DATA MINING AND DIAGNOSING IC FAILS

FRONTIERS IN ELECTRONIC TESTING

Consulting Editor
Vishwani D. Agrawal

Books in the series:

Embedded Processor-Based Self-Test
D. Gizopoulos
ISBN: 1-4020-2785-0
Testing Static Random Access Memories
S. Hamdioui
ISBN: 1-4020-7752-1
Verification by Error Modeling
K. Radecka and Zilic
ISBN: 1-4020-7652-5
Elements of STIL: Principles and Applications of IEEE Std. 1450
G. Maston, T. Taylor, J. Villar
ISBN: 1-4020-7637-1
Fault Injection Techniques and Tools for Embedded systems Reliability Evaluation
A. Benso, P. Prinetto
ISBN: 1-4020-7589-8
High Performance Memory Memory Testing
R. Dean Adams
ISBN: 1-4020-7255-4
SOC (System-on-a-Chip) Testing for Plug and Play Test Automation
K. Chakrabarty
ISBN: 1-4020-7205-8
Test Resource Partitioning for System-on-a-Chip
K. Chakrabarty, Iyengar & Chandra
ISBN: 1-4020-7119-1
A Designers' Guide to Built-in Self-Test
C. Stroud
ISBN: 1-4020-7050-0
Boundary-Scan Interconnect Diagnosis
J. de Sousa, P.Cheung
ISBN: 0-7923-7314-6
Essentials of Electronic Testing for Digital, Memory, and Mixed Signal VLSI Circuits
M.L. Bushnell, V.D. Agrawal
ISBN: 0-7923-7991-8
Analog and Mixed-Signal Boundary-Scan: A Guide to the IEEE 1149.4 Test Standard
A. Osseiran
ISBN: 0-7923-8686-8
Design for At-Speed Test, Diagnosis and Measurement
B. Nadeau-Dosti
ISBN: 0-79-8669-8
Delay Fault Testing for VLSI Circuits
A. Krstic, K-T. Cheng
ISBN: 0-7923-8295-1
Research Perspectives and Case Studies in System Test and Diagnosis
J.W. Sheppard, W.R. Simpson
ISBN: 0-7923-8263-3
Formal Equivalence Checking and Design Debugging
S.-Y. Huang, K.-T. Cheng
ISBN: 0-7923-8184-X
Defect Oriented Testing for CMOS Analog and Digital Circuits
M. Sachdev
ISBN: 0-7923-8083-5

DATA MINING AND DIAGNOSING IC FAILS

LEENDERT M. HUISMAN
IBM Systems and Technology Group

Springer

Library of Congress Cataloging-in-Publication Data

A C.I.P. Catalogue record for this book is available
from the Library of Congress.

ISBN-10: 0-387-24993-1
ISBN-13: 9780387249933

ISBN-10: 0-387-26351-9 (e-book)
ISBN-13: 9780387263519

Printed on acid-free paper.

Printed in the United States of America.

9 8 7 6 5 4 3 2 1 SPIN 11379102

springeronline.com

To Ans and Niels

Contents

List of Figures

List of Tables

Preface

This book grew out of an attempt to describe a variety of tools that were developed over a period of years in IBM to analyze Integrated Circuit fail data. The selection presented in this book focuses on those tools that have a significant statistical or datamining component. The danger of describing statistical analysis methods is the amount of non-trivial mathematics that is involved and that tends to obscure the usually straigthforward analysis ideas. This book is, therefore, divided into two roughly equal parts. The first part contains the description of the various analysis techniques and focuses on ideas and experimental results. The second part contains all the mathematical details that are necessary to prove the validity of the analysis techniques, the existence of solutions to the problems that those techniques engender, and the correctness of several properties that were assumed in the first part. Those who are interested only in using the analysis techniques themselves can skip the second part, but that part is important, if only to understand what is being done.

Several of the analysis techniques presented here were described previously in journal and conference articles: SLAT was described in [6][1] and [30][1], Embedded Object Analysis and Commonality Analysis were presented briefly in [8][1] and [31][1], respectively, and the relationship between coverage and yield was explored in [28][1] and [29]. The treatment in this book adds many details, and corrects some errors in the previous publications. The work presented here was not the work of its author alone, as is clear from the list of contributors in the Acknowledgements. The details of the mathematical analysis and of the analysis of the experimental data, however, are. Consequently, any errors are his responsibility.

LEENDERT HUISMAN
IBM Systems and Technology Group

[1] ©2004 IEEE

Acknowledgements

This book was written over a period of years, and has benefitted from many discussions with my colleagues. The chapter on Commonality analysis is based on a collaboration with Maroun Kassab and Leah Pastel, and the chapter on Instance Analysis on one with Greg Bazan, Francis Gravel, Anne Pardee, Leah Pastel and Ken Rowe. The chapter on SLAT arose from several discussions with Tom Bartenstein, Doug Heaberlin, David Sliwinski and Kevin Stanley. The explanation of why the stuck-at fault model is so succesful is the result of many lunch-time discussions with Peter Wohl.

Chapter 1

Introduction

Diagnosis is the extraction of information from fail data. In this book, the things that fail are Integrated Circuits (ICs), and the failures are those that occur during manufacturing test; more precisely, during the application of specially designed series of electrical stimuli to the integrated circuits on finished wafers or in separate modules. The information that is extracted concerns the causes of the failures. It can vary from the simplest, like the average number of distinct defects on an IC, to the most detailed, for example, the exact location and nature of the anomalies that caused the failure. The information that can be extracted depends on the tests during which the IC failed, and on the amount of fail data collected.

Extracting information from fail data can take many forms. For example, the failure probabilities of identifiable units on the device, like embedded memories or PLLs, can be estimated; different fail mechanisms may be identified by comparing fails at different phases of the test, or by analyzing the patterns of passes and fails on wafers; groups of chips may be identified that seem to share an underlying fail mechanism that did not affect other chips; if enough fail data has been collected, the location or the nature of the defect that caused the fail may even be estimated.

All these forms of extracting diagnostic information help identify the nature and the causes of the defects that occur on integrated circuits. Clustering chips into groups that seem to have failed similarly provides a first estimate of what defects occur, even though the identity of the defects is not yet known. Estimating occurrence probabilities helps focusing on the most prevalent failure mechanisms. And determining the location of the real defect then makes it possible to study the actual defect mechanism: why they occur, and how they affect the electronic circuit.

What is learned during such an analysis is fed back to the process or design engineers, so design and/or manufacturing changes can be made to prevent, or, at least, decrease the possibility of future occurrences of similar defects. Such changes increase the yield, that is, lead to fewer failing ICs, but may also increase the reliability of the finished products.

There are many types of fail data that can be used in diagnostic activities. Not all of the data will unambiguously identify a cause. Some of the data can only point in certain directions, like the probability that some defect, known only through its effects, will occur, or the most likely location on the device where this defect might be situated.

Thus defined, diagnosis covers a wide range of activities.This book will be limited to only one of them, software based analysis of routinely collected fail data during manufacturing testing. Fail data collected at the tester is usually very limited, showing only which phase of the test exposed the defect, and just enough to establish that the device indeed failed the test. The manufacturing tests are sometimes enhanced, however, with additional data collection to support diagnosis. Such additional data collection can be done systematically on a fraction of the (failing) devices, or can be done automatically in response to certain types of fail events. This added information is still part of routine data collection, and will of course also be considered here. In fact, one of the purposes of this book is to determine what additional data collection gives the most diagnostic benefit for the lowest added cost.

Other diagnostic techniques, like probing the failing IC or interactively applying tests based on what has been learned already about the cause of the fail will not be addressed. Such techniques are very important, and are almost always required in the end for the final determination of the cause of the fail, but they fall outside the scope of this book.

The process of diagnosis starts with applying tests to batches of chips. The test sequence is usually the same for all devices, even though the data collection strategy may not be, and is typically divided into a number of steps. The first few steps are intended to verify that the tester probe has made good electrical contact with the chip, and that there are no gross defects that would cause a large current to flow through the pins of the probe. The observations in these initial steps, called the gross tests, consist of measurements of currents and voltages. The results of the measurements are compared to predefined ranges of allowed values, and a chip fails any of the gross tests if the corresponding measurement is outside its accompanying range. If a failure does occur, there is either no good contact between the tester and the internal electronics of the device, or the current flowing through the probe and the contacts on the chip is so large that it might damage the probe. In either case, no further testing can be done.

If the chip does not fail the gross tests, more detailed tests are applied. These are generally of two kinds. Tests of the first kind consists of further measurements, for example of IDDq currents, flush delays, or ring oscillator frequencies. Those of the second kind consists of patterns that exercise selected portions of the chip electronics. A pattern consists of the application of a series of electrical stimuli to the inputs of the chip, and the observation of the electrical responses at its outputs. It is generally digital in nature, and exercises some portion of the chip electronics directly, as in deterministic (like LSSD) tests, or causes the chip electronics to generate internal patterns, as in Built-In Self Test (BIST). The chip can fail any of the patterns applied to it.

The tests can be enriched further by applying portions of the patterns and measurements at different voltages and temperatures. The fails occurring at one voltage or temperature should be distinguished from fails occurring at other voltages or temperatures, because they may be caused by different defect mechanisms. Also, the chip may fail some pattern at one voltage, but pass the same test at some other voltage, and this pattern of corner specific passes and fails provides more information about the defect.

The test sequence consists therefore of a large number of steps: the measurements in the initial phase, and subsequent patterns and more measurements. A chip that fails any of these steps is defective, and should be removed. The removal need not be immediate, however, unless the failing test is one of the gross tests, as it might be desirable for diagnostic purposes to obtain the response of a chip to other tests in the sequence.

The responses of the devices to the patterns and measurements are recorded in the test history, which is the complete record of which chips failed at which step. If the number of steps is large, as it typically is because of the large number of patterns, they can be grouped together into more meaningful major steps. Examples are all the deterministic patterns at a given voltage and temperature, the measurements of the oscillator frequencies, and detailed IDDq measurements. Each such major step is assigned a code, and the test history is merely a listing of the codes for each chip, indicating which major steps it failed. Each failing chip is also assigned an overall code, called the sort code, which is the code of that major test step at which the chip failed for the first time in the test sequence.

Various types of analysis can be performed on the accumulated fail data, and four different types will be discussed in this book. The first type consists of statistical analyses of the fail data of large numbers of devices that were all tested with the same test sequence. Examples of such large groups of devices are the chips on a single wafer, or on all the wafers in a lot. The second type is that of the spatial patterns of passes and fails on wafers, and the subsequent classification of those patterns into random ones and ones that are distinctly non-random. The third analysis type is that of potential commonalities between fails; that is, the attempt to identify common fail mechanisms by comparing fail syndromes. The fourth and final analysis type to be discussed in this book is using the fail syndrome of a particular device to identify the location of the defect that caused that device to fail.

Statistical analysis fits very naturally in the IC test process, because much data that is collected is statistical in nature. Examples are the fractions of good devices on individual wafers, the number of devices that were exposed for the first time by a certain pattern, and the number of times a given object on the device, like an embedded memory, failed across all the devices on a wafer. The statistical analysis is largely descriptive, and its purpose is, broadly speaking, to establish the normal behavior of the ICs, that is, to describe how the ICs, on average, respond to the manufacturing test.

Establishing a normal behavior of the real hardware has two benefits. First, it makes it possible to compare that normal behavior to what is expected based on our understanding of the manufacturing process and test. Such a comparison can then either confirm our understanding, or point at limitations thereof; for example, holes, or, even worse, completely erroneous ideas of what goes on during manufacturing and test.

The second benefit is that, once the normal behavior has been established, it is possible to estimate normal statistical fluctuations around the expected normal values, and to separate those from the truly deviant ones. In other words, it makes it possible to identify ICs that truly deviate from the normal behavior, even when taking statistical fluctuations into account. Non-normal fluctuations often point at systematic problems, and finding them is the first step in identifying, and then removing the underlying cause.

This book uses a limited number of statistical distribution and techniques. Their main features are briefly reviewed in Chapter 2.

The most immediate statistical analysis, taken up in Chapter 3, is that of the distribution of the yields, that is, of the fractions of good devices on single wafers. Using pass/fail information only is a poor use of the test history, however, and a more detailed statistical analysis of the distribution of all the sort codes will be taken up subsequently in the same chapter.

The device yield depends on the area of the device: the larger the area, the lower the yield. It has been noted over the years that this area dependence is more complex than a simple Arrhenius factor, and the cause of this complexity is usually assumed to be some clustering of the defects over the wafer. Clustering of the defects is, of course, related to what causes the defects in the first place, and this area dependence may, therefore, provide additional information about the defect mechanisms. It is discussed in some detail in Chapter 4.

Even more detailed yield information can be obtained when the device contains identifiable embedded objects that are tested separately by specific portions of the test sequence. Examples of such units are embedded memories and scan chains. The former are tested using specific memory tests, applied from the tester or generated on chip (memory BIST). The latter are tested with specific scan tests before the beginning of scan based testing.

Such object oriented yield analysis is valuable for all the reasons mentioned above regarding device yield analysis. There is a normal rate at which such objects are expected to fail, and any significant deviations from the normal rates point at systematic problems. In addition, however, these objects often have a size associated with them, like the number of cells in an embedded memory, and this size makes it possible to compare the failure rates of the objects not just to their counterparts on different chips, but also to different ones on the same chip that are similar in nature but have different sizes. The statistical analysis associated with embedded objects is the topic of Chapter 5.

In addition to rates, the fails of chips or objects on chips are distributed in some fashion over the wafer. This distribution is, of course, limited by the locations of the devices on the wafer, but, with enough such devices, distinct patterns can still be recognized. The primary goal of analyzing fail patterns is to identify those patterns that deviate significantly from a random one, and, therefore, may indicate some systematic problem. In addition, many process problems lead to distinct patterns of fails over the wafer that can be classified. Fairly standard pattern recognition techniques can then be used to identify the presence of such patterns, and, thereby, the potential occurrence of the associated process problem. This type of analysis is treated in Chapter 7.

The goal of commonality analysis, the third type of analysis to be considered in this book, is almost the opposite of statistical analysis: cluster the devices with similar fail syndromes into separate groups. There is no expectation of normality, or conformance to some model, but, instead, the fail data, whatever they are, are taken as signs of the underlying defect, and used to identify instances of the same or similar defects.

This clustering is important, because it attempts to catalogue the types of defects that occur and to determine their occurrence rates. If the devices can be divided into groups at least some of which are large and clearly separated, the obvious conclusion can be drawn that those larger groups correspond to unique fail mechanisms that need to be investigated further. Once such clusters have been identified, they can be selected for further, more detailed analysis. Various forms of commonality analysis are discussed in Chapter 6, with additional examples briefly mentioned in Chapter 3 and Chapter 5.

The most detailed analysis that can be done, and that is still statistical in nature, occurs when there is a notion of coverage. Coverage is a number between 0 and 1 that is attached to any initial section of the scan based patterns (that is, all patterns up to and including some selected one), and that is equal to the fraction of defects that are exposed by the patterns in that initial section. It exists, in particular, for that portion of the test that uses scan based patterns, and for them coverage is in fact routinely calculated.

Coverage clearly is related to the fraction of devices that fail during the application of the patterns in such an initial section. The form of that relationship depends on the nature of the defects, and, consequently, analyzing the progressive fallout when scan based patterns are applied should give useful information about the defects. Chapter 8 will address both the relationship between coverage and fallout, and how to use this relationship to extract defect specific information.

Chapter 9 is devoted to the fourth type of fail data analysis, that of using the collected fail data to identify the location of the defect. This type of analysis is far more complex than the previous types, because it uses a detailed logical model of the device, in addition to the fail data. Consequently, it is far more time consuming, and places far more stringent requirements on the fail data that need to be collected for it to be applicable. On the other hand, if successful, it can locate the defect exactly within the device, and is one of the main enablers of a successful physical failure analysis.

The simplest form of logic diagnosis is that based on the single stuck-at fault model. It gives good results surprisingly often, even though many realistic defects cannot be modeled by single stuck-at faults. Chapter 9 discusses this approach in detail, even though more powerful logic diagnosis techniques are available, because it is the classical form of logic diagnosis, and because many of the issues that complicate more powerful techniques already occur here.

SLAT is a far more powerful logic diagnosis technique that relies on two assumptions. The first assumption is almost an observation, and states that any defect behaves as some set of stuck-at faults under the application of any particular pattern that detects it. The defect may, and often will behave as different sets of stuck-at faults with different detecting patterns. The second assumption is that there will be some detecting patterns that cause the defect to behave as a single stuck-at fault. This assumption is the crucial one, for it reduces logic diagnoses to the standard problem of stuck-at fault diagnosis discussed in Chapter 9. It is more complicated than the latter one, though, for each detecting pattern has to be diagnosed as if it is the only one available. SLAT is the careful simultaneous analysis of all these single pattern diagnoses, and will be described in Chapter 10.

The discussion in this book is ordered roughly according to the amount of detail and the computational effort used in the various analyses. This ordering corresponds more or less to what a diagnostic engineer might do when faced with a large volume of failing devices, and having to find the main causes of the fails. The corresponding flow is shown in Figure 1.

Design data are needed to generate the test sequence and in some of the diagnoses. Not all diagnostic techniques require design data, though. Yield analysis, for example, does not need it at all. Most diagnostic techniques do, however, and those that can be done in its absence may still increase their effectiveness when design data is available. Its importance for the various

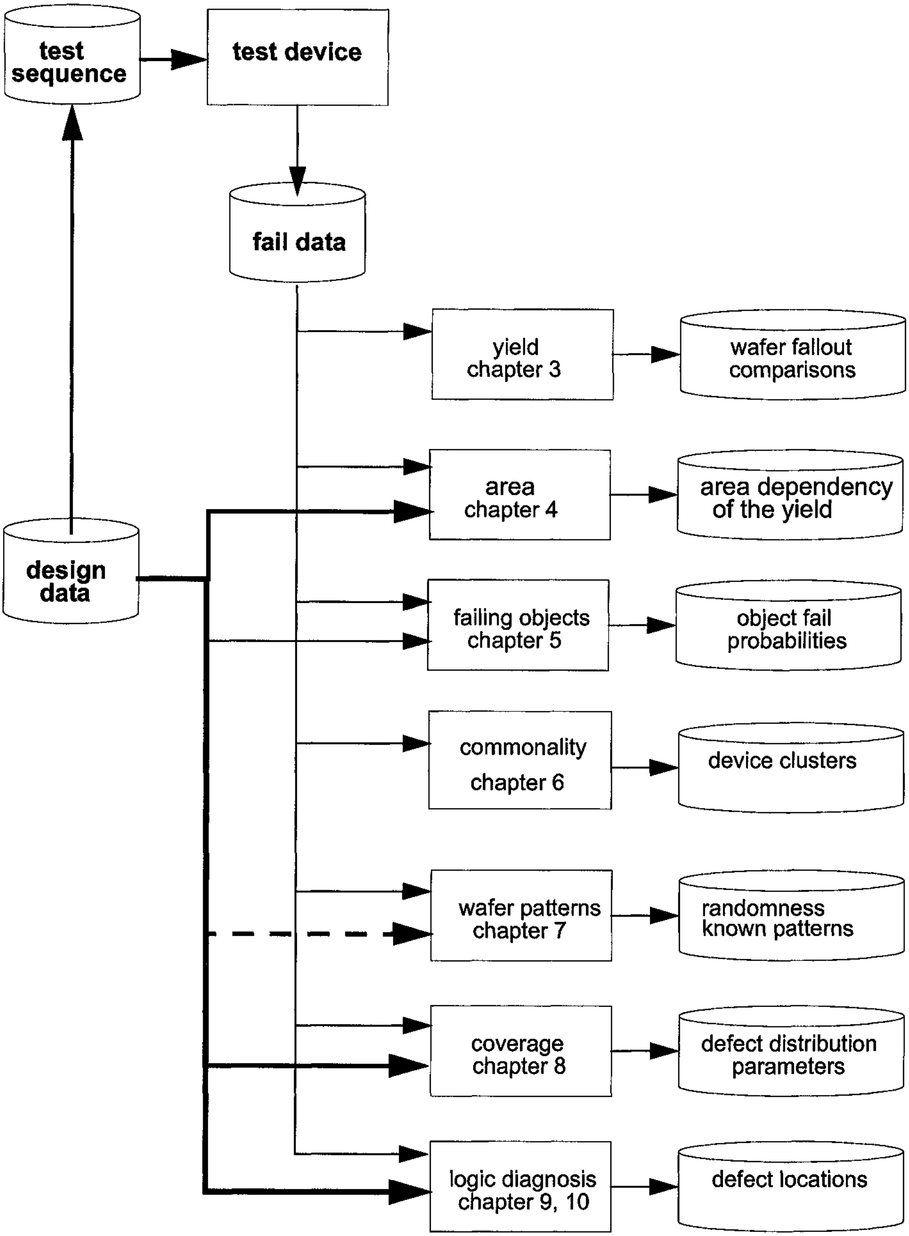
test sequence
test device
fail data
design data
yield chapter 3
wafer fallout comparisons
area chapter 4
area dependency of the yield
failing objects chapter 5
object fail probabilities
commonality chapter 6
device clusters
wafer patterns chapter 7
randomness known patterns
coverage chapter 8
defect distribution parameters
logic diagnosis chapter 9, 10
defect locations

Figure 1 Schematic diagnostic flow

diagnostic analysis techniques is indicated roughly by the heaviness of the arrows from the design data icon to the analysis icons. The results of the various analyses is indicated briefly in the data icons on the right.

The diagnostic techniques to be discussed in this book have certain data collection requirements, and these requirements have repercussions for both design and test. For test, because the required data needs to be collected and made available to the diagnostic software; for design, because it has to be possible to collect the test data, and because certain design data have to be available during diagnosis. An example of the former is the latch contents after the application of a test pattern, while an example of the latter is the positions of those latches in the scan chains. It will become clear from the main text what those requirements are, but they will be briefly summarized in Chapter 11 for the sake of convenience.

Chapter 2

Statistics

In the present chapter, I will briefly review some statistical distributions that are used often in this book. I will also discuss some statistical techniques that are important in this book, but that may not be very well known. Good introductions to practically all the statistical techniques used here can be found in, for example, Lindgren [38], or Casella and Berger [10]. The group of techniques that are used most often is centered on the likelihood function, but in some instances bootstrapping will be used as well. They will be described briefly.

Many chapters in this book rely strongly on the difference between random variables and model parameters. To accentuate this difference, the general custom will be followed of labeling random variables with upper case letters, and parameters with lower case ones.

1 STATISTICAL DISTRIBUTIONS

The number of distributions used in this book is small, basically the binomial and Poisson distributions, and some variations on them.

1.1 Binomial and multinomial distributions

The binomial distribution is that of the number of fails in a given number of attempts, given the fail probability. To simplify notation, I will use Feller's one [22] for the probability density function of the binomial distribution. The probability that n fails will be observed in N tries if the fail probability is p is

$$b(n;N,p) = \binom{N}{n} p^n (1-p)^{N-n} . \tag{2.1}$$

The expected value of n is Np, and its variance is Np(1 - p).

When p is very close to 0 or 1, the relationship between the expected value of n and its fluctuations becomes very simple. When p is very small, it can be neglected with respect to 1. The standard deviation of n is then roughly equal to the square root of its expected value. Likewise, when p is very close to 1, the standard deviation of N - n is roughly equal to the square root of that number. In other words, when p is either very small or very large, the typical size of the variations in the number of the rarer events (failures with very low fail probability, passes otherwise,) is roughly equal to the square root of the number of those events, and does not depend on the number of the more common events.

The binomial distribution can be generalized by compounding [12]. In that case, the binomial parameter p is a random variable itself, with a probability distribution h(p). The expected value of p will be indicated by

$$\langle p\rangle = \int h(p)p\,dp\ , \tag{2.2}$$

and its variance by $\sigma^2(p)$.

The expected value of the number of fails in the compounded distribution equals N<p>, and its variance is equal to

$$N\langle p\rangle(1-\langle p\rangle)+N^2\sigma^2(p)\ . \tag{2.3}$$

The first term in this variance is the standard binomial one, the second one is the contribution from the finite width of h(p). It has the important consequence that, when N becomes large, the ratio of the standard deviation of the number of fails to its expected value does not go to 0, as in a pure binomial distribution, but, instead, to the finite ratio $\sigma(p)/\langle p\rangle$. Even with large N, therefore, the variability in the number of fails cannot be ignored, and can, in fact, be substantial.

Another extension of the binomial distribution is the multinomial [22] one, in which more than two outcomes are possible, each with their own probability of occurrence. There is no standard notation for this distribution. The one that will be used here was inspired by that for the binomial distribution. If there are k choices, with probabilities p_i for i = 1, ..., K, the probability P(n_1, ..., n_k) of n_i occurrences of choice i is given by the multinomial probability

$$m(\{n_i\};N,\{p_i\}) = \frac{N!}{\prod n_i!}\prod p_i^{\,n_i}\ , \tag{2.4}$$

where n! stands for the factorial of n, and all products are from i = 1 to k. The sets $\{p_i\}$ and $\{n_i\}$ obey the obvious sum rules $\sum_i p_i = 1$, and $\sum_i n_i = N$.

By summing over all n_i except one, say n_j, we find that the probability of n_j occurrences out of N trials equals $b(n_j;N, p_j)$ Consequently, the expected value of any n_i equals Np_i.

1.2 Poisson and compound Poisson distributions

The Poisson distribution is that of the number of occurrences of some event in a given space, given the probability of an occurrence in a unit amount of space, and given that occurrences are independent. Typical examples are the number of events in a given amount of time or the number of defects in a given area. The latter example is the important one in this book.

The probability of an occurrence in a unit amount of space is also called the strength of the Poisson distribution. When the strength is ν, the probability of n occurrences in a unit amount of space equals

$$\frac{\nu^n}{n!}e^{-\nu}. \tag{2.5}$$

The expected value and variance of n are both equal to ν. The probability of no occurrence is $e^{-\nu}$.

A more general version of the Poisson distribution is the compound Poisson distribution, in which the strength ν is itself a random variable with some distribution h(ν) [12]. The probability of n occurrences is then equal to

$$\int h(\nu)\frac{\nu^n}{n!}e^{-\nu}\,d\nu. \tag{2.6}$$

It is easy to show, by interchanging integration and summation, that the expected value μ of n is now equal to $\langle\nu\rangle = \int h(\nu)\nu d\nu$, and that its variance equals $\mu + \sigma^2(\nu)$, in which $\sigma^2(\nu)$ is the variance of the Poisson strength ν. Compounding, therefore, always increases the variance of the observed yields.

Another effect of compounding is to increase the probability of no occurrences at all, at least when $\langle\nu\rangle$, the expected number of occurrences stays the same. This probability equals

$$p_0 = \int h(\nu)\,e^{-\nu}\,d\nu\,. \tag{2.7}$$