Intermittent and Transient Fault Diagnosis on Sparse Code Signatures

Kochte, Michael; Dalirsani, Atefe; Bernabei, Andrea; Omana, Martin; Metra, Cecilia; Wunderlich, Hans-Joachim


doi: http://dx.doi.org/10.1109/ATS.2015.34

Abstract: Failure diagnosis of field returns typically requires high quality test stimuli and assumes that tests can be repeated. For intermittent faults with fault activation conditions depending on the physical environment, the repetition of tests cannot ensure that the behavior in the field is also observed during diagnosis, causing field returns diagnosed as no-trouble-found. In safety critical applications, self-checking circuits, which provide concurrent error detection, are frequently used. To diagnose intermittent and transient faulty behavior in such circuits, we use the stored encoded circuit outputs in case of a failure (called signatures) for later analysis in diagnosis. For the first time, a diagnosis algorithm is presented that is capable of performing the classification of intermittent or transient faults using only the very limited amount of functional stimuli and signatures observed during operation and stored on chip. The experimental results demonstrate that even with these harsh limitations it is possible to distinguish intermittent from transient faulty behavior. This is essential to determine whether a circuit in which failures have been observed should be subject to later physical failure analysis, since intermittent faulty behavior has been diagnosed. In case of transient faulty behavior, it may still be operated reliably.

Preprint

General Copyright Notice

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

This is the author’s “personal copy” of the final, accepted version of the paper published by IEEE.¹

¹ IEEE COPYRIGHT NOTICE

©2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Intermittent and Transient Fault Diagnosis on Sparse Code Signatures

M. A. Kochte¹, A. Dalirsani¹, A. Bernabei², M. Omana², C. Metra², H.-J. Wunderlich¹
¹ Institut für Technische Informatik, Universität Stuttgart, 70569 Stuttgart, Germany
² University of Bologna, Bologna 40136, Italy

Abstract—Failure diagnosis of field returns typically requires high quality test stimuli and assumes that tests can be repeated. For intermittent faults with fault activation conditions depending on the physical environment, the repetition of tests cannot ensure that the behavior in the field is also observed during diagnosis, causing field returns diagnosed as no-trouble-found.

In safety critical applications, self-checking circuits, which provide concurrent error detection, are frequently used. To diagnose intermittent and transient faulty behavior in such circuits, we use the stored encoded circuit outputs in case of a failure (called signatures) for later analysis in diagnosis. For the first time, a diagnosis algorithm is presented that is capable of performing the classification of intermittent or transient faults using only the very limited amount of functional stimuli and signatures observed during operation and stored on chip.

The experimental results demonstrate that even with these harsh limitations it is possible to distinguish intermittent from transient faulty behavior. This is essential to determine whether a circuit in which failures have been observed should be subject to later physical failure analysis, since intermittent faulty behavior has been diagnosed. In case of transient faulty behavior, it may still be operated reliably.

Index Terms—Diagnosis, intermittent, transient, concurrent error detection, code signature, self-checking, online testing.

I. INTRODUCTION

In safety critical applications, such as medical, industrial, or automotive applications, failures in the field must be analyzed and diagnosed to identify potentially systematic root causes and avoid risks during future operation. Once a failure has been observed, the circuit can be analyzed in a workshop or during scheduled maintenance intervals.

In the automotive industry, suspected electronic control units (ECUs) are typically replaced for safety reasons and sent back to the OEM (original equipment manufacturer) even if the execution of functional tests does not reveal any failures in the workshop. However, returned units are often classified as no-trouble-found (NTF) or no defect found, i.e. the reported failure could not be reproduced or diagnosed at the OEM site [1], [2]. According to [3], the NTF rate of field returns in VLSI circuits can exceed 50%, which incurs high costs both for workshops and OEMs for replacements and failure analysis.

If an intermittent fault is the root cause of a failure in the field, a thorough test in the workshop or at the OEM may not excite the faulty behavior because of complex fault activation and different operation conditions. For transient faults, the faulty behavior can also not be reproduced. However, without an evidence that the observed failures have been transient, the unit must be replaced to prevent security risks.

By conventional failure analysis based on (re-)execution of tests followed by diagnosis, permanent faults can be effectively identified. However, it is in principle difficult to diagnose and distinguish intermittent from multiple transient faults.

To detect errors at runtime, self-checking circuits, which provide concurrent error detection, can be used. The employed concurrent error detection methods include structural redundancy (duplication with comparison), information redundancy by use of codes [5]–[7], watchdog circuitry [8], or synthesized assertions [9]. Here we consider self-checking circuits using separable codes, where data and check bits can be easily split apart. The generated check bits provide some information about failures at the time of their occurrence and can be captured for later analysis. In the following, these check bits are called signatures and stored in a signature log (SilO, Fig. 1). After a certain number of observed failures or in regular intervals, the data in the SilO is extracted and diagnosed offline to distinguish transient from intermittent root causes.

However, such a diagnosis is very difficult due to 1) extremely limited amount of failure data (low number of defects or marginal hardware, circuit aging, adverse operation conditions (temperature, voltage, electromagnetic radiation, vibrations, etc.), or combinations thereof. Resulting errors can occur in bursts [4].

Fig. 1. Signature-based diagnosis flow
stimuli and signatures, signatures are highly compacted responses),
2) low quality of stimuli (not deterministically generated test
patterns or patterns with high diagnostic resolution, but
random or functional input stimuli), and
3) non-repetitiveness of the test.

Conventional diagnosis algorithms focus on the location of
a fault and often assume a permanent nature of the fault.
The model-independent diagnosis in [10] classifies the observed
fault effects as conditional stuck-at faults that are active
depending on an (unknown) arbitrary activation condition. Yet
the algorithm is unable to classify transient and intermittent
effects correctly. Algorithms tailored to highly compacted test
responses [11] also exist, but they cannot distinguish transient
or intermittent faults that happen during operation.

Diagnosis algorithms specifically targeting intermittent
faults build a probabilistic model of the system and refine fault
classification by repeated application of tests and measurement
up BIST into sessions, repeats failed sessions and analyzes
intermediate test signatures. It uses Bayesian reasoning to
classify ambiguous fault effects. Such approaches are not
suitable for the problem at hand, since the repeated execution
of tests with operation conditions encountered in the field is
in general very difficult.

At system level, errors in memory can be statistically
analyzed at runtime to identify similarities in the observed
behavior and to conclude if a component suffers from transient
or permanent impairments [16]. However, if very few failures
are observed, the result of such reasoning has only little
confidence.

This work presents how failure data available in a self-
checking structure can be used to diagnose intermittent and
transient faults. Code signatures along with the corresponding
input stimuli are stored when failures are detected. A diagnosis
method is presented for offline evaluation of stored data to
distinguish intermittent and transient faults.

The next section gives an overview of the proposed method,
followed by a brief discussion of the used terminology. Section
IV presents the diagnosis algorithm, and Section V discusses
the experimental results.

II. OVERVIEW

A self-checking circuit is capable of detecting errors con-
current to the functional operation. Typically, a checker circuit
checks whether an encoded output is a codeword or not. If it
is not a codeword, an error indication is given.

In the considered architecture (cf. Fig. 2), we consider self-
checking circuits based on separable codes. The separable
check bits are called signature in the following. The circuit
is extended with a storage, called signature log (SILO). Upon
error detection, both the input stimulus and the code signatures
are stored. This requires that the primary and pseudo-primary
inputs are equipped with shadow registers to allow extraction
of the stimuli. The memory of the SILO can be (persistent) sys-
tem memory or chip-internal or external FLASH, for instance.

Available design-for-debug structures such as trace buffers
can be reused. A small controller is responsible to store the stimuli
and signature data. In principle, the input data can also be
compressed using a low-overhead compression algorithm if a
CPU is available.

Apart from failing input stimuli and corresponding sig-
natures, it is also important to store a set of input stimuli
that did not trigger an error. This information helps during
diagnosis to narrow down possible fault locations and fault
activation patterns. The selection of non-failing input stimuli
can be deterministic (e.g. at regular intervals, or a number of
stimuli after a detected error) or random. However, storing
consecutive input stimuli may impose additional overhead for
at-speed high bandwidth data storage. Here, we assume that
non-failing stimuli are selected at random intervals.

During a maintenance session, or if a certain number of
errors have been detected, the data in the SILO is extracted for
offline analysis. A diagnosis algorithm then analyzes the
stored stimuli and signatures to distinguish intermittent from
transient behavior. If the SILO contains only a single failing
entry, a distinction between a transient and an intermittent
faulty behavior is in principle not possible.

Storing the environmental and internal operation conditions
of the system, like temperature, VDD, or the system state at
the moment of failure, may further improve the diagnostic
resolution. But this is beyond the scope of this paper.

III. TERMINOLOGY AND PROBLEM DESCRIPTION

The goal is to distinguish intermittent and transient faults in
the circuit under diagnosis (CUD) using only the information
stored in the SILO. Intermittent faults shall not be falsely
classified as transient faults to avoid overlooking systematic
reliability threats in the system.

Let $S$ be the set of input stimuli stored in the SILO and $|S|
be the number of stored stimuli. Furthermore, $F \subseteq S$ is the
set of failed stimuli, i.e. stimuli that cause an error indication
in the self-checking circuit. For the failed stimuli, the code
signatures are also stored. For the remaining stimuli $S \setminus F$,
the code signature is error-free and can be obtained by logic
simulation.

The task is to determine whether the observed failures have
been caused by an intermittent fault or multiple transient faults
using only the \(|S|\) stimuli and \(|F|\) failed signatures stored in the SILO. Since different input stimuli lead to different signatures, a simple comparison of stored signatures is not meaningful to distinguish intermittent from transient faults.

In the following, we do not further investigate the classification of permanent faults since they can be easily detected and diagnosed by a structural test or repeated application of the stored failing stimuli.

IV. DIAGNOSIS ALGORITHM FOR SIGNATURE CLASSIFICATION

The fault model independent diagnosis algorithm “Pointer” of [10] is adopted here. Pointer performs an effect-cause analysis of the input stimuli and responses. It uses the conditional stuck-at line calculus to characterize the faulty behavior of the circuit under diagnosis (CUD) by considering both the topology and the defect behavior. A conditional stuck-at line fault is a stuck-at fault at a location \(f\) with an arbitrary activation condition. This activation condition may for instance be of Boolean or temporal nature and allows to model intermittent or transient fault activation.

For each fault location \(f\) in the fault machine (circuit model) and for each stored stimulus \(s \in S\), an evidence \(e(f, s) := (\Delta s, \Delta t_s, \Delta \tau_s, \Delta \gamma_s)\) is computed by fault simulation and response comparison between the fault machine under \(f\) and the CUD. \(\Delta s\) denotes the number of outputs that fail both in the CUD and the fault machine (cf. Fig. 3). \(\Delta t_s\) and \(\Delta \tau_s\) are the number of outputs mispredicted by the fault machine. \(\Delta \gamma_s\) is the number of outputs that fail in the fault machine, but are correct in the CUD. \(\Delta \gamma_s\) is the number of outputs that fail in the CUD, but are correct in the fault machine. \(\Delta \gamma_s\) is the minimum of \(\Delta t_s\) and \(\Delta s\). If \(\Delta \gamma_s = 0\), the conditional stuck-at fault at \(f\) partially explains the faulty behavior of the CUD: \(f\) is active and the failing outputs in the fault machine are a subset of the CUD (\(\Delta t_s = 0\)), or \(f\) is not active (\(\Delta s = 0\)). If \(\Delta \gamma_s > 0\), the conditional fault at \(f\) is a less suitable suspect.

\[\phi := \{s \in S | \sigma_s > 0\}\]

\(\phi\) can be used to reason about the temporal nature of the observed faulty behavior since higher values of \(\phi\) indicate frequent observation of failures at one fault location. For transient events, \(\phi\) of the respective affected fault locations equals 1 with high probability. This is because it is unlikely that different transient faults affect the same location multiple times (typically only one failing stimulus per transient). In case of intermittent behavior that has been detected multiple times, \(\phi\) will typically have values greater than 1. If there are multiple transient faults with overlapping output cones (some common failing signature bits), \(\phi\) of the top suspect can be greater than one. This makes the distinction to intermittent faults impossible. In the worst case, a circuit that suffered from transient faults is replaced without need. As shown by the experiments below, this happens in 12.7% of the investigated scenarios. In most cases, however, \(\phi\) of the top suspect is a good classifier to distinguish between transient and intermittent effects.

V. EVALUATION

The proposed diagnosis scheme is evaluated in experiments with fault injection of permanent, intermittent, and single and multiple transient faults (cf. Sec. V-A). In the evaluation, the achievable diagnostic resolution by use of the signature log SILO is assessed. The experimental flow is presented in Sec. V-B, the results are discussed in Sec. V-C.

A. Considered Fault Models

The following fault models are evaluated in the experiments:

Permanent faults: A Byzantine bridge fault between two signal lines \((k, l)\) is injected into the circuit [17]. Depending on the input pattern, this fault may alter the values of line \(l\), line \(k\), or of both lines in the circuit.

The Byzantine bridge fault model has been included in the evaluation since it causes fault effects originating at two
different lines in the circuit. Since it may be easily confused with transient effects at different lines, this complicates the diagnosis task.

**Intermittent faults:** A Byzantine bridge fault is injected into the circuit, but it is considered active only for a fraction \( \rho \) \((0 < \rho < 1)\) of detecting input stimuli. For the remaining stimuli that could detect the fault, the fault will not generate erroneous output values. These stimuli are called *detecting but non-failing stimuli*. The lower the values of \( \rho \) for intermittent faults, the more difficult is their distinction from transient faults.

As second type of intermittent fault, one intermittent stuck-at fault is injected into the circuit. The stuck-at fault is considered active only for the fraction \( \rho \) of the detecting input stimuli.

**Transient faults:** Different numbers of single event transients are injected into the circuit. Depending on the technology and transferred particle energy, a single or multiple signals in the circuit can be affected. The time period until the effect of a transient has been mitigated, may vary as well. Here we assume that a single line is affected for at most one clock period and inject a stuck-at fault for one clock cycle. The fault effect may still propagate to multiple circuit outputs.

### B. Evaluation Flow

The fault injection experiments are conducted for the discussed permanent, intermittent \((\rho = 0.5)\), and transient faults. The faults are selected randomly from the fault universe.

In the experiments, random pattern fault simulation is used to select the failing stimuli. \(|F|\) failing patterns and signatures are stored in the SILO (cf. Sec. III). For an intermittent fault, the parameter \( \rho \) defines the fraction of detecting stimuli for which the fault generates erroneous output values.

If the SILO does not contain *detecting but non-failing stimuli*, the fault appears as permanently active. To reflect the intermittent nature of the fault (and increase the difficulty for the diagnosis), \(|F| \cdot \frac{1-\rho}{\rho} \) *detecting but non-failing stimuli* are stored in the SILO.

Each injection, pattern selection, and diagnosis is repeated \( N = 20 \) times, and the results are averaged. In the experiments, \(|S|\) is set to 20, and \(|F|\) is set to 5. For transient faults, the SILO contains 1, 3, or 5 failing patterns depending on the number of injected transients. These numbers are chosen as an example. Experiments with a much larger number of non-failing stimuli \((|S| = 100)\) have been conducted as well. The diagnosis quality did not significantly improve. If \(|F| << 5\), the distinction between intermittent and transient faults is more difficult since the available data becomes insufficient. Experiments with \(|F| = 3\) are briefly discussed at the end of the section.

In the experiments, a double error detecting (DED) Hamming code, a triple error detecting (TED) Hsiao code, and a cyclic code are investigated for self-checking circuits. The original circuit (plain) is also included to assess the diagnostic quality when responses are not encoded/compacted. This corresponds to the achievable diagnostic quality when duplication with comparison or triple modular redundancy is used and failing responses are stored.

### C. Results

Table I shows the characteristics of the used ISCAS benchmark circuits and industrial circuits \((p^\%)^2\) kindly provided by NXP. The third and fourth columns give the number of primary and pseudo-primary inputs and outputs. The last three columns give the number of signature (check) bits depending on the used error detection code.\(^1\)

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Gate count</th>
<th>Inputs (Plain)</th>
<th>Outputs</th>
<th>Check bits for code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>s5378</td>
<td>3223</td>
<td>214</td>
</tr>
<tr>
<td></td>
<td></td>
<td>s9234</td>
<td>5944</td>
<td>247</td>
</tr>
<tr>
<td></td>
<td></td>
<td>s13207</td>
<td>8668</td>
<td>700</td>
</tr>
<tr>
<td></td>
<td></td>
<td>s15850</td>
<td>10211</td>
<td>611</td>
</tr>
<tr>
<td></td>
<td></td>
<td>s35932</td>
<td>16353</td>
<td>1763</td>
</tr>
<tr>
<td></td>
<td></td>
<td>s38417</td>
<td>23537</td>
<td>1664</td>
</tr>
<tr>
<td></td>
<td></td>
<td>s38584</td>
<td>21462</td>
<td>1464</td>
</tr>
<tr>
<td>p45k</td>
<td></td>
<td>38811</td>
<td>3739</td>
<td>2550</td>
</tr>
<tr>
<td>p77k</td>
<td></td>
<td>65015</td>
<td>3487</td>
<td>3400</td>
</tr>
<tr>
<td>p78k</td>
<td></td>
<td>68263</td>
<td>3148</td>
<td>3484</td>
</tr>
<tr>
<td>p81k</td>
<td></td>
<td>106450</td>
<td>4029</td>
<td>3952</td>
</tr>
<tr>
<td>p89k</td>
<td></td>
<td>80963</td>
<td>4632</td>
<td>4557</td>
</tr>
<tr>
<td>p100k</td>
<td></td>
<td>84356</td>
<td>5902</td>
<td>5829</td>
</tr>
<tr>
<td>p141k</td>
<td></td>
<td>152808</td>
<td>11290</td>
<td>10502</td>
</tr>
</tbody>
</table>

\(1\) The check bits for a cyclic code are set to at most twice the number of DED bits and may be smaller if a generator is not found for that number.

---

\(^2\) Typical conditions of NXP for a production circuit are \((p^\%)\) of 1, 2, or 3.

---

1. The check bits for a cyclic code are set to at most twice the number of DED bits and may be smaller if a generator is not found for that number.
The values of $\phi$ are given as black data points again. For the intermittent stuck-at fault and 5 stored failing stimuli, the diagnosis identifies a top suspect that is detected exactly 5 times. For the bridge faults, $\phi$ may also equal 1. In that case, the fault effects could not be classified as intermittent but are considered transient because of insufficient failure data. This happens in the worst case four times in the 20 iterations (cyclic code, $act = 1.0$). On average over the different injected Byzantine and intermittent stuck-at faults, such a wrong diagnosis is observed for only 5.42% of the iterations. If the circuit goes into operation again and additional failure data is collected, the diagnosis can be repeated with the union of all failure data extracted so far, which improves the diagnostic result.

In summary, the identification of the top suspect using the diagnosis algorithm on the stored stimuli and signatures and using the value of $\phi$ to reason about the temporal fault activation is robust for most of the injected faults.

3) Results for all circuits: Table II shows the values of $\phi$ of the top suspects for all investigated circuits, also averaged over the 20 iterations. The results of the injection of a single transient fault is omitted since $\phi = 1$ for all circuits and codes. The last four rows of the table give the averaged values of $\phi$ over all the circuits.

For the plain circuits (no compacted signatures), $\phi$ is very close to 1 for multiple transient events, with a maximum of $\phi = 2$. For the DED, TED, and cyclic codes, the maximum value of $\phi$ is 4 (5 injected transient faults). Over all circuits, in 12.7% of the experiments, $\phi > 1$. In such a case, the faulty behavior is falsely classified as intermittent, and a defect-free unit would be replaced to avoid safety risks. On average over all circuits, the values of $\phi$ are still below 1.5.

For the intermittent stuck-at fault, $\phi$ equals 5. Thus, $\phi$ is a robust and safe means to classify the intermittent nature of this type of fault. For the Byzantine bridge faults, the values of $\phi$ can be as low as 1, which happens for 4.55% of the iterations. For these faults, the failure data was insufficient to conclude the intermittency of the faulty behavior.

The runtime of the diagnosis algorithm for one circuit is at most 34.9s and has only low memory requirements. It can be easily conducted on a workstation in a workshop.

4) Results for $|F| = 3$: The diagnosis experiments are repeated assuming that only three failing signatures are stored in the SILO. This makes the distinction of intermittent and transient faults even more difficult. On average, for intermittent Byzantine bridges, $\phi$ ranges between 2.36 and 2.63 for the different codes, and for permanent Byzantine bridges between 2.45 and 2.56. Even with this limit on available failure data, it is possible to correctly identify intermittent faulty behavior for the majority of fault injections.

VI. Conclusion

The failure analysis of field returns incurs high costs. Often, circuits are classified as no-trouble-found. Here, we propose a diagnosis method for self-checking circuits. When failures...
are observed, the erroneous code signature and corresponding input stimulus are stored in a signature log on chip. The data in the signature log is later extracted for offline analysis. A diagnosis algorithm is presented that allows to classify the temporary nature of the observed failures. Experimental results using fault injection demonstrate that the vast majority of faults are correctly classified. This classification and the stored information can help to pinpoint the fault location and activation conditions for intermittent faults and to avoid needless replacements of units in case of transient faults.

ACKNOWLEDGMENT

This work was partially supported by the German Research Foundation (DFG) under grant WU 245/13-1 (RM-BIST).

REFERENCES


<table>
<thead>
<tr>
<th>Circuit</th>
<th>Code</th>
<th>Transient</th>
<th>Interm. act=0.5</th>
<th>Perm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>s5378</td>
<td>Plain</td>
<td>1.20 1.30 5.00 3.85 3.85</td>
<td>DED 1.15 1.70 5.00 3.70 3.55</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.20 1.50 5.00 4.35 4.45</td>
<td>TED 1.40 2.05 5.00 4.70 4.60</td>
<td></td>
</tr>
<tr>
<td>s9234</td>
<td>Cyclic</td>
<td>1.40 1.50 5.00 4.25 4.10</td>
<td>TED 1.25 1.45 5.00 4.40 4.20</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.30 1.40 5.00 4.50 4.40</td>
<td>TED 1.15 1.30 5.00 4.10 4.10</td>
<td></td>
</tr>
<tr>
<td>s13207</td>
<td>Plain</td>
<td>1.00 1.00 5.00 4.10 4.15</td>
<td>DED 1.25 1.50 5.00 3.95 4.05</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.15 1.25 5.00 4.50 4.35</td>
<td>TED 1.00 1.15 5.00 4.00 4.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cyclic</td>
<td>1.05 1.40 5.00 4.85 4.75</td>
<td>TED 1.00 1.15 5.00 3.15 3.50</td>
<td></td>
</tr>
<tr>
<td>s15850</td>
<td>Plain</td>
<td>1.00 1.00 5.00 4.05 3.75</td>
<td>DED 1.05 1.25 5.00 4.45 4.40</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.20 1.40 5.00 3.75 4.20</td>
<td>TED 1.20 1.40 5.00 4.45 4.25</td>
<td></td>
</tr>
<tr>
<td>s35932</td>
<td>Cyclic</td>
<td>1.10 1.25 5.00 3.50 3.80</td>
<td>TED 1.10 1.15 5.00 3.90 3.70</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.05 1.05 5.00 3.55 3.60</td>
<td>TED 1.05 1.05 5.00 3.80 3.80</td>
<td></td>
</tr>
<tr>
<td>s38417</td>
<td>Cyclic</td>
<td>1.30 1.75 5.00 3.75 3.85</td>
<td>TED 1.30 1.30 5.00 3.35 3.45</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.05 1.00 5.00 3.80 4.00</td>
<td>TED 1.05 1.00 5.00 3.75 3.75</td>
<td></td>
</tr>
<tr>
<td>s38584</td>
<td>Cyclic</td>
<td>1.10 1.60 5.00 3.45 3.70</td>
<td>TED 1.10 1.40 5.00 4.15 4.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.10 1.25 5.00 3.15 3.50</td>
<td>TED 1.10 1.25 5.00 3.85 3.85</td>
<td></td>
</tr>
</tbody>
</table>

TABLE II. VALUES OF $\phi$ OF THE TOP SUSPECT FOR $|S| = 20$