On Efficient Error Diagnosis of Digital Circuits

*Nandini Sridhar**
Intel Corporation
Dupont, WA 98327
nandini.sridhar@intel.com

Michael S. Hsiao†
Bradley Dept. of ECE, Virginia Tech
Blacksburg, VA 24061
mhsiao@vt.edu

Abstract

The rising trend in large scale integration and design complexity has greatly increased the need for efficient design error diagnosis. We present techniques for fast and efficient error diagnosis of digital circuits by eliminating to a large extent the set of false candidates identified by the diagnosis. The elimination of false candidate regions is conducted via distinguishing X's, flipping of values at the output of candidate regions, and combination of these techniques. Our algorithms help to improve both the speed and resolution of error diagnosis. Experimental results on combinational benchmark circuits showed that up to 92% improvement in diagnostic resolution and 74% speedup over the original region-based diagnosis can be achieved with our approaches.

1 Introduction

Error diagnosis occurs early in the design flow before the actual chip is fabricated. It is invoked once a verification tool has identified that the the design implementation does not conform to the specification. This is due to errors introduced during the design process. The complexity of present day designs makes it cumbersome for the designer to locate these errors manually. Thus, efficient automatic error diagnosis tools can be extremely helpful to the designer in providing feedback about potential error sites in the circuit. Only these sites need be targeted for correction of the implementation.

Veneris et al. [1,2] presented an efficient approach for Design Error Detection and Correction(DEDC) based on test vector simulation and Boolean function manipulation. The error location and correction algorithm provides a list of all possible actual and equivalent single modification locations along with their respective corrections. Abadir et al. [3] presented a single design error model for the DEDC problem, a subset of which is used in most cases of DEDC [4] and [5]. Pomeranz and Reddy [6, 7], proposed test vector simulation methods for the DEDC problem. However, their method did not always guarantee a solution and was demonstrated only on small circuits. Tomita et al. [8, 9] proposed the use of IPLDEs (input pattern for locating design errors) for both single and multiple errors. Ayman Wahba and Dominique Borrione [10] presented an efficient technique to localize connection errors in combinational circuits. To do so, they generated special test patterns that could rapidly locate the error. Some of the earlier work carried out in error diagnosis presented in [11–13] dealt only with gate errors. Other techniques such as [6, 9] dealt with gate connection errors too, but were limited in that they were not precise in locating the error and used a fairly large number of test patterns to locate the error. Huang et al. [14] presented techniques for diagnosis for both sequential and combinational circuits by extensive enumeration and simulation. In some error model based approaches such as [15, 16], after the diagnosis is completed, the error is matched with an error type in the model and the implementation is rectified accordingly.

A general model for both fault and error diagnosis was proposed by Boppana et al. [17] and has been used to effectively diagnose single errors in combinational circuits. The model was then extended to locate multiple errors and used the concept of locality (Region-based error model) [18,19]. This work was furthered by D'Souza et al. [20] to tackle diagnosis of sequential circuits.

In this work, we present three algorithms to perform enhanced error diagnosis by eliminating as many false candidates as possible from an initial list of candidate error regions obtained from the original region-based model. Experiments are conducted on ISCAS85 combinational benchmark circuits and the results indicate that up to 92% improvement in diagnostic resolution and 74% speedup over the original region-based diagnosis can be achieved with our approaches.

The rest of this paper is organized as follows. Section 2 gives an overview of the region-based model and terminologies. Section 3 explains the proposed algorithms for eliminating false candidates. Section 4 discusses experimental results, and Section 5 concludes the paper.

*Formerly with Dept. of ECE, Rutgers University

†Supported in part by an NSF Career Award, under contract CCR-0093042, NSF grant CCR-0098304, and NJ Commission on Science & Technology. Formerly with Rutgers.

0-7803-7169-0/01 $10.00 ©2001 IEEE

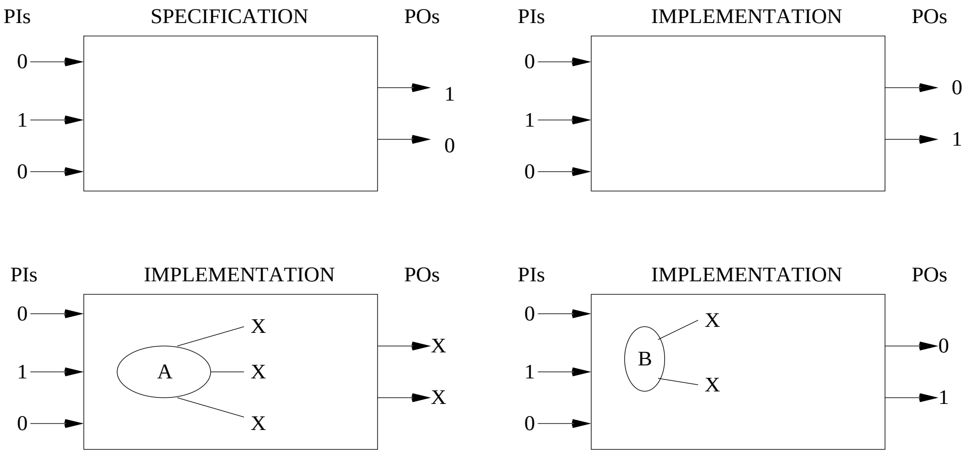

Figure 1: Region-Based Diagnosis: Region A - candidate region; Region B - not a candidate region.

2 Preliminaries

The region-based model was introduced by Bopanna et al. in [18]. Typically a region consists of a node or a gate in the circuit and its surrounding gates. The "radius" of a region specifies the actual size of the region. A region of radius 0 consists of just one single node in the circuit. A region of radius 1 consists of a "center" node, its immediate predecessors and successors. Thus, there will be as many regions as the number of nodes in the circuit (i.e., each node can be the *center node* for exactly one region of a given radius); however, the regions will be overlapping with each other when their radii are greater than 0. These regions form the basis to model, inject, simulate and locate errors.

Simulation is used during diagnosis. During this process, all the output nodes of a region are first set to unknown value X to cover any arbitrary error that may occur in the region. If no X propagates to a primary output for a given vector V, we can conclude that any error within this region is not detectable for this vector because there does not exist any sensitizable path from the outputs of this region to a primary output. Otherwise, we can assume that the region is a possible candidate that might have caused the erroneous behavior. This is illustrated in Figure 1. The specification (top-left) gives the expected response for input vector 010 and the implementation (top-right) gives the actual response (erroneous in this example). If an input vector distinguishes the response of the implementation and the specification, it is an *erroneous vector.* In this case, 010 is an erroneous vector. The output(s) that exhibit differing values between the specification and implementation are called *erroneous outputs.* Because both primary outputs are in error, both outputs are erroneous outputs. During diagnosis (using region-based model), it is observed that for region A, the X values on its fanouts propagate to every erroneous output for the particular erroneous vector. Thus, it is a candidate region. In case the Xs propagate to only some of the erroneous outputs, the error may not be fully contained within the region. For region B, no X propagated to any erroneous output for this vector, indicating that no sensitizable paths exist to propagate the Xs to any output. If the error is restricted to within the size of one region, we can safely discard region B from the candidate list.

3 Design Error Diagnosis

We propose three techniques to perform efficient error diagnosis. The basis of our techniques is the attempt to differentiate between true and false candidate regions. To achieve this, we vary the manner in which each region is excited. These changes in the region-based model give us highly improved error diagnosis.

3.1 Method 1: Flip Fanout Bits

In this method, we try to magnify those factors that can distinguish between the actual region in error and those that are not in error. The initial list of candidates is first obtained using the original region-based model for all erroneous vectors. We aim at eliminating false candidates from this list by trying to correct the actual region in error while at the same time making the false candidates (error-free regions) erroneous. These effects can be observed at the primary outputs of the circuit and can help us to distinguish between true and false candidate regions. Unlike in the original model, we observe all the primary outputs of the circuit and not only the erroneous ones.

Every candidate obtained by the region-based model is examined for every erroneous vector. For each region, all except one fanouts of the region are forced to an 'X' value. The one that is not forced to an 'X' value is flipped to its opposite value. Then, an event-driven simulation is performed from this region onward. This process is repeated for every fanout of the region. By flipping one fanout to its opposite value and keeping all others as Xs, we are trying to correct the error effect at the fanouts of the underlying region. For false candidates (regions not containing any error), with error-free inputs, flipping of a bit makes at least one fanout erroneous. Because this injected error may be propagated to any output (not only to the erroneous outputs), we consider all primary outputs during this stage of diagnosis.

If a particular region causes all the primary outputs of the circuit to have either the correct circuit values or don't care values, for at least one fanout-flip case, then the region is still considered a candidate region. This is because flipping of one fanout's value causes the error to disappear altogether, making this region a possible candidate containing *the* error that affected the flipped signal. On the other hand, if in all fanout-flip cases, at least one erroneous output remains or additional errors are observed, then this region is no longer a candidate region and can be discarded from the list of candidates. This is because if the region did not contain any error and its inputs are error-free, then all of its fanouts will have error-free values.

Flip-Fanout-Bits Method: Flip each of the region's fanouts in turn, while all others are set to X, simulate and observe the primary output values. If the error effects for each flipped fanout can be observed at the primary outputs, then the region is definitely a false candidate region.

Theorem 1:
A candidate region identified by the original region-based model is no longer a candidate region, if it is nullified by The Flip Fanout Bits technique.

Proof:
If a particular region α contains the actual error, then it will affect one or more of its region fanouts for every erroneous vector. For a given erroneous vector, by flipping every fanout of the region in turn while keeping all others 'Xs', we are trying to correct at least one of the fanouts which have been affected by the error in the region. If an erroneous fanout of the region is flipped to its opposite value, it has become correct. Consequently, with an erroneous fanout flipped and all other fanouts set to Xs, there should not be any erroneous value propagated to any primary output for this vector; all the primary outputs of the circuit should have either correct or don't care (X) values. Thus, the true candidate region will not be excluded by this method. On the other hand, if at least one error propagates to a primary output for every flip at the region fanouts, then we can conclude that the error is definitely not contained within this region. ◇

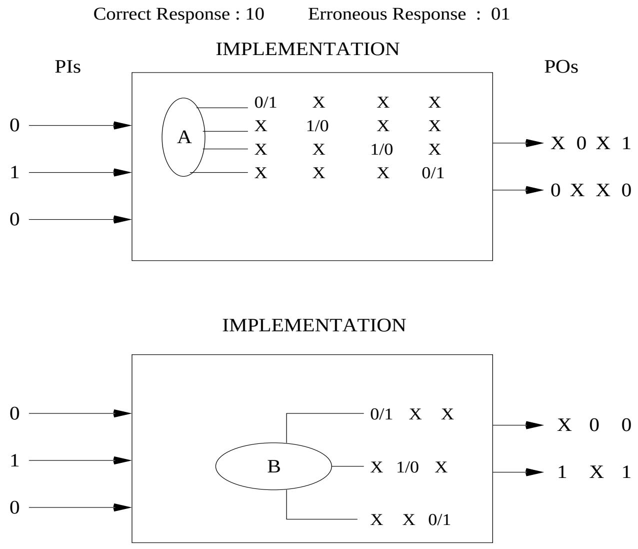

Figure 2: Flip Fanout Bits Example.

In doing so, for the true candidate region α, flipping at least one of its fanouts, with all other Xs, all the primary outputs of the circuit should have either error-free values or X values. In contrast, for a false candidate region β, if flipping *each* of its fanouts in turn causes an erroneous value at at least one primary output, then we can conclude that β is a false candidate and can be discarded from the candidate list.

Figure 2 shows how the fanouts of a region are set. The original specification-implementation model for vector 010 is the same as that in Figure 1. Let us assume that after the initial region-based diagnosis is performed, regions A and B are both considered candidate regions. Now we apply our Flip-Fanout-Bits technique where we force all except one fanout to 'Xs' and then flip the other fanout to assume its opposite value. For region A, we see that by flipping the first, third and fourth fanout, the output which was erroneous became either a correct value or an X value. Hence region A is still considered a candidate region. Since the circuit needs to be correct in only one of the cases, we need not proceed with flipping subsequent fanouts after having flipped the first fanout to save execution time. For region B, we see that at least one of the erroneous primary outputs remains erroneous for every fanout being flipped. Therefore region B is no longer considered a potential error site and can be dropped from the list of candidates. The algorithm for this technique is shown in Figure 3.

3.2 Method 2: Distinguishing Xs

In this method, we do not alter the manner in which the region is excited; instead, we alter the type of X

```
begin
  Error List = Candidate List of Region-based model
  for each **erroneous vector V**
    for each **region R** in Error List
      for each **fanout F**
        Flip F //set fanout F to it's opposite value
        Set other fanouts to X
        Simulate(V,R)
        for each **primary output PO**
          if ($PO \neq X$ and $PO \neq correctvalue$)
            flag = flag + 1
            break
          endif
        endfor
      endfor
      if(flag = = No. of Fanouts of R)
        Error List = Error List - R
      endif
    endfor
  endfor
  Final Candidate List = Error List
end
```

Figure 3: Flip Fanout Bits Algorithm.

that is used. In this technique, every X value injected is unique; no X injected at the output of a region is the same as any other X. Also we establish relationships between some of the Xs. For instance, although an X and its complement are different from each other, they are also related - they may nullify each other when combined. The concept of differentiating *unknowns* was first introduced by Carter et al. in [21]. This was done in order to prevent the loss of information during simulation with these unknowns. Instead of having a set of *named unknowns* together with the 3 values 0, 1 and X, we apply the concept to differentiate every X in the circuit from another. These Xs are termed as "Distinguishing Xs". Each X is represented as an X with an associated id. The illustrations in Figure 4 show the difference between simple Xs and distinguishing Xs. The Xs with even ids are even-polarity Xs. If an X is complemented, it is either incremented or decremented, depending on the id. Thus, two Xs with ids of an even-number k and k+1 may nullify each other through an And or Or gate. For instance, X[0] AND X[1] yields a logic 0; similarly, X[2] OR X[3] produces a logic 1.

We know that most non-fanout-free circuits contain fanout-reconvergence. Thus, by using distinguishing Xs, we may be able to eliminate some Xs at reconvergence points, thus preventing them from propagating all the way to the erroneous outputs. This is depicted in a sample circuit fragment shown in Figure 5. When using simple Xs, region A was considered a candidate error region by the original region-based model because the X's propagate to the primary output. However, with the Distinguising-X approach Region A gets eliminated since the Distinguishing Xs are masked and do not propagate to the erroneous output.

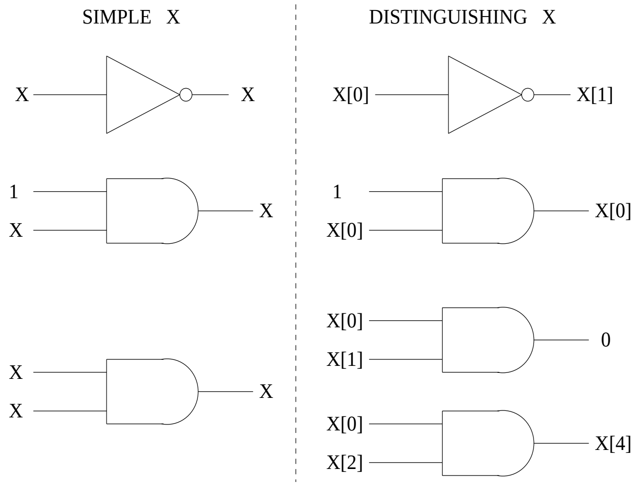

Figure 4: Differentiating Simple and Distinguishing Xs.

We use this concept of Distinguishing Xs in our diagnosis algorithm. In this method, all regions of the circuit are considered to be in the initial list. Every region is taken in turn and its fanouts are set to Distinguishing X values. If the Xs are able to propagate to every erroneous output for every respective erroneous vector, then the region is a candidate region, otherwise it is not. In Figure 6, region A has 4 fanouts. They are set to X values with ids of 0, 2, 4 and 6 respectively. It is seen that the X value propagates to the erroneous outputs. Thus, region A is a candidate. On the other hand, for region

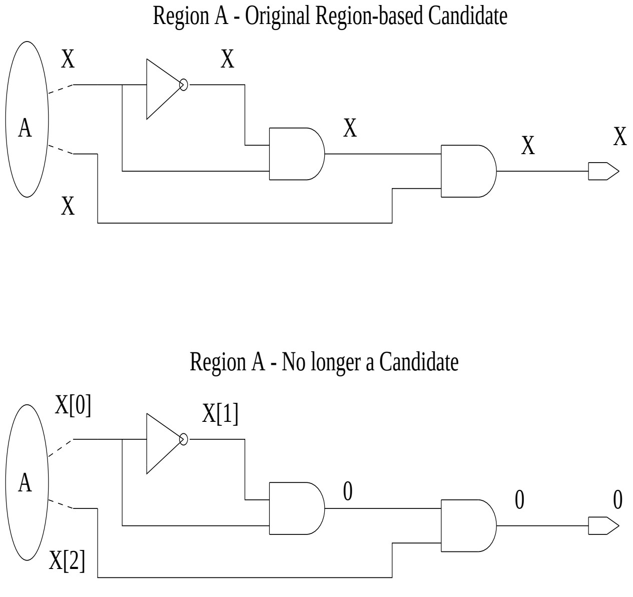
Region A - Original Region-based Candidate
A
X
X
X
X
X
X
Region A - No longer a Candidate
A
X[0]
X[1]
0
0
0
X[2]

Figure 5: False Region Identified Using Dist. Xs.

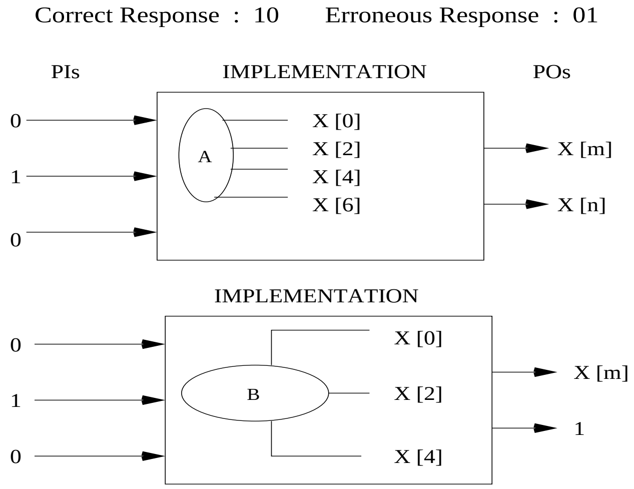

Figure 6: Distinguishing X Example.

B, the distinguishing X value gets masked along the way and does not propagate to the erroneous outputs. It is no longer a candidate region.

<u>**Distinguishing-X Method:**</u> Assign each of the region's fanouts a distinguishing X, simulate and observe the primary output values. If the distinguishing Xs do not propagate to every erroneous output for every respective erroneous vector, then the region is not a candidate region.

Theorem 2:
A candidate region in the circuit identified by the original region-based model is no longer a true candidate if it is eliminated by the Distinguishing X technique.

Proof:
A region that contains the actual error (true candidate) will affect one or more of its region fanouts, and its error effect(s) are able to be propagated to the respective erroneous outputs for *every* erroneous vector. Hence, there exists sensitizable paths for each error-effect, starting at the fanouts of the region to the erroneous primary outputs for each erroneous vector. Along these paths, no reconvergence can nullify the propagation of error effects, if the region is indeed a true candidate region. Thus, if distinguishing X's prevent propagation of an X to an erroneous output, we can conclude that the region is a false candidate. ⋄

Non-distinguishing Xs (in original region-based model) are never masked at a reconvergent point along the way, forcing them to be propagatable to the erroneous outputs. Distinguishing Xs, in contrast, may force the unknown values to be masked along the way, marking the region a false candidate. The algorithm for the Distinguishing X model is shown in Figure 7. In this algorithm, we do not require the initial candidate list from the region-based model. Since we do not have to wait to first obtain the list from the region-based model, the execution times can potentially be faster with this technique than the original region-based model, which is shown later in the experimental results.

```
begin
 Error List = All Regions in the circuit
 for each erroneous vector V
  for each region R in Error List
   for each fanout F
    Set to Distinguishing X value //X value with unique id.
   endfor
   Distinguishing X-Simulate(V,R)
   for each erroneous primary output EPO
    if ($EPO \neq DistinguishingX$ )
     Error List = Error List -R
     break
    endif
   endfor
  endfor
 endfor
 Final Candidate List = Error List
end
```

Figure 7: Distinguishing X Algorithm.

3.3 Method 3: Combined Approach

Methods 1 and 2 each has its merit. Therefore, in this method, we try to combine the effectiveness of the previous two to achieve even better performance. Simply cascading the two techniques does not prove beneficial and hence we try a *different* technique. We maintain the concept of the Flip Fanout Bits model, wherein every fanout of a region is flipped in turn, but instead of keeping the other fanouts as Xs we make them Distinguishing Xs. Thus, some of the candidate error regions that could not be eliminated by either technique alone, can now be eliminated by this model.

In this method, we again start with the candidate list obtained from the original region-based model as the initial list. Every region in the list has its fanouts flipped in turn with all others kept as Distinguishing Xs. As explained earlier, for a true candidate region, at least one case of flip should result in a non-erroneous primary output. With the other fanouts being Distinguishing Xs, the primary outputs should be either X or error-free values. If this condition is not met in any case of fanout-flip, then it can be asserted that the region is a false candidate.

<u>**Combined Method:**</u> Flip each of the region's fanouts in turn, while all others are set to Distinguishing-Xs, simulate and observe the primary output values. If the

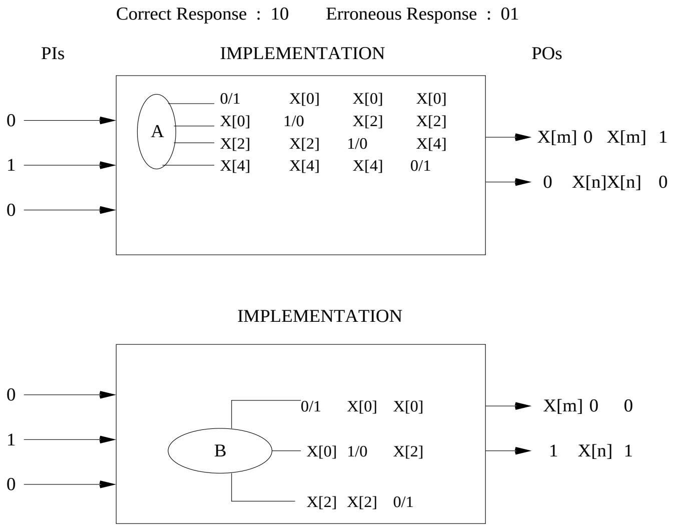

Figure 8: Combined Approach Example.

error effects for each flipped fanout can be observed at the primary outputs, then the region is definitely a false candidate region.

Theorem 3:
A candidate region in the circuit identified by the original region-based model is no longer a true candidate if eliminated by the Combined approach.

Proof:
The proof follows from proofs for Theorems 1 and 2. ◇

In Figure 8, we see how method 3 proceeds. Region A has four fanouts. Thus each fanout in turn is flipped to its opposite value while all others have Distinguishing X values. In the Distinguishing X model, since each X is different we start with a unique unrelated id. for each of them (0, 2 and 4). When applying simple Xs, for the region-based model or the flip fanout bits model, these Xs can never mask each other. With distinguishing Xs, however, this may be achieved which helps in elimination of some false candidates. Also, in the distinguishing X model, there was no way of further differentiating between true and false candidates. This can be achieved by flipping the fanouts in turn and observing the response at the primary outputs of the circuit. In this method, we also observe all primary outputs of the circuit, the reason for which was explained in section 3.1. Region A gives no erroneous response for more than one case of flip for the erroneous input vector and hence is still a candidate error region. Region B, which was originally considered a candidate for all previous methods discussed, now gets eliminated from the candidate list, because it is not able to propagate the distinguishing X values to all the erroneous primary outputs or to correct them, for the current erroneous input vector.

4 Experimental Results

The effectiveness of all three error diagnosis approaches is presented in this section. We are able to enhance the performance of error diagnosis to a great extent. The experiments are conducted for ISCAS85 combinational benchmark circuits on a Sun Ultra 10 workstation with 256MB of memory and 440MHz clock. Table 1 shows the circuit parameters for the benchmark circuits; they include for each circuit, the total number of input vectors used (Tot.Vec.) and the total number of regions present in the circuit (Tot.Reg.). These parameters are common to all the different experiments.

For all experiments, average values for 10 different random errors injected into each benchmark circuit are reported. Single errors are definitely within a region of radius 1. We also experimented with multiple errors, in which we restrict the multiple errors to be within a region of radius 1.

Table 1: Circuit Parameters

Ckt.Name	Tot.Vec.	Tot.Reg.
c432	54	203
c499	184	275
c880	178	469
c1355	198	619
c1908	280	938
c2670	102	1566
c3540	350	1741
c5315	162	2608
c6288	40	2480
c7552	221	3827

4.1 Results for Flip Fanout Bits Model

The results for this technique are shown in Tables 2 and 3 for single and multiple gate substitution errors, respectively. The columns of the tables indicate for each circuit, the number of erroneous vectors, the final number of candidate regions (Cand.), execution time (T(s)) and the Hit rate (Hit), for both the original region-based model and our model. Hit rate indicates whether the actual error was included within one of final candidate regions. Finally, the last column (% Red.) indicates the percentage reduction in the number of candidates from the original region-based model using our approach.

For single error diagnosis, we observe that there is a considerable improvement in the number of candidate error regions compared to the results obtained using the original region-based model for all circuits. For some circuits, the number of candidate regions is almost halved. For instance, in circuit c499, with 82.4 erroneous vectors, the original region-based model yielded 59.30 candidate regions in 0.51 seconds, while the Flip Fanout

Table 2: Diagnosis Using Flip Fanout Bits Model - Single Gate Sub. Error

Ckt. Name	Err. Vec.	Region-Based Model			Flip Fanout			% Red.
		Cand.	T(s)	Hit	Cand.	T(s)	Hit	Cand.
c432	8.80	27.50	0.07	1	25.30	0.10	1	8.0
c499	82.40	59.30	0.51	1	27.30	0.95	1	54.0
c880	65.50	20.50	0.28	1	17.60	0.42	1	14.0
c1355	66.10	61.60	0.79	1	47.20	1.43	1	23.3
c1908	133.80	85.20	2.15	1	46.40	3.91	1	45.5
c2670	59.30	37.90	0.85	1	29.80	1.75	1	21.3
c3540	96.00	45.50	2.26	1	37.90	4.95	1	16.7
c5315	74.20	79.00	2.64	1	59.00	5.30	1	25.3
c6288	35.90	703.60	18.00	1	545.60	54.00	1	22.5
c7552	101.90	84.60	5.23	1	63.40	12.30	1	25.1

Table 3: Diagnosis Using Flip Fanout Bits Model - Multiple Gate Sub. Error

Ckt. Name	Err. Vec. .	Region-Based Model			Flip Fanout			% Red.
		Cand.	T(s)	Hit	Cand.	T(s)	Hit	Cand.
c432	14.43	22.57	0.06	1	19.29	0.12	1	14.5
c499	121.71	31.86	0.35	1	17.57	0.95	1	44.8
c880	51.86	30.14	0.22	1	25.00	0.59	1	16.1
c1355	93.14	37.14	0.55	1	29.00	0.90	1	21.9
c1908	89.43	22.86	0.80	1	15.52	1.05	1	32.1
c2670	38.86	54.43	0.79	1	46.85	1.25	1	13.9
c3540	209.86	17.14	1.93	1	14.00	3.75	1	18.3
c5315	80.43	10.00	1.40	1	7.70	2.30	1	23.0
c6288	24.29	485.86	11.00	1	374.14	30.75	1	23.0
c7552	75.43	73.43	4.53	1	55.43	9.56	1	24.5

Bits model obtained 27.30 candidates in 0.95 seconds, a 54% improvement in diagnostic resolution. A significant reduction of candidate error regions is also observed for c1908. The execution time increases slightly due to repeated flipping of region fanouts, but this increase is relatively small for most circuits. The Hit rate always remains 1, indicating that there is never a mis-diagnosis, i.e. the actual error was always contained in one of the regions of the final candidate list.

The results for multiple gate substitution errors also show similar reduction in candidate error regions. For instance, in circuit c499, the original region-based model reported 31.86 final candidate regions from 121.71 erroneous vectors, while our Flip-Fanout-Bits technique reported only 17.57 candidate regions, an almost 45% improvement in resolution. Likewise, reduction of candidate regions was achieved for other circuits. We see that our approach is independent of single or multiple error model, the results validate the effectiveness of this technique in improving diagnosis of arbitrary errors.

4.2 Results for Distinguishing X Model

The results for the Distinguishing X technique are shown in Tables 4 and 5 for single and multiple gate substitution errors, respectively. For single error diagnosis, significant improvement in diagnosis is observed, where the number of final candidate regions is greatly reduced. The execution time is sometimes slightly greater than the region-based model while in some cases it is less. Although the algorithm for distinguishing Xs is more complex than simple non-distinguishing Xs, the execution time can be reduced because the reduction in the number of candidates takes place very quickly with the first few erroneous vectors, so that only these few regions have to be dealt with for later vectors. For example, in c1355, we have 66.1 erroneous vectors. These vectors give a final list of 61.6 candidates in 0.790 seconds for the original region-based model and 28.8 candidate error regions in 0.70 seconds for our model. The number of final candidate regions is more than halved. In addition, the execution time is also reduced. The same is true for c3540. In the case of c6288, there is a remarkable reduction in the number of candidate regions because

Table 4: Diagnosis Using Distinguishing X Model - Single Gate Sub. Error

Ckt. Name	Err. Vec. .	Region-Based Model			Dist. X Model			% Red.
		Cand.	T(s)	Hit	Cand.	T(s)	Hit	Cand.
c432	8.80	27.50	0.07	1	14.00	0.07	1	49.1
c499	82.40	59.3	0.51	1	40.20	0.50	1	32.2
c880	65.50	20.50	0.28	1	17.50	0.35	1	14.6
c1355	66.10	61.60	0.79	1	28.80	0.70	1	53.2
c1908	133.80	85.20	2.15	1	60.50	2.32	1	29.0
c2670	59.30	37.90	0.85	1	33.00	1.04	1	12.9
c3540	96.00	45.50	2.26	1	22.60	2.02	1	50.3
c5315	74.20	79.00	2.64	1	59.10	3.12	1	25.2
c6288	35.90	703.60	18.00	1	85.50	4.77	1	87.8
c7552	101.90	84.60	5.23	1	60.80	5.43	1	28.1

Table 5: Diagnosis Using Distinguishing X Model - Multiple Gate Sub. Error

Ckt. Name	Err. Vec.	Region-Based Model			Dist. X Model			% Red.
		Cand.	T(s)	Hit	Cand.	T(s)	Hit	Cand.
c432	14.43	22.57	0.06	1	9.86	0.07	1	56.3
c499	121.71	31.86	0.35	1	21.29	0.48	1	33.1
c880	51.86	30.14	0.22	1	24.71	0.4	1	18.0
c1355	93.14	37.14	0.55	1	18.57	0.51	1	50.0
c1908	89.43	22.86	0.80	1	17.14	0.94	1	25.0
c2670	38.86	54.43	0.79	1	46.14	0.95	1	15.23
c3540	209.86	17.14	1.93	1	9.57	1.88	1	44.2
c5315	80.43	10.00	1.40	1	7.43	1.70	1	25.7
c6288	24.29	485.86	11.00	1	54.86	3.51	1	88.7
c7552	75.43	73.43	4.53	1	48.00	4.15	1	34.6

of a significant presence of reconvergence in c6288. The original region-based model gives a final list of 703.6 candidates after an execution time of 18 seconds, whereas our model gives a final list of 85.5 candidate regions, just after 4.77 seconds. This is nearly an 85% improvement in resolution and a 74% improvement in speed.

For multiple gate substitution errors, we also note this trend of performance for most circuits. In c6288, 485.86 candidate regions were trimmed down to only 54.86 regions in 3.51 seconds.

4.3 Results for Combined Model

Tables 6 and 7 show the results for the combination model for single and multiple gate substitution errors, respectively. Again we can see that in all cases there is a significant improvement in resolution of diagnosis as compared to the original region-based approach. Take circuit c6288 again, a drastic improvement in terms of number of final candidates was achieved. For single errors the number of candidate regions is reduced from 703.6 regions to only 56.10 regions. In the case of multiple errors, the region-based model gives 485.86 candidate error regions whereas our technique results in only 38.86 candidate error regions. This is again due to the high degree of convergence in this circuit. The execution times are slightly longer compared to the region-based approach, but the increase is tolerable considering the highly improved resolution. In all other circuits we also see that the diagnostic resolution is greatly improved.

4.4 Comparison of Various Techniques

Finally, we present a comparison among all the approaches that we have carried out in Table 8. This gives us useful information as to which technique is better applied depending on requirements. We compared methods 1, 2 and 3 with each other and also with the original region-based approach. If speed of diagnosis is the main criterion, then Method 2 using the Distinguishing Xs would prove to be the best option. This is the only technique that does not use the candidate list obtained from the region-based model to start diagnosis. Thus, in many cases the execution time is even less than that of the region-based approach and the number of candidates obtained is few. If the final number of candidates

Table 6: Diagnosis Using Combined Model - Single Gate Sub. Error

Ckt. Name	Err. Vec.	Region-Based Model			Combined Model			% Red.
		Cand.	T(s)	Hit	Cand.	T(s)	Hit	Cand.
c432	8.80	27.50	0.07	1	14.60	0.10	1	46.9
c499	82.40	59.30	0.51	1	27.10	1.02	1	54.3
c880	65.50	20.50	0.28	1	17.70	0.54	1	13.7
c1355	66.10	61.60	0.79	1	28.50	1.36	1	53.7
c1908	133.80	85.20	2.15	1	39.70	3.91	1	53.4
c2670	59.30	37.90	0.85	1	29.60	1.86	1	21.9
c3540	96.00	45.50	2.26	1	21.20	4.44	1	53.4
c5315	74.20	79.00	2.64	1	49.80	5.47	1	37.0
c6288	35.90	703.60	18.00	1	56.10	39.50	1	92.0
c7552	101.90	84.60	5.23	1	51.30	11.00	1	39.4

Table 7: Diagnosis Using Combined Model - Multiple Gate Sub. Error

Ckt. Name	Err. Vec.	Region-Based Model			Combined Model			% Red.
		Cand.	T(s)	Hit	Cand.	T(s)	Hit	Cand.
c432	14.43	22.57	0.06	1	11.00	0.12	1	51.3
c499	121.71	31.86	0.35	1	17.14	1.02	1	46.2
c880	51.86	30.14	0.22	1	25.29	0.62	1	16.1
c1355	93.14	37.14	0.55	1	17.86	0.88	1	51.9
c1908	89.43	22.86	0.80	1	15.29	1.00	1	33.1
c2670	38.86	54.43	0.79	1	41.57	1.22	1	23.6
c3540	209.86	17.14	1.93	1	8.00	3.00	1	53.2
c5315	80.43	10.00	1.40	1	7.40	2.32	1	26.0
c6288	24.29	485.86	11.00	1	38.86	20.00	1	92.0
c7552	75.43	73.43	4.53	1	41.00	7.50	1	44.1

obtained is of primary concern, then the combined approach (method 3) would prove highly beneficial. This is because method 3 combines the merits of methods 1 and 2 to achieve superior results. In most cases the number of candidates obtained is the least for this combined method. For instance, in circuit c6288, with 24.29 erroneous vectors, the original region-based technique resulted in 485.86 candidate regions, the Flip-Fanout-Bits method reduced that number to 374.14, Distinguishing X's technique pruned the candidate regions list down to 54.86, and finally the combined approach reported only 38.86 regions that needed to be considered. In other words, 447 of the original list of candidates turned out to be false.

In circuit c880, however, we see that the number of candidates obtained with the combined model is slightly greater than the other two approaches. This can be understood by considering a scenario where a particular region has 2 fanouts. If distinguishing Xs were used and the Xs were complements of each other then it would get eliminated due to which the region could be dropped from the candidate list. However, with the combined model, with one distinguishing X value at one fanout and the other fanout being flipped to a non-controlling value, the X could propagate to a primary output and the region would still be considered a potential error candidate. This situation is rare as indicated from the results of other circuits. For a fanout-free circuit or a circuit with a small number of reconvergent fanouts, Distinguishing Xs may not improve the results, and it would be more beneficial to use the Flip Fanout Bits technique.

5 Conclusion

We have presented efficient algorithms to improve the speed, resolution, and accuracy of arbitrary error diagnosis in combinational circuits. Our techniques aim to eliminate false candidate regions via flipping fanout bits, distinguishing X's, or a combination of the two. Each method enhances the diagnosis in different ways, so as to get the best performance possible. All of our techniques showed improvement over the original region-based approach in terms of both resolution and speed. The combined approach achieved the most significant improvement for most circuits. In some cases, the number of candidate regions is more than halved; up to 92%

Table 8: Comparison of Various Techniques - Multiple Gate Sub. Error

Ckt. Name	Tot. Vec.	Err. Vec.	Tot. Reg.	Region-Based Model		Flip Fanout Model		Distinguishing X Model		Combined Model	
				Cand.	Time(s)	Cand.	Time(s)	Cand.	Time(s)	Cand.	Time(s)
c432	54	14.43	203	22.57	0.06	19.29	0.12	9.86	0.07	11.00	0.12
c499	184	121.71	275	31.86	0.35	17.57	0.95	21.29	0.48	17.14	1.02
c880	178	51.86	469	30.14	0.22	25.00	0.59	24.71	0.40	25.29	0.62
c1355	198	93.14	619	37.14	0.55	29.00	0.90	18.57	0.51	17.86	0.88
c1908	280	89.43	938	22.86	0.80	15.52	1.05	17.14	0.94	15.29	1.00
c2670	102	38.86	1566	54.43	0.79	46.85	1.25	46.14	0.95	41.57	1.22
c3540	350	209.86	1741	17.14	1.93	14.00	3.75	9.57	1.88	8.00	3.00
c5315	162	80.43	2608	10.00	1.40	7.70	2.30	7.43	1.70	7.40	2.32
c6288	40	24.29	2480	485.86	11.00	374.14	30.75	54.86	3.51	38.86	20.00
c7552	221	75.43	3827	73.43	4.53	55.43	9.56	48.00	4.15	41.00	7.50

improvement in diagnostic resolution and 74% speedup over the original region-based diagnosis were achieved with our approaches.

References

[1] A. Veneris and I. N. Hajj, "A fast algorithm for locating and correcting simple design errors in VLSI digital circuits", in *Proc. of GLS-VLSI*, Mar. 1997, pp. 45-50.

[2] A. G. Veneris, S. Venkatraman, I. N. Hajj, and W. K. Fuchs, "Multiple design error diagnosis and correction in digital VLSI circuits", in *Proc. of IEEE VLSI Test Symposium*, Apr. 1999, pp. 58-63.

[3] M. S. Abadir, J. Ferguson and T. E. Kirkland, "Logic design verification via test generation", in *IEEE Transactions on Computer-Aided Design*, Vol. 7, No. 1, January 1988, pp. 138–148.

[4] P. Y. Chung, Y. M. Wang and I. Hajj, "Logic design error diagnosis and correction", in *IEEE Trans. on VLSI Systems*, Vol.2, September 1994, pp. 320-332.

[5] P.-Y. Chung, Y.-M. Wang and I. N. Hajj, "Diagnosis and correction of logic design errors in digital circuits", in *Proc. Design Automation Conf.*, 1993, pp. 503–508.

[6] I. Pomeranz and S. M. Reddy, "On diagnosis and correction of design errors", in *Proc. Intl. Conf. Computer-Aided Design*, 1993, pp. 500–507.

[7] I. Pomeranz and S. M. Reddy, "On error correction in macro-based circuits", in *Proc. Intl. Conf. Computer-Aided Design*, 1994, pp. 568–575.

[8] M. Tomita and H.-H. Jiang, "An algorithm for locating logic design errors", in *Proc. Intl. Conf. Computer-Aided Design*, 1990, pp. 468–471.

[9] M. Tomita, T. Yamamoto, F. Sumikawa and K. Hirano, "Rectification of multiple logic design errors", in *Proc. of ACM/IEEE DAC*, 1994, pp. 212-217.

[10] A. M. Wahba and D. Borrione, "Connection error location and correction in combinational circuits.", in *Proc. of European Design and Test Conference*, March 1997, pp. 235-241.

[11] K. A. Tamura, "Locating functional errors in logic circuits", in *Proc. of DAC'89*, 1989, pp. 185-191.

[12] J. C. Madre, O. Coudert, J. P. Billon, "Automating the diagnosis and rectification of design errors with PRIAM", in *Proc. of ICCAD'89*, 1989, pp. 30-33.

[13] H.-T. Liaw, J.-H. Tsaih and C.-S. Lin, "Efficient automatic diagnosis of digital circuits", in *Proc. Intl. Conf. Computer-Aided Design*, 1990, pp. 464–467.

[14] S.-Y. Huang and K.-T. Cheng, "ErrorTracer: design error diagnosis based on fault simulation techniques", in *IEEE Trans. Computer-Aided Design*, vol. 18, pp. 1341-1352, Sept. 1999.

[15] A. Wahba and D. Borrione, "A method for automatic design error location and correction in combinational logic circuits", in *J. of Electronic Testing: Theory and Applications*, 1996, pp. 113-127.

[16] R. Ubar and D. Borrione, "Single gate design error diagnosis in combinational circuits", in *Proc. of Estonian Acad.Sci.Engng*, 1999,5,1, pp. 3-21.

[17] V. Boppana and M. Fujita, "Modeling the unknown! towards model-independent fault and error diagnosis", in *Proc. Intl. Test Conf.*, 1998, pp. 1094–1101.

[18] V. Boppana, R. Mukherjee, J. Jain and M. Fujita, "Multiple error diagnosis based on xlists", in *Proc. Design Automation Conf.*, 1999, pp. 100–110.

[19] A. Jain, V. Boppana, R. Mukherjee, J. Jain, M. S. Hsiao, M. Fujita, "Testing, verification, and diagnosis in the presence of unknowns", in *Proc. IEEE VLSI Test Symp.*, 2000, pp. 263-269.

[20] A. L. D'Souza and M. S. Hsiao, "Error diagnosis of sequential circuits using region-based model", in *Proc.of IEEE VLSI Design Conf.*, Jan. 2001, pp. 103-108.

[21] J. L. Carter, B. K. Rosen, G. L. Smith and V. Pitchumani, "Restricted symbolic evaluation is fast and useful", in *Proc. of Int. Conf. Computer-Aided Design*, Nov. 1989, pp. 38-41.