Correcting Multiple Design Errors in Digital VLSI Circuits *

Andreas G.Veneris
University of Illinois at Urbana-Champaign
Department of Computer Science
e-mail: aveneris@sal.cs.uiuc.edu

Ibrahim N.Hajj
University of Illinois at Urbana-Champaign
Department of Electrical and Computer Engineering
e-mail: hajj@uivlsi.csl.uiuc.edu

Abstract

With the increase in the complexity of VLSI circuit design, logic design errors can occur during the synthesis process. In this paper we present an efficient test–vector simulation approach for multiple design error diagnosis and correction. We also compare the quality of test vector simulation and BDDs for this problem and show the competitive performance of the former. Experimental results exhibit the robustness of our approach and confirm the theoretical results.

1 Introduction

Design errors may occur in many stages during the design of VLSI digital circuits. In this paper, we are interested in the problem of multiple Design Error Diagnosis and Correction (DEDC) in an erroneous netlist. In our treatment of this problem, design errors are functional mismatches between the specification and the (erroneous) gate–level description. Sources of these mismatches can be the manual interference of the designer with the synthesis process, errors at a higher level of the design flow, and occasionally, software bugs in CAD tools.

Usually, design errors belong to a small predetermined set of possible error types, known as the design error (correction) model and presented in Abadir et al. [2] (Fig. 1). Experimental data has shown that this model covers 97.8% of the errors that occur in a real life synthesis environment [1]. For this reason and because of its simplicity, the model of Abadir et al. [2] has been used by the majority of the existing literature including the work presented here. In the work of Aas et al. [1], experiments show that the average number of design errors is usually less than or equal to 2.

Since there may be more than one way to synthesize a particular function, there may be equivalent corrections along with the actual correction that rectify an erroneous design [4]. We will refer to these corrections as *valid* corrections. It can be shown that the problem of finding a valid correction(s) for DEDC is a challenging one because the space of erroneous line candidates grows exponentially with the number of design errors [8] [9].

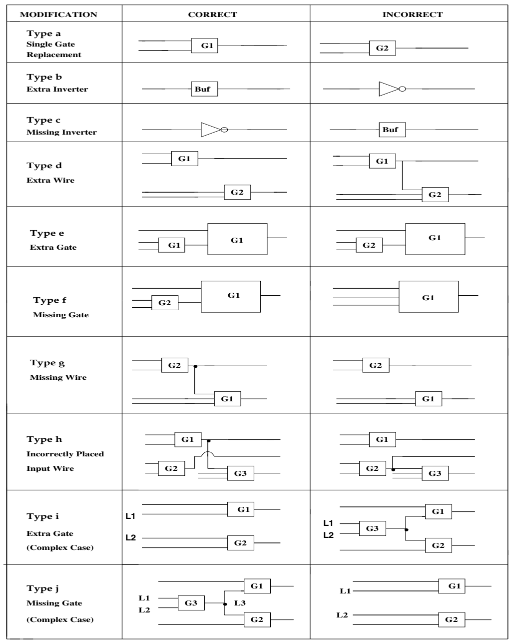

Figure 1: Abadir et al Design Error Model

Previous approaches for DEDC are divided into two categories with respect to the underlying technique used for error location and error correction, those based on **Boolean function manipulation (symbolic)** techniques [3] [4] [6] and those based on **test vector simulation** [5] [7] [8] [9]. Most techniques are fast and exhibit good resolution for single errors but heuristics are usually employed for multiple errors due to the increase of the error space. In addition, little work has been performed in the area of multiple design error correction with test vector simulation [7] [9].

Our DEDC method uses the results of test–vector simulation for both diagnosis and correction. The advantage of this method is the fact that it is applicable to circuits where other methods may fail, *i.e.* the circuits that do not have a global Binary Decision Diagram (BDD) representation. Our experimental results show that it is also run–time efficient, as it can return results for 1 and 2 errors within seconds of CPU time.

This paper is organized as follows. In the next section we describe our diagnosis approach. Section 3 contains the cor-

*This research was supported by the Joint Services Electronics Program under Contract #N00014-96-1-0129.

rection algorithm. In section 4 we present the experimental results and section 5 concludes this paper.

2 Design Error Diagnosis

In this work, we examine incorrect combinational gate–level descriptions with `NOT`, `BUFFER`, `AND`, `NAND`, `OR` and `NOR` gates. A `BUFFER` is introduced at every fan-out line of a branch. The *input* to the diagnosis algorithm is the functional specification F_C, the erroneous gate level description G_C, an initial guess of the required number of modifications N, and a set of test vectors that activate the inconsistencies V_{act}, *i.e.* each of them produces erroneous primary output responses in F_C and G_C. V_{act} is compiled during an initial test–vector simulation based verification step. During this step, we also create two *bit-lists*, $Fgroup$ and $Tgroup$ at every line l of G_C. The i-th entry of the $Fgroup$ ($Tgroup$) bit-list contains the value of l when the i-th vector that activates (does not activate) the inconsistencies is simulated. An example of an $Fgroup$ bit list is shown in Fig. 3(a).

Throughout our discussion, the words modification and correction are used interchangeably and they represent any single error from the model of Abadir et al. [2] (Fig. 1). Finally, we say that a line l of G_C, fan-in to an `AND` or `NAND` (`OR` or `NOR`) gate, has controlling value for input test vector v if the value of l under simulation for v is 0 (1).

In the following definition, we borrow from the concept of Error Possibility Index [8]:

Definition 1 *Let vector $v \in V_{act}$ and let $PO_i \in PO_{G_C}$ with* incorrect *output value under v. We define the* **Observability Measure,** *OM_l^i, of a line l for vector v and over PO_i recursively as follows:*

(a) If l is a primary output, then $OM_l^i = 1$, if $l = PO_i$, and $OM_l^i = 0$ otherwise.

(b) if l is fan-in of a gate G and l' is the fan-out of G then $OM_l^i = 0$, if l has non-controlling value and some other fan-in of G has controlling value, $OM_l^i = OM_{l'}^i$, if l has non-controlling value and all fan-ins of G have non-controlling value, and $OM_l^i = \frac{OM_{l'}^i}{cv}$, if l has controlling value, where cv is the number of fan-ins of G with controlling value.

(c) if l is a fan-out stem, then $OM_l^i = min(1, \sum_{l'} OM_{l'}^i)$, l' drives a buffer at the fan-out branch of l.

We also define the **Accumulated Observability Measure** *AOM_l^v to be $AOM_l^v = \sum_{err.\ PO_i\ for\ v} OM_l^i$.*

Intuitively, the accumulated observability measure, AOM_v^l, of a line l of G_C for $v \in V_{act}$ denotes the *potential* of l to change all erroneous primary outputs of G_C into their correct values (for v).

Theorem 1 that follows is essential for the correctness of the diagnosis algorithm. Before we prove this theorem we need to prove the following lemma.

Lemma 1 *Let $v \in V_{act}$, $PO_e \in PO_{G_C}$ erroneous for v, and let $L_v = (l_1, \ldots, l_k)$ be a minimal set of lines such that when their values are complemented, PO_e has correct value. If l is a line such that there exists at least one sensitized path from l to each member of L_v, then:*
$OM_l^e \geq \sum_{l_i \in L_v} OM_{l_i}^e$

Proof. The proof of the lemma is by induction. Let a *k-cut* on G_C be the set of all lines of G_C at level k. We define the level of the members of L_v to be 0, their immediate fan-ins 1 and so on. The level of a fan-out stem is equal to the maximum level of its branches plus one. The level of all lines not in the backtrace cone of some member of L_v need not be defined. Let j be the level of l, and let MS_i be any minimal set of lines of the i-cut, $i \leq j$, such that for every member of L_v there exists at least one sensistized path from some member of MS_i. We claim that $\sum_{l' \in MS_i} OM_{l'}^e \geq \sum_{l_v \in L_v} OM_{l_v}^e$. Proving the claim proves the lemma.

To prove the claim, we use induction on the level of cuts starting from the members of L_v towards l. For the base case, the claim obviously holds since the members of L_v are a minimal set. Assume it is true for k, we prove that it holds for all minimal sets of the $(k+1)$-cut. Let MS_{k+1} be any such set. Let S_k be the set of lines of the k-cut compiled as follows: for every set of lines of MS_{k+1} that drive the same gate G, S_k contains the output of G and for every line $l' \in MS_{k+1}$ that drives a branch, S_k contains *all* fan-out branches of l'. It is straightforward to see that S_k contains at least one minimal set of lines MS_k (otherwise MS_{k+1} is not minimal). Moreover, if l' is a line of MS_k driven by gate G, then by Definition 3(b) the minimal set of lines fan-in to G that belong in MS_{k+1} should have a sum of observability measures equal to $OM_{l'}^e$ or they will not be able to complement the value of l'. If l'' is a fan-out stem of MS_{k+1}, then by Definition 3(c) l'' will have an observability measure value equal to the sum of the observability measure values of *all* of its branches (or 1). By the pigeonhole principle, the sum of the observability measures of all lines of MS_{k+1} should be greater or equal to that of the MS_k ones.

Theorem 1 *Let $L = \{l_1, l_2, \ldots, l_N\}$, $N \geq 1$ be a set of valid modification locations for an incorrect G_C. If $v \in V_{act}$ then: $\sum_{l_i \in L} AOM_{l_i}^v \geq \#$ erroneous POs for v*

Proof. It suffices to show that if PO_e is an erroneous primary output for v then $\sum_{l_i \in L} OM_{l_i}^e \geq 1$. To prove this we use induction on the number of modification locations N. For this theorem, the level of the PO_e is 0 and the level of the lines in tha fan-in cone of PO_e is defined as in the proof of Lemma 1. The base case, $N = 1$, holds if we let L_v in Lemma 1 be the singleton PO_e and l be the only member of L. If it is true for N, we show that it holds for a set L with $N+1$ elements. Let $L' = L - \{l \in L\ with\ maximum\ level\}$. Let also l' be the line of L' with maximum level k and let MS_k be a minimal set of lines, as defined in Lemma 1, that contains l' and there exists at least one sensitized path from l to every member of $MS_k - l'$. It is straightforward to see that such a minimal set exists because L is a set of valid modification locations. Let pseudo primary input PI_p drive all lines of MS_k. By the induction hypothesis we have that $\sum_{l'' \in \{PI_p, L' - l'\}} OM_{l''}^e \geq 1$.

Using Lemma 1 and the inequality above, the induction step follows and completes the proof.

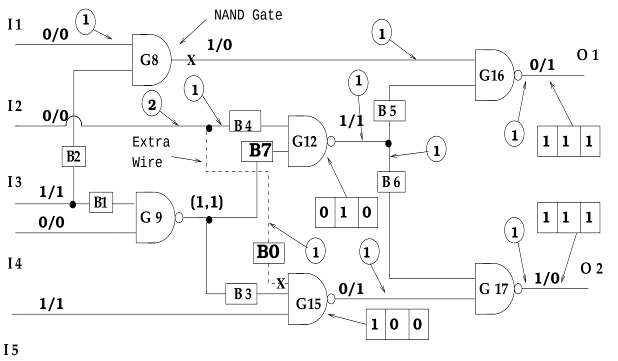

Figure 2: Erroneous Circuit For Examples 1 and 2

The recursive nature of Definition 1 along with Theorem 1 provide the basis for the first step of our diagnosis algorithm, **Total Observability Measure (TOM)**. We simulate vectors $v \in V_{act}$, one at a time, compute the AOM values for each line and delete all sets of lines $L = \{l_1, l_2, \ldots, l_N\}$ that violate the condition of Theorem 1. This is repeated until all vectors of V_{act} are simulated. It can be seen, that at every iteration the procedure reduces the number of candidate erroneous locations.

Example 1: The circuit in Fig. 2 is erroneous. Originally, G_8 is an `NAND` gate and wire B_0 is an extra wire. The circuit is simulated for vector $v = (0, 0, 1, 0, 1)$ and both primary outputs yield erroneous responses. The pairs of values on the lines are pairs of incorrect/correct values but the designer can observe the faulty circuit values. AOM is recursively computed, from primary outputs towards primary inputs, and it is in a circle when non-zero. When $N = 2$, the initial number of error pairs is 361. After completion of the TOM diagnosis procedure for vector v, the size of the error list reduces to 109 pairs, the ones with $\sum_{l_i \in \{l_1, l_2\}} AOM^v_{l_i} \geq 2$.

The second stage of diagnosis, **Inverted Simulation (IS)**, improves on error resolution and obtains information for correction, presented later. During IS, for every set of canditate erroneous lines $L = \{l_1, l_2, \ldots, l_N\}$ that qualified TOM and every vector $v \in V_{act}$, we do $2^N - 1$ simulations. Each such simulation, represented uniquely by an excitation bit-number m, $0 \leq m \leq 2^N - 1$, represents an *error excitation scenario* where l_i maintains the golden simulator value in G_C for v, if the i-th bit of m is 1, and has complemented value, if the i-th bit of m is 0. The intuition behind the procedure is that a line l with complemented simulation value for v indicates a line with a design error that is excited.

If there is $v \in V_{act}$ such that no error excitation scenario for L yields correct primary output responses, then L gets deleted from the error list. Otherwise, L qualifies and we also attach to it *all* error excitation scenario numbers that produce correct primary output responses.

Example 2: The boxes in Fig. 2 contain the initial line values at the beginning of IS for candidate error pair $(G_{12},\ G_{15})$. The first entry contains the lines values for error excitation scenario $(G_{12} = excited,\ G_{15} = not\ excited)$, the second entry contains the lines values when $(G_{12} = not\ excited,\ G_{15} = excited)$, and the third one when $(G_{12} = excited,\ G_{15} = excited)$. Observe, that all these steps can be performed efficiently in parallel at the union of the fan-out cones of (G_{12} and G_{15}) with the use of the $Fgroup$ bit-lists. Since none of these three simulations yields correct primary output responses, $(G_{12},\ G_{15})$ is deleted from the error list.

3 Error Correction

During correction, for every set of lines $L = \{l_1, l_2, \ldots, l_N\}$ that qualified IS with error excitation scenarios $\{m_0, m_2, \ldots, m_k\}$ for vector v, and for every $v \in V_{act}$, we *exhaustively* compile a list of correction from Fig. 1. From these corrections we keep the ones that, when applied, they produce an $Fgroup$ value for v that is in the set $\{2^i\ AND\ m_1, 2^i\ AND\ m_2, \ldots, 2^i\ AND\ m_k\}$, for *every* line $l_i \in L$ and *every* $v \in V_{act}$ [9]. For line related corrections, we consider line additions that do not create loops in the combinational circuit.

This procedure is illustrated in the example that follows.

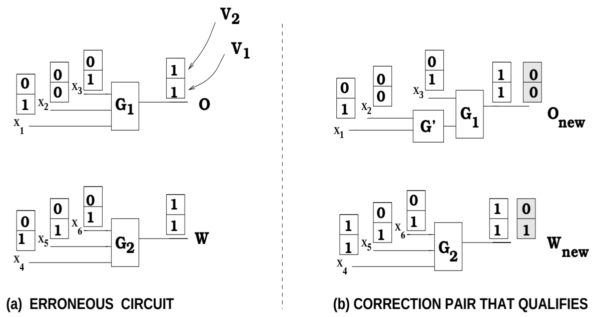

Figure 3: Wrong Gate Correction on Error Pair

Example 3: Correction is applied on error pair $\{G_1, G_2\}$ and excitation scenarios $(1, V_1)$, $(1, V_2)$, and $(3, V_2)$ that qualified from IS, as shown in Fig. 3(a). As explained earlier, excitation configuration 1 for $\{G_1, G_2\}$ and vector v says that v produces correct primary output results for IS if the potential error on G_1 is activated and the error on G_2 is not. Equivalently, excitation configuration 3 for vector v means that both potential errors on G_1 and G_2 during IS for v are activated and produce correct primary output values.

Assume that the correction pair under consideration is a missing gate for G_1 and a gate replacement error for G_2 as shown in Fig. 3(b). Assume that when we simulate the "new" gate structures driving Y and W we get the new values in the shaded boxes. We observe that the new values respect excitation scenario 1 for V_1. The same holds for V_2 and this correction pair qualifies. Observe that if excitation scenario $(3, V_2)$ was not an output of IS, then the above correction pair would not qualify.

ckt name	# of lines	1 design error					2 design errors				
		$\|V_{act}\|$	TOM time	IS time	corr. time	# corr. returned	$\|V_{act}\|$	TOM time	IS time	corr. time	# corr. returned
C432	545	40	1.9	0.9	1.7	1.9	150	18.2	10.7	22.3	8.2
C499	1224	50	11.1	4.9	4.9	2.5	220	37.4	25.4	43.9	19.1
C880	880	50	4.9	2.3	4.4	1.0	150	30.1	17.2	10.6	2.3
C1908	1908	80	7.0	9.4	6.9	2.1	220	89.9	138.4	106.5	4.2
C3540	3540	80	9.9	7.1	8.2	2.3	300	78.1	111.2	138.4	8.9
C7552	7552	100	10.8	11.5	12.0	1.4	320	128.3	199.8	169.9	18.2

Table 1: Multiple DEDC Results

Once we compile exhaustively a list of corrections as described above, a last test vector simulation based verification step is performed for the vectors of $Tgroup$. Corrections that give correct primary output responses qualify as the *output* of our DEDC algorithm.

4 Experimental Results

We implemented our DEDC algorithm in C language and ran it on a Sparc 10 workstation for ISCAS'85 benchmark circuits corrupted with 1 and 2 errors. For each error case and each circuit we ran 30 experiments. The average results of these experiments are shown in Table 1, where all run times are in seconds.

The second column of Table 1 contains the number of circuit lines which also is the size of the error space, for the 1 error case, and to the power of two, for the 2 error case [9]. For the purpose of the initial design verification and to compile the $Fgroup$ and $Tgroup$ bit-lists, we use test–vector simulation of $10,000$-$15,000$ random and stuck-at fault vectors. The next columns contain the size of V_{act}, the run-time for diagnosis (TOM and IS), the run–time needed for the correction procedure to return *all* corrections and the number of corrections returned, for the 1 and 2 errors cases, respectively. The procedure returns one correction within a fraction of the CPU time indicated in columns 6 and 11. It can be seen, that our DEDC approach is indeed robust and computationally efficient.

The corrections returned by our algorithm and shown in columns 7 and 12 guarantee to correct the design *only* for the vectors of $Fgroup$ $Tgroup$. Since, for most circuits, exhaustive test vector simulation is prohibited, we use the term **correction hit-ratio** to represent the percentage of valid corrections in the set of all corrections returned by our algorithm. The correction hit-ratio, for the vectors of $Fgroup$ and $Tgroup$, is 100% for all circuits.

Nevertheless, this ratio degrades as the number of input test–vectors decreases [9]. This is not a drawback since our approach can prune the error space efficiently so that a formal verification tool can screen the output and propose valid corrections. Therefore, our test vector simulation based approach is an attractive alternative to symbolic approaches for DEDC.

5 Conclusion

We proposed a test–vector simulation based method for multiple design error detection and correction. The results of test vector simulation provide information on suspicious error lines and necessary corrections. The method is run-time efficient and it is applicable to circuits that have no BDD representation. Our future plans include extending the above methodology to sequential circuit DEDC.

References

[1] E.J.Aas, K.Klingsheim, and T.Steen, "Quantifying design quality: A model and design experiments," in *Proc. of EURO–ASIC, pp.172–177,* 1992.

[2] M.S.Abadir, J.Ferguson, and T.E.Ferguson, "Logic Verification via Test Generation," in *IEEE Trans. on CAD, vol.7, pp.138–148,* January 1988.

[3] N. Bellas, I.N. Hajj, C. Polychronopoulos, and G. Stamoulis, "Architectural and compiler support for energy reduction in the memory hierarchy of high performance microprocessors," in *Proc. of Int'l Symp. of Low Power Electronics and Design, pp. 70–75*, 1998.

[4] P.-Y. Chung, and I.N.Hajj, "Diagnosis and Correction of Multiple Design Errors in Digital Circuits," in *IEEE Trans. on VLSI Systems, vol. 5, no. 2, pp. 233-237,* June 1997.

[5] S.-Y.Huang, K.-T.Cheng, and K.-C.Chen, "Error-Tracer: A Fault Simulation–Based Approach to Design Error Diagnosis, " in *Proc. of IEEE Int'l Test Conf., pp.974–981,* 1997.

[6] C.-C. Lin, K.-C. Chen, S.-C. Chang, and M.M-.Sadowska, "Logic Synthesis for Engineering Change," in *Proc. of ACM/IEEE Design Automation Conf., pp.647–652,* 1995.

[7] I.Pomeranz and S.M.Reddy, "On diagnosis and correction of design errors," in *IEEE Trans. on CAD, vol.14, pp.255–264,* February 1995.

[8] M.Tomita, T. Yamamoto, F.Sumikawa, and K. Hirano, "Rectification of Multiple Logic Design Errors in Multiple Output Circuits," in *Proc. of the Design Automation Conf., pp.212–217,* 1994.

[9] A.G.Veneris and I.N.Hajj, "Multiple Design Error Diagnosis and Correction in Digital VLSI Circuits," in *Proc. of IEEE VLSI Test Symp.,* 1999 (accepted).