Title: Real Time Dynamic Application of Part Average Testing (PAT) at Final Test

Douglas Pihlaja
TriQuint Semiconductor, Hillsboro Oregon, USA (doug.pihlaja@tqs.com, 503-615-9229)

Keywords: … Statistics, Test, PAT

Abstract

The application of Part Average Testing (PAT) during semiconductor final test is typically applied to population data from numerous batches, called static PAT limits. This can lead to excessively wide distributions compared to batch by batch estimates. This paper will explain a method that allows the application of PAT limits dynamically using the specific distribution of the batch being tested. It also will look into PAT solutions used for distributions that are not Gaussian.

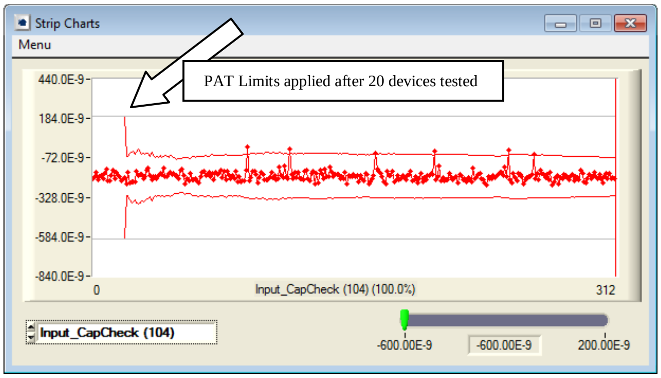
INTRODUCTION

PAT limits are well established in the automotive industry. They are used to separate parts that are different than what is "normally" being produced. That is to say they screen out the outliers from the typical population. The Automotive Electronics Council document AEC-Q001 [1] defines the standard for application of PAT limits in the automotive industry. The strict methodology of this document works fine when you are producing widgets with typical machining processes. In these cases the distributions are known to be Gaussian in nature and the output can be serialized for later selection. But in today's semiconductor final test process, devices are tested and then immediately put onto tape and reel for shipment to the customer. You must make your decision just after the individual unit is tested. In addition, in RF module manufacturing it is not unusual to have other than Gaussian distributions including ones that are highly skewed or bimodal in nature. Because of these issues the standard methods for application of PAT limits would lead to over rejection or worse yet, under rejection.

IMPLEMENTATION

We have addressed these issues in a custom test shell that runs between the native environment on the test system and the user GUI. The same shell runs on multiple test systems, from ASL and LTX ATE's to custom rack and stacks. This shell handles all statistical calculations as well as binning and data logging.

The basic method follows closely to AEC-Q001 in that the first 20 devices of a new batch are tested using the normal static limits. It is important to note that all devices are fed through a bowl handler and as such, devices are randomized within the batch. At this point a good estimate of the "robust" mean and standard deviation of key parameters can be calculated. Using these estimates, PAT limits for the batch are applied (Figure 1).



PAT Limits applied after 20 devices tested

Figure 1: Trend chart showing PAT limits applied after testing 20 devices.

The shell continues to refine the estimate as more devices are tested until the sample size reaches 200. Then a moving average of the last 200 devices is used for the "robust" estimates. In this way the limits move dynamically to allow for small system drifts due to socket wear without adversely effecting yield. To put it another way, the shell is answering the basic question:

"Is any given device different from the last 200?"

ROBUST MEAN AND SIGMA

Calculation of the "robust" mean and standard deviation (sigma) can be accomplished using many different methods.

We initial used an iterative process where the population mean and sigma were calculated. Then devices outside of a predetermined sigma value were discarded from the data set and the mean and sigma were calculated again. This process was then repeated a "few" times until the number of remaining devices in a population settled to the main population. This was typically around three iterations.

The problems with this method were that it did not work well with highly skewed populations and considering the large number of parameters being tested, was very computationally intensive. Another method had to be found.

As it turned out a colleague within TriQuint was working on this very problem. This method was described in the introduction section of CSMantech 2012 paper, *A Robust, Non-Parametric Method to Identify Outliers and Improve Final Yield and Quality* [2].

This method is also mostly described in AEC-Q001 where the "robust" sigma is described as:

Robust Mean = Median (Also known as Q2, the point half way through the ranked data)

Robust Sigma = IQR / 1.35

Where IQR ("Interquartile Range") = (Q3 – Q1)

(Q1 is the point 1 / 4 through the ranked data and Q3 is 3 / 4 through the ranked data.)

AEC-Q001 then describes the upper and lower PAT limits as:

PAT limit = Robust Mean +- 6 * Robust Sigma

We take it one step further:

Substituting (Robust Mean) Q2 = Q3 – IQR/2 or Q1 + IQR/2

Upper PAT limit = Q3 + 3.95*IQR
Lower PAT limit = Q1 – 3.95*IQR

We take this yet one more step to the general case. This is:

Lower limit: Q1 – ((Desired #Sigma – 0.675)/1.35)*IQR
Upper limit: Q3 + ((Desired #Sigma – 0.675)/1.35)*IQR

This gives us options to filter outliers and set our PAT limits using very simple sorting and arithmetic functions. It also is tolerant of distributions that are not Gaussian in nature.

This also allows for asymmetrical application of sigma values which is required for highly tailed and/or bimodal distributions.

LOSE ENDS

As testing continues any device that passes all spec limits but fails any dynamic limit is binned to a separate failure bin for retest and failure analysis as appropriate. The first parameter that failed dynamic limits is logged in the data file along with the limits that were applied at the time of testing. In addition, dynamic limit yield by parameter is also tracked within the shell for Pareto and future Statistical Bin Limit application.

We then had to consider the case of retest. Since retest devices are, by definition, not from the main population we could not use the same methods for arriving at the estimates of robust mean and sigma.

What we came up with is a way to periodically save the limits being applied in the shell back ground. Then upon closing of the virgin test of a given batch the last know set of PAT limits are saved to a text file for future access by the test shell. Then once retest has started this set of limits are applied to the entire retest sublot.

When we started beta testing this process we ran into a small conflict when retest was completed on a test system that was different than originally used to test the virgin batch. To fix this we sent the text file of the last known set of PAT limits to an FTP site on a server that is accessible to all testers. In this way the shell will first look locally for the known limits but if not found will go to the FTP server for these limits. This has proved to be effective.

INITIAL SETUP

In addition to the shell development that went into this process, a Spotfire^{TM} dashboard was developed to help with initial limit setting (Figure 2.).

This dashboard allows the import of historic batch data and facilitates enabling or disabling dynamic limits for any given parameter and allows for asymmetrical setting of sigma limits. This helps to minimize arbitrary yield loss while allowing for concentration on key quality parameters.

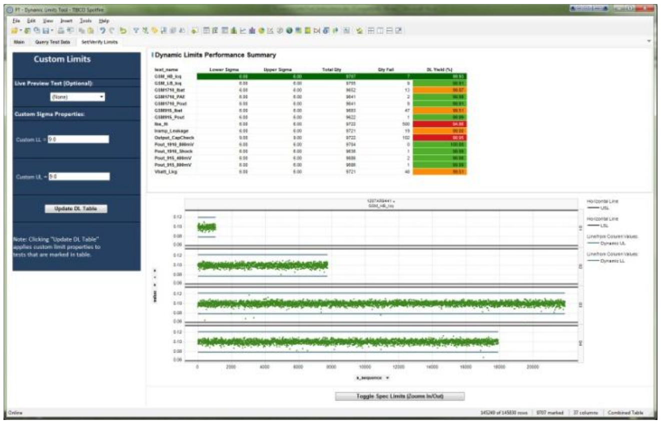
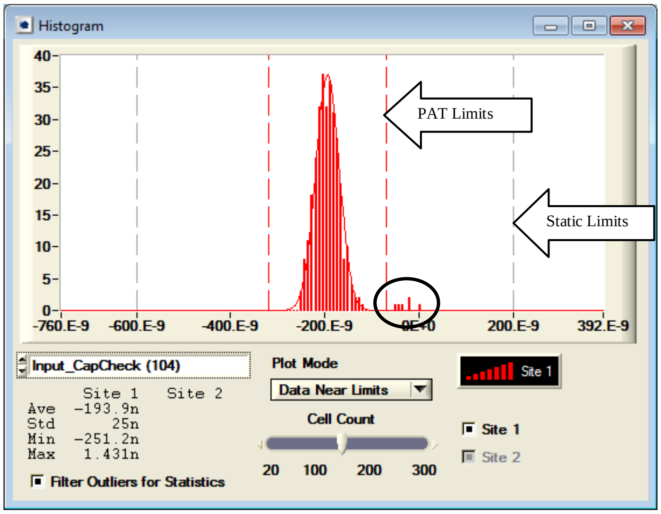


Figure 2: Example of Spotfire^{TM} dashboard showing yield and trend scenarios for PAT limit setup.

Once the sigma value for parameters are decided on then a separate script writes the limits file which includes static limits and enabling of PAT limits. For PAT limit parameters the applicable sigma values are also stated.

RESULTS

This system has been implemented on multiple platforms and products. Since its start we have tested over 250 million devices using this methodology. Results show a yield loss on the order of 0.2 % as expected. But devices that have otherwise passed the static specification limits but were clearly outliers from the main population have been effectively screened out to both keep them from shipment to the customer and to analyze for ongoing yield improvements (Figure 3).



```
         Site 1    Site 2
Ave   -193.9n
Std       25n
Min   -251.2n
Max    1.431n
```

☒ Filter Outliers for Statistics

Plot Mode: Data Near Limits

Figure 3: Devices circled passed static test limits but failed PAT limits and were effectively screened out.

ACKNOWLEDGMENTS

The author would like to thank Doug Knudsen for his excellent shell programming. He would also like to thank Brian Kleven for his dashboard work and Neill Patterson for his statistics.

REFERENCES

[1] Automotive Electronics Council doc, AEC-Q001, *Guidelines for Part Average Testing.*

[2] N. Patterson, *A Robust, Non-Parametric Method to Identify Outliers and Improve Final Yield and Quality* , 2012 GaAs MANTECH Digest, pp. 59-61, April 2012.

ACRONYMS

PAT: Part Average Testing
ATE: Automated Test Equipment
IQR: Interquartile Range
FTP: File Transfer Protocol

6
b