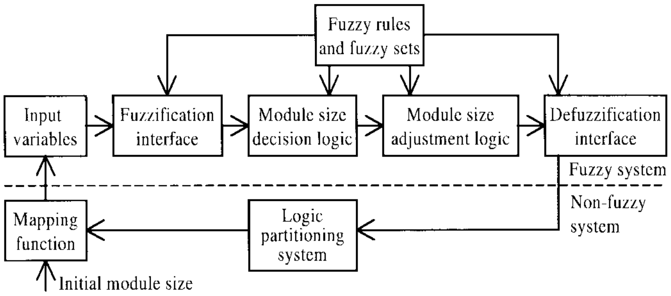
Fuzzy rules and fuzzy sets
Input variables
Fuzzification interface
Module size decision logic
Module size adjustment logic
Defuzzification interface
Fuzzy system
Non-fuzzy system
Mapping function
Logic partitioning system
Initial module size

Fig. 1. Block diagram of our fuzzy system.

to *input variables*. The input variables are then fuzzified. Based on the fuzzified variables and fuzzy rules, a module size state (MSS) is determined by the *module size decision logic*. As soon as the MSS is determined, the *module size adjustment logic* adjusts the module size. Since the inference result from the modules size adjustment logic is a fuzzy set, a *defuzzification interface* is used to convert it into a crisp value. The *logic partitioning system* then takes the crisp value to conduct logic partitioning or initiates another iteration. In Fig. 1, those units below the dashed line are not part of the fuzzy system and are not within the scope of this paper.

A. Design of Fuzzy Membership Functions

Fig. 2 shows the following three sets of membership functions used in our fuzzy system:

1) input variable membership functions;
2) MSS membership function;
3) module size adjustment membership function.

Due to the nonlinear feature of the five input variables, the sinusoidal shape membership function, which is shown in Fig. 2(a), is used to specify the input variables. We use the following three linguistic terms:

1) *small*;
2) *medium*;
3) *big*

to describe three possible levels of an input variable. The number of linguistic terms used in our system is based on the tradeoff between the complexity of fuzzy rules and the level precision of an input variable. A general equation to represent the sinusoidal shape membership function is shown as follows [6]:

$$\mu(X) = \left|\sin\left[a - b\left(\frac{X-c}{d}\pi\right)\right]\right|, \qquad u < X < w \tag{1}$$

where $\mu(X)$ is the membership value for an input variable; and a, b, c, d, u, and w are constants. The fuzzy variable used in the MSS is specified by a trapezoidal shape membership function, as shown in Fig. 2(b). The wide and flat form of the trapezoidal shape membership function makes the MSS be able to tolerate more errors from the input variables. There are three linguistic terms labeled small, medium, and big to describe three possible MSS's. A general equation to represent the trapezoidal shape membership function is shown as follows [6]:

$$\mu(Y) = mY + n, \qquad p < Y < q \tag{2}$$

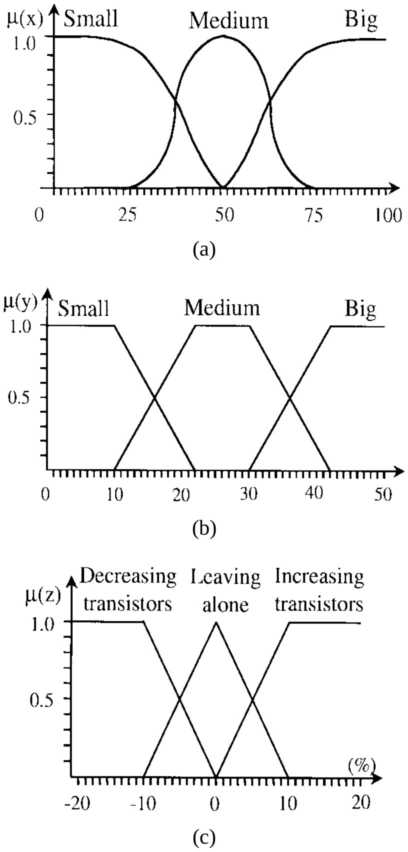

Fig. 2. Membership functions of (a) input variables, (b) MSS, and (c) module size adjustment for the fuzzy system.

where $\mu(Y)$ is the membership value for the MSS; and m, n, p, and q are constants. A triangular shape membership function for the module size adjustment is shown in Fig. 2(c). The steep slope feature of the triangular shape membership function makes the module size adjustment be able to reflect tiny variations of the MSS. We use the following three linguistic terms:

1) *increasing transistors*;
2) *leaving alone*;
3) *decreasing transistors*

to describe three possible levels of the module size adjustment. A general equation to represent the triangular shape membership function is shown as follows [6]:

$$\mu(Z) = rZ + s, \qquad i < Z < j \tag{3}$$

where $\mu(Z)$ is the membership value for the module size adjustment, and r, s, i, and j are constants.

B. Design of Fuzzy Rules for MSS

Each of the five input variables in our fuzzy system is classified into three levels of fuzzy sets. Therefore, the maximum number of possible rules for the fuzzy system is $3^5 = 243$. It is not necessary to take the entire space of possible rules into account. For example, no rule is specified for the case of big resolution and small area overhead. We select 63 fuzzy rules that cover all reasonable situations for the MSS determination. The following shows an example fuzzy rule for the MSS:

Rule 1: IF resolution is big and area overhead is big and noise immunity is small and performance is big and testing timing is small THEN MSS is small.

C. Fuzzy Inference for MSS and Module Size Adjustment

We use an example to illustrate how the MSS and module size adjustment are inferred from the values of input variables as well as fuzzy rules. Assume that the initial module size of a circuit under partitioning is 5000 transistors and the corresponding values of the input variables are $R = 32$ (nA)$^{-1}$, $A = 37$ (μm^2), $N = 60$ (pF), $P = 45$ (MHz), and $T = 67$ (ms). These values are first transformed into fuzzy sets based on (1) and Fig. 2(a). The fuzzy set for each input variable is obtained as follows: $\mu(R) = \{0.90/\mathrm{small}, 0.43/\mathrm{medium}\}$, $\mu(A) = \{0.73/\mathrm{small}, 0.68/\mathrm{medium}\}$, $\mu(N) = \{0.81/\mathrm{medium}, 0.59/\mathrm{big}\}$, $\mu(P) = \{0.31/\mathrm{small}, 0.95/\mathrm{medium}\}$, $\mu(T) = \{0.48/\mathrm{medium}, 0.88/\mathrm{big}\}$. There are $2^5 = 32$ combinations for the fuzzy sets. Excluding the invalid combinations, there are 16 rules fired for MSS big and 12 rules fired for MSS medium. By using the max–min compositional rule of inference operation, the membership value for each possible MSS can be evaluated as follows:

$$\begin{aligned}\mu(\mathrm{MSS})_{\mathrm{big}} &= \bigvee_{i=1}^{16} (\mu(R) \wedge \mu(A) \wedge \mu(N) \wedge \mu(P) \wedge \mu(T))\\ &= 0.73\\ \mu(\mathrm{MSS})_{\mathrm{medium}} &= \bigvee_{j=1}^{12} (\mu(R) \wedge \mu(A) \wedge \mu(N) \wedge \mu(P) \wedge \mu(T))\\ &= 0.48.\end{aligned}$$

Thus, the MSS membership values can be expressed as the following fuzzy set: $\mu(\mathrm{MSS}) = \{0.73/\mathrm{big}, 0.48/\mathrm{medium}\}$. A defuzzification method using the center of gravity method is used to derive a crisp value from this fuzzy set. In Fig. 3, the crisp value of the MSS (CMSS) is calculated as follows:

$$\begin{aligned}\mathrm{CMSS} &= \frac{\int_{10}^{35.45} [\mu(Y)_M \times Y]\,dY + \int_{35.45}^{50} [\mu(Y)_B \times Y]\,dY}{\int_{10}^{35.45} [\mu(Y)_M]\,dY + \int_{35.45}^{50} [\mu(Y)_B]\,dY}\\ &= 33.25.\end{aligned}$$

Note that the module size adjustment process is based on the MSS to increase, decrease, or leave alone the module size. The fuzzy rules for the module size adjustment are given as follows:

1) IF MSS is small THEN increase transistors (IT);
2) IF MSS is medium THEN leave alone (LA);
3) IF MSS is big THEN decrease transistors (DT).

Since $\mathrm{CMSS} = 33.25$, both rules 2 and 3 are fired, as shown on the left-hand side of Fig. 4. We have two module size adjustment fuzzy sets. The union of these two fuzzy sets is illustrated on the right-hand side of Fig. 4. Once again, the center of gravity method is used to get a crisp value from the fuzzy set obtained after the union. In Fig. 4, the crisp value of the module size adjustment (CMSA) fuzzy set is calculated as follows:

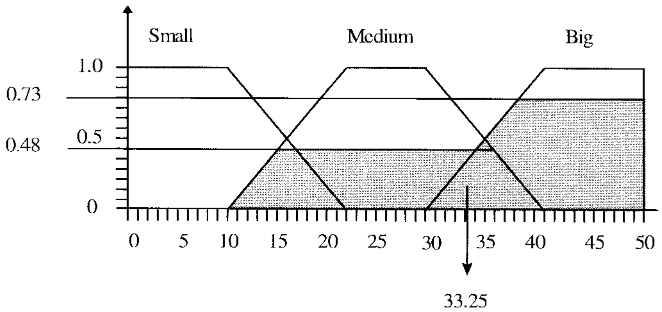

Fig. 3. Defuzzification using center of gravity method.

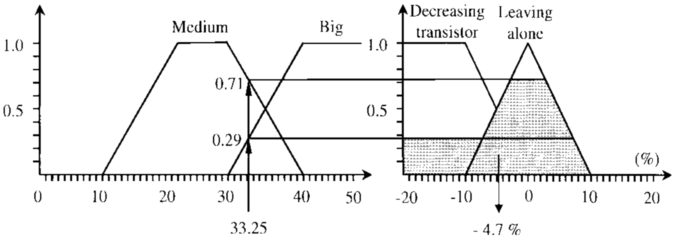

Fig. 4. Fuzzy inference process for module size adjustment with rules 2 and 3 fired.

$$\begin{aligned}\mathrm{CMSA} &= \frac{\int_{-20}^{-7.1} [\mu(Z)_{DT} \times Z]\,dZ + \int_{-7.1}^{10} [\mu(Z)_{LA} \times Z]\,dZ}{\int_{-20}^{-7.1} [\mu(Z)_{DT}]\,dZ + \int_{-7.1}^{10} [\mu(Z)_{LA}]\,dZ}\\ &= -4.7.\end{aligned}$$

This value means that the module size should be decreased by 4.7%; i.e., $5000 + 5000 \times (-0.047) = 4765$ transistors should be included in a module. As shown in Fig. 1, the inference process will not terminate until the module size changing rate converges to a predetermined value.

IV. Experimental Results

We will illustrate how the module size changes as the specifications of the five parameters are modified. The module size of a circuit is assumed to include 4765 transistors initially and the parameter values associated with this module size are the same as those in Section III-C. The value of a parameter is increased or decreased from 10% to 50% and the values of the other four parameters are fixed. All the parameters are changed by turn. Table I shows the module size changing rate with a single parameter variation, where the "+" ("−") sign associated with a parameter is used to indicate the increase (decrease) of the parameter value. The change in noise immunity that affects the module size is more dramatic than that in any other parameter. This is because the absolute value of the slope of noise immunity in Fig. 5 is the largest. The results in Table I also conform to the description in Section II: the larger the module size, the lower the circuit resolution. The same observations can be obtained by checking the other parameters in Table I. Similarly, we can derive the module size changing rate varying with two or more parameters. The

TABLE I
MODULE SIZE CHANGING RATE WITH A SINGLE PARAMETER VARIATION

Changing rates & Parameters	10%	20%	30%	40%	50%
Resolution(+)	−3.65	−4.23	−5.91	−5.91	−6.13
Area overhead(−)	−3.31	−3.90	−3.90	−5.11	−5.11
Noise immunity(+)	3.87	4.72	6.39	6.39	6.39
Performance(+)	−2.81	−3.44	−3.44	−3.44	−4.70
Testing time(−)	2.71	2.71	3.84	3.84	4.38

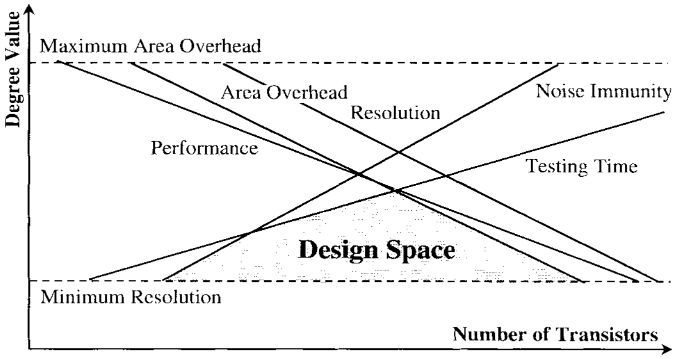

Fig. 5. Module size design space exploration.

experimental results validate that our approach is more flexible than other approaches in the determination of the module size when the specifications of the five parameters are modified.

V. Discussion

A. The Basis of the Membership Functions Used

In Section III, we have described why each of the three fuzzy membership functions is selected. To meet other special circuit requirements, we can modify the membership functions. For example, in low-power supply CMOS circuits, the gap between normal and abnormal I_{DDQ} values is very small. For I_{DDQ} testing to be effective, the gap should be discriminative. In such circuits, we can refine the membership function of resolution to decrease the module size and thus lower the normal I_{DDQ} value and enlarge the gap between the normal and abnormal I_{DDQ} values.

B. Design Space Exploration

In Fig. 5, the shadow region is our design space. For ease of illustration, the relationships between these parameters and the module size are assumed to be linear. However, our approach is equally applicable to other kinds of relationships. Two dashed lines, specified by the minimum resolution and the maximum area overhead, depict the lower and upper bounds of the design space, respectively. The minimum resolution limits the module size to a reasonable value to keep the correctness of testing, and the maximum area overhead limits the number of modules to maintain a reasonable yield of chips. Our approach can exploit the design space of the module size by defining different membership functions. Fig. 6 shows a design example of the membership functions for noise immunity and MSS. The higher the slope of the noise immunity line is, the stronger the effect of the noise immunity on the module size is. This is because the range of the MSS membership function is shrunk. This result is also consistent with that in Table I.

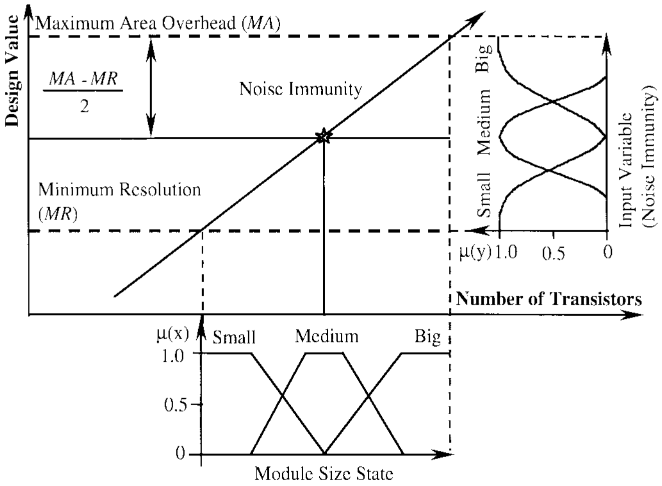

Fig. 6. A design example of membership functions for noise immunity and MSS.

C. Design Space Comparison

A statistical BICT partitioning approach, which was proposed by Malaiya *et al.* [2], is used to compare against our approach. Unlike our approach, which uses four extra parameters besides resolution to determine the module size, the Malaiya *et al.* approach only considers the resolution parameter. A potential difficulty of the Malaiya *et al.* approach is that it is hard to determine the parameters, σ_{iADDQ} and μ_{if}, exactly, where σ_{iADDQ} is the standard deviation of the current through an average cell, and μ_{if} is the mean of the additional quiescent supply current. Also, it assumes a normal (Gaussian) distribution of current values. However, several studies report that the current distributions are non-Gaussian [7], [8]. Due to different hypotheses, it is hard to compare our approach with the Malaiya *et al.* approach directly. Nevertheless, we can map the design space of the Malaiya *et al.* approach to the design space of our approach in Fig. 5. We use the same example described in Section III-C to demonstrate this point. In order to compute the module size as a function of minimum resolution, we set the resolution requirement to the minimum value and leave the other four parameters alone. The module size as a function of minimum resolution, after conducting similar computation as in Section III-C, is 5380 (transistors). This value represents the maximum module size of those approaches, such as the Malaiya *et al.* approach, which provide a fixed threshold to limit the module size. The design space of such approaches is bounded between 0–5380 (transistors). Any module size between this interval is acceptable, but there is no further guide for designers to select a proper module size to meet special requirements. In our approach, we may add a second parameter such that the module size as a function of the maximum area overhead after computation is 4258 (transistors). Thus, the design space of our approach is bounded between 4258 and 5380 (transistors). By adding more constraints of other parameters, the interval can be further shrunk. That is, after the constraints of all the parameters are applied, a suitable module size can be determined such that the circuit properties are met. This example illustrates the merits of our approach over other approaches, which originate from its flexibility to adjust the module size to adapt circuits to different environments by adjusting different parameters.

VI. Conclusions

We have presented an efficient fuzzy-based inference system to decide the module size for the partitioning of large CMOS circuits in BICT. Our approach provides a robust mathematical framework to deal with the real-world imprecision and uncertainty of I_{DDQ} measurements. The experimental results have supported that our approach can consider BICT partitioning in all its aspects in determining better module size compared with existing approaches.

References

[1] A. E. Gattiker and W. Maly, "Current signatures," in *Proc. IEEE VLSI Test Symp.,* Apr. 1996, pp. 112–117.

[2] Y. K. Malaiya, A. P. Jayasumana, Q. Tong, and S. M. Menon, "Enhancement of resolution in supply current based testing for large IC's," in *Proc. IEEE VLSI Test Symp.,* Apr. 1991, pp. 291–296.

[3] W. Maly and M. Patyra, "Built-in current testing," *IEEE J. Solid-State Circuits,* pp. 425–428, Mar. 1992.

[4] B. Preas and M. Lorenzetti, Eds., *Physical Design Automation of VLSI Systems.* Menlo Park, CA: Benjamin/Cummings, 1988.

[5] N. Sherwani, *Algorithms for VLSI Physical Design Automation.* Norwell, MA: Kluwer, 1995.

[6] F. R. Biglari and X. D. Fang, "Real-time fuzzy logic control for maximizing the tool life of small-diameter drills," *Fuzzy Sets and Syst.,* vol. 72, pp. 91–101, Mar. 1995.

[7] K. M. Wallquist, "On the effect of I_{SSQ} testing in reducing early failure rate," in *Proc. Int. Test Conf.,* Sept. 1995, pp. 910–914.

[8] P. C. Maxwell, R. C. Aitken, K. R. Kollitz, and A. C. Brown, "IDDQ and AC Scan: The war against unmodeled defects," in *Proc. Int. Test Conf.,* Sept. 1996, pp. 250–258.