Consiglio Nazionale delle Ricerche

Technical Report

Self-Validating Diagnosis of Hypercube Systems

Paolo Santi, Piero Maestrini

I E I: B4-12-04-99

I.E.I.
ISTITUTO DI
ELABORAZIONE DELLA
INFORMAZIONE

Pisa

Self-Validating Diagnosis of Hypercube Systems

Paolo Santi and Piero Maestrini
Department of Computer Science, University ofPisa
and Istituto di Elaborazione dell'Informazionedel CNR, Pisa
santi@iei.pi.cnr.it, maestrini@iei.pi.cnr.it

Abstract

A novel approach to the diagnosis of hypercubes, called Self-Validating Diagnosis, is introduced. An algorithm based on this approach, called SVD algorithm, is presented and evaluated. Given any fault set and the resulting syndrome, the algorithm returns a diagnosis and a syndrome-dependent bound, T_σ, with the property that diagnosis is correct (although possibly incomplete) if the actual number of faulty units is less than T_σ. The average of T_σ is very large and the diagnosis is almost complete even when the percentage of faulty units in the system approaches 50%. Moreover, the diagnosis correctness can be validated deterministically by individually probing a very small number of units. These results suggest that the SVD algorithm is suitable for applications requiring a large degree of diagnosability, as it is the case of wafer-scale testing of VLSI chips, where the percentage of faulty units may be as large as 50%.

Categories and Subject Descriptors: B.8.1 [**Performance and Reliability**]: Reliability, Testing and Fault-Tolerance; C.4 [**Performance of Systems**]: Fault-Tolerance, Reliability, Availability and Serviceability.
General Terms: Algorithms, Reliability

1. Introduction

Self-diagnosis, also called *system-level diagnosis*, was introduced by Preparata, Metze and Chien in 1967 ([1]). In self-diagnosis, a system composed of several units connected by bi-directional links is diagnosed using the information provided by tests performed by the units comprising the system itself. Every test involves two units, called the *testing* and the *tested* unit, which must be interconnected. Every unit in the system may be in a *faulty* or *non-faulty* state. The set of faulty units is called the *Actual Fault Set* (AFS in the following). Only permanent faults are considered, i.e. units retain their (faulty or non-faulty) state throughout the diagnosis process. Essentially, a test is performed as follows:

- the testing unit requests the tested unit to run a job;
- the tested unit returns a result to the testing unit;
- the testing unit compares the actual and the expected results and provides a binary test outcome. The outcome is 0 if the actual and the expected results match ("the test passes"), 1 otherwise ("the test fails").

The set of tests utilized for the purpose of diagnosis is defined by digraph $DG=(V,E)$, where V is the set of units and $E=\{[u,v]$:unit u tests unit $v\}^{(1)}$. DG is called the *diagnostic graph* of

(1) Notations (u,v) and $[u,v]$ denote and undirected edge connecting u and v, and a directed arc from u to v, respectively.

the system. The test of unit v performed by unit u with outcome x is denoted by $u \xrightarrow{x} v$. In the following we will also use notation $u \xleftrightarrow{y \quad x} v$ to collectively denote the test of unit v performed by unit u with outcome x and the test of u performed by v with outcome y. Given an AFS, the set of the outcomes of all tests in *DG* is called *syndrome* and is denoted σ.

The reliability of any test outcome depends on the state of the testing unit. Different hypotheses upon the test outcomes returned by faulty units lead to different *invalidation rules*, and consequently to different *diagnostic models*. The most widely used diagnostic model is the PMC model introduced in [1], which assumes arbitrary test outcomes for tests performed by faulty units. The invalidation rule of the PMC model is shown in Table 1.

Tester unit's state	Tested unit's state	Test outcome
non-faulty	non-faulty	0
non-faulty	faulty	1
faulty	non-faulty	0 or 1
faulty	faulty	0 or 1

Table 1. Invalidation rule in the PMC model

It is immediate from Table 1 that any given fault set may yield different syndromes; conversely, any given syndrome may derive from different fault sets. A *diagnosis algorithm* is an algorithm which, given a syndrome, provides a *diagnosis* of the system, i.e. classifies each units as *faulty*, *non-faulty* or *suspect*. The diagnosis algorithm is usually executed on an external and reliable computer, called the *diagnoser* (*centralized diagnosis*). The diagnosis algorithm is *correct under syndrome* σ if, given σ, no faulty unit is classified as non-faulty and no non-faulty unit is classified as faulty. The diagnosis algorithm is said to be *correct* if it is correct under all possible syndromes. The diagnosis algorithm is *complete under syndrome* σ if, given σ, every unit in the system is classified as either faulty or non-faulty (i.e. there are no suspect units). The diagnosis algorithm is said to be *complete* if it is complete under all possible syndromes.

A diagnosis which is both correct and complete is called *one-step diagnosis*. It is known ([1]) that one-step diagnosis is possible only if the cardinality of the AFS does not exceed the *diagnosability*, a parameter of the diagnostic graph which is limited above by the minimum of the nodes in-degrees. This requirement may be relaxed if it is admissible that the diagnosis be divided into several test and repair phases, where every phase identifies at least one faulty unit, which is repaired or replaced, thus reducing the number of faulty units. Under this approach, called *sequential diagnosis*, the system diagnosability is increased, at the expense of longer time requirements to perform the diagnosis.

One-step diagnosis can be trusted only if it can be reliably assumed that the cardinality of the AFS is not above the diagnosability. One alternative approach, called *probabilistic diagnosis*, is based on diagnosis algorithms which provide a (usually complete) diagnosis, whose correctness is evaluated probabilistically. Using this approach, the probability of providing correct diagnosis may be close to 1 even when the cardinality of the AFS is considerably above the diagnosability.

A natural domain of application of system-level diagnosis is provided by massively parallel computers, where large numbers of processors, or units, cooperate to perform computations.

In most cases, the units are linked by point-to-point connections, each to a limited number of neighbors and using a regular scheme. As the number of units increases, the probability of faults in the system increases as well, and the propagation of errors may corrupt the entire computation. System-level diagnosis may be exploited to identify faulty units and, in combination with replacement and recovery, may be used as a relatively low cost alternative to error codes or to hardware redundancy.

Many different interconnection schemes have been proposed for massive parallel systems. The hypercube is one of the most widely used, due to its structure which eases both process allocation and routing tasks. For example, the Intel iPSC [11] and the nCUBE [12] machines are structured as hypercubes.

Another promising application area of system-level diagnosis is the wafer-scale testing of VLSI chips during the manufacturing process ([2],[3],[4]). The goal of the diagnosis is to select good chips and to discard faulty chips in an early stage of the manufacturing process, thus avoiding the costly process of bonding and packaging in the case of faulty elements. In order to implement the diagnosis process, the wafer is seen as a regular arrangement of identical components, which should be complemented with communication links, power supply, ground, clock and so on. Application to wafer-scale testing of integrated circuits poses a serious challenge to the theory of system-level diagnosis, since:

- the expected fraction of faulty chips in the wafer can be as large as 50%, depending on the complexity of the chip;
- in order to be feasible, the interconnection structure to be implemented on the wafer has to be regular and the degree of the nodes has to be small.

Among the interconnection schemes which appear suitable for wafer-scale testing, hypercube connection deserves special attention. In fact, as proved by Scheinerman ([16]) and Blough ([17]), probabilistic diagnosis is *almost certainly correct* (i.e. correct with probability approaching 1 as the size of the system increases) if the in-degree of nodes in the diagnostic graph is order of *logN*, where *N* is the size of the system[(2)].

Self-diagnosis of hypercube-structured systems has been studied quite extensively. Kavianpour and Kim ([7],[8]) proposed four different strategies for the diagnosis of hypercubes, based on both the one-step and the sequential diagnosis approaches. They showed that the diagnosability of the system increases by allowing the replacement of at most one possibly non-faulty unit. Feng, Bhuyan and Lombardi ([9]) introduced two algorithms (called HADA and IHADA) to perform adaptive diagnosis of hypercubes. They deserved special attention at reducing the cost associated to the diagnosis process. Khanna and Fuchs ([10]) introduced an algorithm for sequential diagnosis of hypercubes and they evaluated its diagnosability and cost.

In this paper we introduce a novel approach to the diagnosis of hypercube systems. This approach, called *self-validating diagnosis* (SVD), may be easily extended to other regular structures. We present a diagnosis algorithm whose correctness is proved deterministically if the total number of faults in the system is less than a syndrome-dependent bound, T_σ, which is asserted by the algorithm itself. The diagnosis correctness can be trusted if the cardinality of

[(2)] In this paper, all logarithms are to base 2.

the AFS is expected to be below T_σ; if not, it can be checked by individually probing N- T_σ+1 units which were diagnosed as non-faulty. The expected values of T_σ have been evaluated by means of simulation, as well as the degree of completeness of the diagnosis. The average of T_σ resulted very close to N as long as the cardinality of the AFS is not above $N/2$, and the diagnosis resulted almost complete under the same condition. Both results agree with the probabilistic evaluations byScheinerman andBlough.

This paper is structured as follows. In Section 2 a Self-Validating Diagnosis algorithm for hypercube systems is introduced and the bound T_σ is derived. In Section 3 the completeness of the algorithm is considered. In Section 4 the performance of the SVD algorithm introduced in Section 2 is evaluated by means of simulation. In Section 5 the SVD algorithm is compared with previous algorithms for hypercubes. Finally, Section 6 draws some conclusions.

2. The diagnosis algorithm

In this section we introduce a self-diagnosis algorithm for hypercube-connected systems which validates its own diagnosis by asserting a syndrome-dependent bound for correctness. Because of this feature, the algorithm is called SVD algorithm, or SVDA.

An hypercube system can be represented by an undirected graph $H=(V,E)$, where vertices correspond to units and edges represent point to point interconnections. The number $\#V$[3] of vertices is $N=2^n$ and the number $\#E$ of edges is $n2^{n-1}$, for some integer n greater than 1. Integer n is called the dimension of the hypercube. Every $v \in V$ is labeled with a n digit binary number denoted $lab(v)$. Vertices are connected based on the Hamming distance of their labels, denoted d_H: edge (u,v) exists if and only if $d_H(lab(u),lab(v))=1$. In the following, words vertex and unit, as well as words edge and interconnection, will be used indifferently. Since it is assumed that any two adjacent units test each other, the set of tests utilized for the purpose of diagnosis is defined by the arcs in the digraph$DH=(V,E')$, where $E'=\{[u,v]: (u,v) \in E\}$.

The SVD algorithm is divided into three steps: *Local Diagnosis*, *Fault-Free Core Identification* and *Augmentation*. Every step of the algorithm is described in a separate subsection. In the last subsection the complexity of the algorithm is evaluated. The algorithm is summarized in Table 2.

2.1 Local Diagnosis

The objective of local diagnosis is to classify every unit as either F (faulty unit) or D (dual unit), or S (suspect unit). F-units are known to be faulty. D-units are defined in disjoint pairs with the property that, for every pair, at least one unit is faulty. The state of S-units remains unidentified and, in most cases, will be determined in the subsequent steps of the algorithm. The sets of units classified S, F or Dare denoted by *S*, *F* and *D*, respectively.

Unit classification is based upon the following Lemma:

Lemma 1: *Let u and v be adjacent units in H:*

a) if $u \xleftrightarrow{1\ \ 0} v$ *then u is faulty;*

b) if $u \xleftrightarrow{1\ \ 1} v$ *then at least one unit between u and v is faulty;*

c) if $u \xleftrightarrow{0\ \ 0} v$ *then u and v are in the same state (that is, both units are*

(3) Given any set H, notation $\#H$ denotes its cardinality.

either faulty or non-faulty);

d) if v is faulty and $u \xrightarrow{0} v$ *then u is faulty.*

Proof: Immediate from the invalidation rule of Table 1. ❑

Local Diagnosis proceeds as follows. Initially F-units are identified using statements a) and d) of Lemma 1. Then D-units (statement b) of Lemma 1) are defined in disjoint pairs, with the property that in every pair at least one unit is faulty. Since maximizing the cardinality of set D tends to increase the average of bound T_σ defined in step 2 of the algorithm, pairs of D-units are determined as a maximum matching ([19]) on the subgraph H' of H induced by the vertex set $V' \subseteq V$, which contains all vertices incident to edges labeled with outcomes 11. Finally set S is defined as the set of units which were classified as neither F nor D in the preceding steps.

2.2 Fault-Free Core Identification

In this step, the subgraph H'' of H induced by the units classified S is considered, and the vertex set of every connected component of H'' is defined as an *aggregate*. Isolated S-units (that is, S-units which are adjacent only to F-units or to D-units) are trivial aggregates. The collection of all the aggregates is denoted $\{A_1, A_2, \ldots, A_k\}$.

It is immediate from Table 1 that all units in an aggregate are in the same state. However, we cannot decide whether the actual state is faulty or non-faulty.

Letting α be the maximum of $\#A_1, \#A_2, \ldots, \#A_k$, the Fault-Free Core (denoted FFC) is defined as the union set of all the aggregates of cardinality α. The syndrome-dependent bound T_σ is also defined at this step, as $T_\sigma = \#F + \#D/2 + \alpha$.

The FFC plays a fundamental role in the diagnosis algorithm since, under certain conditions to be stated in subsection 2.4, all units in the FFC are non-faulty. This also implies that tests performed by units in the FFC are completely reliable.

2.3 Augmentation

The third step of the algorithm is aimed at augmenting sets FFC and F, by identifying the state of (as many as possible) units in sets D and $\bigcup_{i=1,..,k} A_i - FFC$.

Augmentation is based on the tests of units in set $V-(FFC \cup F)$ performed by units in FFC, and on tests of units in set F performed by units in $V-(FFC \cup F)$. For every test $u \xrightarrow{x} v$, with $u \in FFC$ and $v \in V-(FFC \cup F)$:

- if $x=0$, then v is identified as non-faulty and set FFC is redefined as $FFC \cup \{v\}$. If v belongs to some aggregate A_i, then all the units belonging to A_i are identified as non-faulty and included in FFC.
- if $x=1$, then v is identified as faulty and set F is redefined as $F \cup \{v\}$. If v belongs to some aggregate A_i, then all the units belonging to A_i are identified as faulty and included in F.

For every test $u \xrightarrow{y} v$, with $u \in V-(FFC \cup F)$ and $v \in F$:

- if $y=0$, then u is identified as faulty and set F is redefined as $F \cup \{u\}$. If u belongs to some aggregate A_i, then all the units belonging to A_i are identified as faulty and included in F.

At the end of this step, the set S of suspect units is redefined as $V-(FFC \cup F)$.

2.4 Self-Validation

The diagnosis algorithm returns sets *F*, *FFC* and *S* as defined in step 3 (Augmentation). If set *S* is non-empty, the diagnosis is incomplete. The SVDA also returns the maximum α of the aggregate cardinalities, the syndrome-dependent bound T_σ and it validates its own output by asserting that the diagnosis is correct if $\alpha>0$ and the cardinality of the AFS is less than T_σ.

The validation is based on the following theorem.

Theorem 1. *Given any syndrome σ, the Fault-Free Core is non-empty and completely fault-free, provided $\alpha>0$ and the total number of faults in the system is less than T_σ.*

Proof. Consider sets *FFC*, *F* and *D* as defined in steps 1 and 2 of the SVD algorithm. The hypothesis $\alpha>0$ ensures that there exists at least one aggregate and, consequently, set *FFC* is non-empty. Suppose that some unit in *FFC* is faulty. Since set *FFC* is defined as the union set of all the aggregates of cardinality α, at least one such aggregate must be completely faulty. Recalling that the number of faulty units in *F* is #*F* and the number of faulty units in *D* is at least #*D*/2, the total number of faulty units in the system is at least $\#F+\#D/2+\alpha$. This is a contradiction, since the total number of faults in the system is less than $T_\sigma = \#F + \#D/2 + \alpha$ by hypothesis. ❑

If set *FFC* is non-empty and completely fault-free, the diagnosis provided by the SVD algorithm is correct. In fact, units assigned to set *F* during Local Diagnosis are guaranteed to be faulty by statement a) of Lemma 1, and units added to *FFC* or to *F* during Augmentation are guaranteed to be non-faulty or faulty, respectively, by the reliability of tests performed by units in *FFC*, or by statement d) of Lemma 1.

From this reasoning, the following result is immediate:

Corollary 1: *The SVD algorithm is correct under syndrome σ if $\alpha>0$ and the cardinality of the AFS is less than T_σ.*

Observe that the existence of at least one aggregate (that is, $\alpha>0$) is guaranteed if $N_F=\#AFS<N/2$, because the admissible classifications of non-faulty units are S or D, and the number of non-faulty units in set *D* is at most $\#D/2\leq N/2$.

Corollary 1 states that the diagnosis returned by the SVDA may be incorrect only if $N_F\geq T_\sigma$. This means that it is reasonable to trust the diagnosis whenever the expected number of faults is considerably below T_σ. This situation actually occurs in most cases of interest, as it is seen from simulation results reported in Section 4: in fact, the average of T_σ always resulted above N-2 as long as $N_F\leq N/2$.

A deterministic validation of the diagnosis correctness is also suggested from Corollary 1. To this purpose, it is sufficient to test individually as few as $N- T_\sigma+1$ units, among those declared non-faulty by the algorithm. If all of them are confirmed to be non-faulty, then must be $N_F<T_\sigma$ and the diagnosis is correct. This is a valuable result since, as simulation shows, the expected number of units to be tested is at most 2 if $N_F\leq N/2$, regardless of the size of the system.

2.5 *Complexity of SVD algorithm*

In this subsection, the time complexity of SVD algorithm is evaluated as follows:

Theorem 2. *The time complexity of the algorithm SVD is $O(N^{3/2}logN)$.*

Proof. In the order, the complexity of the algorithm is the maximum of complexities of the operations executed in a step. Consider an implementation of the algorithm which, for every unit, defines boolean flags associated to sets F, D, S and FFC. If unit u belongs to a set, the corresponding flag is set to TRUE, otherwise it is FALSE. Under this implementation, determining whether a unit belongs to a given set, adding a unit to, or removing a unit from a set can be done in O(1). To identify aggregates, an integer value is associated to each unit: this value is k if the units belongs to A_k, 0 otherwise. Hence all the operations $\in$, $\notin$, $\cup$ and - between a unit and an aggregate can be done in O(1). Moreover, an auxiliary queue of units is utilized throughout the algorithm.

Let us consider the first step of the SVDA. Initially, set F is constructed in O($NlogN$) by scanning all the edges of graph H and considering the 01 test outcomes (statement a) of Lemma 1). In order to detect increase the cardinality of set F, faulty units are inserted in queue Q, from which they are removed one at a time. For every unit u removed from Q, the *zero-ancestors*, which are identified among the neighbors of u in set V-F according to statement d) of Lemma 1, are added to set F and inserted in Q. As soon as a new unit is identified as belonging to F, it is inserted in the queue. Since every unit is inserted in Q at most once, and every unit has $logN$ neighbors, this operation also requires time O($NlogN$).

Set D is determined by finding a maximum matching on subgraph H' of H, induced by the vertex set $V' = \{u: u \overset{1 \quad 1}{\longleftrightarrow} v$ for some $v \neq u\}$.Since graph H is bipartite ([5]) and H' is a subgraph of H, H' is also bipartite, and a maximum matching for H' can be determined in $O(N^{3/2}logN)$ ([19]).

The set S is easily constructed in time O(N) by scanning the units.

In step 2, aggregates are identified in time O($NlogN$) by determining the connected components of subgraph H'' of H induced by set S ([20]). Set FFC is defined as the union set of all the aggregates of maximum cardinality.

In step 3, set FFC is augmented by inserting in queue Q units in set V-($F \cup FFC$) which are adjacent to units in the FFC. Units in the queue are removed one at a time, and added to set FFC or F depending on the outcome of any test performed by units in set FFC. In the former case, the neighbors of the unit extracted which are in set V-($F \cup FFC$) are also inserted in Q. Since every unit is inserted in Q at most once, the time complexity of this operation is O($NlogN$). The augmentation of set F with units which are zero-ancestors of units in F is also executed in time O($NlogN$) using a similar technique, where queue Q is used to contain units in set S which are neighbors of units in F.

❑

The complexity of the algorithm reported in Table 2, which is dominated by the operation of determining a maximum matching on subgraph H', could be reduced to

O(*NlogN*) by using a simple heuristic to define a (non maximum) bipartite matching. From simulation, it was seen that this simplification would cause a negligible reduction of the average of T_σ.

```
Local Diagnosis;
 begin
   F:= {v:∃u∈V such that v <-1 0-> u };
   while (((∃u∈F) and (∃v∈V-F))such that v -0-> u) do F:=F∪{v};
   D:=∅;
   E':={(u,v)∈E: u <-1 1-> v};
   H':=SubGraphE(H,E');  {Given a graph H and a set E' of edges, function SubGraphE
                          returns the subgraph H' of H of vertex set V'={u∈V:(u,v)∈E' for
                          some v}}
   D:=DualSet(H'); {DualSet(H') is a function which returns the set D of units incidents
                    to the edges belonging to a maximum matching of H' }
   S:=V-(D∪F)
 end.
 FFC identification;
  begin
   H'':=SubGraphV(H,S);  {Given a graph H and a set S of vertices, function SubGraphV
                          returns the subgraph H'' of H of vertex set S}
   {A1,A2,...,Ak}:=ConnComp(H'');  {this function returns the vertex sets of the connected
                                    components of graph H'' }
   α:=MaximumCardinality(A0,..,Ak); {This function returns the maximum of cardinalities
                                     #A1,...#Ak}
   FFC:= ∪_{#Ai=α} Ai ;
   Tσ:=#F+#D/2+α      {Tσ is the syndrome-dependent bound for correctness }

  end.
 Augmentation;
  begin
    {Given any unit v, function Agg(v) returns set Ai if v∈Ai for some i=1,..,k, set {v}
    otherwise}
    while ∃u∈FFC, v∈V-(FFC∪F) such that u -0-> v do FFC:=FFC∪Agg(v);
    while ∃u∈FFC, v∈V-(FFC∪F) such that u -1-> v do F:=F∪Agg(v);
    while ∃u∈V-(FFC∪F), v∈F such that u -0-> v do F:=F∪Agg(v);
    S:=V-(FFC∪F);
    return(F,FFC,S,α,Tσ)
  end.
```

Table 2. The SVD diagnosis algorithm

3. Diagnosis Completeness

Given any syndrome σ, the diagnosis returned by the SVD algorithm is incomplete if set S is non-empty. This situation may occur if no units in set S is adjacent to units in the set *FFC* and, consequently, cannot be reliably tested during Augmentation. More precisely, let $S_i \subseteq S$ be a connected component of H induced by vertex subset S, and $B_i \subseteq V\text{-}S_i$ be the boundary of S_i defined as the set of units not in S_i which are adjacent to units in S_i. If $B_i \subseteq F$, then units in *FFC* are unable to test units in S_i, whose state cannot be identified unless some unit in S_i tests some neighbor in B_i with outcome 0 (statement d) of Lemma 1). The most critical situation is depicted in Figure 1, where S={u} and a boundary of n faulty units is sufficient to prevent the testing of u from the *FFC*. For arbitrary syndromes, diagnosis returned by the SVDA is guaranteed to be complete under the hypothesis of the following theorem, whose proof is immediate from the preceding reasoning.

Theorem 3. *The diagnosis returned by the SVD algorithm is always complete only if N_F is less than n.*

The result stated by Theorem 3 agrees with [6], in which it is proved that n is the diagnosability of an hypercube of dimension n. In fact, in the worst case depicted in Figure 1, unit u, with $S=\{u\}$ and $\#B=n$, could still be identified as non-faulty, since otherwise would be $F \supseteq B \cup \{u\}$ and thus $\#F>n$. However, this result is very pessimistic, as confirmed by the simulation results reported in the next section.

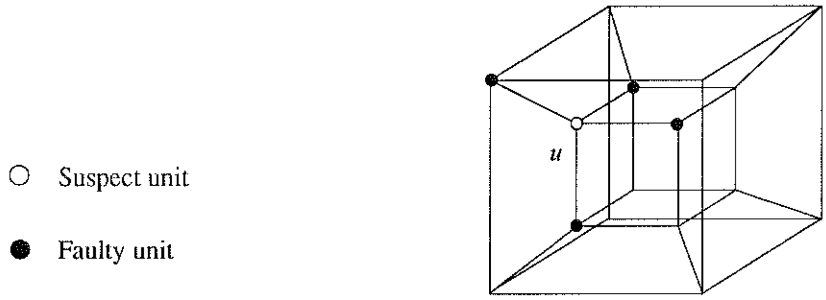

Figure 1. A boundary of faulty units encloses a suspect unit

4. Experimental evaluation

The expected value of T_σ, the percentage of complete diagnoses and the average number of units left unidentified by the algorithm SVD were evaluated by means of simulation. Results reported in this Section refer to hypercubes of different sizes and to fault sets of different cardinalities distributed uniformly over the vertex set. Table 3 reports the averages of the syndrome-dependent bound T_σ, denoted $E(T_\sigma)$. The averages were calculated over a sample of 1000 fault sets of cardinality ranging from 0,1N to 0,5N. For every entry, the variance v and the confidence interval[4] i are reported in parentheses. As seen from Table 3, the average of T_σ remains very close to N, irrespective of the size of the system, even when the fraction of faulty units reaches 0,5. In all cases, the confidence interval is very narrow and the variance is quite small.

N		N_F=0,1*N*	N_F=0,2*N*	N_F=0,3*N*	N_F=0,4*N*	N_F=0,5*N*
64	E(Tσ)	63.9 (*v*=0.13,*i*=0.05)	63.8 (*v*=0.32,*i*=0.07)	63.7 (*v*=0.25,*i*=0.06)	63.2 (*v*=1.60,*i*=0.16)	62.7 (*v*=2.04,*i*=0.18)
128	E(Tσ)	127.9 (*v*=0.07,*i*= 0.03)	127.8 (*v*=0.29,*i*=0.07)	127.7 (*v*=0.63,*i*=0.10)	127.4 (*v*=0.87,*i*=0.12)	126.9 (*v*=1.67,*i*=0.17)
256	E(Tσ)	255.9 (*v*=0.12,*i*=0.05)	255.9 (*v*=0.11,*i*=0.04)	255.6 (*v*=0.40,*i*=0.08)	255.5 (*v*=0.45,*i*=0.09)	254.9 (*v*=1.72,*i*=0.17)
1024	E(Tσ)	1023.9 (*v*=0.11,*i*=0.04)	1023.8 (*v*=0.14,*i*=0,05)	1023.6 (*v*=0.86,*i*=0.12)	1023.5 (*v*=0.48,*i*=0.09)	1023.0 (*v*=1.01,*i*=0.13)
16384	E(Tσ)	16383.9 (*v*=0.07,i=0.04)	16383.9 (*v*=0.14,*i*=0.05)	16383.7 (*v*=0.30,*i*=0.07)	16383.6 (*v*=0.35,*i*=0.08)	16383.1 (*v*=0.89,*i*=0.12)

Table 3. Evaluation of diagnosis correctness: $E(T_\sigma)$ is the average of T_σ over a sample of 1000 fault sets. The variance v and the confidence interval i are reported in parentheses.

Based on results reported in Table 3, it can be concluded that SVDA is able to provide a reliable diagnosis in all circumstances of practical interest.

---

(4) The confidence interval, calculated with precision p, is defined as the number i such that the probability $P(|E(T_\sigma)-\mu(T_\sigma)| \leq i)$ is greater than or equal to p, where $\mu(T_\sigma)$ is the average of T_σ calculated over the universe of all possible fault sets of given cardinality. In Table 3 p was set to 0.98.

Simulation results related to diagnosis completeness, which are reported in Table 4, are also excellent. Here, the percentages of complete diagnosis were calculated over samples of 1000 fault sets and the average number of units which remained not diagnosed was calculated over samples of at least 100 fault sets which led to incomplete diagnosis. For every entry $E(n_d)$, the variance and the confidence interval (calculated with a precision p=0.98) are reported in parentheses.

N		N_F=0,1*N*	N_F=0,2*N*	N_F=0,3*N*	N_F=0,4*N*	N_F=0,5*N*
64	c_p	100%	99.9%	97.2%	90.7%	65.51%
	$E(n_d)$	0 (*v*=0,*i*=0)	1.0 (*v*=0,*i*=0)	1.04 (*v*=0.04,*i*=0.05)	1.04 (*v*=0.04,*i*=0.03)	1.38 (*v*=0.56,*i*=0.10)
128	c_p	100%	99.5%	99.0%	90.7%	64.67%
	$E(n_d)$	0 (*v*=0,*i*=0)	1.0 (*v*=0,*i*=0)	1.0 (*v*=0,*i*=0)	1.08 (*v*=0.09,*i*=0.04)	1.29 (*v*=0.29,*i*=0.07)
256	c_p	100%	99.9%	99.1%	92.4%	63.64%
	$E(n_d)$	0 (*v*=0,*i*=0)	1.0 (*v*=0,*i*=0)	1.0 (*v*=0,*i*=0)	1.03 (*v*=0.03,*i*=0.02)	1.37 (*v*=0.50,*i*=0.10)
1024	c_p	100%	100%	99.6%	95.4%	60.63%
	$E(n_d)$	0 (*v*=0,*i*=0)	0 (*v*=0,*i*=0)	1.0 (*v*=0,*i*=0)	1.04 (*v*=0.04,*i*=0.04)	1.29 (*v*=0.40,*i*=0.09)
16384	c_p	100%	100%	100%	96.3%	60.94%
	$E(n_d)$	0 (*v*=0,*i*=0)	0 (*v*=0,*i*=0)	0 (*v*=0,*i*=0)	1.0 (*v*=0,*i*=0)	1.34 (*v*=0.35,*i*=0.08)

Table 4. Evaluation of diagnosis completeness. c_p is the percentage of complete diagnoses. $E(n_d)$ is the average number of unidentified unit over a sample of 100 fault sets which provided incomplete diagnosis. The variance v and the confidence interval i are reported in parentheses.

5. Comparison with previous algorithms

In this section, performance and cost of the SVD algorithm are compared with similar parameters of some previously known diagnosis algorithms for hypercube systems.

The performance measure considered are the *degree of diagnosability*, i.e. the maximum number of faults which would not impair the correctness of diagnosis, and the *degree of completeness*, defined as the expected percentage of complete diagnoses (in those cases where the diagnosis is not guaranteed to be complete).

To main cost parameter is related to the time needed to execute all the tests and to collect their outcomes (*syndrome generation and collection time*). Considering that tests involving different units can be performed in parallel, the cost is evaluated as the number of *testing rounds* needed to generate and collect the syndrome. In every testing round the largest possible number of tests are performed in parallel, with the constraint that any unit cannot act as a testing and a tested unit within the same testing round. In the case of sequential diagnosis another cost parameter is the number of *test and repair phases*, defined as the number of sessions, each of which aims at identifying and replacing at least one faulty unit, which are needed in order to remove all faulty units in the system. In general, every test and repair phase requires one or more testing round, but the replacement of units identified as faulty contributes to the cost. Another cost parameter could account for the time needed to execute the diagnosis algorithm on the external diagnoser. However, this time is usually negligible as compared to the syndrome generation and collection time; for this reason, this parameter is omitted in the following evaluations.

In [7] and [8], Kavianpour and Kim introduced four different strategies for the diagnosis of hypercubes. The first two strategies are based on the one-step approach, the latter on the sequential approach. The first strategy, called *precise* one-step diagnosis, provides correct and complete diagnosis, with degree of diagnosability *logN*. The strategy uses all the possible tests in the system; hence the number of testing rounds is $2 \cdot logN$. The second strategy, called *pessimistic* one-step diagnosis, tolerates the erroneous identification as faulty of at most one non-faulty unit. This is the price to be paid to increase the degree of diagnosability, which, however, refers to the ability of obtaining a *weakly correct* diagnosis, where one non-faulty unit could possibly be diagnosed as faulty. Under the pessimistic one-step diagnosis, the weak degree of diagnosability is raised to $2 \cdot logN - 2$. Again, the diagnosis uses all the possible tests in the system, which means that the number of testing rounds is $2 \cdot logN$. As an alternative, pessimistic diagnosis may be used to reduce the total number of required tests to $N \cdot (\lfloor logN/2 \rfloor + 1)$, while keeping the weak degree of diagnosability to *logN*. Although the authors do not give a detailed diagnosis algorithm, it appears that the number of testing rounds is at least $2(\lfloor logN/2 \rfloor + 1)$. The third strategy performs precise sequential diagnosis with a degree of diagnosability $t \le T \in \Theta(\sqrt{N} \log N)$. This approach achieves correct and complete diagnosis using R test and repair phases, with $R \le logN$. Although a detailed algorithm is not provided, it is clear that the total number of testing rounds needed to complete the diagnosis must be at least R. Finally, a trivial strategy which raises the degree of diagnosability to N-2 is given for the sequential diagnosis under the pessimistic assumption. The strategy needs R' test and repair phases to complete diagnosis, with $R \le R' \le log\ N$. The number of testing rounds needed to diagnose all the unit in the system is at least R'. The strategies proposed by Kavianpour and Kim are appropriate for application to massively parallel systems, but they appear not to be suitable for wafer-scale testing. In fact, the degree of diagnosability of both the precise and pessimistic one-step strategies is too small for the latter application, and sequential diagnosis strategies are not feasible in wafer-scale testing.

In [9], Feng, Bhuyan and Lombardi introduced two algorithms (called HADA and IHADA) for the adaptive diagnosis of hypercubes. In the adaptive diagnosis approach, special attention is deserved at reducing the total number of tests needed to perform diagnosis. HADA has lower performance but is easier to analyze than IHADA. It is shown that the number of tests and testing rounds needed by HADA to perform diagnosis is O(N) and 4+*logN* respectively with degree of diagnosability *logN*. IHADA has a degree of diagnosability $t \in O(\sqrt{N})$ and its cost, which was evaluated by means of simulation, is significantly below the cost of HADA. However, diagnosis returned by IHADA is not guaranteed to be complete and no evaluation is given about diagnosis completeness. Again, both algorithms are not suitable for wafer-scale testing, because degrees of diagnosability of *logN* or $O(\sqrt{N})$ are inadequate for this application.

In [10], Khanna and Fuchs introduced an algorithm (KF algorithm in the following) for the sequential diagnosis of hypercubes. The algorithm is based upon a partitioning of the set of units of the system into m clusters of size p ($N=m \cdot p$). A simple cycle is embedded in each cluster, and the syndromes associated to each cluster are considered. Only clusters associated to *0_syndromes* (i.e. syndromes where all test outcomes are 0s) are considered in the following step of the algorithm. The algorithm achieves a $\Theta(m)$ degree of diagnosability. The testing and repair phases needed to complete the diagnosis are O(*logN*), and the number of required tests is $2N+2ps$, where s is a parameter that depends on the number of 0_syndromes which actually occur. The number of testing rounds needed to perform diagnosis is $3+2s+\log m$. Since

$s \leq \frac{(m-1)m}{N}$ and m has to be $\Theta\left(\sqrt{N \log N}\right)$ in order to maximize the degree of diagnosability, it follows that the total number of testing rounds needed to complete the diagnosis is O(*logN*). However, it was observed that the actual degree of diagnosability of algorithm KF is significantly larger than $\sqrt{N}$ only if N is very large. Furthermore, the KF algorithm performs quite poorly if the number of actual faults is above the degree of diagnosability. In fact, observe that if every cluster contains at least one faulty unit, then either the algorithm is unable to provide a diagnosis (because no 0_syndrome is generated) or the diagnosis provided by the algorithm is incorrect (because all the clusters generating a 0_syndrome are entirely composed by faulty units). Assuming uniform distribution of faults, the probability that every cluster contains at least one faulty unit has been calculated using the Poincarè formula ([18]), from which we derive:

$$P(k) = \sum_{h=0}^{m} (-1)^h \binom{m}{h} \binom{mp}{k}^{-1} \binom{(m-h)p}{k},$$

where k is the number of faulty units and $P(k)$ is the probability that every cluster contains at least one faulty unit when $N_F = k$. It follows that the probability that the algorithm provides a correct diagnosis is at most than $1\text{-}P(k)$. An upper bound $b(k)$ to $1\text{-}P(k)$ is reported in Table 5 for N=64, m=p=8 and fault sets of different cardinalities: it is seen that $b(k)$ decreases dramatically as the percentage of faulty units approaches 50%. This implies that KF algorithm is not suitable for wafer-scale testing, where the expected fraction of faulty units may approach 50% and sequential-diagnosis is not feasible.

k	**percentage of faulty units (*k/N*)**	***b(k)***
7	11%	1
19	30%	0.3584
26	40%	0.0884
32	50%	0.0190

Table 5. Values of the bound $b(k)$ for N=64, m=p=8 and fault sets of different cardinalities.

The SVD algorithm introduced in this paper outperforms all the previous algorithms in terms of degree of diagnosability. In fact, the diagnosis provided by SVD algorithm is correct if $N_F < T_\sigma$. Although the bound T_σ is related to individual syndromes, it is almost certain that this inequality always holds if $N_F \leq N/2$. Moreover, the diagnosis correctness can be validated deterministically by individually probing a small number of units which were diagnosed as non-faulty. Simulation reveals that the average of this number is 1 or 2 as long as $N_F \leq N/2$. More noticeable, the diagnosis provided by the SVDA is *almost complete*. Although the occurrence of incomplete diagnoses increases as N_F increases, even when N_F=0,4N the percentage of complete diagnoses is above 90%. Moreover, when diagnosis is incomplete, the average number n_d of units left unidentified by the SVD algorithm is very small. Simulation shows that when N_F=0,5N, n_d is between 1 and 2, irrespective of the dimension of the hypercube. This means that the SVD algorithm is suitable for the application to wafer-scale testing, because it provides diagnosis which is almost complete and can be expected to be always correct, even when the percentage of faulty chips approaches 50%, with the additional feature that the correctness can be easily checked. The number of testing rounds needed by the SVD algorithm to perform diagnosis is $2{\cdot}logN$, since the algorithm executes all possible tests.

6. Concluding remarks

A new diagnosis algorithm for hypercube systems, called Self-Validating Diagnosis Algorithm (SVDA), has been introduced.

The SVDA, which runs on an external reliable processor, performs the diagnosis in a single step, using a syndrome consisting of the binary outcomes of all the mutual tests of units which are adjacent in the hypercube. The diagnosis can be expected to be complete or almost complete even if the percentage of faulty units approaches 50%.

The most notable feature of SVDA is that, together with the diagnosis, it returns a syndrome-dependent bound (T_σ), with the property that the diagnosis is correct if the cardinality N_F of the actual fault set is less than T_σ. This implies that it should be assumed (as it is usual in the so called deterministic diagnosis) that the expected number of faults can be reliably upper bounded. In the case of SVDA this bound can be very rough, since the average of T_σ, which was evaluated by means of simulation, is far above N_F even if this number approaches $0.5N$. Furthermore, the diagnosis provided by the SVDA can be validated deterministically by individually probing a small number of units (at most 2 on the average) that were declared non-faulty. This is a significant improvement over most probabilistic and deterministic algorithms where the cost of validating the diagnosis by individually probing individual units would be intolerable.

Because of the preceding properties, the SVDA appears to be a strong candidate for application to wafer-scale testing, where the percentage of faulty units may be large and correctness and completeness of diagnosis are vital requirements, in order to avoid the bonding and packaging of faulty chips, which would cause a serious economic loss.

References

[1] Preparata, F.P, Metze, G. and Chien, R.T., "On the Connection Assignment Problem of Diagnosable Systems". *IEEE Trans. Comput.*, vol. EC-16, pp. 848 - 854, Dec. 1967.

[2] Rangarajan, S., Fussel, D., Malek, M., "Built-in Testing of Integrated Circuit Wafers", *IEEE Trans. Comp.* vol. 39 n. 2, pp. 195 - 205, February 1990.

[3] LaForge, L.E., Huang, K., Agarwal, V.K., "Almost Sure Diagnosis of Almost Every Good Element", *IEEE Transaction on Computer* vol. 43 n. 3, pp. 295 - 305, March 1994.

[4] Huang, K., Agarwal, V.K., LaForge, L., Thulasiraman, K., "A Diagnosis Algorithm for Constant Degree Structures and Its Application to VLSI Circuit Testing", *IEEE Transaction on Parallel and Distributed Systems* vol. 44 n.4, pp.363-372, April 1995.

[5] Saad, Y. and Schultz, M.H., "Topological Properties of Hypercubes", *IEEE Trans. on Computers*, vol. C-37, pp. 867-872, July 1998.

[6] Armstrong, J. R., Gray, F. G., "Fault Diagnosis in a Boolean *n*-cube Array of Microprocessors", *IEEE Trans. Computers*, vol. C-30, no. 8, pp. 587-590, August 1981.

[7] Kavianpour, A., Kim, K. H., "Diagnosabilities of Hypercubes Under the Pessimistic One-Step Diagnosis Strategy", *IEEE Trans. Comp.*, vol. 40, no. 2, pp. 232-237, Feb. 1991.

[8] Kavianpour, A., Kim, K. H., "A Comparative Evaluation of Four Basic System-Level Diagnosis Strategies for Hypercubes", *IEEE Trans. Reliability*, vol. 41, no. 1, pp. 26-37, March 1992.

[9] Feng, C., Bhuyan, L. N., Lombardi, F., "Adaptive System-Level Diagnosis for Hypercube Multiprocessors", *IEEE Trans. Computers*, vol. 45, no. 10, pp. 1157-1170, October 1996.

[10] Khanna, S., Fuchs, W. K., "A Linear Time Algorithm for Sequential Diagnosis in

Hypercubes", *Journal of Parallel and Distributed Computing*, vol. 26, pp-48-53, 1995.

[11] Rattner, J., "Concurrent Processing: A New Direction in Scientific Computing", *AFIPS Conf. Proc.*, pp. 157-166, 1985.

[12] "nCUBE 2 Processor Manual", *nCUBE Corp.*, 1980.

[13] Maestrini, P., Santi, P., "Self-Diagnosis of Processor Arrays Using a Comparison Model", *Proc. 14'th SRDS-Symposium on Reliable and Distributed Systems*, Bad Neuenahr, Germany, pp.218-228, September 1995.

[14] Chessa, S., Maestrini, P., "Correct and Almost Complete Diagnosis of Processor Arrays", *Internal Report B4-27,* IEI, Pisa, June 1995.

[15] Chessa, S., Maestrini, P., Mangione, M., Polacci, S., Santi, P., "Self-Diagnosing Algorithms for Processor Arrays: Survey and Evaluation", *Proc. 2th ICAuto International Conference on Automation*, Indore India, December 1995.

[16] Scheinerman, E. R., "Almost Sure Fault Tolerance in Random Graphs", *SIAM Journal of Computing*, vol. 16, no. 6, pp. 1124-1134, December 1987.

[17] Blough, D. M., "Fault Detection and Diagnosis in Multiprocessor Systems", *Ph.D. Dissertation*, The Johns Hopkins Univ. Baltimore, MD, 1988.

[18] Rao, M. M., "Measure Theory and Integration", John Wiley and Son, 1987.

[19] Hopcroft, J., Karp, R. M., "An $n^{5/2}$ Algorithm for Maximum Matching in Bipartite Graphs", *SIAM Journal of Computing*, pp. 225-231, 1975.

[20] Cormen, T. H., Leiserson, C. E., Rivest, R. L., "Introduction to Algorithms", The MIT Press, 1990.