

6 Der Ausbeute- und Zuverlässigkeitssimulator CALYPSO

Basierend auf den Abschnitten 2 bis 5 wurde ein Simulationsprogramm entwickelt, um einerseits die Kernelfunktion für beliebige Layouts und die damit verbundene zu erwartende Produktausbeute zu berechnen und andererseits Aussagen zur Wirkung von Zuverlässigkeitsdefekten, wie sie im Abschnitt 5.1 betrachtet wurden, zu erhalten.

Nach einer Auswertung bereits bestehender Berechnungsverfahren wird das neu entwickelte, modular aufgebaute, Programm CALYPSO (Critical Area, Lifetime and Yield Predicting Software) vorgestellt. Während einer Beschreibung der wichtigsten Module des Programms werden neue Verfahren zur Auswertung sogenannter Zuverlässigkeitsdefekte ("Soft-Faults") und "eingebauter Schwachstellen" ("build-in reliability") vorgestellt und deren Wirkungsweise an Beispiellayouts demonstriert.

6.1 Berechnung von Kernelfunktionen und kritischen Flächen

6.1.1 Bekannte Algorithmen

Es wurde schon im Abschnitt 2.3.1 darauf hingewiesen, daß die Kernelfunktion -und damit die kritische Fläche- nur für sehr einfache Layouts analytisch berechenbar ist. Zur Berechnung der kritischen Fläche A_{krit} für kompliziertere Schaltungen wurden in der Vergangenheit mehrere Verfahren entwickelt:

- Eine in [34] beschriebene Methode besteht darin, einzelne Teile des Layouts durch Mustererkennung zu identifizieren. Für die einzelnen Muster ist jeweils eine analytische Lösung für alle Defektgrößen bekannt. Erst wenn man das gesamte Layout mit erkannten Mustern abdecken kann, ist es möglich, eine exakte Lösung für A_{krit} anzugeben. Da Schaltungslayouts jedoch voneinander sehr verschieden sein können, ist es schwierig, immer alle zu erwartenden Muster im Programm vorrätig zu haben. Erschwerend kommt hinzu, daß die erkennbaren Layoutteile sehr einfach geformt sein müssen, um überhaupt eine analytische Lösung angeben zu können.

- In [33] wird ein weiteres Verfahren vorgestellt. Es beruht auf dem Prinzip, ein Layout in mehrere Teile zu zerlegen, deren kritische Fläche sich einfach berechnen läßt. Voraussetzung für dieses Vorgehen ist, daß es sich um ein sogenanntes Manhattan-Layout handelt. Das heißt, alle Figuren der Schaltung lassen sich durch Aneinanderfügen von Rechtecken erzeugen. Umgekehrt kann ein solches Layout solange zerlegt werden, bis nur noch rechteckige Polygone übrigbleiben. Um herauszufinden, ob ein Defekt mit dem Durchmesser d einen Kurzschluß verursachen kann, muß der Abstand zweier benachbarter Rechtecke bekannt sein. Es besteht also zunächst die Aufgabe, für jedes einzelne Rechteck eine Liste aufzustellen, in der die Abstände zu den anderen benachbarten Rechtecken festgehalten werden. Aus dieser Nachbarschaftsbeziehung läßt sich im weiteren Verlauf des Algorithmus die kritische Fläche berechnen.
- In [32], [111] und [113] betrachtet man zum Beispiel einen Defekt, der zwei benachbarte Leitbahnen kurzschließen soll. Ist sein Durchmesser genau so groß wie der Abstand der Leitbahnen zueinander, dann kann er das aber nur dann, wenn sein Zentrum genau in der Mitte des Leitbahnzwischenraumes liegt. Erst wenn der Defekt größer ist, als der zu überbrückende Leitbahnabstand, kann er aus dieser Lage etwas in Richtung der einen oder anderen Leitbahn verschoben werden. Es ergibt sich sozusagen ein Korridor zwischen den benachbarten Leitbahnen, in den der Defekt fallen muß, um wirksam zu werden. Dieser Korridor ist die gesuchte kritische Fläche. Man kann sich nun aber vorstellen, daß beide Leitbahnen um den Betrag des Defektradius vergrößert werden. Ab einem gewissen Radius beginnen sich die Figuren zu überlappen. Im Falle benachbarter Leitbahnen ergibt sich eine Überlappungsfläche, die dem erwähnten Korridor, also der kritischen Fläche entspricht. Auf diese Weise werden die kritischen Flächen aller sich überlappender Figuren ermittelt und addiert. Sollten sich benachbarte kritische Flächen überlappen, muß das natürlich beim Aufsummieren beachtet werden.

- Einen vollkommen anderen Ansatz machen Computerprogramme, die die kritische Fläche eines Layouts mit Hilfe der Monte-Carlo-Methode berechnen. Diese Programme stellen an das Schaltungslayout keine besonderen Ansprüche, wie das bei den bisher besprochenen Verfahren der Fall war. Sie erzeugen nach dem Zufallsprinzip Koordinaten für Defekte, deren Durchmesser in einem vorgegebenen Größenbereich liegt und prüfen die Wirkung auf das Layout. Es wird also für jeden generierten Defekt geprüft, ob er vielleicht zwei Leitbahnen kurzschließt oder eine Unterbrechung verursacht. Die Komplexität eines solchen in [39] beschriebenen Verfahrens ist allerdings enorm. Um auch großflächige Layouts dicht genug mit den virtuellen Defekten abdecken zu können, werden nicht selten einige zehntausend bis mehrere Millionen Defektkoordinaten erzeugt. Gelingt es, sowohl die Layoutdaten, also die Koordinaten der einzelnen Polygone, als auch die Defektkoordinaten in zwei sortierten Listen zu speichern, so kann man die Komplexität des Verfahrens einschränken. Da die Entwicklung aber zu immer größeren Schaltungslayouts führt, beanspruchen die Sortieralgorithmen für die Layoutdaten immer mehr Speicher und Rechenzeit. Eine Optimierung des Monte-Carlo-Verfahrens ist also nicht immer rentabel bzw. durchführbar.
- In [31] wird ein Verfahren beschrieben, bei dem die virtuellen Defekte nicht per Zufall erzeugt werden, statt dessen wird ein festes Raster über die gesamte Layoutfläche bzw. interessierende Layoutteile gelegt. An jedem Rasterpunkt werden die kleinsten Entfernungen r_1 und r_2 zu den nächstliegenden zwei Polygonen bestimmt. Dem Maximum r_{\max} aus r_1 und r_2 kommt eine besondere Bedeutung zu: Ein Defekt, dessen Mittelpunktskoordinaten auf dem Rasterpunkt liegen, kann zwei Polygone kurzschließen, wenn der Defektradius mindestens so groß wie r_{\max} ist. Jeder größere Defektradius führt ebenfalls zum Kurzschluß. Das heißt also, man speichert für jeden Rasterpunkt die Entfernung r_{\max} und kann schließlich angeben, bei wieviel Prozent aller Rasterpunkte die Entfernung r_{\max} in einem vorgegebenen Größenbereich ($\text{Min.} \leq r_{\max} \leq \text{Max.}$) liegt. Diese Angabe entspricht aber gerade dem in Abschnitt 2.3.1 definierten Begriff des Kerns, mit dem sich die kritische Fläche des Layouts berechnen läßt.

Der Vorteil dieses letztgenannten Verfahrens liegt darin, daß man in jedem Bereich des Layouts per Definition eine gleichbleibende virtuelle Defektdichte erreicht und deshalb die beim Monte-Carlo-Verfahren auftretenden statistischen Schwankungen der Defektdichte von vornherein ausgeschlossen werden. Außerdem muß man für einen Rasterpunkt nur einmal die Berechnung von r_{\max} durchführen und kann dann davon ausgehen, daß jeder größere Defekt an dieser Stelle mit Sicherheit einen Kurzschluß verursacht.

Problematisch gestaltet sich allerdings der Aufbau des Rasters: Einerseits dürfen nicht zu viele Rasterpunkte verwendet werden, um die Rechenzeit einzugrenzen, andererseits müssen die Abstände der Rasterpunkte so klein sein, daß selbst kleinflächige Gebiete zwischen den Polygonen korrekt behandelt werden. Die Maschengröße des Rasters richtet sich also nach dem kleinsten im Layout zu erwartenden Abstand r_{\min} zweier benachbarter Polygone. Ist dieser Abstand nicht bekannt, muß man entweder eine Analyse des Layouts vornehmen, die dann allerdings sehr zeitaufwendig wird, oder r_{\min} muß abgeschätzt werden. Generell läßt sich sagen, daß mit immer feiner werdenden Layoutstrukturen, die Anzahl der Rasterpunkte und somit die Rechenzeit nicht nur linear, sondern quadratisch ansteigen.

Bei Berechnungen an Metallebenen, die für Busleitungen verwendet werden, muß man beachten, daß diese zunehmend automatisch vom Layoutentwurfsprogramm erzeugt werden und deshalb einer gewissen Systematik unterliegen. Überlagern sich die automatisch erzeugten Busleitungen mit dem gewählte Raster, dann kommt es zu sogenannten Abtasteffekten, die bei der Berechnung der Kernelfunktion zu unerwünschten Effekten bzw. Verfälschungen führen können.

Weitere Simulationsprogramme, die ebenfalls nach den genannten Prinzipien arbeiten, sind in [36]-[38], [40], [42]-[44], [106], [107] und [110] beschrieben.

6.1.2 Programmübersicht

Auf den bisherigen Ausführungen aufbauend wurde ein neues Simulationsprogramm zur Designbewertung hinsichtlich zu erwartender Produktausbeute und Schwachstellenanalyse entwickelt. Die folgende Grafik soll die gedankliche Einordnung dieses Defektsimulators unterstützen. Es handelt sich dabei um eine prinzipielle Darstellung, die das Zusammenspiel der Eingangsparameter, des Simulators und der Endergebnisse verdeutlichen soll.

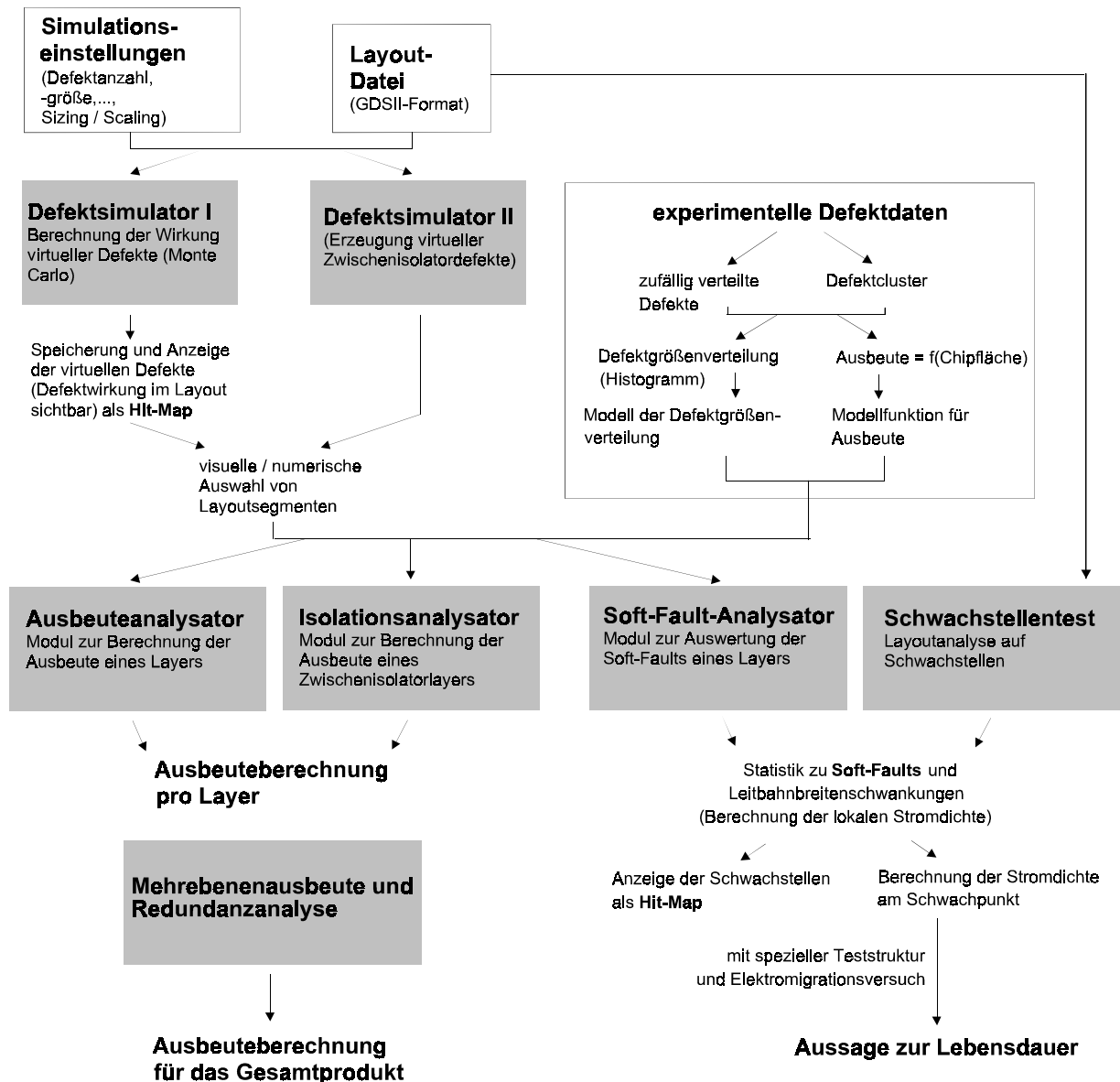


Abbildung 44: Prinzipskizze zur Darstellung des Zusammenspiels der Eingangsgrößen, des Defektsimulators und der Simulationsergebnisse

Man kann aus der Abbildung 44 erkennen, daß im wesentlichen drei Kategorien von Eingangsdaten benötigt werden. Es handelt sich dabei um die Simulationseinstellungen, eine Layoutdatei und Daten zu experimentell beobachteten Defekten. Das Programm selbst besteht aus mehreren Modulen, von denen hier nur die wichtigsten sieben hervorgehoben wurden. Die Funktionsweise dieser Module soll in den folgenden Abschnitten näher erläutert werden.

6.1.3 Defektsimulator I

Der Defektsimulator I dient dazu, Defekte innerhalb eines physikalischen Layers zu erzeugen und deren Wirkung auf das Layout (Kurzschluß oder Unterbrechung) in einer gesonderten Datei festzuhalten.

Der Simulator benutzt eine Kombination aus dem Monte-Carlo- und dem im Abschnitt 6.1.1 beschriebenen Rasterverfahren nach [31]. So werden die Defektkoordinaten entsprechend der Monte-Carlo-Methode per Zufall erzeugt, während die prinzipielle Auswertemethode des Rasterverfahrens erhalten bleibt.

Dadurch nutzt man einerseits den Vorteil der kurzen Rechenzeit des Rasterverfahrens und umgeht andererseits die Problematik, ein geeignetes Raster für das gesamte Layout bzw. die interessierenden Layoutbereiche zu finden.

Prinzipiell läßt sich der Algorithmus zur Berechnung der Kernelfunktion in folgende Schritte einteilen:

1. Zunächst wählt der Anwender eine ihn interessierende Ebene (Layer) des zu untersuchenden Layouts aus. Werden mehr als ein Layer ausgewählt, werden die jeweiligen Ebenen vom Defektsimulator für die Dauer der Simulation miteinander verschmolzen.
2. Da es nicht immer notwendig ist, die Kernelfunktion für das gesamte Layout zu berechnen -weil eventuell nur ausgewählte Bereiche von Interesse sind- besteht die Möglichkeit, den Bereich der zu erzeugenden Defektkoordinaten vorzugeben. Zusätzlich hat man die Möglichkeit, bei der Auswertung der erzeugten Ergebnisdatei erst später eine Auswahl einzelner Gebiete zu treffen.

3. Weiterhin legt der Benutzer die Anzahl der zu generierenden virtuellen Defekte fest. Per Definition wirkt dabei eine Hälfte der Defekte additiv (extra material) und die andere Hälfte subtraktiv (missing material).
4. Im nächsten Schritt wird die Layoutdatei gelesen. Die nachfolgend beschriebenen Schritte werden nur durchgeführt, wenn die gerade eingelesene Layoutfigur zum ausgewählten Layer gehört und im ausgewählten Layoutbereich liegt.
5. Wenn die aktuelle Layoutfigur in der Reichweite eines Defekts liegt -der größte anzunehmende Defektradius wird ja vom Benutzer vorgegeben- dann wird diese in geeigneter Weise gespeichert. Sollten bereits andere Polygone des Layouts in der Reichweite desselben Defekts gelegen haben, dann muß geprüft werden, ob das aktuelle Polygon mit den bereits gespeicherten Polygonen in einer direkten elektrischen Verbindung steht. Wenn das der Fall ist, müssen diese Polygone durch eine OR-Verknüpfung miteinander verschmolzen werden (merging).
6. Nachdem die bisherigen Schritte für alle Layoutfiguren abgearbeitet wurden, muß die für jeden Defekt angelegte Liste von Polygonen, die in dessen Reichweite liegen, nochmals auf sich berührende Polygone untersucht werden. Diese werden dann miteinander verschmolzen. Die Operation kann bei großen maximalen Defektausdehnungen (Vorgabe des Anwenders) sehr lange dauern, da dann viele Polygone in der Reichweite des Defekts liegen und somit in der Liste stehen können.
7. Nachdem die Polygonliste eines jeden virtuellen Defekts von sich überlappenden bzw. berührenden Polygonen "bereinigt" worden ist, wird für alle in der Liste verbliebenen Polygone geprüft, ob der jeweilige virtuelle Defekt mit seinem Zentrum innerhalb oder außerhalb des Polygons liegt.
8. Liegt der Defekt außerhalb der Figur, so wird der geringste Abstand zu dem aktuellen Polygon der Liste berechnet (Abstand r_1). Außerdem wird die Defektausdehnung bestimmt, die notwendig ist, um dieses Polygon zu unterbrechen (Abstand r_2). Bei einem Defekt, dessen Zentrum innerhalb der Layoutfigur liegt, werden die zwei kleinsten Abstände zu den sich gegenüberliegenden Polygonkanten in r_1 bzw. r_2 gespeichert (siehe Abbildung 45).

9. Falls es sich um ein komplizierter geformtes Polygon handelt, muß untersucht werden, ob der Defekt einen Kurzschluß innerhalb der Figur erzeugen kann. Die dazu notwendige kleinste Defektausdehnung wird festgestellt und gespeichert (Abstand r_3 , siehe Abbildung 45).
10. Nachdem für jeden Defekt die erwähnten Abstände gefunden wurden, beginnt die eigentliche Auswertung, die darüber entscheidet, ob ein Defekt einen Kurzschluß oder eine Unterbrechung verursacht. Dazu wird der Defektdurchmesser in einem Bereich, der durch den Anwender vorgegeben wurde, schrittweise verändert. Ist der resultierende Defektradius größer als einer der Abstände r_1 bis r_3 , dann erzeugt der betreffende Defekt einen Kurzschluß bzw. eine Leitbahnunterbrechung. Es handelt sich um einen die Ausbeute mindernden virtuellen Defekt.

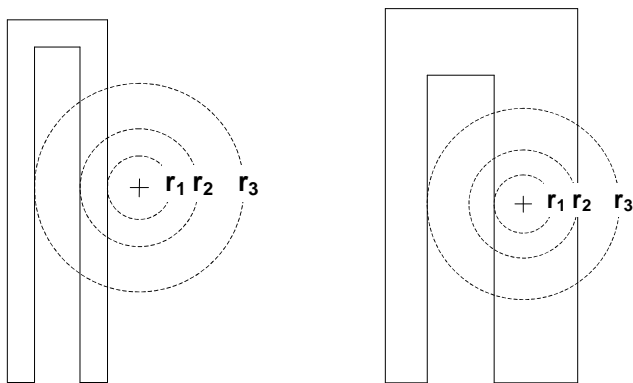


Abbildung 45: Bedeutung der Abstände r_1 bis r_3 ; weitere Erklärungen im Text; links: Defektzentrum außerhalb des Polygons, rechts: Defektzentrum innerhalb des Polygons

Sollte ein Defekt eine Leitbahn nicht vollständig unterbrechen oder zwei Leitbahnen nur "fast" miteinander verbinden, so soll er als ein "die Zuverlässigkeit mindernder Defekt" bezeichnet werden. Während manueller optischer Bewertungen realer Teststrukturen wurden, wenn auch in geringer Anzahl, solche Defekte tatsächlich registriert. Diese Art von Defekten ist in der Literatur unter der Bezeichnung "Soft-Faults" bekannt ([26], [82], [86], [87]). Sie lassen sich mit Defektmonitorstrukturen nicht feststellen. Nur Geräte, die auf optischer Basis arbeiten, sind *prinzipiell* in der Lage, solche "Soft-Faults" durch einen Bildvergleich mit originalen Layoutdaten zu erkennen.

Um einen Defekt als Soft-Fault zu klassifizieren, benötigt der Defektsimulator entsprechende Entscheidungskriterien. Es handelt sich dabei um jeweils zwei Angaben für subtraktive und additive Defekte:

1. Für subtraktiv wirkende Soft-Faults, also "fast" Leitbahnunterbrechungen, gibt man eine Restbreite der Leitbahn vor. Wird diese unterschritten, soll der Defekt als ausbeuterelevant eingestuft werden (siehe Abbildung 32).
2. Um einen Defekt als additiven Soft-Fault zu klassifizieren, muß er ein Polygon berühren bzw. in diesem Polygon liegen und über mindestens eine Polygonkante hinausragen. Dabei darf er sich aber nicht näher als eine vom Benutzer anzugebende Entfernung an ein benachbartes Polygon annähern. Wird diese Entfernung unterschritten, so soll der Defekt als Kurzschluß und damit als ausbeuterelevant betrachtet werden.

Die festgestellte Wirkung jedes einzelnen Defekts wird vom Defektsimulator in einer Ergebnisdatei festgehalten. Diese Datei soll im weiteren Text als Defektdatei bezeichnet werden. Sie enthält Angaben über die Defektkoordinaten, die Defektgröße und das Ergebnis (Kurzschluß / Unterbrechung) des oben beschriebenen Auswerteschrittes.

6.1.4 Modul zur Berechnung der Ausbeute im Leitbahnkomplex

Mit diesem sogenannten Ausbeuteanalysator ist der Anwender in der Lage, die Ausbeute für ein Layout zu bestimmen, für das vorher mit dem Defektsimulator I die Wirkung virtueller Defekte bestimmt und in einer Defektdatei abgelegt wurde.

Da ein Chip häufig in mehrere Funktionsgruppen aufgeteilt ist, erscheint es zweckmäßig, die Teilausbeuten dieser Funktionsgruppen getrennt voneinander zu bestimmen. So können verhältnismäßig leicht diejenigen Gruppen gefunden werden, die die Gesamtausbeute des Chips maßgeblich verringern. Der Ausbeuteanalysator bietet deshalb die Möglichkeit an, nur für ausgewählte Bereiche die Ausbeuteberechnung durchzuführen.

Der Ausbeuteanalysator stellt dem Anwender zudem mehrere verschiedene Defektgrößenverteilungsfunktionen zur Auswahl:

- $S(d) \sim d^{-n}$
- $S(d) \sim e^{-c \cdot d}$
- $S(d)$ nach Prabhu (vgl. [47])
- $S(d)$ nach Stapper (vgl. [48])
- neu entwickeltes Modell (wird im Abschnitt 7.1.4 beschrieben)

Die für das jeweils ausgewählte Modell nötigen Parameter müssen dem Anwender bereits bekannt sein. Im Analysator sind außerdem eine Reihe von Ausbeutemodellen integriert:

- Poisson-Modell (Gleichung (2.3))
- Modell von Seeds und Price (Gleichung (2.6))
- Negatives Binomialmodell (Gleichung (2.13))
- Modell von Okabe (vgl. [47])
- Modell von Murphy (vgl. [47], [48])
- Modell von Moore (vgl. [48])
- neu entwickeltes Modell (wird im Abschnitt 7.3 beschrieben)

Auch hier muß der Anwender die entsprechenden Parameter vorgeben (siehe dazu Abschnitt 7.3).

In der Praxis werden in relativ kurzen Zeitabständen Skalierungen des Layouts vorgenommen. Es wäre sehr zeitaufwendig, nach jeder Skalierung den Defektsimulator aufzurufen, um eine neue Defektdatei für das skalierte Layout zu erzeugen. Deshalb hat der Anwender an dieser Stelle die Möglichkeit, den gewünschten Skalierungsfaktor des originalen Layouts anzugeben. Die zu erwartende Ausbeute wird dann entsprechend neu berechnet.

Um für die berechnete Ausbeute den zugehörigen Fehler angeben zu können, übermittelt der Defektsimulator I die statistische Sicherheit der Kernelfunktionen für den Fall, daß sie für das gesamte Layout bestimmt werden soll. Wird zur Ausbeuteberechnung vom Anwender nur ein Layoutsegment ausgewählt, rechnet das Analysemodul diese Fehlerangaben entsprechend um. Als weitere Fehlerquelle ist vom Anwender der Fehler der Defektdichte anzugeben. Nun läßt sich für jedes Ausbeutemodell neben der Ausbeute auch deren Schwankung abschätzen.

Die in den Abschnitten 4.3 und 4.4 beschriebenen Effekte der Maßübertragungsfehler und des Sizing werden dabei natürlich berücksichtigt. Auch das unter 3.6 beschriebene Verfahren, die reale Defektform in die Ausbeutebetrachtungen einzubeziehen, ist in diesem Modul vertreten. Unter Abschnitt 7.3 wird auf die Wirksamkeit dieser Auswerteroutinen noch näher eingegangen.

6.1.5 Defektsimulator II

Der zweite Defektsimulator erzeugt Defekte, die zwischen zwei Ebenen des Layouts liegen und somit die Zwischenisolatorschicht schädigen können. Unter Defekten sollen hier sowohl Löcher im Zwischenisolator als auch darin eingeschlossene elektrisch leitfähige Partikel aus Metallisierungs- und Polyebenen verstanden werden. Dabei sind nur diejenigen Defekte von Interesse, die ein Polygon des unteren mit einem Polygon des oberen Layers verbinden können. Das heißt, der Defekt muß mindestens in der Überlappungsfläche der beiden gegeneinander isolierten Figuren liegen, um die Schaltung in ihrer Funktion zu schädigen.

Die hier erzeugten Defekte werden hinsichtlich dieses Kriteriums ausgewertet. In einer separaten Datei werden anschließend die Defektkoordinaten und die jeweilige Defektwirkung (Kurzschluß / keine Wirkung) abgelegt.

6.1.6 Modul zur Berechnung der Ausbeute von Zwischenisolatorschichten

Bei diesem Modul handelt es sich um eine Auswerteroutine der vom Defektsimulator II erzeugten Zwischenisolatordefekte. Der grundsätzliche Aufbau des Analysators entspricht dem im Abschnitt 6.1.4 beschriebenen Auswertemodul.

Der Anwender gibt zunächst auch hier einen ihn interessierenden Layoutbereich und den jeweiligen Zwischenisolator vor. Es wird nun überprüft, inwieweit die virtuellen Defekte innerhalb des ausgewählten Bereiches sowohl in einem Polygon der unteren als auch in einem Polygon der oberen Ebene liegen. Aus dem Verhältnis der insgesamt in diesem Bereich erzeugten Defekte zu den Defekten, die einen Schaden verursachen, läßt sich die Überlappungsfläche beider Layer bestimmen.

Um einen Kurzschluß zwischen beiden Ebenen zu erzeugen, muß der Durchmesser eines Defektes größer oder gleich der Zwischenisolatorschichtdicke sein, und der Defekt muß in der Überlappungsfläche der beiden gegeneinander isolierten Layer liegen. Defekte, deren Durchmesser kleiner sind als die Schichtdicke des Zwischenisolators, haben keine Auswirkung auf die Ausbeute. Ein Vergleich dieser Ausführungen mit denen des Abschnitts 2.3.2 zeigt, daß die Überlappungsfläche in ihrem Wesen der reduzierten Chipfläche entspricht (siehe auch [19], [25], [41], [70], [73], [113]).

Die jetzt noch fehlenden Komponenten zur Ausbeuteberechnung sind die Defektgrößenverteilung $S(d)$ und die Defektdichte D_0 . Um eine Größenverteilung der Zwischenisolatordefekte festzustellen, müßte man zwischen zwei Metallebenen die Schichtdicke des Isolators variieren und die Anzahl der jeweils registrierten Kurzschlüsse zählen. Wenn für jede Schichtdicke auch die Überlappungsfläche der oberen und unteren Ebene bekannt ist, kann man für die einzelnen Schichtdicken eine Defektdichte angeben.

In diesem Auswertemodul wird davon ausgegangen, daß die Schichtdicke des Isolationsmaterials zwischen der ausgewählten unteren und oberen Ebene konstant ist. Es reicht demnach aus, nur eine Defektdichte D_0 für die interessierende Schichtdicke anzugeben und für $S(d)$ folgende Formel zu benutzen:

$$S(d) = \begin{cases} 1 & d \geq d_{Schicht} \\ 0 & d < d_{Schicht} \end{cases} \quad (6.1)$$

d Defektdurchmesser

$d_{Schicht}$ Schichtdicke des Isolators

Der Analysator stellt nun dem Anwender wieder eine Anzahl Modellfunktionen zur Ausbeuteberechnung zur Auswahl und gibt die zu erwartende Ausbeute an. Zur Anwendung dieses Moduls sei auf den Abschnitt 7.4.2 verwiesen.

6.1.7 Mehrebenenausbeute und Redundanzanalyse

Im Abschnitt 4.5 wurde erläutert, wie sich die Ausbeute eines Schaltkreises ändert, wenn bestimmte Teile der Schaltung redundant ausgelegt werden. Zur Berechnung der Ausbeute können solche Designs mit dem ebenfalls implementierten Mehrebenen- und Redundanzanalysator untersucht werden. Der Anwender selektiert einen ihn interessierenden Bereich, gibt die entsprechende vom Defektsimulator I oder -II erzeugte Defektdatei vor und legt den Grad der Redundanz fest.

Wie der Name dieses Moduls bereits aussagt, können neben der Redundanzanalyse die Einzelausbeuten beliebiger Layer miteinander zur Produktausbeute kombiniert werden. Aufgrund der hohen Flexibilität dieses Moduls hinsichtlich der Möglichkeiten zur Bereichsselektion, der Ebenenkombination und der Redundanzanalyse können relativ leicht präzise Ausbeuteprognosen durchgeführt werden.

6.1.8 Module zur Anzeige der Defekte und des Layouts

Bei diesem Programmteil handelt es sich um ein Modul, das entwickelt wurde, um sich einen Überblick über die Lage und die Dichte der virtuellen Defekte, die von den Defektsimulatoren I + II erzeugt wurden, zu verschaffen. Zusätzlich zu den Defekten kann auch das Layout selbst angezeigt werden. So hat man eine gute Kontrolle über die vorhergesagte Wirkung (Unterbrechung / Kurzschluß) eines jeden Defekts. In der Abbildung 46 ist ein Ausschnitt eines speziellen Testlayouts dargestellt. Die Leitbahnen sind 2 µm breit

und auch $2\ \mu\text{m}$ voneinander entfernt. Im linken Teil wurden zusätzlich zum Layout die Mittelpunkte simulierter Defekte eingezeichnet, die mit einem Durchmesser von $3\ \mu\text{m}$ einen Kurzschluß am Layout verursachen. Man sieht deutlich, daß gewisse Bereiche des Layouts von vielen Defekten und andere wiederum von gar keinem Defekt betroffen sind. Der rechte Bildteil zeigt die Mittelpunkte von Unterbrechungsdefekten mit einem Durchmesser von $3\ \mu\text{m}$. Die Darstellung verdeutlicht, daß ein solcher Defekt nur dann die Leitbahnen unterbrechen kann, wenn er annähernd in der Mitte der abgebildeten Leitbahn liegt, denn nur so kann ein Defekt dieser Größe die Leitbahnbreite vollständig überbrücken. Die von den Defektmittelpunkten der linken bzw. rechten Bildhälfte belegte Fläche entspricht nach Abschnitt 2.3.2 der resultierenden kritischen Fläche für Defekte dieser Größe.

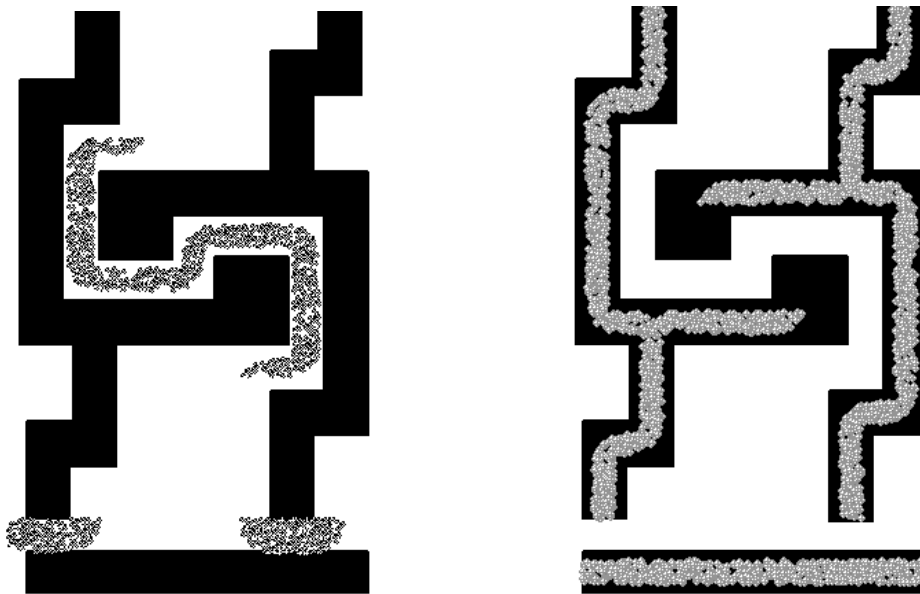


Abbildung 46: Ausschnitte eines Testlayouts zur Darstellung von Wirkungstreffern; links: Mittelpunktslage von Kurzschlußdefekten mit $d = 3\ \mu\text{m}$, rechts: Mittelpunktslage von Unterbrechungsdefekten mit $d = 3\ \mu\text{m}$

Eine Übersichtsanzeige zu allen die Ausbeute verringernden Defekten (nachfolgend auch als Wirkungstreffer bezeichnet) deckt meist auf, daß es gewisse Bereiche auf dem Chip gibt, auf denen mehr Wirkungstreffer auftraten, als in anderen Gebieten. Bereiche mit vielen Wirkungstreffern verursachen den größten Ausbeuteverlust und sollten einer genaueren Analyse unterzogen werden, zum Beispiel durch die in 6.1.4 und 6.1.6 beschriebenen Ausbeuteanalytoren. Die folgende Abbildung zeigt eine solche Darstellung von Wirkungstreffern:

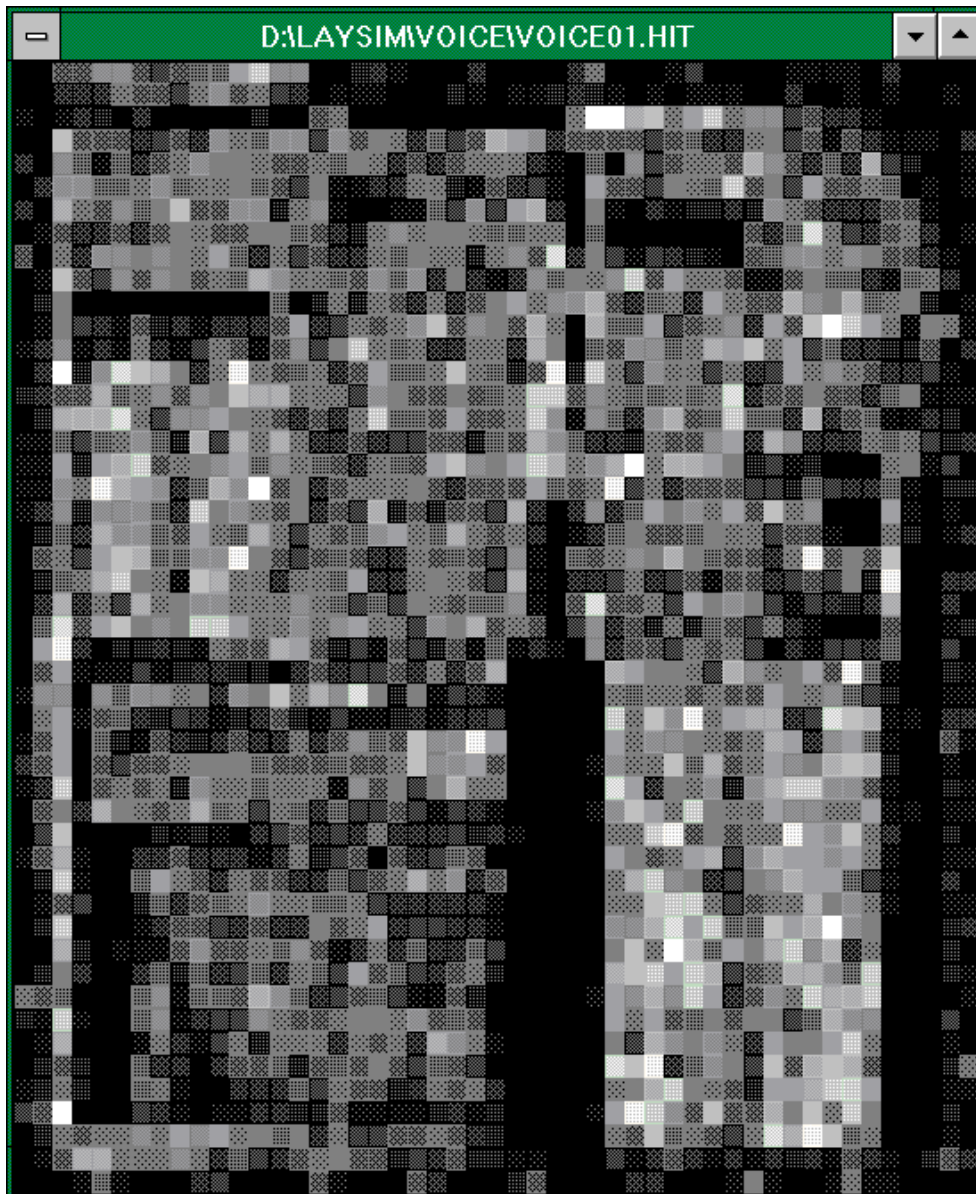


Abbildung 47: Anzeige der örtlichen Häufigkeit der Wirkungstreffer

Es sind deutlich Bereiche zu erkennen, die heller dargestellt sind, als Gebiete in der Nachbarschaft. Bei der hier gewählten Darstellungsweise wurde gedanklich ein Netz über das Layout gelegt und die Anzahl der registrierten Wirkungstreffer in jeder einzelnen Masche in eine entsprechende Graufärbung umgerechnet. Viele Treffer bewirken eine hellere Färbung, wenige eine dunklere. Der Anwender kann nun recht einfach einzelne Funktionsblöcke des Chips erkennen und für die weitere Auswertung visuell oder numerisch, durch Angabe der Koordinaten, auswählen.

6.2 Produktzuverlässigkeit

6.2.1 Bewertung von Zuverlässigkeitsdefekten

In der vom Defektsimulator I erzeugten Defektdatei finden sich neben Defekten, die die Ausbeute eines Produkt verringern, auch Defekte, die dessen Zuverlässigkeit beeinträchtigen. Es sind dies nach den Ausführungen des Abschnitts 5.1 Defekte im Leitbahnkomplex, die nicht groß genug sind, um zwei benachbarte Leitbahnen miteinander zu verbinden bzw. eine Leitbahn vollständig zu durchtrennen (vgl. Abbildung 32).

In das Auswertemodul für diese Klasse von Defekten wurden die Modellvorstellungen, die in den Abschnitten 5.4 und 5.5 dargestellt wurden, integriert. Der Anwender gibt zur Zuverlässigkeitsanalyse die gewünschte Defektdatei an und stellt die folgenden Parameter ein:

- Angaben zur Defektgrößenverteilung und zur Defektdichte tatsächlich vorhandener ausbeuterelevanter Defekte
- Stromdichteexponent der Blackschen Gleichung (5.13)
- mittlere Betriebstemperatur des Schaltkreises
- eine Leitbahnbreite mit zugehöriger mittlerer Stromdichte

Aus diesen Angaben und der vom Simulationsprogramm durchgeführten Analyse kann nach den Abschnitten 5.4 und 5.5 eine Wahrscheinlichkeitsverteilung der Ausfallzeiten für das jeweilige Produkt angegeben werden. Weitere Einzelheiten dazu werden in den Abschnitten 6.2.3 und 7.5 erläutert.

6.2.2 Layoutbezogene Schwachstellenanalyse

Nicht nur Zuverlässigkeitsdefekte können die Lebensdauer eines Schaltkreises verkürzen, manchmal finden sich auch im Layout selbst Stellen, die Leitbahneinengungen darstellen oder sich durch eine besonders hohe Stromdichtedivergenz von den übrigen Figuren abheben. Man spricht bei dieser Art der Zuverlässigkeitsbegrenzung auch von der sogenannten "build-in reliability", also einer Art "eingebauter Zuverlässigkeit" (vgl. [84], [94]). Es wurde deshalb ein weiteres Modul entwickelt, das ein Layout auf das Vorkommen "eingebauter" Schwachstellen analysiert. Der sogenannte "Edge-Finder" geht im Fall einer gefundenen Leitbahneinengung nach dem gleichen Prinzip wie im Abschnitt 5.3 vor und schätzt die Lebensdauer der jeweiligen Leitbahn ab.

Dabei arbeitet der Edge-Finder in zwei Modi: Zum einen tastet er jedes Polygon einzeln ab, ohne sich berührende Figuren miteinander zu verschmelzen, im zweiten Modus wird eine vom Benutzer vorzugebende Anzahl von Verschmelzungen durchgeführt. Dieser Vorgang ist auch als sogenanntes "Merging" bekannt und erfordert für große Layouts selbst auf leistungsfähigen Workstations Rechenzeiten von mehreren Stunden oder Tagen. Um den Rechenaufwand einzugrenzen, wurde hier das Merging auf Stichproben begrenzt. Dabei wird in einem Stichprobenbereich wählbarer Ausdehnung versucht, alle sich in diesem Segment befindlichen Figuren zu verschmelzen. Die Genauigkeit der durchgeführten Tests hängt somit von der Stichprobenanzahl und der Ausdehnung des Stichprobenbereiches ab. Um schnell zu einem ersten Ergebnis zu gelangen, sollte man den Edge-Finder zunächst im Einzelscan-Modus betreiben und anschließend die Funktion zur Polygonverschmelzung des zweiten Betriebsmodus verwenden.

Zusätzlich lassen sich mit diesem Modul Stellen des Layouts mit besonders hoher Stromdichtedivergenz finden und markieren. Somit können Änderungen am Layout vorgenommen werden, um die Zuverlässigkeit bezüglich der Stromdichte und -divergenz zu erhöhen. Auf die weitere Auswertung der im Layout erkannten Schwachstellen wird in den Abschnitten 6.2.3 und 7.5 eingegangen.

6.2.3 Auswertung von Zuverlässigkeitsdefekten und Schwachstellen

Sowohl die vom Defektsimulator I erzeugten Zuverlässigkeitsdefekte als auch die von vornherein im Layout vorhandenen Schwachstellen (Leitbahneinengungen) werden hinsichtlich ihrer Lebensdauer mittels der bereits erwähnten Module ausgewertet. Dabei wird das Verhältnis aus der Lebensdauer einer Schadstelle zu der einer unbeschädigten Leitbahn bestimmt. So lassen sich im nächsten Schritt diejenigen Bereiche im Layout markieren, die im Vergleich zu anderen Gebieten eine besonders kurze Lebensdauer erwarten lassen. An späterer Stelle (Abschnitt 7.5) werden ein Beispiel- und ein Produktlayout auf das Vorhandensein solcher Gebiete untersucht.

Es ist möglich, aufgrund der unterschiedlichen Lebensdauern der erkannten Schwachstellen eine Wahrscheinlichkeitsverteilung der Ausfallzeiten ähnlich der folgenden Grafik zu bestimmen. Um den mehrere Größenordnungen umfassenden Bereich der berechneten relativen Lebensdauern besser darstellen zu können, wurde die x-Achse logarithmisch geteilt.

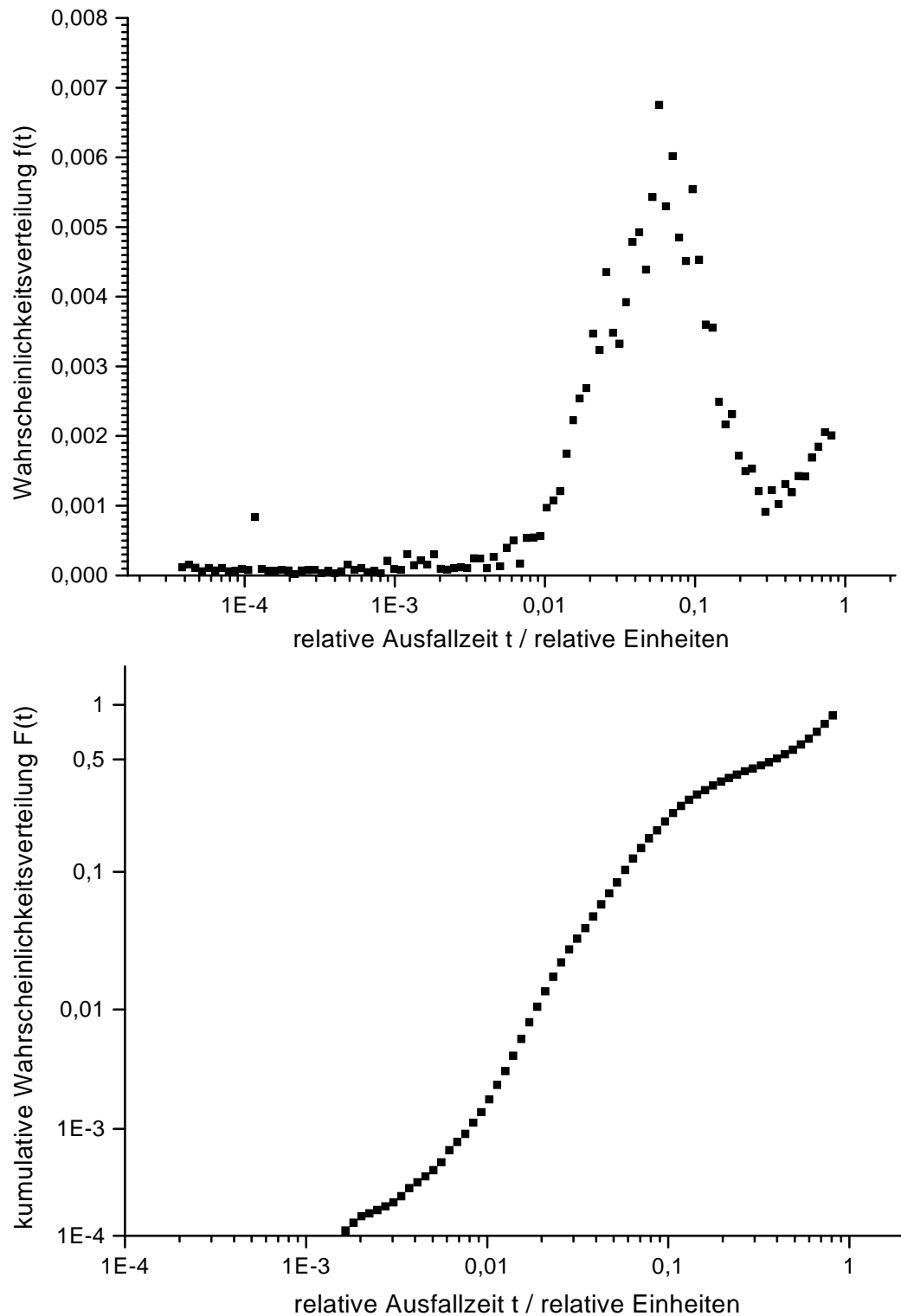


Abbildung 48: oben: Wahrscheinlichkeitsverteilung der berechneten relativen Lebensdauern nach der Schwachstellenanalyse; unten: kumulative Wahrscheinlichkeitsverteilung der Lebensdauern unter Benutzung der Lognormalverteilung

In der Literatur wird häufig die Annahme vertreten, daß logarithmierte Lebensdauern normalverteilt sind ([45], [83], [94]). Die Wahrscheinlichkeitsdichtefunktion $f(t)$ der Ausfallzeiten ist demnach durch folgende Gleichung gegeben:

$$f(t) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp\left[-\frac{1}{2}\left(\frac{\ln(t-t_{50})}{\sigma}\right)^2\right] \quad (6.2)$$

σ *Standardabweichung*

t *Zeit*

t_{50} *mittlere Ausfallzeit*

$$F(t) = \int_0^t f(u) du \quad (6.3)$$

$F(t)$ *kumulative Wahrscheinlichkeits-
verteilung der Ausfallzeit*

Berechnet man die kumulative Wahrscheinlichkeitsverteilung $F(t)$ nach Gleichung (6.3), dann erhält man für die Ausfallzeiten eine Form der Darstellung, wie sie in Abbildung 48 im unteren Teil gezeigt ist. Die x-Achse wurde logarithmisch geteilt, während die y-Achse einer sogenannten Wahrscheinlichkeitsskala entspricht (vgl. [123]). Der Vorteil dieser Darstellung ist die Möglichkeit, eine Gerade an die Meßdaten anzupassen, mit der eine genaue Angabe sowohl der mittleren Ausfallzeit als auch einer Ausfallzeit, die einem beliebigen vorgegebenen kumulativen Ausfall entspricht, möglich ist. Die mittlere Ausfallzeit, auch Median Time To Failure (MTTF) genannt, entspricht der Zeit, zu der die Hälfte aller Proben (hier die gefährdeten Leitbahnen) ausgefallen sind.

Da es keinen physikalischen Grund für die Anwendung der logarithmischen Normalverteilung gibt, wurden weitere Verteilungsfunktionen gesucht, die eine Ausfallzeitverteilung beschreiben können. Die bekanntesten Modelle sind die Weibullverteilung und die multilognormale Verteilung, die hier nur genannt werden sollen. Eine ausführliche Beschreibung dieser und weiterer Verteilungsfunktionen findet man zum Beispiel in [122]. Je nach benutztem Modell ändert sich lediglich die Formel der Wahrscheinlichkeitsdichtefunktion $f(t)$ und die jeweilige grafische Darstellung. Die Bedeutung der MTTF bleibt dagegen erhalten.