Test und Verlässlichkeit Foliensatz 6: Hardware-Test und Selbsttest.

Prof. G. Kemnitz

6. Oktober 2025

Inhaltsverzeichnis

6.2 Auf diesem Foliensatz

Die aus Testsicht kompliziertesten und anspruchsvollsten Hardware-Systeme, aber auch die, mit denen es die meisten Erfahrungen gibt, sind hochintegrierte digitale Schaltkreise. Der Foliensatz behandelt hierzu:

- tatsächliche Fehler, Fehlermodelle und Erfahrungen, was sich bewährt hat und was nicht.
- Fehlersimulation, Testberechnung, prüfgerechter Entwurf,
- Lösungen für den Selbsttest.

Andere Bereiche, insbesondere Software, können von den mit Hardware gesammelten Erfahrung lernen.

Weitere Themen:

- Baugruppentest,
- Ausfallverhalten, Wartungstests und Redundanzen.

6.3 Wiederholung einiger Begriffe

Dynamische Tests sind in der Regel nur für eine winzige Stichprobe der möglichen Eingaben durchführbar. Auswahl der Testeingaben:

- gezielt für eine Menge unterstellter Modellfehler,
- zufällig ohne Rücksicht auf die interne Struktur und zu erwartende Fehler oder
- Mischformen, z.B. operationprofilorientiert.

Wiederholung einiger Begriffe (Abschn. 2.1.3):

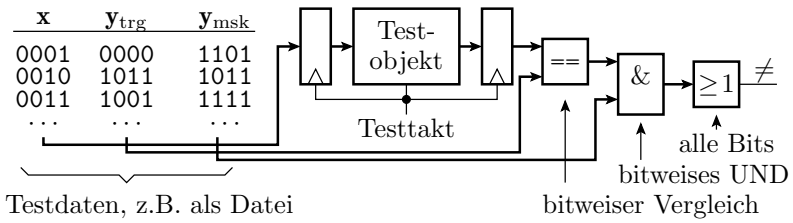
Fehlermodell: Algorithmus, der aus einer simulier- oder abarbeitbaren Beschreibung eine Modellfehlermenge berechnet.

Modellfehler: geringfügige Verhaltens- oder Beschreibungsänderung.

Haftfehler: Annahme von ständig eins oder ständig null für jeden Gatteranschluss.

Eingabeprofil: Relative Nutzungshäufigkeit der unterschiedlicher Äqivalenzklassen von Systemfunktionen (Abschn. 2.3.8).

6.4 Testausführung für DIC (Abschn. 5.2.2)



Wiederhole üblicherweise für sehr viele Testeingaben:

- zeitgleiche Übergabe der Eingabebits in ein Eingaberegister,
- Ausgabeabtastung mit der nächsten Taktflanke,
- Vergleich mit Sollwerten, opt. Bitausmaskierung.

Probleme: begrenzte Anzahl von Schaltkreisanschlüssen, viele Ein- und Ausgabesignale, hohe Geschwindigkeit, große Datenmengen, ...

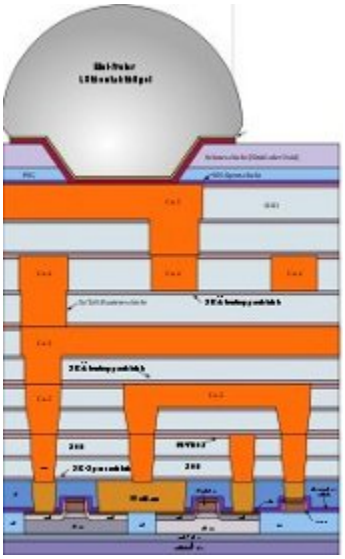
---

x, y Testeingaben, Testausgaben.
y_{msk} Maskenwerte zum Ausschluss von Testausgaben vom Soll-Ist-Vergleich.
DIC Integierter digitaler Schaltkreis (Integrated digital circuit).

1 DIC-Fehler

1.1 Fertigungsfehler

6.5 Entstehungsprozess, zu erwartende Fehler



Schaltkreise entstehen schichtenweise:

- Auftragen von Schichten (z.B. Fotolack oder Metall),
- Belichten des Fotolacks durch eine Maske, die die Geometrie der zu erzeugenden Schichtelemente festlegt,
- Entfernen der belichteten (unbelichteten) Bereiche des Fotolacks,
- Fortätzen der freiliegenden Schichten neben dem Fotolack und entfernen des Fotolacks.

Zu erwartende Fehler aus der Fertigung:

- fehlendes (zu wenig aufgetragenes zu viel weggeätztes Material),
- überflüssiges (zu viel aufgetragen, zu wenige weggeätzt Material)
- Maskenversatz, Prozesssteuerfehler, ...

DIC Integierter digitaler Schaltkreis (Integrated digital circuit).

6.6 Globale Fehler und Parameterfehler

Globale Fehler:

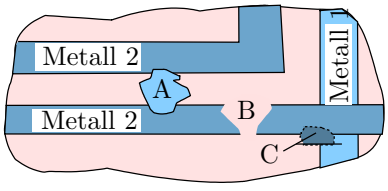
- Fehlerhafte Schichteigenschaften durch Prozesssteuerfehler. Betroffen sind alle Strukturelemente derselben Schicht (Halbleiter-, Leitungs- oder Isolationsschicht).
- Großflächig überflüssiges oder fehlendes Material. Mehrfachkurzschlüsse oder Unterbrechungen.

Meist erhebliche Funktionsbeeinträchtigung und problemlos zu finden:

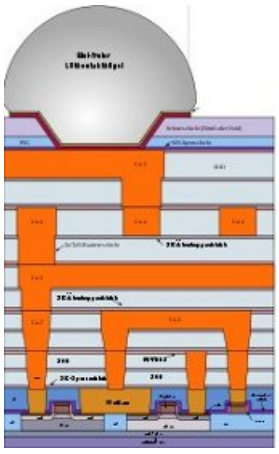
- Parametertest (statische Tests): Stichprobenkontrolle der Transistoreigenschaften, Leitwerte und Kapazitäten an den Schaltkreisanschlüssen und speziellen Teststrukturen, auch schon nach Fertigungszwischenschritten.
- Grobtest: Spannung anlegen und kleine Funktionsstichprobe ausprobieren.

Parameter- und Grobtest finden ca. je 1/3 der Schaltkreisfehler und stellen kaum Anforderungen an die Fehlermodellierung und Testsuche.

6.7 Lokale Fehler



A zuätzliches Metall
B fehlendes Metall
C fehlende Isolation



Einzelfehler durch fehlendes und überflüssiges Material:

- kurzgeschlossene und unterbrochene Verbindungen,
- nicht richtig ein- oder ausschaltende Transistoren

bilden sich ab auf ...

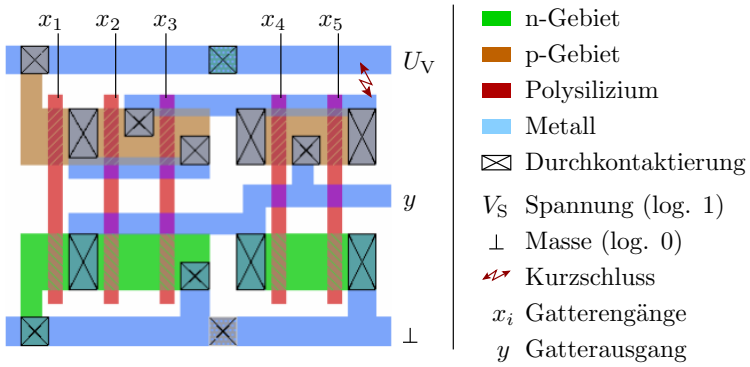
... eine große Menge möglicher unterschiedlicher elektrischer Fehlerwirkungen mit

- z.T. sehr geringer Fehlfunktionsrate und Nachweiswahrscheinlichkeit,
- die z.T. logisch nur über verursachte Zusatzverzögerung oder eine erhöhte Ruhestromaufnahme nachweisbar sind.

Fehlermodellierung durch eine für die Testobjektgröße angemessene Anzahl von Modellfehlern,

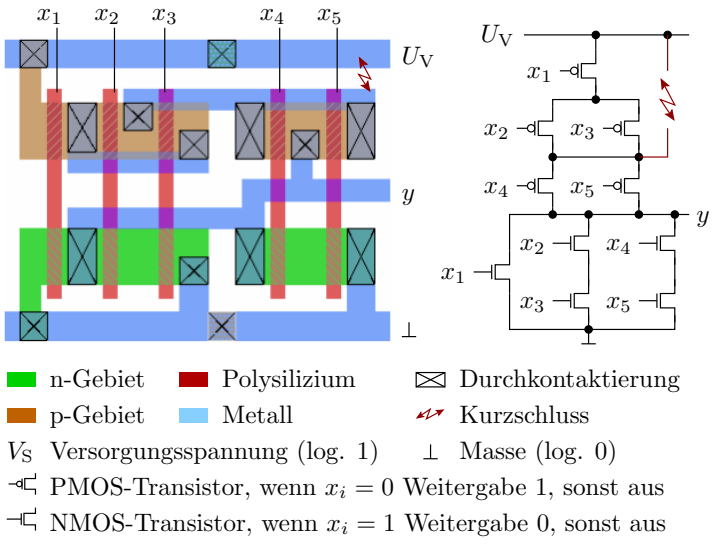
- die sich mit den tatsächlichen Fehlern Anregungs- und Beobachtungsbedingungen teilen und
- möglichst unabhängig von einander nachweisbar sein sollten (Abschn. 4.3.2).

6.9 Geometrie und Funktion

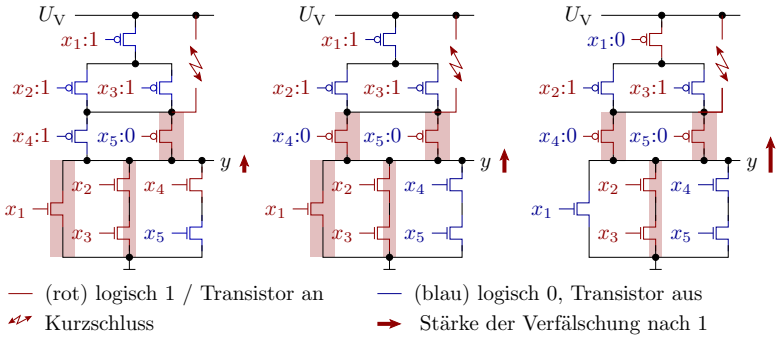


In dem Layoutausschnitt des Logikgatters im Bild ist ein Kurzschluss zwischen zwei benachbarten Leiterbahnen der Metallebene durch nicht weggeätztes Metall unterstellt. Unter welchen Bedingungen verfälscht der Fehler kontrollierbare Ausgaben?

6.10 Extraktion der Schaltung

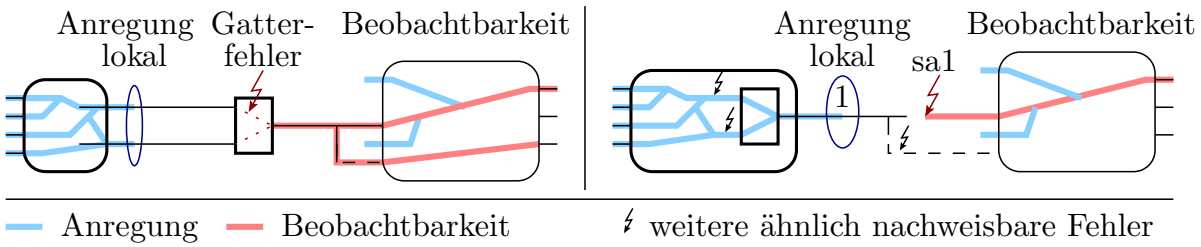


6.11 Lokale Fehlernachweisbedingungen



Für den Nachweis muss der Schaltzweig parallel zum Kurzschluss aus und der in Reihe eingeschaltet sein. Die Stärke der Ausgabeverfälschung hängt vom den Leitwerten des Kurzschlusses und der eingeschalteten Transistoren ab. Die Anregung $x_5x_4x_2x_1 = 00110$ hat die größten Nachweischancen. Statisch nicht »ausreichende« Verfälschungen verzögern den Schaltvorgang und erhöhen den Ruhestrom.

6.12 Globale Nachweisbedingungen



Fehlernachweis verlangt außer einer Gattereingabe aus der Fehlernachweismenge (lokale Nachweisbedingung), globale

- Anregungsbedingungen, Schaltungseingaben, die die Gattereingaben steuern und
- Beobachtungsbedingungen, Schaltungseingaben, die vom Gatterausgang einen Beobachtungspfad sensibilisieren.

Die Fehlermodelle, die sich in der Praxis bewährt haben, generieren für jeden zu erwartendenen Fehler mehrere Modellfehler mit denselben globalen Beobachtungsbedingungen und ähnlichen lokalen und globalen Anregungsbedingungen.

1.2 Praxistaugliche Fehlermodelle

6.13 Bewährte DIC-Fehlermodelle

- Haftfehler,
- Gatterverzögerungsfehler und
- Gatterverzögerungsfehler mit Zusatznachweisbedingungen.

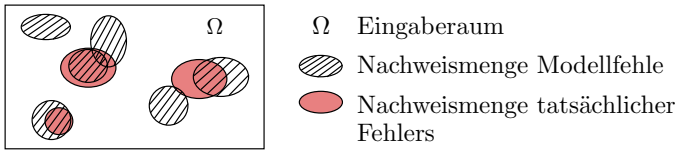
Identisch nachweisbare und redundante Modellfehler beseitigen, ... Die Fehlermodelleigenung hängt auch von der Art der Testauswahl ab.

Interessante Ergänzung für CMOS-Schaltkreise: IDDQ-Test, zusätzliche Überwachung der Ruhestromaufnahme nach jedem Testschritt.

Zur Untermauerung der späteren Bewertung der aktuell als ausreichend geltenden Vollständigkeitskriterien für Softwaretests (Abschn. 7.4.1) werden in einem Folgeabschnitt weitere Testvollständigkeitsmaße für digitale Schaltungen beschrieben, die vor Jahrzehnten genutzt oder als vielversprechend diskutiert wurden und sich nicht bewährt haben.

DIC Integierter digitaler Schaltkreis (Integrated digital circuit).
(F)CMOS Schaltkreise in (vollständig) komplementärer MOS-Schaltungstechnik.
MOS Feldeffektransitoren mit isoliertem Steuer-Gate (MOS – metall oxid semiconductor).
IDDQ Test von CMOS-Schaltkreisen mit Rohestrom-Überwachung.

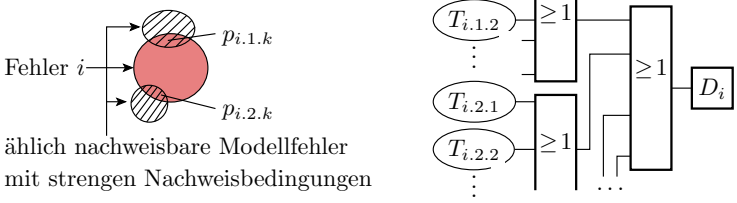
6.14 Nachweisbeziehung Zufallstests



Bei (etwa) gleicher Verteilung der Fehlfunktionsrate gleiche zu erwartende Fehlfunktionsabdeckung für modellierte und tatsächliche Fehler. Eine im Mittel etwas größere oder kleinere Fehlfunktionsrate lässt mit der Testlängenskalierung C herausrechnen (vergl. Abschn. 3.2.3).

Unser Kurzschlussbeispiel hat gezeigt, dass die möglichen Fehlerwirkungen sehr vielfältig sind und sich schwer exakt modellieren lassen. Die ähnlich nachweisbaren Modellfehler je tatsächlicher Fehler haben als gemeinsame Nachweisbedingung »Beobachtbarkeit Gatterausgang« und eine Schnittmenge der lokalen Nachweismengen. Die lokalen Modellfehlernachweismengen sind, wie noch gezeigt wird, je nach Fehlermodell tendentiell größer, gleich oder kleiner als die der Fehler.

6.15 Gezielte Testsuche



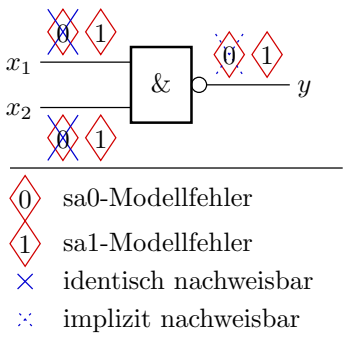
Ein tatsächlicher Fehler i wird nachgewiesen (Ereignis D_i), wenn ein gefundener Test für ein ähnlich nachweisbaren Modellfehler j Fehler i nachweist (Ereignis T_{ijk}, Wahrscheinlichkeit p_{ijk}). Eine hohe Erkennungswahrscheinlichkeit $p_{\mathrm{D}.i}$ verlangt vom Fehlermodell

- ähnlich nachweisbare Modellfehler, deren Nachweis mit
- hohe Wahrscheinlichkeiten p_{ijk} den Fehlernachweis impliziert.

Tendentiell nimmt p_{ijk} mit der Strenge der Nachweisbedingungen des Fehlermodells zu. Die Nachweisbedingungen dürfen aber nur so streng sein, dass der Erfolg der Testsuche nicht beeinträchtigt wird.

p_{ijk} Wahrscheinlichkeit, dass Test k für Modellfehler j den tatsächlichen Fehler i nachweist.

6.16 Gatterhaftfehler (Folie 2.15)



x_2	x_1	$\overline{x_2 \wedge x_1}$	sa0(x_1)	sa1(x_1)	sa0(x_2)	sa1(x_2)	sa0(y)	sa1(y)
0	0	1	1	1	1	1	0	1
0	1	1	1	1	1	0	0	1
1	0	1	1	0	1	1	0	1
1	1	0	1	0	1	0	0	1

Nachweisidentität (gleiche Nachweismenge)

Nachweisimplikation
zugehörige Eingabe ist Element der Nachweismenge

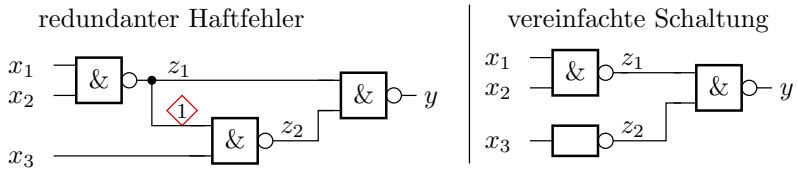
Für jeden Gatteranschluss wird unterstellt:

- ein sa0 (stuck-at-0) Fehler
- ein sa1 (stuck-at-1) Fehler

Nachbearbeitung der Anfangsfehlermenge (bei jedem Fehlermodell):

- Zusammenfassung identisch nachweisbarer Fehler.
- Weglassen von redundanten und optional auch von implizit nachweisbaren Modellfehlern.

6.17 Redundante Haftfehler (Folie 2.18)



Bei $z_1 = 0$ ist der Fehler an y nicht beobachtbar und mit $z_1 = 1$ wird der Fehler nicht angeregt. Nicht nachweisbare Modellfehler sind keine Fehler, aber der Nachweis der »Nichtnachweisbarkeit« ist schwieriger, als für nachweisbare Fehler Tests zu finden.

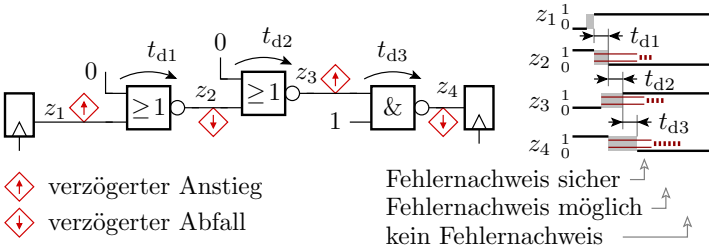
Redundante nicht als solche erkannte Modellfehler verringern die berechnete Modellfehlerabdeckung und verfälschen darüber die Schätzergebnisse für die tatsächliche Fehlerabdeckung.

Fehlermodelle, die einen hohen Anteil redundanter Fehler generieren, sind sind nicht praxistauglich.

---

x_i, y_i, z_i Eingabe-, Ausgabe- und interne Signale.

6.18 Gatterverzögerungsfehler



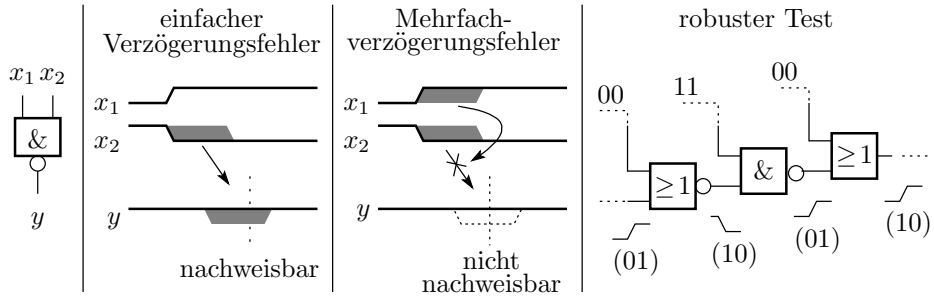
Annahme an allen Gatteranschlüssen

- Slow-To-Raise (verzögerter Signalanstieg) und
- Slow-To-Fall (verzögerter Signalabfall).

Änderung des logischen Werts am Fehlerort als strengere Nachweisbedingung gegenüber Haftfehlern. Längere Zufallstests für gleiche Modellfehlerabdeckung (Nachteil). Vergrößert die p_{ijk} für Kurzschlüsse und andere zu erwartende tatsächliche Fehler für die gezielte Testsuche (Vorteil).

$t_{d.i}$ Gatterverzögerungszeit.

6.19 Robuster Nachweis

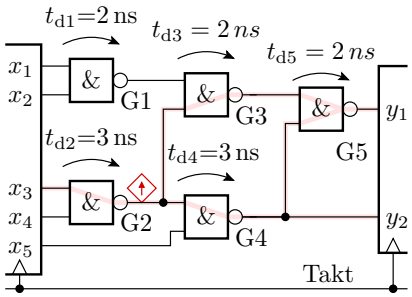


- Es sind Mehrfachfehler konstruierbar, die sich bei bestimmten Tests gegenseitig maskieren.
- Robuster Nachweis: Ausschluss der gegenseitigen Maskierung durch max. eine Eingabesignaländerung je Gatter und Testschritt.

Strengere Nachweisbedingung als »Gatterverzögerungsfehler«:

- für Zufallstest Nachteil,
- für gezielte Testsuche Vorteil.

6.20 Minimal erkennbare Zusatzverzögerung



Pfade	$\sum t_{d.i}$
G1-G3-G5	6 ns
G2-G3-G5	7 ns
G2-G4-G5	8 ns
G2-G4	6 ns
G4-G5	5 ns
G4	3 ns

- Die minimal erkennbare Zusatzverzögerung ist die Differenz aus Taktperiode und Soll-Verzögerung.
- Je länger die Sollverzögerung, desto höher die Wahrscheinlichkeit, fehlerverursachte Zusatzverzögerungen zu erkennen.

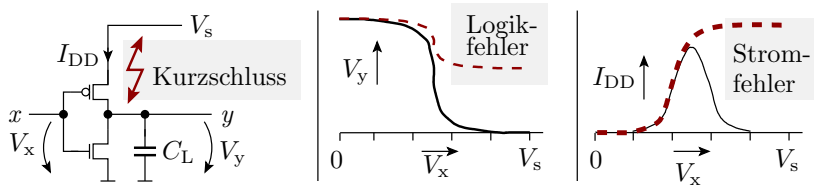
Test mit $f_{\mathrm{Clk.max}}$ auch für Zufallstest gut. Test über die Pfade mit der max. Sollverzögerung nur für gezielte Testauswahl vorteilhaft.

$t_{d.i}$ Gatterverzögerungszeit.

x_i, y_i Schaltungeingabesignale, Schaltungsausgabesignale.

1.3 IDDQ-Test

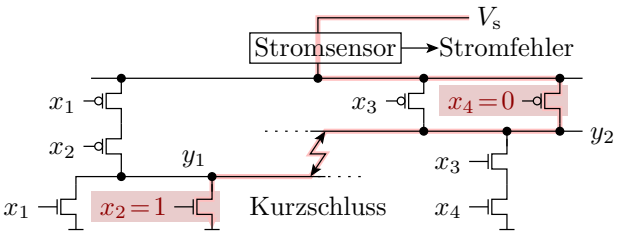
6.21 Ruhestromfehler



In einer CMOS-Schaltung ist der Gatterausgang nur entweder über NMOS-Transistoren mit 0 (Masse) oder über PMOS-Transistoren mit 1 (Versorgungsspannung V_s) verbunden. Nach jedem Schaltvorgang klingt der Strom auf einen sehr kleinen Wert ab. Kurzschlüsse, nicht richtig ausschaltende Transistoren, ... verursachen, wenn sie Gatterausgaben verfälschen, ein messbar erhöhten Ruhestrom I_{DDQ}.

I_{DDQ} Versorgungsruhestrom (supply current).
V_x, V_y Spannung (Potential) von Signal x bzw. y.
V_s, $\perp$ Versorgungsspannung (log. 1), Masse (log. 0).

6.22 Nachweiseigenschaften



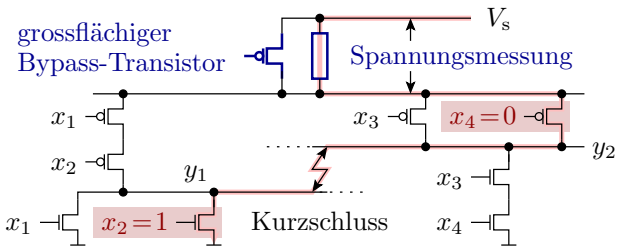
Die Überwachung des Ruhestroms vereinfacht die Nachweisbedingungen für viele FCMOS-typische Fehler erheblich. Nachweisbededingung Kurzschluss im Bild mit IDDQ-Überwachung: $y_1 \neq y_2$.

Allgemein genügt die lokale Anregung ohne sensibilisierten Beobachtungspfad zu einem Ausgang:

- Zufallstests: erheblich höhere effektive Testsatzlänge,
- gezielte Testsuche: viel einfachere Testberechnung.

x_i, y_i Eingabe- und Ausgabesignale.

6.23 Ruhestrommessung schwierig

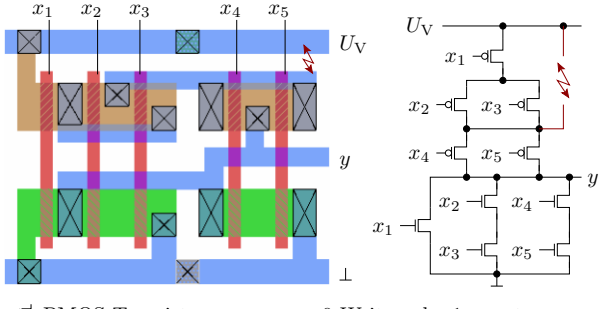


Zum erhöhten Ruhestrom des defekten Gatters addieren sich die Ruheströme aller anderen Gatter. Sicheres Erkennen ohne Phantomfehler nur bis zu einigen tausend Gattern. Für größere Schaltungen sind die Stromsensoren der Gattergruppen in die Schaltkreise zu integrieren. Elekrische Probleme, Chip-Fläche für Bypass-Transistoren, ...

Trotzt des hohen Fehlernachweispotentials Einsatz nur für Produkte, die sehr zuverlässig sein müssen und / oder für die geringer Stromverbrauch sehr wichtig ist (Batteriebetrieb).

1.4 Untersuchung einiger Beispielfehler

6.24 Beispielkurzschluss



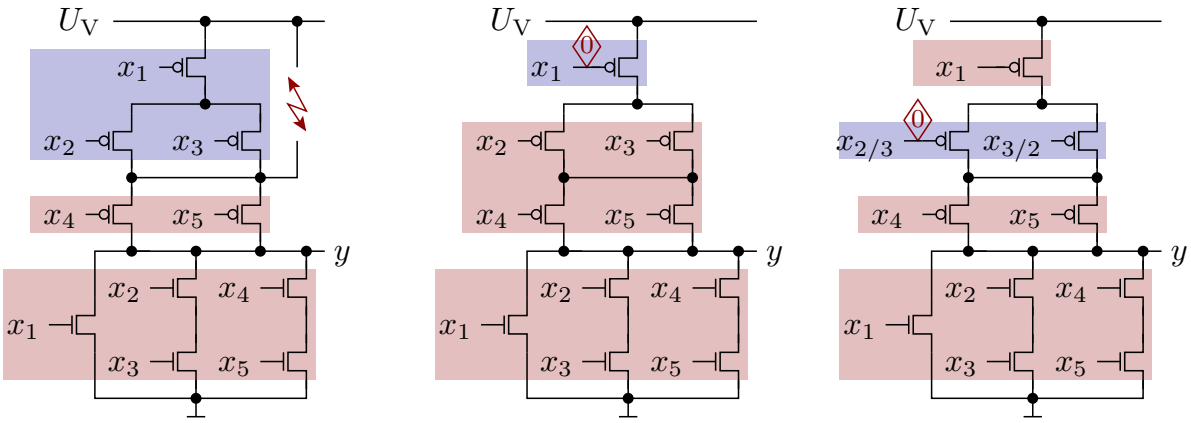
PMOS-Transistor, wenn $x_i = 0$ Weitergabe 1, sonst aus
NMOS-Transistor, wenn $x_i = 1$ Weitergabe 0, sonst aus

Lokale Nachweisvoraussetzungen:

- PMOS: Netzwerk für x_1 bis x_3 aus, Paralleschaltung x_4 und x_5 an.
- NMOS-Netzwerk: insgesamt an

Ähnlich nachweisbare Haftfehler: sa0(x_1), sa0(x_2), sa0(x_3).

V_s, $\perp$ Versorgungsspannung (log. 1), Masse (log. 0).



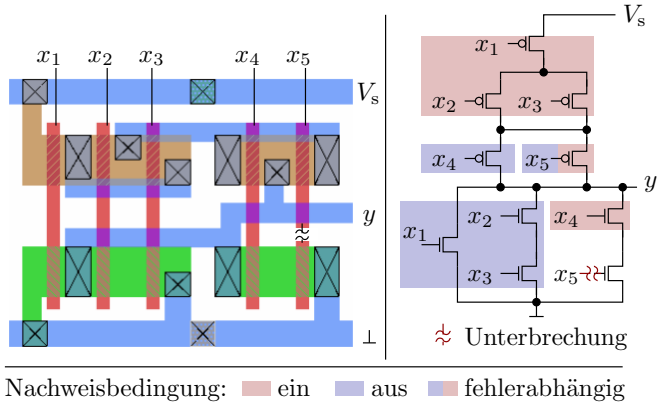
Für $x_5 = x_4 = 1$ sind weder Kurzschluss noch einer der betrachteten Hafttfehler nachweisbar

x_1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
x_2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
x_4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
x_5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
Kurzschl.	x		x		x	x	x		x		x	x	x		x		x	x	x		x	x	x		x	x	x	
sa0(x_1)	x		x		x		x		x		x		x		x		x	x	x		x	x	x		x			
sa0(x_2)						x						x												x				
sa0(x_3)						x						x												x				

Kurzschluss
sa0-Fehler
insgesamt leitend
insgesamt gesperrt

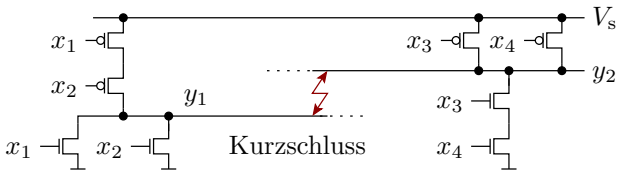
Drei (davon zwei identisch nachweisbare) Haftfehler mit kleineren Nachweismengen. Zufallstest und Haftfehler $C \approx 0{,}5 \ldots 1$. Geziele Suche und Gatterverzögerungsfehler $p_{ijk} < 0{,}5$, für Haftfehler kleiner.

6.26 Offenes Gate



Der Beispielfehler ist ein abgetrennter Gate-Anschluss. Das isolierte Gate kann auf null oder eins aufgeladen sein und seinen Wert nur sehr langsam ändern. Gleiche lokale Anregungsmenge wie sa0 oder sa1 bzw. slow-to-rise oder slow-to-fall an x_5. Zufallstest und Haftfehler: $C \approx 1$. Geziele Suche mindestens ein Gatterverzögerungsfehler mit $p_{ijk} = 1$.

6.27 Kurzschluss zweier Gatterausgänge



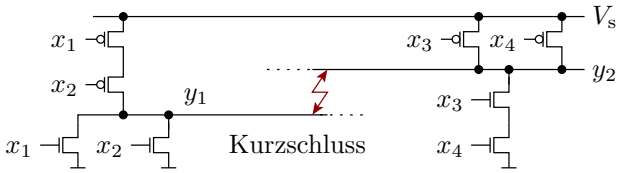
Mögliche Nachweisbedingungen:

1. $\bar{x}_1 \wedge \bar{x}_2 = 1$ und $x_3 \wedge x_4 = 1$ ($y_{1.\mathrm{trg}} = 1$ und $y_{2.\mathrm{trg}} = 0$)
2. $x_1 \vee x_2 = 1$ und $\bar{x}_3 \vee \bar{x}_4 = 1$ ($y_{1.\mathrm{trg}} = 0$ und $y_{2\mathrm{trg}} = 1$)

Ob sich dabei $y_1 = y_2 = 0$ oder $y_1 = y_2 = 1$ durchsetzt, hängt von den Transistorbreiten bzw. Transistorsteilheiten ab. Der verfälschte y-Wert muss zusätzlich beobachtbar sein.

---

x_i, y_i Gattereingangssignale, Gatterausgangssignale.
V_s, $\perp$ Versorgungsspannung (log. 1), Masse (log. 0).



Ähnlich nachweisbare Haftfehler:

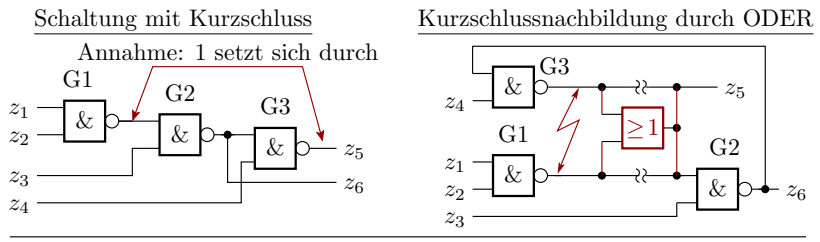
- sa0 (x_1), sa0 (x_2) wenn $y_2 = 1$ ist und sich durchsetzt
- sa1 (x_1), sa1 (x_2) wenn $y_2 = 0$ ist und sich durchsetzt
- sa0 (x_3), sa0 (x_4) wenn $y_1 = 1$ ist und sich durchsetzt
- sa1 (x_3), sa3 (x_4) wenn $y_1 = 0$ ist und sich durchsetzt

Zufallstest und Haftfehler: Nachweiswahrscheinlichkeit grob: Wahrscheinlichkeit inverser Pegel auf der anderen Leitung mal Summe der Nachweiswahrscheinlichkeiten von zwei Leitungshaftfehlern, $C \lesssim 1$. Gezielte Auswahl und Haftfehler: p_{ijk} tendentiell 50%.

---

x_i, y_i Gattereingangssignale, Gatterausgangssignale.
V_s, $\perp$ Versorgungsspannung (log. 1), Masse (log. 0).
C Fehlermodellspezifische Skalierung der effektiven Testanzahl.

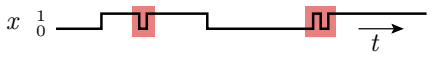
6.29 Speicherverhalten durch Kurzschluss



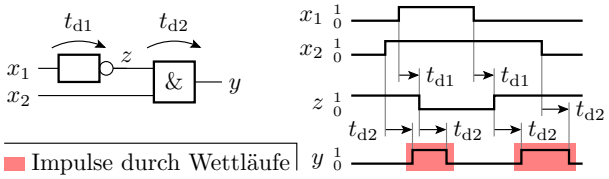
$z_4\ z_3\ z_2\ z_1$	$z_6\ z_5$
0 1 1 1	0 1 (setzen)
1 0 1 1	1 0 (löschen)
1 1 1 1	speichern

$z_4\ z_3\ z_2\ z_1$	$z_6\ z_5$
0 0 1 1	1 1
⇓⇓	⇓⇓ ändern nach
1 1 1 1	unbestimmt

6.30 RS-Flipflops, Glitches und Wettläufe



Glitches sind kurze Pulse, die bei logischen Verknüpfungen von Signalen mit unterschiedlicher Verzögerung entstehen, die unser RS-Flipflop für $x_4x_3x_2x_1 = 0011$ setzen oder rücksetzen können.



Ursache Wettläufe und rekonvergente Signalflüsse. Wettläufe sind fast zeitgleiche Änderungen an mehreren Eingängen einer vorgelagerten Teilschaltung. Auch wenn vor und nach der Änderung derselbe Wert ausgegeben wird, treten im Änderungsmoment kurze Pulse auf.

6.31 Fehlerwirkung vs. Modellierung

Die wenigen Beispiele haben gezeigt, dass die Wirkungen einzelnen Kurzschlüsse und Unterbrechungen sehr vielfältig und kaum exakt zu modellieren ist, weil auch von Laufzeiten, Leitwerten und der Fehlerausprägung abhängig. Nicht betrachtet aber auch möglich sind Abhängigkeiten von der Versorgungsspannung, der Temperatur, ...

Je exakter ein Fehlermodell die möglichen tatsächlichen Fehlerwirkungen annähert, desto mehr sehr ähnlich nachweisbare Modellfehler. Zumindest für Zufallstest nicht zielführend. Bei gezielter Testsuche beeinträchtigen zu komplizierte Nachweisbedingungen den Sucherfolg.

Etabliert haben sich statt dessen die am einfachsten zu modellierenden lokalen logischen Fehlerannahmen:

- Haftfehler und
- Haftfehler mit Zusatzbedingungen.

Für einen Zufallstest deuten die Beispiele auf fehlermodellspezifische Skalierung der effektiven Testanzahl von mindesten $C \leq 1$.

- Ohne redundante Modellfehler ist die zu erwartende Fehlerabdeckung nicht schlechter als die Modellfehlerabdeckung.
- Eine ausreichende Testanzahl für Modellfehler ist auch ausreichend für tatsächliche Fehler.

Für gezielte Testauswahl sind die Haftfehler-Zusatzbedingungen zur Erhöhung von p_{ijk} nutzbar:

- Signalwechsel am Fehlerort (Gatterverzögerungsfehler),
- robuster Nachweis und
- Nachweis über den Pfad mit der längsten Soll-Verzögerung.

Alle drei Zusatzbedingungen haben den Vorteil, dass sie sich auf die Eingaben vor dem Haftfehlernachweis beziehen, und damit die Testsuche für den Haftfehlernachweis nicht erschweren.

Auch bei strengem Fehlermodell Suche mehrerer zufällig ausgewählter Tests aus der Nachweismenge von jedem Modellfehler zweckmäßig.

6.33 Modellfehler- und Defektabdeckung DIC

In der Beispielrechnung 3.2 wurde abgeschätzt, dass Schaltkreistests bei den typischen Schaltkreisausbeuten von $Y \approx 30\% \ldots 90\%$ Defektabdeckungen von $DC \approx 99{,}9\%$ haben müssten, um auf die in der Literatur zu findenden Defektanteile von $DL = 200\,\text{dpm}$ bis $1.000\,\text{dpm}$ zu kommen.

Schaltkreistests habe Haftfehlerabdeckungen in der Größenordnung $FC_\text{M} = 95\% \ldots 99\%$. Der Anteil der nicht nachweisbaren defekten Schaltkreise müsste mindestens eine Zehnerpotenz kleiner als der Anteil der nicht nachweisbaren Haftfehler sein.

> Ist das glaubwürdig?

Y Ausbeute (Yield).
DL Defektanteil nach Aussortieren oder Ersatz erkannter defekter Produkte.
DC Defektabdeckung (defect coverage), Anteil der erkennbar defekten Produkte.
FC_M Modellfehlerabdeckung.

- Typ. 2/3 der Schaltkreisdefekte sind globale Fehler, die von den Parameter- und Grobtests zu 100% nachgewiesen werden. Verringerung der Lücke bis um Faktor 3.
- Die zu erwartende Defektabdeckung ist die Wahrscheinlichkeit, das mindestens ein Fehler erkannt wird. Bei Ausbeuten $Y \approx 30\% \ldots 90\%$ sind Mehrfachfehler typisch. Clusterung erhöht die den Anteil der Schaltkreise mit Mehrfachfehlern. Mehrfachfehler sind einfacher zu erkennen als Einzelhaftfehler.
- Auch Kurzschlüsse Kurzschlüsse und Unterbrechungen tendentiell dazu, einfacher als Haftfehler nachweisbar zu sein.
- Die Angaben zur Haftfehlerabdeckung werden durch nicht erkannte redundante Haftfehler verfälscht.

Ein Anteil der nicht nachweisbaren Defekte von weniger als einem Zehntel des Anteils der nicht nachweisbaren Modellfehler ist denkbar, aber eine höher Dunkelziffer des Defektanteils aufgelieferter Schaltkreise ist auch möglich. Forschungsbedarf!

1.5 Veraltetet Testvollständigkeitsmaße

6.35 Veraltetet Testvollständigkeitsmaße

Nicht alle Fehlermodelle und Testvollständigkeitsmaß, die vor Jahrzehnten für digitale Schaltungen als vielversprechend diskutiert wurden, haben sich gehalten.

Software hat gegenüber Hardware die Vorzüge; einfache Änderbarkeit und große funktionale Redundanz. Wenn die Anwender Probleme bemerken, stellten sie ein Change-Request oder suchen ein Workaround. Das hat die Forschung zum Testen gegenüber Hardware um Jahrzehnte zurückgestellt.

Die heute für den Softwaretests als ausreichend geltenen Testvollständigkeitsmaße (Abschn. 7.4.1):

- alle Anweisungen ausprobieren (Anweisungsabdeckung),
- alle Zweige ausprobieren (Zweigabdeckung), ...

gabe es vor Jahrzehnten in ähnlicher Form für Hardware.

Vor diesem Hintergrund lohnt ein Blick, was sich für digitale Schaltungen alles nicht bewährt hat und in Vergessenheit geraten ist.

6.36 Toggle-Test

Vor der Zeit der fehlermodellbasierten Testauswahl und -bewertung diente die Toggle-Abdeckung als Testvollständigkeitsmaß:

```
für alle Leitungen

    0 einstellbar,
    1 einstellbar.
```

Unberücksichtigte Nachweisbedingungen:

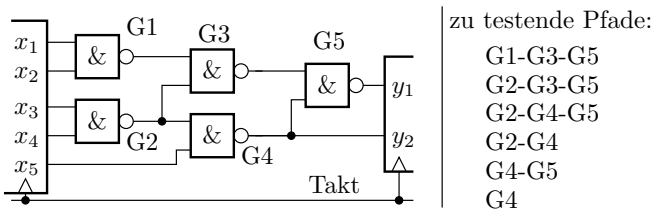
- Beobachtbarkeit lokaler Verfälschungen,
- paarweise unterschiedlicher Werte für Kurzschlussnachweis, ...

Eine 0 oder 1 am Fehlerort impliziert nur mit geringer Wahrscheinlichkeit einen Fehlernachweis.

- Zufallstests müssen für denselben Abdeckungswert um einen Skalierungsfaktor $C \gg 1$ länger als der Toogle-Test sein und
- bei gezielter Suche ist jedes Kriterium $w \gg 1$ mal abzudecken.

Toggle-Tests haben keine praktische Bedeutung mehr und über die Größenordnung von C und w gibt es keine belastbaren Untersuchungen.

6.37 Pfadverzögerungsfehler



Fehlerannahme »Slow-To-Rise« / »Slow-To-Fall« für alle Signalpfade. Die Beispielschaltung hat fünf Gatter und 6 Signalpfade. 100% Pfadfehlerabdeckung schließt die Tests über die Pfade mit der längsten Solllaufzeit immer mit ein. Als Fehlermodell ungeeignet weil:

- Im ungünstigen Fall exponentielle Zunahme der Pfadanzahl mit der Gatteranzahl. Große Nachweisähnlichkeiten, großes κ ...
- Nicht über alle Pfade lassen sich Signalwechsel propagieren. Viel redundante Fehler.

6.38 Zellenfehler

Idee: Definition von Bausteintestsätzen, die durch die umgebende Schaltung gesteuert und beobachtbar gemacht werden müssen. Jeder Bausteintest wird als Modellfehler gezählt.

Soll-Funktion			Zellenfehler			
x_1	x_0	$x_1 \vee x_0$	F00	F01	F10	F11
0	0	0	1	0	0	0
0	1	1	1	0	1	1
1	0	1	1	1	0	1
1	1	1	1	1	1	0

Für ein Gatter könnte der lokalen Tests der mit allen Eingaben sein. Im Beispiel $2^2 = 4$ Zellenfehler statt 6 initiale Haftfehler. Als Fehlermodell meist nicht geeignet weil:

- Beim Test der Zellen mit allen Eingabemöglichkeiten exponentielle Zunahme der Modellfehleranzahl mit der Anzahl der Gattereingänge und viele ähnlich nachweisbare Modellfehler.
- Viel der lokalen Tests nicht durch die umgebende Schaltung steuer- und beobachtbar. Viele redundante Fehler.

6.39 C-Test

Bewährt hat sich die Idee des Zellenfehlermodells für die Konstruktion von sog. C-testbaren regelmäßig strukturierten Schaltungen, Speicher, Addierer, Multiplizierer, ...

Dabei werden die Zellenfunktion, der Zellentestsatz und die Schaltungsstruktur so aufeinander abgestimmt entwickelt, dass sich ein Testsatz mit 100% Zellenfehlerabdeckung (ein C-Test) generieren lässt (sie später Abschn. 6.2.7).

Zusammenfassung

6.40 Global und lokale Fehler, Testauswahl

- Schaltkreise entstehen schichtenweise. Globale Fehler sind großflächig, leicht nachweisbar und für die Testauswahl uninteressant.
- Lokale Fehler unterbrechen oder verbinden einzelne Schichtelemente, modellierbar als fehlendes oder zusätzliches Material.
- Kurzschlüsse und Unterbrechungen von Schichtelementen bilden sich auf Kurzschlüsse und Unterbrechungen von Verbindungen und nicht richtig bzw. zu langsam schaltende Transistoren ab.
- Die elektrische und logische lokale Wirkung und Nachweisbarkeit ist vielfältig und oft nicht exakt vorhersagbar.
- Modellfehler teilen sich mit den tatsächlichen lokalen Fehlern Anregungsbedingungen und die Beobachtbarkeit des Gatterausgang.
- Für Zufallstest genügen Modellfehler mit ähnlich großen lokalen Nachweismengen.
- Für gezielte Testauswahl ist es wichtiger, dass Tests für ähnlich nachweisbaren Modellfehler mit großer Wahrscheinlichkeit p_{ijk} tatsächliche Fehler nachweisen.

6.41 Bewährte Fehlermodelle

Für Zufallstest ist das Haftfehlermodell Favorit. Die untersuchten Beispiele deuten auf eine fehlermodellspezifische Skalierung der effektiven Testanzahl $C < 1$.

- Ohne redundante Modellfehler ist die zu erwartende Fehlerabdeckung nicht schlechter als die Haftfehlerabdeckung.
- Eine ausreichende Testanzahl für Haftfehler ist auch ausreichend für tatsächliche Fehler.

Für gezielte Testauswahl sind Zusatzbedingungen zur Verbesserung der p_{ijk} zielführend:

- Signalwechsel am Fehlerort (Gatterverzögerungsfehler),
- robuster Nachweis und
- Nachweis über den Pfad mit der längst Soll-Verzögerung.
- Statt oder in Ergänzung der Zusatzbedingungen, zufällige Auswahl mehrerer Tests aus der Nachweismenge von jedem Modellfehler.

6.42 Sonstiges

IDDQ-Test: Beobachtung von Fehlern am erhöhten Ruhestrom nach Schaltvorgängen. Erhöht die effektive Testanzahl und vereinfacht Testsuche erheblich. IDDQ-Überwachung aber so problematisch, das Einsatz nur bei sehr hohen Anforderungen an die Verlässlichkeit und Batterielaufzeit.

Anteil der nicht nachweisbaren defekten Schaltkreise um einen Faktor zehn kleiner als Anteil der nicht nachweisbaren Modellfehler? Denkbar, aber genauere Untersuchen wünschenswert.

Ein ähnliches Testvollständigkeitsmaß wie heute die Anweisungsabdeckung für Software gab es früher auch für Hardware, den Toggle Test. Heute in Vergessenheit geraten.

2 Testsuche

6.43 Testauswahl, gezielte Suche

Beginn mit zufälliger Auswahl:

Wiederhole, solang Nachweis von genug neuen Fehler je Test
Fehlersimulation mit neuen Zufallseingaben und Inkrement der Nachweisanzahl w_i für alle nachweisbaren Modellfehler i

Dann Umschaltung auf gezielte Testsuche:

Wiederhole für alle Modellfehler i:
Suche w Tests oder Redundanznachweis Abbruch bei Überscheitung einer gesetzten Rechenzeitgrenze

Problematisch sind redundante Fehler, genauer der Nachweis, dass es für einen Modellfehler keine Nachweismöglichkeit gibt und er folglich nicht als Modellfehler zählt.

---

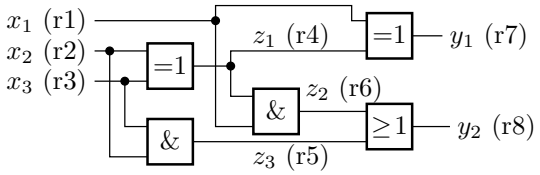
w Nachweisanzahl, Anzahl der je Nachweismenge zufällig auszuwählender Tests.

w_i Anzahl der für Modellfehler i gefundenen Tests.

2.1 Fehlersimulation

6.44 Logiksimulation digitaler Schaltungen

Schaltung eines Volladdierers



r1 bis r8 Prozessorregister

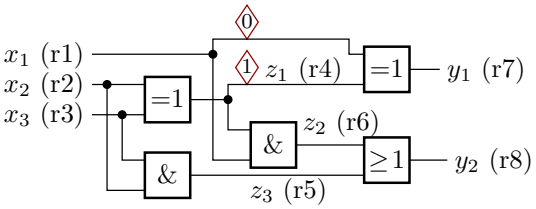
Programm für die Gutsimulation

```
 1 lade x1 in Register r1
 2 lade x2 in Register r2
 3 lade x3 in Register r3
 4 r4 = r2 xor r3
 5 speichere Inhalt r4 in z1
 6 r5 = r2 and r3
 7 speichere Inhalt r5 in z3
 8 r6 = r1 and r4
 9 speichere Inhalt r6 in z2
10 r7 = r1 xor r4
11 speichere Inhalt r7 in y1
12 r8 = r5 or r6
13 speichere Inhalt r8 in y2
```

- Jede zweistellige Logikoperation ist ein Maschinenbefehl.
- In jeder der 32 oder 64 Bits der Operanden kann ein anderer Testfall oder ein anderer Fehler simuliert werden.
- Mit 64-Bit-Prozessor ca. 16 simulierte Gatterfkt. / Prozessortakt.

6.45 Haftfehler und Zusatzbedingungen

Fehlerparallele Haftfehlerimulation




```
 9 ...
   ; sa0(x1) in Bit 0
   r8 = r1 and ...1110
   ; sa1(z1) in Bit 1
   r9 = r1 or ...0010
10 r7 = r8 xor r9
11 speichere Inhalt r7 in y1
12 ...
```

- Haftfehler werden durch setzen bzw. Löschen von Bits durch UND/ODER mit Bitvektorkonstanten simuliert.
- Zusatzbedingungen wie Signalwechsel am Fehlerort, wenn Haftfehler nachweisbar (Gatterverzögerungsfehler), sind Logikbeziehungen zu Bitergebnissen der Simulation für Testschritt davor, modellierbar durch wenige zusätzlichen Lade- und Logikbefehle.

Haftfehler und Zusatzbedingungen sind bitparallel mit wenigen Prozessorbefehlen simulierbar.

6.46 Aufwandsabschätzung am Beispiel

- Schaltungsgröße: 10^4 Gatter
- Anzahl der Testschritte / Testeingaben: 10^4
- Anzahl der Modellfehler: 10^4
- Simulationsaufwand je Gatter: 10 ns

Rechenaufwand:

- Wenn jeder Fehler mit allen Testeingaben simuliert wird und ohne bitparallele Simulation: 10^4 s, ca. 3 h.
- Wenn mit jedem der 32 bzw. 64 Bits ein anderer Fehler simuliert wird, nur 6 bzw. 3 Minuten.
- Wenn bereits ausreichend oft nachgewiesene Modellfehler nicht weiter mit simuliert werden, unter 1 Minute.

Erheblicher Rechenaufwand entfällt auf die nicht erkannten redundanten Modellfehler.

In 10 s Testzeit können bei einer Taktfrequenz von 100 MHz bis zu 10^9 Tests ausgeführt werden. Eine Fehlersimulation von 10^9 Test dauert auch bitparallel auf einem Prozessor viele Tage.

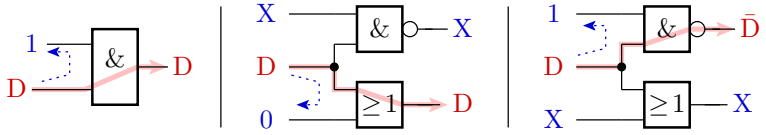
2.2 D-Algorithmus

6.47 D (Discrepancy)-Kalkül von Roth

Erweiterung der Logikwerte um 3 Pseudo-Werte [Deahn97]:

X	Signalwert nicht festgelegt oder beliebig
D	0 wenn unverfälscht, 1 wenn verfälscht.
$\bar{D}$	1 wenn unverfälscht, 0 wenn verfälscht.

Regeln für die Sensibilisierung eines Beobachtungspfades:

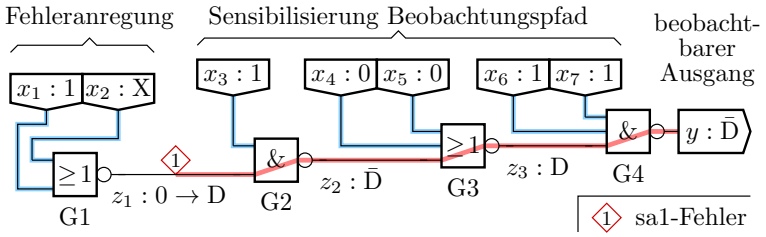


D Discrepanz-Kalkül, 0 wenn unverfälscht, 1 wenn verfälscht.

X Signalwert ungültig oder für den Fehlernachweis ohne Bedeutung.

[Daehn97] Testverfahren in der Mikroelektronik: Methoden und Werkzeuge. Springer 1997.

6.48 Testsuche für Haftfehler



Ein Haftfehler unterstellt für den Fehlerort, dass der Wert

- entweder ständig 0 (sa0) oder
- ständig 1 ist (sa1) ist.

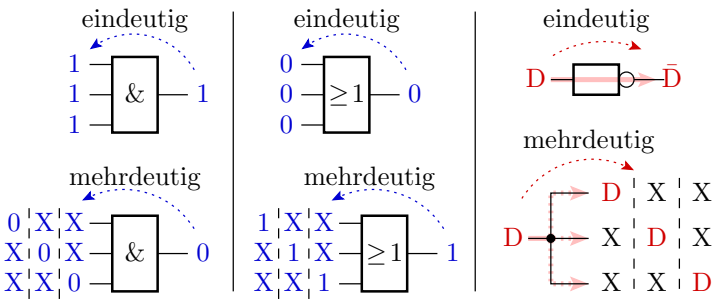
Ausgehend vom Fehlerort werden Eingaben gesucht,

- für die der Wert am Fehlerort invertiert wird und
- bei denen die Invertierung am Fehlerort an einem Ausgang beobachtbar ist.

x_i, y_i, z_i Eingabe-, Ausgabe- und interne Signale.

D Discrepanz-Kalkül, 0 wenn unverfälscht, 1 wenn verfälscht.

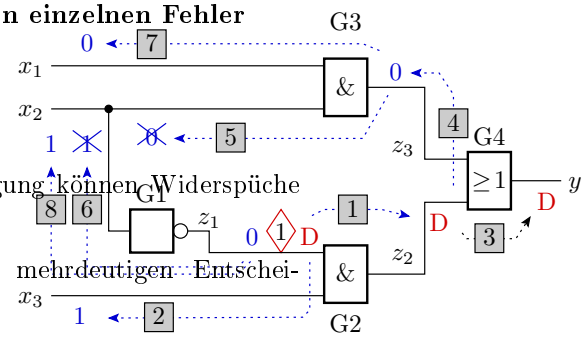
6.49 Ein- und mehrdeutige Pfade



Ausgehend vom Fehlerort:

- Wertefestlegung zur Weiterführung von Beobachtungs- oder Steuerpfaden.
- Für nicht eindeutige Festlegungen sind Entscheidungen zu treffen.
- Bei Widersprüchen Änderung der letzten Entscheidung mit einer noch nicht untersuchten Alternative (Baumsuche).
- Widerspruch und keine Alternative ⇒ Modellfehler redundant.

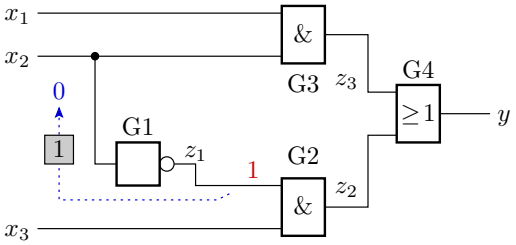
6.50 Testsuche für einen einzelnen Fehler



Baumsuche: nicht

- Bei der Wertefestlegung können Widersprüche auftreten.
- Zurück zur letzten mehrdeutigen Entscheidung.
- Keine Lösung nach Durchmusterung des gesamten Baums. ⇒ Modellfehler redundant (nicht nachweisbar).

6.51 Zusatzbedingungen



Die Zusatzbedingungen »Gatterverzögerungsfehler«, »robuster Nachweis« und »Pfad mit längster Sollverzögerung« beschreiben Logikbedingungen für die Testeingaben davor.

Für den sa1-Fehler im Beispiel ist für den Zeitschritt davor eine eins am Fehlerort einzustellen.

Suche geeigneter Testeingaben dafür in der Regel viel einfacher als für den Haftfehler, weil kein Beobachtungspfad sensibilisiert werden muss.

6.52 6.52 Optimierung Suchalgorithmus

- Der Suchraum wächst exponentiell mit der Anzahl der mehrdeutigen Festlegungen.
- Am aufwändigsten ist der Redundanznachweis für einen Modellfehler, weil dafür der gesamte Suchraum zu durchmustern ist.
- Suchräume der Größen $> 2^{30...40}$ nicht mehr vollständig durchsuchbar. Abbruch der Suche nach einer vorgegebenen Zeitschranke.

Erfolgsrate der Testberechnung:

- Anteil der Fehler, für die ein Test gefunden oder für die der Beweis »nicht nachweisbar« erbracht wird.

Heuristiken zur Verbesseerung der Erfolgsrate:

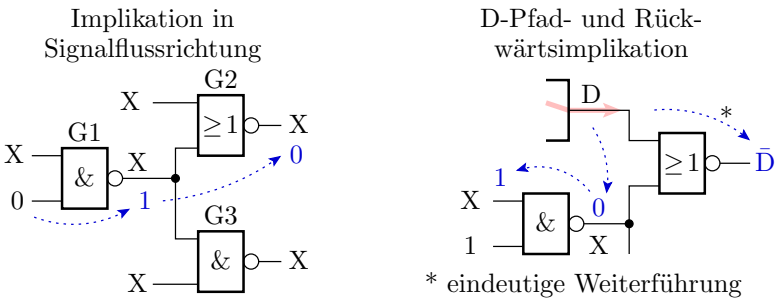
- Frühe Erkennung von Widersprüchen (Äste im Suchbaum abschneiden),
- Suchraumbegrenzung und
- gute Suchraumstrukturierung.

2.3 Implikationstest

6.54 Implikationstest

...zur frühzeitigen Erkennung von Widersprüchen:

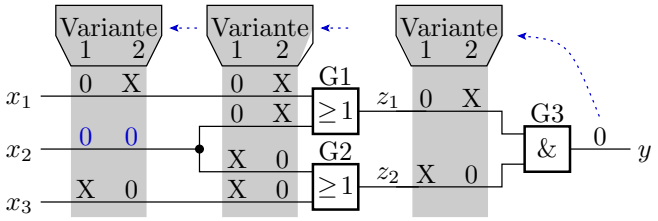
- Nach jeder Wertefestlegung alle eindeutig folgenden Werte festlegen.



- Im Beispiel (Folie 6.50) sind alle Werte eindeutig.

D Discrepanz-Kalkül, 0 wenn unverfälscht, 1 wenn verfälscht.
X Signalwert ungültig oder für den Fehlernachweis ohne Bedeutung.

- Rückwärtsimplikation über mehrere Gatterebenen:



- Für $y = 0$ gibt es zwei Einstellmöglichkeiten.
- Für beide Möglichkeiten muss $x_2 = 0$ sein.
- Das Erkennen von Implikationen dieser Art mindert die Backtracking-Häufigkeit um bis zu 80 %.

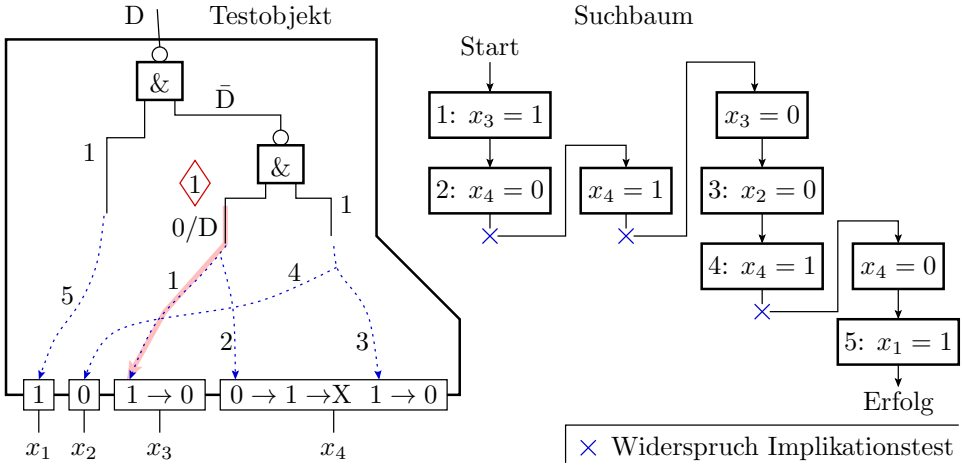
x_i, y_i, z_i Eingabe-, Ausgabe- und interne Signale.
X Signalwert ungültig oder für den Fehlernachweis ohne Bedeutung.

2.4 Suchraumstrukturierung

6.56 Suchraumbegrenzung

- Der D-Algorithmus baut den Suchbaum über alle mehrdeutigen Wertefestlegungen auf.
- Nur die Schaltungseingänge können unabhängig voneinander alle Wertekombinationen annehmen.
- Es genügt, den Suchbaum mit den Eingabewertefestlegungen aufzubauen.
- Begrenzt Suchraum auf $2^{\#x}$. Verringert Rechenaufwand um Zehnerpotenzen.

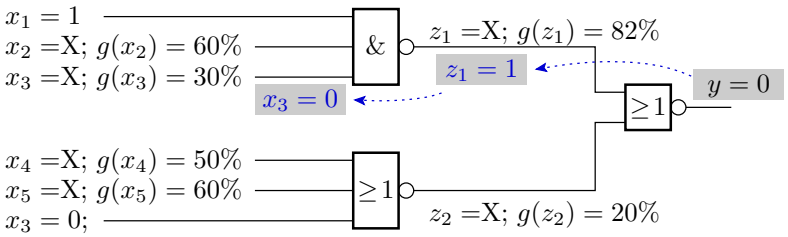
$\#x$ Anzahl der Schaltungseingänge.



- Lange Steuerpfade vom Fehlerort und vom D-Pfad zu Eingängen.
- Aufbau des Suchbaums über Eingangssignale.
- Wenn Implikationstest-Widerspruch, letzte Eingabefestlegung invertieren.

X Signalwert ungültig oder für den Fehlernachweis ohne Bedeutung.
D Discrepanz-Kalkül, 0 wenn unverfälscht, 1 wenn verfälscht.

6.58 Geschätzte Erfolgswahrscheinlichkeiten



- Schätzen der Signalwichtungen $g(x_i)$ über eine kurze Simulation mit Zufallswerten oder analytisch.
- Wahl der Steuerwerte / Beobachtungspfade, die mit größerer Wahrscheinlichkeit aktivierbar / sensibilisierbar sind.

x_i, y_i, z_i Eingabe-, Ausgabe- und interne Signale.
X Signalwert ungültig oder für den Fehlernachweis ohne Bedeutung.
$g(\ldots)$ Wichtung, Auftrittshäufigkeit des Signalwerts 1.

2.5 Komplexe Funktionsbausteine

6.59 Komplexe Funktionsbausteine

- Beschreibung durch Tabellenfunktion (Bsp. Volladdierer):

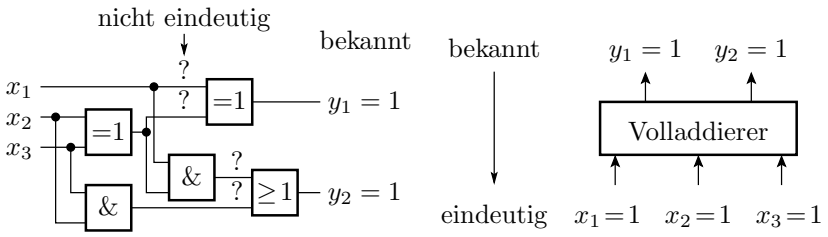
x_2	x_1	x_0	s	c	gegeben	Lösungsmenge
0	0	0	0	0	XXX00	⇒ 00000 (eindeutig)
0	0	1	1	0		
0	1	0	1	0	01DXX	⇒ 01DD̄D (eindeutig)
0	1	1	0	1		
1	0	0	1	0	1XXXD	⇒ 10DD̄D, 1D0D̄D (2 Alternativen)
1	0	1	0	1		
1	1	0	0	1	11XX1	⇒ 11111, 11001 (2 Alternativen)
1	1	1	1	1		

- Vervollständigung des Vektors der gegebenen Anschlusswerte durch Vergleich mit allen Tabellenzeilen:
  - »1« und »0« passen nur auf »1« und »0«.
  - »X« passt immer.
  - »D« muss für eine Zeile mit »D=0« und eine Zeile mit »D=1« passen.

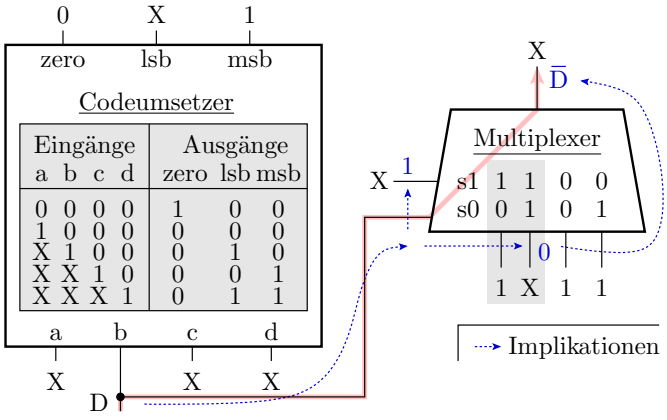
D Discrepanz-Kalkül, 0 wenn unverfälscht, 1 wenn verfälscht.

X Signalwert ungültig oder für den Fehlernachweis ohne Bedeutung.

6.60 Implikationstest an einem Volladdierer



- An der Gatterbeschreibung eines Volladdierers ist die Implikation $y_1 = y_2 = 1 \Rightarrow x_1 = x_2 = x_3 = 1$ nicht zu erkennen. Lösungsfindung über Baumsuche.
- Bei Zusammenfassung zu einer Tabellenfunktion wird die Lösung bereits bei der Anschlusswertevervollständigung erkannt.



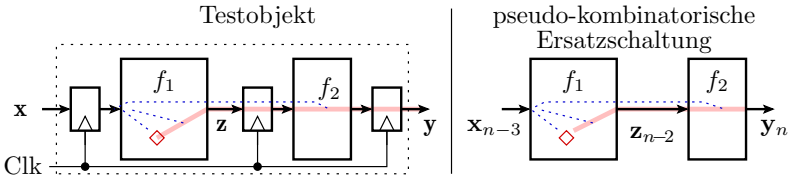
- »lsb« hängt bei »zero=0« und »msb=1« nicht von »b« ab. Nur über den Multiplexer lässt sich der D-Pfad weiterführen, ...
- Tabelleneingabewerte »X« führt zu Tabellen mit $\ll 2^w$ Zeilen.

D Discrepanz-Kalkül, 0 wenn unverfälscht, 1 wenn verfälscht.
X Signalwert ungültig oder für den Fehlernachweis ohne Bedeutung.
w Anzahl der Eingabebits.

2.6 Sequentielle Schaltungen

6.62 Pseudo-kombinatorische Ersatzschaltung

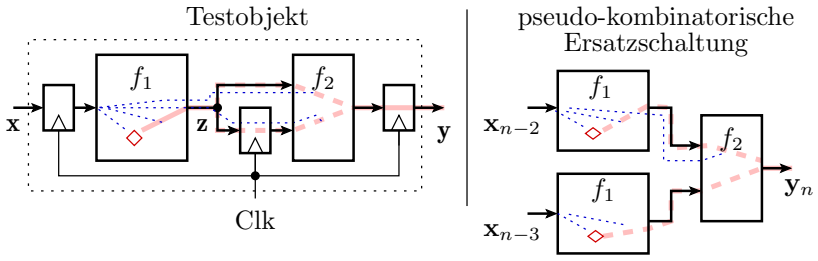
Schaltungen mit Speicherelementen werden für die Testsuche zu einer pseudo-kombinatorischen Ersatzschaltung aufgerollt. Abtastregister in einem geradlinigen Berechnungsfluss werden weggelassen:



- Testberechnung wie für eine kombinatorische Schaltung.
- Die Verzögerungen der Ausgabe gegenüber der Eingabe wird erst beim Zusammensetzen der Testeingaben und Sollausgaben berücksichtigt.

x, y, z Bitvektor der Eingabe-, Ausgabe- und internen Signale, Index ist Zeitschrittnummer.
Clk Taktsignal (Clock signal).

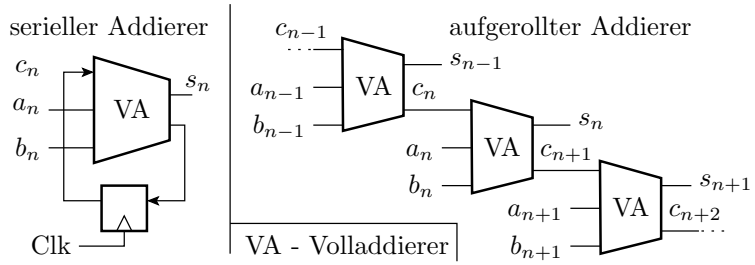
6.63 Verarbeitung in mehreren Zeitebenen



- Mehrere Kopien gleicher Schaltungsteile in der pseudo-kombinatorischen Ersatzschaltung.
- Der eingebaute Haftfehler ist in jeder Kopie der Teilschaltung.
- Für jeden Fehler wird eine Folge von Testeingaben für mehrere Zeitschritte berechnet (Mehr-Pattern-Test).
- Erschwert die Einstellung der Zusatzbedingungen in den Zeitschritten zuvor.

x, y, z Bitvektor der Eingabe-, Ausgabe- und internen Signale, Index ist Zeitschrittnummer.
Clk Taktsignal (Clock signal).

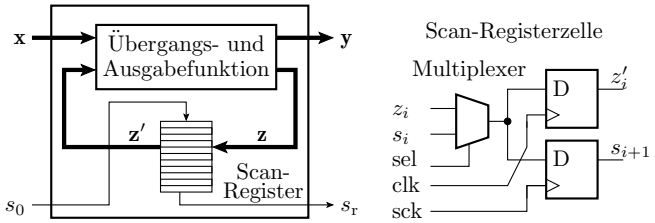
6.64 Schaltungen mit Rückführung



- Pseudo-kombinatorischen Ersatzschaltung mit endlos vielen Kopien der Übergangsfunktion.
- Längenbegrenzung der Steuer- und Beobachtungspfade.
- Alternative: Scan-Zugriff auf Übertragsbit (Folie 5.37).

a_i, b_i	Summandenbit i.
s_i	Summenbit i.
c_i	Übertragsbit i.
Clk	Taktsignal (Clock signal).

6.65 6.65 Scan-Verfahren (Folie 5.37)



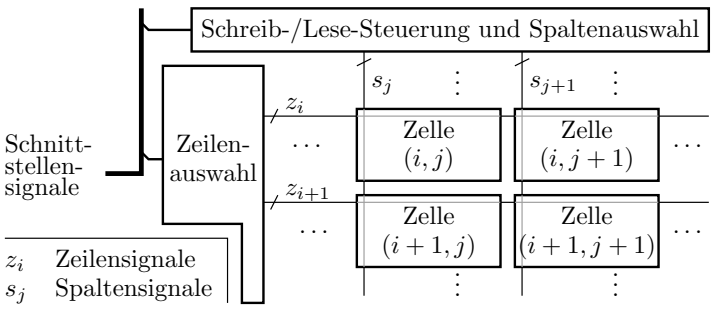
Lese- und Schreibzugriff während des Tests durch Umschalten des Zustandsspeicher in ein r-Bit Schieberegister.

- Alternierend r Schiebeschritte zum Beschreiben und Lesen des Scan-Registers und eine Testschritt.
- Optional getrennte Register für Schieben und Daten für Zusatzbedingung »Gatterverzögerungsfehler.

x, **y**	Bitvektor der Eingabe- und Ausgabesignale.
z, **z**′	Bitvektor der Zustands- und der abgetasteten Zustandssignale.
s_i, sel	Serielles Schiebesignal i, Umschaltsignal zwischen Schiebe- und Normalmodus.
clk, sck	Normales Taktsignal, Schiebetaktsignal (Normal clock signal, shift clock signal).

2.7 Speichertest

6.66 Blockspeicher



Große Speicher besteht im Kern aus einer regelmäßigen 2D-Anordnung von flächenminimierten Speicherzellen umgeben von der Zeilen- und Spaltenauswahlschaltung. Die Grundfunktionen (nur lesbar, beschreib- und lesbar, ...) hängen von der Zellenfunktion ab.

Testauswahl: Festlegen der Test für eine Zelle. Konstruktion eine Gesamttestablaufs, der alle Zellentests enthält (C-Test, Folie 6.39).

6.67 Zellenfehler für RAM

beteiligte Zellen	Name	Definition	Fälle	Testfolge für den Nachweis
1	Haftfehler	Wert der Speicherzelle ist nicht setzbar	stuck-at-0 stuck-at-1	$W(i)1$, $R(i)1$ $W(i)0$, $R(i)0$
	Übergangsfehler	Wert der Speicherzelle i ist nur in einer Richtung änderbar	kein Übergang $1 \rightarrow 0$ $0 \rightarrow 1$	 $W(i)1$, $R(i)1$, $W(i)0$, $R(i)0$ $W(i)0$, $R(i)0$, $W(i)1$, $R(i)1$
	Stuck-open-Fehler	kein Zugriff auf Speicherzelle i (Ausgabe des Wertes der vorherigen Leseoperation)		$W(i)0$, $R_1(j)$, $R(i)0$, $W(i)1$, $R_0(j)$, $R(i)1$
	zerstörendes Lesen	Inhalt von Speicherzelle i wird beim Lesen verändert	$R(i) \Rightarrow C(i) = \overline{C(i)}$	$W(i)0$, $R(i)0$, $R(i)0$ $W(i)1$, $R(i)1$, $R(i)1$
2	Kopplung Typ 1	Veränderung des Inhalts von Zelle i bestimmt Zustand in Zelle j	$W(i)0 \Rightarrow C(j) = 0$ $W(i)0 \Rightarrow C(j) = 1$ $W(i)1 \Rightarrow C(j) = 0$ $W(i)1 \Rightarrow C(j) = 1$	$W(j)0$, $W(i)0$, $R(j)0$, $W(i)1$, $R(j)0$ $W(j)1$, $W(i)0$, $R(j)1$, $W(i)1$, $R(j)1$
	Kopplung Typ 2	Veränderung des Inhalts von Zelle i bewirkt eine Änderung in Zelle j	$C(i) = \overline{C(i)} \Rightarrow C(j) = \overline{C(j)}$	$W(j)0$, $W(i)0$, $R(j)0$, $W(i)1$, $R(j)0$, $W(i)0$, $R(j)0$ $W(j)1$, $W(i)0$, $R(j)1$, $W(i)1$, $R(j)1$, $W(i)0$, $R(j)1$

$W(i)0$ Schreibe in Zelle i eine 0
$W(i)1$ Schreibe in Zelle i eine 1
$R(j)$ Lese eine beliebige andere Zelle
$C(\ldots)$ Inhalt Zelle ...
$R(i)0$ Lese Inhalt Zelle i und vergleiche mit Sollwert 0
$R(i)1$ Lese Inhalt Zelle i und vergleiche mit Sollwert 1
$R_0(j)$ Lese eine andere Zelle, in der 0 steht
$R_1(j)$ Lese eine andere Zelle, in der 1 steht

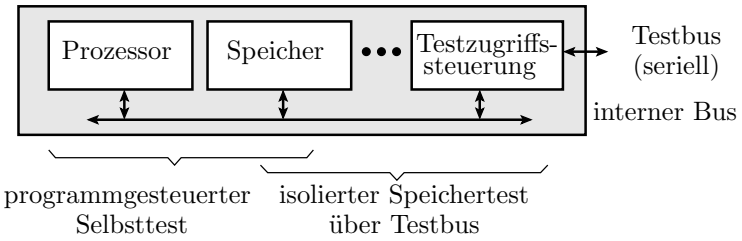
6.68 Marching Test

Adresse i	Initialisierung	March 1	March 2	March 3
0	$W(i)0$	$R(i)0$, $W(i)1$	$R(i)1$, $W(i)0$	$R(i)0$, $W(i)1$
1	$W(i)0$	$R(i)0$, $W(i)1$	$R(i)1$, $W(i)0$	$R(i)0$, $W(i)1$
2	$W(i)0$	$R(i)0$, $W(i)1$	$R(i)1$, $W(i)0$	$R(i)0$, $W(i)1$
⋮	⋮	⋮	⋮	⋮
$N-1$	$W(i)0$	$R(i)0$, $W(i)1$	$R(i)1$, $W(i)0$	$R(i)0$, $W(i)1$

	March 4		March 1a		March 2a
0	$R(i)1$, $W(i)0$	Wartezeit	$R(i)0$, $W(i)1$	Wartezeit	$R(i)1$
1	$R(i)1$, $W(i)0$		$R(i)0$, $W(i)1$		$R(i)1$
2	$R(i)1$, $W(i)0$		$R(i)0$, $W(i)1$		$R(i)1$
⋮	⋮		⋮		⋮
$N-1$	$R(i)1$, $W(i)0$		$R(i)0$, $W(i)1$		$R(i)1$

- Mehrfaches Durchwandern des Speichers mit der Operationsfolge Zelle Lesen, Wert kontrollieren, inversen Wert zurückschreiben.
- Umkehrung der Adressierungsreihenfolge, um Überkopplungen in beiden Richtungen nachzuweisen.
- Wartezeit für den Nachweis von Datenhalteproblemen.

6.69 Test eingebetteter Blockspeicher



Eingebettete Blockspeicher werden vorzugsweise isoliert von ihrer Schaltungsumgebung getestet:

- über herausgezogene Bussignale,
- über den Testbus oder
- programmgesteuert vom Prozessor als eingebauter Selbsttest.

Zusammenfassung

6.70 Gezielte Testsuche

- Gezielte Testsuche beginnt in der Regel mit einer Fehlersimulation mit zufälligen Eingaben und Abhaken der nachweisbaren Fehler bzw. Zählen der Nachweisanzahl.
- Zur Fehlersimulation werden die logischen Berechnung in Maschinenbefehlsfolgen übersetzt und bitparallel ausgeführt. Die Simulation größerer Funktionsbausteine mit tausenden von Fehlern und millionen von Tests kann dennoch Tage dauern.
- Für die geziele Suche haben wir den D-Algoritmus kennen gelernt. Er treibt vom Fehlerort Beobachtungspfade zu Augängen und Steuerpfade zu Eingängen.
- Mit den Mehrdeutigkeiten bei der Pfadsensibilisierung wird ein Suchbaum aufgebaut und bei Wiedersprüchen immer zum letzten Ast mit noch nicht durchmusterten Alternativen zurückgekehrt.
- Widerspruch und keine weitere Alternative weist einen Modellfehler als redundant aus, d.h. Redundanznachweis aufwändiger, als existierende Tests zu finden.

6.71 Heuristiken zur Erhöhung des Sucherfolgs

- Frühe Widerspruchserkennung durch Implikationstests.
- Suchraumbegrenzung auf unabhängig steuerbare Eingänge.
- Bevorzugung erfolgsversprechenderer Entscheidungen z.B. anhand geschätzter Wahrscheinlichkeiten für die Steuer- und Beobachtbarkeit.
- Beibehaltung komplexer Teilfunktionen als Wertetabellen statt Nachbildung durch Gatter.

6.72 Sequentielle Schaltungen

- Sequentielle Schaltungen werden für die Testsuche zu einer pseudo-kombinatorischen Ersatzschaltung aufgerollt.
- Solange gespeicherte Werte nicht auf Eingänge der kombinatorischen Verarbeitung rückgeführt werden, Testberechung wie für Schaltungen ohne Speicherzellen.
- Rückführung erschweren die Testsuche und mindern die Erfolgsrate, Tests zu finden.
- Problemvermeidung durch Schreib-/Lesezugriff auf interne Speicherzellen über Scan-Register.

6.73 Speicher

- Für Speicher und andere regelmäßig strukturierte Schaltungen gibt es sog. C-Tests. Das sind parametrierte auf unterschiedliche Größe und Konfigurationen anpassbare Testsatzalgorithmen, die für jede Zelle alle Modellfehler nachweisen.

- Der Marching-Algorithmus für RAM durchläuft beispielsweise alle Adressen vor- und rückwärts und führt dabei bestimmte Zugriffsoperationen aus.

3 Selbsttest

6.74 Selbsttest

Selbsttestfunktionen für digitale Schaltungen werden realisiert durch Erweiterung der Testobjekte um:

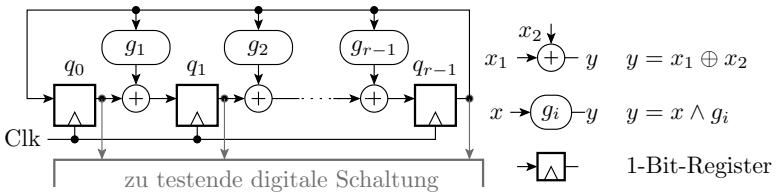
- Pseudo-Zufallsgeneratoren an den Eingängen, vorzugsweise LFSR (Linear Feedback Shift Register), Zähler, ...
- Signaturegister oder andere Überwachungsfunktionen an den Ausgängen und
- eine Steuerung für den Testablauf:
  - Initialisierung Testmustergenerator, Prüfkennzeichen, Testobjekt
  - Wiederhole für alle Testschritte
    - Testmustergenerator weiterschalten,
    - Bildung der Schaltungsausgaben aus den Testeingaben,
    - Weiterschalten Signaturregister oder andere Kontrolle.
- Steuerung und Abfrage Prüfkennzeichen und anderer Kontrollergebnisse über seriellen Testbus (Folie 5.40).

Vorteile gegenüber externer Prüftechnik:

- sehr viele Tests mit voller Systemgeschwindigkeit,
- auch in der Einsatzphase als Einschalt- oder Wartungstest, ...

3.1 Pseudo-Zufallsregister

6.75 Linear rückgekoppelte Schieberegister

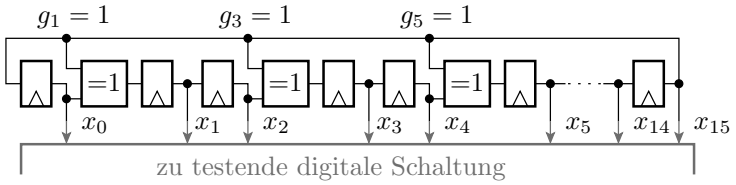


In einer ersten Ausführung verschiebt ein linear rückgekoppeltes Schieberegister (LFSR **L**inear **F**eeback **S**hift **R**egister) seinen r-Bit-Zustand $\mathbf{s} = (s_{r-1}, s_{r-2}, \ldots, s_0)$ um eine Stelle nach links und addiert, wenn das herausgeschobene Bit s_{r-1} gleich »1« ist, eine Bitvektorkonstante $\mathbf{g} = (g_{r-1}, g_{r-1}, \ldots, g_1, 1)$ zum Zustand **s.**

Konstanteneliminition entfernt in der Schaltung die EXOR mit $g_i = 0$.

Für jede Bitanzahl r des Zustandsvektors gibt es Konstantenvektoren $\mathbf{g}$, sog. »primitive Polynome«, bei denen alle Zustände außer 000...0 in einander übergehen (Folie 5.69 ff.). Diese werden bevorzugt.

6.76 Primitive Polynome und die Konstante g



Mit dem Internet-Suchbegriff »Primitive Polynome« findet man z.B. für 16-Bit LFSR:

$$z^{16} \oplus z^5 \oplus z^3 \oplus z \oplus 1$$

Das bedeutet $g_1 = g_3 = g_5 = 1$ und alle anderen $g_i|_{i \notin \{1,3,5\}} = 0$. In Realisierung als Digitalschaltung für $g_i = 1$ EXOR-Gatter einfügen und für $g_i = 0$ EXOR-Gatter weglassen. g_0 und g_r sind immer 1.

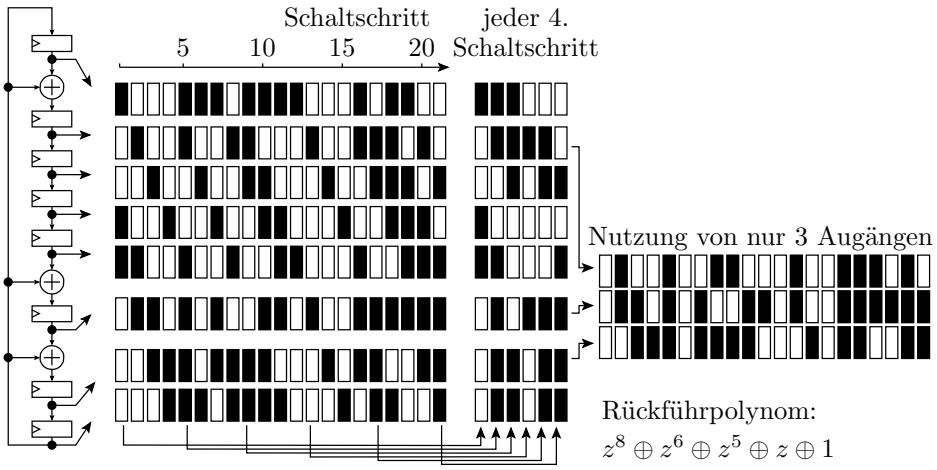
---

z^i Operator für eine Verschiebung um i Bitstellen.

g_i, q_i Registerbits und Rückführkoeffizienten des LFSR.

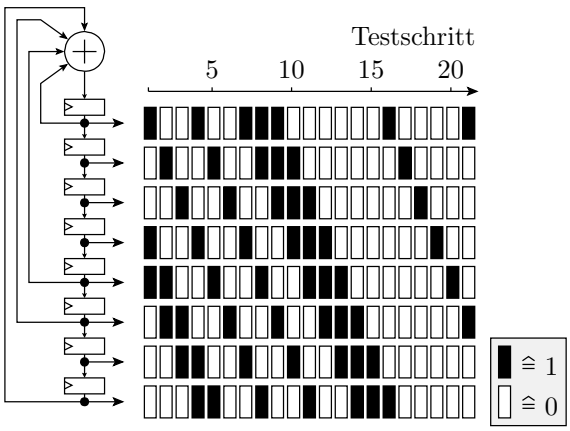
r Bitanzahl des linear rückgekoppelten Schieberegisters.

6.77 Pseudo-Zufallsfolge eines 8-Bit-LFSR



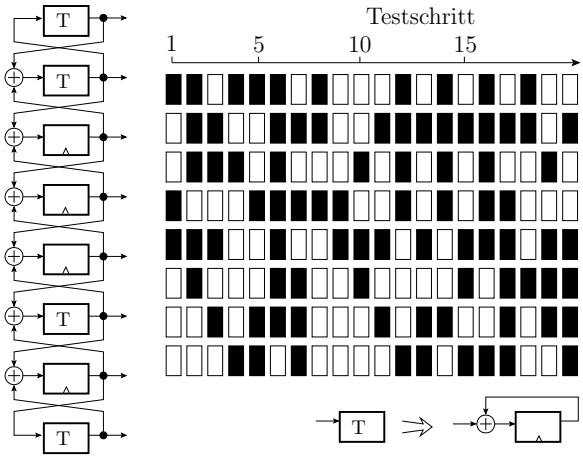
Falls die »Streifenmuster« durch die Schiebeoperationen stören, z.B. für Verzögerungsfehlernachweis, nur einen Teil der Ausgaben nutzen.

6.78 LFSR mit zentraler Rückführung



Statt Ausgang auf mehrere Bits, Rückkopplung mehrere Bits auf den Eingang. Bei gleichen Rückführstellen haben LFSR mit zentraler und dezentraler Rückführung gleiche Zyklusstruktur.

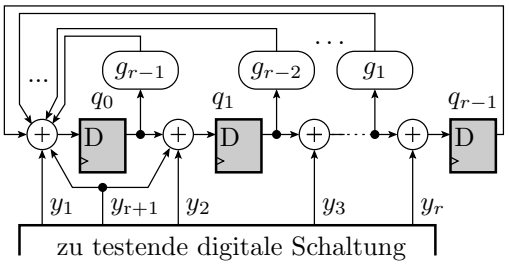
6.79 Zellenautomaten



- Folgebit gleich EXOR Nachbarbits optional plus eigener Bitwert.
- Mit ausgewälter Toggle-Zellenzuordnungen Maximalzyklus $2^r - 1$.

3.2 Signaturregister

6.80 Paralles Signaturregister



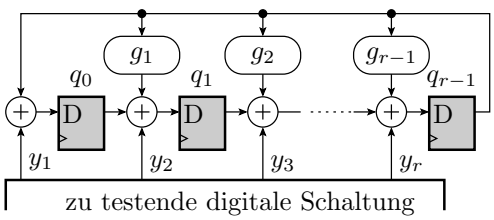
EXOR mit $g_1 = 0$ oder $x_i = 0$ entfallen durch Konstantenelimination

auch mehr als r Eingänge

Für die Bildung auf Prüfkennzeichen ist es nur wichtig, dass die Abbildung pseudo-zufällig hinsichtlich der zu erwartenden Verfälschungen erfolgt. Diese Eigenschaft hat auch ein rückgekoppeltes Schieberegister, bei dem die Daten modulo-2 als Bitvektoren zu den Registerzuständen addiert werden (paralleles Signaturregister, Folie 5.69 ff.) .

y_i Testobjektausgang i.
r Registerlänge.

6.81 Dezentrale Rückführung auch ok



- Autonome Zyklusstruktur bei gleichen Rückführkoeffizienten g_i für zentrale und dezentrale Rückführung gleich.
- Für Signaturregister werden, wie für Pseudo-Zufallsgeneratoren auch, primitive Rückkopplungen bevorzugt, bei denen bei $x_i = 0$ alle Zustände $\mathbf{q} \neq 00\ldots0$ zyklisch durchlaufen werden.

- Keine primitive Rückführung bedeutet aber nicht, dass dann die Fehlermaskierungswahrscheinlichkeit signifikant größer 2^{-r} ist.

---

g_i, q_i Registerbits und Rückführkoeffizienten des LFSR.
y_i Testobjektausgang i.

6.82 Experiment Maskierungswahrscheinlichkeit

Ein Signaturregister maskiert verfälschte Testausgabedaten durch Abbildung auf die Sollsignatur mit einer Wahrscheinlichkeit von:

$$p_{\mathrm{M}} = 2^{-r}$$

(vergl. Gl. 1.35). Wegen der geringen Eintrittswahrscheinlichkeit sollte die Anzahl der Maskierungen X poisson-verteilt sein:

(4.40) $$\mathbb{P}[X = k] = \mathrm{e}^{-\lambda} \cdot \frac{\lambda^k}{k!}$$

mit dem Erwartungswert $\mu = \lambda = \#F_{\mathrm{D}} \cdot 2^{-r}$.

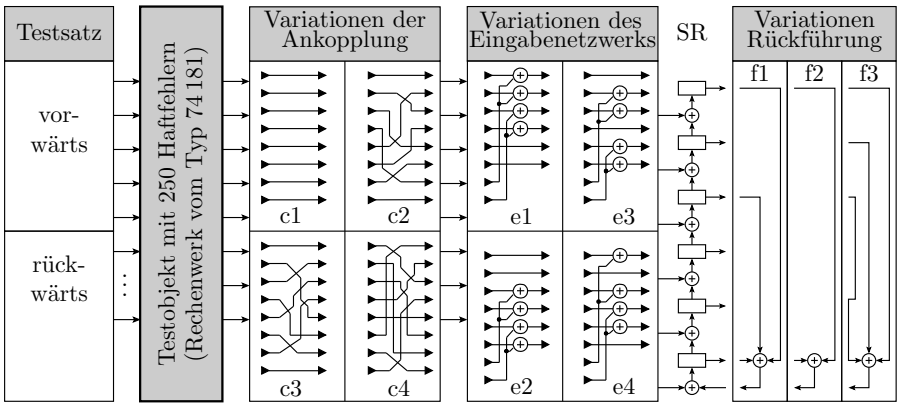
Kontrolle der Verteilung der Fehlermaskierung mit $\lambda = 250 \cdot 2^{-6} = 3{,}9$:

- Simulation einer 4-Bit-ALU mit einem Testsatz, der alle 250 unterstellten Haftfehler erkennt.
- Zählen der Maskierungen durch ein 6-Bit Signaturregister.
- für 96 Kombination der Testanordnung.

---

p_{M} Maskierungswahrscheinlichkeit (Mask probability).
$\#F_{\mathrm{D}}$ Anzahl der vom Testsatz nachweisbaren Fehler.
k Anzahl der von den $\#F_{\mathrm{D}} = 96$ vom Signaturregister maskieren Fehlern.

6.83 Versuchsplan

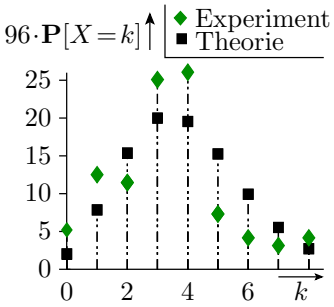


$2 \cdot 4 \cdot 4 \cdot 3 = 96$ Selbsttestanordnungen durch Kombination von

- 2 Testreihenfolgen,
- 4 Anschlussvarianten Testobjekt Signaturregister (c1 bis c4),
- 4 EXOR-Zusammenfassung von 8 Eingängen auf 6 Bit (e1 bis e4),
- 3 Rückführungen (f1 und f3 zufällig gewählt, f2 primitiv).

6.84 Anzahl der maskierten Haftfehler k

		e1				e2				e3				e4			
		c1	c2	c3	c4	c1	c2	c3	c4	c1	c2	c3	c4	c1	c2	c3	c4
vor-wärts	f1	3	4	1	2	3	4	3	3	4	2	4	3	4	3	4	6
	f2	3	4	1	7	2	2	1	4	2	1	1	3	2	5	3	7
	f3	5	2	2	8	4	5	3	4	3	6	3	7	5	3	3	4
rück-wärts	fb	6	4	4	2	3	4	3	4	3	4	3	4	4	8	4	5
	f2	2	0	0	1	4	1	4	1	0	0	0	1	1	1	4	1
	f3	2	4	3	4	4	8	5	8	3	3	3	6	3	3	4	3



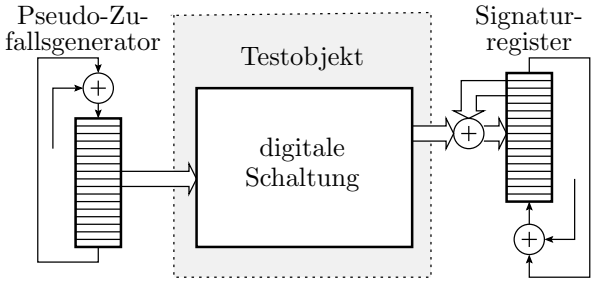
Erwartete und tatsächliche Häufigkeit der Anzahl der Maskierungen:

k	0	1	2	3	4	5	6	7	$\mathbb{E}[X=k]$
$96 \cdot e^{-3,9} \cdot \frac{3,9^k}{k!}$	1,3	7,5	14,7	19,2	18,7	14,6	9,5	5,3	3,9
Simulation	5	12	11	25	26	6	4	3	3

- Für nur 96 verschiedene Testanordnungen gute Übereinstimmung.
- Keine auffällig gute oder schlechte Testanordnung.
- Zufällig gewählte Rückführung nicht schlechter als primitive.

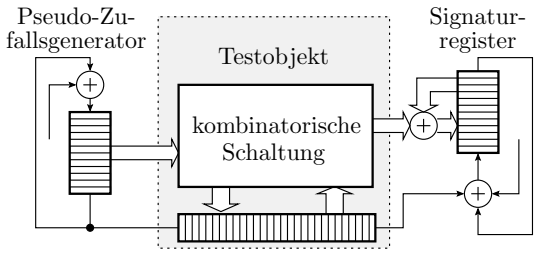
3.3 Selbsttest mit LFSR

6.85 Selbsttest mit LFSR



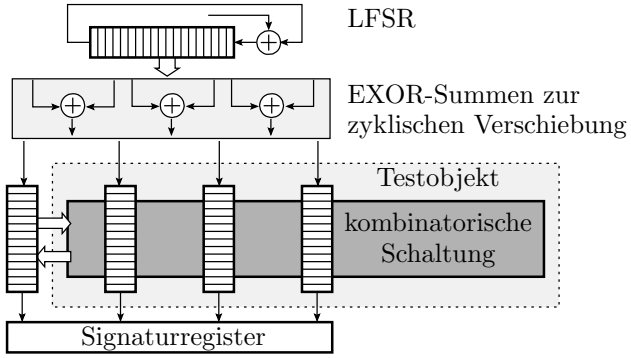
- Einrahmen der Schaltung mit Schieberegistern und Ergänzung von EXOR-Gattern für Rückführungen und SR-Eingänge.
- Init-Wert und Rückführung für Pseudo-Zufallsgenerator so wählen, dass sich keine Testeingabefolgen wiederholen.
- Wenn als Schieberegister vorhandene Ein- und Ausgaberegister verwendet werden, besonders niedriger Zusatzaufwand.
- Test mit voller Schaltungsgeschwindigkeit von Millionen bis Milliarden Tests pro Sekunde.

6.86 Selbsttest mit LFSR und Scan-Registern



- Ersatz interner Speicherzellen, vor allem solche, die Zustände speichern, durch Scan-Register.
- Zwischen den Testschritten wird das Scan-Register seriell in das Signaturregister auslesen und neu beschreiben.
- Lese- und Schreibzugriff auf die Zustandsspeicher reduzieren die erforderliche Testsatzlänge für dieselbe Fehlerabdeckung.
- Insgesamt kann sich die erforderliche Testzeit sogar verkürzen.

Für sehr große Systeme, z.B. Multi-Chip-Module auch mehrere Scan-Register, die zwischen den Testschritten parallel gelesen und mit neuen Zufallswerten beschrieben werden.



EXOR-Summen von LFSR-Bits bewirken zyklische Verschiebung. Vermeidung identischer Bitwerte in Scan-Registerzellen [Ke07].

[Ke07] G. Kemnitz: Test und Verläßlichkeit von Rechnern. Springer 2007.

3.4 Fehlerorientierte Wichtung

6.88 Fehlerorientierte Wichtung

Wenn ungewichtete Zufallsfolgen akzeptabler Länge nicht genug Modellfehler nachweisen, kann eine Testfortsetzung mit gewichteten Testprofilen helfen. Testprofile sind Eingabeprofile, die Eingaben aus Nachweismengen schlecht nachweisbarere Fehler extrem bevorzugen (Abschn. 2.3.8).

Wichtung und Beobachtbarkeit eines Bits x_i:
Wichtung ist die Auftrittswahrscheinlichkeit von Signalwert $x_i = 1$.

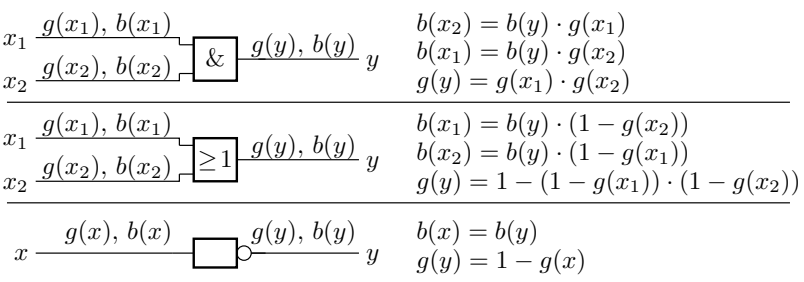
$$g(x_i) = \mathbb{P}[x_i = 1]$$

Beobachtbarkeit ist die Wahrscheinlichkeit, dass eine fehlerverursachte Invertierung eines Bits x_i mindestens ein Ausgabebit y_i verfälscht.

Geänderte Eingabewichtungen ändern die Wichtungen und Beobachtbarkeiten interner Signale und darüber die Fehlfunktionsraten der einzelnen Fehler zum Teil erheblich.

6.89 Wichtungen und Beobachtbarkeit

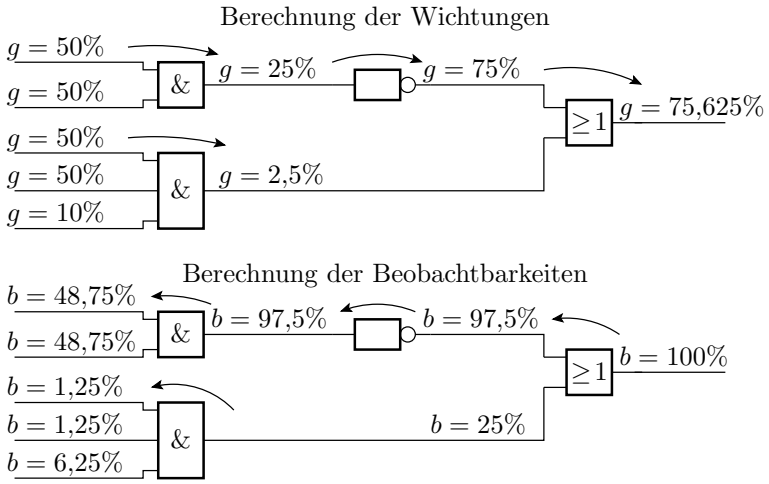
Die Wichtung einer UND-Verknüpfung ist das Produkt der Wichtungen der Operanden. ... Die Eingabe einer UND-Operation ist beobachtbar, wenn die andere Eingabe eins ist, ...



Wichtungen werden in Richtung und Beobachtbarkeiten entgegen der Richtung des Berechnungsflusses bestimmt.

$g(\ldots)$ Wichtung, Auftrittshäufigkeit des Signalwerts 1.

$b(\ldots)$ Beobachtbarkeit, Wahrscheinlichkeit, das eine Invertierung ein Ausgabebit ändert.



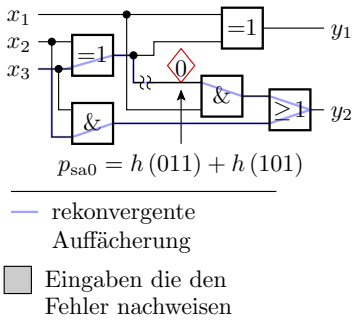
Berechnung der Nachweiswahrscheinlichkeiten:

$$p_{\text{sa0}}(x_i) = b(x_i) \cdot g(x_i)$$
$$p_{\text{sa1}}(x_i) = b(x_i) \cdot (1 - g(x_i))$$

6.91 Rekonvergente Auffächerung

Bei rekonvergenter Auffächerung werden abhängige Bitwerte verknüpft, so dass die einfachen Regel für $\mathbb{P}[A \wedge B]$ etc. nicht gelten.

Berechnung der Nachweiswahrscheinlichkeiten über Wertetabellen:



Eingabe			Ausg.		$h(\mathbf{x})$ für $g(x_i) = g$		
x_3	x_2	x_1	y_2	y_1	g=0,5	g=0,3	g=0,7
0	0	0	0	0	0,125	0,343	0,027
0	0	1	0	1	0,125	0,147	0,036
0	1	0	0	1	0,125	0,147	0,036
0	1	1	1	0	0,125	0,036	0,147
1	0	0	0	1	0,125	0,147	0,036
1	0	1	1	0	0,125	0,036	0,147
1	1	0	1	0	0,125	0,036	0,147
1	1	1	1	1	0,125	0,027	0,343
				p_{sa0} :	0,25	0,072	0,294

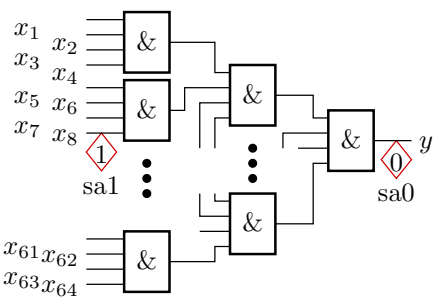
x_i, y_i Eingabe- und Ausgabesignale.

g Übereinstimmende Wichtung der drei Eingabesignale x_1 bis x_3.

$h(\mathbf{x})$ Auftrittshäufigkeit des Eingabevektors $\mathbf{x} = x_3x_2x_1$.

p_{sa0} Nachweiswahrscheinlichkeit des eingezeichneten sa0-Fehlers.

6.92 Fehlerabdeckung und Wichtung



Nachweiswahrscheinlichkeiten für $g(x_i) = g$:

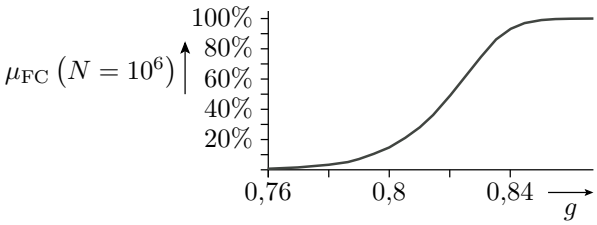
- der 64 sa1-Fehler
  $p_{\mathrm{sa1}}(1) = g^{63} \cdot (1-g)$
  $p_{\mathrm{sa1}}(N) = 1 - \mathrm{e}^{-g^{63}\cdot(1-g)\cdot N}$
- des einen sa0-Fehlers
  $p_{\mathrm{sa0}}(1) = g^{64}$
  $p_{\mathrm{sa0}}(N) = 1 - \mathrm{e}^{-g^{64}\cdot N}$

Zu erwartende Fehlerabdeckung als Mittelwert der Nachweiswahrscheinlichkeiten aller 65 Haftfehler:

$$\mu_{\mathrm{FC}}(N) = 1 - \frac{64 \cdot \mathrm{e}^{-g^{63}\cdot(1-g)\cdot N} + \mathrm{e}^{-g^{64}\cdot N}}{65}$$

g	Übereinstimmende Wichtung der Eingabesignale x_1 bis x_{64}.
$\mu_{\mathrm{FC}}(N)$	Zu erwartende Fehlerabdeckung in Abhängigkeit von der Testanzahl N.

$$\mu_{\mathrm{FC}}\left(N = 10^6\right) = 1 - \frac{64 \cdot \mathrm{e}^{-g^{63}\cdot(1-g)\cdot 10^6} + \mathrm{e}^{-g^{64}\cdot 10^6}}{65}$$



Eine Erhöhung der Wichtung an allen Eingängen von $g \leq 76\%$ auf $g \geq 86\%$ erhöht die Fehlerabdeckung für $N = 10^6$ Tests von 0 auf 1.

Allgemein sind zur Erzielung einer hohen Fehlerabdeckung die Wichtungen $g(x_i)$ für alle Eingänge x_i individuell anzupassen.

g	Übereinstimmende Wichtung der Eingabesignale x_1 bis x_{64}.
$\mu_{\mathrm{FC}}(N)$	Zu erwartende Fehlerabdeckung in Abhängigkeit von der Testanzahl N.

6.95 Fehlerorientierte Wichtungsanpassung

Pragmatischer Ansatz aus [HaK93]:

1. Zusammenstellung der Haftfehlermenge.
2. Längerer Zufallstest, z.B. $N = 10^6$ und Abhaken aller damit nachweisbaren Modellfehler.
3. Gezielte Berechnung von Tests für die übrigen Modellfehler. Eingabebits, die für den Fehlernachweis keinen bestimmten Wert haben müssen, bleiben »X«.
4. Wenn die gefundenen Tests für ein Eingabebit x_i überwiegend null verlangen $g(x_i) < 0{,}5$ und für x_i überwiegend eins $g(x_i) > 0{,}5$ (Details siehe nächste Folie).
5. Längerer gewichteter Zufallstest, z.B. auch wieder $N = 10^6$, und auch wieder Abhaken aller damit nachweisbaren Modellfehler.
6. Wenn immer noch nicht alle Modellfehler nachweisbar sind, Wiederholung von Schritt 4 und 5.

N	Anzahl der Tests.
[HaK93]	J. Hartmann, G. Kemnitz: How to do weighted random testing for BIST? ICCAD 1993.

Zur Schaltungsminimierung des Testmustergenerators ist eine Beschränkung auf nur 5 verschiedene Wichtungswerte zweckmäßig:

$$g(x_i) = \begin{cases} 0 & \text{wenn } \forall x_{i.j} \in \{0, \mathrm{X}\} \\ 1 & \text{sonst wenn } \forall x_{i.j} \in \{1, \mathrm{X}\} \\ 2^{-m_{\mathrm{AND}}} & \text{sonst wenn } \#\mathtt{0} \gg \#\mathtt{1} \\ 1-2^{-m_{\mathrm{AND}}} & \text{sonst wenn } \#\mathtt{0} \ll \#\mathtt{1} \\ 0{,}5 & \text{sonst} \end{cases}$$

Beispiel für $m_{\mathrm{AND}} = 3$:

Testvektor:		1	2	3	4	5	6	7	8	9	10	11	$g(x_i)$
Eingabebit	x_1	1	X	0	1	0	0	X	1	0	1	1	0,5
	x_2	1	1	X	1	1	X	1	X	X	X	X	1
	x_3	0	0	1	0	0	X	0	X	1	0	0	2^{-3}
	x_4	1	0	X	0	X	0	1	X	1	X	0	0,5
	x_5	1	1	1	X	0	1	1	1	X	0	1	$1-2^{-3}$

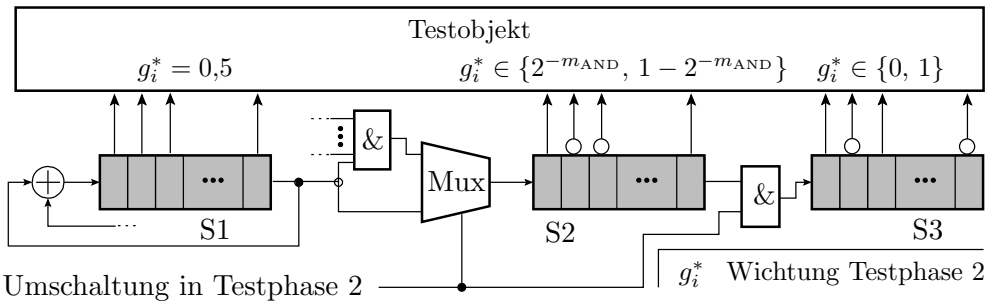
X Signalwert ungültig oder für den Fehlernachweis ohne Bedeutung.

#0 Anzahl der Nullen im Testsatz für Eingang i des Testobjekts.

#1 Anzahl der Einsen im Testsatz für Eingang i des Testobjekts.

m_{AND} Anzahl UND-verknüpfter ungewichteter Bits für die Wichtung $0 < g \ll 0{,}5$.

6.97 Implementierung als Selbsttest



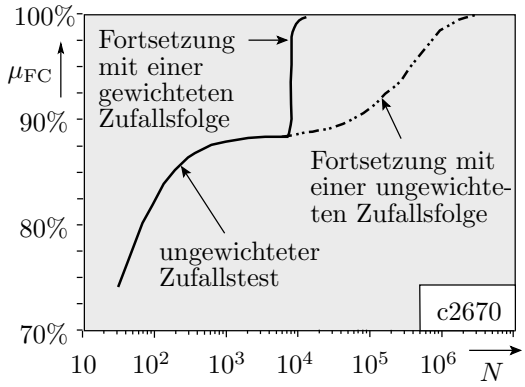
- Testphase 1: Erzeugung ungewichteter Pseudo-Zufallseingaben mit LFSR S1. Serielle Weitergabe an die Schiebereg. S2 und S3.
- Testphase 2: Verringerung der Wichtung
  - in S2 durch UND-Verknüpfung von m_{AND} Ausgabefolgen von S1 und für S2 und
  - in S3 durch »UND 0«.

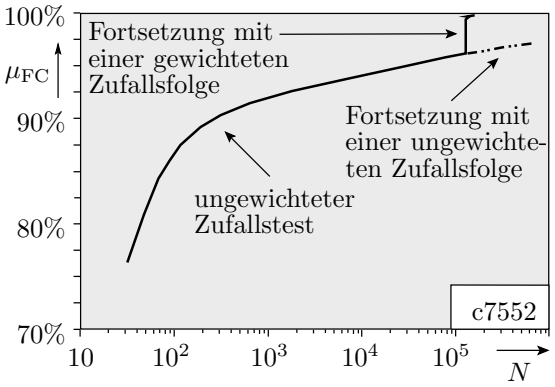
  Erzeugung $g(x_i) = 1-2^{-m_{\mathrm{AND}}}$, $g(x_i) = 1$ durch Inverierung.

Kaum aufwändiger als Selbstest nur mit ungewichtetem Zufallswerten.

6.98 Beispielentwurf mit ISCAS-Benchmarks

- Test mit 10^4 bzw. 10^5 ungewichteten Zufallsmustern, die 90% bzw. 95% der Haftfehler nachweisen.
- Gezielte Testberechnung und Wichtung wie beschreiben
- Weitere 10^4 bzw. 10^5 gewichtet Zufallstests weisen alle verbleibenden nichtredundanten Modellfehler nach.
- Bestätigung der These, dass der Test mit Testprofil den Anteil der nicht nachweisbaren Fehler noch einmal etwa um $(N/N_0)^{-K}$ und mit beiden Profilen zusammen um $(N/N_0)^{-2K}$ verringern.





μ_{FC}, N Zu erwartende Haftfehlerabdeckung, Testanzahl.

3.5 RAM-Selbsttest

6.99 Marching-Test für RAM (Folie 6.67)

Adresse i	Initialisierung	March 1	March 2	March 3
0	$W(i)0$	$R(i)0, W(i)1$	$R(i)1, W(i)0$	$R(i)0, W(i)1$
1	$W(i)0$	$R(i)0, W(i)1$	$R(i)1, W(i)0$	$R(i)0, W(i)1$
2	$W(i)0$	$R(i)0, W(i)1$	$R(i)1, W(i)0$	$R(i)0, W(i)1$
⋮	⋮	⋮	⋮	⋮
$N-1$	$W(i)0$	$R(i)0, W(i)1$	$R(i)1, W(i)0$	$R(i)0, W(i)1$

	March 4		March 1a		March 2a
0	$R(i)1, W(i)0$		$R(i)0, W(i)1$		$R(i)1$
1	$R(i)1, W(i)0$		$R(i)0, W(i)1$		$R(i)1$
2	$R(i)1, W(i)0$	Wartezeit	$R(i)0, W(i)1$	Wartezeit	$R(i)1$
⋮	⋮		⋮		⋮
$N-1$	$R(i)1, W(i)0$		$R(i)0, W(i)1$		$R(i)1$

Die Testeingabebereitstellung und -auswertung vorlangt einen Vor-/Rückwärtszähler und eine kleine Steuerung, die Zählrichtung, Schreibwert und Soll-Lese-Wert auswählt. Wartezeiten können von außen über den Testbus gesteuert werden.

Zusammenfassung

6.100 Selbsttest digtaler Schalkreise

- Ergänzung von Pseudo-Zufassgeneratoren an den Eingängen, in der Regel LFSR, für die Durchführung langer Zufallstests und Signaturregistern an den Ausgängen.
- Für die Testmustergeneratoren sind die Rückführungen und Startwerte so zu wählen, dass nicht mehrfach dieselbe Testfolge wiederholt wird und die Musterabhängigkeiten den Fehlernachweis nicht stören (bevorzugt primitive Rückführungen).
- Die Signaturregister sind so zu wählen, dass die Maskierungswahrscheinlichkeit z.B. durch falsche Rückführung den Wert 2^{-r} nicht übersteigt. Ein Experiment hat gezeigt, dass man da nicht viel falsch machen kann.
- Zur Verbesserung der Fehlerabdeckung werden auch interne Speicherzellen zu Scan-Registern zusammengefasst, die zwischen den Testschritten in das Signaturregister ausgelesen und mit neuen Pseudo-Zufallswerten beschrieben werden.

6.101 Fehlerorientierte Wichtung und RAM

- Wenn die Fehlerabdeckung mit $N = 10^{6...9}$ Zufallstests nicht ausreicht, gibt es auch die Möglichkeit, in weiteren Testphasen die Auftrittshäufigkeiten der Nullen und Einsen an den Testobjekteingängen so anzupassen, dass die mit ungewichteten Zufallsmustern zu schlecht nachweisbaren Fehler bevorzugt nachgewiesen werden.

- Dazu werden im einfachten Fall für alle in der ersten Testphase mit ungewichteten Zufallswerten nicht nachweisbaren Modellfehler Tests gesucht und die Wichtungen an die Häufigkeiten der Nullen und Einsen im gefundenen Testsatz angepasst.

- Für regelmäßig strukturierte Schaltungen wie RAM gibt es oft einfachere Lösungen mit exzellenter Fehlerabdeckung. Als Beispiel wurde die Implementierung eines Marching Tests für einen RAM-Block skizziert.

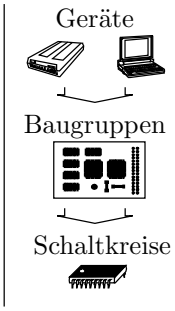
4 Baugruppentest

6.102 Hierarchischer Test der Hardware

Rechnerhardware besteht aus tauschbaren Komponenten:

- tauschbare Teilsysteme.
- tauschbare Leiterplatten,
- austauschbare Bauteile.

Die Komponenten werden vor Einbau in das übergeordnete System gründlich getestet.

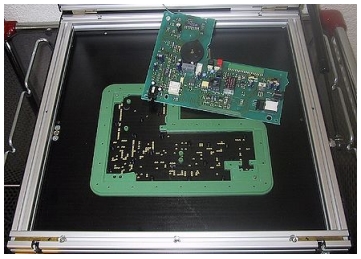


Wiederholungsfragen

1. Warum tauschbare Komponenten bzw. unter welcher Bedingung nicht erforderlich (nicht zweckmäßig)?
2. Warum ist ein gründlicher Komponententest zu fordern?

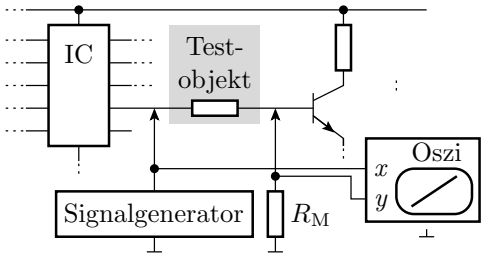
6.103 Leiterplattentest (Folie 2.47)

Hauptfehler auf Baugruppen sind Kurzschlüsse und Unterbrechungen. Nachweis durch Widerstandsmessungen zwischen und entlang der Verbindungen.
In der Serienfertigung erfolgt die Kontaktierung mit einem mit Unterdruck angesaugten Nadeladapter.



Die Nadeln sind mit einer Relais-Matrix verbunden, über die vom Testprogramm Prüfgeräten angeschlossen werden. Auch Bestückungsfehler lassen sich überwiegend mit Zweipunktmessungen von Strom-Spannungsbeziehungen erkennen. Fehlerlokalisierung im Vergleich zur Rückverfolgung falsche Ausgaben von dynamischen Tests sehr einfach (vergl. Abschn. 2.2.2).

6.104 Zweipunktmessung Bestückungkontrolle

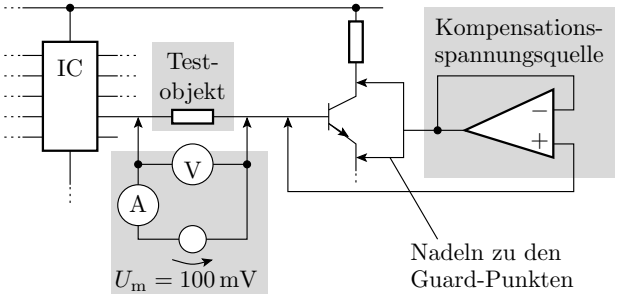


Kontrolle auf Bestückungsfehler durch Überprüfung ausgewählter Zweipunktmerkmale:

- Widerstandswerte,
- Kapazitäten,
- Flussspannungen von Dioden, ...

Verbindungskontrolle IC-Anschlüsse zu den Kontakt-Pads durch Ausmessen der Schutzdioden zur Versorgungsspannung oder Masse.

6.105 Analoger In-Circuit Test

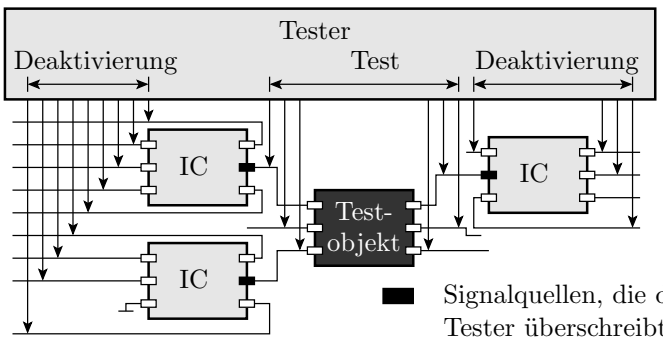


Die Strom-Spannungs-Beziehung zwischen zwei Punkten hängt nicht nur vom Bauteil zwischen den Nadeln, sondern von allen Strompfaden, im Beispiel durch Transistor und Schaltkreis ab.

Unterdrückung von Parallelströmen zum Testobjekt durch Kompensation der Spannungsabfälle über den wegführenden Bauteilen auf einer Testobjektseite auf null über »Guard-Punkte«:

- Vereinfacht die Testprogrammerstellung und Fehlerlokalisierung, insbesonderen die Sollwert- und Sollwerttoleranzfestlegung .
- Mindert die Häufigkeit von Fehlklassifikationen.

6.106 Digitaler In-Circuit-Test

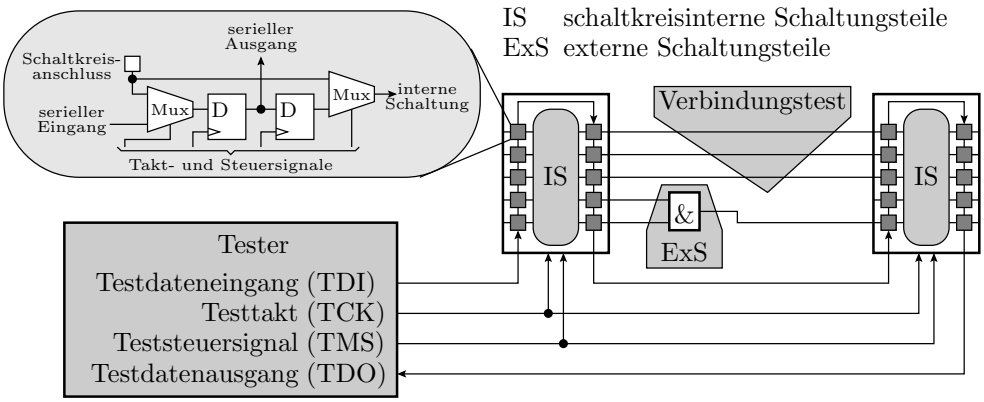


- Kontaktierung der Baugruppe über ein Nadelbetadapter.
- Isolierter Test der Schaltkreise durch Überschreiben der digitalen Schaltkreiseeingaben mit stromstarken Treibern.

- Im Gegensatz zum analogen ICT unter Spannung.
- Andere Schaltkreise möglichst deaktivieren (Ausgänge trennen).

Erlaubt auch den isolierten Test der Bausteine, aber im Gegesatz zum analogen In-Circuit-Test unter Spannung.

6.107 Boundary-Scan (Folie 5.38)



Die ursprüngliche Ideee von Boundary-Scan war der Ersatz der mechanischen Nadeln durch »silicon nails« als Alternative zu den teuren, für jede Baugruppe speziell anzufertigenden Nadeladaptern. Verbindungs- und Bestückungstest über die seriell les- und beschreibbaren Scan-Register an den Schaltkreisanschlüssen.

6.108 Optische Inspektion

Bild links korrekt bestückter SMD-Widerstand, rechts Lötfläche durch Kleber verschmutzt. Elektrisch leitende aber keine feste Lötverbindung. Nachweis nur durch visuelle Kontrolle möglich. Nach Ausfall der Baugruppe z.B. durch Vibration in einem Fahrzeug ist sofort erkennbar, dass es sich um einen optisch nachweisbaren Fertigungsfehler handelt.

Wenn ein solcher Fertigungsfehler Schaden verursacht, z.B. einen Unfall, greift die Produkthaftung, d.h. der Hersteller muss für den entstandenen Schaden aufkommen. Damit ist eine optische Inspektion zwingend, für Prototypen und kleine Stückzahlen manuell, in der Serien und Massenfertigung mit einem Bildverarbeitungssystem, dass das Bild jeder Lötstelle mit einem Referenzbild vergleicht.

Kontrollfragen

- Zählen Sie Maßnahmen des prüfgerechten Entwurfs für den Baugruppentest auf.
- Was bedeutet isolierter Test von Komponenten und wie setzen der analoge und der digitale In-Circuit-Test dieses Prinzip um?
- Warum ist für SMD-bestückte Baugruppen von Steuergeräten für KFZ eine optische Inspektion zwingend?
- Welche prüftechnischen Probleme des Baugruppentests löst Boundary-Scan?

5 Ausfälle

6.110 Ausfälle

Hardware und Mechanik unterliegt einem Verschleiß, der zu Ausfällen führen kann. Bei einem Ausfall entsteht ein Fehler, der oft mehr Fehlfunktionen (MF) als alle vom Test nicht erkannten Fehler zusammen oder ein komplettes Versagen (NS) verursacht.

Kenngrößen für das Ausfallverhalten:

- Lebensdauer, Überlebenswahrscheinlichkeit,
- Ausfallrate, ...

Maßnahmen zum Umgang mit Ausfällen:

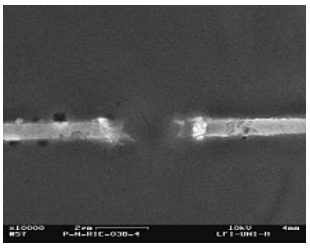
- Voralterung,
- Wartung,
- Redundanz (kalte oder heiße Reserve).

In Software entstehen während des Betriebs keine neuen Fehler, ausgenommen geplante Obsoleszenz (Programm-Features zur Vortäuschung von Ausfällen) und Fehler, die bei Beseitigungsversuchen anderer Fehler entstehen.

MF Fehlfunktion (Malfunction).
NS Keine Service-Leistung.

6.111 Verschleiß elektronischer Bauteile

Auch elektronische Bauteile unterliegen einem Verschleiß.



Langsam ablaufende physikalische Vorgänge:

- Korrosion (Stecker, Schalter, Isolationen, Leiterbahnen, ...).
- Elektromigation: strombedingte Wanderung von Metalatomen bei hohen Stromdichten.
- Parameterdrift: Widerstandswerte, Kapazitäten, Schwellspannungen etc.
- Gateoxiddurchschlag: Hochschaukelnde Tunnelströme, Ladungseinlagerung bis zum lokalen Schmelzen des Oxids. Bildung von Kurzschlüssen. Phänomen: Zunahme des Stromverbrauchs über Monate bis zum Ausfall.
- Heiße Elektronen im Einschnürrbereich, die durch Gitterstreuungen ins Gateoxid gelangen.

5.1 Modelle, Kenngrößen

6.112 Kenngrößen des Ausfallverhaltens

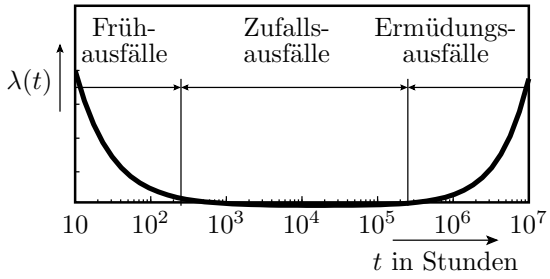
- Lebensdauer L: Zufallsvariable. Zeit vom Beanspruchungsbeginn bis zum Ausfall. Verteilungsfunktion:
$$F_L(t) = \mathbb{P}[L \leq t] \tag{6.1}$$
- Überlebenswahrscheinlichkeit:
$$V(t) = \mathbb{P}[L > t] = 1 - F_L(t)$$
- Ausfallrate λ: Relative Abnahme der Überlebenswahrscheinlichkeit:
$$\lambda(t) = -\frac{1}{V(t)} \cdot \frac{\mathrm{d}V(t)}{\mathrm{d}t}$$

- Mittlere Lebensdauer:

$$\mu_{\mathrm{L}} = \int_0^{\infty} t \cdot \frac{\mathrm{d}(F_L(t))}{\mathrm{d}t} \cdot \mathrm{d}t$$

L	Lebensdauer, Zufallsvariable.
$V(t)$	Überlebenswahrscheinlichkeit als Funktion der Lebensdauer.
$\lambda(t)$	Ausfallrate als Funktion der Zeit.
μ_{L}	Zu erwartende Lebensdauer.

6.113 Ausfallphasen



- Frühausfälle (infant mortalities): Erhöhte Ausfallrate zu Beginn der Nutzung durch Schwachstellen (Materialrisse, lokal stark überhöhte Feldstärke oder Stromdichte, ...).
- Zufallsausfälle: Näherungsweise konstante Ausfallrate.
- Ermüdungsausfälle: Anstieg der Ausfallrate in der Nutzungsendphase: Materialermüdung, ...

$\lambda(t)$	Ausfallrate als Funktion der Zeit.

6.114 Zufallsausfälle

In der Hauptnutzungsphase ist die Ausfallrate konstant:

$$\lambda(t) = -\frac{1}{V(t)} \cdot \frac{\mathrm{d}V(t)}{\mathrm{d}t} = \lambda = \text{konst.}$$

Maßeinheit der Ausfallrate: fit (failure in time)

$$1\,\text{fit} = 1\,\text{Ausfall in } 10^9 \text{ Stunden}$$

Überlebenswahrscheinlichkeit und Verteilung der Lebensdauer*:

$$V(t) = \mathrm{e}^{-\lambda \cdot t} \tag{6.2}$$

$$F_L(t) = 1 - \mathrm{e}^{-\lambda \cdot t} \tag{6.3}$$

Die Lebensdauer ist exponentialverteilt und hat den Erwartungswert:

$$\bar{t}_{\mathrm{FL}} = \mu_{\mathrm{L}} = \int_0^{\infty} t \cdot \lambda \cdot \mathrm{e}^{-\lambda \cdot t} \cdot \mathrm{d}t = \frac{1}{\lambda} \tag{6.4}$$

$V(t)$	Überlebenswahrscheinlichkeit als Funktion der Lebensdauer.
fit	Failure in Time, Anzahl der Ausfälle in 10^9 Stunden.
*	Kontrolle: $\lambda(t) = -\frac{1}{\mathrm{e}^{-\lambda \cdot t}} \cdot \frac{\mathrm{d}\left(\mathrm{e}^{-\lambda \cdot t}\right)}{\mathrm{d}t} = \lambda \surd$.

6.115 System aus mehreren Komponenten

Ein System aus mehreren notwendigen Bauteilen überlebt, solange alle Bauteilen überleben:

$$V(t) = \prod_{i=1}^{\#Prt} V(t)_i$$

Mit einer konstanten Ausfallrate λ_i für alle Bauteilen (Gl. 6.2):

$$V(t) = \prod_{i=1}^{\#Prt} \mathrm{e}^{-\lambda_i \cdot t} = \mathrm{e}^{-\left(\sum_{i=1}^{n} \lambda_i\right) \cdot t}$$

ist die Gesamtausfallrate die Summe der Ausfallraten aller Bauteile:

$$\lambda_{\mathrm{Sys}} = \sum_{i=1}^{\#Prt} \lambda_i \tag{6.5}$$

$V(t)$ Überlebenswahrscheinlichkeit als Funktion der Lebensdauer.
$\#Prt$ Anzahl der Bauteile.
λ_{Sys} Ausfallrate des Systems.
λ_i Ausfallrate Komponente i.

6.116 Ausfallraten Hauptnutzungsphase

Bauteiltyp	λ in fit	Bauteiltyp	λ in fit
diskrete SC	1 bis 100	Widerstände	1 bis 20
digitale IC	50 bis 200	Kondensatoren	1 bis 20
ROM	100 bis 300	Steckverbinder	1 bis 100
RAM	bis 500	Lötstellen	0,1 bis 1
analoge IC	20 bis 300		

λ Ausfallrate (Failure rate).
SC Halbleiterbauteile (Semiconductor components).
IC Integrierte Schaltkreise (Integrated circuits).
fit Failure in Time, Anzahl der Ausfälle in 10^9 Stunden.

Kataloge für Ausfallraten:

- MIL-HDBK-217-Military Handbook: Reliability Prediction for Electronic Equipment. United States of America, Department of Defense. Washington, 1991
- Siemens AG: SN 29500: Siemens Norm
- IEC International Electrotechnical Commission: IEC/TR 62380:2004(E) : Reliability data handbook - Universal model for reliability prediction of electronic components, PCBs and equipment, 2004

6.117 Ausfallrate einer Baugruppe

Bauteiltyp i	Anzahl $\#Prt_i$	Ausfallrate λ_i	$\#Prt_i \cdot \lambda_i$
Schaltkreise	20	150 fit	3000 fit
diskrete SC	15	30 fit	450 fit
Kondensatoren	15	10 fit	250 fit
Widerstände	30	10 fit	300 fit
Lötstellen	2000	0,5 fit	1000 fit
λ Baugruppe			5000 fit

- Im Mittel 1 Ausfall in $2 \cdot 10^5$ Stunden (≈23 Jahre) Betriebsdauer.
- Von den heutigen PCs, Handys, ... fallen pro Jahr und hundert Stück nur wenige aus.

- Nach 2 ... 5 Jahren erste Ermüdungsausfälle, insbesondere der Akkus.

SC Halbleiterbauteile (Semiconductor components).
$\#Prt_i$ Anzahl der Bauteile vom Typ i.
λ_i Ausfallrate der Bauteile vom Typ i.
fit Failure in Time, Anzahl der Ausfälle in 10^9 Stunden.

6.118 Frühausfälle

- Auf 100 richtige Fehler kommt etwa ein Beinahefehler, der zu einem Frühausfall führt [BS03].
- Bei 50% fehlerfreien und 50% aussortierten Schaltkreisen $50\%/100 = 0{,}5\%$ Beinahefehler.
- Die Hälfte wird mit dem Ausschuss aussortiert.
  - $\approx 0{,}25\%$ (jeder 400ste) Schaltkreis verursacht einen Frühausfall.
  - Bei 20 Schaltkreisen pro Gerät jedes zwanzigste Gerät.
  - Bei großen Systemen fast jedes System.
- Frühausfälle sind Garantiefälle und verursachen Kosten für Reparatur, Ersatz, Auftragsabwicklung, ... Was tun?

  $\Rightarrow$ Voralterung

  (Problem existiert auch bei Mechanik.)

$\lambda(t)$ Ausfallrate als Funktion der Zeit.
[BS03] Singh, A. D.: Relating Yield Models to Burn-In Fall-Out in Time. ITC, 12/2003, S.77-84.

5.2 Gegenmaßnahmen

6.119 Ausfallschwere und Gegenmaßnahmen

Einteilung ausfallverursachter Fehler nach Beeinträchtigung:

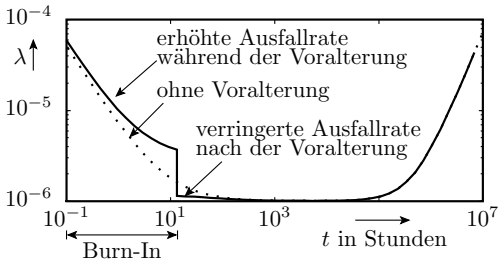
- schwer: System erst nach Fehlerbeseitigung wieder benutzbar.
- toleriert: Service von Redundanzen übernommen. Bis zur Reparatur erhöhtes Risiko für schwere Ausfälle.
- leicht: Verringerte Zuverlässigkeit bis zur nächsten Wartung.
- vorhersagbar: Messbare Anzeichen für bevorstehende Ausfälle.
- verborgen: Unbemerkte Zuverlässigkeitsminderung.

Für jede Ausfallschwere gibt es andere Gegenmaßnahmen.

Als von der Fehlerschwere unabhängige Gegenmaßnahmen kommen hinzu die Vermeidung erhöhten Ausfallraten

- in der Frühphase durch Voralterung und
- in der Verschleißphase durch rechtzeitigen Ersatz.

6.120 Voralterung (Burn-In)

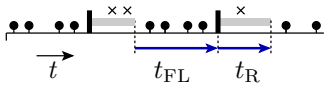


Beschleunigung der Alterung vor dem Einsatz durch »harte Umgebungsbedingungen«:

- überhöhte Spannung,
- überhöhte Temperatur,
- Stress (Burn-In).

Einsatzbeginn erst nach der Frühausfallphase, wenn die kränklichen Bauteile »gestorben« und ausgetauscht sind.

6.121 Sofortige Reparatur



Nur die schweren Ausfälle beeinträchtigen die Verfügbarkeit. Die Ausfallrate zur Abschätzung der mittleren Zeit $\bar{t}_{FL}$ zwischen Ausfällen in

(1.6)
$$A_H = \frac{\bar{t}_{FL}}{\bar{t}_{FL}+\bar{t}_R} \overset{\bar{t}_{FL}\gg\bar{t}_R}{=} 1 - \frac{\bar{t}_R}{\bar{t}_{FL}} \tag{6.4}$$

$$\bar{t}_{FL} = \mu_L = \int_0^\infty t \cdot \lambda \cdot e^{-\lambda\cdot t} \cdot dt = \frac{1}{\lambda}$$

ist nur die für schwere Ausfälle. Hardwareverfügbarkeit und mittlere Wahrscheinlichkeit der Nichtverfügbarkeit:

$$A_H = 1 - \lambda \cdot \bar{t}_R \tag{6.6}$$

$$PFD = 1 - A_H = \lambda \cdot \bar{t}_R \tag{6.7}$$

$\bar{t}_{FL}$, $\bar{t}_R$ Mittlere Nutzungsdauer zwischen schweren Ausfällen, mittlere Reparaturdauer.

λ Ausfallrate für schwere Ausfälle, die die Verfügbarkeit beeinträchtigen (Failure rate for serious failures that affect availability).

A_H, PFD Hardware-Verfügbarkeit, Probability of Failure on Demand.

6.122 Hohe Hardware-Verfügbarkeit

(6.6)
$$A_H = 1 - \lambda \cdot \bar{t}_R$$

Zulässige mittlere Reparaturdauer:

$$\bar{t}_R = \frac{1-A_H}{\lambda}$$

A_H	$\lambda = 10^7$ fit	$\lambda = 10^6$ fit	$\lambda = 10^5$ fit
$A_H = 99\%$	1 h	10 h	100 h
$A_H = 99{,}9\%$	0,1 h	1 h	10 h
$A_H = 99{,}99\%$	0,01 h	0,1 h	1 h

$A_H \approx 99\%$ ist normal. Hohe Verfügbarkeit ab 99,9% verlangt Redundanzen zur schnellen Service-Übernahme bei Ausfall:

- unterbrechungsfreie Stromversorgung,
- RAID (**R**edundant **A**rray of **I**ndependent **D**isks, Abschn. 6.6.4),
- gespiegelte Server, ...

λ, $\bar{t}_R$ Ausfallrate für schwere Ausfälle, mittlere Reparaturdauer.
A_H Hardware-Verfügbarkeit.
fit Failure in Time, Anzahl der Ausfälle in 10^9 Stunden.

6.123 Wartung(LIFW)

»Werkstattbesuch« zur Fehlervermeidung und -beseitigung durch Ausfälle. Oft konstantes Wartungsintervall τ.

1. Reparatur oder Ersatz von Komponenten, die an Redundanzen übergaben haben.
2. Wartungstests und Beseitigung der damit erkennbaren zuverlässigkeitsmindernder Fehler durch Ausfälle.
3. Ersatz von Betriebsstoffen und Verbrauchsmitteln, Schmierstoffe bei Getrieben, Papier und Toner bei Druckern, ...
4. Planmäßiger Austausch von Verschleißteilen vor der Ermüdungsphase, z.B. BIOS-Batterien und Server-Festplatten.
5. Auswertung von Gesundheitsdaten, Ursachensuche und -beseitigung gehäuft beobacht- und lokalisierbarer Probleme.

(1, 2) reaktive Wartung Beseitigung erkennbarere entstandener Fehler, (2-4) vorausschauende Wartung zur Ausfallvermeidung [F. Beichelt, Reliability and Maintenance Theory,Teubner, Stuttgart, 1993].

LIFW Großes eigenständiges Forschungs- und Arbeitsgebiet.
τ Wartungsintervall.

6.124 Gesundheitsüberwachung(LIFW)

Gesundheitsmonitor: Eigenständiges Teilsystem zur Sicherheitsverbesserung, das Auffälligkeiten für die nächste Wartung mitschreibt:

- Häufung von Busfehlern, Zeitüberschreitungen und anderer von Steuergeräten signalisierte Fehlfunktionen,
- Anzeichen auf Sensor- oder Aktorenfehler,
- Hinweise auf Korrosion, Verformung, Rissbildung, Abnutzung, ... mechanischer Bauteile von resitiven, kapazitiven, induktiven, Dehnungs- und Beschleinigungssensoren.
- erhöhter Stromaufnahme, verkürzte Akkulaufzeiten (Folie 6.111).

Bei Schwellwertüberschreitung der Häufigkeit und Schwere erfasster Anormalien vorausschauende Maßnahmen zur Risikominderung:

- Funktionen deaktiviert, Weiternutzung unterbindet (Maximalgeschwindigkeit von Autos drosselt oder Startverbot für Flugzeuge),
- Wartung anfordert (z.B. Anzeige Werkstattsymbol), ...

LIFW Großes eigenständiges Forschungs- und Arbeitsgebiet.

6.125 Zuverlässigkeitsminderung

Leichte Aufälle: Fehleranzahl nach der Wartung null. Zunahme mit einer Ausfallrate λ_{FW}. Am Ende des Wartungsintervalls $\lambda_{\mathrm{WF}} \cdot \tau$. Fehlfunktionsrate je leichter Fehler im Mittel $\bar{\gamma}_{\mathrm{WF1}} \gg 1$. Zuverlässigkeitsminderung:

$$\frac{1}{R_{\mathrm{FW}}} = \bar{\gamma}_{\mathrm{FW}} = \frac{\lambda_{\mathrm{WF}} \cdot \tau \cdot \bar{\gamma}_{\mathrm{WF1}}}{2} \tag{6.8}$$

Verborgene Ausfälle: Zunahme der zu erwartenden Anzahl mit Ausfallrate λ_{FH} mit der Nutzungsdauer t_{OP}. Viele kleinere mittlere Fehlfunktionsrate je Fehler $\bar{\gamma}_{\mathrm{WH1}} \ll \bar{\gamma}_{\mathrm{WF1}}$als leichte Ausfälle:

$$\frac{1}{R_{\mathrm{FH}}} = \bar{\gamma}_{\mathrm{FH}} = \lambda_{\mathrm{WH}} \cdot t_{\mathrm{OP}} \cdot \bar{\gamma}_{\mathrm{WH1}} \tag{6.9}$$

λ_{WF}, $\bar{\gamma}_{\mathrm{WF1}}$	Ausfallrate und mittlere Fehlfunktionsrate je Fehler für schwache Ausfälle.
R_{FW}, γ_{FW}	Teilzuverlässigkeit und gesammte Fehlfunktionsrate für schwache Ausfälle.
λ_{WH}, $\bar{\gamma}_{\mathrm{WH1}}$	Ausfallrate und mittlere Fehlfunktionsrate je Fehler für unsichtbare Ausfälle.
R_{FH}, γ_{FH}	Teilzuverlässigkeit und gesammte Fehlfunktionsrate für unsichtbare Ausfälle.
τ, t_{OP}	Wartungsintervall, Dauer der Nutzung.

6.126 Gegenmaßnahmen

(6.8) $\frac{1}{R_{\mathrm{FW}}} = \bar{\gamma}_{\mathrm{FW}} = \frac{\lambda_{\mathrm{WF}} \cdot \tau \cdot \bar{\gamma}_{\mathrm{WF1}}}{2}$ (6.9)

$\frac{1}{R_{\mathrm{FH}}} = \bar{\gamma}_{\mathrm{FH}} = \lambda_{\mathrm{WH}} \cdot t_{\mathrm{OP}} \cdot \bar{\gamma}_{\mathrm{WH1}}$ **Gegenmaßnahm**

Zuverlässigkeitsminderung **leichte Ausfälle**:

- $\bar{\gamma}_{\mathrm{WF1}}$ durch bessere Überwachung auf Fehlfunktionen während des Betriebs und
- kürzere Wartungsintevalle τ mindern.

Gegenmaßnahmen Zuverlässigkeitsmind. **verborgene Ausfälle**:

- höhere effektive Testanzahl der Wartungstests und
- prophylaktischen Ersatz nach einer kürzeren Betriebsdauer t_{OP}.

Wartung verbessert die Verfügbarkeit, Zuverlässigkeit und Sicherheit.

λ_{WF}, $\bar{\gamma}_{\mathrm{WF1}}$	Ausfallrate und mittlere Fehlfunktionsrate je Fehler für schwache Ausfälle.
R_{FW}, γ_{FW}	Teilzuverlässigkeit und gesammte Fehlfunktionsrate für schwache Ausfälle.
λ_{WH}, $\bar{\gamma}_{\mathrm{WH1}}$	Ausfallrate und mittlere Fehlfunktionsrate je Fehler für unsichtbare Ausfälle.
R_{FH}, γ_{FH}	Teilzuverlässigkeit und gesammte Fehlfunktionsrate für unsichtbare Ausfälle.
τ, t_{OP}	Wartungsintervall, Dauer der Nutzung.

Zusammenfassung

6.127 Ausfall von Hardware

Ausfall bedeutet, dass durch Verschleiß oder Zerstörung neue Fehler entstehen. Verschleißmechanismen elektronischer Bauteile sind z.B. Korrosion und Elektromigration. Kenngrößen des Ausfallverhaltens:

- Lebensdauer L (Zufallsgröße) und deren Erwartungswert μ_{L},
- Überlebenswahrscheinlichkeit $V(t)$ und
- Ausfallraten λ.

Die Ausfallrate hat typisch eine Badewannenkurve und unterscheidet:

- Frühausfälle: Zu Beginn der Lebensdauer fallen «kränkliche« Bauteile mit Beinahefehlern aus. Abnahme λ mit der Lebensdauer.
- Zufallsausfälle: Wenn alle Bauteile mit Beinahefehlern ausgefallen sind, λ über lange Zeit konstant.
- Ermüdungsausfälle: Nach langer Nutzungsdauer häufen sich die Ausfälle durch normalen Verschleiß (Materialermüdung). Zunahme λ mit t.

6.128 Hauptnutzungsphase, Voralterung, ...

Nach den Frühausfällen und vor den Verschleißausfällen ist die Ausfallrate am geringsten und konstant. Hauptnutzungsphase. Überlebenswahrscheinlichkeit, Verteilung und Erwartungswert der Lebensdauer:

$$V(t) = \mathrm{e}^{-\lambda \cdot t} \tag{6.2}$$

$$F_L(t) = 1 - \mathrm{e}^{-\lambda \cdot t} \tag{6.3}$$

$$\bar{t}_{\mathrm{FL}} = \mu_{\mathrm{L}} = \int_0^\infty t \cdot \lambda \cdot \mathrm{e}^{-\lambda \cdot t} \cdot \mathrm{d}t = \tfrac{1}{\lambda} \tag{6.4}$$

Für ein System auf mehreren notwendigen Komponenten ist die gesammte Ausfallrate die Summe der Ausfallraten aller Komponenten:

$$\lambda_{\mathrm{Sys}} = \sum_{i=1}^{\#Prt} \lambda_i \tag{6.5}$$

Vermeidung der Phasen erhöhter Ausfallrate:

- Voralterung: Betrieb unter Stress, damit »kränkliche« Bauteile mit Beinahefehler vor der Nutzung ausfallen und ersetzt werden.
- Rechtzeitige Einplanung Ersatz Verschleißteile bzw. Gesamtsystem zur Ausfallvermeidung durch Materialermüdung.

6.129 Ausfallschwere und Gegenmaßnahmen

Schwer: Ersatz vor der weiteren Nutzung. Hardware-Verfügbarkeit:

$$A_{\mathrm{H}} = 1 - \lambda \cdot \bar{t}_{\mathrm{R}} \tag{6.6}$$

Gegenmaßnahmen:

- kurze Reparaturzeiten,
- Redundanzen für sofortige Service-Übernahme.

Toleriert: Service von Redundanzen übernommen. Bis zur Reparatur erhöhtes Risiko für schwere Ausfälle. Gegenmaßnahmen:

- schnellere Fehlerbeseitigung.

Leicht: Verringerte Zuverlässigkeit bis zur nächsten Wartung:

$$\tfrac{1}{R_{\mathrm{FW}}} = \bar{\gamma}_{\mathrm{FW}} = \tfrac{\lambda_{\mathrm{WF}} \cdot \tau \cdot \bar{\gamma}_{\mathrm{WF1}}}{2} \tag{6.8}$$

Gegenmaßnahmen

- verbesserte Überwachung zur schnellern Ausfallerkennung,
- kürzere Wartungsintervalle.

Vorhersagbar: Messbare Anzeichen für bevorstehende Ausfälle. Gegenmaßnahmen

- Gesundheitsüberwachung und Wartungstests auf solche Indikatorenen.
- Prophylaktischer Ersatz bei der nächsten Wartung.
- Funktionseinschränkung zur Vermeidung erhöhter Sicherheitsrisiken.

Verborgen: Unbemerkte Zuverlässigkeitsminderung:

$$\tfrac{1}{R_{\mathrm{FH}}} = \bar{\gamma}_{\mathrm{FH}} = \lambda_{\mathrm{WH}} \cdot t_{\mathrm{OP}} \cdot \bar{\gamma}_{\mathrm{WH1}} \tag{6.9}$$

Gegenmaßnahmen

- bessere Wartungstests ,
- prophylaktischer Ersatz Gesamtsystem.

6 Redundanz

6.1 Kalt, heiß, warm

6.131 Reserveeinheiten und Ausfalltoleranz

Reserve- oder redundante Einheiten sind Komponenten,

- die für die Funktion nicht erforderlich sind und
- nach Ausfall Aufgaben ausgefallener Komponenten übernehmen.

Reserveeinheiten sind erforderlich für

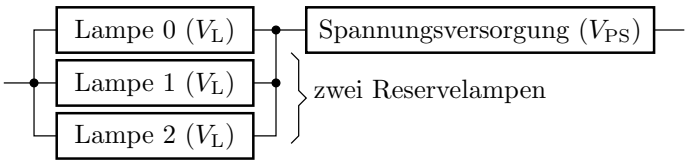
- Systeme ohne Reparaturmöglichkeit, die lange verfügbar sein müssen (z.B. Satelliten) und
- bei hoher geforderter Verfügbarkeit (z.B. für Serverdienste) zur Überbrückung von Ausfallzeiten der Hauptkomponenten.

Ausfalltoleranz*: Aufrechterhaltung der Funktion nach Komponentenausfall. Mögliche Maßnahmen außer Ersatz durch Reserveeinheiten:

- Aufgabenübernahme durch andere Komponanten,
- Notbetrieb mit reduzierter Funktionalität und Leistung, ...

* In der Literatur wird die Tolerierung von Ausfällen, Störungen, Fertigungs- und Entwurfsfehlern in der Regel ohne Unterscheidung nach dem zu tolerierenden Problem als Fehlertoleranz bezeichnet.

6.132 Verfügbarkeitsplan mit Reserveeinheiten



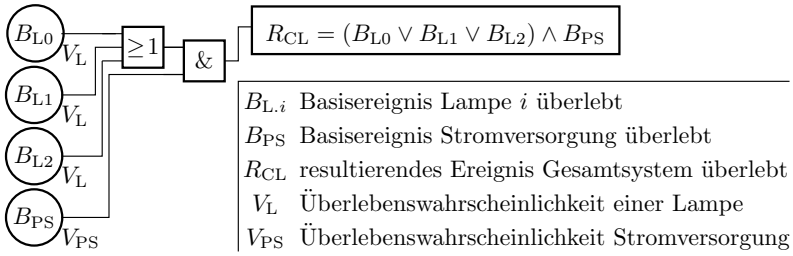
Im Verfügbarkeitsplan werden

- notwendige Komponenten als Reihenschaltung und
- Reserveeinheiten (Redundanzen) als Parallelschaltung dargestellt.

Eine Flurbeleuchtung sei verfügbar, wenn mindestens eine von drei Lampen und die Spannungsversorgung funktionieren. Parallelschaltungen beschreiben eine ODER- und Reihenschaltungen eine UND-Verknüpfung der Überlebenswahrscheinlichkeiten.

$V_L(t)$ Überlebenswahrscheinlichkeit einer einzelnen Flurbeleuchtungslampe.
$V_{PS}(t)$ Überlebenswahrscheinlichkeit der Spannungsversorgung.
$V_{CL}(t)$ Überlebenswahrscheinlichkeit der gesamten Flurbeleuchtung.

6.133 Fehlerbaum zum Verfügbarkeitsplan



In der gleichwertigen Darstellung als Fehlerbaum ist das Überleben jeder Komponente eine Basisereignis. Regeln für UND und ODER unabhängiger Ereignisse:

$$\mathbb{P}[A \wedge B] = \mathbb{P}[A] \cdot \mathbb{P}[B] \tag{3.5}$$

$$\mathbb{P}[A \vee B \vee C] = 1 - (1 - \mathbb{P}[A]) \cdot (1 - \mathbb{P}[B]) \cdot (1 - \mathbb{P}[C]) \tag{3.9}$$

Überlebenswahrscheinlichkeit der Flurbeleuchtung:

$$V_{\mathrm{CL}}(t) = \left(1 - (1 - V_{\mathrm{L}}(t))^3\right) \cdot V_{\mathrm{PS}}(t)$$

6.134 Kalte, warme und heiße Reserve

- Heiße Reserve*: Reservekomponenten arbeiten parallel (z.B. die redundante Platte in einem RAID5) und fallen mit derselben Wahrscheinlichkeit aus wie das aktive System.
- Kalte Reserve: Reservekomponenten werden geschont und funktionieren idealerweise noch alle zum Ausfallzeitpunkt der aktiven Komponente.
- Warme Reserve: Reserveeinheiten (z.B. das Reserverad im Auto) altern auch bei Nichtnutzung, nur langsamer.

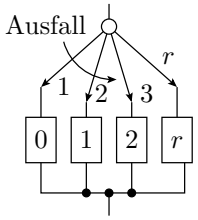
Die beiden zusätzlichen Lampen auf der Folie zuvor, die für die Verfügbarkeit der Treppenbeleuchtung nicht unbedingt funktionieren müssen, bilden eine heiße Reserve, Ersatzlampen, die erst nach Ausfall der »Hauptlampe« eingeschraubt werden, eine kalte Reserve, ein Ersatzrad im Auto eine warme Reserve.

* Abgeleitet von Glühlampen. Reihenschaltung fällt mit der ersten Glühlampe aus, bei Parallelschaltung Komplettausfall der Beleuchtung erst mit der letzte Glühlampe.

6.135 Zu erwartende Lebensdauer kalte Reserve

Für jede Komponente beginnt die Belastung erst nach Ausfall der vorherigen Komponente.

Phase	mittlere Lebensdauer
1	$\mu_{\mathrm{LC}.0}$
2	$\mu_{\mathrm{LC}.1}$
3	$\mu_{\mathrm{LC}.2}$
...	...



Die zu erwartenden Lebensdauern aller Komponenten addieren sich*:

$$\mu_{\mathrm{LS}} = \sum_{i=0}^{r} \mu_{\mathrm{LC}.i} \tag{6.10}$$

Mir r gleichen Reserveeinheiten (plus Komponente 0) erhöht sich die zu erwartende Lebensdauer auf das $r+1$ fache.

μ_{LS} Zu erwartende Lebensdauer des Systems.
r Anzahl der Reserveeinheiten.
$\mu_{\mathrm{LC}.i}$ Zu erwartende Lebensdauer Komponente i.
* Annahme, dass Umschalter und ungenutzten Reserveeinheiten Ausfallrate null haben.

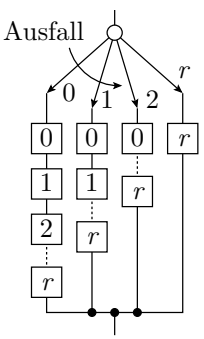
6.136 Zu erwartende Lebensdauer heiße Reserve

Alle noch lebenden Komponenten können gleichermaßen ausfallen:

$$\mu_{\mathrm{L}.i} = \frac{1}{\sum_{j=1}^{i+1} \lambda_j}$$

Bei gleicher Lebensdauer μ_{LC} aller Komponenten:

defekte Komponenten	mittlere Phasendauer
0	$\mu_{\mathrm{L}.1} = \mu_{\mathrm{LC}}/(r+1)$
1	$\mu_{\mathrm{L}.2} = \mu_{\mathrm{LC}}/r$
...	...



Zu erwartende Gesamtlebensdauer:

$$\mu_{\mathrm{LS}} = \mu_{\mathrm{LC}} \cdot \sum_{i=0}^{r} \frac{1}{i+1} \tag{6.11}$$

$\mu_{\mathrm{L}.i}$, λ_j Zu erwartende Lebensdauer Komponente i, Ausfallrate noch lebende Komponente j.
μ_{LC}, μ_{LS} Zu erwartende Lebensdauer der Komponenten, zu erwartende Lebensdauer Systems.

r Anzahl der Reserveeinheiten.

Beispiel 6.1 Dreifache Lebensdauer

Wie viele Reserveeinheit werden für eine Verdreifachung der zu erwartendene Lebensdauer benötigt?

a) *bei kalter Reserve?*

(6.10) $\mu_{\mathrm{LS}} = \sum_{i=0}^{r} \mu_{\mathrm{LC}.i}$

Insgesamt drei, also zwei Ersatzkomponenten.

b) *bei heißer Reserve?*

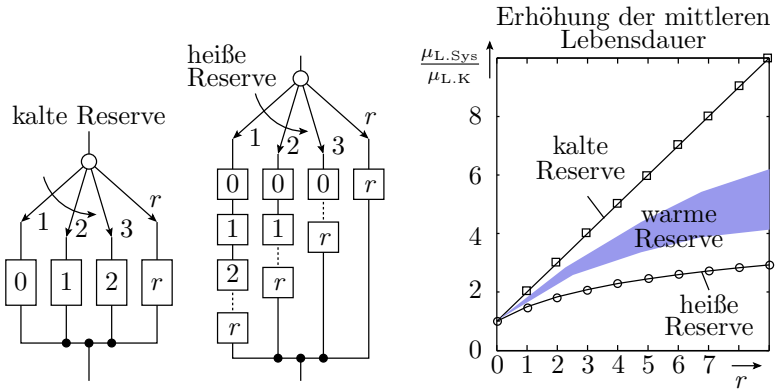
(6.11) $\mu_{\mathrm{LS}} = \mu_{\mathrm{LC}} \cdot \sum_{i=0}^{r} \frac{1}{i+1}$

r	1	2	3	4	5	6	7	8	9	10
$1/(r+1)$	0,5	0,33	0,250	0,20	0,17	0,14	0,13	0,11	0,10	0,09
$\sum_{i=0}^{r} \frac{1}{i+1}$	1,5	1,83	2,08	2,28	2,45	2,59	2,72	2,83	2,93	3,02

Insgesamt 11, also 10 Ersatzkomponenten.

μ_{LC}, μ_{LS} Zu erwartende Lebensdauer der Komponenten, zu erwartende Lebensdauer Systems.
r Anzahl der Reserveeinheiten.

6.138 Warme Reserve



- Die Ausfallrate der »kalten« Ersatzkomponenten ist kleiner als im aktiven Zustand, aber größer null.
- »Warme« Reserveeinheiten verlängert die Lebensdauer mehr als »heiße« und weniger als »kalte«.

6.2 KOON-Strukturen

6.139 KooN-Systeme (k out of n)

Systeme aus n gleichartigen Komponenten mit Verfügbarkeit A, von denen mindestens k verfügbar sein müssen. Bekanntes A impliziert heiße Reserve mit Wartung. Für kalter Reserve Worst-Case-Abschätzung.

Anwendungen:

- Standby-Reserve im Maschinenbau (1oo2),
- Systeme ohne kurzzeitig erreichbaren sicheren Zustand, z. B. Flugzeuge, Atomkraftwerke, Chemiereaktoren (auch 2oo3, 2oo4),
- Ausfalltolerante Massenspeicher, Server-Cluster und Verbindungsnetzwerke, (RAID, $(n-1)$oo(n)), ...

KooN-Systeme tolerieren fast alle unabhängigen Ausfälle. Ihre größte Schwachstelle sind die nie ganz ausschließbaren Risiken einer gemeinsamen Ursache für den gleichzeitigen Ausfall aller Komponenten (common-cause failure):

- Sabotage,
- Feuer, Blitzschlag, ...

6.140 Unabhängige und gemeinsame Ausfälle

Zur Vereinfachung werden nur Systeme aus n gleichen Komponenten mit derselben Verfügbarkeit $A = p$, Gegenwahrscheinlichkeit q betrachtet, die entweder unabhängig voneinander oder gemeinsam ausfallen. Für unabhängige Ausfälle ist die Anzahl der verfügbaren identischen Komponenten binomialverteilt

(4.30) $$\mathbb{P}[X=k] = \binom{n}{k} \cdot p^k \cdot (1-p)^{n-k}$$

Für Common-Cause-Ausfälle sind mit Wahrscheinlichkeit p alle und mit Wahrscheinlichkeit $q = 1-p$ null Komponenten verfügbar. Insgesammt hat die Anzahl der ausgefallenen Komponenten die Mischverteilung:

$$\mathbb{P}[X=k] = \left(h_{\mathrm{CC}} \cdot \begin{cases} q & k=0 \\ p & k=n \\ 0 & \text{sonst} \end{cases} \right) + (1-h_{\mathrm{CC}}) \cdot \binom{n}{k} \cdot p^k \cdot q^{n-k}$$

n, k Anzahl aller Komponenten, min. Anzahl der erforderlichen Komponenten.
h_{CC} Bedingte Wahrscheinlichkeit für gemeinsame Ursache wenn Ausfall.
p, q Wahrscheinlichkeit, Komponente verfügbar, Gegenwahrscheinlichkeit $1-p$.

Verfügbarkeit von mindestens $k > 0$ der n Komponenten:

$$\begin{aligned} A_{koon} &= h_{\mathrm{CC}} \cdot p + (1-h_{\mathrm{CC}}) \cdot \sum_{i=k}^{n} \binom{n}{i} \cdot p^i \cdot q^{n-i} \\ &= h_{\mathrm{CC}} \cdot (1-q) + (1-h_{\mathrm{CC}}) \cdot \left(1 - \sum_{i=0}^{k-1} \binom{n}{i} \cdot p^i \cdot q^{n-i} \right) \end{aligned} \tag{6.12}$$

KooN-Systeme verlangen eine sehr hohe Verfügbarkeit der einzelnen Komponenten ($p \to 1$, $q \to 0$) und geringes Common-Cause-Risiko ($h_{\mathrm{CC}} \to 0$). Erlaubt Vereinfachung*:

$$\sum_{i=0}^{k-1} \binom{n}{i} \cdot p^i \cdot q^{n-i} = q^{n-(k-1)} \cdot \underbrace{\sum_{i=0}^{k-1} \binom{n}{i} \cdot p^i \cdot q^{k-1-i}}_{\approx \binom{n}{k-1} \cdot p^{k-1} \cdot q^0 = \binom{n}{k-1}}$$

$$A_{koon} \approx h_{\mathrm{CC}} \cdot (1-q) + (1-h_{\mathrm{CC}}) \cdot \left(1 - \binom{n}{k-1} \cdot q^{n-(k-1)} \right)$$

$$A_{koon} \approx 1 - h_{\mathrm{CC}} \cdot q - \binom{n}{k-1} \cdot q^{n-(k-1)} \tag{6.13}$$

h_{CC} Bedingte Wahrscheinlichkeit für gemeinsame Ursache wenn Ausfall.
p, q Wahrscheinlichkeit, Komponente verfügbar, Gegenwahrscheinlichkeit $1-p$.

* $A_{koon} \approx \cancel{h_{CC}} - h_{CC} \cdot q + 1 - \cancel{h_{CC}} - (1 - \cancel{h_{CC}}) \cdot \binom{n}{k-1} \cdot q^{n-(k-1)}$.

6.142 Beispielkonfigurationen

(6.13) $$A_{koon} \approx 1 - h_{CC} \cdot q - \binom{n}{k-1} \cdot q^{n-(k-1)}$$

System aus n Komponenten, von denen mindesten eine benötigt wird:

$$A_{1oon} = 1 - h_{CC} \cdot q - \binom{n}{0} \cdot q^n$$
$$A_{1oon} = 1 - h_{CC} \cdot q - q^n \tag{6.14}$$

Nichtverfügbarkeit q je Komponente als Gegenwahrsch. zu (Gl. 6.6):

$$q = \lambda \cdot \bar{t}_R \tag{6.15}$$

Für $q^{n-1} \ll h_{CC}$ kann q^n gegenüber $h_{CC} \cdot q$ vernachlässigt werden:

$$A_{1oon} = 1 - h_{CC} \cdot \lambda \cdot \bar{t}_R \tag{6.16}$$

Bei hoch verfügbaren Einzelsystemen lohnt nicht mehr als eine Reserveeinheit.

A_{koon} Verfügbarkeit von min. k der n Komponenten.
p, q Wahrscheinlichkeit, Komponente verfügbar, Gegenwahrscheinlichkeit $1-p$.
h_{CC} Bedingte Wahrscheinlichkeit für gemeinsame Ursache wenn Ausfall.
$h_{CC} \cdot q$ Absolute Wahrscheinlichkeit für gleichzeitigen Ausfall durch gemeinsame Ursachen.

6.143 Nur eine Reserveeinheit

(6.13) $$A_{koon} \approx 1 - h_{CC} \cdot q - \binom{n}{k-1} \cdot q^{n-(k-1)}$$

Eine Reserveeinheit bedeutet $k = n - 1$ notwendige Einheiten. Mit

$$\binom{n}{n-1-1} = \binom{n}{2}$$

erhöht eine Reserveeinheit bei hohe Komponentenverfügbarkeit und geringem Common-Cause-Risiko die Gesamtverfügbarkeit auf:

$$A_{(n-1)oo(n)} = 1 - h_{CC} \cdot q - \binom{n}{2} \cdot q^2 \tag{6.17}$$
$$A_{(n-1)oo(n)} = 1 - h_{CC} \cdot \lambda \cdot \bar{t}_R - \binom{n}{2} \cdot (\lambda \cdot \bar{t}_R)^2 \tag{6.18}$$

Für $q \ll h_{CC}/\binom{n}{2}$ kann der Term $\binom{n}{2} \cdot q^2$ vernachlässigt werden und die Gesamtverfügbarkeit ist wieder eins abzüglich der Wahrscheinlichkeit für Common-Cause-Ausfälle:

$$A_{(n-1)oo(n)} = 1 - h_{CC} \cdot \lambda \cdot \bar{t}_R \tag{6.19}$$

A_{koon} Verfügbarkeit von min. k der n Komponenten.
p, q Wahrscheinlichkeit, Komponente verfügbar, Gegenwahrscheinlichkeit $1-p$.
h_{CC} Bedingte Wahrscheinlichkeit für gemeinsame Ursache wenn Ausfall.
$h_{CC} \cdot q$ Absolute Wahrscheinlichkeit für gleichzeitigen Ausfall durch gemeinsame Ursachen.

6.144 1oo2-Standby-Reserve

Bei Ausfall des Hauptsystems, z.B. eine Anlage mit hohen Stillstandskosten, steht eine Reserveanlage für den vollwertigen oder einen Notbetrieb bereit. Unter der Annahme gleicher Verfügbarkeit* 1oo2-System:

$$A_{1oo2} = 1 - h_{CC} \cdot \lambda \cdot \bar{t}_R - (\lambda \cdot \bar{t}_R)^2 \tag{6.20}$$

Zulässige Reparaturdauer, damit der quadratische Term entfällt:

$$\bar{t}_R \ll \frac{h_{CC}}{\lambda}$$

Standby-Reserve muss vor allem vor Common-Cause-Ausfällen geschützt werden, Zerstörung durch Sabotage, Feuer, ...

A_{koon} Verfügbarkeit von min. k der n Komponenten.
λ, $\bar{t}_R$ Ausfallrate für schwere Ausfälle, mittlere Reparaturdauer.
h_{CC} Bedingte Wahrscheinlichkeit für gemeinsame Ursache wenn Ausfall.
$h_{CC} \cdot q$ Absolute Wahrscheinlichkeit für gleichzeitigen Ausfall durch gemeinsame Ursachen.
* Gleiche Verfügbarkeit impliziert heiße Reserve mit Wartung. Standby bedeutet jedoch in der Regel kalte oder warme Reserve. In dem Fall Worst-Case-Abschätzung.

6.145 Mehrheitseinscheid mit MS-Notbetrieb

(6.13) $$A_{koon} \approx 1 - h_{CC} \cdot q - \binom{n}{k-1} \cdot q^{n-(k-1)}$$

Dreiversionssysteme wurden erstmalig von Von-Neumann zur Erhöhung der Hardware-Verfügbarkeit vorgeschlagen. Heute Einsatz in Systemen, die nicht in kurzer Zeit einen sichereren Zustand hergestellen können (Flugzeuge, Atomkraftwerke, ...). Verfügbarkeit mit Mehrheitsentscheid (3oo3):

$$A_{3oo3} = 1 - h_{CC} \cdot q - \binom{3}{2} \cdot q \overset{h_{CC} \ll 3}{=} 1 - 3 \cdot \lambda \cdot \bar{t}_R \tag{6.21}$$

Notbetrieb als Master-Checker-Paar:

$$A_{2oo3} = 1 - h_{CC} \cdot q - \binom{3}{1} \cdot q^2 = 1 - h_{CC} \cdot \lambda \cdot \bar{t}_R - 3 \cdot (\lambda \cdot \bar{t}_R)^2 \tag{6.22}$$

Für hoch verfügbare Einzelsysteme. Gesamtverfügbarkeit mindestens als Master-Checker-Paar wieder eins abzüglich der Wahrscheinlichkeit für Common-Cause-Ausfälle.

A_{koon} Verfügbarkeit von min. k der n Komponenten.
h_{CC} Bedingte Wahrscheinlichkeit für gemeinsame Ursache wenn Ausfall.
p, q Wahrscheinlichkeit, Komponente verfügbar, Gegenwahrscheinlichkeit $1 - p$.
λ, $\bar{t}_R$ Ausfallrate für schwere Ausfälle, mittlere Reparaturdauer.

6.146 3oo4 mit Mehrheitseinscheid

(6.13) $$A_{koon} \approx 1 - h_{CC} \cdot q - \binom{n}{k-1} \cdot q^{n-(k-1)}$$

Für den Mehrheitseinscheid genügen $k = 3$ der $n = 4$ Einzelsysteme. Verfügbarkeit:

$$A_{3oo4} = 1 - h_{CC} \cdot q + (1 - h_{CC}) \cdot \left(1 - \binom{4}{2} \cdot q^2\right)$$

Mit der Nichtverfügbarkeit

(6.15) $$q = \lambda \cdot \bar{t}_R$$

$$A_{3oo4} = 1 - h_{CC} \cdot \lambda \cdot \bar{t}_R - 6 \cdot (\lambda \cdot \bar{t}_R)^2$$

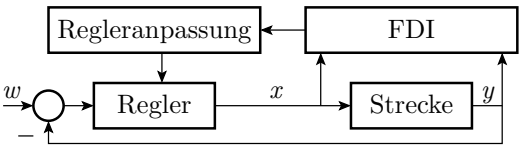
Zulässige Reparaturdauer, damit der quadratische Term entfällt:

$$\bar{t}_R \ll \frac{h_{CC}}{6 \cdot \lambda}$$

Komplette Verfügbarkeit mit Mehrheitsentscheid wieder eins abzüglich der Wahrscheinlichkeit für Common-Cause-Ausfälle (Sabotage, Naturkatastrophen, Feuer, ...).

6.3 Spezielle Lösungen

6.147 Fehlertolerantes Regelungssystem



In einem Reglersystem wird vom Sollwert w der zu regelnde Ist-Wert y abgezogen. Aus der Differenz bildet der Regler den Stellwert x für die Regelstrecke (z.B. eine Heizung, wenn y eine Temperatur ist). Gefährlichste Fehlfunktionen:

- unzulässige Stellwerte für die Strecke und
- Schwingungen.

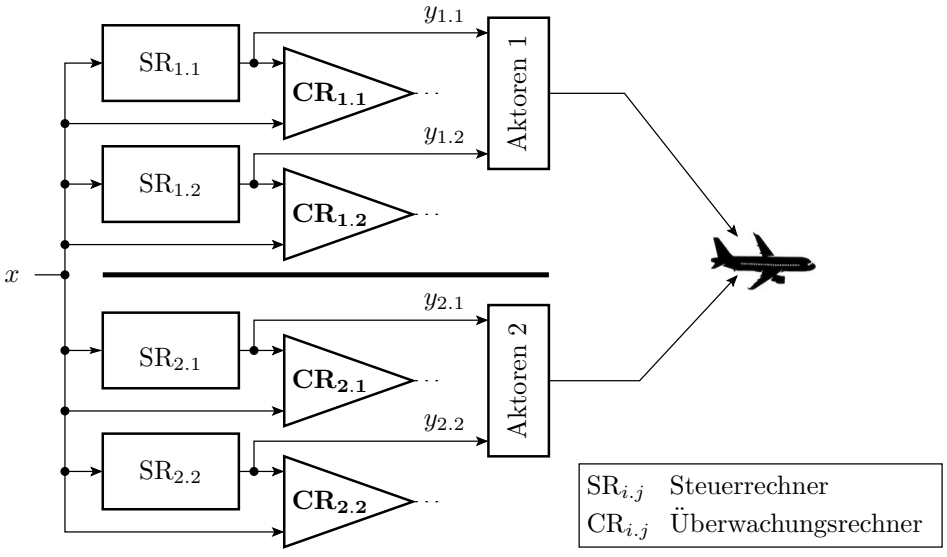
Hinzufügen einer zusätzlichen Überwachungs- und Fehlerbehandlungsschicht (FDI) mit den Aufgaben:

- Überwachung der Werte (-verläufe). Wenn Fehlerzustand:
- Fehlerdiagnose (Abschätzung von Fehlerursache und -ort) und
- Anpassung der Regelung so, dass eine Mindestfunktionalität gewährleistet bleibt.

Toleriert auch Fehler und Störung auch an der Strecke, ...

FDI Fehlerdetektion, -isolation und -identifikation.

6.148 Flugsteuersystem Airbus A3XX [1]



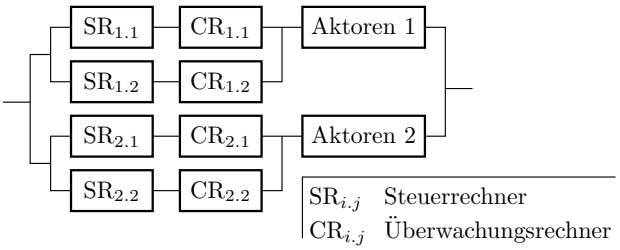
Beispiel für extremen Aufwand für max. Sicherheit: Zwei identische Systeme mit Sensoren, Aktoren und zwei Master-Checker-Paaren.

- Jedes Rechnerpaar besteht aus einem Steuerrechner $SR_{i.j}$, der die Aktoren ansteuert, und einem Überwachungsrechner $CR_{i.j}$.
- Normalzustand Rechner $SR_{1.1}$ steuert und $CR_{1.1}$ überwacht. Zweites Rechnerpaar Stand-By. System 2 abgeschaltet.

- Bei Ausfall von Rechnerpaar 1 übernimmt Rechnerpaar 2. Bei Ausfall des kompletten ersten Stems oder dessen Aktoren übernimmt System 2.

Zur zusätzlichen Tolerierung auch von Fehlfunktionen durch Entwurfsfehler werden Rechner von unterschiedlichen Herstellern verwendet mit getrennt entwickelter Software nach Spezifikationen, die unabhängig voneinander von einer gemeinsamen Basisspezifikation abgeleitet sind.

6.150 Verfügbarkeitsplan Airbus A3XX



Im Verfügbarkeitsplan bilden

- die Master- und Checker-Rechern Reihenschaltungen,
- parallel zum jeweils anderen Paar des eigenen Teilsystems,
- in Reihe zu den Aktoren und
- beide Teilsysteme parallel.

Unterschiedliche Komponenten unterscheiden sich im Ausfallverhalten und Redundanzbedarf. Die vielen Rechner und Checker dienen vor allem auch zur Tolerierung von Fehlfunktionen durch Programmierfehler.

6.4 RAID und Backup

6.151 RAID

Organisation mehrerer physischer Massenspeicher (Festplattenlaufwerke oder SSD) zu einem logischen Laufwerk, ab RAID 1, insbesonder auch zum Schutz vor Datenverlust bei Plattenausfällen.

Wichtige RAID-Techniken:

- Verteilung der Daten auf mehrere Platten mit Parallelzugriff (RAID 0). Erhöhung der Schreib- und Lesegeschwindigkeit. Keine Datenschutz.
- Gespiegelte Platten (RAID 1). Datenverlust erst nach Ausfall der letzten Platte (1ooN).
- Fehlerkorrigiernder Hamming-Code (RAID 2): Tolerierung Einzelbitverfälschungen je Datenwort oder Festplattenausfall.
- Paritätsblöcke auf getrennten Festplatten, $(n-1)\,\mathrm{oo}\,(n)$ zur Tolerierung eines Plattenausfalls (RAID 5), ... eines zweiten Plattenausfalls vor der Datenwiederherstellung (RAID 6).

RAID Redundantes Array unabhängiger (urspünglich kostengünstiger) Festplatten.
SSD Solid State Drive, Festplattennachbildung mit Halbleiterspeichern.

6.152 RAID mit Paritätsblöcken (RAID 2 bis 7)

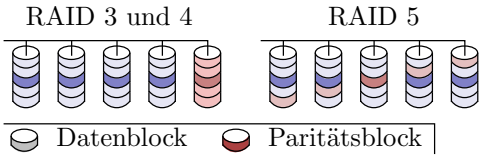
	Paritätsbildung	Ausfall Disc 2	Ausfall Disc 4
Disc 1: Daten	0001 0011 0010	0001 0011 0010	0001 0011 0010
Disc 2: Daten	1101 0101 0000	xxxx xxxx xxxx	1101 0101 0000
Disc 3: Daten	1101 1111 1100	1101 1111 1100	1101 1111 1100
Disc 4: Parität	**0001 1001 1110**	0001 1001 1110	xxxx xxxx xxxx
	Wiederherstellung:	**1101 0101 0000**	**0001 1001 1110**

Datenverteilung blockweise über mehrere Platten. Bildung und Speicherung von Paritätsblöcken so, dass

- nur Blöcke unterschiedlicher Platten zusammengefasst werden,
- der Paritätsblock auf noch einer anderen Platte gespeichert wird.

Nach Ausfall einer Platte lassen sich alle Blöcke auf der ausgefallenen Platte aus denen auf anderen Platten gespeicherten Blöcken durch bitweise EXOR rekonstruieren.

6.153 Verteilung Paritätsblöcke auf alle Platten



Bei jeder Schreiboperation auf einer Platte muss die Parität korrigiert werden. Wenn alle Paritätsblöcke auf derselben Festplatte stehen (RAID 3 und 4), erfolgen auf diese viel mehr Zugriffe und erhöhen deren Ausfallrate. Zu Sicherstellung gleicher Ausfallrate, gleichmäßige Verteilung der Paritätsblock auf alle Platten (RAID 5).

Ein RAID 5 ist ein »$n-1$ out of n«–System aus gleichen Komponenten (Festplatten oder SSDs) mit gleichen Ausfallraten:

(6.18) $$A_{(n-1)\text{oo}(n)} = 1 - h_{\text{CC}} \cdot \lambda \cdot \bar{t}_{\text{R}} - \binom{n}{2} \cdot (\lambda \cdot \bar{t}_{\text{R}})^2$$

Die Verfügbarkeit wird maßgeblich durch die Reparaturdauer $\bar{t}_{\text{R}}$, d.h. der Zeit für Plattenwechsel und Datenrekonstruktion, bestimmt.

6.154 Reparaturdauer, höhere Redundanz

Verkürzung der mittleren Reparaturdauer:

- Hot-Fix: eingebaute Reserveplatte, die bei Ausfall für die ausgefallene Platte in das RAID eingebunden wird.
- Hot-Swap: Plattenaustausch und Rebuilt im Betrieb.

Für Massenspeichern ab Terra-Byte-Bereich dauert die Datenwiederherstellung (Rebuilt), enthalten in der mittleren Reparaturzeit $\bar{t}_{\text{R}}$ in

(6.18) $$A_{(n-1)\text{oo}(n)} = 1 - h_{\text{CC}} \cdot \lambda \cdot \bar{t}_{\text{R}} - \binom{n}{2} \cdot (\lambda \cdot \bar{t}_{\text{R}})^2$$

so lange, dass Zufallsausfälle nicht mehr vernachlässigbar sind. Erhöhung der Redundanz auf $(n-2)$ out of n durch weitere Zusatzplatten und Paritätsblöcke.

6.155 Backup

Neben HW-Ausfällen gibt es weitere Ursachen für den Verlust schwer wiederzubeschaffender Daten. Auftrittshäufigkeiten:

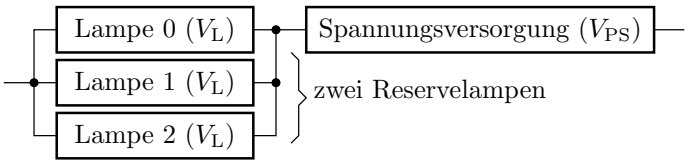
- 59% Hardwareprobleme
- 26% Anwenderfehler
- 9% Softwarefehler
- 2% Schadware (Vieren, ...)
- 2% Naturkatstrophen
- 2% sonstiges.

Selbst ausgefeilte RAIDs tolerieren nur HW-Probleme. Den einzig wirklich zuverlässigen Schutz gegen Datenverluste bieteten konsequent geplante und durchgeführte Backups.

Schadware wird oft erst längere Zeit nach der Infizierung erkannt. Typisch Backup-Aufbewahrung über ein Jahr. Abstandszunahme auf der Zeitachse zurück.

Zusammenfassung

6.156 Reserveeinheiten



Reserveeinheiten übernehmen nach Ausfall die Aufgaben ausgefallener Komponenten. Im Verfügbarkeitsplan bilden notwendige Komponenten Reihenschaltungen und Reservekomponenten Parallelschaltung.

Das gesamte System überlebt, solage

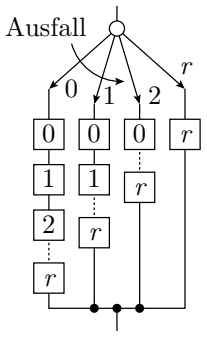
- von allen parallelen Komponenten mindestens eine überlebt (ODER*) und
- von allen Komponenten in Reihe alle überleben (UND*)
- ODER-Verknüpfung der Überlebenswahrscheinlichkeiten.

* der Überlebensereignisse.

6.157 Erhöhung der Lebensdauer

- Kalte Reserve: Belastungsbeginn erst bei Ausfall der zu ersetzenden Komponente. Addition der zu erwartenden Lebensdauern:

(6.10) $$\mu_{LS} = \sum_{i=0}^{r} \mu_{LC.i}$$



- Heiße Reserve: Mittlere Lebensdauer mit i verfügbaren Komponenten $\mu_{L.i} = \mu_{LC}/i$. Erhöhung der erwartende Gesamtlebensdauer mit der ersten Ersatzkomponente $1/2$ mit der zweiten $1/3$ etc.:

(6.11) $$\mu_{LS} = \mu_{LC} \cdot \sum_{i=0}^{r} \frac{1}{i+1}$$

- Warme Reserve verlängert die Lebensdauer mehr als heiße und weniger als kalte Reserve.

6.158 KooN-Systeme

Systeme aus n gleichartigen Komponenten, von denen k verfügbar sein müssen und ausgefallene Systeme repariert werden. Verfügbarkeit:

(6.13) $$A_{koon} \approx 1 - h_{CC} \cdot q - \binom{n}{k-1} \cdot q^{n-(k-1)}$$

Bei kleiner Nichtverfügbarkeit q der Komponenten genügt eine redundanten Komponente:

(6.17) $$A_{(n-1)oo(n)} = 1 - h_{CC} \cdot q - \binom{n}{2} \cdot q^2$$

Nichtverfügbarkeit ist proportional zur mittleren Reparaturdauer $\bar{t}_R$:

(6.15) $$q = \lambda \cdot \bar{t}_R$$

Bei hinreichen
praktisch nur

(6.19) $$A_{(n-1)oo(n)} = 1 - h_{CC} \cdot \lambda \cdot \bar{t}_R$$

KooN ist ein Modell für einfache Überschläge. Hauptproblem Common-Cause-Ausfälle. Bei realen Systemen unterscheiden sich Ausfallrate und Redundanzbedarf der Komponenten. Auch Mitnutzung Redundanzen zur Tolerierung von Software-Fehler, ...

6.159 RAID und Backup

RAIDs sind logisches Laufwerk aus mehreren physischen Festplatten oder SSDs, die ab RAID1 mindestens Einzelplattenausfälle tolerieren. Das erfordert im einfachsten Fall, dass alle Bits eines Datenworts und das Paritätsbit je Datenwort auf einer anderen Platte stehen.

Selbst ausgefeilte RAID-Technilen tolerieren nur HW-Probleme. Wirklich zuverlässigen Schutz gegen Datenverluste bieteten Backups, die ausreichend gegen Fehlfunktionen des Systems, Bedinfehler, Angriffe, Katastrophen, ... geschützt sind. Ausreichend lange Aufbewahrung, ...

6.160 Literatur

Literatur

4:134–152, 1991.

[1] Pascal Traverse. Dependability of digital computers on board airplanes. *Dependable Computing for critical applications*,