How to predict the Unforeseeable?

Dr. Gunther KARNER

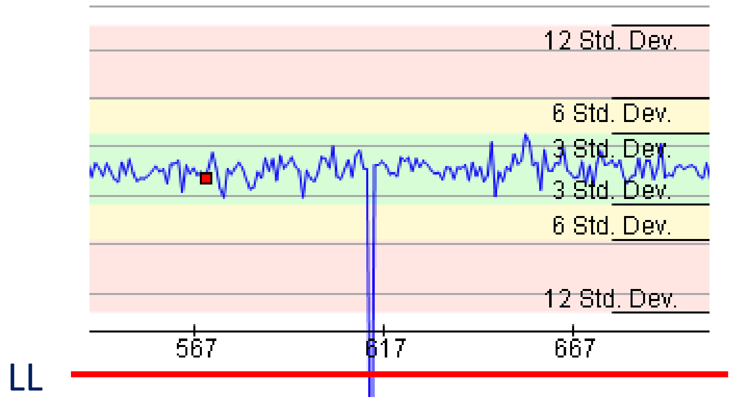
optimiSE GmbH

The Power of [x]

SEMICON® Europa2012

9–11 October
Messe Dresden, Germany

‘Random’ Fails – *the* Challenge in Adaptive Test



- Spot Failures in Devices under Test (DUTs) are among the most notorious sources of yield loss in semiconductor production.
- How to ‘control’ such a test?

By Sampling?

➔ Sampling rate of 10 → Cpk over of 10 fully tested parts before the fail equals 4.71!

→ incapable of predicting ‘random’ defects!

The ‘Miracle’ and what’s best done of it

→ ‘DUT Response Model’

- Devices, failing in a certain test, often ‘announce’ that deficiency in specific tests earlier in the flow by showing ‘abnormal’ readings!
- Thus, not (only) the history of a test may predict its actual result but the outcome of earlier tests on the same DUT!

Our Focus

- Real-time monitoring of specific **‘Signature Test’** results
- ‘Very short-term’ predictions
- Test flows are dynamically changed by adding or removing **Adaptive Test Groups**
- Execution of those groups is a function of prior <u>Signature Test readings on the same device</u>

Knowledge Correlations accentuate critical results

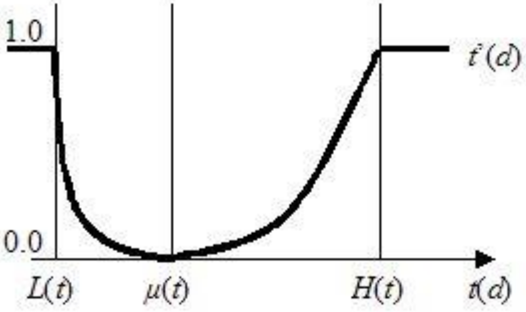
- The crucial ingredient in a reliable DUT Response Algorithm is the determination of the Signature Tests!
- Compute **Knowledge Correlations** from specific test results:

Define the normalized score $t'(d)$ of a test result $t(d)$ for a test t with limits $L(t)$, $H(t)$ and 'target value' of $\mu(t)$ as:

$$t'(d) = \begin{cases} 1 & \text{for } t(d) < L(t) \\ F(t)_{\text{lower}} & \text{for } L(t) \leq t(d) \leq \mu(t) \\ F(t)_{\text{upper}} & \text{for } \mu(t) \leq t(d) \leq H(t) \\ 1 & \text{for } t(d) > H(t) \end{cases}$$

- The function *F* is chosen to stress readings close or beyond test limits

→



Knowledge Correlations (KCs) point the way

- Strategy: Find Signature Tests by reconstructing measurements in Adaptive Test Groups on a part-by-part scale from the best knowledge-correlated tests
- The reconstruction vehicle is a standard multilayer neural net (→FANN)
- KCs perform better than ‘raw’ correlations when it comes to spot fails! (‘Raw’ indicates original test readings.)

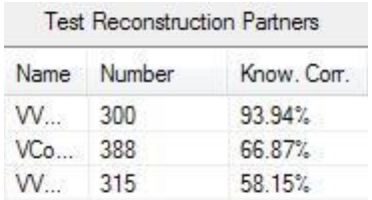
- Example:

Reconstruct a Test #340 by a Test #300 over one wafer, in particular in the vicinity of the spot fail

Name	Number	# Executions	# Fail Flags	Cpk	Raw Corr.	Know. Corr.
AD...	340	2027	1	1.45	81.59%	93.94%

Knowledge Correlations have the edge

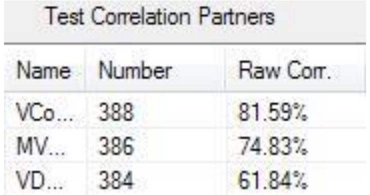
- Training of FANN over half the wafer not including spot fail



Test Reconstruction Partners

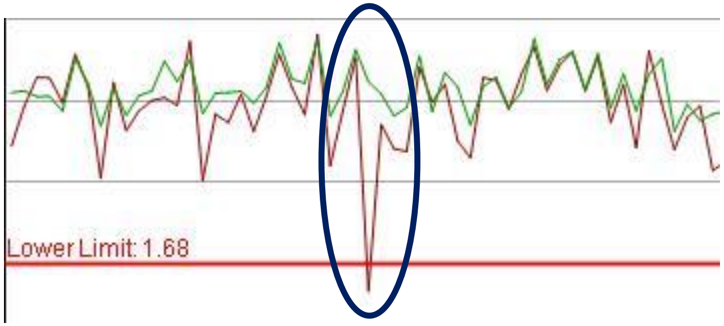
Name	Number	Know. Corr.
W...	300	93.94%
VCo...	388	66.87%
W...	315	58.15%

- Same setting, but highest raw-correlated test as predictor:



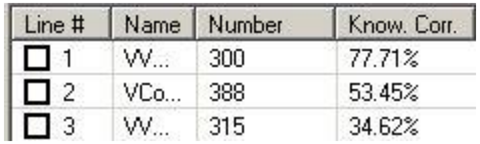
Test Correlation Partners

Name	Number	Raw Corr.
VCo...	388	81.59%
MV...	386	74.83%
VD...	384	61.84%

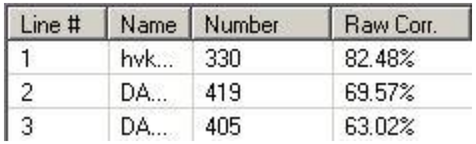


- If above is scalable → #300 is potential Signature Test

Na...	Number	# Executions	# Fail Flags
AD...	340	245827	53



Line #	Name	Number	Know. Corr.
☐ 1	W...	300	77.71%
☐ 2	VCo...	388	53.45%
☐ 3	W...	315	34.62%



Line #	Name	Number	Raw Corr.
1	hvk...	330	82.48%
2	DA...	419	69.57%
3	DA...	405	63.02%

The Decision Criterion

- Test #300 displays distinct spike at the part failing in Test #340, compared to its results just before

To qualify as a Numerical Indicator for future potential fails in #340, the graph suggests

- to evaluate the result of #300 at the fail device against a certain ‘local average region’ of #300
- to calculate that average from the #300-readings at a collection of parts tested just before the defective

The number of free parameters is just two:

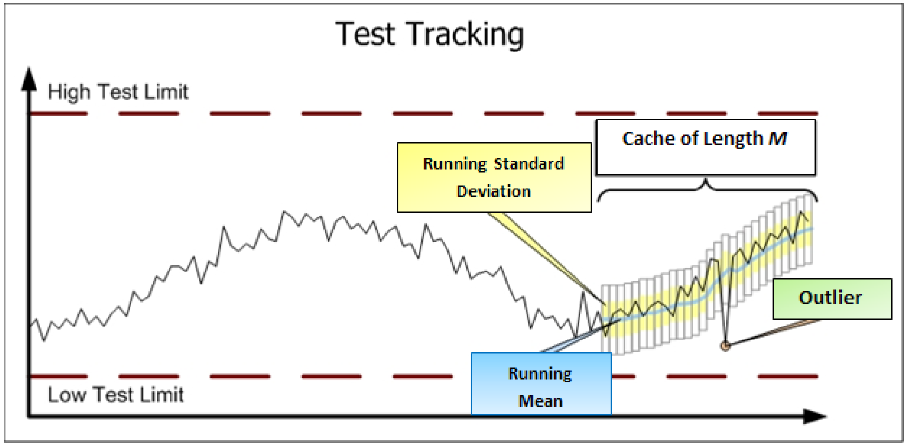
- The weight of the ‘average’ against the spike
- The size of the device window before the fail

→ Ockham's razor …

Feed-Forward Decisions in Real Time

The algorithm is executed in a die-by-die rhythm:

- The most recent results, say [1, *N* -1], of Signature Tests are kept in memory
- Those readings determine the ‘local average regions’, against which the results of the Signature Tests at Part *N* are assessed
- Depending on the ‘being inside or outside’ scores of the latter, a potential fail in a defect-correlated Adaptive Test Group at the Part *N* is predicted
- On Device *N+1* the procedure repeats itself, now with the set [2, *N*] of the Signature Test results as entry of the ‘local average region’ calculation
- And it keeps on going that way…



The Adaptive Group Test Controller

The decision algorithm is the core of a **Feed-Forward Controller**, called **Adaptive Group Test Controller** (**AGTC**), where the latter

- generates the decision to execute Adaptive Groups or not on account of the Signature Test results received in real-time from the tester. Either uni- or multivariat, i.e.,‘$\vee$’ or ‘$\wedge$’ combinations of Signature Tests
- hands back in real-time the instruction to execute Adaptive Groups or not to the tester
- however, does not get information about success (i.e., avoiding escapes)

NOTE

- **No additional hardware** is required to run the **AGTC**, only a minor one-time adaption of the test program
- The **control over the Adaptive Group Testing remains** with the **owner of the test program** → Interesting for fabless companies…

Actual Realisation of the AGTC

- On a Catalyst, for example, a test group can be easily handled inside an “if” statement (1=execute, 0=skip), where *oatcDo* is the automaton call:

```
if (oatcDo(oatc,340)==1)
  seq Adaptive Group()
        {
     TestNr(340)…
        }
```

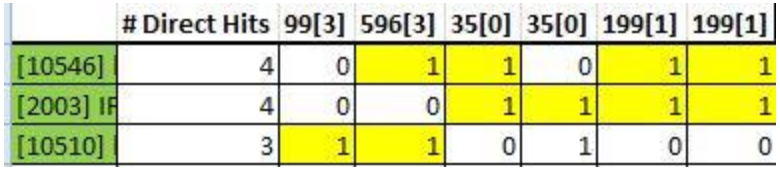
- What about test time overhead due to the AGTC ?
- On a standard Catalyst, a automaton decision takes about 0.3 ms
- Typically, two to four Signature Tests control an Adaptive Group
- The method works independently of the test regime!

A Showcase Simulation

A Group ‘ADC…’ is taken as **Adaptive Test Group**:

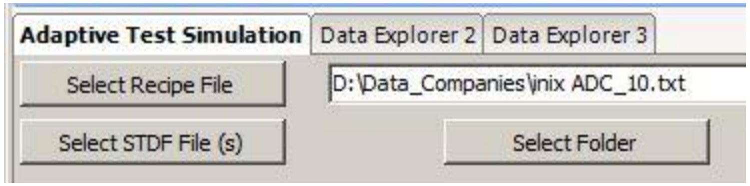
- Test Time ≅ 2 sec (app. 11% of total TT)
- 6 fails over 5 Wafer Sort lots (app.85k parts)

Two **Signature Tests** cover those fails:



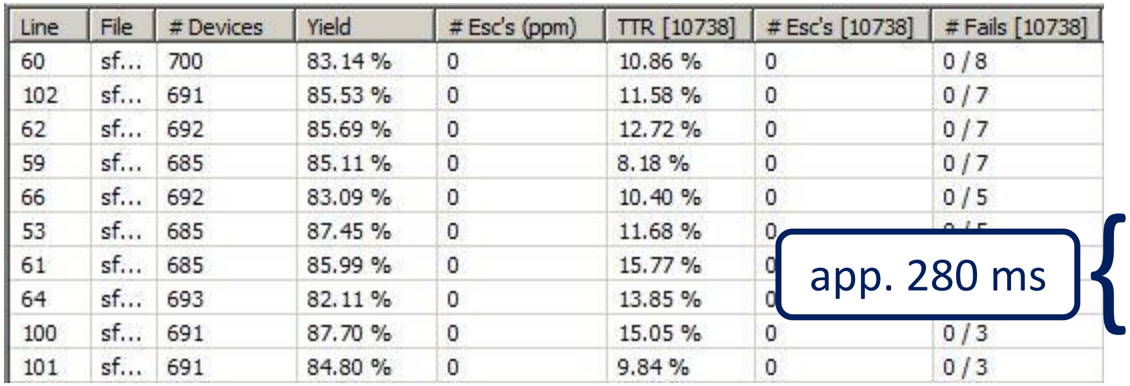
	# Direct Hits	99[3]	596[3]	35[0]	35[0]	199[1]	199[1]
[10546]	4	0	1	1	0	1	1
[2003] IF	4	0	0	1	1	1	1
[10510]	3	1	1	0	1	0	0

Monitoring of #2003 and #10510 permits the dynamic sampling of ‘ADC…’ over another set of five lots



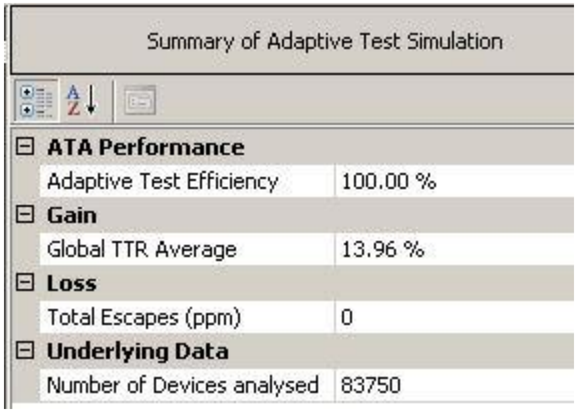
D:\Data_Companies\inix ADC_10.txt

Start with controller parameters on ‘save side‘:



Line	File	# Devices	Yield	# Esc's (ppm)	TTR [10738]	# Esc's [10738]	# Fails [10738]
60	sf...	700	83.14 %	0	10.86 %	0	0 / 8
102	sf...	691	85.53 %	0	11.58 %	0	0 / 7
62	sf...	692	85.69 %	0	12.72 %	0	0 / 7
59	sf...	685	85.11 %	0	8.18 %	0	0 / 7
66	sf...	692	83.09 %	0	10.40 %	0	0 / 5
53	sf...	685	87.45 %	0	11.68 %	0	0 / 5
61	sf...	685	85.99 %	0	15.77 %	0	
64	sf...	693	82.11 %	0	13.85 %	0	
100	sf...	691	87.70 %	0	15.05 %	0	0 / 3
101	sf...	691	84.80 %	0	9.84 %	0	0 / 3

app. 280 ms

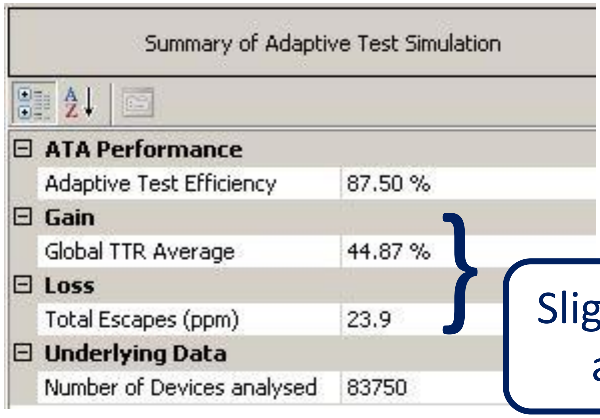


Summary of Adaptive Test Simulation

ATA Performance	
Adaptive Test Efficiency	100.00 %
Gain	
Global TTR Average	13.96 %
Loss	
Total Escapes (ppm)	0
Underlying Data	
Number of Devices analysed	83750

Is that Result good enough? More is in Store…

- Very little fails ➔ Statistics is a somewhat ‘lean’… Nevertheless, widening of controller parameters yields



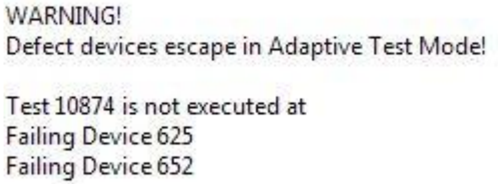
Summary of Adaptive Test Simulation

ATA Performance	
Adaptive Test Efficiency	87.50 %
Gain	
Global TTR Average	44.87 %
Loss	
Total Escapes (ppm)	23.9
Underlying Data	
Number of Devices analysed	83750

Slightly more risk: app. 900 ms

What about the two escapes?

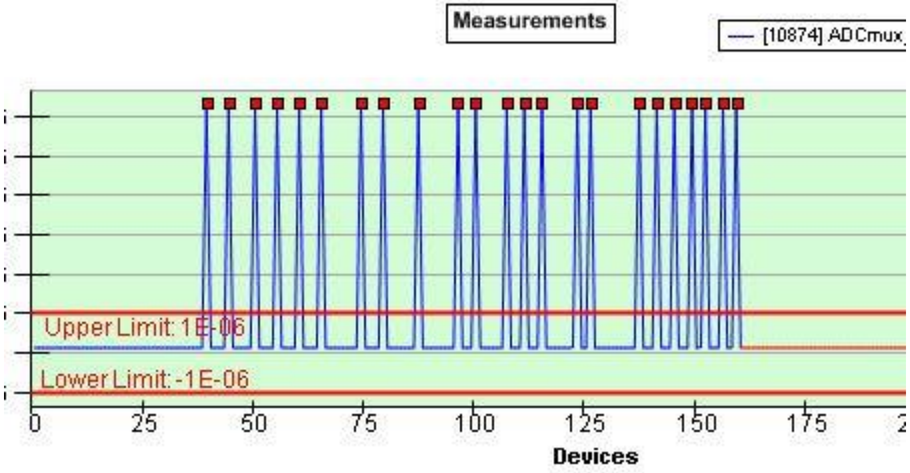
Simulator details show:



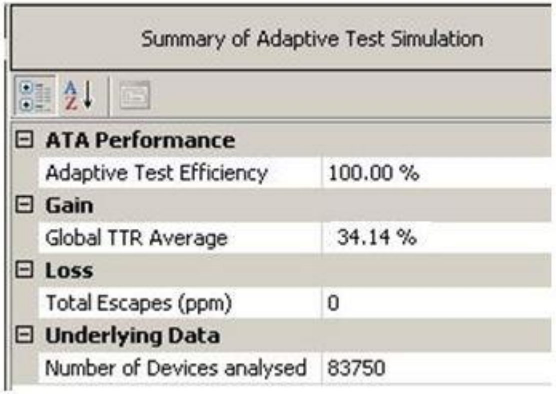
WARNING!
Defect devices escape in Adaptive Test Mode!

Test 10874 is not executed at
Failing Device 625
Failing Device 652

- A ‘local single wafer, single site’ problem:



➔ New analysis, add one Signature Test…o.k.!



Summary of Adaptive Test Simulation

ATA Performance	
Adaptive Test Efficiency	100.00 %
Gain	
Global TTR Average	34.14 %
Loss	
Total Escapes (ppm)	0
Underlying Data	
Number of Devices analysed	83750