An Experiment of Burn-In Time Reduction Based On Parametric Test Analysis

Nik Sumikawa and Li-C. Wang
University of California, Santa Barbara

Magdy S. Abadir
Freescale Semiconductor, Inc

Abstract

Burn-in is a common test approach to screen out unreliable parts. The cost of burn-in can be significant due to long burn-in periods and expensive equipment. This work studies the potential of using parametric test data to reduce the time of burn-in. The experiment focuses on developing parametric test models based on test data collected after 10 hours of burn-in to predict parts likely-to-fail after 24 and 48 hours of burn-in. Our study shows that 24-hour and 48-hour burn-in failures behave abnormally in multivariate parametric test spaces after 10 hours of burn-in. Hence, it is possible to develop multivariate test models to identify these likely-to-fail parts early in a burn-in cycle. This study is carried out on 8 lots of test data from a burn-in experiment based on a 3-axis accelerometer design. The study shows that after 10 hours of burn-in, it is possible to identify a large portion of all parts that do not require longer burn-in time, potentially providing significant cost saving.

1 Introduction

Burn-in stresses parts in order to identify those likely-to-fail early in their life cycle. Burn-in can be an expensive step in a test flow. For example, the burn-in period can be tens of hours, which limits the throughput of testing. The test equipment is expensive and has a short lifetime due to the effects of thermal stress. Failing equipment can further reduce the throughput as fewer chips are tested in parallel.

The possibility of burn-in degrading quality has been a known issue. It has been shown that a part that is subjected to a static burn-in for 10 hours can experience up to 60% of the total NBTI degradation that it would see over its expected 10 year lifetime [13]. The irrecoverable effects caused by burn-in may compromise a product and actually increases the number of field failures.

Due to the cost and quality concerns, it is desirable to minimize the use of burn-in in a test flow. This is challenging for products demanding high quality, such as those sold to the automotive market, as some failing mechanisms may only be revealable through high temperature stress, for a long period of time.

This work is supported in part by National Science Foundation, Grant No. 0915259 and Semiconductor Research Corporation, project 2010-TJ-2093.

In this paper, we report findings based on a burn-in experiment performed on 8 lots of parts for a 3 axis accelerometer designed for the automotive market. Specifically, the experiment consisted of three stages of burn-in: 10 hours, an additional 14 hours (24 hours total), and an additional 24 hours (48 hours total). Parametric (and non-parametric) testing was performed after each burn-in stage to identify failing parts. The $\sim$60 parametric tests from the production test set were repeated at hot, cold and room temperatures.

The focus of the study is to develop a methodology that, at the end of 10 hours of burn-in, predicts parts that are likely-to-fail parametric testing after 24 hours and 48 hours of burn-in (suspect parts). Such a methodology can be used to identify parts that do not require additional burn-in after 10 hours, resulting in savings in terms of burn-in time for those parts.

To predict parts that may fail parametric testing after 24 hours and 48 hours of burn-in, we develop a learning methodology that builds multivariate test models based on parametric test data collected at the end of 10 hours of burn-in. At the core of this methodology, we employ Support Vector Machine (SVM) algorithms [1] for building both linear and non-linear models, depending on the *kernel* in use.

The focus of our study is not on the learning algorithms, but on developing a methodology that applies the learning algorithms to predict parts likely-to-fail parts after 24 and 48 hours of burn-in. In developing this methodology, we discovered several interesting aspects as to how multivariate test analysis can be effectively applied in the context of this work. These findings are summarized below:

- Parametric fails tend to require longer burn-in time. Hence, the focus of the study is on predicting parametric fails early in a burn-in process. Non-parametric fails tend to get exposed earlier in burn-in.
- Site-to-site variability can mislead learning. It is crucial to remove this variability before the learning.
- Before building a multivariate model, it is important to perform test selection. The effectiveness of a model depends on the tests used to build the model.
- Prior to test selection, it is important to partition tests into groups of tests of the same type. Automatic test selection can then be applied to each group separately.

978-1-4673-1595-1/12/$31.00 ©2012 IEEE

- When learning, each multivariate model is built with respect to a particular test temperature, since the failure signatures exposed in different temperatures (hot, cold, room) are usually not compatible to each other.
- To achieve zero test escapes, a few failing examples from 24 and 48 hours of burn-in are required. These examples are needed when the number of 10-hour fails in a failing category (bin) is too small to enable effective learning of the failing space (for that bin).

In this work, we will show that among the 42 parts that fail parametric testing after 24 hours and 48 hours of burn-in, 38 could be predicted using multivariate test models built using the data from 10 hours of burn-in. The remaining four fails fall into two categories: (1) Two (escaping) parts could be predicted using multivariate test models based on test measurements from 10-hour of burn-in, provided the earliest lot went through 48 hours of burn-in and we were able to leverage three parts failing after 24/48 hours of burn-in. These three failing parts were then used to refine the learning of a multivariate test model and the refined model was able to catch the two (escaping) parts. (2) The other two escaping parts could not be predicted because they were unique in the sense that they were the only failing part in their respective categories (sorted by test bins). Because there were no other 10-hour (nor 24/48-hour) failing parts in the same category, we could not learn a multivariate test model to predict them. However, we could apply a rule-based learning approach to explain them. The rule-based models (in contrast to multivariate SVM models) could be validated with domain knowledge and then applied to screen out similar future fails.

The rest of the paper is organized as the following. Section 2 briefly reviews prior efforts for reducing burn-in costs. Section 3 describes the data produced by the burn-in reduction experiment. The learning methodology and its important aspects are discussed in Section 4. Section 5 discusses the experimental results. Section 6 concludes.

2 Related work

For burn-in reduction, one early approach was to better utilize I_{DDQ} tests. I_{DDQ} tests were shown to be able to identify defective parts including those susceptible to fail during burn-in [17]. However, larger leakage currents render I_{DDQ} tests less effective [14]. There were many other works studying I_{DDQ} tests, e.g. [15, 16, 18]. In practice, burn-in is still used because it can be difficult to quantify some burn-in fails with I_{DDQ} measurements.

The alternative is to adopt advanced statistical methods by employing multivariate analysis [3]. In multivariate test analysis, a part is screened based on a model built from several tests collectively. The behavior, or signature, is exposed through the collective analysis of tests to determine the pass or fail status of a part. For example, the work in [4] is among those that pioneered this type of analysis for burn-in reduction. This work analyzed the parametric test measurements for a 90nm SoC and showed that Principal Component Analysis (PCA) could transform the tests data and reveal the abnormal behavior of defective parts. All parts were analyzed in a PCA space to identify outlying parts, which were be shown to be more susceptible to failing during burn-in. A more recent work in [6] analyzed wafer probe measurements and it was shown that known burn-in failures behaved as outliers in the wafer probe test space. A screening methodology was suggested that identified a population of good parts that could skip burn-in.

In the analog/RF space, the author in [5] analyzed a dataset consisting of function and parametric results. Various statistical methods were shown to effectively identify defective devices in the parametric test space.

In general, many other works utilized multivariate test analysis to predict devices susceptible to failing in the future. For example, the authors in [7, 8] analyzed the parametric test data for two products and applied enhanced binary decision forests to identify redundancies in the test set. The parametric test measurements analyzed for one of the devices belong to three final test insertions. The authors identified redundant tests belonging to one insertion and suggested that more expensive tests could be replaced with models built from those in less expensive insertions. In another example, the authors in [11] analyzed parametric wafer sort data from a high quality SoC and showed the potential for building models from the test data which were capable of predicting devices likely-to-fail at final package testing. Similarly, multivariate test analysis was used in [10, 12] to predict parts that would fail in the field, i.e. customer returns.

Following the promising results demonstrated in prior works, using multivariate test analysis, this work was motivated by two questions in the context of our specific burn-in experiment: (1) Can a multivariate test approach to predict failing parts after 24 hours and 48 hours of burn-in? (2) What are the key considerations when creating a learning methodology to enable prediction?

3 Burn-in Reduction Experiment

In an attempt to assess burn-in costs, a burn-in experiment was performed to determine the total burn-in period required to screen unreliable parts. In this specific experiment, the parts were subjected to varying intervals of burn-in where tests measurements were taken after an accumulated burn-in time of 10, 24 and 48 hours. This experimental flow is illustrated in Figure 1.

The parts that passed wafer tests were packaged and burn-in was performed for 10 hours. Then, each part was subjected to the production set of parametric tests which consists of $\sim$60 parametric tests including various current,

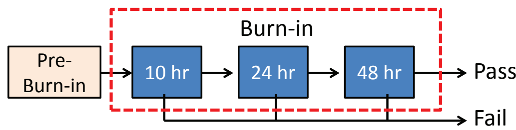

Figure 1. Burn-in Reduction Experiment Framework

voltage and frequency measurements. All tests were performed at cold, hot and then room temperatures. Parts that failed at one temperature were retested up to 6 times and the parts that failed after retest were removed prior to testing at the next emperature. For example, parts that fail cold temperature were removed before hot temperature testing. Parts that passed all 3 temperature tests were subjected to additional burn-in and tested again, after the accumulated burn-in time of 24 and 48 hours.

3.1 Burn-in Experiment Data

The burn-in reduction experiment was performed on 8 lots of packaged parts, where each lot contained more than six thousands parts. The parts that failed during this experiment were put into individual "bins," categorized by their failure mechanisms. Figure 2 shows a Pareto plot of all burn-in failing parts over bins (y-axis normalized). Note that bin 1 contains all failing parts that are non-parametric.

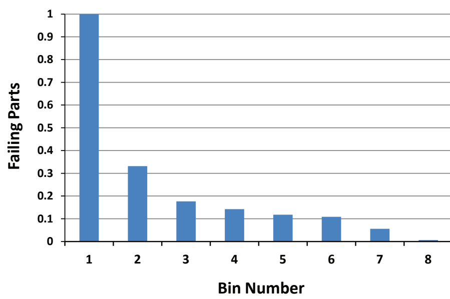

Figure 2. Pareto of all parts that fail burn-in

In Figure 2, most failing parts were captured after 10 hours. Only 34 parts failed after 24 hours and an additional 10 parts failed after 48 hours of burn-in.

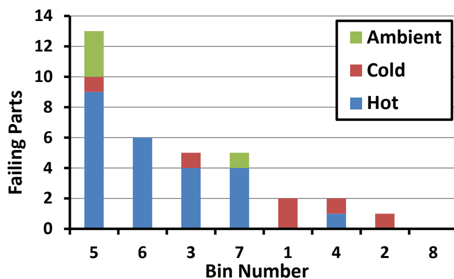

Figure 3. Pareto of parts that fail only after 24 hours of burn-in and before 48 hours of burn-in

Figure 3 shows the Pareto for the 34 failing parts after 24 hours but before 48 hours of burn-in. In this figure, the parts failing at room temperature testing are shown in green, cold temperature testing are shown in red and hot temperature testing are shown in blue. For example, bin 5 contains parts failing at all three temperatures. When comparing the Pareto plot in Figure 2 with the Pareto plot in Figure 3, we observe that the most frequently failing bins in Figure 2 (bin 1 and 2) are less frequent failing bins in Figure 3. After 24 hours, bin 1 (non-parametric fails) contains only two fails.

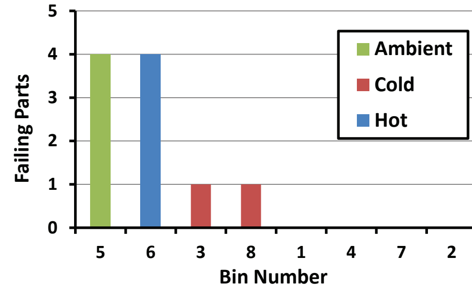

Figure 4. Pareto of parts that fail only after the 48 hours of burn-in

Figure 4 shows the Pareto plot for parts failing after 48 hours. We see that the top three failing bins in Figure 4 are the same as those in Figure 3.

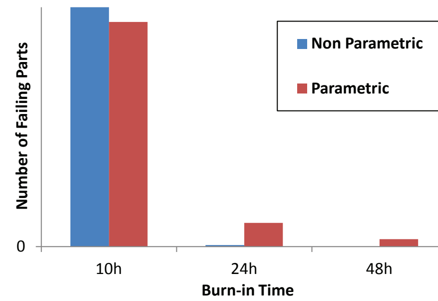

Figure 5. Non-parametric fails Vs. Parametric fails

Figure 5 compares parametric fails and non-parametric fails across the three stages of burn-in. Note that the number of non-parametric fails drops to two after 24 hours of burn-in and becomes zero after 48 hours of burn-in. The number of parametric fails, although drops significantly from 10 hours to 24 hours, does not go to zero after 48 hours of burn-in. Figure 5 shows that parametric fails require a longer burn-in time than non-parametric fails. Hence, when developing the multivariate test analysis methodology, our objective was to first target parametric fails.

4 The Learning Methodology

The methodology consists of two phases: a learning phase and an application phase, as illustrated in Figure 6. During the learning phase, parametric test data collected after 10 hours of burn-in is used to learn multivariate test models. In practice, there can be two learning scenarios: (1) We do not have examples of parts that fail after 24 or 48 hours to learn from. (2) We have examples of parts failing

after 24 or 48 hours to learn from. In the 2nd scenario, those examples can be utilized to guide the learning.

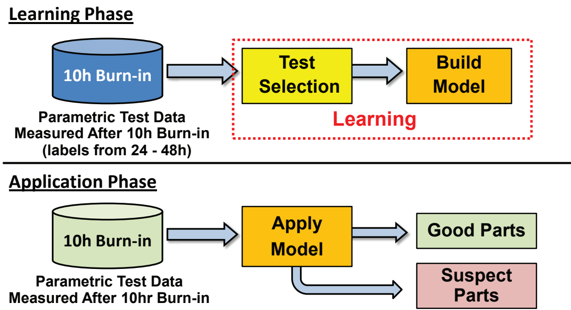

Figure 6. Learning and Application Phases

During the application phase, each model is applied to the parametric test data collected after 10 hours of burn-in. Only the parts failing any model are subjected to additional burn-in, while passing parts are not. Hence, the amount of overkill is less of a concern as it will add to the cost of burn-in. Test escapes are more of a concern because they will become field failures. Hence, the objective of learning is to build models resulting in zero test escapes.

4.1 Handling Site-to-Site Variations

Multi-site testing is often applied at wafer and package level to parallelize parametric testing. Measurements taken from different sites may have a different offset due to calibration issues, differences in probe resistivity, debris on the probes, etc. Figure 7 shows an example of this variation.

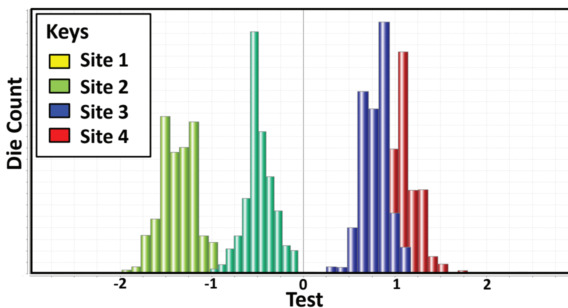

Figure 7. Site-to-site variation seen with a single test

In Figure 7, the distributions of four sites are shown for a single test. If each site is considered individually, each distribution resembles a Gaussian distribution. However, when viewing all results collectively, the distribution is clearly non-Gaussian. This can lead to misinterpreting the data as multi-modal if we do not account for the site information.

Site-to-site variations can mask outlying behavior. For example, the outlying parts in site 3 (blue) can reside in the middle of the distributions of site 2 (green) and of site 4 (red). In this figure, only positive outlying behavior from site 4 and negative outlying behavior from site 1 do not overlap with distributions from other sites. All others may be masked by other sites' distributions.

This masking can also occur in multivariate analysis, as illustrated in Figure 8. This shows the distributions of the 4 sites in a 2-dimensional test space.

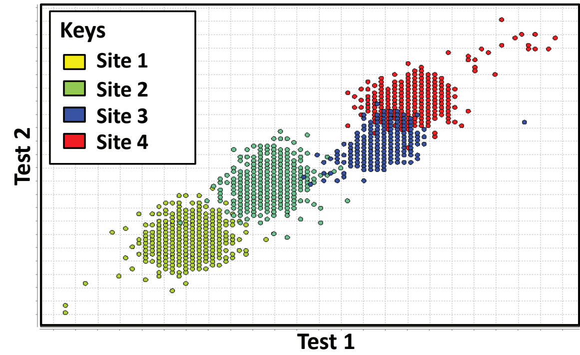

Figure 8. Site-to-site variation can generate misleading trends that impact the effectiveness of learning

When considering all samples in Figure 8 collectively, the data shows a clear linear correlation. However, much of this linear correlation is caused by site-to-site variation. If we focus on only the samples of a single site, we see that the two tests are actually uncorrelated as each individual distribution looks more like a circle. This shows how a misleading linear trend be created by site-to-site variation.

Figure 9 shows an example of masking outlying behavior in the 2-dimensional space. In this figure, failing parts from bin 7 are shown with all passing parts (red). We see that one failing part in site 2 (green) and one failing part in site 3 (yellow) are in the middle of the good distribution.

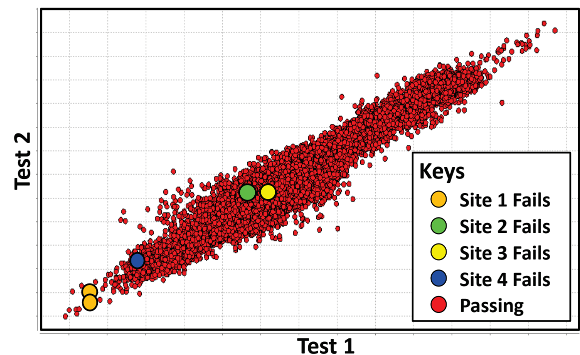

Figure 9. In the presence of site-to-site variation, some fails may reside in the middle of the distribution

Figure 10 shows only the parts based on site 3. In this figure, the outlying behavior of the failing part can be observed. Hence, this behavior is masked in Figure 9 above.

To remove site-to-site variations, each distribution of a site-test pair was repositioned such that the median is zero. After repositioning, the parts in Figure 9 are shown again in Figure 11. As we can see, all failing parts now show outlying behavior. This plot indicates a systematic failing signature for the failing parts in bin 7.

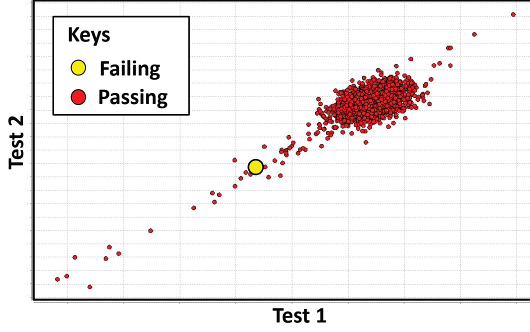

Figure 10. **Focusing on site 3 reveals the outlying behavior of the failing part**

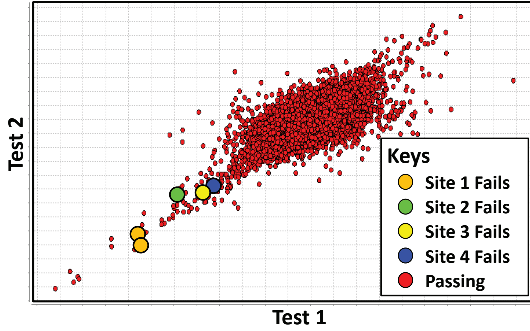

Figure 11. **After removing site-to-site variations, all failing parts show outlying behavior (Vs. Figure 9)**

4.2 Test selection for multivariate test analysis

With many parametric tests, it is important to differentiate tests that are relevant to describing the failure signature of a bin from those that do not. Test selection algorithms were applied in previous work to identify the tests which are most important in describing the failing signature for customer returns [10, 11, 12]. In this work, we apply the same SVM test ranking algorithm, which applies the C-Support Vector Classification (C-SVC) algorithm [1] to determine the importance of each test.

Each part is associated with a vector of test results: $\vec{v} = (v_1, \ldots, v_n)$, where n is the number of tests. In test selection, a linear model is learned with the C-SVC algorithm to separate the two classes of vectors in the n-dimensional space, where one class of results vectors contains passing parts and the other with failing parts. Let t_i be the variable to denote the result of test i. This linear model is represented as $\mathcal{C}(\vec{t}) = w_1t_1 + \cdots + w_nt_n + b = \langle \vec{w}, \vec{t} \rangle + b$, where $\langle \cdot, \cdot \rangle$ denotes the dot-product of two vectors. This equation defines a linear hyperplane in the n-dimensional space.

The C-SVC with a linear kernel solves the following optimization problem:

$$\min \tfrac{1}{2}||\vec{w}||^2 + C^+ \textstyle\sum_{\forall i} \xi_i^+ + C^- \sum_{\forall j} \xi_j^-$$
subject to
$$(\langle \vec{w}, \vec{g}_i \rangle + b) \geq 1 - \xi_i^+ \;\; \forall \;\text{good parts} \qquad (1)$$

$$(\langle \vec{w}, \vec{f}_j \rangle + b) \leq -(1 - \xi_j^-) \;\; \forall \;\text{failing parts} \qquad (2)$$

$$\text{and } \xi_i^+ \geq 0, \xi_j^- \geq 0$$

where $\vec{g}_i$ is the test result vector for a good part i and $\vec{f}_j$ is the vector for a failing part j. The slack variables ξ_i^+ and ξ_j^- are used to measure how much separation the hyperplane can achieve to capture the passing and failing parts respectively. C^+ and C^- are constants that control how hard the model should try to correctly classify the parts in the passing and failing classes respectively. When learning the test importance using C-SVC, we set $C^- \gg C^+$ to ensure that the resulting hyperplane correctly classifies all failing parts.

In an ideal situation, we want to have $\xi_i^+ = 0$, $\xi_j^- = 0$ for all i and j where the model computes a value ≥ 1 for all good parts (1) and a value ≤ -1 for all failing parts (2). This explains why in the objective function, ξ_i^+ and ξ_j^- are minimized. We also want to maximize the *margin* of the hyperplane, $1/||\vec{w}||^2$, which is the distance from the hyperplane to the closest data point. This explains why the objective function minimizes $||\vec{w}||^2$.

Solving the optimization problem leads to values for $w_1, \ldots, w_n, b$. These weights $w_1, \ldots, w_n$ are taken as the importance measures for each individual test. Each weight, w_i, describes the amount the hyperplane is oriented in the direction of the variable t_i, where a large $|w_i|$ means that t_i is in an important direction described by the hyperplane. Hence, the weight $w_1, \ldots, w_n$ can be used to rank tests.

4.3 Model building - binary classifier

In this study, a model is learned from an imbalanced dataset consisting of a large set of passing parts and a much smaller set of failing parts (as small as 1 or 2 parts). When learning, it is important to address this dataset imbalance which was why the SVM algorithms [1] and more specifically, the C-SVC algorithm [2] was chosen over other supervised learning algorithms.

As described in the previous section, we can deal with the imbalanced dataset by setting $C^- \gg C^+$ to ensure that none of the failing parts are misclassified (no test escapes) while allowing for some overkill. This is important in the context of this work, as the primary objective is to prevent failing parts from escaping the remainder of the burn-in process. Overkilling parts when applying a model is more acceptable as it only adds to the cost of burn-in.

The C-SVC algorithm using a linear kernel $k(x_a, x_b) = \langle x_a, x_b \rangle$ was used in a prior work [11], to build models from imbalanced datasets consisting of few customer returns (1-2 fails) and many good parts (thousands). The linear kernel was suitable because the number of customer returns was small and multiple returns exhibiting similar failing behavior were learned together.

In this work, the C-SVC algorithm is applied in the same way but the linear kernel is replaced with the non-linear Gaussian kernel $k(x_a, x_b) = exp(-\gamma||x_a - x_b||^2)$. These non-linear models are recommended for scenarios where there are many (tens) failing parts whose failing signatures point in slightly different directions in the multivariate test space. Both the linear and non-linear models are effective at describing a the failing space in high dimensions, but the non-linear model can achieve this with fewer misclassified samples i.e. overkill.

4.4 Pre-filtering tests

When determining test importance using the C-SVC test selection algorithm, all tests are used. However, this can lead to an over-fit model. To demonstrate this, we use bin 5 tested at hot temperature as an example. We divide 8 lots of data into a training set and validation set, each consisting of 4 lots. We rank all tests by applying the C-SVC using the failing and passing parts in the training set. Using the test ranking, we select the top i tests (with the largest $|w_i|$) to build a binary classifier to separate the failing parts from the passing parts.

When training, we are only aware of parts in the training set. Hence, we select the top i tests such that the training error, measured in terms of the number of overkill, is minimized (recall that each model ensures that all failing parts are correctly classified, i.e. no test escapes). The following explains how this minimization objective could lead to over-fitting the model to the training set, which results in a model that produces test escapes when applied to the validation set.

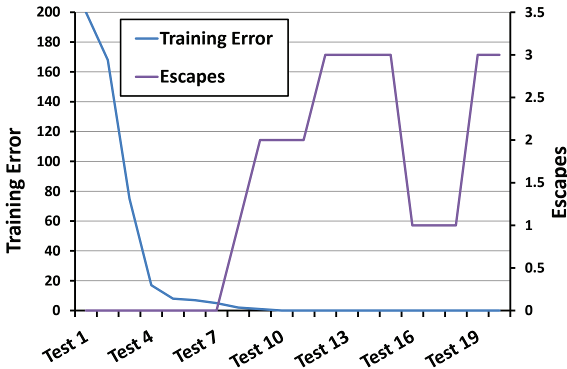

Figure 12. Results of selecting the top i tests when model learning is applied with all tests

Given the test rank, Figure 12 shows the training error as we selected the top i tests to build the model, for $i = 1, \ldots, 20$. Then, these models were applied to the validation dataset to monitor the number of test escapes for each model. In Figure 12 the number of selected tests used to build each model is shown on the x-axis, the training error is shown on the left y-axis, and the number of test escapes in the validation set is shown on the right y-axis.

In Figure 12, we see that the training error decreases rapidly as more tests are used. When $i \leq 7$ tests, the model results in no test escapes in the validation set. For i >7 tests, including additional tests results in test escapes while the training error continues to decrease. This indicates model over-fitting (over-fitting to the training set) which is less generalizable to the validation set and result in overkill.

Figure 12 presents a challenge for test selection. When training, we are not aware of the data from the validation set. Hence, selecting more tests to reduce the training error to zero would seem to be a logical strategy. The problem of model over-fitting could be avoided by filtering out tests unrelated to the failure mechanism prior to test selection. For example, when dealing with clock related failures, we can filter out tests that target the ADC block.

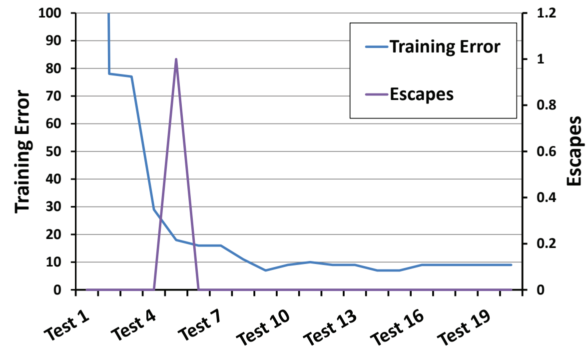

Figure 13. Results of model learning when test selection is performed after filtering out tests unrelated to the failure mechanism

Following the experiment in Figure 12, Figure 13 shows the result when filtering out tests unrelated to the failure mechanism prior to test selection. We see that the training error (# of overkill) decreases rapidly for the first 9 tests but no longer goes to zero. As more tests are used, the number of test escapes remains at zero, except when the model is learned with the first 5 tests where only one part escapes. Figure 13 simplifies test selection because we could select tests up to the point where the training error saturates, e.g. with 10-12 tests. At this point, the model also gives no test escapes in the validation set. This shows that pre-filtering tests can improve the robustness of the test selection.

4.5 Tests in Hot Vs. Cold Temperatures

As mentioned before, each part was tested with each parametric test at hot, cold and room temperatures. In this section, we show that it is important to learn models for different temperatures separately because different temperatures exposed failing parts as outliers in different directions. For the failures shown in Figure 3 and 4, there were several bins where parts failed at different temperatures. In those cases, multiple models were built for each bin.

In Figure 14, the parts from one lot are shown in a 2-dimensional test space consisting of the same test per-

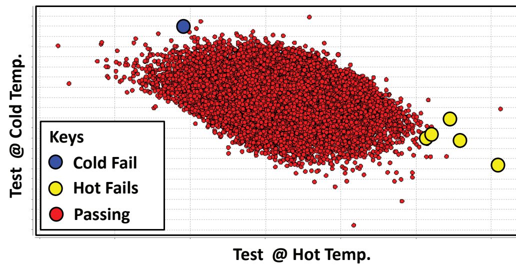

Figure 14. Tests at different temperatures expose failing parts as outliers in different directions

formed at hot and cold temperatures. The passing parts are shown in red. The failing parts in bin 5 are shown where the parts failing in hot temperature are shown in yellow and the parts failing in cold temperature are shown in blue. We see that the two sets of failing parts reside on opposite sides of this parametric test space. The hot failures reside on the bottom right and the cold failure resides on the top left. We can also see that four of the five hot failures do not show outlying behavior at cold temperature. The single cold failure does not show outlying behavior at hot temperature.

As mentioned before, parts in a bin were grouped together because they failed with similar reasons (manually decided). Nevertheless, we see that the failing signatures of these failing parts could be different depending on the test temperature. For this reason, a multivariate test model was learned for each bin-temperature pair separately.

5 Experiment

Recall that there were 32 parametric failing parts after 24 hours, and 10 parametric failing parts after 48 hours. Our objective is to learn models using parametric data after 10 hours to predict these failing parts with a reasonable number of overkill. As discussed in Section 3.1, the two 24-hour non-parametric fails were not considered in the study.

5.1 Learning with Only 10-hour Failing Parts

While the models utilize only parametric measurements after 10-hour of burn-in, in training there are two scenarios to consider. In the first scenario, we assume no information is available for parts failing 24 and 48 hours. In practice, a burn-in process may stop at 10 hours and the objective is to learn models to decide the set of parts that need to continue with the burn-in. In the second scenario, we assume some known failing parts are available from 24 and 48 hours of burn-in. For example, 1-2 lots could go through the longer burn-in time to extract examples of failing parts. These known failing parts can be used to guide the learning. In this section, we assume the first scenario. The second scenario is discussed in the next 2 sections.

When learning a model, we divided 8 lots of data into 4 lots for training and 4 lots for validation. Parts that failed after 10 hours of burn-in were used. The parts failing after 24/48 hours were put into the validation set. They were considered as passing parts in the learning.

There were 8 test bins and three test temperatures. In total, we built 24 models. As discussed in Section 3.1, bin 1 contained non-parametric related failures that were not of our concern. For the remaining 21 bin-temperature pairs, 8 of them did not contain any failing part. As a result, we learned 13 models, one for each bin-temperature pair that contained at least one failing part. Each model was learned with the 4 lots of training data and validated with the 4 lots of validation data to ensure that none of the 10-hour failing parts were misclassified. Together, each model divided all parts in the 8 lots into two classes, the passing and the (likely) failing classes.

In this experiment, we were interested in two numbers: the total number of overkill incurred by each model and the failing parts from 24/48 hours that were captured by the model as failing parts. Table 1 shows the results for the 13 models. The "Tests" column shows the number of tests selected which were used to build the model. The "10h Fails" column shows the number of 10-hour failing parts used in the training set. We use "> 10" to avoid showing the actual numbers so no detailed information on yield is revealed. The "# of overkill" is the number of total passing parts (after 10 hours) classified by the model as "likely-to-fail." These are the parts that need additional burn-in.

Bin	Temp	Tests	10h Fails	# of overkill	24-48h fails	
					Capture	Escapes
2	Cold	10	> 10	3	1	0
3	Hot	10	> 10	11	4	0
3	Cold	2	1	52	2	0
4	Hot	5	> 10	323	1	0
4	Cold	2	> 10	4,040	1	0
5	Hot	10	4	1	4	0
5	Cold	10	2	63	1	0
5	Room	3	> 10	3,370	7	0
6	Hot	4	2	218	13	2
7	Hot	9	5	256	4	0
3 models below do not catch any 24/48h fails						
2	Hot	10	> 10	58	0	0
4	Room	2	8	380	0	0
7	Cold	10	2	56	0	0
Two 24/48h fails are unique; no 10hr fail in the same category						
7	Room	-	0	-	0	1 (24hr)
8	Cold	-	0	-	0	1 (48hr)

Table 1. Learning from parts that fail after 10 hours of burn-in to predict parts that fail after 24/48 hours with a total # of overkill $= 7403 < 15\%$ of total population

Among the 42 failing parts after 24/48 hours of burn-in, Table 1 shows that 38 of them could be predicted by 10 multivariate test models. Three models did not predict

any 24/48-hour fail. However, they did not incur a large number of overkill either. The total number of overkill is 7403, representing less than 15% of the total parts.

Bin 7 at room temperature contained only one fail after 24 hours. It did not have any fail at 10 hours. Hence, there was no model built. Similarly, bin 8 at cold temperature had only one fail, after 48 hours. Hence, no model was built for that bin-temperature pair either.

Figure 15 shows an example 2-dimensional linear test model, with all passing parts (red) and failing parts in bin 5 at room temperature. The model was learned from the parts that failed after 10 hours of burn-in (yellow). We see that no failing parts resides in the passing region and the suspect space has a mixture of passing and failing parts, which includes all parts that failed after 24 (green) and 48 (blue) hours.

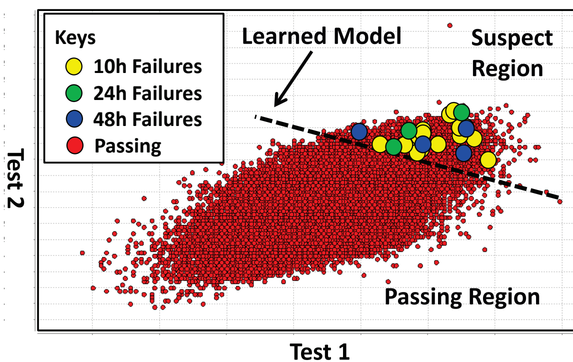

Figure 15. **An example model to illustrate learning from the parts that failed after 10 hours of burn-in to predict parts failing after 24/48 hours of burn-in**

The example model in Figure 15 is shown with a 2-dimensional test space for illustration purpose. In actual application, it is more effective to learn a model in a higher dimensional test space because it separates the failing from passing parts more effectively. For this particular example, the model used to obtain the result shown in Table 1 was actually learned with three tests. Moreover, the non-linear Gaussian kernel was used to build a non-linear model to reduce the number of overkill.

5.2 Reducing overkill by Learning From Parts that Fail After 24 Hours of Burn-in

If we use bin 5 at room temperature in Table 1 as an example, we see that the number of overkill is large, i.e. 3370. There were 7 failing parts after 24 and 48 hours. We selected one of these parts, from the earliest lot, which failed after 24 hours. We assumed the lot went through 24 hours of burn-in so that this part was a known failing part. This part was added to the training dataset and used for learning. Table 2 shows that with this additional failing part, the number of overkill can be reduced to 2269.

Tests	10h Fails	24h Fails	# of overkill	24-48h Fails	
				Capture	Escapes
3	> 10	0	3,370	7	0
3	> 10	1	2,269	6	0

Table 2. **Reducing overkill by learning from a single known failing part after 24 hours**

5.3 Reducing test escapes by learning from Parts that Fail after 24-48 hours of Burn-in

One major issue shown in Table 1 is the two test escapes for the model built from bin 6 at hot temperature. Notice that there are only two 10-hour fails. Hence, the number of failures used in the learning was very small. For illustration purpose, Figure 16 shows a 2-dimensional test space where all passing parts (red) and failing parts after 10 (yellow), 24(green) and 48 (blue) hours are shown. We see that learning from two 10-hour failures is not enough to push the decision boundary left enough to include the two test escapes (the two parts on the left of the dot line, i.e. the "learned model"). As a result, there are two failing parts, one after 24 hour and the other after 48 hour that reside in the passing region. Hence, the test escapes were due to the small number of 10-hour failures available for learning.

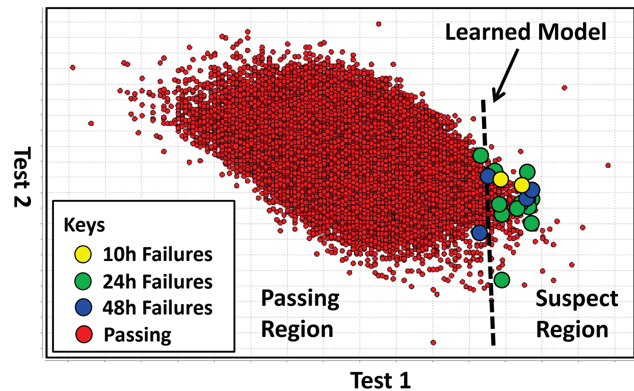

Figure 16. **Learning from too few parts that fail after 10 hours of burn-in may not be effective**

Among the 13 24/48-hour fails in bin 6 at hot temperature, we selected three parts from the earliest lot that were not the two test escapes. We assumed that these three parts were known fails available for learning. Together with the two 10-hour failing parts, we had 5 failing parts for learning. A new set of tests were selected and a non-linear model was learned.

Table 3 shows the effect of adding the three known 24/48-hour fails. The first row is the results from Table 1. The second row shows zero test escapes with increased number of overkill. The two escapes are now among the 12 fails captured by the model.

Tests	10h Fails	24-48h Fails	# of overkill	24-48h Fails	
				Capture	Escapes
4	2	0	218	13	2
3	5	3	641	12	0

Table 3. Reducing test escapes by learning from three known 24/48-hour fails

5.4 Summary of results

Table 4 summarizes the results discussed above. In this table, only 40 24/48-hour fails are considered. The two unique fails will be discussed in the next section.

Fails used in learning	Total overkill	24/48 Fails Considered	Capture	Escapes
10 hours only	7403	40	38	2
10/24/48 hours	7327	36	36	0

Table 4. Summary of results

In the second row of the table, we assumed the earliest lot went through 48 hours of burn-in and four failing parts after 24/48-hours became known failing parts. These four failing parts were used in the learning. The result, from all models collectively, shows slight a reduction in the number of overkill while resulting in no test escapes. Table 4 suggests that prior to applying the proposed methodology, an initial run of 48 hours of burn-in may be required on a few lots to collect examples of parts failing after 24/48 hours. These examples are useful for guiding the model building to avoid test escapes. As mentioned before, the total overkill number shown in the table represents less than 15% of the total population. In other words, less than 15% of the parts require additional burn-in after 10 hours.

6 Explaining Unique Failures

Bin Number	Temp	24/48 hour Fails	10h Fails
7	Room	1 (24h)	0
8	Cold	1 (48h)	0

Table 5. Failing parts that could not be predicted using the previously-described learning approach

Table 5 summarizes the two 24/48-hour failing parts that are unique in the sense that the same bin-temperature pair has no 10-hour nor other 24/48-hour fails. Because these two are unique, we cannot build a model to predict them. We could learn from the failing part itself to build a model, but we would not have another failing part to validate the model. The alternative is to apply a rule-based approach [19] to explain the unique fail.

6.1 Learning a Rule to Describe the Unique Fail in Bin 8 which Fails After 48 Hour of Burn-in

With one failing part and many passing parts, rule learning can be applied to extract a ”test” rule to explain the uniqueness of the failing part. The test vector of the failing part satisfies such a rule while the test vectors of none of the other parts satisfies the rule. The rule found for the failing part in bin 8 in Table 5 is shown in Table 6.

Clause	Test Type	Measured Range
Voltage Test 1	Cold	-∞ - -3.36
Voltage Test 2	Cold	-∞ - -1.80

Table 6. Rule learned for the bin 8 failing part

The rule in Table 6 states that the abnormal behavior for the failing part in bin 8 is characterized by the outlying behavior on the negative side of a regulator voltage test and of a voltage stress test. After discussing this result with the test engineer, we considered this to be a feasible rule because bin 8 characterizes parts that fail to produce a specific voltage value out of the voltage regulator.

This rule can be visualized in a 2-dimensional test space shown in Figure 17. The rule describes the region in the bottom left corner of this test space. As we can see, the unique failing part resides in this tests space where no other parts does.

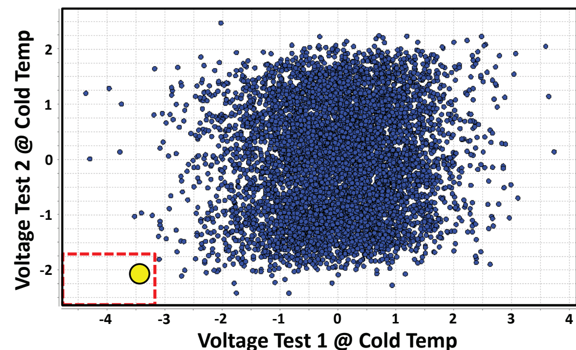

Figure 17. Applying the learned rule to the one lot containing the failing part

In order to validate this rule, the rule was applied to all 8 lots of data. In addition to the unique failing part, only six more parts satisfied the rule. In application, these six parts would be added to the overkill set.

6.2 Learning a Rule to Describe the Unique Fail in Bin 7 which Fails After 24 Hour of Burn-in

Similarly, for the unique fail in bin 7 at room temperature, we extracted a rule to describe its uniqueness. The rule is shown in Table 7. This rule is described by two voltage tests performed at different temperatures. The failure type is related to the clock. The rule suggests that the frequency

issue may be related to the voltage of the chip. When applying this rule to all 8 lots of data, only one additional part satisfied the rule.

Clause	Test Type	Measured Range
Voltage Test 1	Ambient	0.0067 - 2.219
Voltage Test 1	Hot	-4.007 - -2.262
Voltage Test 2	Cold	-2.984 - -2.012

Table 7. Rule learned for the bin 7 failing part

7 Conclusion

In this work, we study the potential of using parametric test data after 10 hours of burn-in to predict the parts that fail with additional hours of burn-in. The experiment was based on 8 lots of parametric test data for a 3-axis accelerometer design. Our findings can be summarized as the following: (1) It is more effective to learn a multivariate binary classification model for each bin-temperature pair. (2) Applying these models can identify a large population of passing parts that do not require additional burn-in after 10 hours. (3) To achieve zero test escapes, it is required to perform an initial experiment to obtain a few known failing examples at 24/48 hours. (4) A diagnosis approach, i.e. rule learning, is required to handle unique fails that only occur after 24/48 hours of burn-in. While we cannot build a model to predict a unique fail, a rule (based on 10-hour parametric measurements) can be learned to explain them. These rule models can then be applied to identify similar fails (using only 10-hour burn-in data) in the future.

The two 24-hour fails are not considered in this study because they are non-parametric fails (in bin 1). This type of failure seems to be exposed earlier in the burn-in process when compared to parametric fails. This suggests that in practice, we could run burn-in for a predetermined amount of time to expose all non-parametric fails. The experimental data suggests that we have to perform burn-in on all parts for 24 hours. Then, more than 85% of the parts could be identified as passing which do not require the additional 24 hours of burn-in. If we consider the original total burn-in time to be 48 hours multiplied by the total number of parts, the result suggests that there could be a saving of more than 42.5% of total burn-in time. In practice, a larger dataset is needed as the number of non-parametric fails is not sufficient to get a confident measure of the burn-in time required to expose the non-parametric failures.

The proposed learning methodology predicts parametric fails and it is not applicable for learning from non-parametric fail as the tests that target their failure mechanism are binary i.e. either pass (1) or fail (0). For example, parts passing all tests after 10 hours would have the same test vector of all 1's. Hence, there's nothing to learn from. To predict non-parametric failing parts, we need to create a parametric test spaces where their abnormal signatures can be exposed earlier in a burn-in cycle. This will be subjected to further research.

8 Acknowledgments

The authors would like to thank Matthew Parker and Cinda Flynn of Freescale Semiconductors for their help in collecting and interpreting the data.

References

[1] B. Scholkopf and A. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization,and Beyond. *The MIT Press*, 2001

[2] M. Tsai, C. Ho and C. Li, Active Learning Strategies Using SVMs. In *IJCNN*, 2010

[3] W. Robert Daasch, C. Glenn Shirley and Amit Nahar. Statistics in Semiconductor Test: Going beyond Yield. IEEE Design & Test of Computers, vol 26, issue 5, 2009, pp. 64-73.

[4] Amit Nahar, Robert Daasch and S. Subramaniam. Burn-in Reduction using Principle Component Analysis. *IEEE International Test Conference*, 2005.

[5] Haralampos-G. D. Stratigopoulos, Petros Drineas, Mustapha Slamani and Yiorgos Makris. Non-RF To RF Test Correlation Using Learning Machines: A Case Study. *IEEE VLSI Test Symposium*, 2007.

[6] Amit Nahar, Kenneth M. Butler, John M. Carulli Jr. and Charles Weinberger. Quality Improvement and Cost Reduction Using Statistical Outlier Methods. *IEEE International Conference on Computer Design*, 2009, pp. 64 - 69.

[7] Biswas, S. and Blanton, R.D. Reducing Test Execution Cost of Integrated, Heterogeneous Systems Using Continuous Test Data. *IEEE Transactions on CAD*, V30, 1, 2011, pp. 145-158.

[8] Biswas, S. and Blanton, R.D. Statistical Test Compaction Using Binary Decision Trees *IEEE Design & Test of Computers*, vol 23, issue 6, 2006, pp. 452-462.

[9] N. Sumikawa, D. Drmanac, L. Winemberg, L. Wang and M. Abadir. Important Test Selection For Screening Potential Customer Returns. International Symposium on VLSI Design, Automation and Test, 2011.

[10] D. Drmanac, N. Sumikawa, L. Winemberg, L. Wang and M. Abadir. Multidimensional Parametric Test Set Optimization of Wafer Probe Data for Predicting in Field Failures and Setting Tighter Test Limits. DATE, 2011.

[11] Nik Sumikawa, Dragoljub (Gagi) Drmanac, Li-C. Wang, LeRoy Winemberg and Magdy S. Abadir Forward Prediction Based on Wafer Sort Data. In *ITC*, 2011.

[12] Nik Sumikawa, Dragoljub (Gagi) Drmanac, Li-C. Wang, LeRoy Winemberg and Magdy S. Abadir Understanding Customer Returns From A Test Perspective. In *VTS*, 2011.

[13] A. Chakraborty and D. Pan Controlling NBTI degradation during static burn-in testing. *ASPDAC*, 2011.

[14] J. Figueras and A. Ferre Possibilities and Limitations of IDDQ Testing in Submicron CMOS. *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, part B, Vol. 21, No.4, Nov. 1998, pp. 352-359.

[15] A. Miller IDDQ Testing in Deep Sub-micron Integrated Circuits. *ITC*, 1999, pp.724-729.

[16] P. Maxwell et al. Current Ratios: A Self-scaling Technique for Production IDDQ Testing. *ITC*, 1999, pp. 738-746.

[17] R. Kawahara, O. Nakayama, and T. Kurasawa The effectiveness of IDDQ and high voltage stress for burn-in Elimination. *IEEE Workshop on IDDQ Testing*, 1996, pp. 9-13.

[18] S.S Sabade and D.M. Walker ; Evaluation of effectiveness of median of absolute deviations outlier rejection-based IDDQ testing for burn-in reduction. *VTS*, 2002, pp. 81-86.

[19] Li-C. Wang; Data Learning Based Diagnosis. *Asia and South Pacific Design Automation Conference*, 2010.