

An Improved Approach for Generating Max-Fault Min-Cardinality Diagnoses

Palo Alto Research Center
 3333 Coyote Hill Road, Palo Alto, CA 94304 USA
 dekleer@parc.com

Abstract

Most approaches to model-based diagnosis focus on isolating defective component(s) by performing additional measurements on the defective system. If internal measurements are expensive it is much less costly to change system inputs and observe outputs to provide diagnostic information. In digital circuits this approach is called test-vector generation. Of particular interest are Max-Fault Min-Cardinality (MFMC) observation vectors which result in the maximum number of faults in the minimal cardinality diagnosis. Prior approaches to MFMC generation either used sampling (which is incomplete) or exhaustively enumerate all possible observation vectors (which is computationally impossible). This paper presents a new direct approach to determining MFMC vectors which shows 4-5 orders of magnitude performance improvement over prior algorithms.

1 Introduction

This paper provides a new approach to identifying Max-Fault Min-Cardinality (MFMC) observation vectors as defined in [Feldman, Provan, & van Gemund, 2007]. Identifying such vectors has widespread applicability to analyzing the diagnosability of systems and evaluating the scalability of diagnostic algorithms. Most existing algorithms work best with single or double faults, but do not scale well to higher cardinalities. MFMC observation vectors can easily yield minimal diagnoses of size 20 and above. The main result of this paper is that MFMC observation vectors can be generated with familiar algorithms and well-known circuit properties at 4-5 orders of magnitude performance over prior algorithms.

We first explain the concepts intuitively with in an example. Consider the circuit illustrated in Figure 1. The digital circuit consists entirely of NAND gates. Figure 2 presents the truth table for a NAND gate. Suppose the inputs are observed to be $I1=1, I2=0, I3=0, I4=0$ and $I5=0$. Given those inputs the outputs must be $O1=1$ and $O2=0$. Consider the subset of the circuit consisting of $G3, G4$ and $G6$. Given $I3=0$, $G3$ drives $N3=1$. Given $I5=0$, drives $N4=1$. Given both inputs to the NAND gate $G6$ are 1, it drives its output to $O2=0$. Suppose $O2=1$ is observed. This can only be explained by at least one

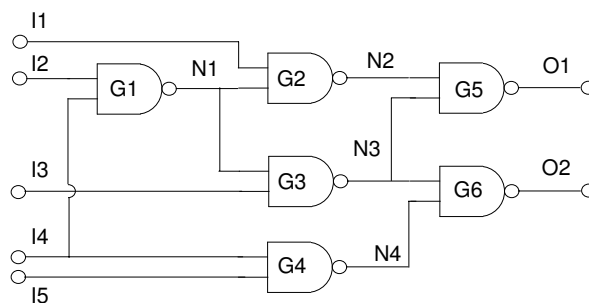


Figure 1: The simplest circuit, c17, from the ISCAS-85 test suite. Inputs are labeled “In,” outputs “On,” gates “Gn,” and corresponding internal nodes “Nn.” All 6 gates are NAND gates.

A			C
B			
A	B		C
0	0		1
0	1		1
1	0		1
1	1		0

Figure 2: Truth table for NAND gate.

of G3, G4 and G6 being faulted. Consider the subset of the circuit consisting of G1, G2 and G5. Given I2=0, G1 drives N1 to 1. Given both N1 and I1 are 1, G2 drives N2 to 0. If one input to a NAND is 0, the output is driven to 1 (O1). Suppose we observe O1=0. Then at least one of G1, G2 or G5 is faulted. The combined observations can only be explained by a double fault (consisting of one of G1,G2 and G5 combined with one of G3, G4 and G6). These are all minimal diagnoses. The observation is an MFMC vector as no other observation has a larger min-cardinality diagnosis. (This will be shown later.)

Prior approaches to identifying MFMC observations utilized either importance sampling, simulated annealing or exhaustive search of the entire space. The first two approaches are not exhaustive and cannot produce absolute bounds. Exhaustive search is impractical for all but the smallest circuits because the complexity is $2^{|INPUTS|+|OUTPUTS|+|COMPS|}$ (where *INPUTS* is the set of inputs, *OUTPUTS* are the set of outputs and *COMPS* is the set of components. This paper proposes an alternative approach to analyze the circuit structure to greatly lower this complexity. This approach has been completely implemented and is based on the GDE implementation and prime implicate algorithms described in [Forbus & de Kleer, 1992]. The result significantly outperforms the previous approach. In future work, we propose to use a more efficient approach outlined in [de Kleer & Williams, 1989] from which we expect further dramatically improved performance.

In this paper we draw all our examples from digital circuits. However, the concepts apply to any system which can be modeled using discrete signals and having distinguished inputs and outputs.

2 Formal Framework

We adopt the formal framework of [de Kleer, Mackworth, & Reiter, 1992] which we briefly summarize here.

Definition 1 A system is a triple $(SD, COMPS, OBS)$ where:

1. *SD*, the system description, is a set of first-order sentences.
2. *COMPS*, the system components, is a finite set of constants.
3. *OBS*, a set of observations, is a set of first-order sentences.

Definition 2 Given two sets of components C_p and C_n define $\mathcal{D}(C_p, C_n)$ to be the conjunction:

$$\left[\bigwedge_{c \in C_p} AB(c) \right] \wedge \left[\bigwedge_{c \in C_n} \neg AB(c) \right].$$

Where $AB(x)$ represents that the component x is ABnormal (faulted).

A diagnosis is a sentence describing one possible state of the system, where this state is an assignment of the status normal or abnormal to each system component.

Definition 3 Let $\Delta \subseteq COMPS$. A diagnosis for $(SD, COMPS, OBS)$ is $\mathcal{D}(\Delta, COMPS - \Delta)$ such that the following is satisfiable:

$$SD \cup OBS \cup \{\mathcal{D}(\Delta, COMPS - \Delta)\}$$

Definition 4 An AB-literal is $AB(c)$ or $\neg AB(c)$ for some $c \in COMPS$.

Definition 5 An AB-clause is a disjunction of AB-literals containing no complementary pair of AB-literals.

Definition 6 A conflict of $(SD, COMPS, OBS)$ is an AB-clause entailed by $SD \cup OBS$.

Definition 7 A minimal conflict of $(SD, COMPS, OBS)$ is a conflict no proper sub-clause of which is a conflict of $(SD, COMPS, OBS)$.

Theorem 1 Suppose that Π is the set of minimal conflicts of $(SD, COMPS, OBS)$, and that Δ is a minimal set such that,

$$\Pi \cup \left\{ \bigwedge_{c \in COMPS - \Delta} \neg AB(c) \right\}$$

is satisfiable. Then $\mathcal{D}(\Delta, COMPS - \Delta)$ is a minimal diagnosis.

This result forms the basis of most model-based diagnosis algorithms: (1) compute the minimal conflicts, (2) compute the diagnoses. For the purposes of MFMC computation we are only interested in minimal diagnoses:

Definition 8 A diagnosis $\mathcal{D}(\Delta, COMPS - \Delta)$ is a minimal diagnosis iff for no proper subset Δ' of Δ is $\mathcal{D}(\Delta', COMPS - \Delta')$ a diagnosis.

We also assume the usual axioms for equality and arithmetic are included in *SD*. For this paper, we assume weak fault models or the Ignorance of Abnormal Behavior property.

3 Representing Component Behaviors as Propositional Clauses

An inverter can be modeled by:

$$INVERTER(x) \rightarrow \left[\neg AB(x) \rightarrow [in(x) = 0 \equiv out(x) = 1] \right].$$

A particular inverter G is thus modeled by the formula

$$\neg AB(G) \rightarrow [in(G) = 0 \equiv out(G) = 1],$$

which is modeled by the following clauses (prime implicates):

$$AB(G) \vee out = 0 \vee in = 1,$$

$$AB(G) \vee out = 1 \vee in = 0.$$

The NAND gate G5 (of Figure 1) with inputs N2 and N3 and output O1 is modeled by the clauses:

$$AB(G5) \vee N2 = 0 \vee N3 = 0 \vee O1 = 0,$$

$$AB(G5) \vee N2 = 1 \vee O1 = 1,$$

$$AB(G5) \vee N3 = 1 \vee O1 = 1.$$

4 MFMC definitions

Following [Feldman, Provan, & van Gemund, 2007] we define:

Definition 9 The cardinality of diagnosis $\mathcal{D}(\Delta, COMPS - \Delta)$ is $|\Delta|$.

Definition 10 A diagnosis $\mathcal{D}(\Delta, COMPS - \Delta)$ is a minimal cardinality diagnosis iff for no other diagnosis $\mathcal{D}(\Delta', COMPS - \Delta')$ is $|\Delta'| < |\Delta|$.

Clearly all minimal cardinality diagnoses are minimal diagnoses. However, the converse does not hold.

Definition 11 $MaxCard(SD)$ is the maximum cardinality of all possible minimal cardinality diagnoses for all possible OBS.

Definition 12 An MFMC observation is any OBS such that the minimal cardinality diagnosis is equal to $MaxCard(SD)$.

In this paper we assume the usual definitions of well-formedness (system forms a DAG, only one component drives any node, all component inputs are driven, components have at least one input and at most one output).

We can now state the analysis of the circuit of Figure 1 more formally. $COMPS = \{G1, G2, G3, G4, G5\}$. $OBS = \{I1 = 1, I2 = 0, I3 = 0, I4 = 0, I5 = 0, O1 = 0, O2 = 1\}$. There are 3 minimal conflicts:

$$\begin{aligned} &AB(G4) \vee AB(G5) \vee AB(G6), \\ &AB(G1) \vee AB(G2) \vee AB(G5), \\ &AB(G3) \vee AB(G4) \vee AB(G6). \end{aligned}$$

One of the minimal diagnoses is:

$$\begin{aligned} &\neg AB(G1) \wedge \neg AB(G2) \wedge AB(G3) \wedge \\ &\neg AB(G4) \wedge AB(G5) \wedge \neg AB(G6). \end{aligned}$$

5 Two Bounds on MaxCard

There are two important upper bounds on $MaxCard$. The first is obvious, but important:

Theorem 2 $MaxCard(SD)$ is bounded by $|COMPS|$.

More importantly:

Theorem 3 $MaxCard(SD)$ is bounded by the number of outputs of SD .

Proof sketch. In a well-formed circuit, every possible input combination is possible. Thus any conflict must involve at least one component driving an output. So the conjunction of all output components faulted will always be a diagnosis. It may not be minimal and thus the number of outputs is only an upper bound on $MaxCard(SD)$.

The ‘‘converse’’ is not true: $MaxCard(SD)$ is not bounded by the number of inputs. Unfortunately, $MaxCard(SD)$ may be far less than the number of outputs. Nevertheless, this second bound is critical to the efficiency of the algorithm described later: when searching for possible MFMC observations, those with a number of incorrect outputs less than the current best estimate of $MaxCard(SD)$ can be skipped.

obs	MC	obs	MC
11111	0	01111	1
11110	1	01110	0
11101	1	01101	2
11100	2	01100	1
11011	2	01011	0
11010	1	01010	1
11001	1	01001	1
11000	0	01000	2
10111	2	00111	0
10110	1	00110	1
10101	1	00101	1
10100	0	00100	2
10011	1	00011	2
10010	2	00010	1
10001	0	00001	1
10000	1	00000	0

Table 1: Minimum cardinality for each possible observation vector for subtractor circuit of Figure 3. Each obs = $[x, y, p, b, d]$.

6 Sources of Complexity

A brute force approach to determining $MaxCard(SD)$ is to use a diagnosis engine to compute a minimal cardinality diagnosis for each possible observation. Consider the subtractor circuit of Figure 3 [Feldman, Provan, & van Gemund, 2007] having 7 components, 3 inputs and 2 outputs. Table 1 lists the minimum cardinality for the given observation vector. $MaxCard(SD)$ for the circuit is 2 which is the upper bound stipulated by Theorem 3.

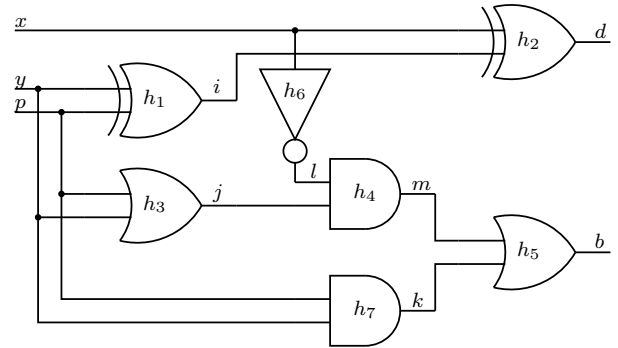


Figure 3: Subtractor with inputs x, y and p and outputs d and b . h_1 and h_2 are exclusive-or gates, h_3 and h_5 are OR gates, h_4 and h_7 are AND gates, and h_6 is an inverter.

Although C17 has fewer components it is more complicated for a brute force algorithm. It has 5 inputs, 2 outputs, and 6 gates. The space to be searched for identifying $MaxCard(c17)$ is $2^{5+2+6} = 8192$ cases.

7 Reducing the number of components

The complexity of searching for $MaxCard(SD)$ is exponential in $|COMPS|$ Therefore, any reduction in number of

components which does not affect $MaxCard(SD)$ provides significant computational advantage. Consider again the circuit of Figure 1. Gates G2 and G5 can be combined into one gate G2-G5 with one output O1 and three inputs I1, N1 and N3. This combination corresponds to the top dashed region of Figure 4. The only other combinable components is the bottom region. No other combination is possible.

The intuition why G2 and G5 can be combined is as follows. The internal node N2 is not observable. Therefore, any conflict involving G2 will necessarily involve G5 as well. If $I1=1$ and $N1=1$, then if G2 and G5 were operating correctly $O1=1$. Observation $O1=0$ yields the conflict $\{G2, G5\}$. (Notice that G2 does not necessarily occur in every conflict involving G5. If $N3=0$, observing $O1=0$, yields the conflict $\{G5\}$.) The same pattern arises in every possible conflict involving G2. Therefore, for the purposes of calculating $MaxCard(SD)$, one can replace G2 and G5 with one composite component thereby reducing the complexity of searching for $MaxCard(SD)$.

In the case of stuck-at faults, there are many algorithms to collapse faults [Bushnell & Agrawal, 2000]. In general:

Theorem 4 Any connected subset of components of SD having only one output can be replaced by a single composite component with equivalent behavior without changing $MaxCard(SD)$.

Identifying all such subsets is typically too expensive. Fortunately, many such equivalent subsets can be identified very quickly:

Theorem 5 If component A drives internal node n and that node is an input into only one component B, then A, B and n can be replaced with a single composite component with logically equivalent behavior without changing $MaxCard(SD)$.

Proof sketch. By construction, the input-output behavior of the circuit remains unchanged with the replacement. As no other component is connected to n, every conflict involving A must also include B. Therefore, B would appear in any diagnosis in which A appears. Thus replacing A and B with a single composite component does not change $MaxCard(SD)$.

For example component G2, node N2 and component G5 of Figure 4 can be replaced with a single component with inputs I1, N1 and N3 and output O1.

By iteratively applying the preceding theorem the number of components can be significantly reduced without changing $MaxCard(SD)$.

The combined G2-G5 (G) component is described by the following 4 clauses (a single NAND gates is encoded by 3 clauses):

$$AB(G) \vee N1 = 1 \vee N3 = 0 \vee O1 = 0,$$

$$AB(G) \vee I1 = 1 \vee N3 = 0 \vee O1 = 0,$$

$$AB(G) \vee N1 = 0 \vee I1 = 0 \vee O1 = 1,$$

$$AB(G) \vee N3 = 1 \vee O1 = 1.$$

Table 2 lists the reduction in component count after applying the reduction rule. The circuits are commonly known

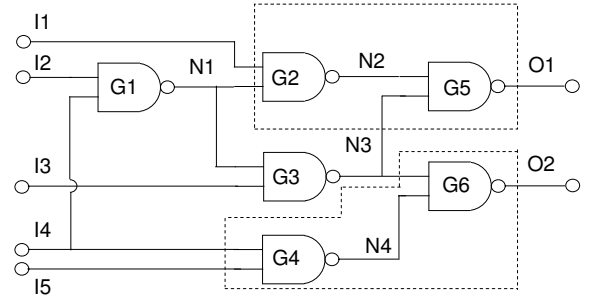


Figure 4: Outlined regions are combinable components.

Table 2: Reduction in component count using the Prime Implicate reduction rule.

circuit	original	reduced
7485	33	15
74181	73	20
74182	19	6
74283	36	14
c17	6	4
c432	160	59
c499	202	58
c880	383	77
c1355	546	58
c1908	880	160
c2670	1193	169
c3540	1669	353
c6288	2416	1456
c7552	3512	545

74nnn circuits or are from the ISCAS 85 [Brglez & Fujiwara, 1985] benchmarks. We use the prime implicate algorithm from [Forbus & de Kleer, 1992] on the clausal form of the component models.

The cones formulation of [Siddiqi & Huang, 2007] yields an almost identical reduction in component count but with a much simpler, fault collapsing, approach. We cannot compare the timings of the underlying diagnostic algorithms or the resulting algorithms to compute MFMC vectors directly.

Our current algorithm cannot complete the MFMC computation for any of the non-trivial ISCAS benchmarks. However, we do know lower bounds on all the benchmarks from partial runs and they typically scale with the number of outputs. For example, C2670 has 140 outputs and an MFMC of at least 20 (probably much higher).

8 Reducing the Number of Inputs and Outputs

Important properties which improve running time dramatically on many circuits are:

- Suppose SD can be divided into two SD_1 and SD_2 such they share no internal nodes. In this case, $MaxCard$

Table 3: The columns are: circuit, original complexity, MaxCard(SD), our wall time, cited wall time reported. Timings are on Common Lisp and a multi-core 3GHz PC and include prime implicate construction time.

circuit	complexity	MC	time	cited
7485	1.4×10^{14}	3	1ms	196.9s
74181	4.0×10^{28}	7	60m	impossible
74182	8.6×10^9	5	1ms	53.4s
74283	1.1×10^{15}	5	40ms	371.9s
c17	8.2×10^3	2	2ms	not given

calculation is greatly simplified: $MaxCard(SD) = MaxCard(SD_1) + MaxCard(SD_2)$.

- Let $SD|_{i=x}$ represent the system SD with input i set to x and the resulting circuit simplified by removing irrelevant components. $MaxCard(SD) = Max(MaxCard(SD|_{i=0}), MaxCard(SD|_{i=1}))$.
- If all of the inputs of a component are fed directly from the inputs (and no other component is driven by those inputs), then this set of inputs can be reduced to one and the component replaced by a buffer.
- Under many conditions, inputs can be discarded without affecting $MaxCard(SD)$. For example, $I1$ can be discarded from $c17$ and $G2$ replaced by an inverter.

9 Basic Algorithm

The algorithm is based on the Common Lisp code of [Forbus & de Kleer, 1992] and exploits all the prior concepts described earlier.

1. Inputs and outputs are reduced.
2. The number of components is reduced by replacing subsets of components with their prime implicates (unless the number of prime implicates exceeds the number of clauses in the original circuit which can happen for the larger ISCAS circuits).
3. Of the resulting system(s), an exhaustive search is made. With two important modifications: (1) only consider those observations whose number of incorrect outputs is greater than the current best estimate of $MaxCard$, and (2) cache the ATMS conflicts of prior observations.

Table 3 presents the performance compared to those reported in [Feldman, Provan, & van Gemund, 2007]. The algorithm presented in [Feldman, Provan, & van Gemund, 2007] is based on a GDE-like LTMS-based algorithm. The algorithm of this paper is a version of GDE using a conventional ATMS. Initial indications are that GDE benefits significantly from the caching capabilities of the ATMS.

The current algorithm based on [Forbus & de Kleer, 1992] constructs all minimal conflicts and then constructs the minimal diagnoses from the minimal conflicts. This approach constructs far more conflicts than are necessary to find one minimal diagnosis. If there are a large number of minimal diagnoses, most of this is useless work. C6288 can never be analyzed by this approach as it yields an exponential number

Table 4: Distribution of observation vectors for each minimal diagnosis cardinality for 74182. As $MaxCard(SD) = 5$, 480 are MFMC observation vectors.

cardinality	MFMC count
0	512
1	2592
2	5184
3	5120
4	2496
5	480
Total	16384

of minimal conflicts. These result from its inherently parallel structure illustrated in Figure 5.

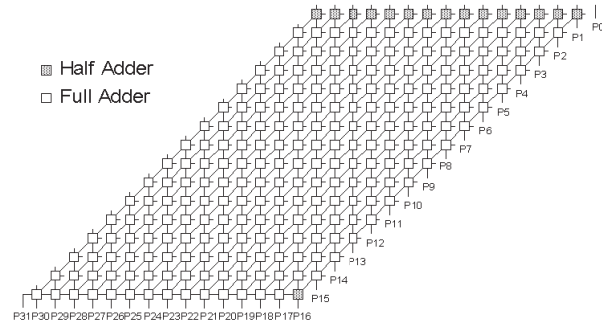


Figure 5: The most difficult to analyze circuit from the ISCAS-85 test suite. It is a 16 by 16 bit parallel multiplier built out of half and full adders.

10 Constructing the MFMC Observation

Once $MaxCard(SD)$ has been identified, the full observation is constructed by inverting the simplifications described earlier. This is always very fast.

One reason MFMC observations are difficult to find is that there are not that many of them. Table 4 lists the number of observation vectors for each minimal cardinality. For the 74182 only 3 % are MFMC observations.

11 Conclusions and Future Work

The approach in this paper has been completely implemented and built upon the Common Lisp code described in [Forbus & de Kleer, 1992]. It outperforms previous approaches and we expect further improvements when implemented with a more modern efficient model-based diagnosis algorithm which does not compute all minimal diagnoses. As this task is easily parallelizable (e.g., using MapReduce [Dean & Ghemawat, 2004]) we plan to use a cluster for future results.

Combining the ATMS-based GDE with the stochastic search could enable finding high cardinality observation vectors for large circuits relatively quickly. Such high cardinality observation vectors are important as they provide high fault coverage for ATPG applications.

12 Acknowledgements

Many conversations with Alex Feldman helped clarify many of the concepts.

References

- [Brglez & Fujiwara, 1985] Brglez, F., and Fujiwara, H. 1985. A neutral netlist of 10 combinational benchmark circuits and a target translator in fortran. In *Proc. IEEE Int. Symposium on Circuits and Systems*, 695–698.
- [Bushnell & Agrawal, 2000] Bushnell, M. L., and Agrawal, V. D. 2000. *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Boston: Kluwer Academic Publishers.
- [de Kleer & Williams, 1989] de Kleer, J., and Williams, B. 1989. Diagnosis with behavioral modes. In *Proc. 11th IJCAI*, 1324–1330.
- [de Kleer, Mackworth, & Reiter, 1992] de Kleer, J.; Mackworth, A.; and Reiter, R. 1992. Characterizing diagnoses and systems. *Artificial Intelligence* 56(2-3):197–222.
- [Dean & Ghemawat, 2004] Dean, J., and Ghemawat, S. 2004. Mapreduce: Simplified data processing on large clusters. 137–150.
- [Feldman, Provan, & van Gemund, 2007] Feldman, A.; Provan, G.; and van Gemund, A. 2007. Generating manifestations of max-fault min-cardinality diagnoses. In Biswas, G.; Koutsoukos, X.; and Abdelwahed, S., eds., *Working Papers of the Eighteenth International Workshop on Principles of Diagnosis*. 83–90.
- [Forbus & de Kleer, 1992] Forbus, K. D., and de Kleer, J. 1992. *Building Problem Solvers*. Cambridge, MA: MIT Press.
- [Siddiqi & Huang, 2007] Siddiqi, S., and Huang, J. 2007. Hierarchical diagnosis of multiple faults. *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)* 581–586.