



Test und Verlässlichkeit

Foliensatz 5: Hardware-Test und Selbsttest.

Prof. G. Kemnitz

Institut für Informatik, TU Clausthal (TV_F5)

6. Februar 2024



Inhalt Foliensatz 5

Fehlermodellierung

- 1.1 Schaltkreisfehler
- 1.2 Lokale Fehler
- 1.3 FM für DIC
- 1.4 Nachweisbeziehungen

Testsuche

- 2.1 Fehlersimulation
- 2.2 D-Algorithmus
- 2.3 Implikationstest

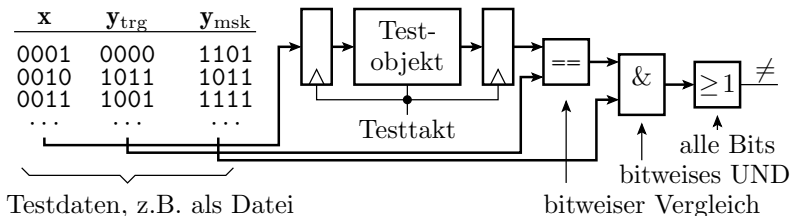
- 2.4 Suchraumstrukturierung
- 2.5 Komplexe Funktionsbausteine
- 2.6 Sequentielle Schaltungen
- 2.7 Speichertest

Selbsttest

- 3.1 Pseudo-Zufallsregister
- 3.2 Signaturregister
- 3.3 Selbsttest mit LFSR
- 3.4 Fehlerorientierte Wichtung
- 3.5 RAM-Selbsttest

Vorlesung	15	16	17
ca. ab Folie	3	51	86

Test digitaler Schaltungen



Die Hardware von IT-Systemen steckt überwiegend in hochintegrierten digitalen Schaltkreisen und enthält unzählige Fehlermöglichkeiten.

Nachweis durch sehr viele dynamische Tests der Form:

- Bereitstellung logischer Eingabewerte und
- Abtasten und Auswertung der Ausgaben.

x	Testeingaben.
y	Testausgaben.
y _{msk}	Maskenwerte zum Ausschluss von Testausgaben vom Soll-Ist-Vergleich.
≠	Vergleichsfehler.

Test mit allen Eingaben unmöglich

Für den Nachweis, dass ein Service mit m Eingabebits für alle Eingabedaten korrekte Ergebnisse liefert, müsste er mindestens mit alle 2^m Kombinationen der Eingabewerte ausprobiert werden. Ab $m \geq 40 \dots 50$ unmöglich:

	w	2^w	t_{test}
Gatter, 4 Eingänge	4	16	16 μs
ALU, 68 Eingänge	68	$3 \cdot 10^{20}$	10^7 Jahre
vier Eingabevariablen vom Typ int32_t	128	$3 \cdot 10^{38}$	10^{25} Jahre

- Die meisten Systeme verarbeiten $w \gg 100$ Eingabebits.
- Hinzu kommen oft tausende oder mehr gespeicherte Bits, die auch mit variiert werden müssten.
- Geschätzte Zeit seit dem Urknall $14 \cdot 10^9$ Jahre.

w Anzahl der Eingabebits.
 t_{test} Testzeit, wenn jeder Einzeltest $1 \mu\text{s}$ dauert.



Testauswahl

Wiederholung einiger Begriffe zum Thema:

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 einen Gatteranschluss.

Geziele Suche: Suche von Testeingaben, für die der Modellfehler die Ausgabe verfälscht.

Zufallstest: Auswahl unabhängig von den Modellfehlern und den zu findenden Fehlern.

Fehlerüberdeckung

Fehlerüberdeckung: Anteil der nachweisbaren Fehler

$$FC = \frac{\#DF}{\#F} \Big|_{ACR} \quad (1.52)$$

Für Modellfehler »messbar«, für tatsächliche Fehler nur schätzbar

- aus der Modellfehlerüberdeckung und
- Erfahrungswerten aus vergleichbaren Projekten und Systemen:
 - vom Test übersehene nachträglich gefundene Fehler und
 - der Zuverlässigkeitszunahmen mit der Fehlerbeseitigung (siehe Abschn. 1.4.6 *Reifeprozess*).

Fortsetzungsthemen:

- Fehlermodellierung für digitale Schaltkreise, Schwerpunkte
 - Zusammenhang zwischen der Modellfehlerüberdeckung und der daraus abzuschätzenden tatsächlicher Fehlerüberdeckung
 - und was eigentlich an einem Fehlermodell wichtig ist.
- Berechnung von Testeingaben für Haftfehler.
- Einbau von Selbsttests in digitale Schaltkreise.



Fehlermodellierung



Schaltkreisfehler



Warum Schaltkreise?

Die Konzepte prüfgerechter Entwurf, Fehlermodellierung, Testberechnung, ... überwiegend für digitale Schaltkreis entwickelt:

- Schlechte Steuer- und Beobachtbarkeit.
- Hohe Fehlerbeseitigungskosten.
- Zwang zu kompromissloser Fehlervermeidung und Beseitigung.
- Hohe Anforderungen an die Fehlerüberdeckung der Tests, ...

Schaltkreise heute fehlerarm und zuverlässig.

Gegenstück SW:

- änderungsfreundlich, relativ geringe Fehlerbeseitigungskosten.
- Features oft besser verkaufbar als Qualität.

SW oft voller Fehler und unzuverlässig, aber das muss nicht so bleiben.

Mit den wachsenden Verlässlichkeitsanforderungen an IT-Systeme insgesamt kann die restliche IT-Branche von den Erfahrungen der Schaltkreisbranche lernen.

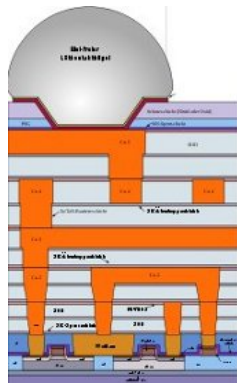
Entstehung und Fehler integrierter Schaltkreise

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.

Typische Herstellungsfehler:

- fehlendes (zu wenig aufgetragenes zu viel weggeätzt) und
- überflüssiges Material (zu viel aufgetragen, zu wenige weggeätzt).



[Kurzschlüsse, Unterbrechungen; Fehlerwirkung nur unter bestimmten Bedingungen (Temperatur, Geschwindigkeit, ...); Beinahefehler, Frühausfälle]



Einteilung in lokale und globale Fehler

Globale Fehler:

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

Lokale Fehler:

- Unterbrechungen einzelner Verbindungen.
- Kurzschlüsse zwischen benachbarten leitenden Gebieten.
- Transistoren, die nicht richtig ein- oder ausschalten.
- Leckströme ohne logische Fehlerwirkung (siehe Folie 3.185 *Frühausfälle*)).



Globale Fehler für Testauswahl uninteressant

- Großflächige Fehler \Rightarrow hohe MF-Rate \Rightarrow Grobtest.
- Prozesssteuerfehler beeinträchtigen die Struktureigenschaften aller Elemente einer Schicht \Rightarrow Parametertest, Teststrukturen.

Grobtest:

- Spannung anlegen und
- kleine Funktionsstichprobe ausprobieren.

Parametertest:

- Stichprobenkontrolle der Transistoreigenschaften, Leitwerte und Kapazitäten an speziellen Teststrukturen, auch schon nach Fertigungszwischenschritten.
- Elektrische Messungen an den Schaltkeisanschlüssen incl. Versorgungsstrom.

Testauswahl ist nur für kleinflächige lokale Fehler, die nur einzelne Transistoren und Verbindungen beeinträchtigen, anspruchsvoll.

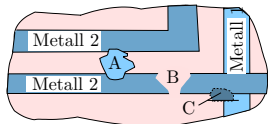
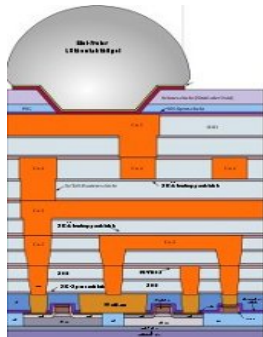


Lokale Fehler

Verbindungs- und Transistorfehler

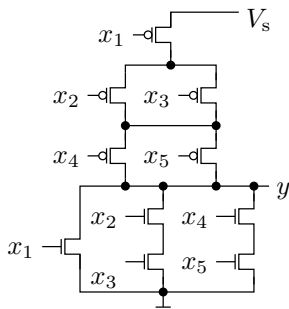
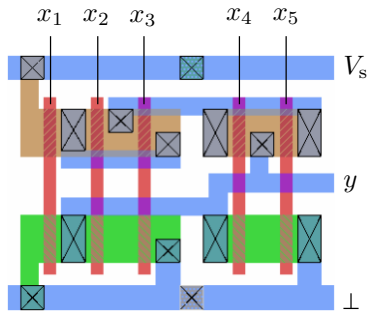
Einzelfehler durch fehlendes und überflüssiges Material:

- kurzgeschlossene und unterbrochene Verbindungen,
- nicht richtig ein- oder ausschaltende Transistoren,
- Leckströme ohne Beeinträchtigung der logischen Funktion, ...
- überhöhte Stromdichten oder Feldstärken, die zu Frühausfällen führen.



- A zusätzliches Metall
- B fehlendes Metall
- C fehlende Isolation

Transistorebene



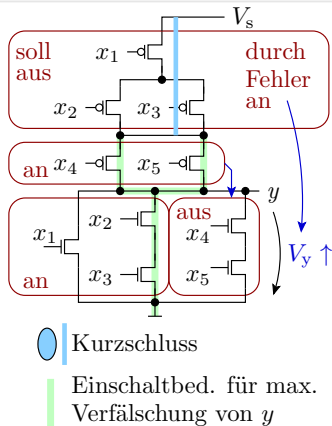
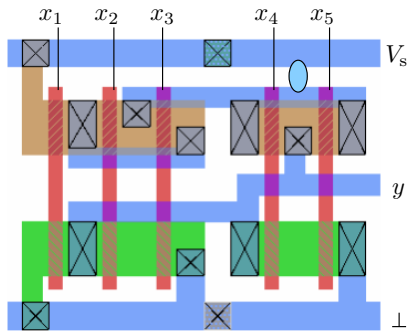
- Durchkontaktierung
- n-Gebiet
- Polysilizium
- p-Gebiet
- Metall

$$y = \begin{cases} 1 & \text{wenn } \bar{x}_1 \wedge (\bar{x}_2 \vee \bar{x}_3) \wedge (\bar{x}_4 \vee \bar{x}_5) \\ 0 & \text{wenn } x_1 \vee (x_2 \wedge x_3) \vee (x_4 \wedge x_5) \end{cases}$$

$$= \overline{x_1 \vee (x_2 \wedge x_3) \vee (x_4 \wedge x_5)}$$

x_i, y_i Eingabe- und Ausgabesignale.
 V_s Versorgungsspannung.
 \perp Masse.

Kurzschluss im Gatters



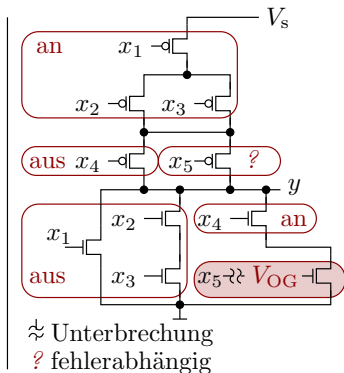
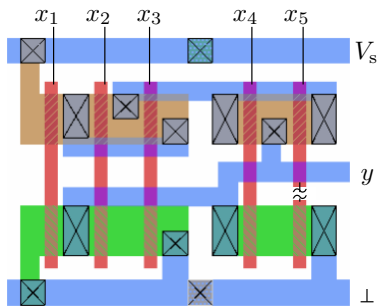
Fehlerbedingte Spannungserhöhung^(IDDQ) an y :

$$(\bar{x}_1 \wedge (\bar{x}_2 \vee \bar{x}_3) = 0 \text{ (aus)}) \wedge (\bar{x}_4 \vee \bar{x}_5 = 1 \text{ (ein)})$$

Größte fehlerbedingte Spannungserhöhung an y :

$$(x_2 = x_3 = 1) \wedge (x_1 = x_4 = x_5 = 0)$$

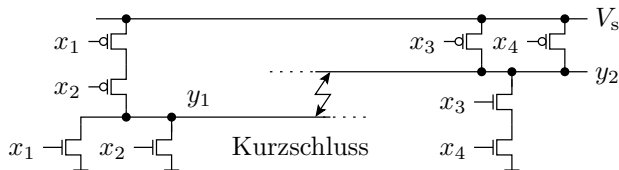
Offenes Gate



Nachweisvoraussetzung: $(x_1 \vee (x_2 \wedge x_3) = 0) \wedge (x_4 = 1) \wedge \dots$

- $(F = 0) \wedge (x_5 = 1)$: $y = 0 \rightarrow 1?$ oder verzögerter Abfall
 - $(F = 1) \wedge (x_5 = 0)$: $y = 1 \rightarrow 0?$, verzögerter Anstieg oder IDDQ
- (? – ob logische Verfälschung hängt ab von Transistorbreiten, ...).

Kurzschluss zweier Gatterausgänge



Mögliche Nachweisbedingungen:

1 $\bar{x}_1 \wedge \bar{x}_2 = 1$ und $x_3 \wedge x_4 = 1$ ($y_{1.trg} = 1$ und $y_{2.trg} = 0$)

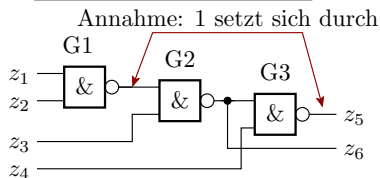
2 $x_1 \vee x_2 = 1$ und $\bar{x}_3 \vee \bar{x}_4 = 1$ ($y_{1.trg} = 0$ und $y_{2.trg} = 1$)

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.

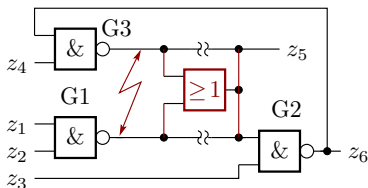
x_i, y_i	Eingabe- und Ausgabesignale.
V_s	Versorgungsspannung.
$y_{i.trg}$	Sollwert am Gatterausgang.

Speicherverhalten durch Kurzschluss

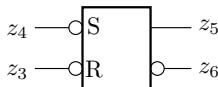
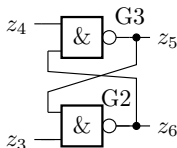
Schaltung mit Kurzschluss



Kurzschlussnachbildung durch ODER



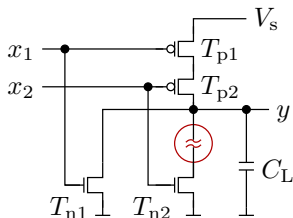
Ersatzschaltung
für $z_1 = z_2 = 1$



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	

Stuck-Open-Fehler



x_2	x_1	y
0	0	1 (setzen)
0	1	0 (rücksetzen)
1	0	speichern
1	1	0 (rücksetzen)

Ausgewählte Unterbrechungen und Transistordefekte können bewirken, dass Gatterausgänge für bestimmte logische Eingaben isoliert sind oder nur zu langsame auf- oder entladen werden. Dieser Fehlertyp wird als Stuck-Open-Fehler bezeichnet und lässt sich nur über Schaltvorgänge am Gatterausgang zuverlässig nachweisen.

x_i, y_i	Eingabe- und Ausgabesignale.
V_s	Versorgungsspannung.
C_L	Lastkapazität.



FM für DIC

Fehlermodelle für digitale Schaltkreise

Etablierte Fehlermodelle für digitale Schaltkreise:

- Haftfehler, Gatterverzögerungsfehler, IDDQ-Fehler und
- Zellenfehler (regelmäßig strukturierte DIC, RAM, ...).

Nicht praxistauglich:

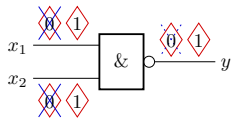
- Toggle-Fehler: wenig Korrelation zwischen der Modellfehler- und Fehlerüberdeckung,
- Kurzschlüsse, Unterbrechungen: erfordert elektrische Simulation, viel höherer Rechenaufwand als reine Logiksimulation.
- Mehrfachfehler, Pfadverzögerungsfehler, ...: überproportionale Zunahme der Modellfehleranzahl mit der Testobjektgröße.

Zu den wichtigen Erfahrungen mit fehlerorientierter Testauswahl für die Übertragung auf andere Systemtypen zählt auch, was für Fehlermodelle sich für Schaltkreise nicht bewährt haben und warum.

Haftfehler

Für jeden Gatteranschluss wird unterstellt:

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



- ◇ 0 sa0-Modellfehler
- ◇ 1 sa1-Modellfehler
- × identisch nachweisbar
- ⋯ implizit nachweisbar

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

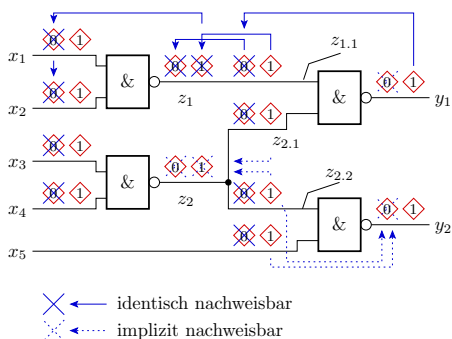
Nachweisidentität (gleiche Nachweismenge)

⋯→ Nachweisimplikation

■ zugehörige Eingabe ist Element der Nachweismenge

Zusammenfassung identisch nachweisbarer Fehler. Optionale Streichung redundanter und implizit nachweisbarer Modellfehler (siehe Abschn. 1.4.4 *Haftfehler*).

Identisch und implizit nachweisbare Fehler im Schaltungsverbund

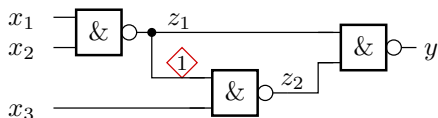


Größe der Anfangsfehlermenge:	24
Anzahl der nicht identisch nachweisbaren Fehler: ohne implizit nachgewiesene Fehler:	14 10

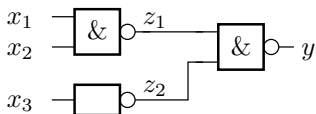
Mengen von identisch nachweisbaren Fehlern	Nachweis impliziert durch
1 sa0(x ₁), sa0(x ₂), sal(z ₁), sal(z _{1.1})	
2 sal(x ₁)	
3 sal(x ₂)	
4 sa0(x ₃), sa0(x ₄), sal(z ₂)	9, 12
5 sal(x ₃)	
6 sal(x ₄)	
7 sa0(z ₂)	5, 6, 8, 11
8 sa0(z ₁), sa0(z _{1.1}), sa0(z _{2.1}), sal(y ₁)	2, 3
9 sal(z _{2.1})	
10 sa0(y ₁)	1, 9
11 sa0(z _{2.2}), sa0(x ₅), sal(y ₂)	
12 sal(z _{2.2})	
13 sal(x ₅)	
14 sa0(y ₂)	12, 13

Redundante Fehler

redundanter Haftfehler

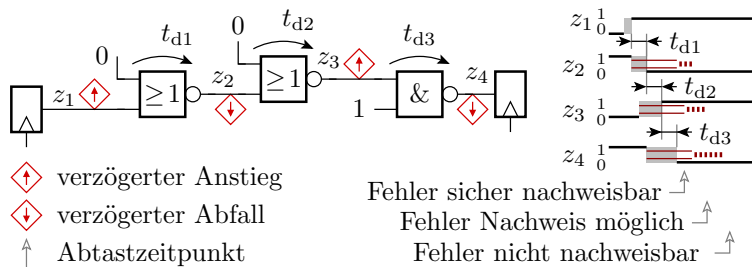


vereinfachte Schaltung



- Bei $z_1 = 0$ ist der Fehler an y nicht beobachtbar und mit $z_1 = 1$ wird der Fehler nicht angeregt. Gatteranschluss mit sa1-Fehler kann mit »1« verbunden werden, ohne dass sich die Funktion ändert. Möglichkeit der Schaltungsvereinfachung.
- Der Nachweis der Redundanz kann schwieriger sein als die Suche eines Tests. Falls nicht erkannt, werden redundante Fehler als nicht erkannt gezählt und mindern die Modellfehlerüberdeckung.
- Fehlermodelle, die viele redundante Fehler erzeugen, sind zur Schätzung der tatsächlichen Fehlerüberdeckung ungeeignet.

Gatterverzögerungsfehler



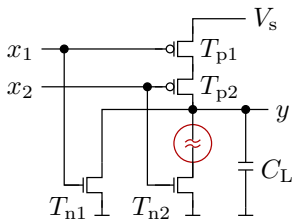
Statt der Haftfehler, Annahme an allen Gatteranschlüssen

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

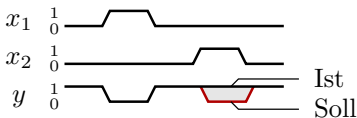
Vereinfachung: Zusammenfassen identischer Fehler, ... Nachweis über einen Signalpfad durch die Schaltung, der an einem Abtastregister beginnt und endet durch eine Folge aus Initialisierungs- und Testeingabe.

$t_{d,i}$ Gatterverzögerungszeit.

Stuck-Open-Fehler



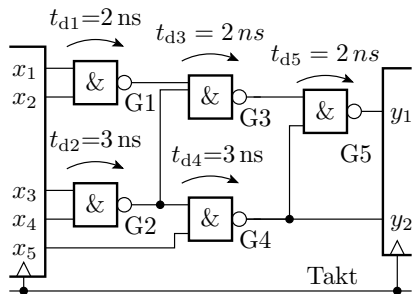
Nachweis (2-Pattern-Test)



- Stuck-Open-Fehler bewirken, dass der Gatterausgang bei lokaler Anregung hochohmig ist.
- Fehlerwirkung wie lange Zusatzverzögerung.
- Nachweis wie Verzögerungsfehler durch 2-Pattern-Tests.

x_i, y_i	Eingabe- und Ausgabesignale.
V_s	Versorgungsspannung.
C_L	Lastkapazität.

Nachweiswahrscheinlichkeit und Pfadlänge

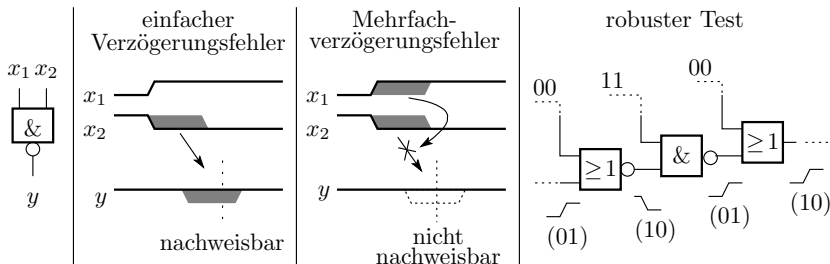


Pfade	$\sum t_{d,i}$
G1-G3-G5	6 ns
G2-G3-G5	7 ns
G2-G4-G5	8 ns
G2-G4	6 ns
G4-G5	5 ns
G4	3 ns

- Die minimal erkennbare Zusatzverzögerung ist die Differenz aus Taktperiode und Soll-Verzögerung.
- Je länger die Sollverzögerung, desto höher die Wahrscheinlichkeit, fehlerverursachte Zusatzverzögerungen zu erkennen.
- Tests über die Pfade mit den längsten Sollverzögerungen erkennen alle nachweisbaren Einzelverzögerungsfehler.

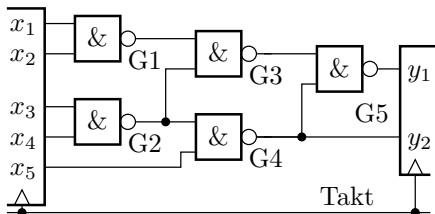
$t_{d,i}$ Gatterverzögerungszeit.
 x_i, y_i Eingabe- und Ausgabesignale.

Mehrfachfehler



- Es sind Mehrfachfehler konstruierbar, die sich bei bestimmten Tests gegenseitig maskieren.
- Robuster Test: Ausschluss der gegenseitigen Maskierung durch max. eine Eingabesignaländerung je Gatter und Testschritt.
- Zusatzattribute wie »minimal erkennbare Zusatzverzögerung« und »Robustheit« bei der Fehlersimulation / Testberechnung mit berechnenbar bzw. berücksichtigbar.

Pfadverzögerungsfehler



Pfade:

G1-G3-G5

G2-G3-G5

G2-G4-G5

G2-G4

G4-G5

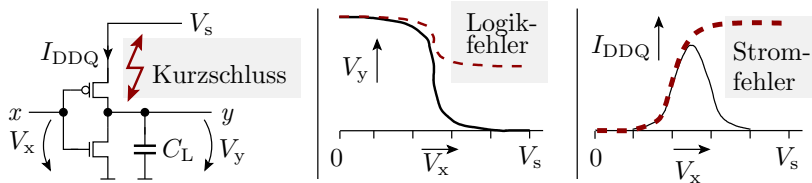
G4

Fehlerannahme »Slow-To-Rise« / »Slow-To-Fall« für alle Schaltungspfade, statt Gatter. Nicht zielführend weil:

- Pfadsanzahl wächst überproportional mit Systemgröße.
- Wegen Nachweisabhängigkeiten sind zu viele Modellfehler je Systemgröße kein Gewinn für die Vorhersagbarkeit der FC .
- Mehr redundante Fehler durch nicht sensibilisierbarer Pfade.

Gatterverzögerungsfehler versprechen mit deutlich weniger Aufwand bessere Vorhersagen für die FC . Folie 5.30: Pfadverzögerungsfehler.

Ruhestromüberwachung

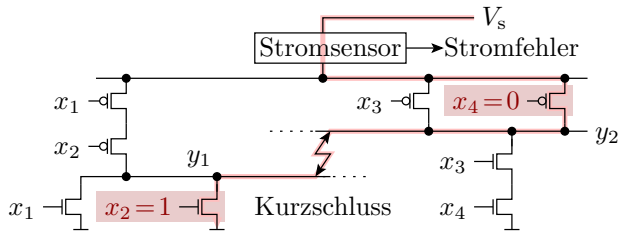


In einer CMOS-Schaltung ist der Gatterausgang nur entweder über NMOS-Transistoren mit 0 (Masse) oder über PMOS-Transistoren mit 1 (Versorgungsspannung V_s) verbunden. Nach jedem Schaltvorgang klingt der Strom auf einen sehr kleinen Wert ab. Kurzschlüsse, nicht richtig ausschaltende Transistoren, ...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

Ruhestromüberwachung erkennt Defekte auch ohne Beobachtungspfad zu einem Ausgang, ist aber messtechnisch problematisch.

I_{DDQ} Versorgungsrुhestrom.
 V_x, V_y Spannung (Potential) von Signal x bzw. y .

Kurzschlussnachweis über den Ruhestrom



Nachweis über statische Stromaufnahme, wenn $y_1 \neq y_2$

- Einfachere logische Nachweisbedingungen, einfachere fehlerorientierte Testsuche, viel kürzere Zufallstests für dieselbe Fehlerüberdeckung.
- Die Unterscheidung von zulässigem und überhöhtem Ruhestrom funktioniert nur bis einige tausend Gatter und verlangt integrierte, im Normalbetrieb zu überbrückende Stromsensoren, ...

x_i, y_i Eingabe- und Ausgabesignale.
 V_s Versorgungsspannung.

Toggle-Test

Alle Signale sind während des Test mindestens einmal auf »0« und einmal auf »1« zu steuern. Als Fehlermodell:

```
für alle Leitungen
    keine 0 einstellbar,
    keine 1 einstellbar.
```

Die Wahrscheinlichkeit, dass ein Toggle-Test einen Leitungshaftfehler nachweist, ist die Wahrscheinlichkeit der Beobachtbarkeit der lokalen Verfälschung. Die bedingten Wahrscheinlichkeiten für Fehlernachweise, wenn Toggle-Bedingung(en) erfüllt, um Zehnerpotenzen kleiner als für ähnlich nachweisbare Haftfehler.

Die Toggle-Überdeckung eines Testsatzes erlaubt wenig Rückschlüsse auf die Haft- oder die tatsächliche Fehlerüberdeckung.



Zellenfehlermodell

- Definition einer Testvorschrift für jede Zelle (jedes Teilsystem).

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

- Durchführbarer Zelltestschritt zählt als nachweisbarer Fehler, z.B. für Gatter jede Eingabe der Wertetabelle.
- Für freistrukturierte Schaltungen sind typisch 10% bis 15% der Zellenfehler redundant.

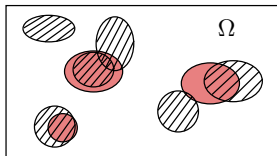
Fehlermodelle mit unbekanntem hohen Anteil redundanter Fehler ungeeignet für Abschätzung von Fehlerüberdeckungen nahe 100%.



Bewährt hat sich das Zellenfehlermodell für die Konstruktion von Testsätzen für regelmäßig strukturierten Schaltungen wie Speicher, Addierer und Multiplizierer. Mit einem geeigneten prüfgerechten Entwurf schafft man hier 100% Zelltestüberdeckung.



Nachweisbeziehungen

Fehler und Modellfehler



- Ω Menge der Eingabewerte / Teilfolgen die einen Fehler nachweisen können
-  Nachweismenge eines Modellfehlers
-  Nachweismenge eines tatsächlichen Fehlers

Die zu findenden Fehler sind zum Zeitpunkt der Testauswahl unbekannt. Deshalb werden für Testauswahl und Abschätzung der Fehlerüberdeckung Fehlermodelle verwendet. Etablierte Fehlermodelle für digitale Schaltkreise sind:

- das Haftfehler- und das,
- Gatterverzögerungsfehlermodell.

Die Nachweiswahrscheinlichkeiten für tatsächliche Fehler hängen außer vom Fehlermodell und der Modellfehlerüberdeckung auch von der Art der Testauswahl ab:

- gezielte Suche,
- zufällige Auswahl, ... [Abschn. 5.1.4: Nachweisbeziehungen.](#)

Nachweisbeziehungen (Wiederholung)

Nachweiswahrscheinlichkeit eines tatsächlichen Fehlers i bei gezielter Suche von w (zufälligen) Tests für den Nachweis von jedem Modellfehler j :

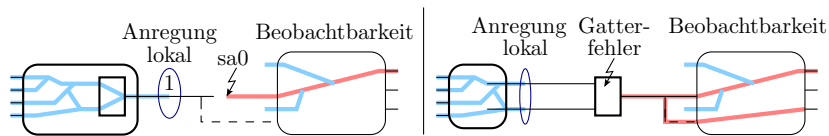
$$p_i = 1 - \prod_{j=1}^{v_i} (1 - (FC_M \cdot (1 - (1 - p_{ij})^w))) \quad (2.10)$$

Fehlerüberdeckung in Abhängigkeit der Modellfehlerüberdeckung bei zufälliger Testauswahl bei einer ähnlicher Verteilung der MF-Raten:

$$N_{\text{Sim}} \approx c_{\text{MF}} \cdot N \quad \text{mit} \quad c_{\text{MF}} = \frac{\bar{\zeta}}{\zeta_{\text{MF}}} \quad (2.11)$$

p_i	Nachweiswahrscheinlichkeit Fehler i .
v_i	Anzahl der ähnlich nachweisbaren Modellfehler für Fehler i .
FC_M	Modellfehlerüberdeckung.
p_{ij}	Wahrscheinlichkeit, dass ein Test, der Modellfehler j nachweist, auch Fehler i findet.
w	Anzahl der je Modellfehler gesuchten Tests. Gefunden werden alle oder keiner.
$\mu_{FC}(N)$	Zu erwartende Fehlerüberdeckung in Abhängigkeit von der Testanzahl.
$\mu_{FM}(N)$	Zu erwartende Modellfehlerüberdeckung in Abhängigkeit von der Testanzahl.
c_{MF}	Fehlermodellspezifische Skalierung der effektiven Testanzahl.
N	Anzahl der Tests.
$\bar{\zeta}$	Mittlere Fehlfunktionsrate je Fehler der tatsächlichen Fehler.
ζ_M	Mittlere Fehlfunktionsrate je Fehler der Modellfehler.

Lokale Gatterfehler und Haftfehler



Der eingezeichnete Haftfehler verlangt

- als Anreibungsbedingung $y = 1$ und
- Beobachtbarkeit des Zweigs von y mit dem Haftfehler.

Der ähnlich nachweisbare Gatterfehler verlangt

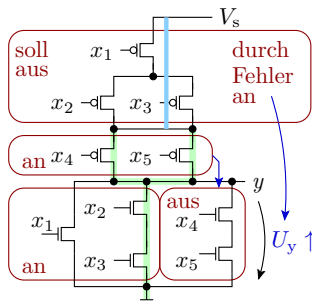
- $y = 1$ und weitere Zusatzbedingungen für die Gattereingabe und
- Beobachtbarkeit von y über einen beliebigen Zweig.

Ein Test, der die Nachweisbedingungen des Haftfehlers erfüllt, erfüllt auch mit nennenswerte Wahrscheinlichkeit die des Gatterfehlers:

$$((\text{Nachweisbedingung } sa0) \wedge \dots) \vee \dots$$

Für die MF-Raten beider Fehler ist eine ähnliche Größenordnung zu erwarten.

Gatterinterner Kurzschluss



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

Kurzschluss

Einschaltbed. max.
Verfälschung von y

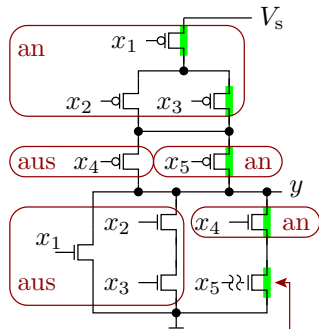
- Nachweis $sa0(x_2)^*$: 2× gute und 1× sicherste Nachweisbed.
- Nachweis $sa0(x_1)$: mehrere unsichere Nachweisbedingungen
- Slow-to-fall-Fehler für x_1 bis x_3 versprechen höhere p_{ij} .

Parameter FC -Abschätzung:

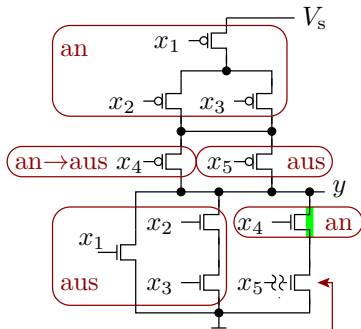
- gezielte Testauswahl: $v_i \approx 2$, $p_{ij} \approx 50\%$
- Zufallstests: $c_{MF} \approx 1$

* $sa0(x_3)$ ist identisch nachweisbar.

Offenes Gate



$sa1(x_5)$: soll aus, ist an



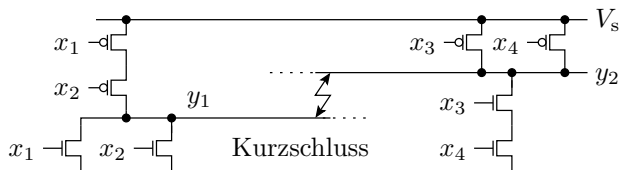
$sa0(x_5)$: soll an, ist aus

- Nachweis $sa1(x_5)$ für $F = 1$, am sichersten wenn $x_2 = \bar{x}_2$.
- Nachweis $sa0(x_5)$ für $F = 0$, wenn y_{soll} von 0 nach 1 schaltet.

Parameter FC -Abschätzung:

- gezielte Testauswahl: $v_i \approx 2$, $p_{ij} \approx 50\%$
- Zufallstests: $c_{MF} \approx 1$

Kurzschluss zweier Gatterausgänge



Bedingungen für den logischen Nachweis:

- y_1 und y_2 müssen sich im fehlerfreien Fall unterscheiden.
- Je nach Fehlerwirkung müssen y_1 oder y_2 dabei beobachtbar sein.

Ähnlich nachweisbare Haftfehler:

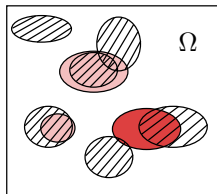
- $sa0(x_1)$, $sa0(x_2)$ wenn $y_2 = 1$ ist und sich durchsetzt
- $sa1(x_1)$, $sa1(x_2)$ wenn $y_2 = 0$ ist und sich durchsetzt
- $sa0(x_3)$, $sa0(x_4)$ wenn $y_1 = 1$ ist und sich durchsetzt
- $sa1(x_3)$, $sa3(x_4)$ wenn $y_1 = 0$ ist und sich durchsetzt

Parameter FC -Abschätzung:

- gezielte Testauswahl: $v_i \approx 8$, $p_{ij} \approx 25\%$
- Zufallstests: $c_{MF} \approx 2$

Trendabschätzung

Die betrachteten Beispielfehler deuten darauf, dass es zu allen lokalen logisch nachweisbaren Fehlern (Kurzschlüssen, Unterbrechungen, Transistorfehler)



- $v_i = 1, 2, \dots$ ähnlich nachweisbare Hauffehler gibt, deren Nachweis mit $p_{ij} = 10\% \dots 50\%$ den Fehlernachweis impliziert.
- eine Zufallstest für dieselbe zu erwartenden Fehlerüberdeckung für Modellfehler die $c_{MF} = 1 \dots 2$ fache Testsatzlänge benötigt.

Gezielte Testsuche

Die Annahmen für gezielte Testauswahl in Beispiel 2.10, Folie 2.57

$p_{ij} = 25\%$, $v_i = 5$ und $w_j = w$ könnten für typische CMOS-Fehler sein:

$p_i(w, FC_M)$	$w = 1$	$w = 2$	$w = 3$	$w = 4$	$w = 5$
$FC_M = 90\%$	72,0%	91,8%	97,5%	99,15%	99,70%
$FC_M = 95\%$	74,2%	93,2%	98,1%	99,47%	99,84%

Bei gezielter Testauswahl hängt die zu erwartende Fehlerüberdeckung weniger von FC_M als von w ab. Eine tatsächliche Fehlerüberdeckungen gleich oder größer der Modellfehlerüberdeckung verlangt eine Zufallsauswahl mehrerer Tests aus jeder Modellfehlernachweismenge.

Anmerkung zum Zufallstests

Typische Abnahme des Anteils der nicht nachweisbaren Fehler bei einem Zufallstest:

$$1 - \mu_{\text{FC}}(N) = \left(\frac{N}{N_0}\right)^{-K} \quad (1.57)$$

$$N_{[\text{eff}]} = c_{\text{MF}} \cdot N_{\text{T}} \quad \text{mit } c_{\text{MF}} = \frac{\bar{\zeta}}{\zeta_{\text{M}}} \quad (1.72)$$

Vernachlässigung Testsatzverlängerung für Modellfehler um $c_{\text{MF}} = 1 \dots 2$ erhöht den Schätzwert für die zu erwartende tatsächliche Fehlerüberdeckung um 1^K bis 2^K mit $0 < K < 1$, im Vergleich zu den vielen anderen Schätzunsicherheit eher unbedeutend.

Modellfehler- und Defektüberdeckung DIC

In Beispiel 2.12 wurde abgeschätzt, dass wenn bei den typischen Schaltkreisausbeuten von $Y \approx 30\% \dots 90\%$ und den Literaturangaben für den Defektanteil getesteter Schaltkreise von $DL = 200\text{dpm}$ bis 1000dpm die Tests etwa $DC \approx 99,9\%$ aller defekten Schaltkreise erkennen müssten (siehe Folie 2.73 *Beispiel 2.6: Ausbeute und Defektanteil nach Ersatz*).

Von Schaltkreistestsätzen wird eine Haftfehlerüberdeckung in der Größenordnung $FC_M = 95\% \dots 99\%$ verlangt. Ist es glaubwürdig, dass Tests mit $FC_M \leq 99\%$ einen Anteil von $DC \approx 99,9\%$ der defekten Schaltkreise erkennen?

Y	Ausbeute (Yield).
DL	Defektanteil nach Ersatz der Produkte mit erkannten Fehlern.
DC	Defektüberdeckung (defect coverage), Anteil der erkennbar defekten Produkte.
FC_M	Modellfehlerüberdeckung.



Die Nachweisbeziehungen zwischen Modellfehlern und tatsächlichen Fehlern erklären nur eine Verringerung des zu erwartenden Anteils der nicht nachweisbaren tatsächlichen Fehler gegenüber dem Anteil der nicht nachweisbaren nicht-redundanten Haftfehler bis auf die Hälfte, aber nicht ein Zehntel.

Mögliche Erklärungen für die »fehlende Zehnerpotenz«:

- Nur ein kleiner Anteil $\eta_{LD} \ll 1$ der defekte Schaltkreise hat einen kleinen, schlecht nachweisbaren Fehler. Die anderen $1 - \eta_{LD}$ defekten ICs werden fast sicher erkannt:

$$DC \approx 1 - (1 - FC_M) \cdot \eta_{LD}$$

- Nur ein kleiner Anteil $\eta_{Ret} \ll 1$ der defekt ausgelieferten Schaltkreise verursacht im Einsatz eine so hohe MF-Rate, dass er erkannt und reklamiert wird. Der Defektanteil in der Literatur ist vermutlich die Reklamationsrate:

$$DL_{Ret} = DL \cdot \eta_{Ret}$$

DC	Defektüberdeckung (defect coverage), Anteil der erkennbar defekten Produkte.
FC_M	Modellfehlerüberdeckung.
η_{LD}	Anteil der Schaltkreise mit schlecht nachweisbaren lokalen Einzelfehlern.
η_{Ret}	Anteil der defekt ausgelieferten ICs, die erkannt und reklamiert werden.



Es ist zu vermuten, dass

- von den ausgelieferten defekten Schaltkreisen nur die Hälfte oder weniger als fehlerhaft erkannt und reklamiert werden, und
- das nur die Hälfte oder weniger der defekten Schaltkreise nur einen kleinflächigen lokalen Fehler haben, der aufwändige Tests verlangt.

Es gibt Untersuchungen in der Literatur, die besagen, dass

- etwa ein Drittel der defekten Schaltkreise von einem Grobtest und ein Drittel vom Parametertest erkannt werden,
- Schaltkreisfehler zur Clusterbildung neigen, ...

Das untermauert die Hypothese.



Testsuche



Testauswahl, gezielte Suche

Auswahl einiger zufälliger Testeingaben:

Wiederhole, bis Mindest-Incremtrate unterschritten:

Fehlersimulation mit neuen Zufallseingaben und Inkrement der Nachweisanzahl w_j aller nachweisbaren Modellfehler j .

Umschalten auf gezielte Suche:

Wiederhole für alle Modellfehler solange $w_j < w$:

Suche Test für Modellfehler j

Fehlersimulation und Inkrement w_j für jeden nachweisbaren Modellfehler j .

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

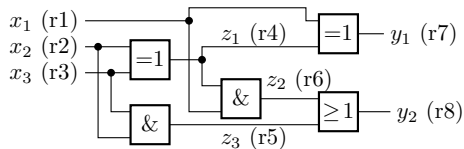
Abschn. 5.2: Testsuche.



Fehlersimulation

Simulation von Haftfehlern

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.



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 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. Das zu simulieren dauert auch bitparallel viele Stunden oder Tage.



D-Algorithmus

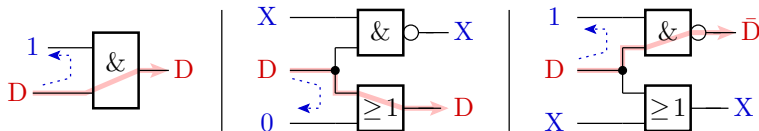
D (Discrepancy)-Kalkül von Roth

Abschn. 5.2.2: D-Algorithmus.

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

- D 0 wenn unverfälscht, 1 wenn verfälscht.
- \bar{D} 1 wenn unverfälscht, 0 wenn verfälscht.
- X Signalwert ist ungültig oder für den Fehlernachweis ohne Bedeutung.

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.

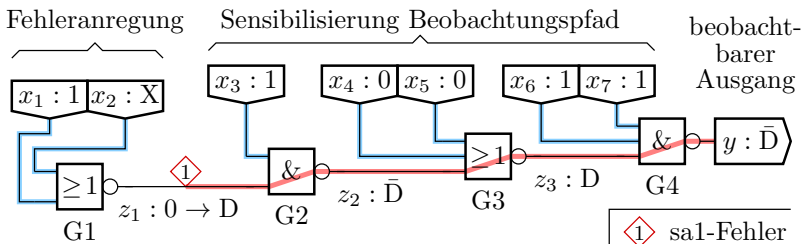
Testsuche für Haftfehler

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,

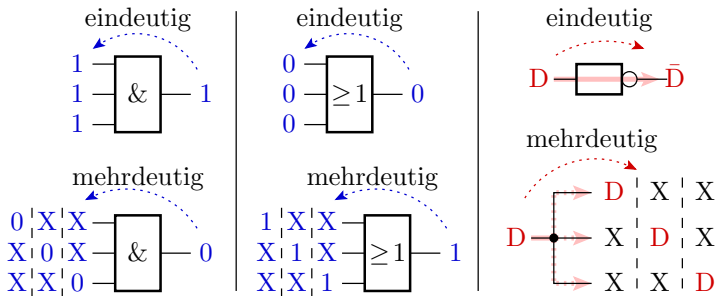
- für die der Wert am Fehlerort invertiert wird und
- bei denen die Invertierung am Fehlerort an einem Ausgang beobachtbar ist.



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

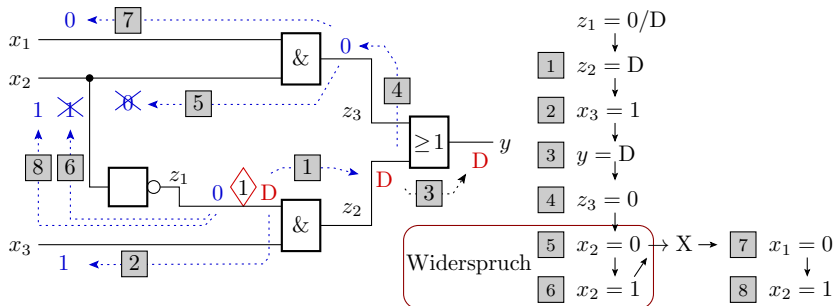
D Discrepanz-Kalkül, 0 wenn unverfälscht, 1 wenn verfälscht.

Ein- und mehrdeutige Pfade



Ausgehend vom Fehlerort:

- Festlegen von Werten zur Weiterführung des Beobachtungs- oder eines Steuerpfads.
- Bei Widersprüchen zurück zu letzten Möglichkeit einer Alternativentscheidung, ... \Rightarrow Baumsuche
- Widerspruch und keine Alternative \Rightarrow Modellfehler redundant.



Baumsuche:

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

	x_3	x_2	x_1	z_3	z_2	z_1	y
	X	X	X	X	X	0D	X
1	X	X	X	X	D	0D	X
2	1	X	X	X	D	0D	X
3	1	X	X	X	D	0D	D
4	1	X	X	0	D	0D	D
5	1	0	X	0	D	0D	D
6	1	X	X	0	D	0D	D
7	1	X	0	0	D	0D	D
8	1	1	0	0	D	0D	D

[Suchreihenfolge verbesserbar, z.B. eindeutige Wertefestlegungen vor mehrdeutigen]



Optimierung Suchalgorithmus

Erfolgsrate der Testberechnung:

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

- Die Testsuche für einen Fehler kann hunderte von Wertefestlegungen beinhalten.
 - Der Suchraum wächst exponentiell mit der Anzahl der mehrdeutigen Festlegungen. Suchräume der Größen $> 2^{30 \dots 40}$ nicht mehr vollständig durchsuchbar.
 - Abbruch der Suche nach einer bestimmten Rechenzeit.
-

Heuristiken:

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

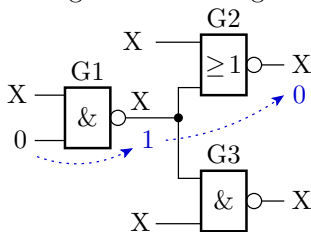


Implikationstest

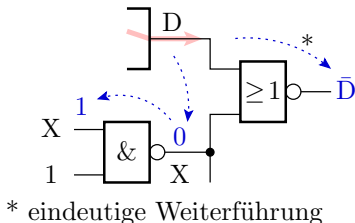
Implikationstest (Widerspruchsfrüherkennung)

- Aus den berechneten Wertefestlegungen alle eindeutig folgenden Werte berechnen.

Implikation in
Signalflussrichtung



D-Pfad- und Rückwärtsimplikation



- Mindert die Entscheidungsbaumtiefe.

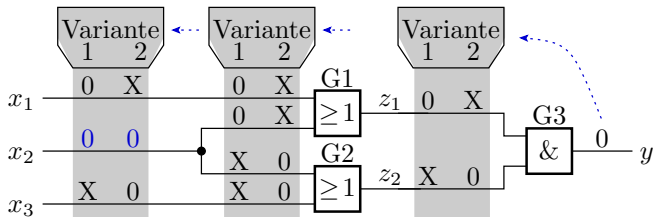
D

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.



Suchraumstrukturierung

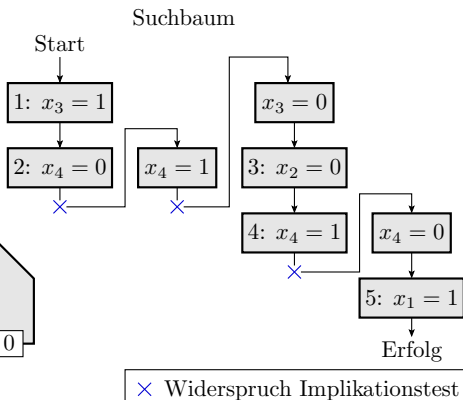
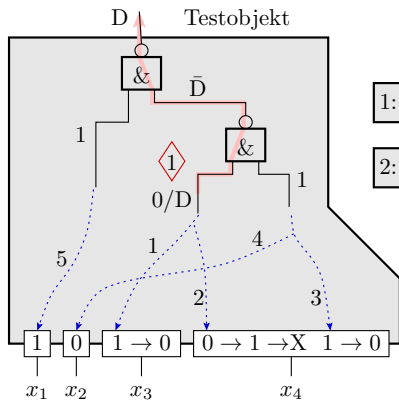


Suchraumbegrenzung

- 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^w . Verringert Rechenaufwand um Zehnerpotenzen.

w

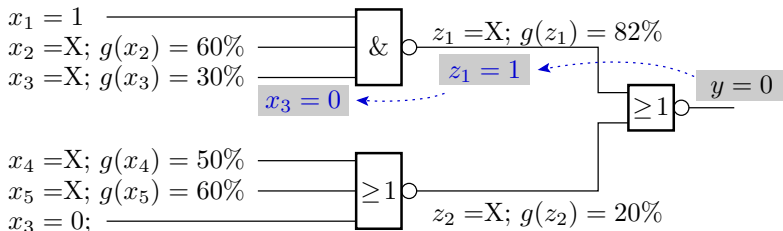
Anzahl der Eingabebits.



- 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.

Geschätzte Erfolgswahrscheinlichkeiten



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

Folie 5.63

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



Komplexe Funktionsbausteine



Komplexe Funktionsbausteine

- Beschreibung durch Tabellenfunktion (Bsp. Volladdierer):

x_2	x_1	x_0	s	c	gegeben	Lösungsmenge
0	0	0	0	0	XXX00	\Rightarrow 00000 (eindeutig)
0	0	1	1	0	01DXX	\Rightarrow 01D \bar{D} D (eindeutig)
0	1	0	1	0		
0	1	1	0	1	1XXXD	\Rightarrow 10D \bar{D} D, 1D0 \bar{D} D (2 Alternativen)
1	0	0	1	0		
1	0	1	0	1	11XX1	\Rightarrow 11111, 11001 (2 Alternativen)
1	1	0	0	1		
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.

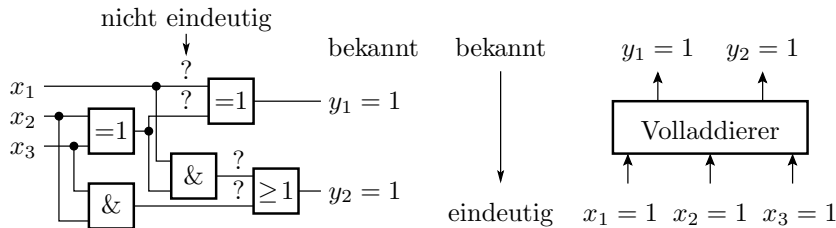
D

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

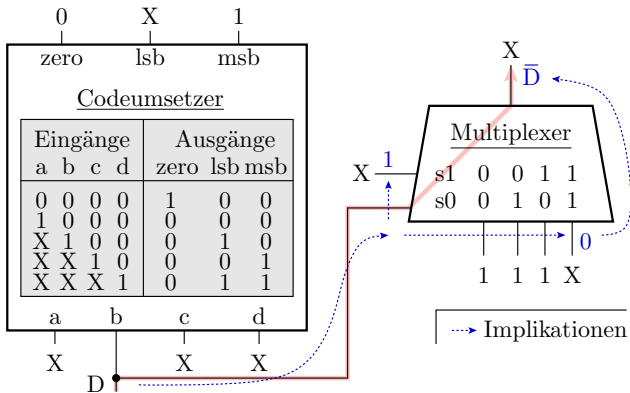
X

Signalwert ungültig oder für den Fehlernachweis ohne Bedeutung.

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.

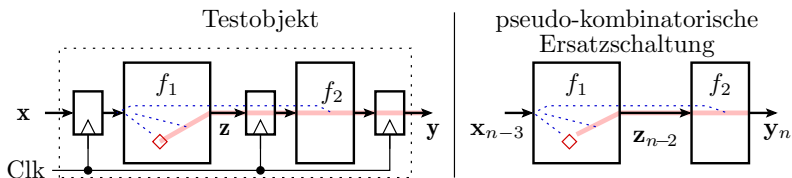
D Discrepanz-Kalkül, 0 wenn unverfälscht, 1 wenn verfälscht.
 X Signalwert ungültig oder für den Fehlernachweis ohne Bedeutung.
 w Anzahl der Eingabebits.



Sequentielle Schaltungen

Pseudo-kombinatorische Ersatzschaltung

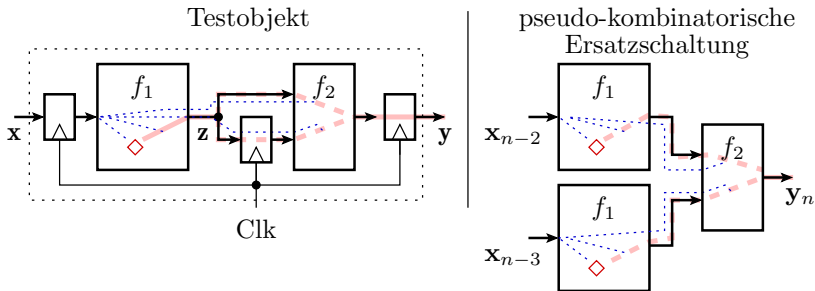
Schaltungen mit Speicherelementen werden für die Testsuche zu einer pseudo-kombinatorischen Ersatzschaltung aufgerollt. Abtastregister in einem geradlinigen Berechnungsfluss werden weggelassen:



- 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).

Verarbeitung in mehreren Zeitebenen

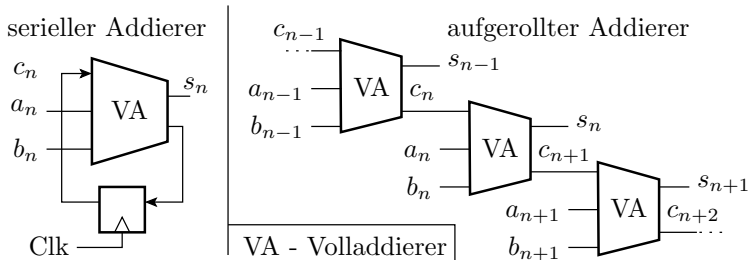


- Mehrere Kopien gleicher Schaltungsteile in der pseudo-kombinatorischen Ersatzschaltung.
- Der eingebaute Haftfehler ist in jeder Kopie der Teilschaltung.
- Für jeden Fehler wird eine Folge von Testeingaben für mehrere Zeitschritte berechnet (Mehr-Pattern-Test).

x, y, z
Clk

Bitvektor der Eingabe-, Ausgabe- und internen Signale, Index ist Zeitschrittnummer.
Takt signal (Clock signal).

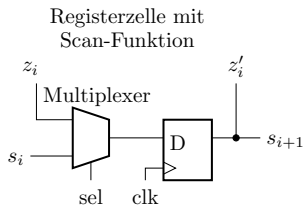
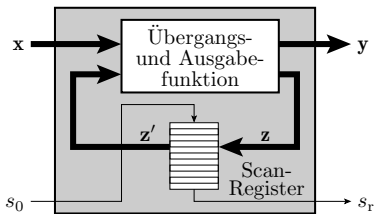
Schaltungen mit Rückführung



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

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

Scan-Verfahren



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

- Mindestschaltungsaufwand ein Multiplexer je Speicherzelle.
- Ablauf eines Testschritts: r Schiebeschritte zum Beschreiben des Zustandsspeichers, Testschritt, r Schiebeschritte zum Lesen (und Überschreiben) des Zustandsspeichers.

x, y	Bitvektor der Eingabe- und Ausgabesignale.
z, z'	Bitvektor der Zustands- und der abgetasteten Zustandssignale.
s_i	Seriellles Schiebesignal i .
sel	Auswahlsignal zwischen Schiebe- und Normalmodus.

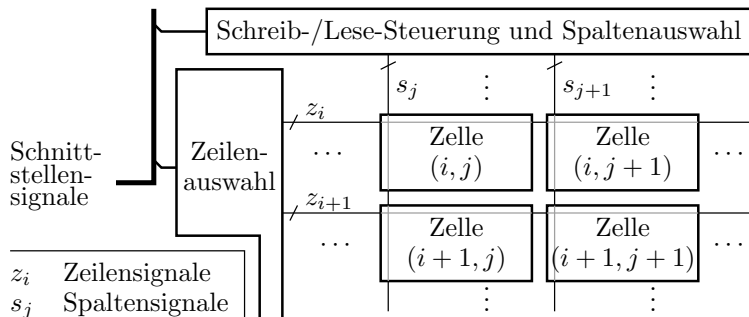


Speichertest



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.



[Zellenfehler, c-Test]



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	$C(i) = \overline{\overline{C(i)}} \Rightarrow$ $C(j) = \overline{C(j)}$	$W(j)0, W(i)0, R(j)0, W(i)1,$ $R(j)0, W(i)0, R(j)0$ $W(j)1, W(i)0, R(j)1, W(i)1,$ $R(j)1, W(i)0, R(j)1$

 $W(i)0$ Schreibe in Zelle i eine 0 $W(i)1$ Schreibe in Zelle i eine 1 $R(j)$ Lese eine beliebige andere Zelle $C(\dots)$ 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



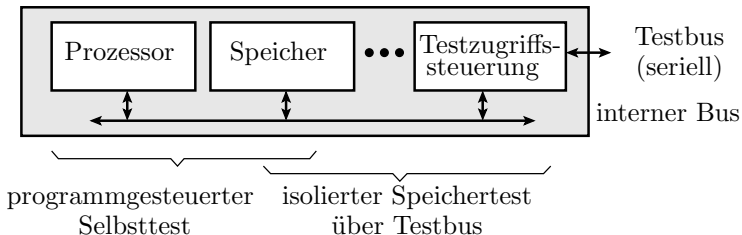
Beispiel Marching Test

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	$W(i)0$	$R(i)0, W(i)1$	$R(i)1, W(i)0$	$R(i)0, W(i)1$	
\vdots	\vdots	\vdots	\vdots	\vdots	
$N - 1$	$W(i)0$	$R(i)0, W(i)1$	$R(i)1, W(i)0$	$R(i)0, W(i)1$	
	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	$R(i)1, W(i)0$		$R(i)0, W(i)1$		$R(i)1$
\vdots	\vdots		\vdots		\vdots
$N - 1$	$R(i)1, W(i)0$		$R(i)0, W(i)1$		$R(i)1$

Mehrfaches Durchwandern des Speichers in unterschiedlicher Reihenfolge mit der Operationsfolge Zelle Lesen, Wert kontrollieren und inversen Wert zurückschreiben.

[Umkehrung Adressierungsreihenfolge \Rightarrow alle Überkopplung; Datenhaltetest]

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



Gezielte Testsuche

- Für Modellfehler, die häufig MF verursachen, ist der einfache Weg, Zufallswerte zu nehmen, Fehlersimulation und abhaken der ausreichend oft nachgewiesenen Modellfehler.
- Zur Fehlersimulation werden die logischen Berechnung in Maschinenbefehlsfolgen übersetzt und bitparallel ausgeführt. Die Simulation größerer Funktionsbausteine mit tausenden von Fehlern und millionen von Tests dauert dennoch Tage und länger.
- Für die gezielte Suche haben wir den D-Algorithmus kennen gelernt. Er treibt vom Fehlerort Beobachtungspfade zu Ausgängen und Steuerpfade zu Eingängen.
- Mit den Mehrdeutigkeiten bei der Pfadsensibilisierung wird ein Suchbaum aufgebaut und bei Widersprüchen 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.

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.



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, Testberechnung 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.

Speicher

- Für Speicher und andere regelmäßig strukturierte Schaltungen gibt es sog. C-Tests. Das sind parametrisierte auf unterschiedliche Größe und Konfigurationen anpassbare Algorithmen zur Testdatenbereitstellung, die alle von einem bestimmten Zellenfehlermodell unterstellten Modellfehler nachweisen.
- Beispiel war ein DRAM für den für alle Speicherzellen bzw. Speicherzellenpaare Haftfehler, Stuck-open-Fehler, zerstörendes Lesen und zwei Arten von Überkopplungsfehler unterstellt wurden.
- Der aus den Fehlerannahmen abgeleitete Marching-Test marschiert mehrfach durch alle Adressen und liest und invertiert den Wert, ein programmtechnisch und auch durch eine Schaltung einfach zu realisierender Algorithmus.



Selbsttest



Selbsttest

[engl. BIST, IC-Testproblem begrenzter Zugang, DFT Scan, isolierter RAM-Test]

Erweiterung zum Selbsttest durch Ergänzung von

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

Vorteile gegenüber externer Prüftechnik:

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

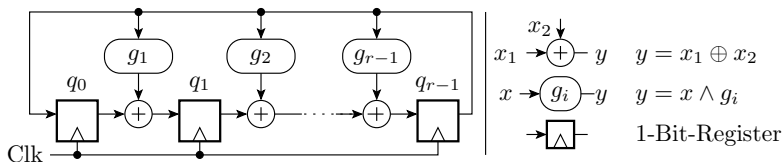


Pseudo-Zufallsregister

Linear rückgekoppelte Schieberegister (LFSR)

Gebräuchlichste Pseudo-Zufallsgeneratoren für Schaltungsselbsttest.

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



Für jede Bitanzahl r des Zustandsvektors gibt es Konstantenvektoren \mathbf{g} , sog. »primitive Polynome«, bei denen alle Zustände außer 000...0 ineinander übergehen. Die werden bevorzugt.

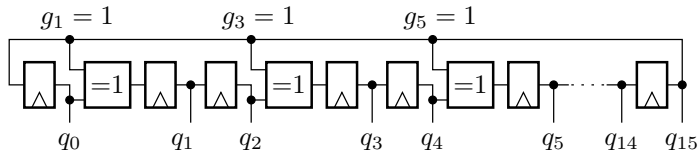
[Konstantenelimination entfernt in der Schaltung die EXOR mit $g_i = 0$]

Primitiven Polynome und die Konstante g

Mit dem Internet-Suchbegriff »Primitive Polynome« findet man z.B. für 16-Bit LFSR (siehe Folie 4.101 *Polynom-Multiplikation und -division*):

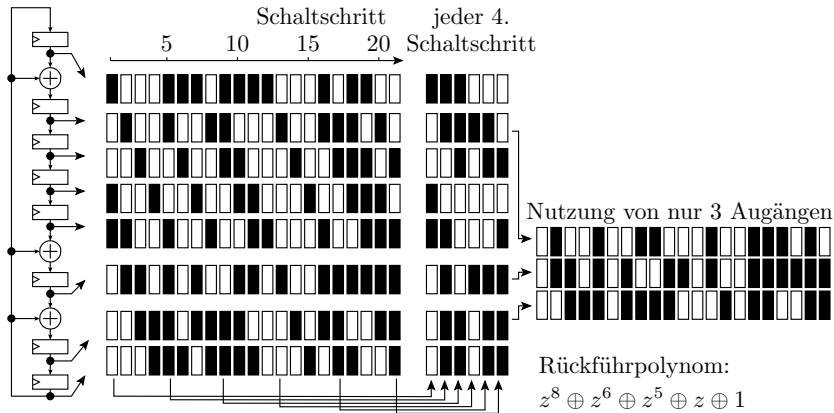
$$z^{16} \oplus z^5 \oplus z^3 \oplus z \oplus 1$$

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 Logische 1 um i Bitstellen verschoben.
- q_i Registerbit i des linear rückgekoppelten Schieberegisters.
- g_i Rückführstellen und gleichzeitig Bits des Generatorpolynoms.
- r Bitanzahl des linear rückgekoppelten Schieberegisters.

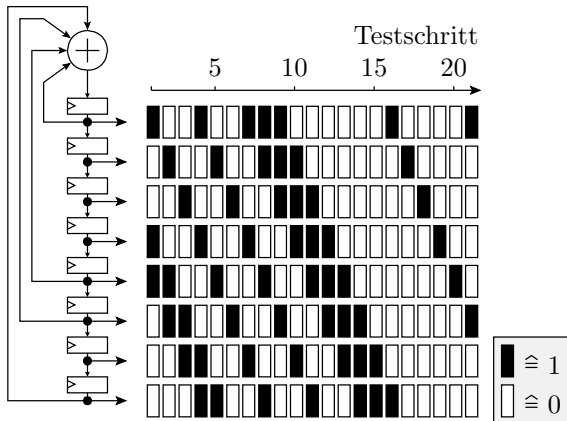
Pseudo-Zufallsfolge eines 8-Bit-LFSR



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

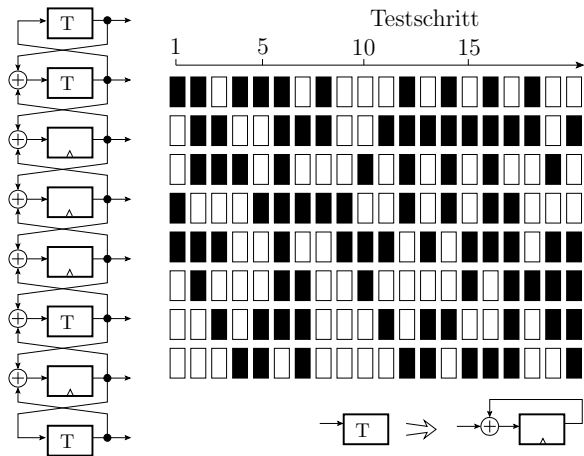
[nach EXOR keine sichtbare Korrelation; genauer phasenverschobene zykl. Bitfolgen]

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.

Zellenautomaten



Folgebit gleich EXOR der Nachbarbits optional plus eigener Bitwert.

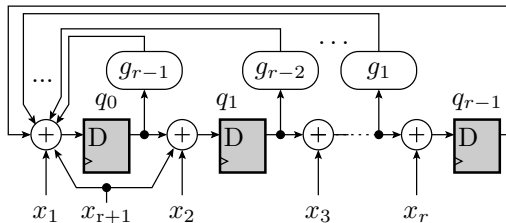
[Ausgewählte Toggle-Zellenzuordnungen \Rightarrow max. Zykluslänge]



Signaturregister

LFSR für parallele Datenströme

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).

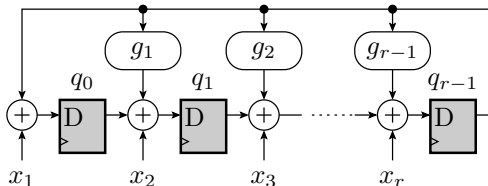


EXOR mit $g_1 = 0$
oder $x_i = 0$ ent-
fallen durch Kon-
stantenelimination

auch mehr als r
Eingänge

- q_i Registerbit i des linear rückgekoppelten Schieberegisters.
- g_i Rückführstellen und gleichzeitig Bits des Generatorpolynoms.
- x_i Testobjektausgang i .
- r Registerlänge.

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 $q \neq 00 \dots 0$ zyklisch durchlaufen werden.
- Keine primitive Rückführung bedeutet aber nicht, dass dann die Fehlermaskierungswahrscheinlichkeit signifikant größer 2^{-r} ist.

q_i	Registerbit i des linear rückgekoppelten Schieberegisters.
g_i	Rückführstellen und gleichzeitig Bits des Generatorpolynoms.
x_i	Testobjektausgang i .



Experiment Maskierungswahrscheinlichkeit

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

$$p_M = 2^{-r}$$

(vergl. Gl. 1.24). Wegen der geringen Eintrittswahrscheinlichkeit sollte die Anzahl der Maskierungen X poisson-verteilt sein:

$$\mathbb{P}[X = k] = e^{-\lambda} \cdot \frac{\lambda^k}{k!} \quad (3.43)$$

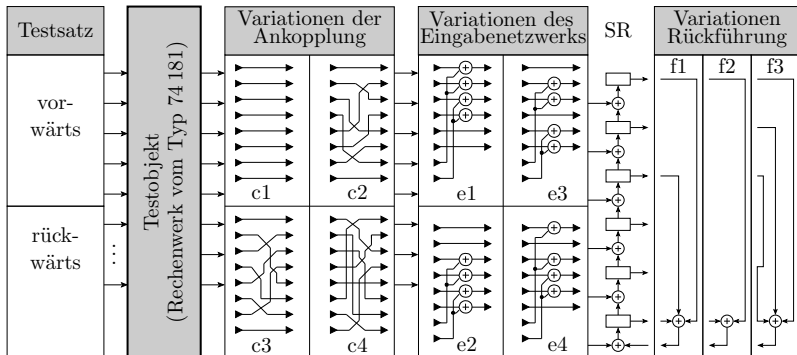
mit dem Erwartungswert $\lambda = \#F_D \cdot 2^{-r}$.

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 Hauffehler erkennt.
- Zählen der Maskierungen durch ein 6-Bit Signaturregister.
- für 96 Kombination der Testanordnung.

p_M	Maskierungswahrscheinlichkeit (Mask probability).
$\#F_D$	Anzahl der vom Testsatz nachweisbaren Fehler.
k	Anzahl von den $\#F_D = 96$ Fehlern, die vom Signaturregister maskiert werden.

Testanordnungen



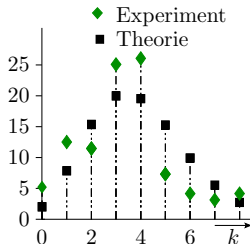
96 Testanordnungen durch Kombination: $(2 \cdot 4 \cdot 4 \cdot 3 = 96)$

- vor-/rückwärts: Varianten der Testreihenfolge
- c1 bis c4: Varianten für eine 8 auf 8 Eingabebezuordnung,
- e1 bis e4: Varianten für eine 8 auf 6 Bit EXOR-Zusammenfassung,
- fb1 bis fb3: Rückführungen (f1 und f3 zufällig gewählt, f2 primitiv).

Anzahl der Maskierungen

Experimentell bestimme Anzahl der Maskierungen k :

		e1				e2				e3				e4			
		c1	c2	c3	c4	c1	c2	c3	c4	c1	c2	c3	c4	c1	c2	c3	c4
vorwärts	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
rückwärts	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



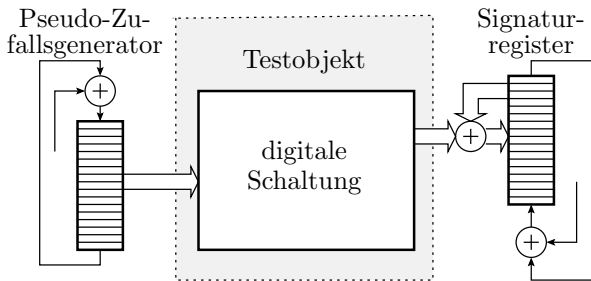
k	0	1	2	3	4	5	6	7
$96 \cdot e^{-3,9} \cdot \frac{3,9^k}{k!}$	1,3	7,5	14,7	19,2	18,7	14,6	9,5	5,3
Experiment	5	12	11	25	26	6	4	3

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



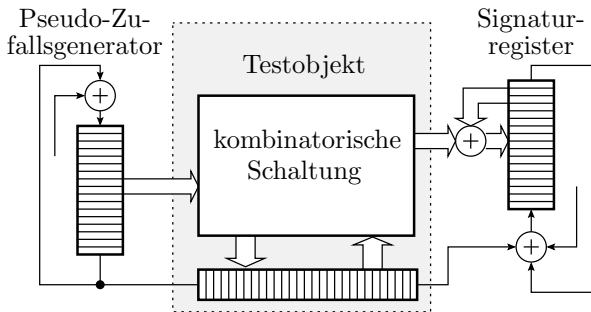
Selbsttest mit LFSR

Selbsttest mit LFSR



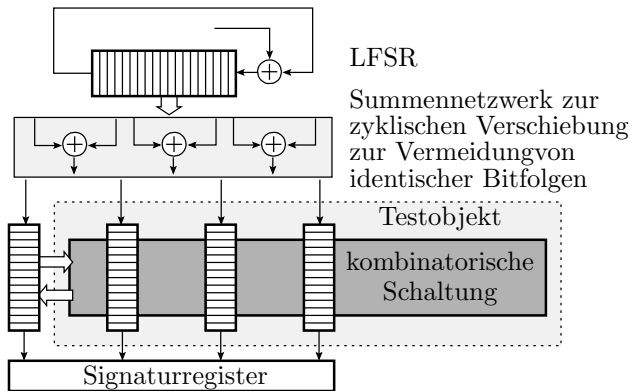
- Einrahmen der Schaltung mit Schieberegistern und Ergänzung von EXOR-Gattern für Rückführungen und SR-Eingänge.
- Init-Wert und Rückführung für Pseudo-Zufallsgenerator so wählen, dass sich keine Testeingabefolgen wiederholen.
- Wenn als Schieberegister vorhandene Ein- und Ausgaberegister verwendet werden,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

Selbsttest mit LFSR und Scan-Registern



- Verschalten der Speicherzellen, aller oder nur die , die die Zustände speichern, zu einem Scan-Register.
- Zwischen den Testschritten Scan-Register seriell in das Signaturregister auslesen und neu beschreiben.
- Der isolierte Test der Übergangsfunktion reduziert in die Regel die erforderliche Testzeit viel mehr, als sie sich durch die zusätzlichen Schiebeschritte erhöht.

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.



Die EXOR-Summen mehrerer LFSR-Bits ergeben dieselbe Folge, nur zyklisch verschoben [Ke07].



Fehlerorientierte Wichtung



Fehlerorientierte Wichtung

Abschn. 5.3.4: Fehlerorientierte Wichtung.

Wenn mit akzeptabler Testsatzlänge zu viele Modellfehler nicht nachgewiesen werden:

- fehlerorientiert Testsuche und
- Ergänzung des Zufallstest um die gezielt berechneten Eingaben.

Wichtung ist eine kostengünstige Alternative zu einem Pattern-Speicher.

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 MF-Raten der Fehler (siehe Folie 5.63 *Testsuche. Geschätzte Erfolgswahrscheinlichkeiten*).

Berechnung Wichtungen und Beobachtbarkeiten

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, ...

x_1	$g(x_1), b(x_1)$		$g(y), b(y)$	y	$b(x_2) = b(y) \cdot g(x_1)$ $b(x_1) = b(y) \cdot g(x_2)$ $g(y) = g(x_1) \cdot g(x_2)$
x_2	$g(x_2), b(x_2)$				

x_1	$g(x_1), b(x_1)$		$g(y), b(y)$	y	$b(x_1) = b(y) \cdot (1 - g(x_2))$ $b(x_2) = b(y) \cdot (1 - g(x_1))$ $g(y) = 1 - (1 - g(x_1)) \cdot (1 - g(x_2))$
x_2	$g(x_2), b(x_2)$				

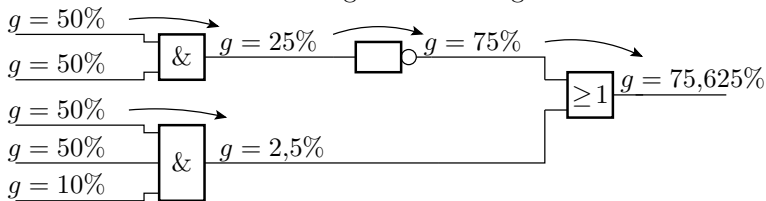
x	$g(x), b(x)$		$g(y), b(y)$	y	$b(x) = b(y)$ $g(y) = (1 - g(x))$
-----	--------------	---	--------------	-----	--------------------------------------

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

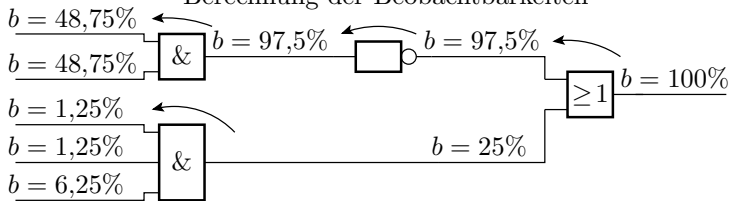
$g(\dots)$ Wichtung, Auftrittshäufigkeit des Signalwerts 1 .

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

Berechnung der Wichtungen



Berechnung der Beobachtbarkeiten



Berechnung der Nachweiswahrscheinlichkeiten:

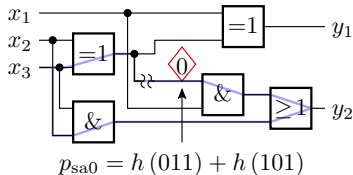
$$p_{\text{sa}0}(x_i) = b(x_i) \cdot g(x_i)$$

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

Rekonvergente Auffächerung

Bei rekonvergenter Auffächerung werden abhängige Zufallswerte log. verknüpft, so dass die einfachen Regel für $\mathbb{P}(A \wedge B)$ etc. nicht gelten.

Berechnung der Nachweiswahrscheinlichkeiten über Wertetabellen:



— rekonvergente
Auffächerung

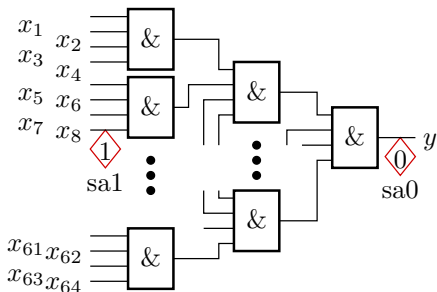
■ Eingaben die den
Fehler nachweisen

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

$$p_{\text{sa}0} : 0,25 \quad 0,072 \quad 0,294$$

x_i, y_i Eingabe- und Ausgabesignale.
 g Wichtung der Eingabesignale x_1, x_2 und x_3 .
 $h(\mathbf{x})$ Auftrittshäufigkeit des Eingabevektors $x_3x_2x_1$.

Fehlerüberdeckung und Wichtung



Nachweiswahrscheinlichkeiten für $g(x_i) = g$:

- der 64 sa_1 -Fehler

$$p_{sa_1}(1) = g^{63} \cdot (1 - g)$$

$$p_{sa_1}(N) = 1 - e^{-g^{63} \cdot (1-g) \cdot N}$$

- des einen sa_0 -Fehlers

$$p_{sa_0}(1) = g^{64}$$

$$p_{sa_0}(N) = 1 - e^{-g^{64} \cdot N}$$

Zu erwartende Fehlerüberdeckung als Mittelwert der Nachweiswahrscheinlichkeiten aller 65 Haftfehler:

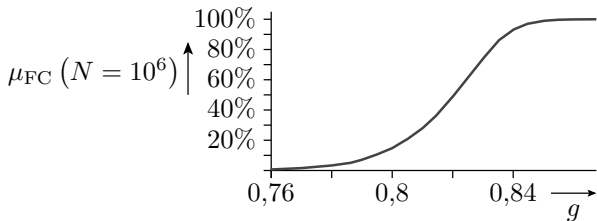
$$\mu_{FC}(N) = 1 - \frac{64 \cdot e^{-g^{63} \cdot (1-g) \cdot N} + e^{-g^{64} \cdot N}}{65}$$

g Wichtung der Eingangssignale x_1 bis x_{64} .

N Anzahl der Tests.

$\mu_{FC}(N)$ Zu erwartende Fehlerüberdeckung in Abhängigkeit von der Testanzahl.

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



Eine Erhöhung der Wichtung an allen Eingängen von $g \leq 76\%$ auf $g \geq 86\%$ erhöht die Fehlerüberdeckung praktisch von 0 auf 1.

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

g Wichtung der Eingabesignale x_1 bis x_{64} .

N Anzahl der Tests.

$\mu_{FC}(N)$ Zu erwartende Fehlerüberdeckung in Abhängigkeit von der Testanzahl.



Fehlerorientierte Wichtungsauswahl

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 Festlegung der Wichtung so, dass die gefundenen Tests bevorzugt erzeugt werden.
- 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:

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

Beispiel für $m_{\text{AND}} = 3$:

Test	1	2	3	4	5	6	7	8	9	10	11	$g(x_i)$
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.

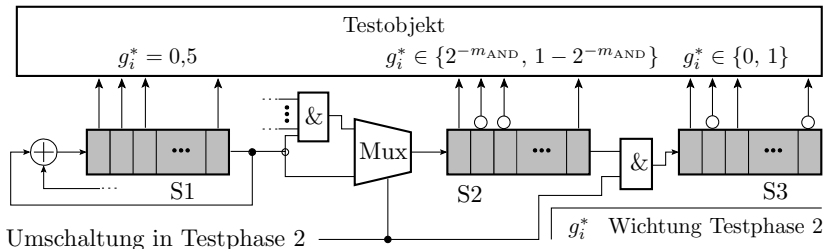
#0

Anzahl der Nullen im Testsatz für Eingang i des Testobjekts.

#1

Anzahl der Einsen im Testsatz für Eingang i des Testobjekts. n_{AND} Anzahl UND-verknüpfter ungewichteter Bits zur Erzeugung der Wichtungen $0 < g \ll 0,5$.

Implementierung als Selbsttest



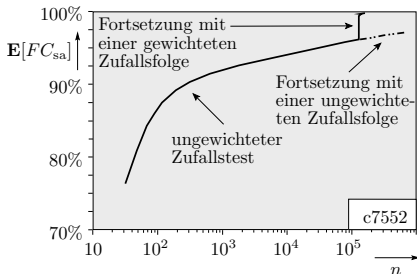
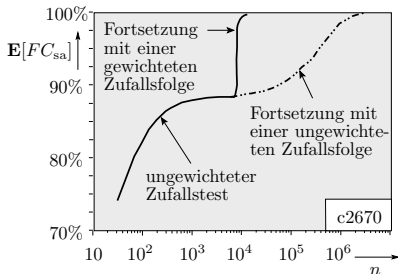
- Testphase 1: Erzeugung ungewichteter Pseudo-Zufallseingaben mit LFSR S1. Serielle Weitergabe an die Schiebereg. S2 und S3.
- Testphase 2: Verringerung der Wichtung
 - in S2 durch UND-Verknüpfung von m_{AND} Ausgabefolgen von S1 und für S2 und
 - in S3 durch »UND 0«.

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

Kaum aufwändiger als BIST mit nur ungewichtetem Zufallstest.

Benchmark-Experimente

- Beispielentwurf für die größten ISCAS-Benchmarks c2670, c7552.
- Test mit 10^4 bzw. 10^5 ungewichteten Zufallsmustern, die 90% bzw. 95% der Haftfehler nachweisen.
- Gezielte Testberechnung für die restlichen Haftfehler.
- Individuelle Wichtung aller Eingabebits zur Maximierung der mittleren Auftretshäufigkeit der berechneten Testeingaben.
- Weitere 10^4 bzw. 10^5 gewichtet Zufallstests weisen alle verbleibenden nichtredundanten Modellfehler nach.





RAM-Selbsttest



Marching-Test für RAM

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	$W(i)0$	$R(i)0, W(i)1$	$R(i)1, W(i)0$	$R(i)0, W(i)1$	
\vdots	\vdots	\vdots	\vdots	\vdots	
$N - 1$	$W(i)0$	$R(i)0, W(i)1$	$R(i)1, W(i)0$	$R(i)0, W(i)1$	
	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	$R(i)1, W(i)0$		$R(i)0, W(i)1$		$R(i)1$
\vdots	\vdots		\vdots		\vdots
$N - 1$	$R(i)1, W(i)0$		$R(i)0, W(i)1$		$R(i)1$

Die Testeingabebereitstellung und -auswertung verlangt 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



Selbsttest digitaler Schaltkreise

- Ergänzung von Pseudo-Zufassgeneratoren an den Eingängen, in der Regel LFSR, für die Durchführung langer Zufallstests, typ. Testanzahl $m = 10^{6 \dots 9}$, 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 Fehlerüberdeckung werden auch interne Speicherzellen zu Scan-Registern zusammengefasst, die zwischen den Testschritten in das Signaturregister ausgelesen und mit neuen Pseudo-Zufallswerten beschrieben werden.



Fehlerorientierte Wichtung und RAM

- Wenn die Fehlerüberdeckung von $m = 10^{6 \dots 9}$ 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 einfachsten 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 in den gefundenen Testsatz angepasst.
- Für regelmäßig strukturierte Schaltungen wie RAM gibt es oft einfachere Lösungen mit exzellenter Fehlerüberdeckung. Als Beispiel wurde die Implementierung eines Marching Test für einen RAM-Block skizziert.



Seitenreferenzen

(siehe Folie 5.63 *Testsuche. Geschätzte Erfolgswahrscheinlichkeiten*)

65