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

Prof. G. Kemnitz

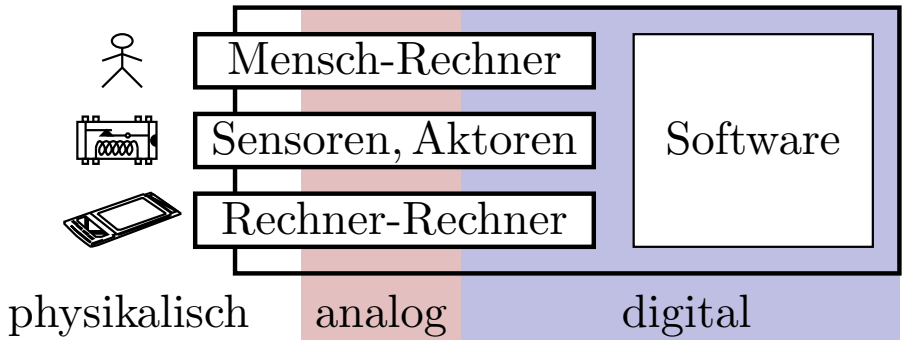
Institut für Informatik, TU Clausthal (TV_F6)

22. Juli 2020

Inhalt TV_F6: Hardware-Test und Selbsttest

Schaltungsstrukt. & Test

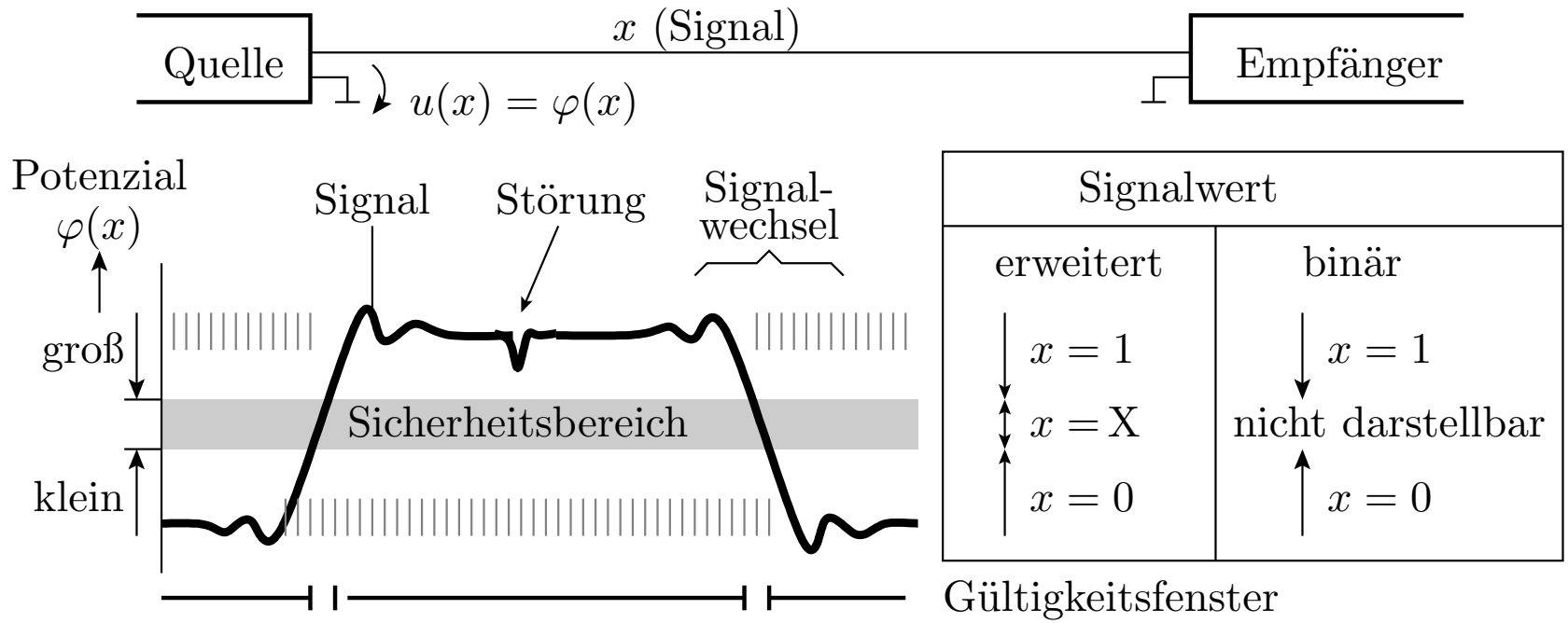
Struktur heutiger Hardware-Systeme



Die Hardware von IT-Systemen besteht aus physikalischen, analogen und digitalten Verarbeitungselementen:

- Physikalisch und analog in der Regel nur an den Schnittstellen zur Ein- und Ausgabe, Spannungversorgung, ... Separat zu testende Bausteinen. Überschaubare Funktionsvielfalt.
- Kompliziertere und spezielle Funktionen werden digital realisiert.
- Noch kompliziertere Funktionen: programmierbare Schaltungen und Rechnerstrukturen + Software.
- Überwachung und Fehlerbehandlung gehört zu den komplizierten und damit typisch in Software realisierten Funktionen.

Digital vs. Analog



Digitalisierung: Informationstdarstellung durch Bits

- physikalische Werteunterscheidungt nur groß, klein und ungültig,
- Abtastung im Gültigkeitsfenster.
- Robust gegen Verfälschungen durch Rauschen, induktives und kapazitives Übersprechen, Fertigungsstreuungen, Alterung, ...

Vorteile der analogen Verarbeitung:

- die Information braucht nicht digitalisiert zu werden.
- viel mehr Leitungen zur parallelen Darstellung derselben Informationsmenge.
- viel mehr Bauteile zur Realisierung derselben Funktion.

Vorteile der digitalen Verarbeitung

- Information ist speicherbar.
- Speicherung und Verarbeitung ohne Informationsverlust
- größe relative Fertigungstoleranzen erlauben viel größere Integrationsdichten und höhere Geschwindigkeiten.
- Größerer funktionaler Gestaltungsspielraum.
- programmierbare und Rechnerstrukturen erlauben Funktionsfestlegung durch Software.
- Komplexe Digitalschaltungen werden heute wie Software entworfen.

Statische Tests

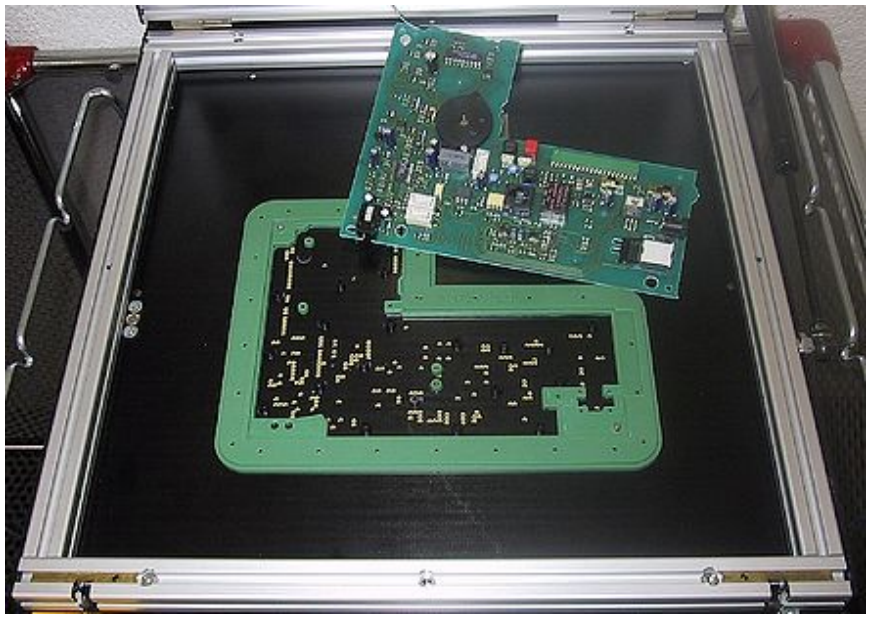
Hierarchie und Test

Der Test der Hardware folgt dem Entstehungsfluss:

- Entwürfe werden durch Simulation und mit Prototypaufbauten getestet. Fehlerentstehung und Testauswahl siehe später Foliensatz 7.
- Bei der Fertigung von Schaltkreisen, anderer Bauteile und Verdrahtungsträgen folgen nach ausgewählten Fertigungsschritten statische Tests durch physikalische Messung, mit bildgebenden Verfahren, ...
- Nach der Fertigung folgen gründliche statische und dynamische Tests.
- Die Baugruppenfertigung verwendet getestete Bauteile.
- Nach der Baugruppenfertigung folgen statische Verbindungs- und Bestückungstests ohne Versorgungsspannung.
- Erst nach Beseitigung aller Verbindungsfehler werden Funktionen ausprobiert.
- Optional Burn-In (siehe FS 4), Tests in der Anwendungsungebung, ...

Bestückungs- und Verbindungstests

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



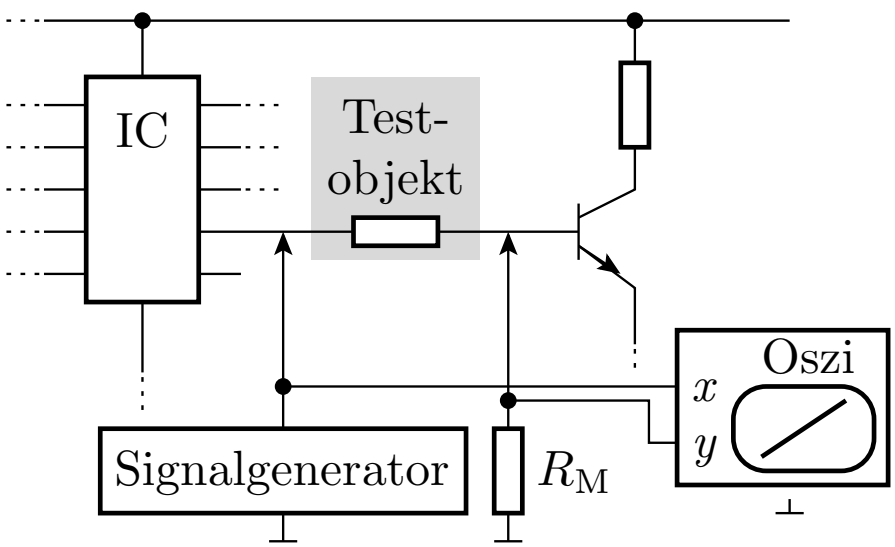
Die Nadeln sind mit reine Relais-Matrix zur Verbindung mit den Prüfgeräten angeschlossen. Auch Bestückungsfehler lassen sich überwiegend mit Zweipunktmessungen von Strom-Spannungsbeziehungen erkennen. Fehlerlokalisierung im Vergleich zu der für einen dynamischen Test, der versagt, sehr einfach.

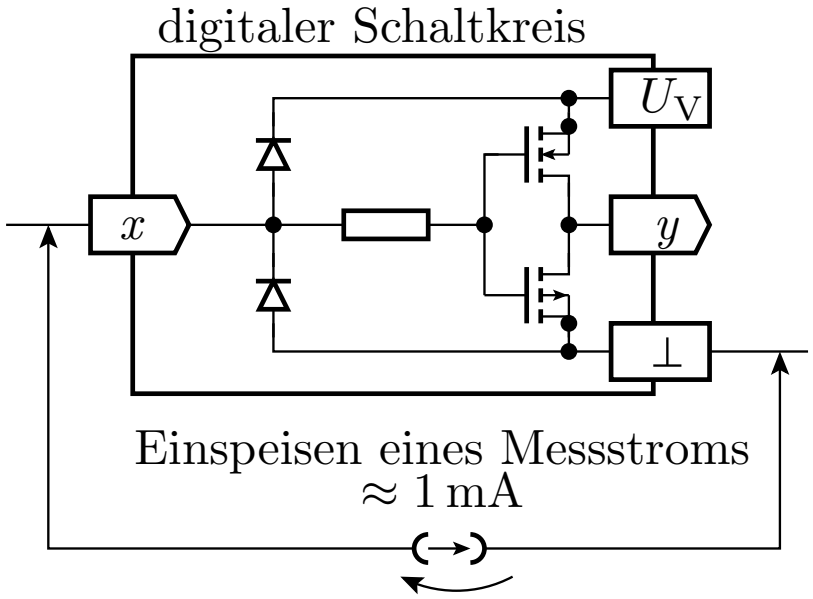
Bestückungstests

Kontrolle auf Bestückungsfehler durch Überprüfung ausgewählter Zweipunktmerkmale:

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

Zu Kontrolle, dass Schaltkreisanschlüsse mit dem Kontakt-Pad verbunden sind, werden die Schutzdioden zur Versorgungsspannung und Masse ausgemessen.

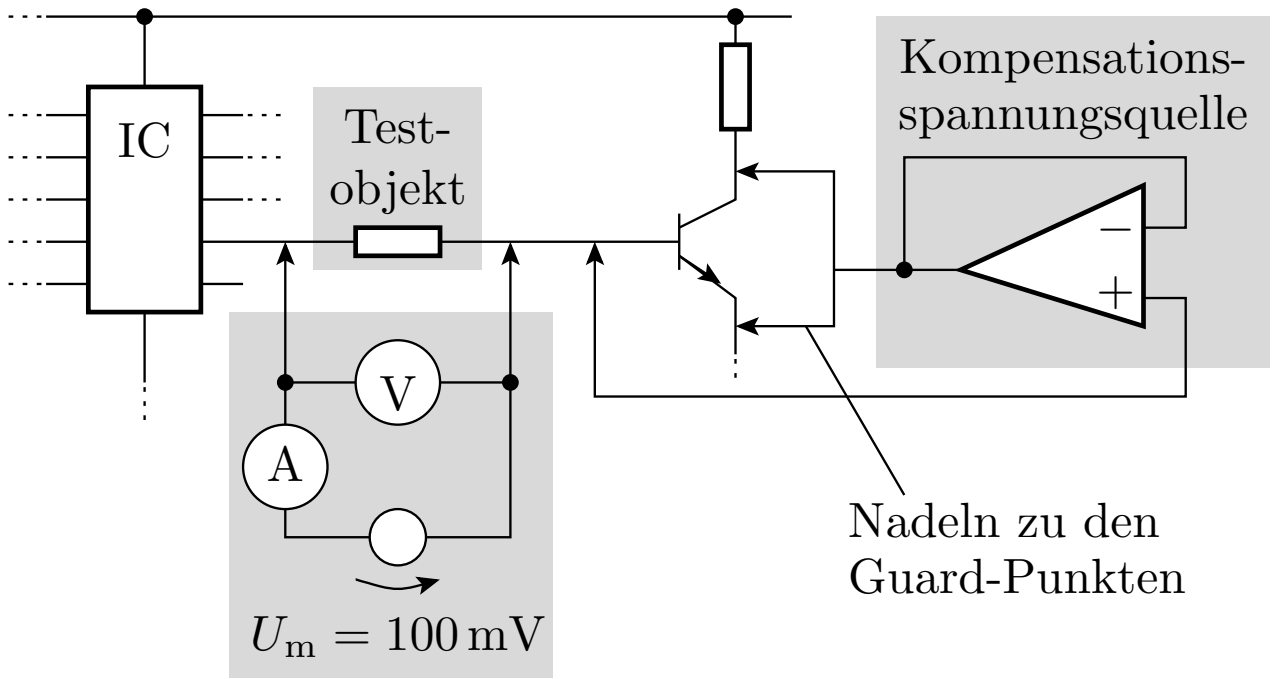


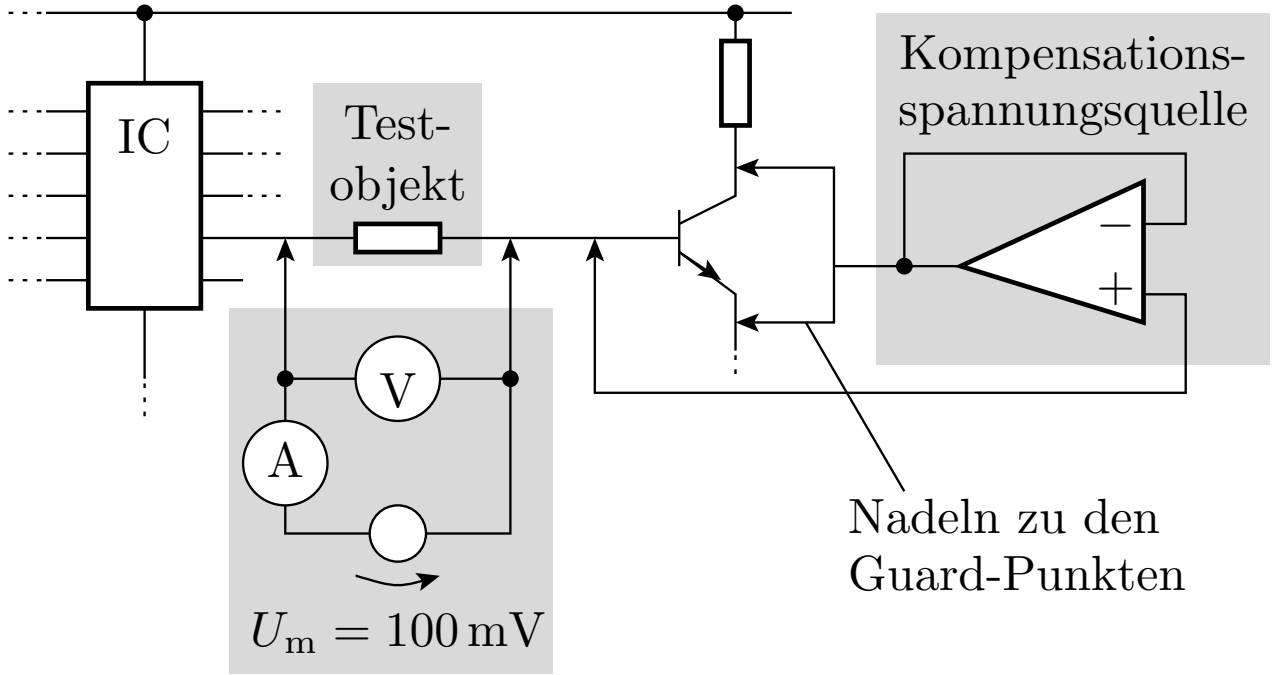


Analoger In-Circuit Test

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. Problematisch:

- die Toleranzbereiche der Sollwerte mit allen Bauteilstreuungen,
- die Erkennungssicherheit für Fehlbestückungen, z.B. bei sehr kleinen Kapazitäten.



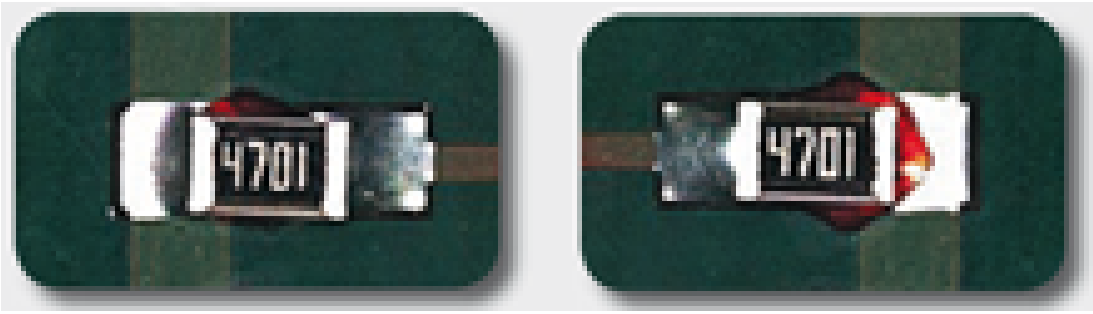


Unterdrückung von Parallelströmen zum Testobjekt durch Kompensation der Spannungsabfälle über den wegführenden Bauteilen auf einer Testobjektseite auf null über »Guard-Punkte«. Erlaubt einen isolierten Zweipoltest.

- Vereinfacht die Testauswahl, Sollwertfestlegung, ...
- Mindert die Häufigkeit von Fehlklassifikationen.

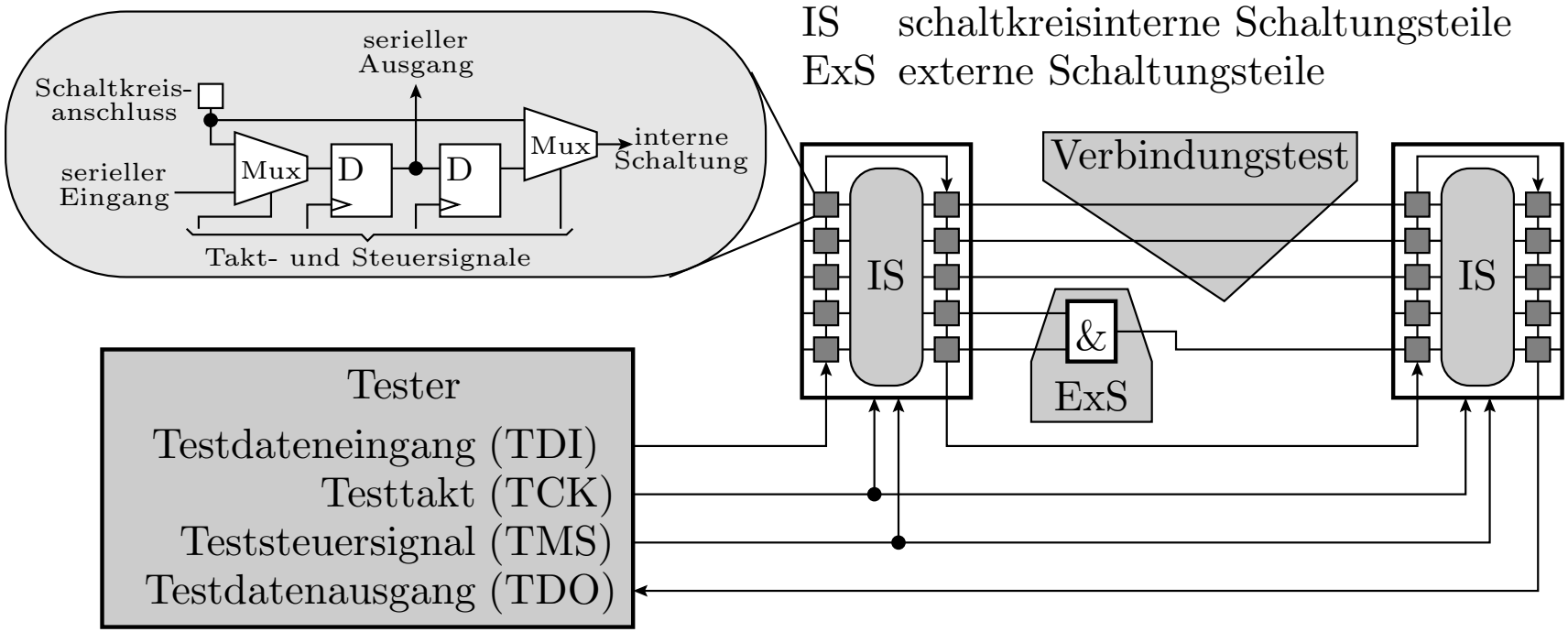
Optische Inspektion

Es gibt Bestückungsfehler, die sind optisch, aber nicht elektrisch erkennbar. Bild links korrekt bestückter SMD-Widerstand, rechts Lötfläche durch Kleber verschmutzt. Elektrisch leitende aber keine feste Lötverbindung:

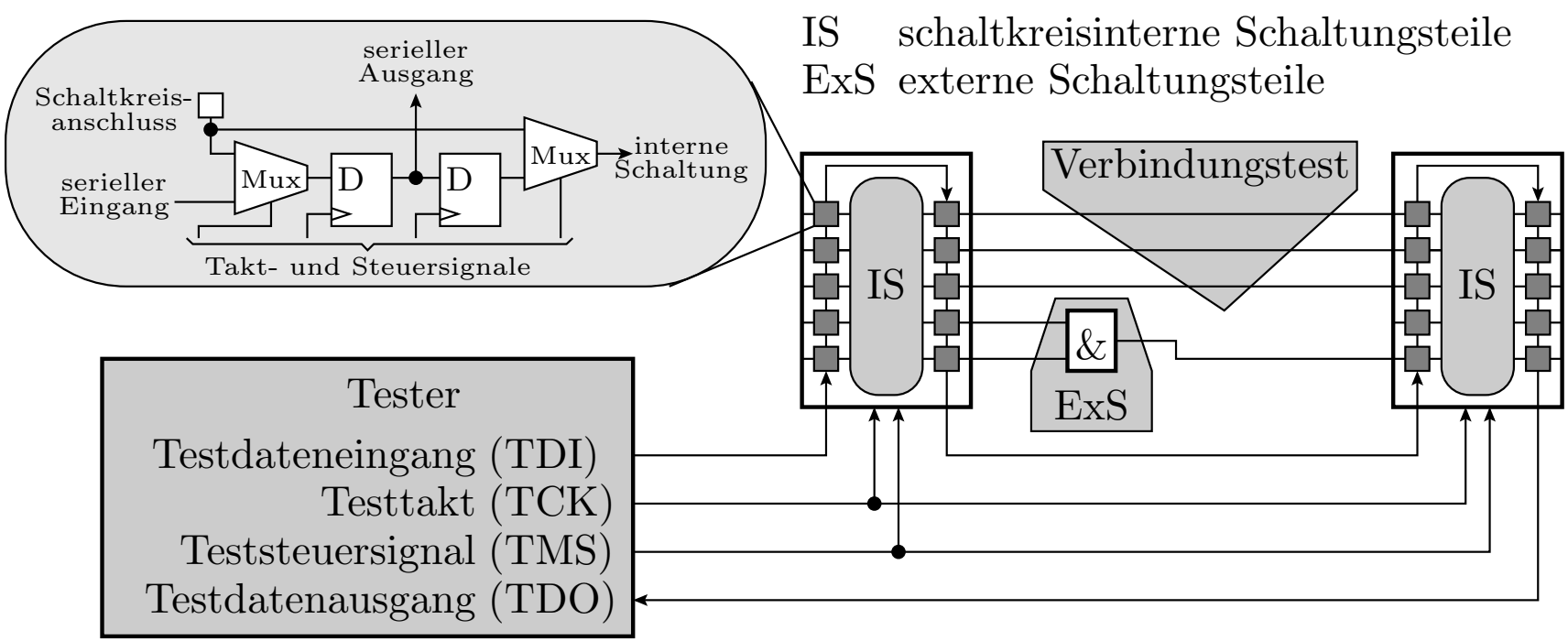


Nachweis nur durch visuelle Kontrolle möglich. Besonderes Problem: Nach einem Ausfall der Baugruppe z.B. bei Vibration in einem Fahrzeug ist sofort erkennbar, dass es sich um einen Fertigungsfehler handelt, der (optisch) erkennbar gewesen wäre. Fall für Produkthaftung.

Boundary-Scan



Ersatz der mechanischen Nadeln durch »silicon nails« (seriell beschreibbare Register an den Schaltkreisanschlüssen, im Normalbetrieb überbrückt). Alternative zu den teuren, für jede Baugruppe speziell anzufertigenden Nadeladaptern.



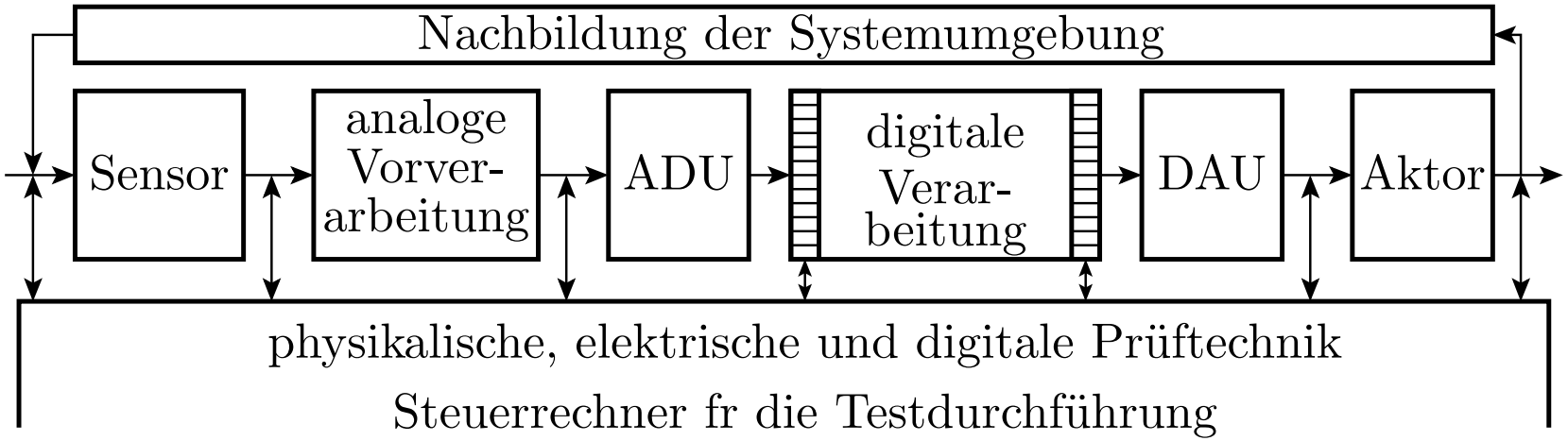
Ablauf eines Testschritts für den Baugruppentest:

- BS-Register aller Schaltkreise auf der Baugruppe seriell beschreiben,
- Datenübergabe (Update),
- Datenübernahme (Capture),
- serielle Ausgabe der übernommen Wert und Laden des Eingabevektors für den nächsten Testschritt.

Fortsetzung Abschn. # »Testbus«.

Funktionstest

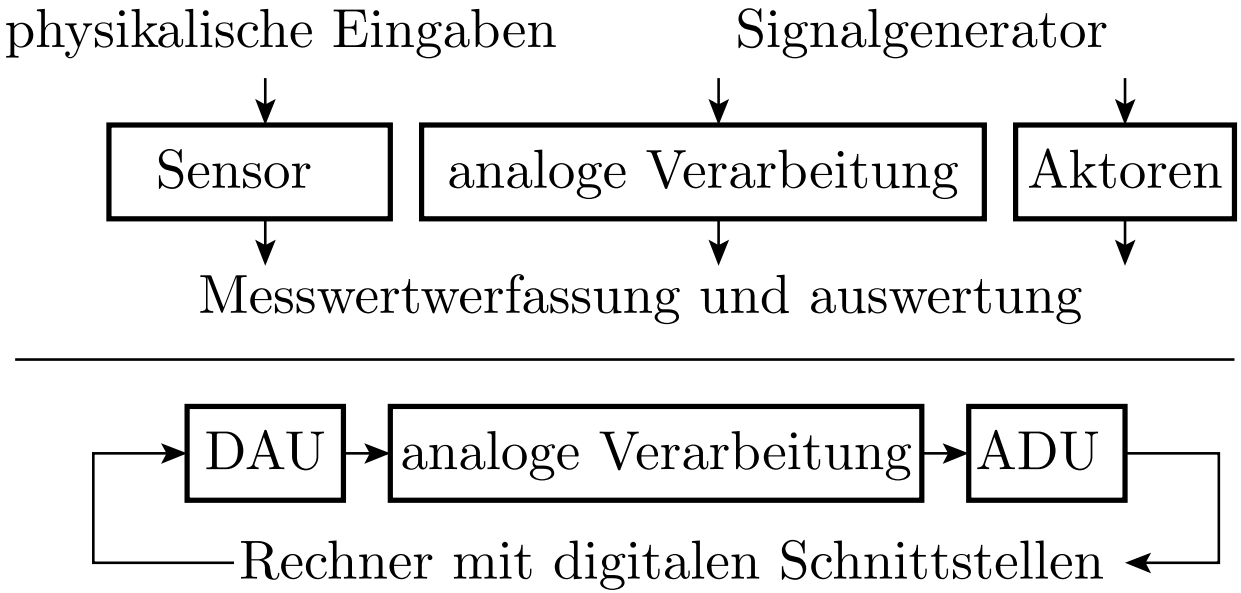
Modularar Test



Analoge Komponenten lassen sich entweder isoliert oder oder eingebettet in die Funktionsumgebung testen. Erfordert:

- Möglichkeiten der Kontaktierung und der Isolation von der Funktionsumgebung.
- Testhardware zur Bereitstellung von Eingaben und zur Auswertung von Ausgaben (physikalisch elektrisch, digital, Busprotokolle).
- Nachbildung der Systemumgebung (HIL – Hardware in the Loop).
- Teststeuerrechner.

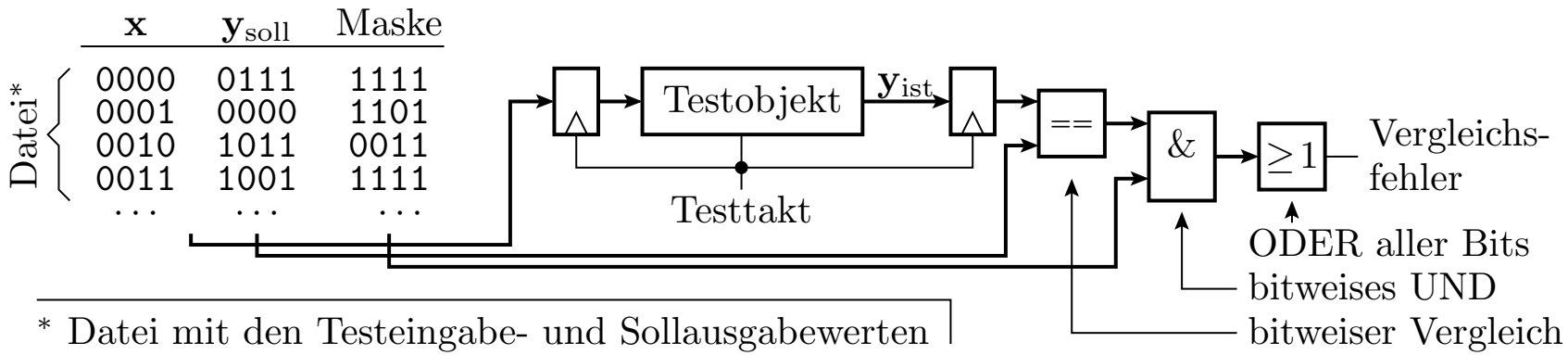
Test analoger Bausteine



Der Sensortest verlangt physikalische Eingaben, der Test von analogen Verarbeitungseinheiten und Aktoren elektrische Eingabesignale. Die Ausgabesignale sind aufzuzeichnen und auf zu testende Merkmal zu kontrollieren.

In einem System mit Rechner ADU, analoger Verarbeitung und DAU kann man Rechner, ADU und DAU nachdem sie getestet sind, mit als Testhardware nutzen.

Test digitaler Bausteine

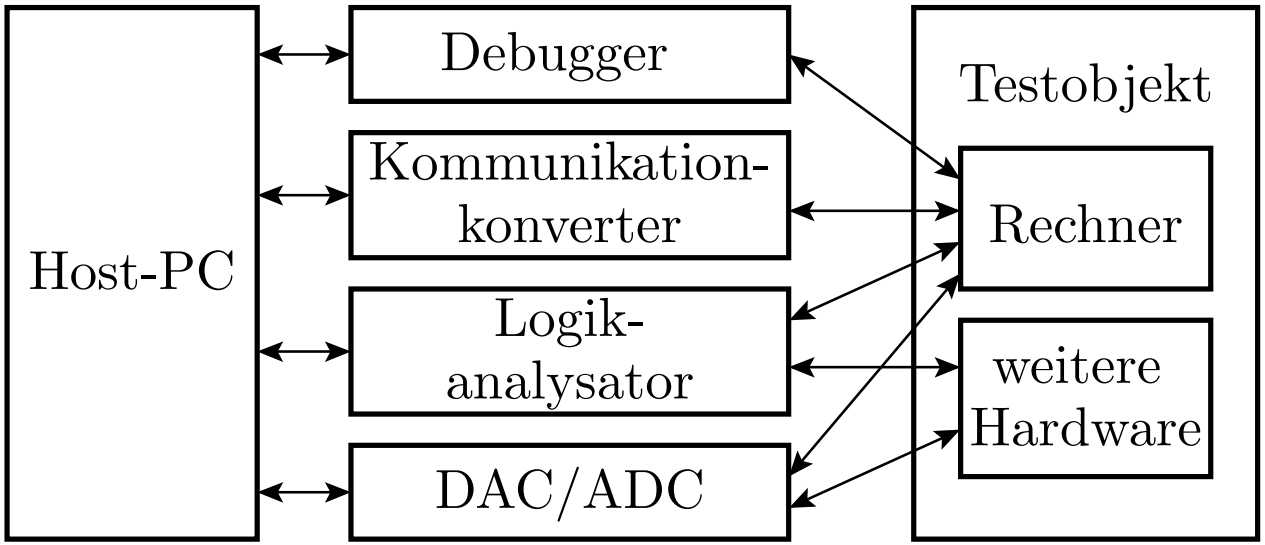


Eingabebereitstellung und Ausgabeüberwachung / -aufzeichnung.

- Manuell, nur Logikverhalten mit Schaltern und LEDs.
- Automatisch einschließlich Laufzeitkontrolle mit Signalgenerator und Logikanalysator[1].

[1] Aufzeichnung in der Regel mit mehrfachem Schaltungstakt. Bessere LA hohe Aufzeichnungsgeschwindigkeit und Erkennung/Darstellung »Glitches«, »ungültig« ... zur genaueren Diagnose des Laufzeitverhaltens.

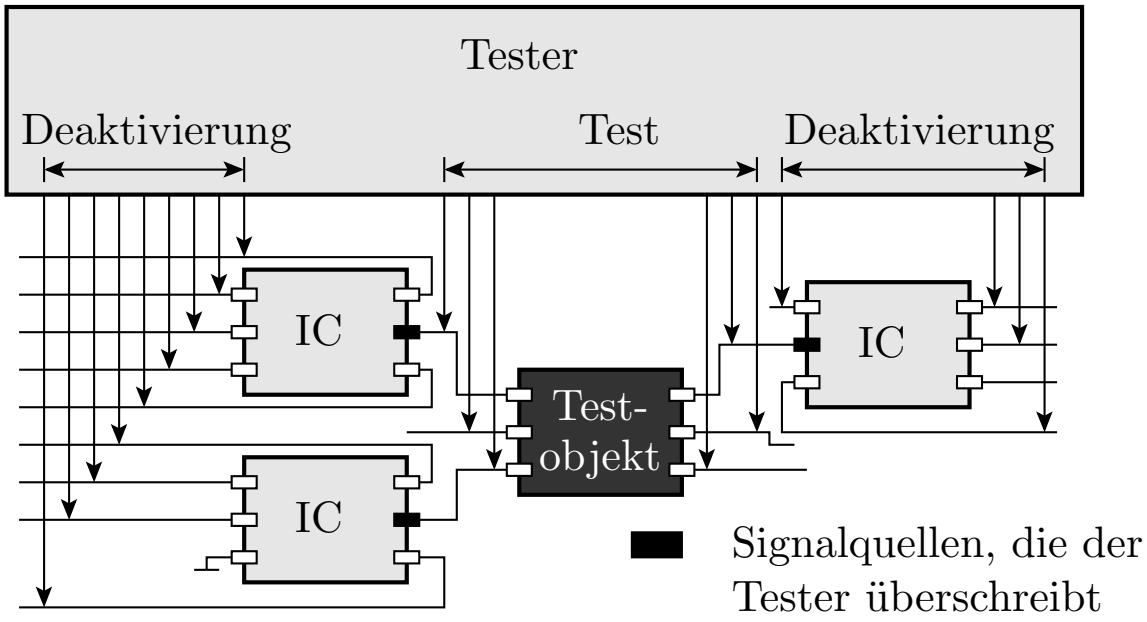
Test von Rechnersystemen



Der Test eines Rechnersystems benötigt zusätzlich zu Funktionseinheiten für die Bereitstellung der Testeingaben und Testausgaben (physikalisch, elektrisch, digital):

- Möglichkeiten zum Laden und Starten von Testprogrammen.
- Einen Debugger für Schrittbetrieb, Setzen von Haltepunkten, Trace-Aufzeichnung Lesen und Schreiben von Variablen und Speicherinhalten.

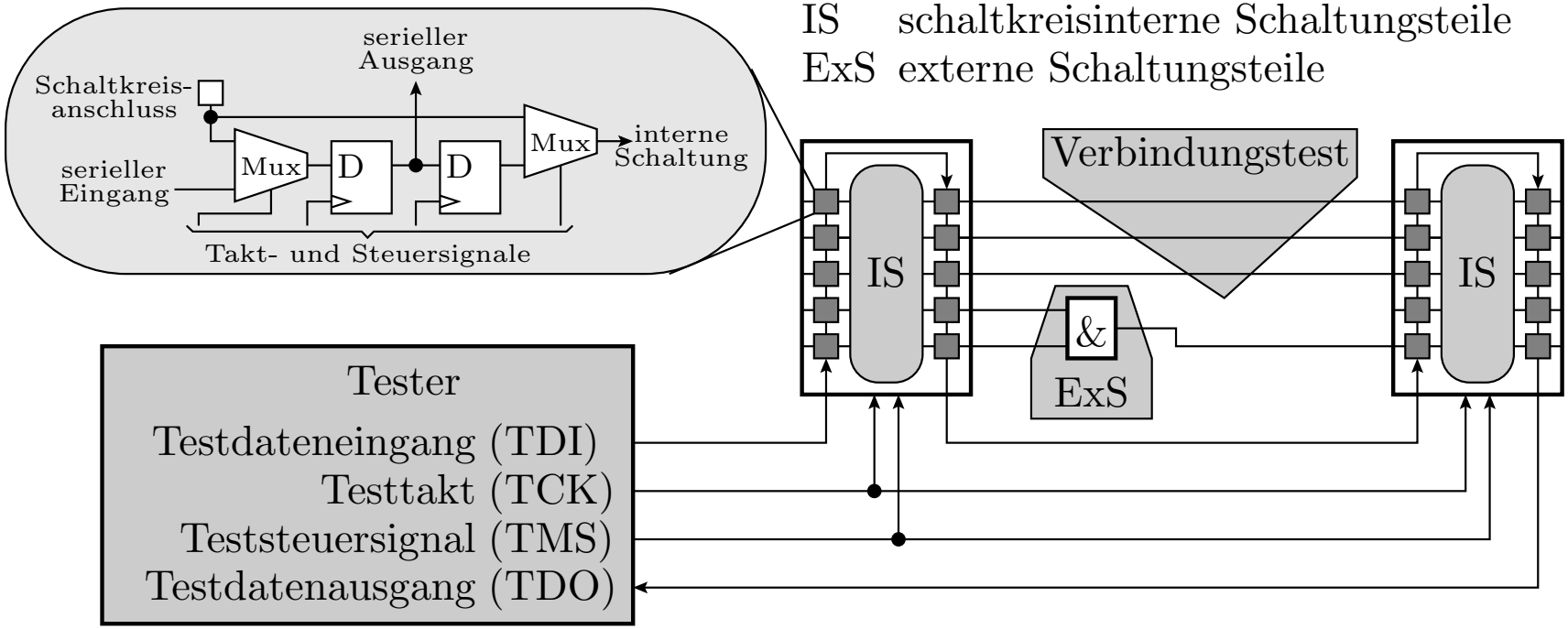
Digitaler In-Circuit-Test



- 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 werden möglichst deaktiviert (Anschlüsse hochohmig).

Testbus

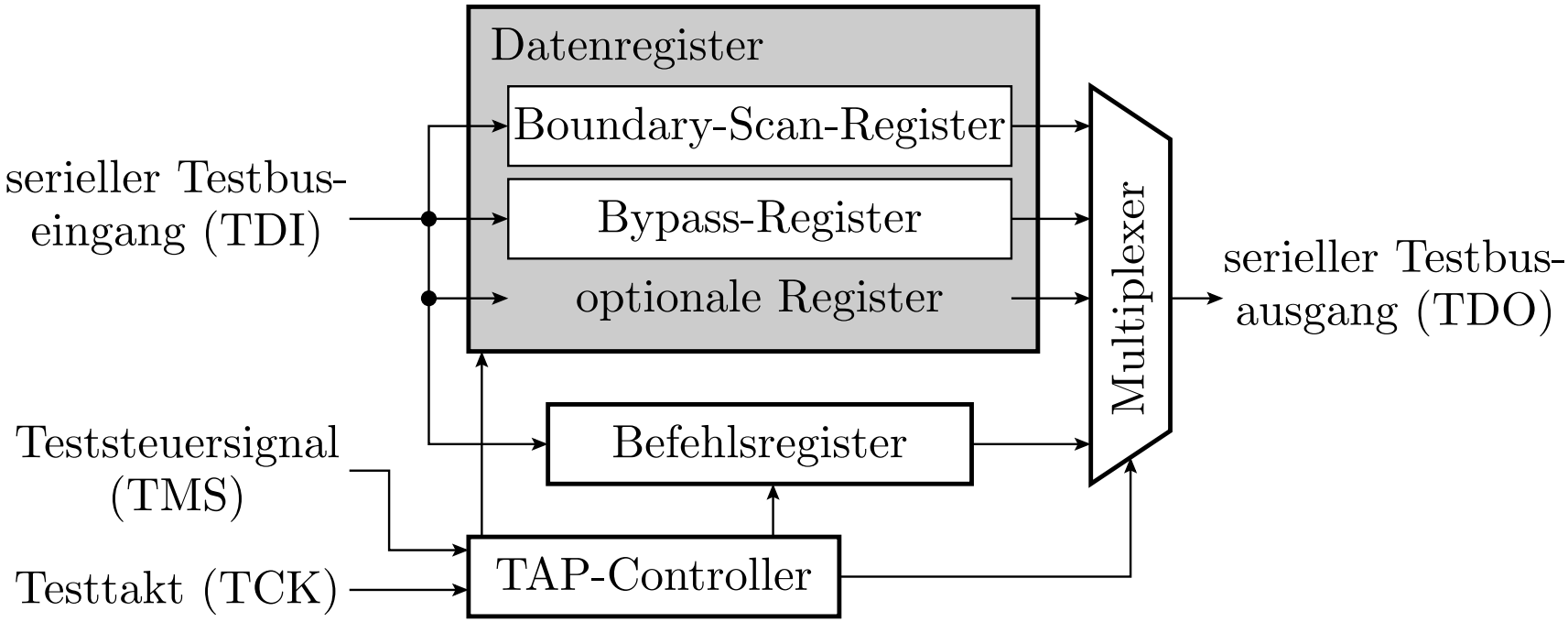
Testbus – JTAG (IEEE 1149.1)



Steuerung integrierter Testhilfen in Schaltkreisen und auf Baugruppen über einen seriellen Bus (minimale Anzahl von Verbindungsleitungen):

- Ersatz der mechanischen Nadeln für den Verbindungs-, Bestückungs- und isolierten Test integrierte Teststrukturen.
- Laden und Lesen von Programm- und Konfigurationsdaten, ...

JTAG-Testbusarchitektur der Schaltkreise



Eine Boundary-Scan-Implementierung umfasst:

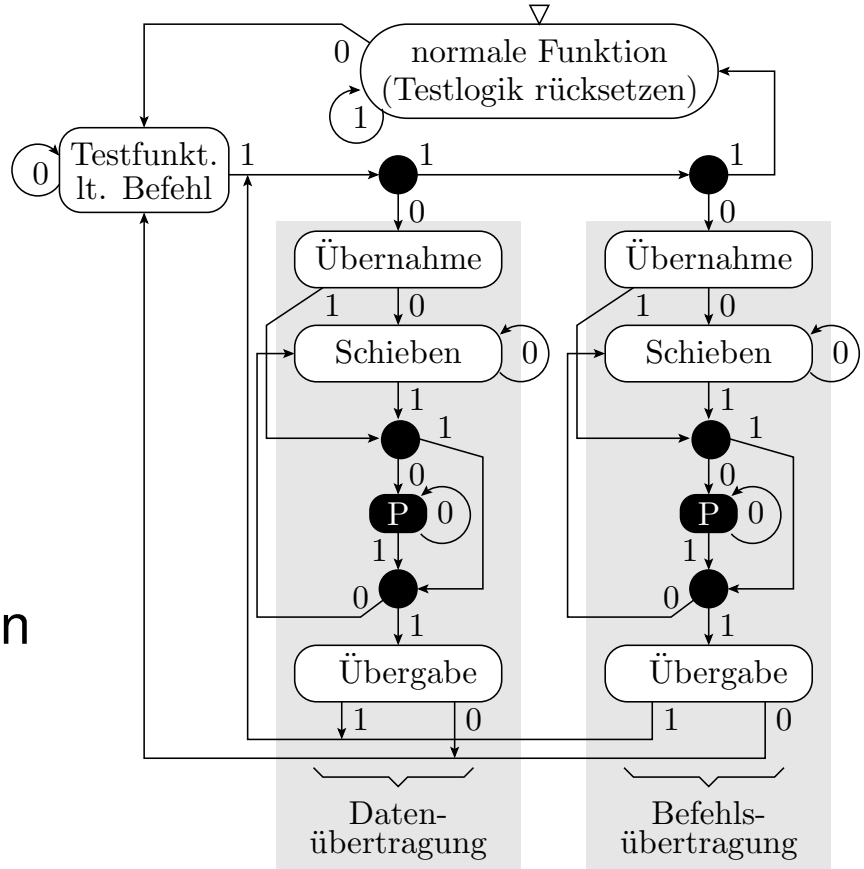
- den TAP- (Test Access Port) Controller
- ein Befehlsregister
- mehrere Testdatenregister (mindestens das Boundary-Scan- und das Bypass-Register).

Lesen und Schreiben der Testregister

- Automat mit 16 Zuständen
- Kantenauswahl über TMS- (**T**est **M**ode **S**elect) Signal

Typischer Testablauf:

- Befehlsregister lesen (Bestückungskontrolle[2]),
- Bauteilnummern lesen (Bestückungskontrolle),
- Einen Teil der Schaltkreise auf Bypass setzen. Für die anderen Datenregister auswählen.
- Verbindungstest.
- Adressregister Auswählen,
- Adressregister Schreiben,
- Adressierten Speicherplatz lesen, Datenregister auswählen, ...



[2]Das Befehlsregister übergibt beim Lesen ein Muster zur Erkennung von Unterbrechungen der Schieberegisterkette auf der Baugruppe.

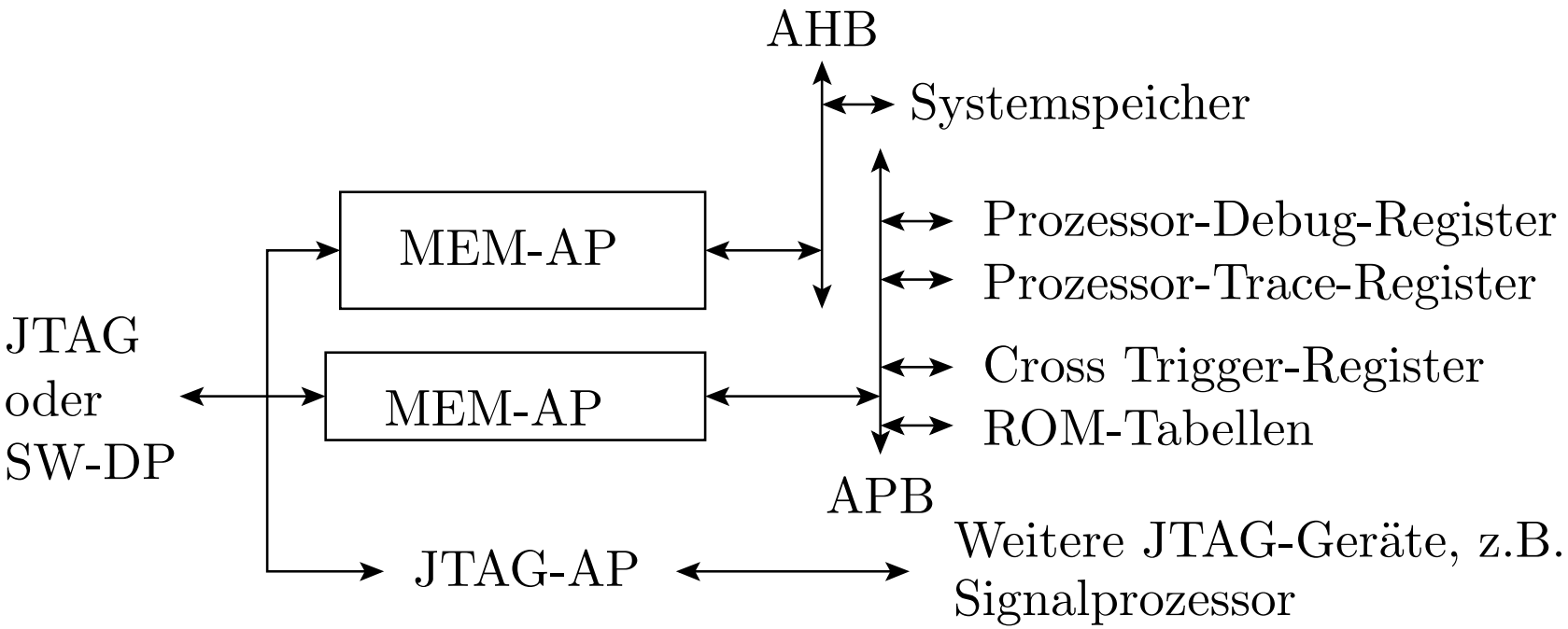
Über das Befehlsregister lässt sich eine

- beliebige Anzahl von Datenregistern adressieren und
- eine beliebige Anzahl von »Testfunktionen laut Befehl« steuern.

Typische unterstützte Testfunktionen:

- lesbare Hersteller- und Bauteilidentifikationsregister für den Bestückungstests.
- Steuerung von Steuerung 0, 1, hochohmig und Lesen der Logikwerte der IC-Anschlüsse für den Verbindungstest.
- Laden und Lesen von Programm- und Konfigurationsdaten.
- Steuerung integrierter Debugger-Funktionen,
- Steuerung von Selbsttests, ...

ARM-Testbusarchitektur



AHB	Advanced High Performance Bus
APB	Advanced Peripherial Bus
MEM-AP	Memory Acces Port
SW-DP	Serial-Wire Debug Port

Fehlermodellierung

Fehlermodellierung und Testauswahl

Dynamischer Tests: Ausprobieren der Systemfunktion mit einer Stichprobe von Beispieleingaben. Auswahlmöglichkeiten:

- Überprüfung aller Eingabemöglichkeiten.
- Suche von Eingaben zum Nachweis von Fehlern bzw. Fehlerannahmen.
- skalierbare Testalgorithmen für den vollständigen Fehlernachweis für ausgewählte Schaltungsklassen.
- zufällige Auswahl in Bezug auf die zu findenen Fehler.

Die Fehler in einem System sind immer erst nach ihrer erfolgreichen Beseitigung genau bekannt. Für vorausschauende Abschätzungen:

- potentielle (zu erwartende / mögliche) Fehler. In der Regel lassen sich nur Beispielmengen angeben.
- Modellfehler. Durch ein Fehlermodell definierte abzählbare Mengen von simulierbaren Fehlerannahmen, die ähnlich wie potentielle Fehler nachweisbar sind.

Test mit allen Eingabemöglichkeiten

Für den Nachweis, dass ein Service für alle Eingabedaten korrekte Ergebnisse liefert, müsste er mindestens mit allen Eingaben ausprobiert werden. Bereits ab wenigen Eingabebits unmöglich:

	m	2^m	t^*
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

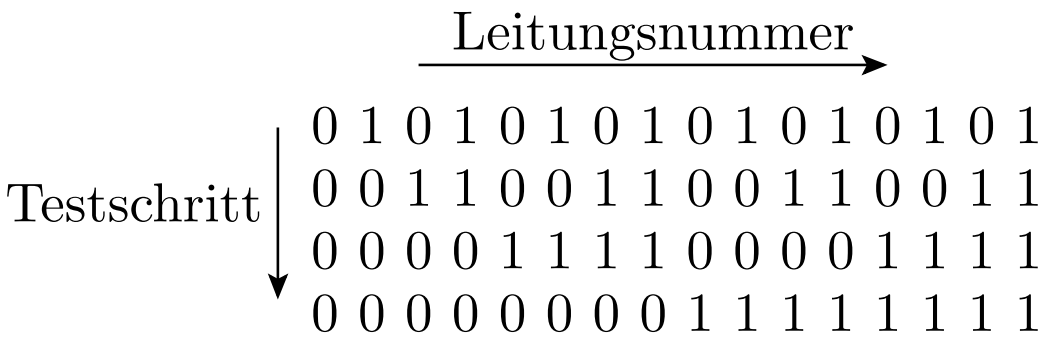
(m – Anzahl der Eingabebits; 2^m – Anzahl der Eingabemöglichkeiten; t^* – Testdauer bei einer Service-Ausführungszeit von 1µs.)

- Die meisten Systeme verarbeiten mehr als 128 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.
- Es gibt auch Fehler, die verlangen Eingabefolgen für den Nachweis. Alle Variationen von 2, 3, ... Eingabevektoren ...

Vollständiger Fehlernachweis

Für regelmäßig strukturierte Schaltungen lassen sich oft Algorithmen für die Eingabegenerierung so formulieren, dass alle Fehler nach einem bestimmten Fehlemodell nachweisbar sind. Beispiel Nachweis alle Kurzschlüsse auf einer Leiterplatte, wenn auf allen Leitungen 0 oder 1 gesteuert werden kann und die Leitungen beobachtbar sind.

Hinreichende Nachweisbedingung für alle Kurzschlüsse ist, dass sich die gesteuerten Leitungspegel paarweise in mindestens einem Testschritt unterscheiden.



Logarithmische Zunahme der Testsatzlänge mit der Anzahl der potentiell kurzgeschlossenen Leitungen.

Ein Beispiel für einenen Speichertest folgt später in Abschn. 3.7.

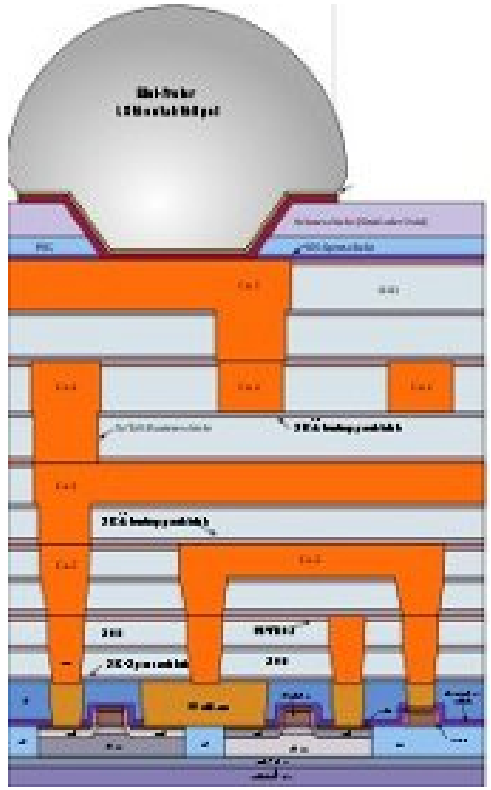
Schaltkreisfehler

Entstehung und Fehler integrierter Schaltkreise

Die Herausforderung für die Testauswahl sind hochintegrierte Schaltkreise mit ihren extrem vielen logischen Gattern und Verbindungen, die nicht von außen direkt steuerbar und beobachtbar sind.

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 weggeätztes) und
- überflüssiges Material (zu viel aufgetragen, zu wenige weggeätzt).

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 von Verbindungen,
- Kurzschlüsse zwischen benachbarten leitenden Gebieten.
- Transistoren, die nicht richtig ein- oder ausschalten,
- Leckströme ohne logische Fehlerwirkung.

Globale Fehler für Testauswahl uninteressant

Fehlerhafte Schichteigenschaften durch Prozesssteuerfehler:

- Überwachung auf Prozesssteuerfehler während der Fertigung.
- Kontrolle der Eigenschaften erzeugter Schichten nach Prozessschritten.
- Stichprobenkontrolle der Transistoreigenschaften, Leitwerte und Kapazitäten nach der Fertigung an speziellen Testschaltungen auf dem Wafer.
- Ausmessen der elektrischen Eigenschaften an den Anschlüssen incl. Versorgungsstrom.

Großflächig überflüssiges oder fehlenden Material:

- verursachen häufig FF oder komplette Funktionsunfähigkeit,
- erkennbar in der Regel beim Ausmessen der elektrische Anschlusseigenschaften oder vom sich anschließenden Grobtest.

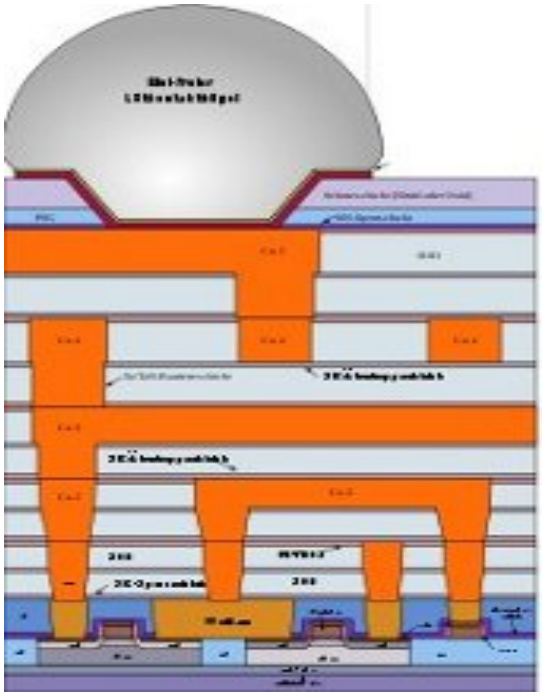
Anspruchsvoll ist die Suche nach kleinen Defekten, die überall sein können und nur selten FF verursachen.

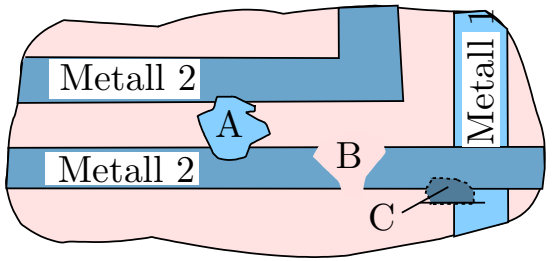
Lokale Fehler

Verbindungs- und Transistorfehler

Einzelfehler durch fehlendes und überflüssiges Material:

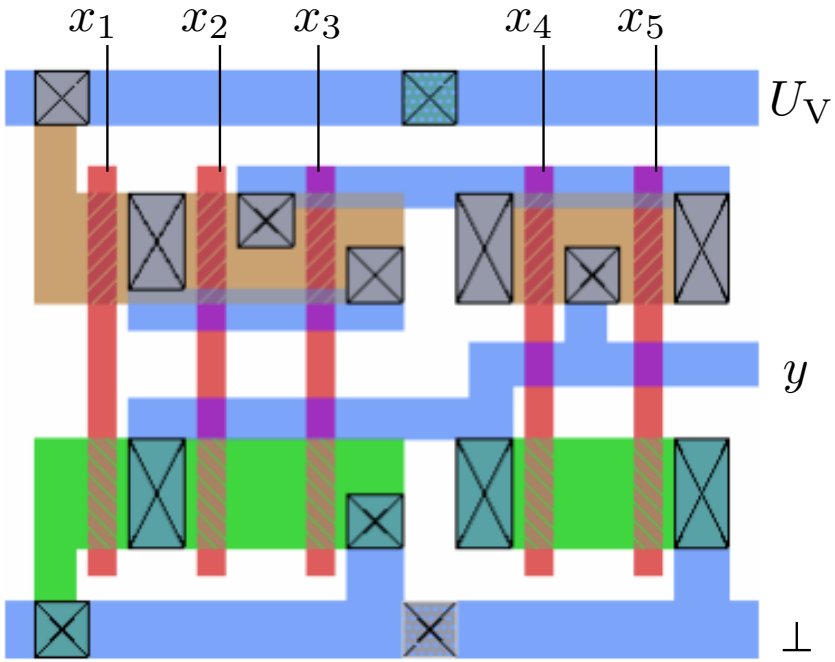
- kurzgeschlossene und unterbrochene Verbindung,
- nicht richtig ein- oder ausschaltende Transistoren,
- Leckströme ohne Beeinrächtigung der logischen Funktion, ...
- überhöhte Stromdichten oder Feldstärken, die zu Frühausfällen führen.

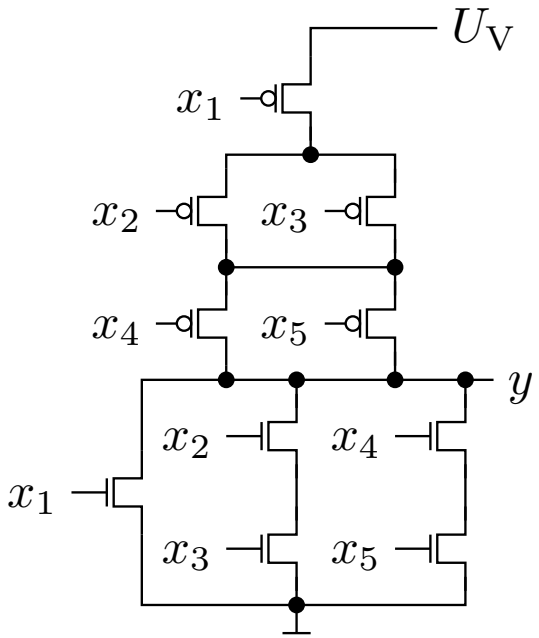




A zuätzliches Metall
B fehlendes Metall
C fehlende Isolation

Transistorebene



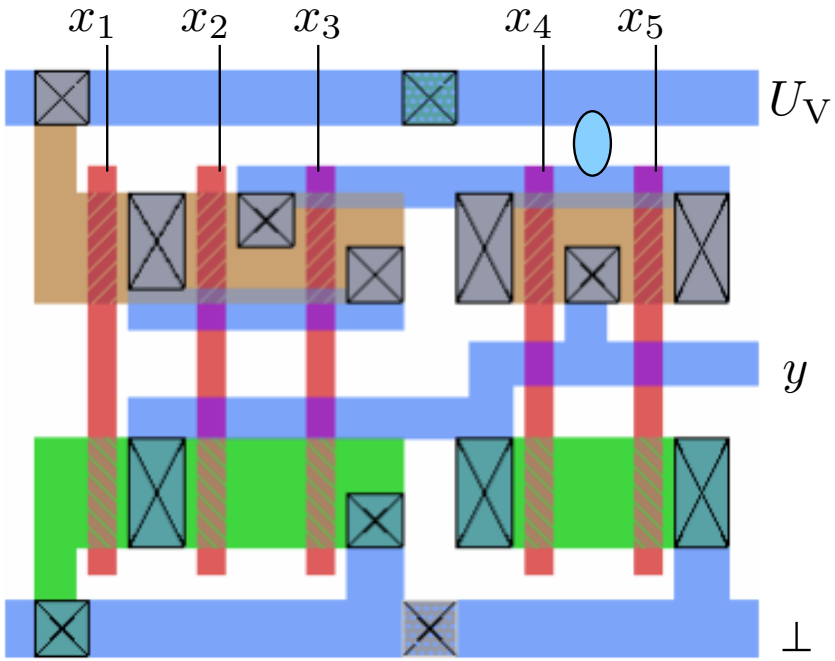


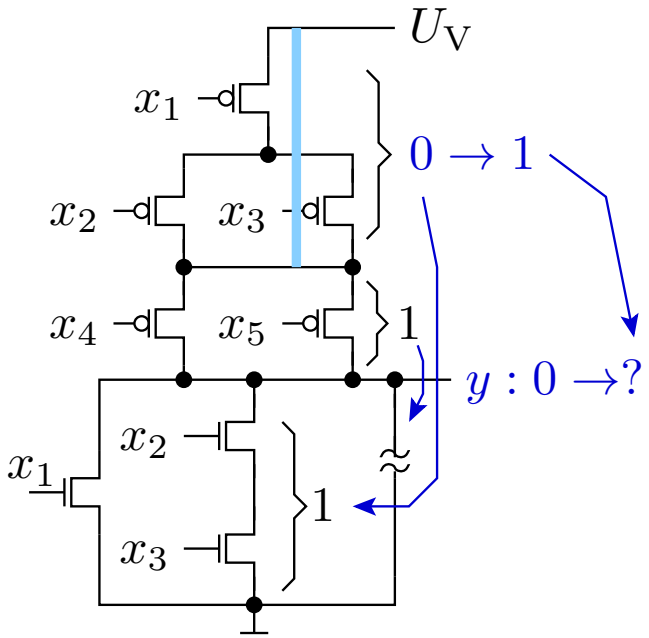
Durchkontaktierung
n-Gebiet
p-Gebiet
Polysilizium
Metall

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

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

Kurzschluss im Gatters

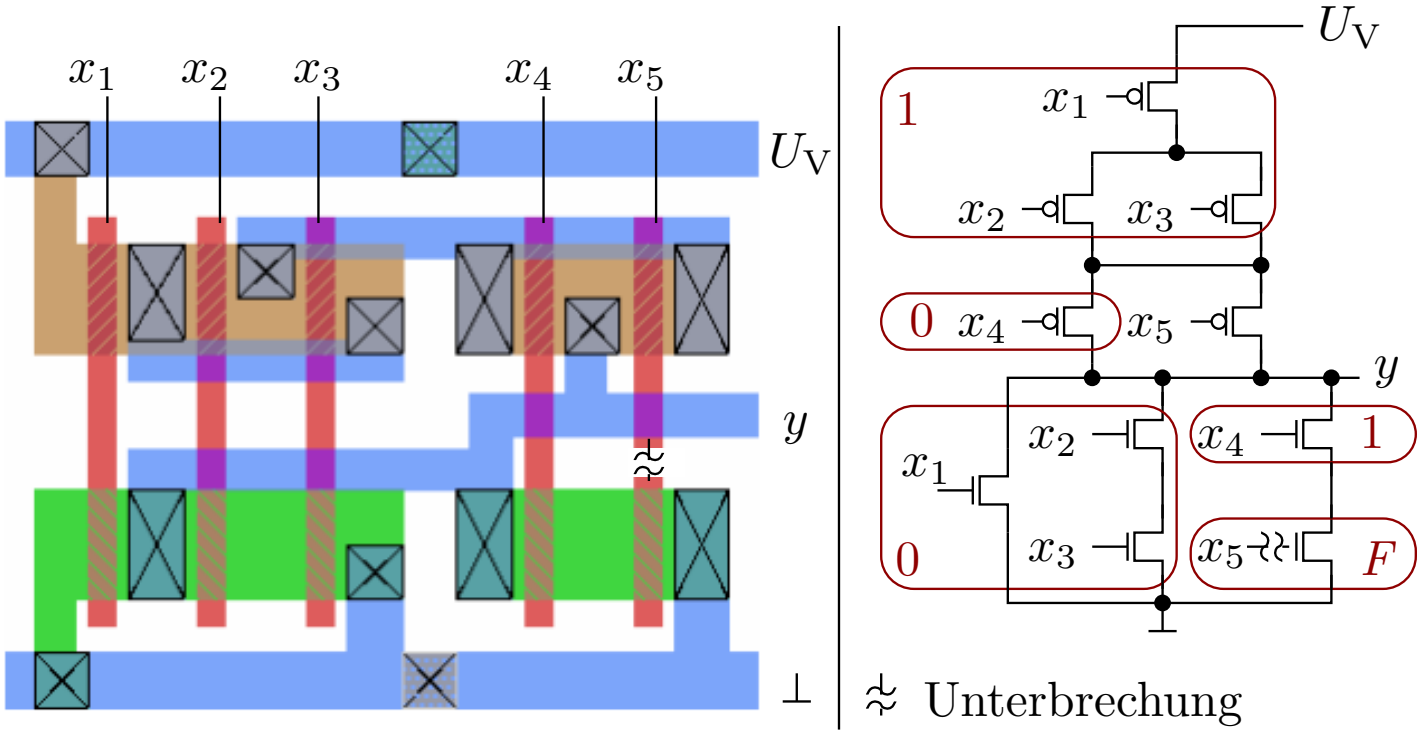




Nachweis bei $\bar{x}_1 \wedge (\bar{x}_2 \vee \bar{x}_3) = 0 \ \wedge \ \bar{x}_4 \vee \bar{x}_5 = 1$ durch

- statischen Ruhestrom (NMOS- und PMOS-Netzwerk gleichzeitig ein),
- eventuelle Verfälschung y von 0 nach 1.

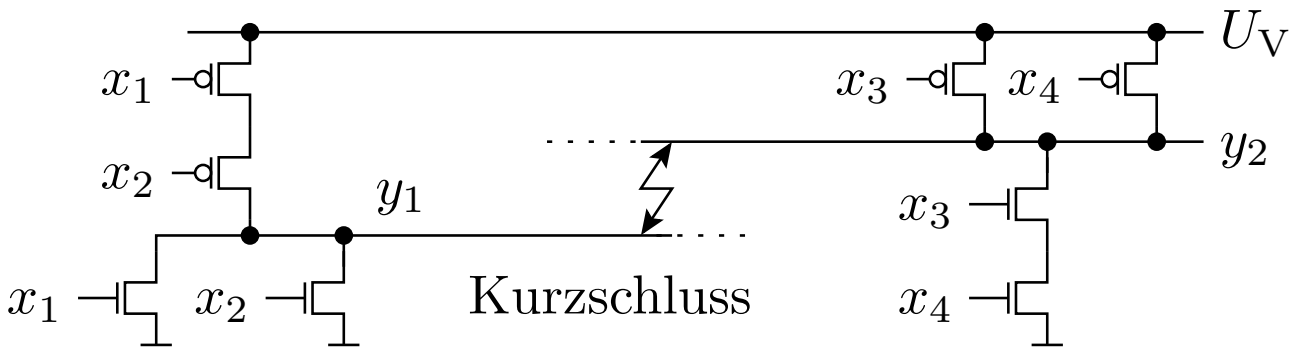
Offenes Gate



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

- wenn $F = 0$ zusätzlich $x_5 = 1$: kein Wechsel $y : \Downarrow \rightarrow 1$
- wenn $F = 1$ zusätzlich $x_5 = 0$: Ruhestrom [+ $y : 1 \rightarrow 0$]
- sicherer Nachweis: $y : \Uparrow\Downarrow$ durch $y \neq x_5$ nach 1. oder 2. Wechsel

Kurzschluss zweier Gatterausgänge



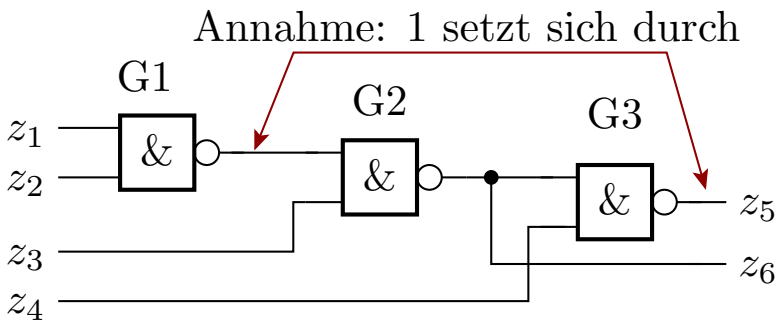
Mögliche Nachweisbedingungen:

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

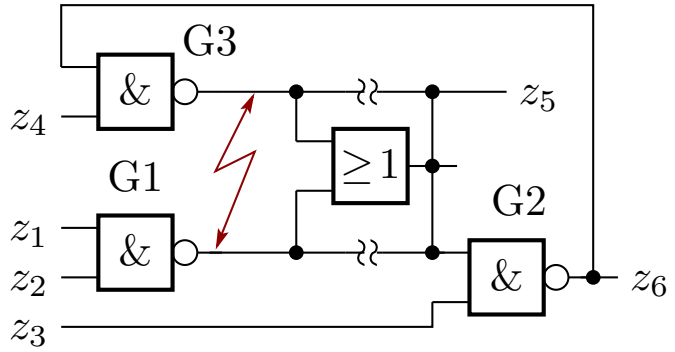
Ob sich dabei $y_1 = y_2 = 0$ oder $y_1 = y_2 = 1$ durchsetzt, hängt von den Transistorbreiten, bzw. Transistorsteilheiten ab.

Zusätzliches Speicherverhalten durch Kurzschluss

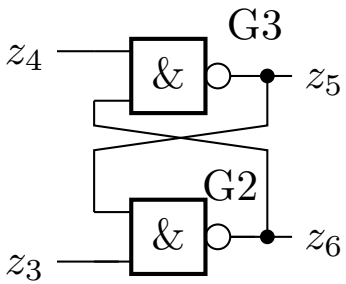
Schaltung mit Kurzschluss

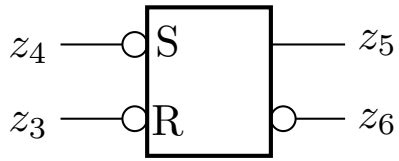


Kurzschlussnachbildung durch ein ODER



Ersatzschaltung für $z_1 = z_2 = 1$



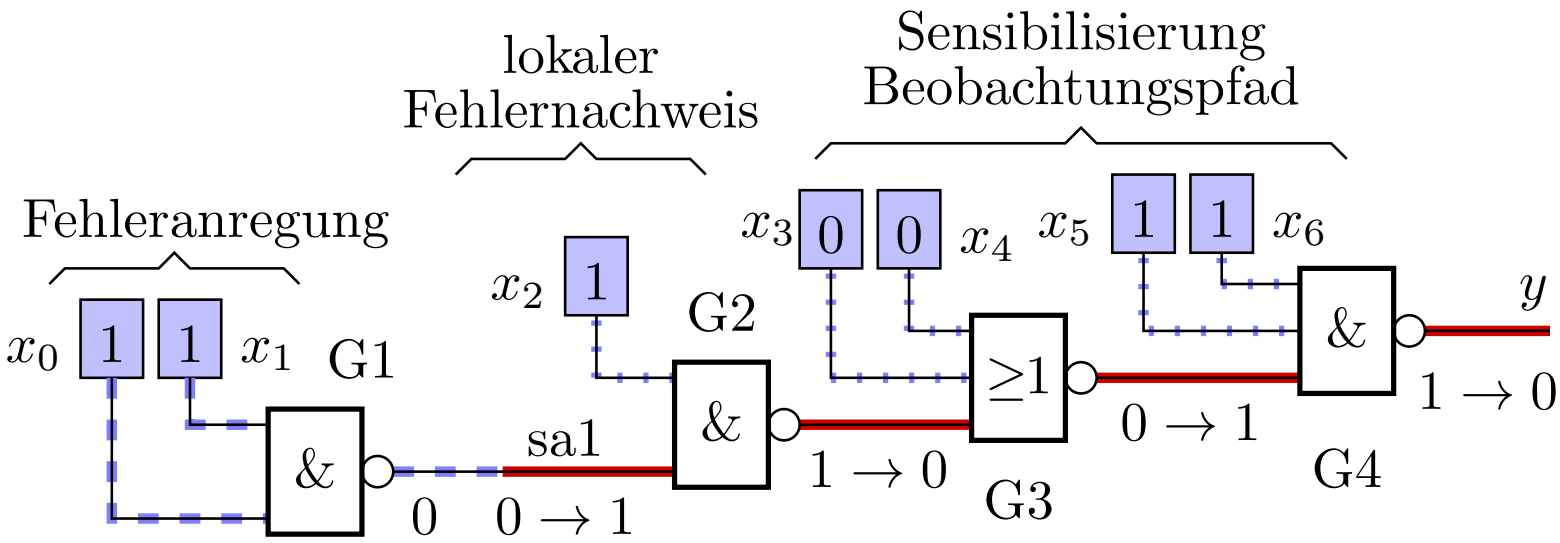


z_4	z_3	z_2	z_1	z_6	z_5	
0	1	1	1	0	1	(setzen)
1	0	1	1	1	0	(löschen)
1	1	1	1	speichern		

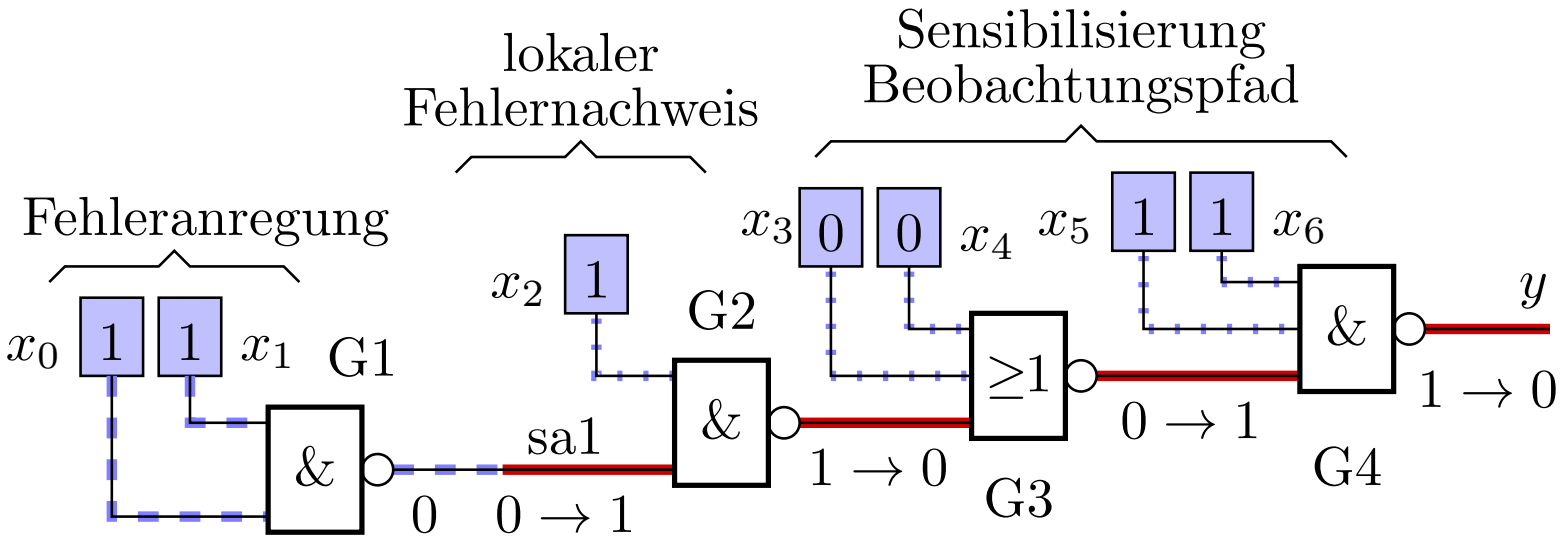
z_4	z_3	z_2	z_1	z_6	z_5	
0	0	1	1	1	1	
↓	↓			↓	↓	ändern nach
1	1	1	1	unbestimmt		

Reale und Haftfehler

Fehlernachweis im Schaltungsverbund



- Fehleranregung: Eingaben zur Einstellung der erforderlichen Eingaben am fehlerhaften Teilsystem.
- Lokale Fehlernachweis: Abbildung des lokalen Fehlers auf eine Verfälschung.
- Beobachtbarkeit: Sensibilisierung eines Pfads, der eine lokale Verfälschung zu einem Ausgang weiterleitet.



Lokale Fehler:

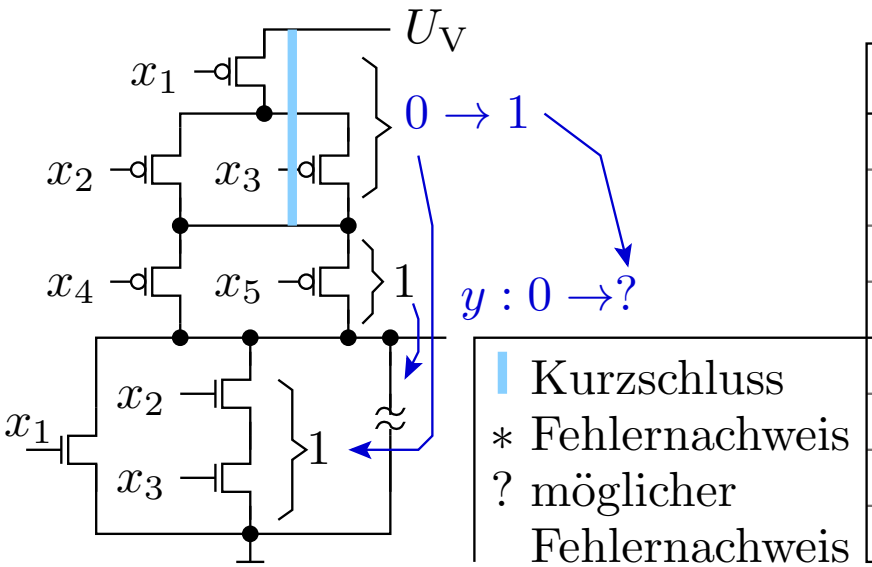
- Gatter mit Kurzschlüssen und Unterbrechungen,
- Unterbrochene Verbindungen,
- Teilschaltungen mit Kurzschlüssen, ...

Vereinfachte Modellfehler:

- Haftfehler: Gatteranschlusses ständig 0 oder ständig 1, ...
- Verzögerungsfehler: verzögerte Änderung $0 \to 1$ oder $1 \to 0$, ...

Für jeden lokalen Fehler gibt es vereinfachte Modellfehler mit gleichen Anregungs- und Beobachtungsbedingungen.

Kurzschlussnachweis mit Haftfehlertests



$x_3\,x_2\,x_1$		sa0(x_3)	sa0(x_2)	sa0(x_1)
0 0 0				
0 0 1	?			∗
0 1 0				
0 1 1	?			∗
1 0 0				
1 0 1	?			∗
1 1 0	?∗	∗	∗	
1 1 1	?			

Statischer Ruhestrom und eventuell $y : 0 \to 1$ für $y_{\text{soll}} = 0$ und ($x_4 = 0$ und $x_5 = 0$). Haftfehler, die auch nur unter den Bedingung $y_{\text{soll}} = 0$ oder ($x_4 = 0$ und $x_5 = 0$) $a = \bar{x}_4 \vee \bar{x}_5$ an $y : 0 \to 1$ nachweisbar sind:

Haftfehler	zusätzliche Nachweisbedingung
sa0 (x_1)	$x_1 = 1$ und ($x_2 = 0$ oder $x_3 = 0$)
sa0 (x_2)	$x_2 = 1$ und $x_1 = 0$ und $x_3 = 1$
sa0 (x_3)	$x_3 = 1$ und $x_1 = 0$ und $x_2 = 1$

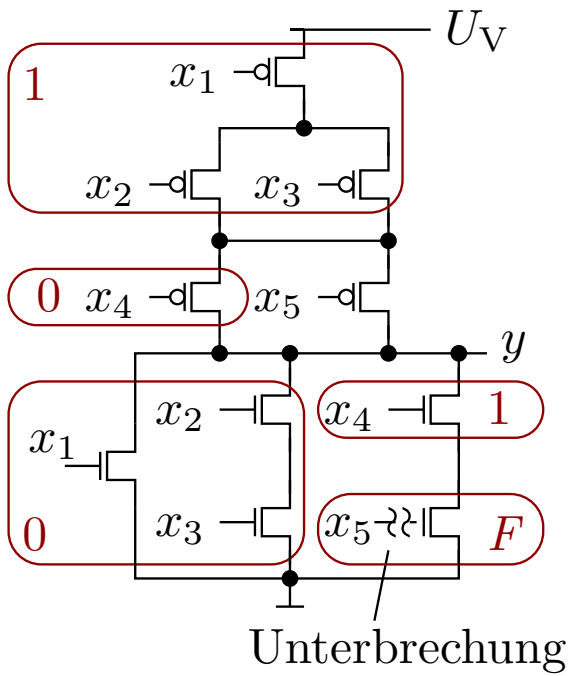
»Offenes Gate« mit Haftfehlertests

Wenn $a = \bar{x}_1 \wedge (\bar{x}_2 \vee \bar{x}_3) \wedge x_4$ und

- $y = 0$ und $x_5 = \Uparrow$: kein $y = \Downarrow$
- abgetrennter Transistor dauerhaft an und $x_5 = 0$: eventuell kein $y = \Uparrow$

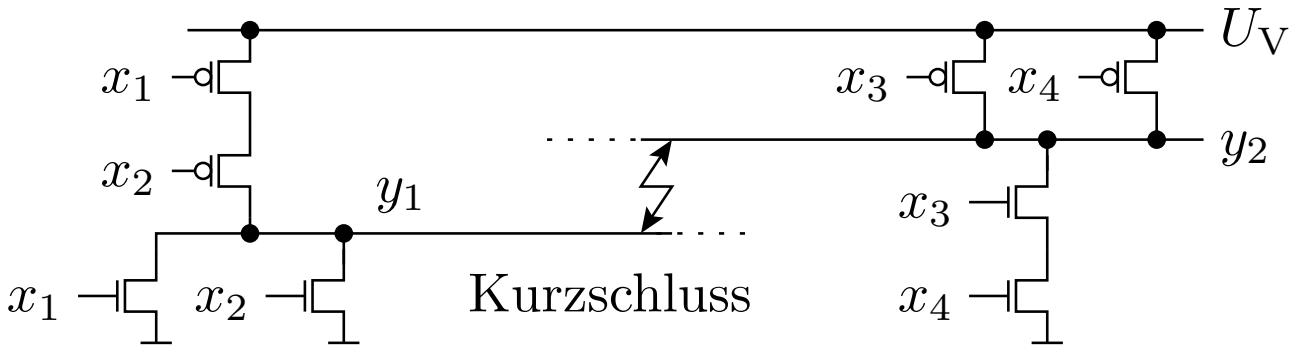
Ähnlich nachweisbare Haftfehler:

- sa0 (x_5) wenn $x_5 = \Uparrow$
- sa1 (x_5) wenn Reihenschaltung NMOS x_5, x_4 niederohmiger als Reihenschaltung PMOS x_5, x_2 [$\|$ x_3], x_1



(a – gemeinsame Anregungsbedingung; $\Uparrow$ – Wechsel 0 nach 1; $\Downarrow$ – Wechsel von 1 nach 0). Zu beiden Nachweismöglichkeiten gibt es einen Haftfehler, bei derem Nachweis die Unterbrechung am Gate unter je einer Zusatzbedingung auch nachgewiesen wird.

Kurzschluss zweier Gatterausgänge



Nachweis über statische Stromaufnahme: $y_1 \neq y_2$

Bedingungen für einen möglichen logischen Nachweis:

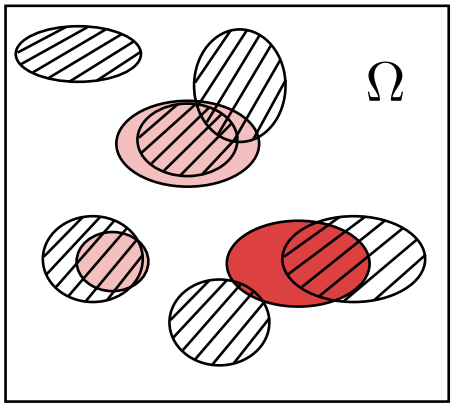
- y_1 und y_2 müssen sich im fehlerfreien Fall unterscheiden.
- Je nach Fehlerwirkung müssen y_1 oder y_2 dabei beobachtbar sein.

Ähnlich nachweisbare Haftfehler:

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

FC und Modell-FC

Fehler und Modellfehler



Ω	Menge aller Testeingabewerte
	Eingabewerte, die den Fehler nachweisen
	Eingabewerte, die den Fehler eventuell (unter Zusatzbedingungen) nachweisen
	Nachweismenge ähnlich nachweisbarer Modellfehler

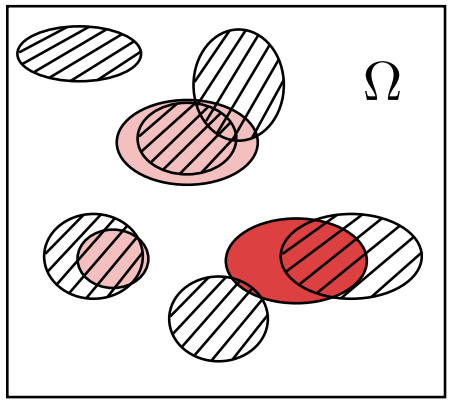
Für alle logisch nachweisbaren lokalen Fehler gibt es ähnlich nachweisbare Haftfehler mit bezüglich logischer Signalwerte

- übereinstimmenden Anregungsbedingungen,
- übereinstimmenden Beobachtungsbedingungen und
- nicht leeren Schnittmengen für der lokalen Nachweis,

unter denen der Fehlernachweis möglich oder sicher ist.

Fehlerüberdeckung bei fehlerorientierter Suche

Für jeden Modellfehler j wird ein Test gesucht und bei Erfolg $a_j \geq 1$ Tests gefunden. Jeder der a_j Tests weist Fehler i mit Wahrscheinlichkeit p_{ij} nach:



$$p_i = 1 - \prod_{j=1}^{n_i} (1 - p_{ij})^{a_j}$$

(n_i – Anzahl der ähnlich nachweisbaren Modellfehler zu Fehler i). Fehlerüberdeckung

$$FC = \frac{1}{\#F} \cdot \sum_{i=1}^{\#F} p_i$$

($\#F$ – Anzahl der tatsächlichen Fehler). Zwei Modellrechnungen mit $p_{ij} = p = 50\%$, Verteilung $\mathbb{P}[X = k]$ mit k gleich der Anzahl der Tests für alle ähnlich nachweisbaren Fehler zusammen :

$$FC = 1 - \frac{1}{\#F} \cdot \sum_{i=1}^{\#F} (1-p)^{k_i = \sum_{j=1}^{n_i} a_i} = \sum_{k=1}^{k_{\max}} \mathbb{P}[X = k] \cdot (1-p)^k$$

: ...

Fehlerüberdeckung bei fehlerorientierter Suche

Zwei Modellrechnungen mit $p_{ij} = p = 50\%$, Verteilung $\mathbb{P}[X = k]$ mit k gleich der Anzahl der Tests für alle ähnlich nachweisbaren Fehler zusammen:

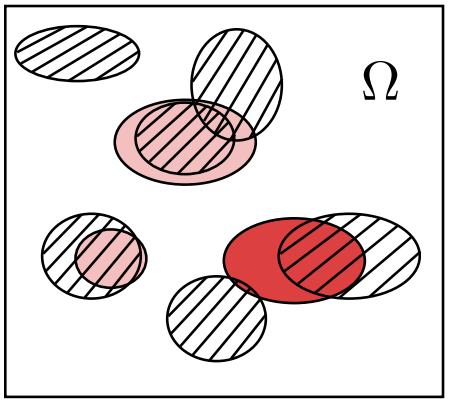
$k = n_i \cdot a_j$	0	1	2	3	4	5	6	7	8
A	1%	2%	50%	40%	7%	0			
B	3%	4%	7%	10%	15%	20%	23%	10%	8%

$$FC_{\mathrm{M}} \approx 1 - \mathbb{P}[k = 0]\,; \quad FC = 1 - \sum_{k=1}^{k_{\max}} \mathbb{P}[X = k] \cdot 0{,}5^k$$

A:	$FC_{\mathrm{M}} \approx 99\%$	$FC = 90\%$
B:	$FC_{\mathrm{M}} \approx 97\%$	$FC = 95\%$

Bei fehlerorientierter Testsuche ist es wichtiger, für jeden Modellfehler mehrere unterschiedliche Tests zu suchen, als für jeden Modellfehler mindestens einen Test zu finden.

Fehlerüberdeckung bei zufälliger Auswahl



Die Nachweismengen lokaler Fehler sind wegen der geteilten Anregungs- und Beobachtungsbedingungen ähnlich groß, wie im Mittel die der ähnlich nachweisbaren Modellfehler. Berücksichtigung tendenzieller Unterschiede durch eine Skalierungsfaktor c:

$$h_{\mathrm{M}}(\zeta) \approx h(c \cdot \zeta)$$

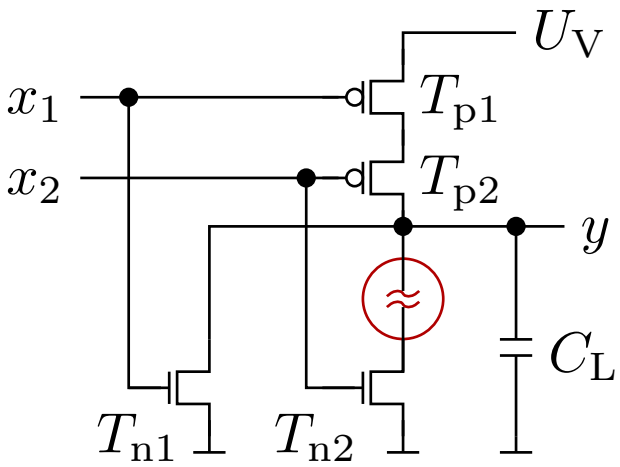
Wenn die ähnlich nachweisbaren Modellfehler tendentiell mehr FF verursachen als die tatsächlichen Fehler $c < 1$ sonst $c \geq 1$. Nach Foliensatz 4, Abschn. 3 »FF-Rate im Einsatz« ist die Fehlerüberdeckung für Testsatzlänge n etwa die der Modellfehlerüberdeckung $c \cdot n$–fachen Testsatzlänge:

$$FC(n) \approx FC_{\mathrm{M}}(c \cdot n)$$

Bei zufälliger Testauswahl ist die Fehlerüberdeckung einfacher und genauer aus der Modellfehlerüberdeckung vorhersagbar als bei gezielter Suche.

Verzögerungsfehler

Verzögerungsfehler, Beispiel Stuck-Open-Fehler



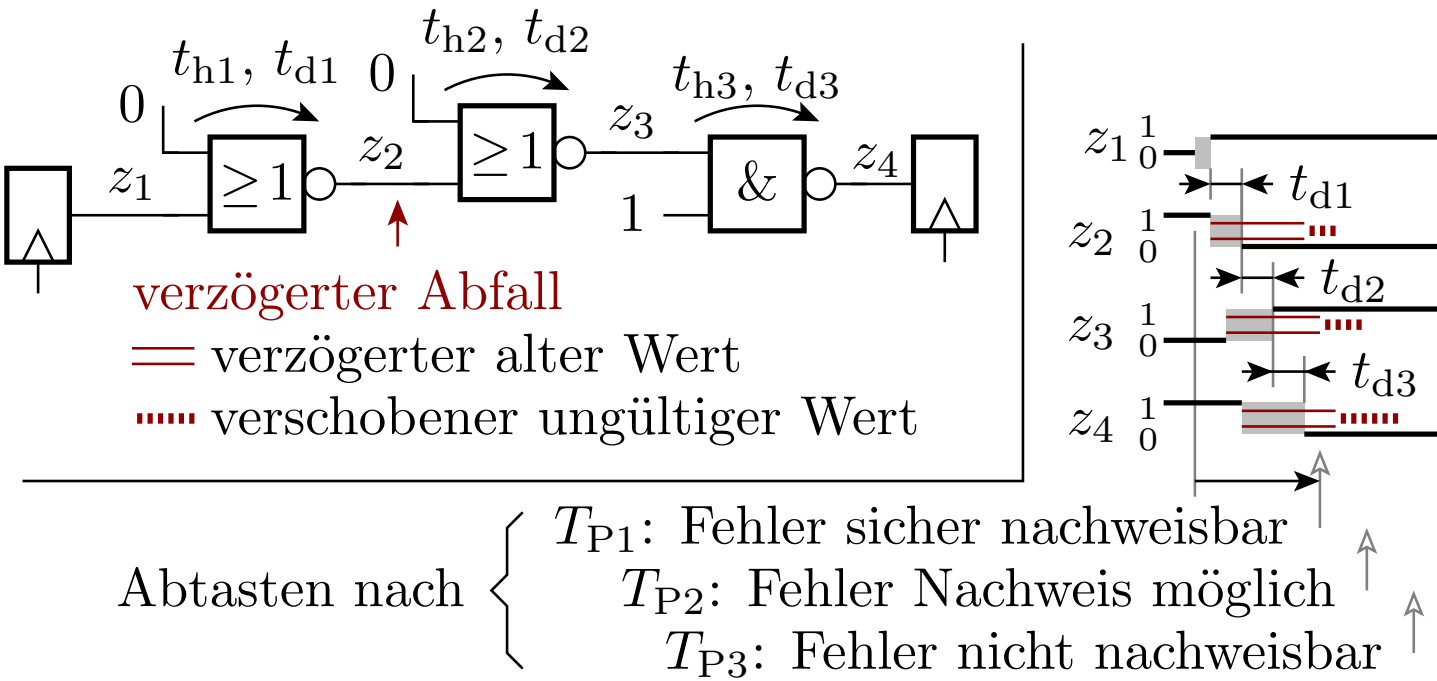
x_2	x_1	y
0	0	1 (setzen)
0	1	0 (rücksetzen)
1	0	speichern
1	1	0 (rücksetzen)

x_2	x_1	y
0	0	1
0	1	0
0	0	1
1	0	0

Grob die Hälfte der zu findenden Schaltkreisfehler sind Unterbrechungen und nicht richtig schaltende Transistoren, nachweisbar oft auch oder nur an einer erhöhten Verzögerung.

Die dargestellte Unterbrechung eines Parallelzweigs wird als Stuck-Open-Fehler bezeichnet und bewirkt ein Speicherverhalten. Nachweisbar an der stark erhöhten Entladezeit von C_L über T_{n2}.

Nachweis von Gatterverzögerungsfehlern



Verzögerter Signalanstieg oder -abfall. Abtastwert nach Abtastzeit:

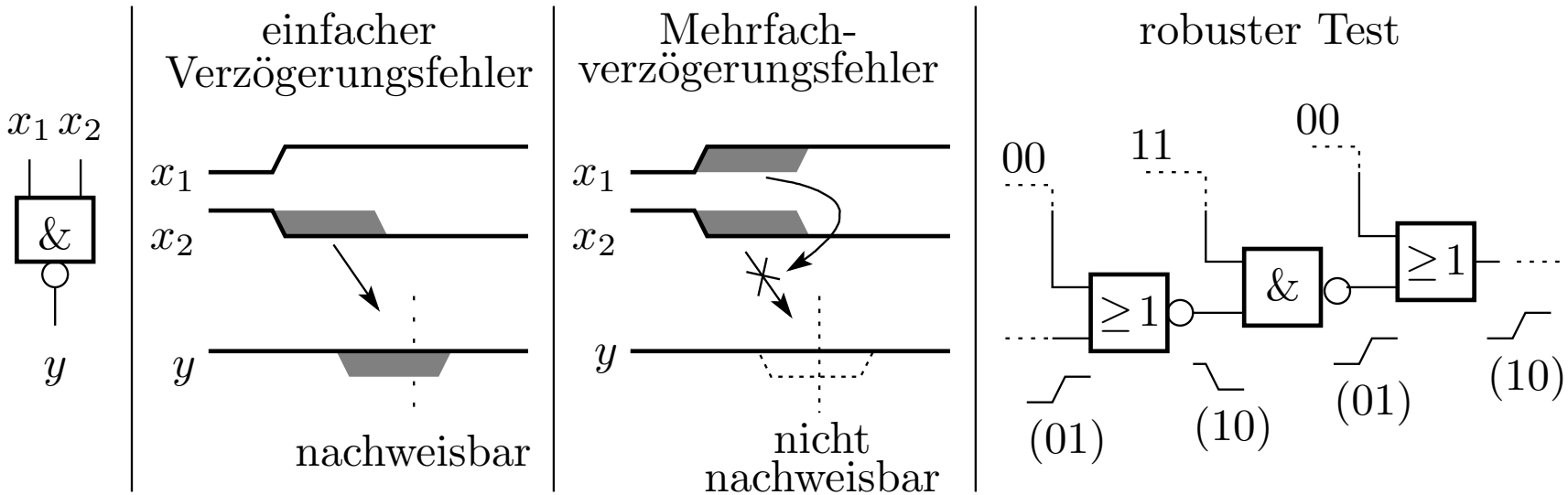
- T_{P1} ohne Fehler neue, sonst alter Wert,
- T_{P2} ohne Fehler neuer, sonst ungültiger Wert,
- T_{P3} ohne und mit Fehler neuer Wert.

(Robuster) Zwei-Pattern-Test

Der Fehlernachweis erfordert zwei Eingaben:

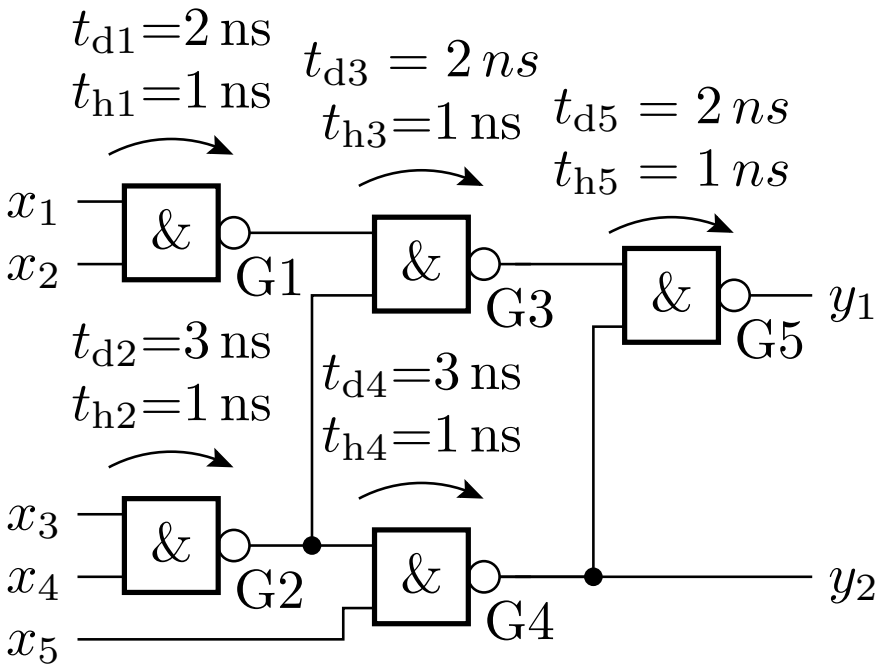
- Einstellung 0 für $\Uparrow$ und 1 für $\Downarrow$ am Fehlerort.
- Haftfehlertest sa0 für $\Uparrow$, sa1 für $\Downarrow$ am am Fehlerort.

($\Uparrow$ – verzögerter Anstieg; $\Downarrow$ – verzögerter Abfall).



Robuster Test: Je Testschritt max. eine Signaländerung an den Eingängen jedes Gatters.

Test aller Gatter über den längsten Pfad

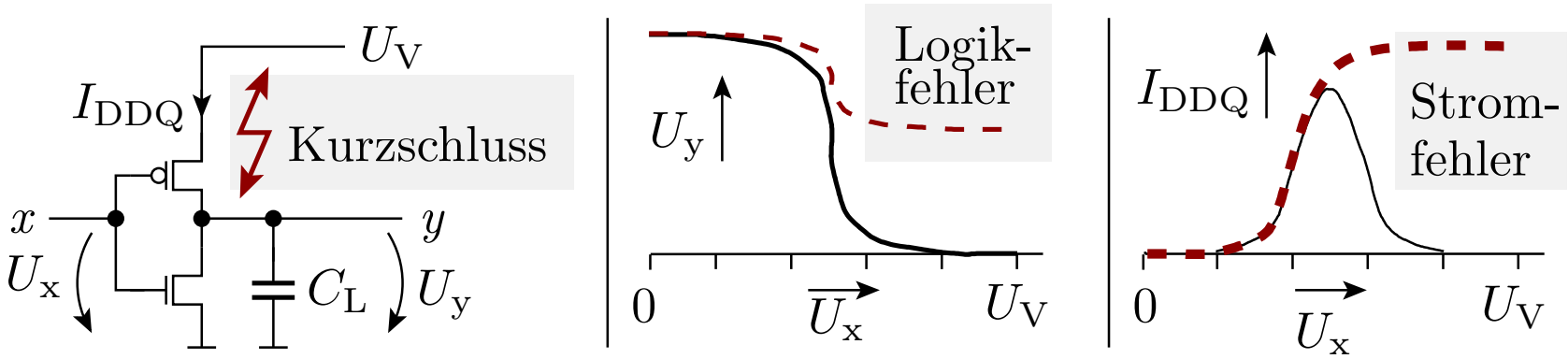


Pfade	$\sum t_{h.i}$	$\sum t_{d.i}$
G1-G3-G5	3 ns	6 ns
G2-G3-G5	3 ns	7 ns
G2-G4-G5	3 ns	8 ns
G2-G4	2 ns	6 ns
G4-G5	2 ns	5 ns
G4	1 ns	3 ns

Die minimal erkennbare Zusatzverzögerung ist die Differenz aus Taktperiode und Soll-Verzögerung. Test auf Gatterverzögerungsfehler vorzugsweise über die Pfade mit der längsten Sollverzögerung.

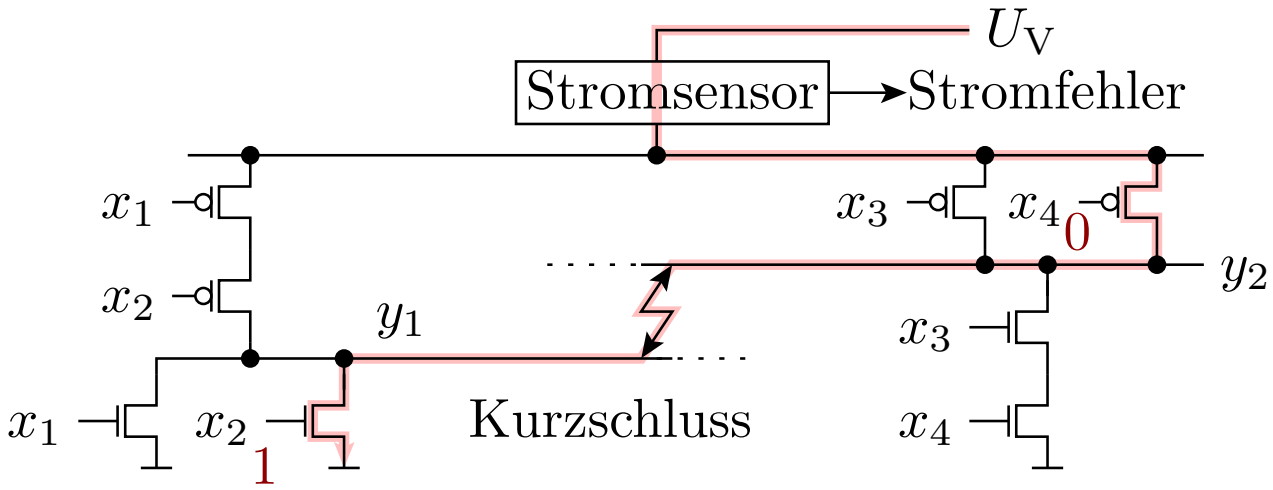
IDDQ-Test

Ruhestromüberwachung



In einer CMOS-Schaltung ist der Gatterausgang nur entweder über NMOS-Transistoren mit 0 (Masse) oder über PMOS-Transistoren mit 1 (Versorgungsspannung) verbunden. Nach jedem Schaltvorgang klingt der Strom auf einen sehr kleinen Wert ab. Die Hälfte der zu erwartenden lokalen Schaltkreisfehler sind Kurzschlüssen, nicht richtig ausschaltende Transistoren, ... die, wenn sie zur Wirkung kommen, ein messbaren Ruhestrom I_{DDQ} verursachen. I_{DDQ}-Überwachung erkennt Defekte auch ohne Beobachtungspfad zu einem Ausgang.

Kurzschlussnachweis über den Ruhestrom



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

- Vorteile I_{DDQ}–Test: Einfachere logische Nachweisbedingungen, einfachere fehlerorientierte Testsuche, kürzere Zufallstetest bei gleicher Fehlerüberdeckung.
- Probleme I_{DDQ}–Test: Unterscheidung von zulässigem und überhöhtem Ruhestrom funktioniert nur bis einige tausend Gatter. Integrierte Stromsensoren, ...

Eignung Fehlermodelle

Kriterien für die Eignung von Fehlermodellen

Fehlermodelle sind praxistauglich, wenn:

1. die Anzahl der vom Fehlermodell erzeugten Modellfehler maximal lineare mit Testobjektgröße zunimmt
2. die Fehlerannahmen sich auf der Logikebenen simulieren lassen.
3. die bedingten Wahrscheinlichkeiten, dass ein Test für einen Modellfehler auch ein Tests für einen richtigen Fehler ist, und umgekehrt nicht zu klein sind.

Geeignet:

- Haftfehler $\surd$,
- Gatterverzögerungsfehler $\surd$,
- IDDQ-Fehler $\surd$,
- Zellenfehler für Block Speicher $\surd$, ...

Ungeeignet für Fehlermodellierung

- Mehrfachfehler: überproportionale Zunahme der Fehlermöglichkeiten mit der Systemgröße.
- Kurzschlüsse und Unterbrechungen: Bestimmung der lokalen Nachweisbarkeit verlangt elektrische Simulation.
- Toggle Fehler (alle Signale sind während des Test mindestens einmal auf »0« und einmal auf »1« zu steuern): bedingte Wahrscheinlichkeiten, dass Tests, die lokal »0« oder »1« einstellen, zufällig einen lokalen Fehler anregen und beobachtbar machen, zu gering.
- Pfadverzögerungsfehler (Unterstellung von slow-to-rise/fall für komplette Pfade, statt Gatteranschlüsse): Im ungünstigen Fall exponentielle Zunahme der Pfad- und damit der Modellfehleranzahl mit der Testobjektgröße.

Testberechnung

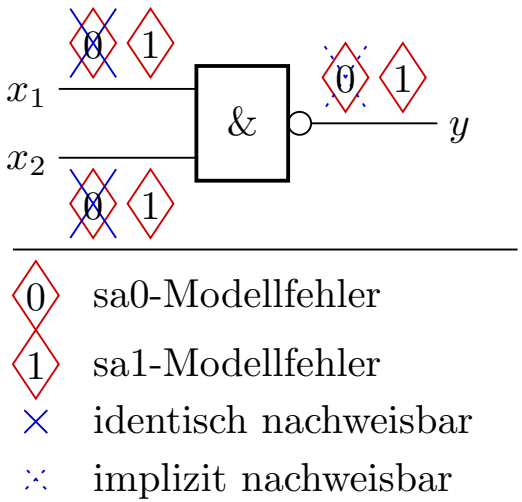
Fehlersimulation

Testsuche durch Fehlersimulation

Wiederhole, bis genügend Modellfehler nachgewiesen sind:

- Gezielte, manuelle oder zufällige Auswahl weiterer Testbeispiele
- Fehlersimulation und Abhaken der nachweisbaren Modellfehler

Zuvor wird mit einem Fehlermodell eine Menge von Modellfehlern zusammgestellt, identisch nachweisbare Modellfehler zu einem zusammengefasst, redundante und optional implizit nachweisbare Fehler gestrichen.



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

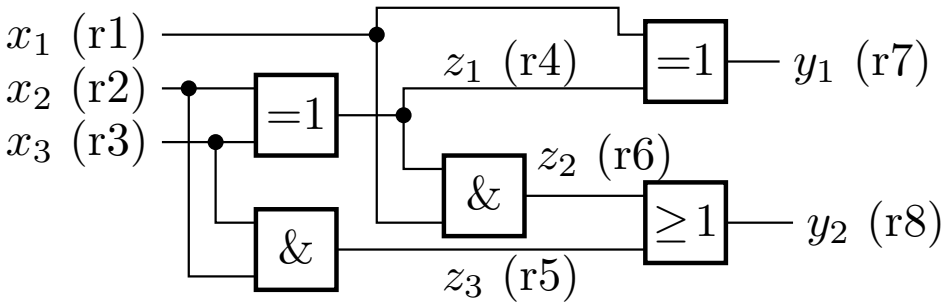
Nachweisidentität (gleiche Nachweismenge)

Nachweisimplikation

zugehörige Eingabe ist Element der Nachweismenge

Haftfehler sind einfach zu simulieren

Schaltung eines Volladdierers



r1 bis r8 Prozessorregister

Programm für die Gutsimulation

lade x_1 in Register r1
lade x_2 in Register r2
lade x_3 in Register r3
r4 = r2 xor r3
speichere Inhalt r4 in z_1
r5 = r2 and r3
speichere Inhalt r5 in z_3
r6 = r1 and r4
speichere Inhalt r6 in z_2
r7 = r1 xor r4
speichere Inhalt r7 in y_1
r8 = r5 or r6
speichere Inhalt r8 in y_2

- Jede zweistellige Logikoperation ist ein Maschinenbefehl.
- In jeder der 8, 16, 32 oder 64 Bits der Operanden kann ein anderer Testfall oder ein anderer Fehler simuliert werden.

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 nachgewiesene Modellfehler nicht weiter mit simuliert werden unter 1 Minute.

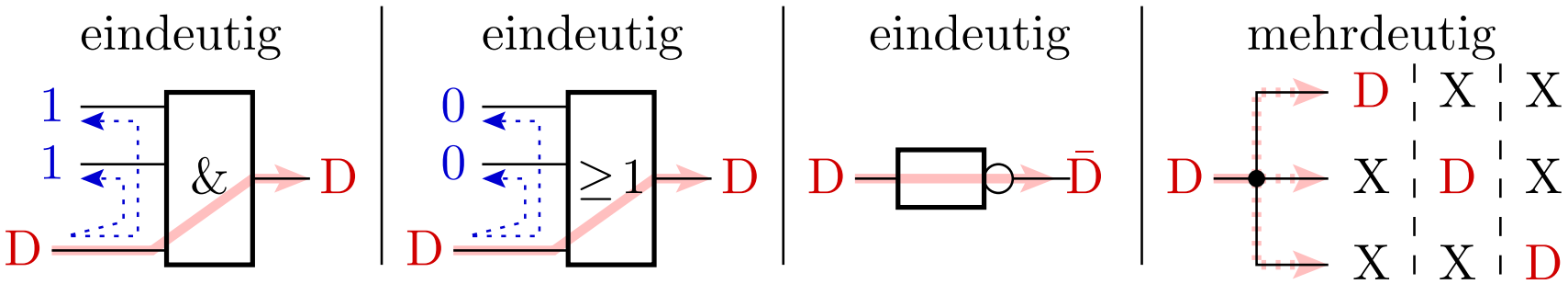
D-Algorithmus

D (Discrepancy)-Kalkül von Roth

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

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:



[3] W. Daehn: Testverfahren in der Mikroelektronik: Methoden und Werkzeuge. Springer 1997.

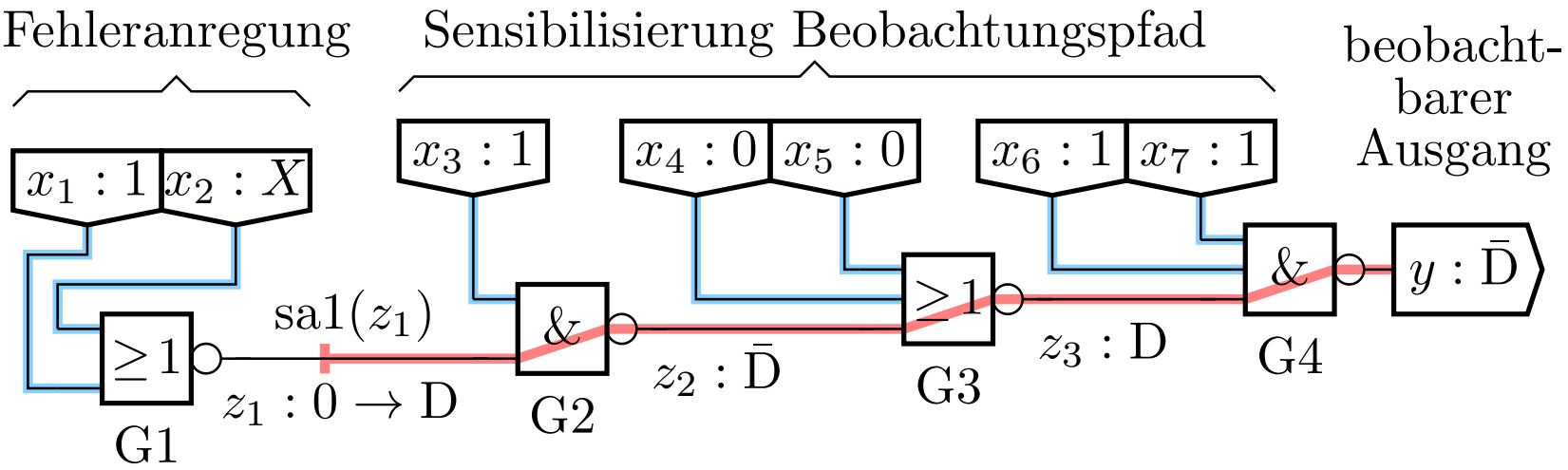
Testsuche für Haftfehler

Ein Haftfehler unterstellt für den Fehlerort, dass der Wert

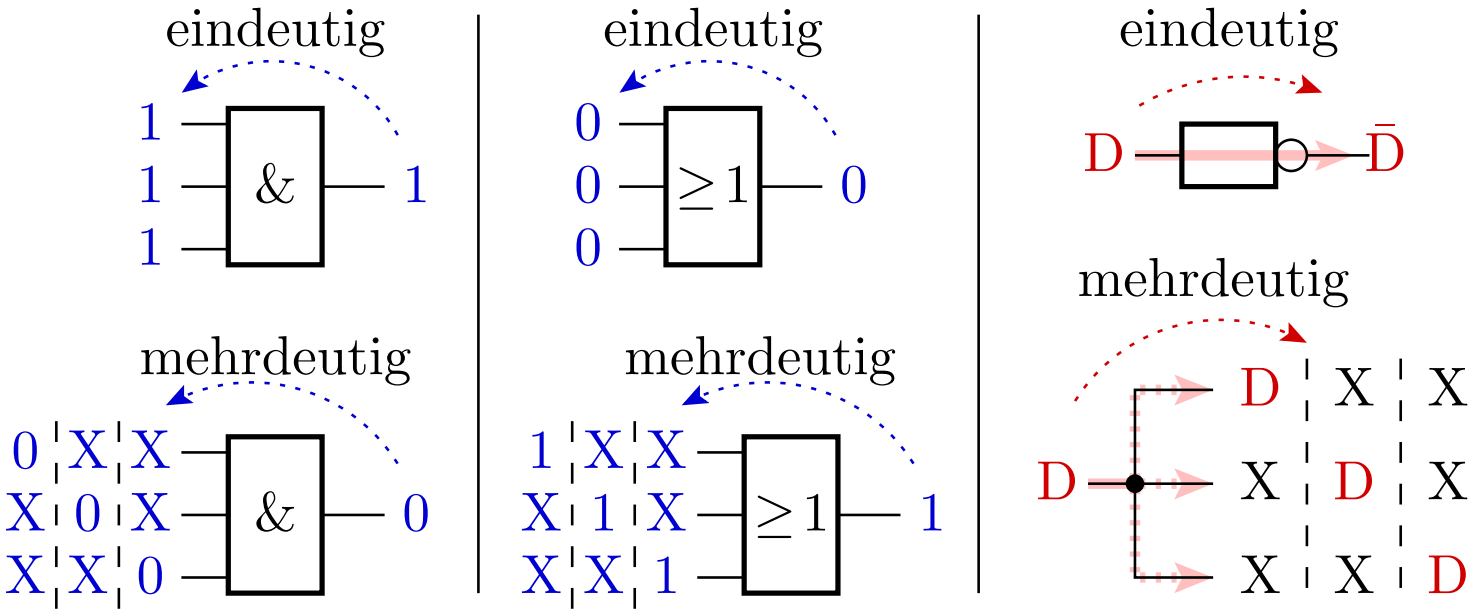
- entweder ständig 0 (sa0) oder
- ständig 1 ist (sa1) ist.

Ausgehend vom Fehlerort werden Eingaben gesucht,

- die den Wert am Fehlerort invertieren und
- bei denen die Invertierung am Fehlerort an einem Ausgang beobachtbar ist.

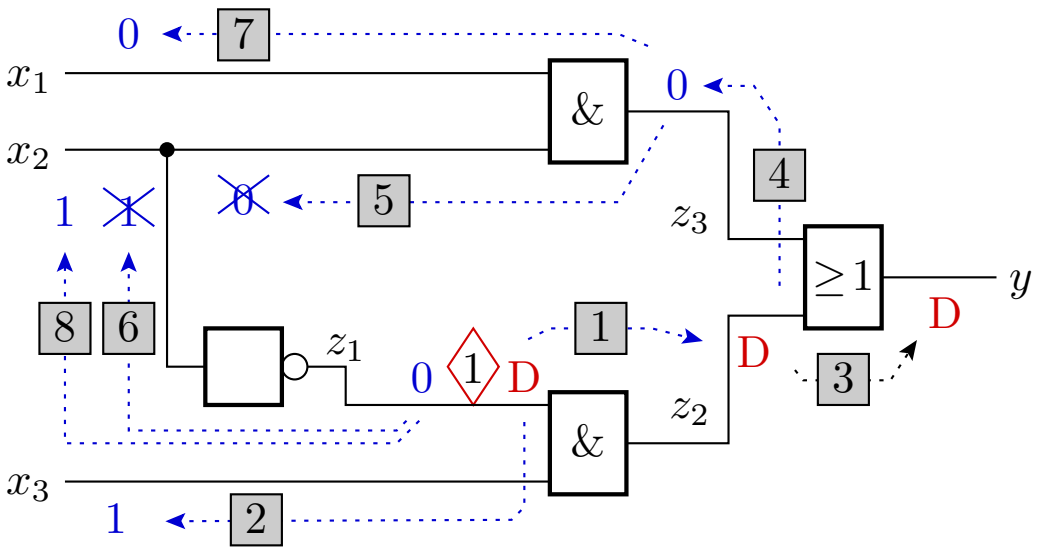


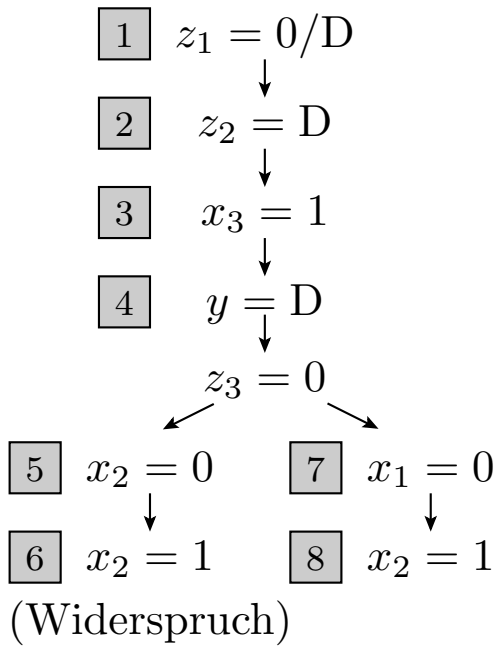
Ein- und mehrdeutige Pfade



Ausgehend vom Fehlerort:

- Festlegen von Werten zur Weiterführung des Beobachtungs- oder eines Steuerpfads.
- Bei Widersprüchen zurück zu letzten Möglichkeit einer Alternativentscheidung, ... $\Rightarrow$ Baumsuche





	x_3	x_2	x_1	z_3	z_2	z_1	y
0	X	X	X	X	X	0D	X
1	1	X	X	X	X	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	~~0~~1	X	0	D	0D	D
7	1	X	0	0	D	0D	D
8	1	1	0	0	D	0D	D

Baumsuche:

- Bei der Wertefestlegung können Widerspüche auftreten.
- Zurück zur letzten mehrdeutigen Entscheidung.
- Keine Lösung nach Durchmusterung des gesamten Baums. $\Rightarrow$ Fehler nicht nachweisbar

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 $> 2^{30\ldots40}$ nicht mehr vollständig durchsuchbar.
- Abbruch der Suche nach einer bestimmten Rechenzeit.

Heuristiken:

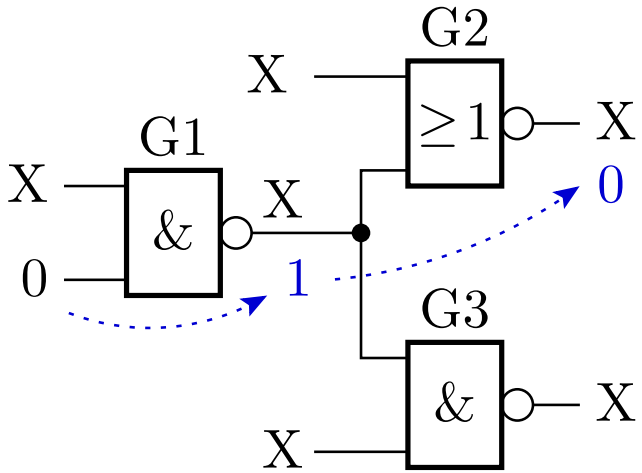
- Frühe Erkennung von Widersprüchen,
- Suchraumbegrenzung und
- gute Suchraumstrukturierung.

Implikationstest

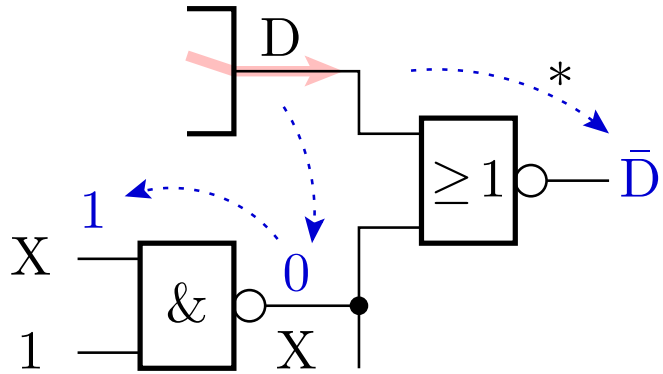
Implikationstest (Widerspruchsfrüherkennung)

- Aus den berechneten Wertefestlegungen alle eindeutig folgenden Werte berechnen.

Implikation in Signalflussrichtung



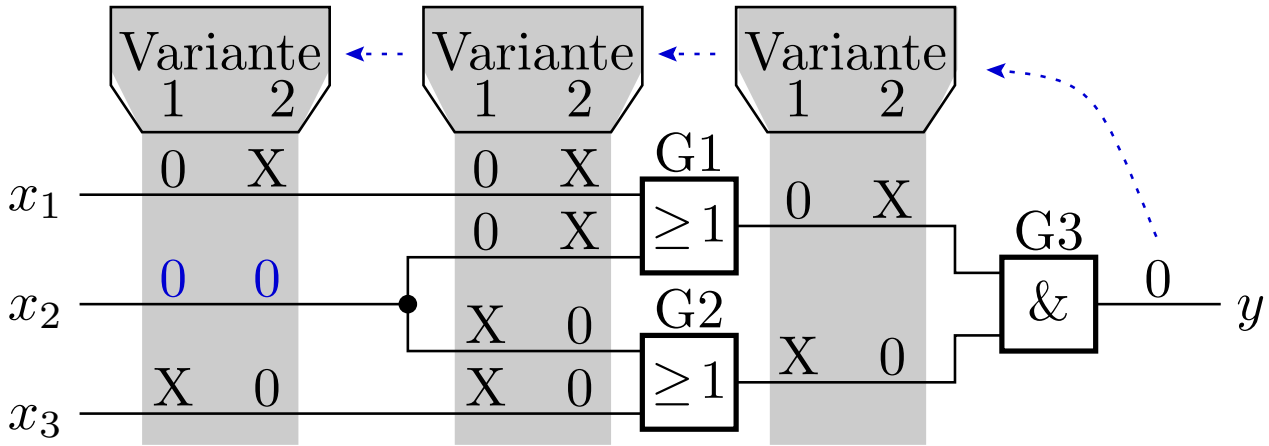
D-Pfad- und Rückwärtsimplikation



* eindeutige Weiterführung

- Mindert die Entscheidungsbaumtiefe.

- Rückwärtsimplikation über mehrere Gatterebenen:

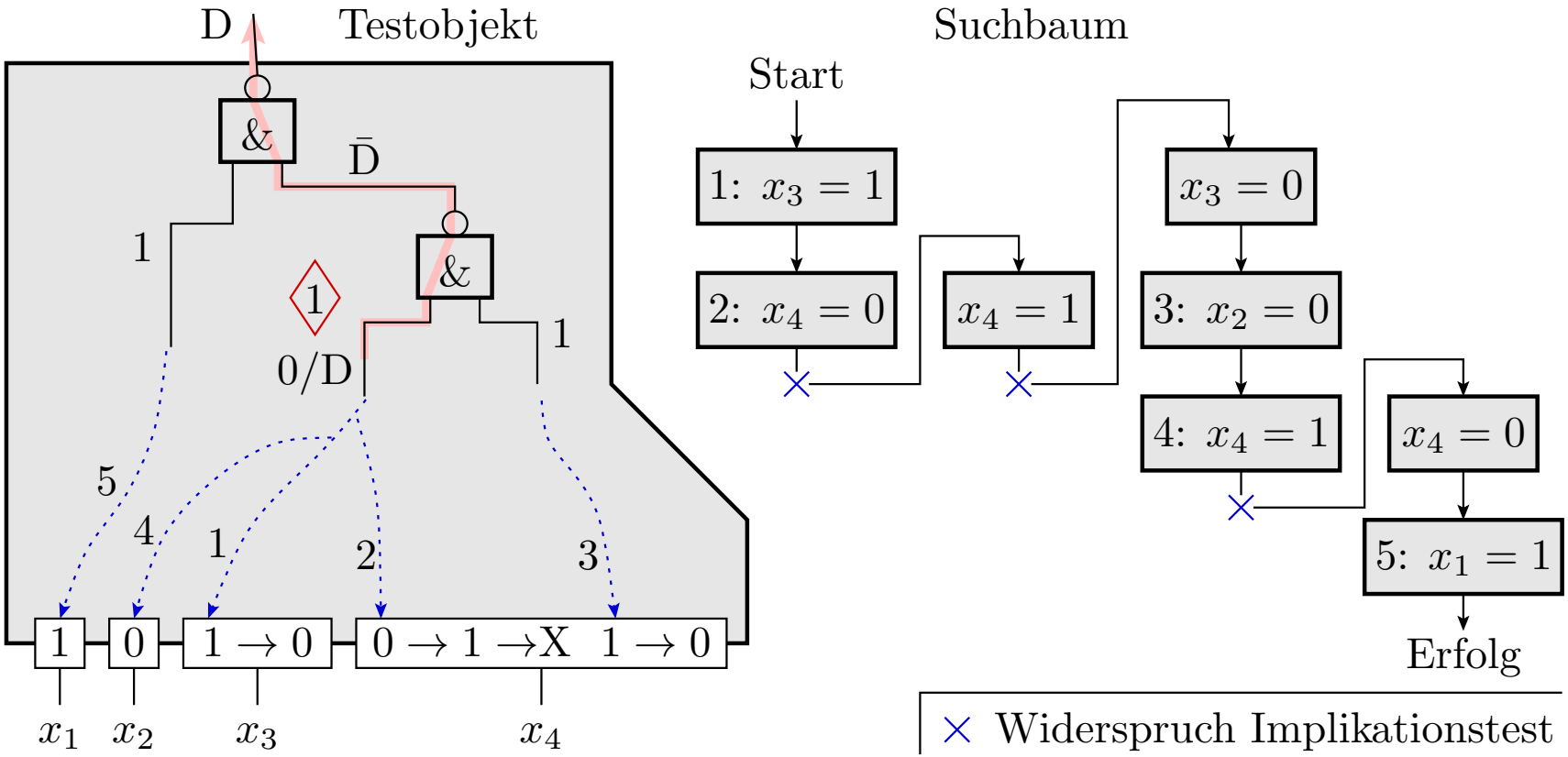


- Für $y = 0$ gibt es zwei Einstellmöglichkeiten.
- Für beide Möglichkeiten muss $x_2 = 0$ sein.
- Das Erkennen von Implikationen dieser Art mindert die Backtracking-Häufigkeit um bis zu 80 %.

Suchraumstrukturierung

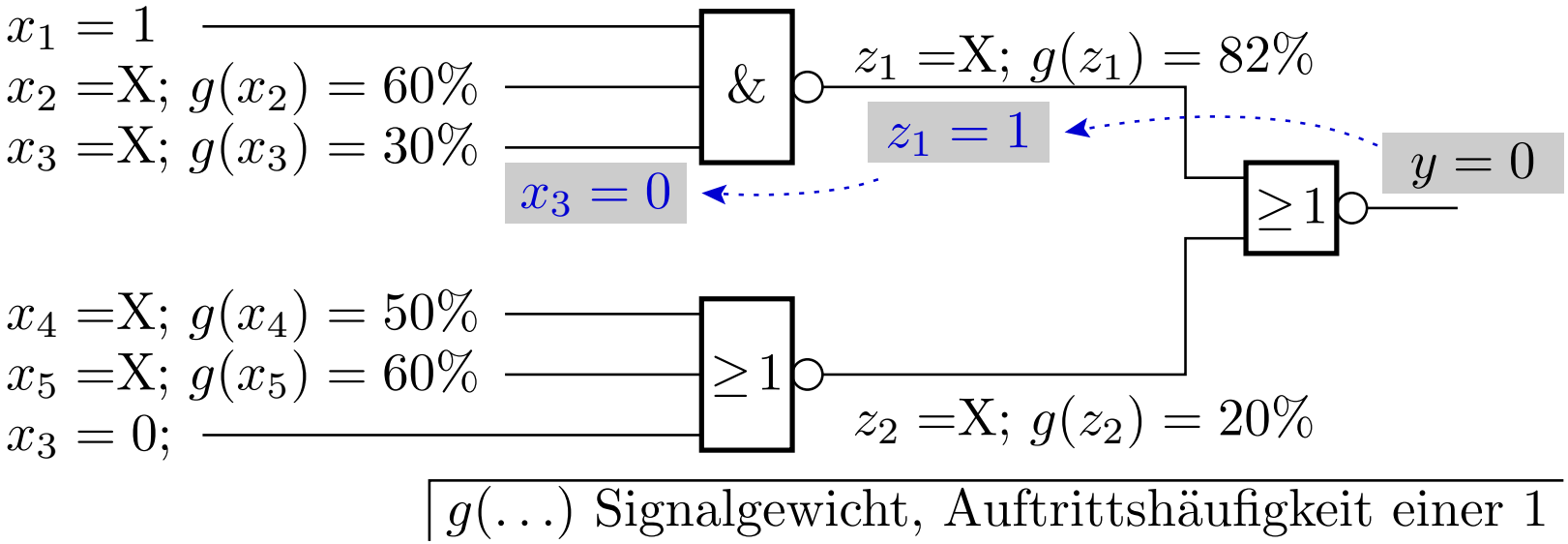
Suchraumbegrenzung

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



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

Geschätzte Erfolgswahrscheinlichkeiten



- Schätzen der Signalwichtungen[4] über eine kurze Simulation mit Zufallswerten oder analytisch.
- Wahl der Steuerwerte / Beobachtungspfade, die mit größerer Wahrscheinlichkeit aktivierbar / sensibilisierbar sind.

[4] Die Wichtung eines Signals ist die Auftrittshäufigkeit einer »1«.

Komplexe Funktionsbausteine

Komplexe Funktionsbausteine

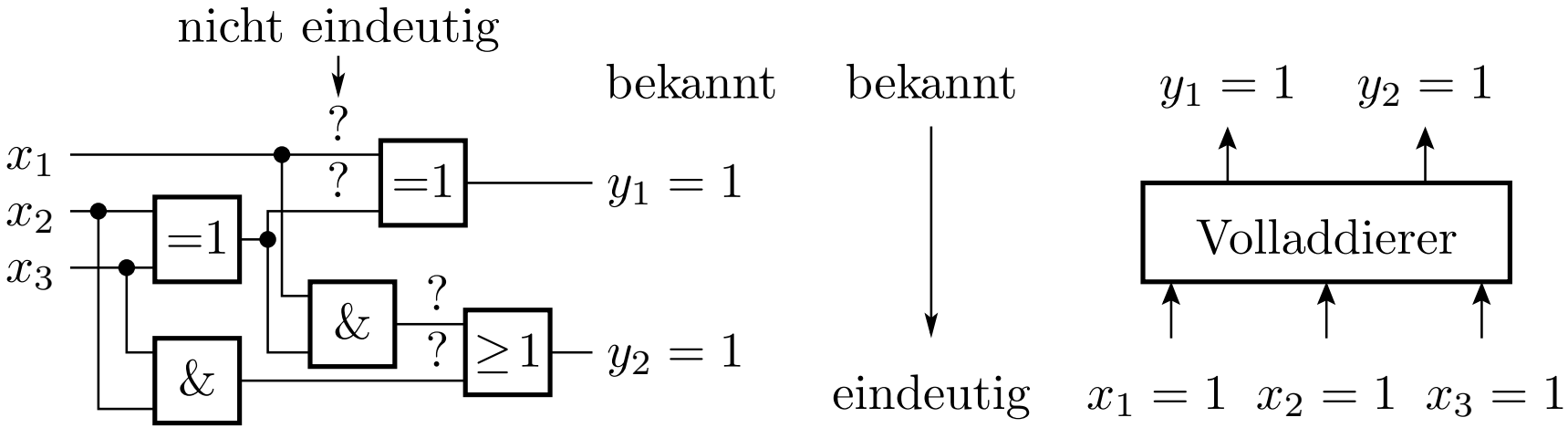
- Beschreibung durch Tabellenfunktion (Bsp. Volladdierer):

x_2	x_1	x_0	s	c
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

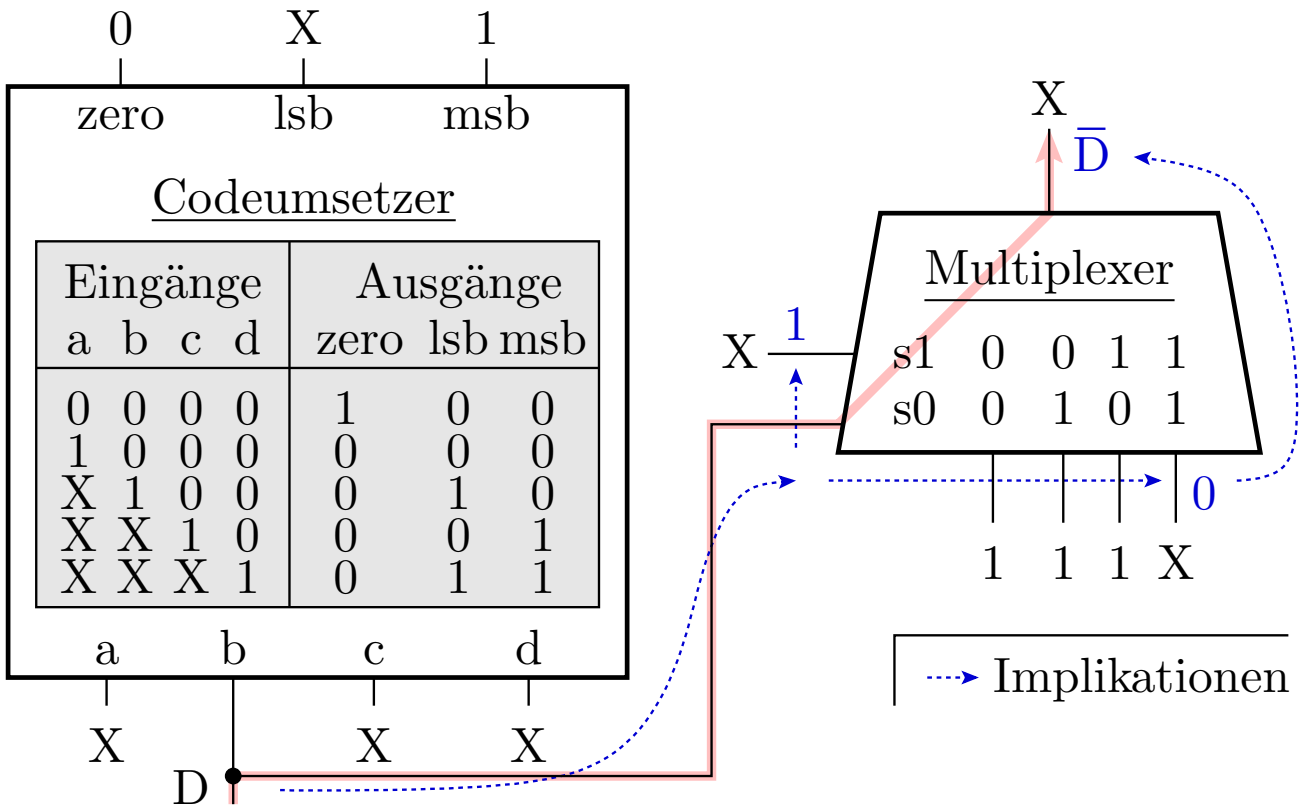
gegeben		Lösungsmenge
XXX00	$\Rightarrow$	00000
01DXX	$\Rightarrow$	01D$\bar{\text{D}}$D
1XXXD	$\Rightarrow$	10D$\bar{\text{D}}$D, 1D0$\bar{\text{D}}$D
11XX1	$\Rightarrow$	11111, 111001

- 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 »D=0« und für »D=1« passen.

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.

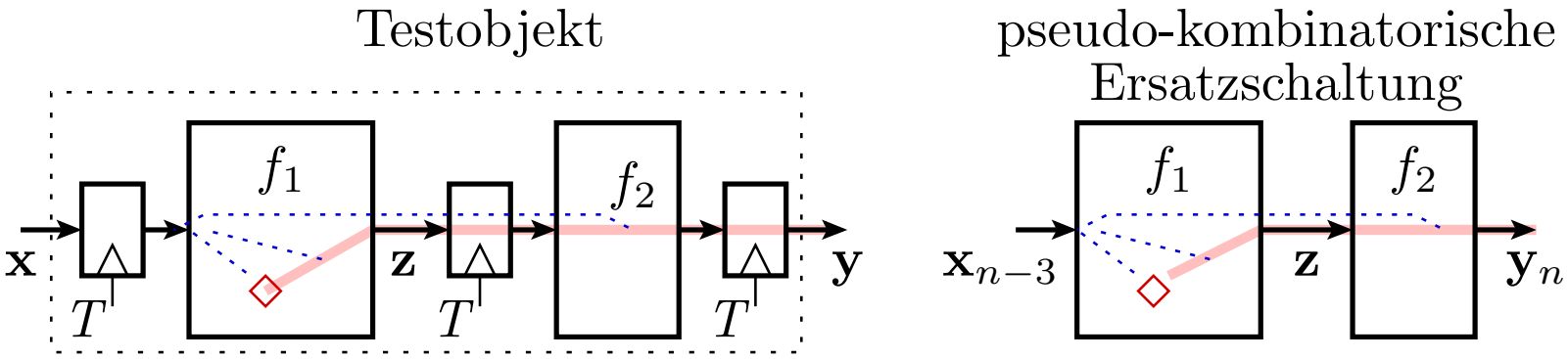


- »lsb« hängt bei »zero=0« und »msb=1« nicht von »b« ab. Eindeutiger D-Pfad über Multiplexer.
- Tabelleneingabewerte »X« (Eingang beeinflusst nicht die Ausgabe) führt zu Tabellen mit $\ll 2^{N_\mathrm{E}}$ Tabellenzeilen (N_E – Anzahl der Eingänge).

Sequentielle Schaltungen

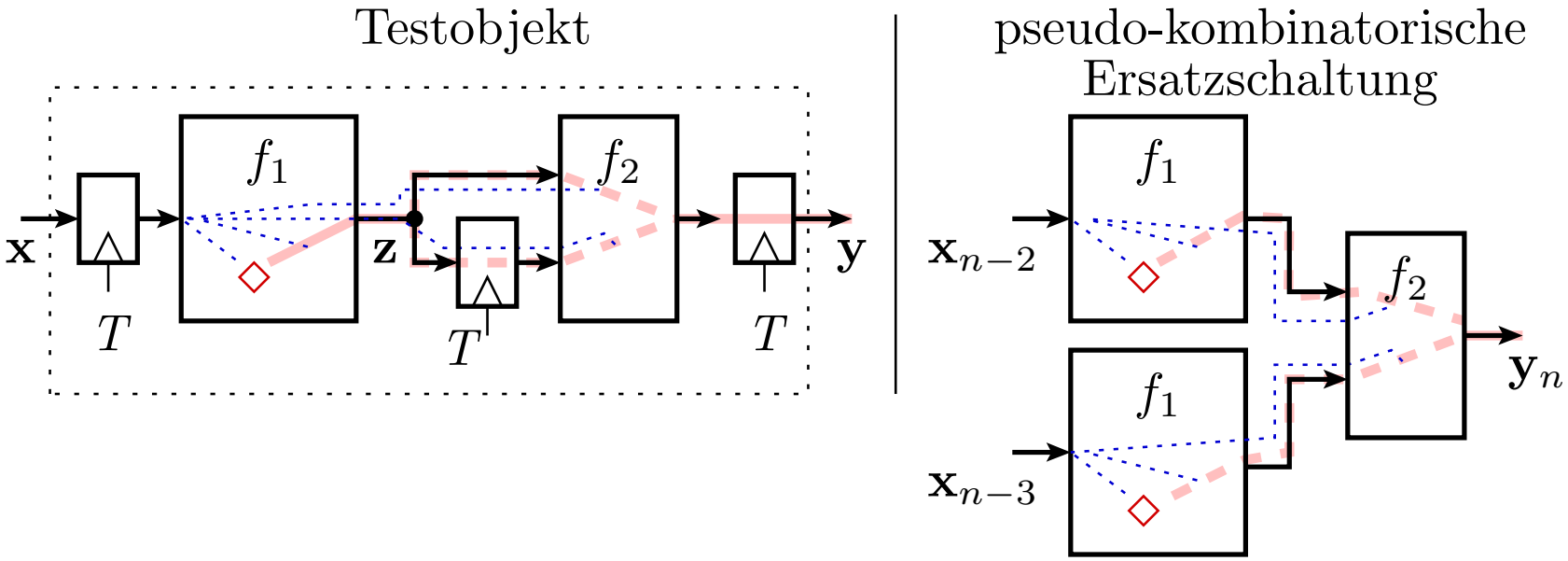
Pseudo-kombinatorische Ersatzschaltung

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



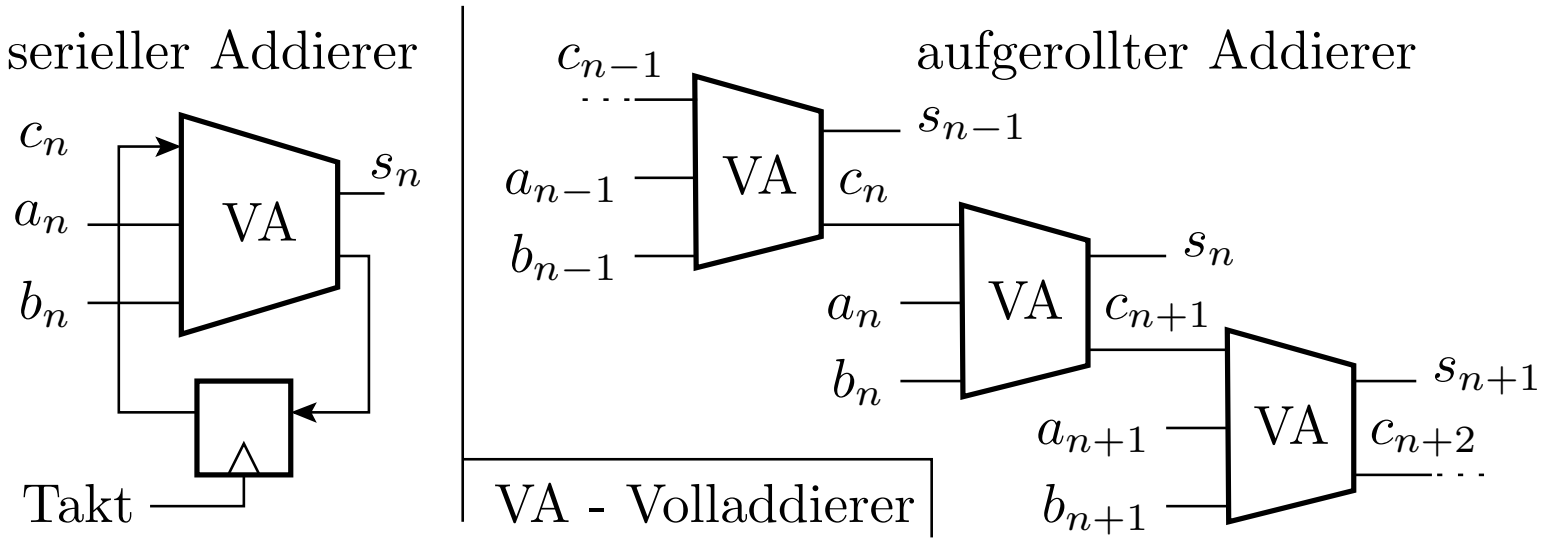
- Testberechnung wie für eine kombinatorische Schaltung.
- Die Verzögerung der Ausgabe gegenüber der Eingabe wird erst bei der Testdurchführung berücksichtigt.

Verarbeitung in mehreren Zeitebenen



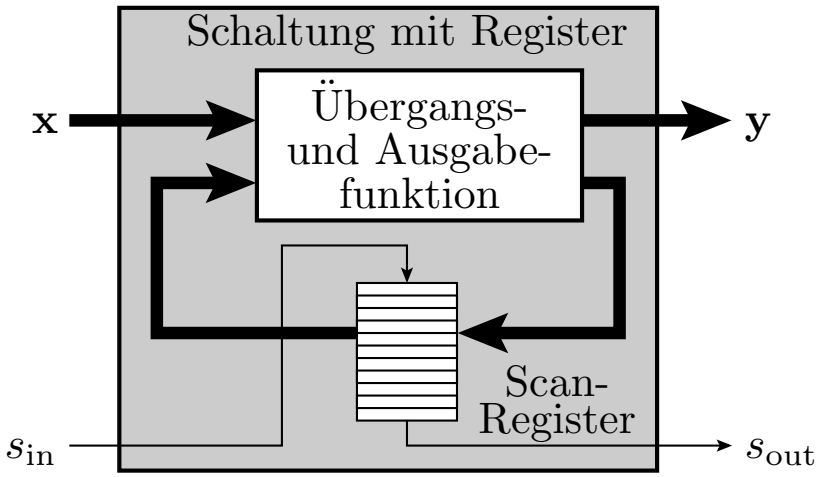
- Mehrere Kopien gleicher Schaltungsteile in der pseudo-kombinatorischen Ersatzschaltung.
- Der eingebaute Haftfehler ist in jeder Kopie der Teilschaltung.
- Berechnet wird eine Folge von Testeingaben für mehrere Zeitschritte (Mehr-Pattern-Test).

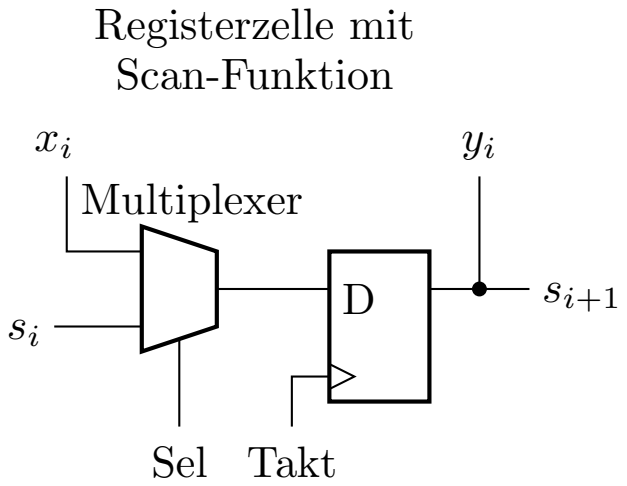
Schaltungen mit Rückführung



- Pseudo-kombinatorischen Ersatzschaltung mit endlos vielen Kopien der Übergangsfunktion.
- Längenbegrenzung der Steuer- und Beobachtungspfade.
- Alternative: Lese- und Schreibzugriff auf Zwischenergebnisse und -zustände, z.B. durch Verschalten der internen Speicherzeichen zu einem Scan-Register (vergl. Folie 91).

Scan-Verfahren





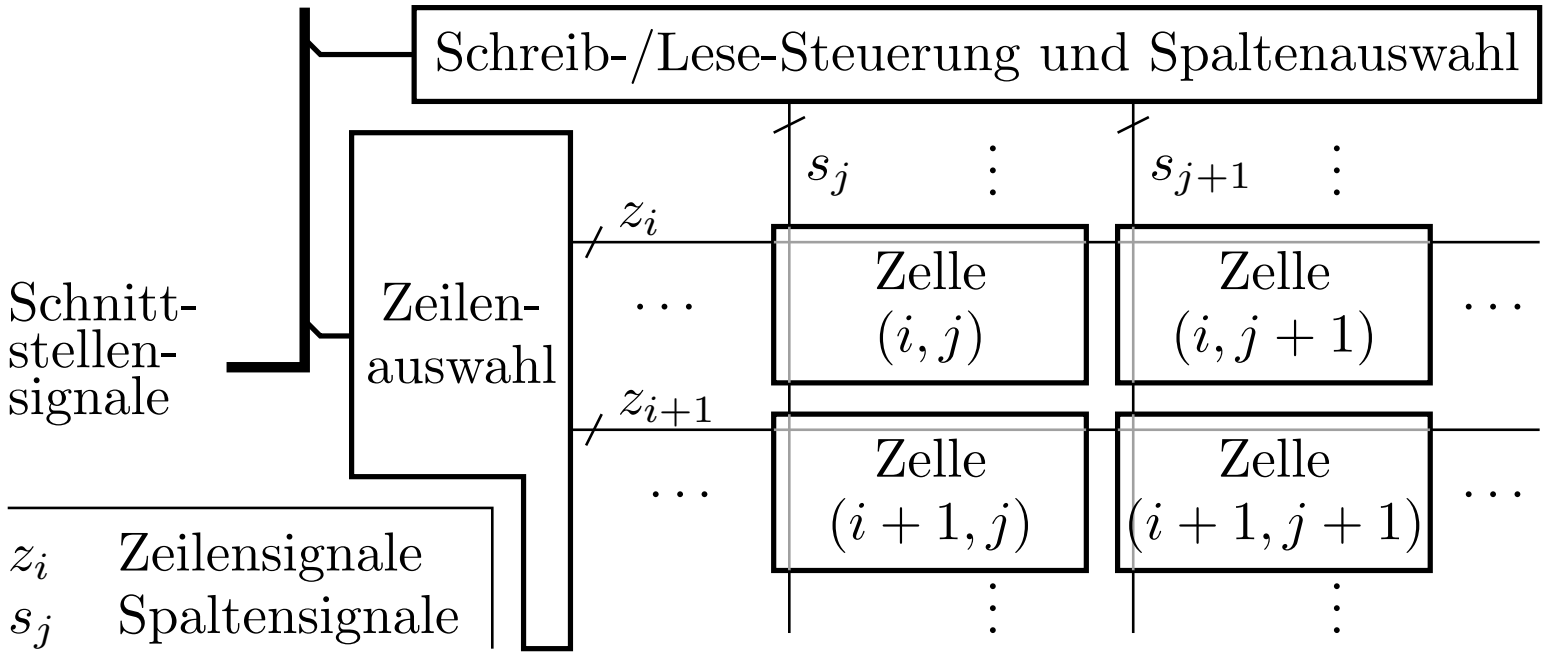
Lese- und Schreibzugriff während des Tests durch Umschalten des Zustandsspeicher in ein Schieberegister.

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

Speichertest

Blockspeicher

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



Typische Fehlerannahmen für einen SRAM siehe nächste Folie.

beteiligte Zellen	Name	Definition	Fälle	Testfolge für den Nachweis
1	Haftfehler	Wert der Speicherzelle ist nicht setzbar	stuck-at-0 stuck-at-1	$W(i)1$, $R(i)1$ $W(i)0$, $R(i)1$
	Übergangsfehler	Wert der Speicherzelle i ist nur in einer Richtung änderbar	kein Übergang $1 \to 0$ $0 \to 1$	 $W(i)1$, $R(i)1$, $W(i)0$, $R(i)0$ $W(i)0$, $R(i)0$, $W(i)1$, $R(i)1$
	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

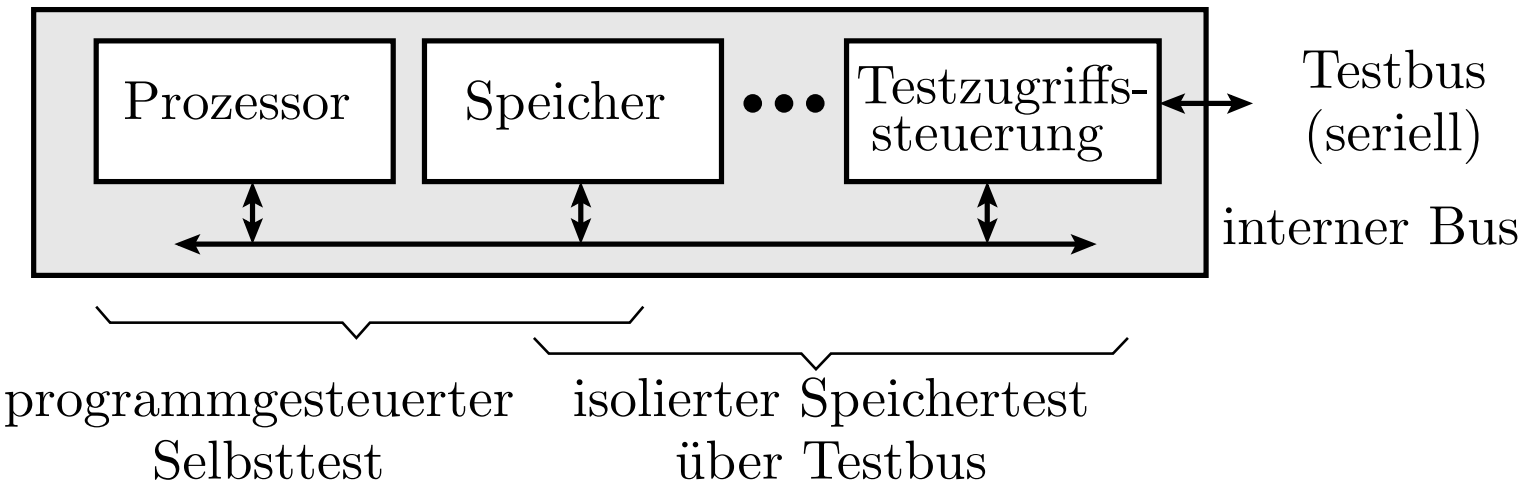
Beispiel Marching Test

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.

Test eingebetteter Blockspeicher



Eingebettete Blockspeicher werden vorzugsweise isoliert von ihrer Schaltungsumgebung getestet:

- über herausgezogenen Bussignale,
- über den Testbus oder
- programmgesteuert vom Prozessor als eingebauter Selbsttest (BIST – Built-In Self-Test).

Selbsttest

Selbsttest

Einbau der Testfunktionseinheiten mit in den Schaltkreis:

- Testmustergenerator: Pseudo-Zufallsgenerator, Zähler, Schieberegister.
- Testablaufsteuerung, in der Regel überTestbus (vergl. Abschn. 3.20).
- Ausgabekontrolle, vorzugsweise durch Bildung eines Prüfkennzeichens mit LFSR (Linear Feedback Shift Register).

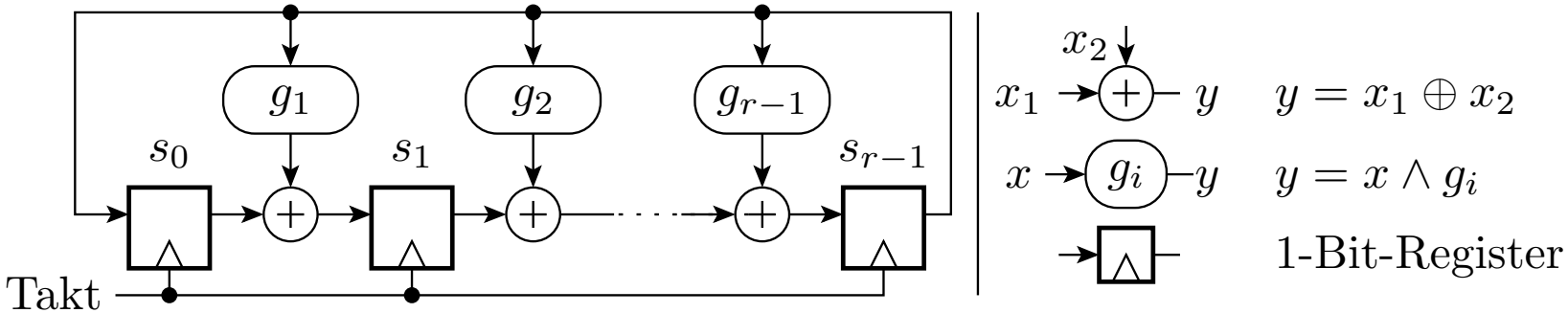
Vorteile:

- Erlaubt sehr große Testsatzlängen, Test mit voller Geschwindigkeit,
- nutzbar auch später im Zielsystem für den Einschalttest.

Pseudo-Zufallsregister

Linear rückgekoppelte Schieberegister

Ein linear rückgekoppelte Schieberegister (LFSR **L**inear **F**eeback **S**hift **R**egister) in einer ersten Ausführung verschiebt 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**:



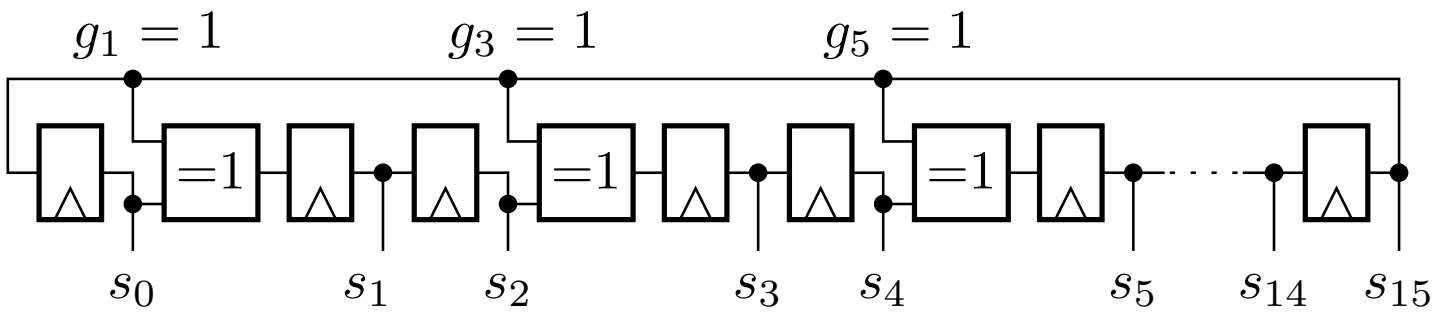
Für jede Bitanzahl r des Zustandsvektors gibt es Konstanten **g**, sog. »primitive Polynome«, bei denen alle Zustände außer 000...0 ineinander übergehen. Nur solche Konstanten **g** werden verwendet.

Primitiven Polynome und die Konstante g

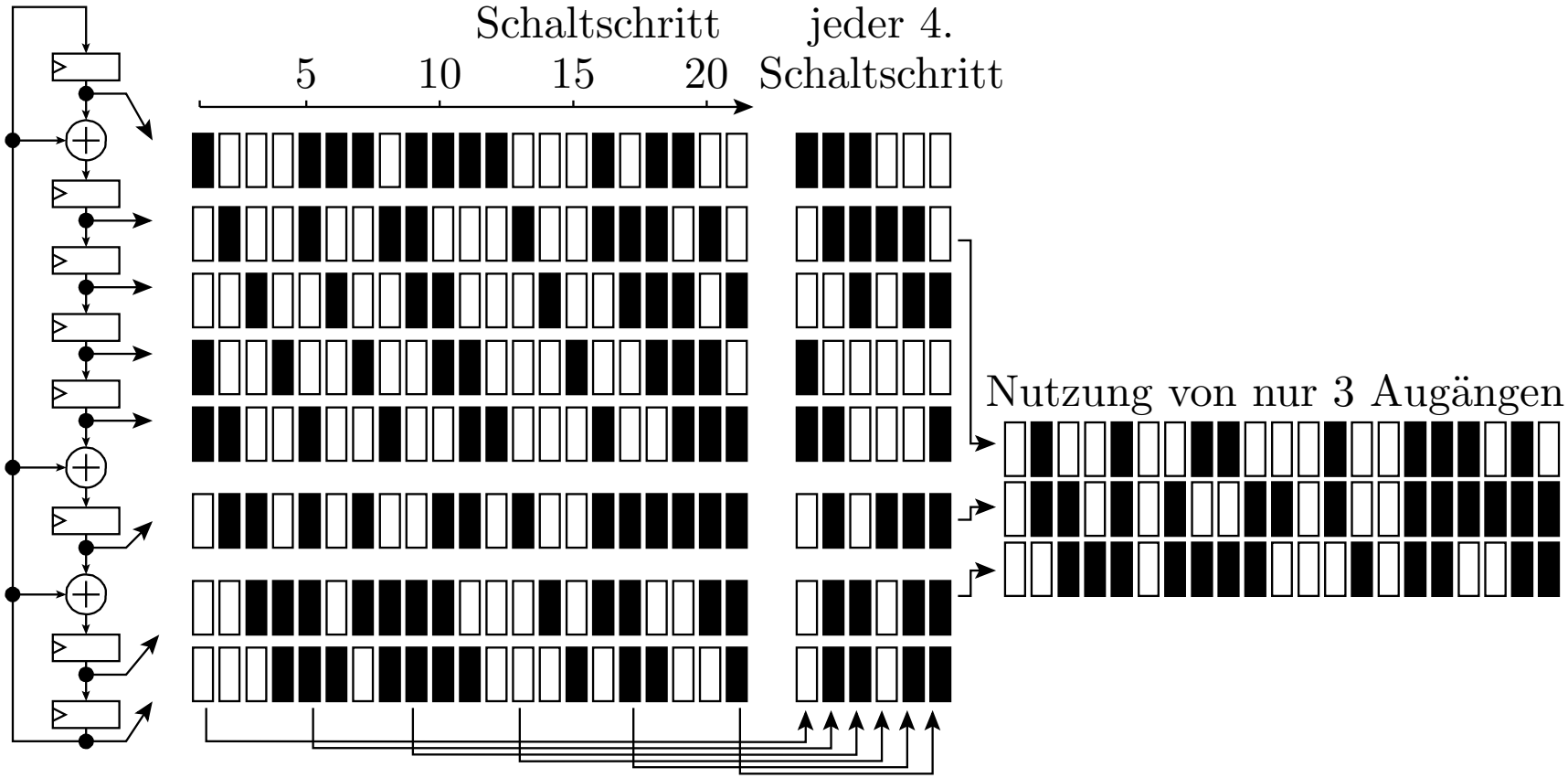
Mit dem Internet-Suchbegriff »Primitive Polynome« findet man z.B. für 16-Bit LFSR:

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

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.

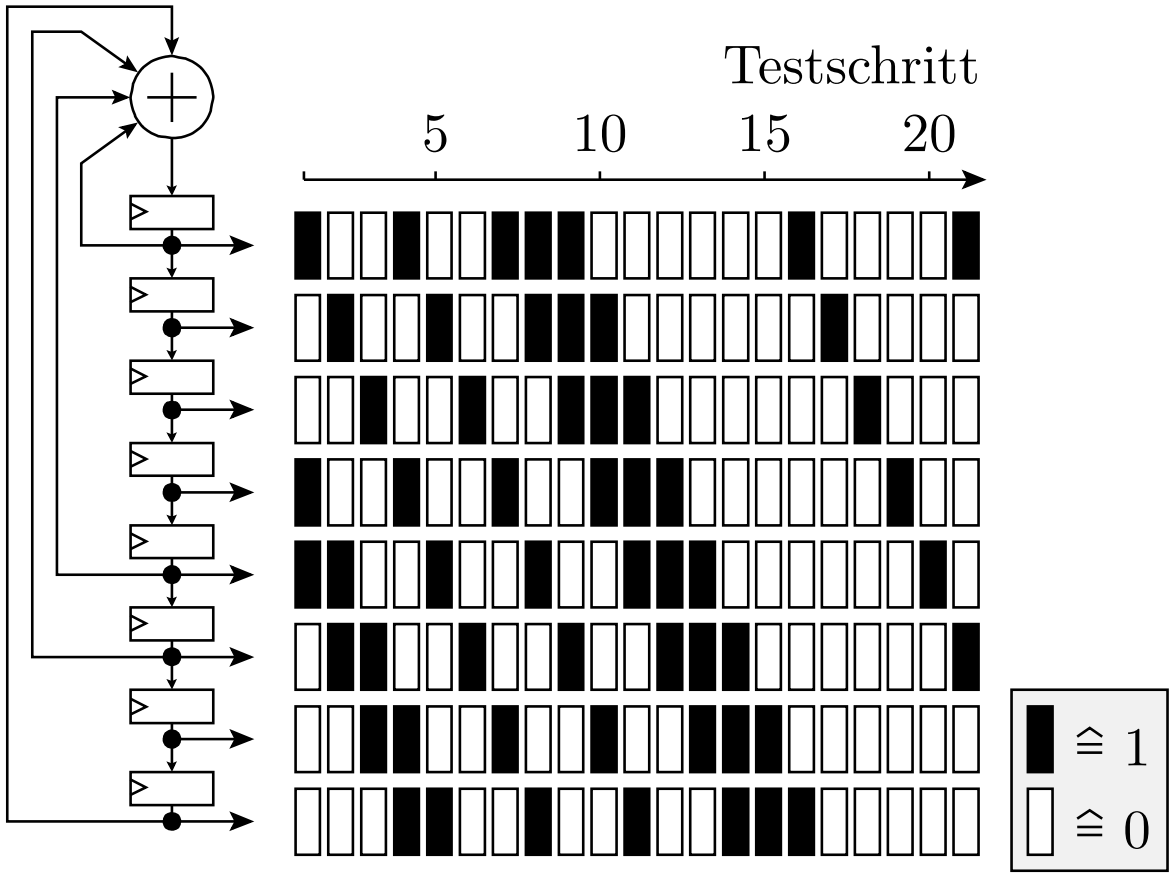


Pseudo-Zufallsfolge eines 8-Bit-LFSR



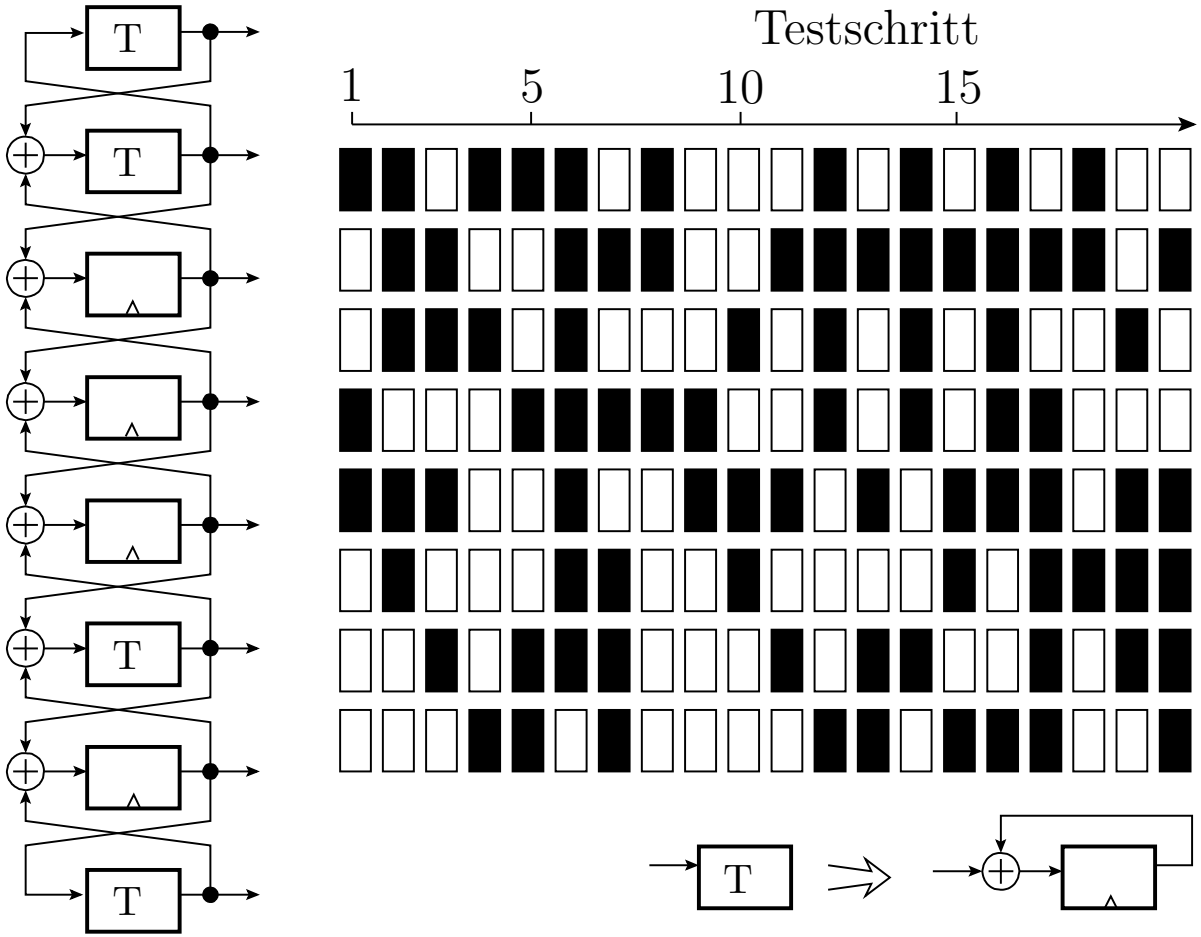
Falls die »Streifenmuster« durch die Schiebeoperationen stören, nur einen Teil der Ausgaben nutzen.

Bei »Umkehrung« der Signalflussrichtung wird aus den verteilten EXOR-Gattern ein zentrales EXOR-Netzwerk am Eingang.



Gleiche Zyklusstruktur bei gleichen Rückführstellen. Bitfolgen mit Phasenverschiebung größer 1 auch durch EXOR mehrerer Bitströme.

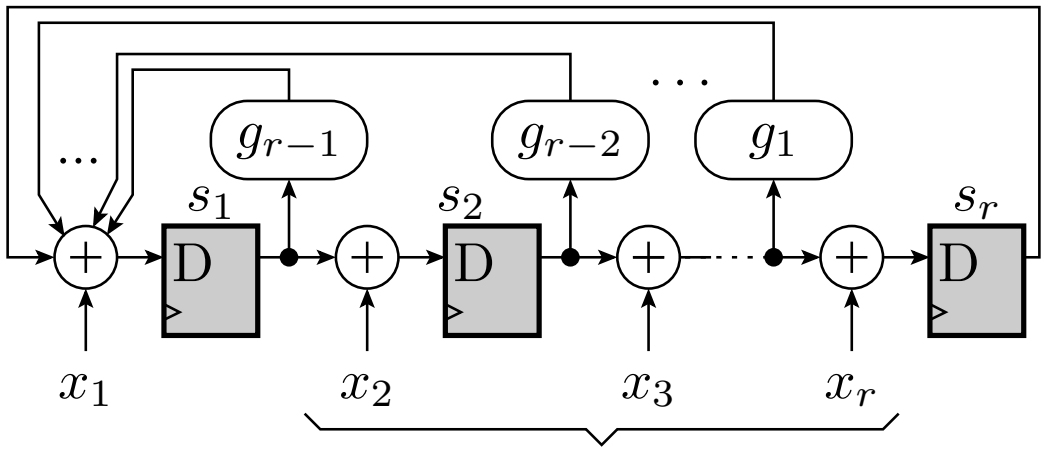
Es gibt viele weitere lineare Automaten, die auch zyklisch Bitfolgen in zufälliger Reihenfolge erzeugen. Beispiel Zellenautomaten, bei denen jedes Folgebit aus dem eigenen und den Zuständen der Nachbarbits gebildet wird:



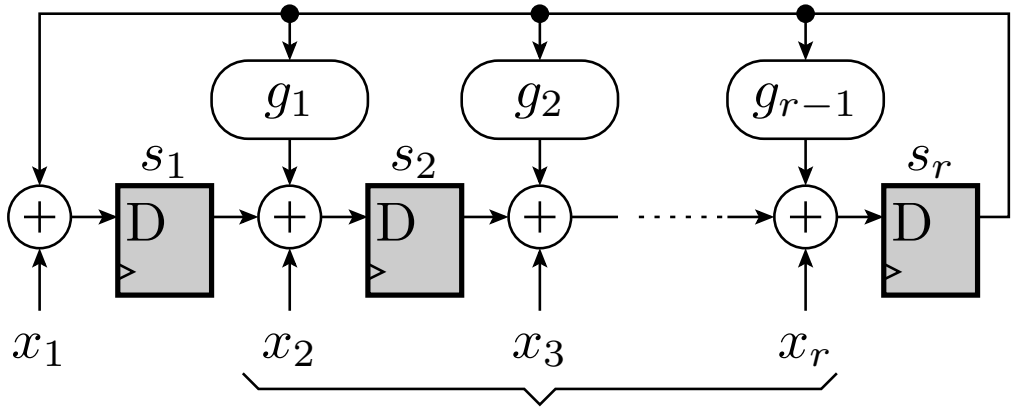
Signaturregister

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).



Die Rückführung darf dabei auch wie bei der Polynom-Division dezentral sein.



Erweiterungsmöglichkeit auf mehrere Eingänge

Die Koeffizienten g_i der Rückführung, bei der Polynom-Division das Divisor-Polynom, bestimmen die autonome Zyklusstruktur[5]. Die autonome Zyklusstruktur ist bei zentraler und dezentraler Rückführung mit denselben Rückführkoeffizienten gleich. Bevorzugt werden lange Zyklen, insbesondere sog. primitive Polynome, bei denen alle Zustände außer »alles null« einen $2^r - 1$ langen Maximalzyklus bilden.

[5] Zyklusstruktur ohne Eingaben.

Experiment Fehlererkennungssicherheit von LFSR

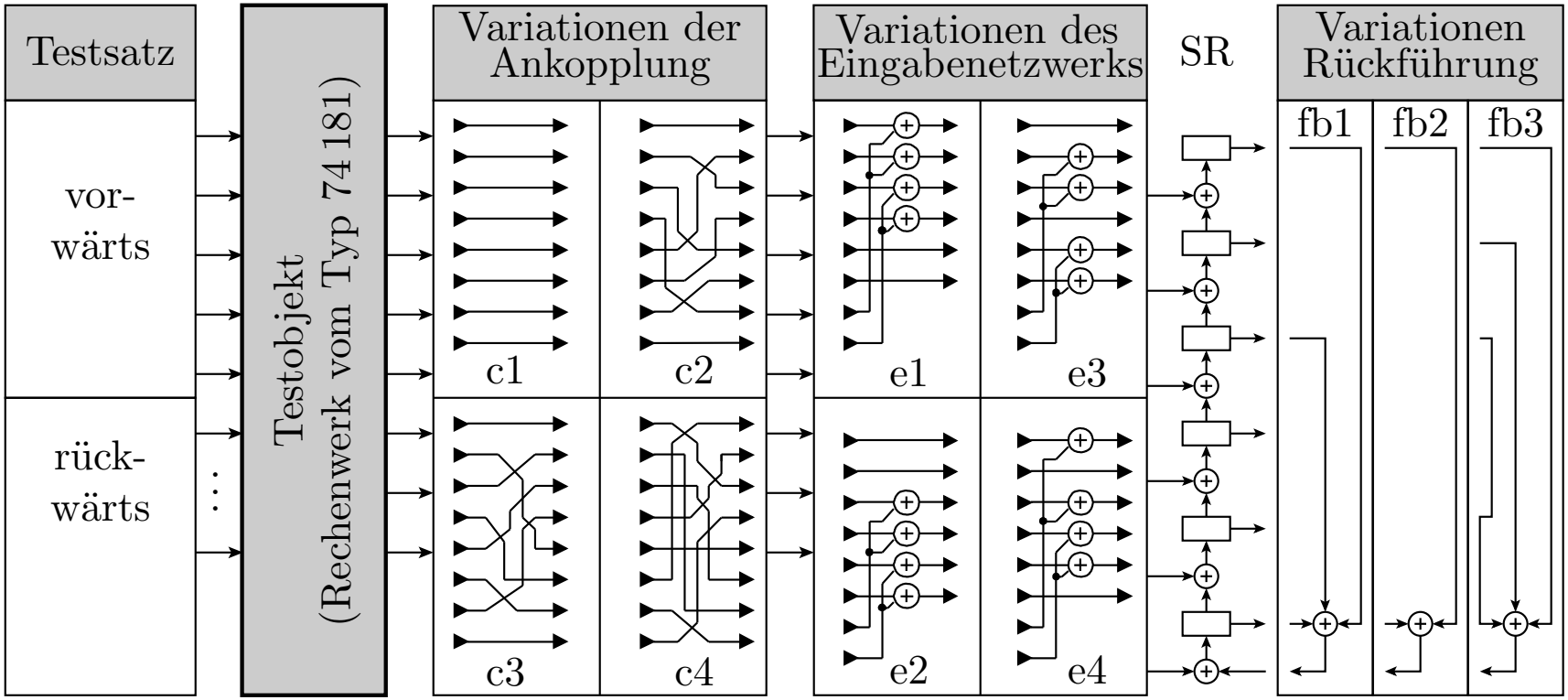
Es ist schwer zu glauben, dass

- mit r-Bit Prüfkennzeichen beliebige Verfälschung mit einer Wahrscheinlichkeit $p_{\mathrm{E}} = 1 - 2^{-r}$ erkannt werden und
- die Schaltungsstruktur, die Rückführung etc. kaum Einfluss auf die Erkennungswahrscheinlichkeit haben sollen.

Deshalb ein Experiment:

- Simulation einer Schaltung (4-Bit-Rechenwerk) mit einem Testsatz und 250 verschiedenen Haftfehlern. Berechnung des Prüfkennzeichens für jeden Fehler.
- Variation der Testsatzreihenfolge,
- Variation der Ankopplung an das LFSR und
- Variation der Rückführung.

Zählen der nachweisbaren Fehler für jede Konfiguration.



Aus $r = 6$ bit folgt, dass jeder Fehler mit einer Wahrscheinlichkeit $p_{\mathrm{E}} = 1 - 2^{-6} = 98{,}44\%$ erkenn- und mit einer Wahrscheinlichkeit $p_{\mathrm{F}} = 2^{-6} = 1{,}36\%$ nicht erkennbar sein müsste. Definition einer Zufallsgröße X_i zum Zählen der nicht erkennbaren Fehler:

$$\begin{aligned} \mathbb{P}\left[X_i = 0\right] &= 1 - 2^{-6} \text{ Fehler } i \text{ nachweisbar} \\ \mathbb{P}\left[X_i = 1\right] &= 2^{-6} \text{ Fehler } i \text{ nicht nachweisbar} \end{aligned}$$

Wenn die Theorie stimmt, müsste die Anzahl der maskierten Fehler

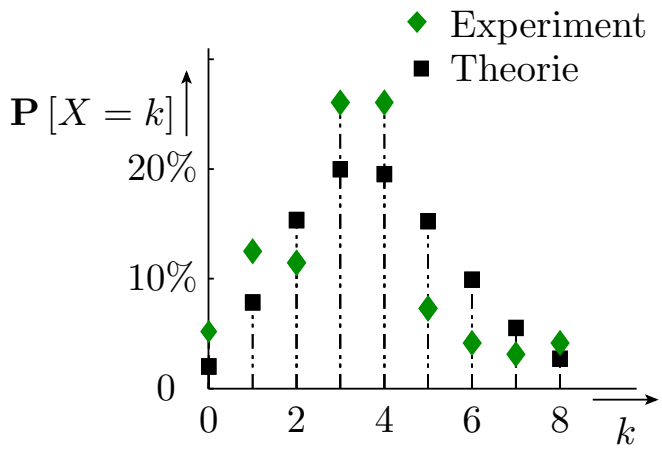
$$X = \sum_{i=1}^{250} X_i$$

binomialverteilt mit $p_{\mathrm{F}} = 2^{-6}$ sein (siehe Foliensatz 3):

$$\mathbb{P}\left[X = k\right] = \left(\begin{array}{c} N \\ k \end{array} \right) \cdot p_{\mathrm{F}}^{k} \cdot \left(1 - p_{\mathrm{F}}\right)^{N-k}$$

Anzahl der maskierten Fehler

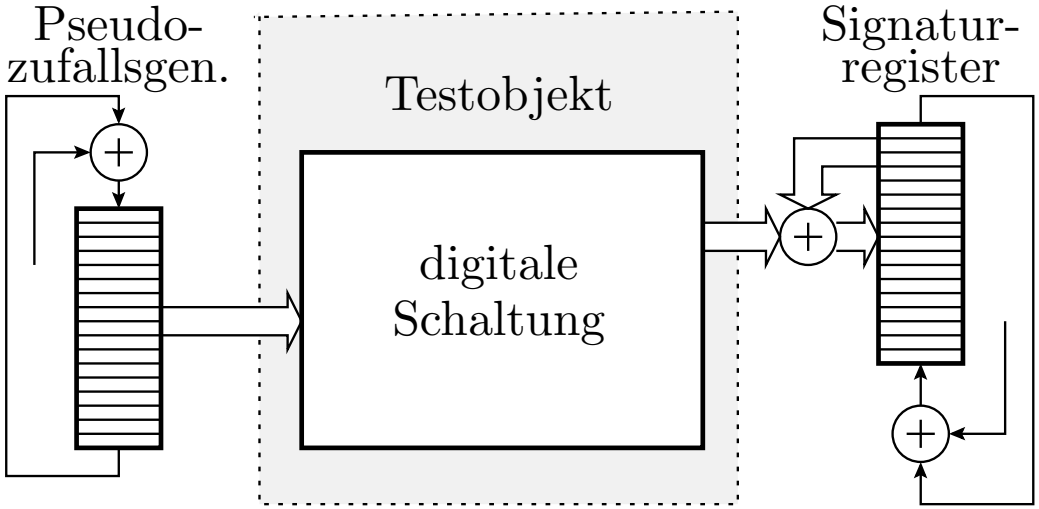
		e1				e2				e3				e4			
		c1	c2	c3	c4	c1	c2	c3	c4	c1	c2	c3	c4	c1	c2	c3	c4
vor-wärts	fb1	3	4	1	2	3	4	3	3	4	2	4	3	4	3	4	6
	fb2	3	4	1	7	2	2	1	4	2	1	1	3	2	5	3	7
	fb3	5	2	2	8	4	5	3	4	3	6	3	7	5	3	3	4
rück-wärts	fb1	6	4	4	2	3	4	3	4	3	4	3	4	4	8	4	5
	fb2	2	0	0	1	4	1	4	1	0	0	0	1	1	1	4	1
	fb3	2	4	3	4	4	8	5	8	3	3	3	6	3	3	4	3



Abweichungen zwischen Vorhersage und Experiment nicht signifikant.
Hypothese $p_{\mathrm{F}} = 2^{-6}$ bestätigt.

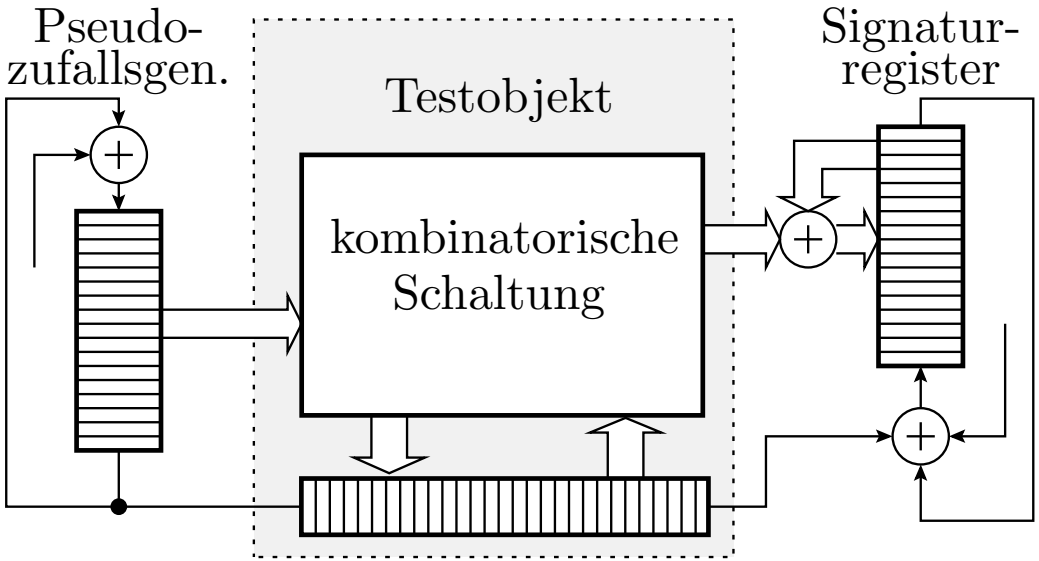
Selbsttest mit LFSR

Selbsttest (BIST **B**uilt-in **S**elf **T**est)



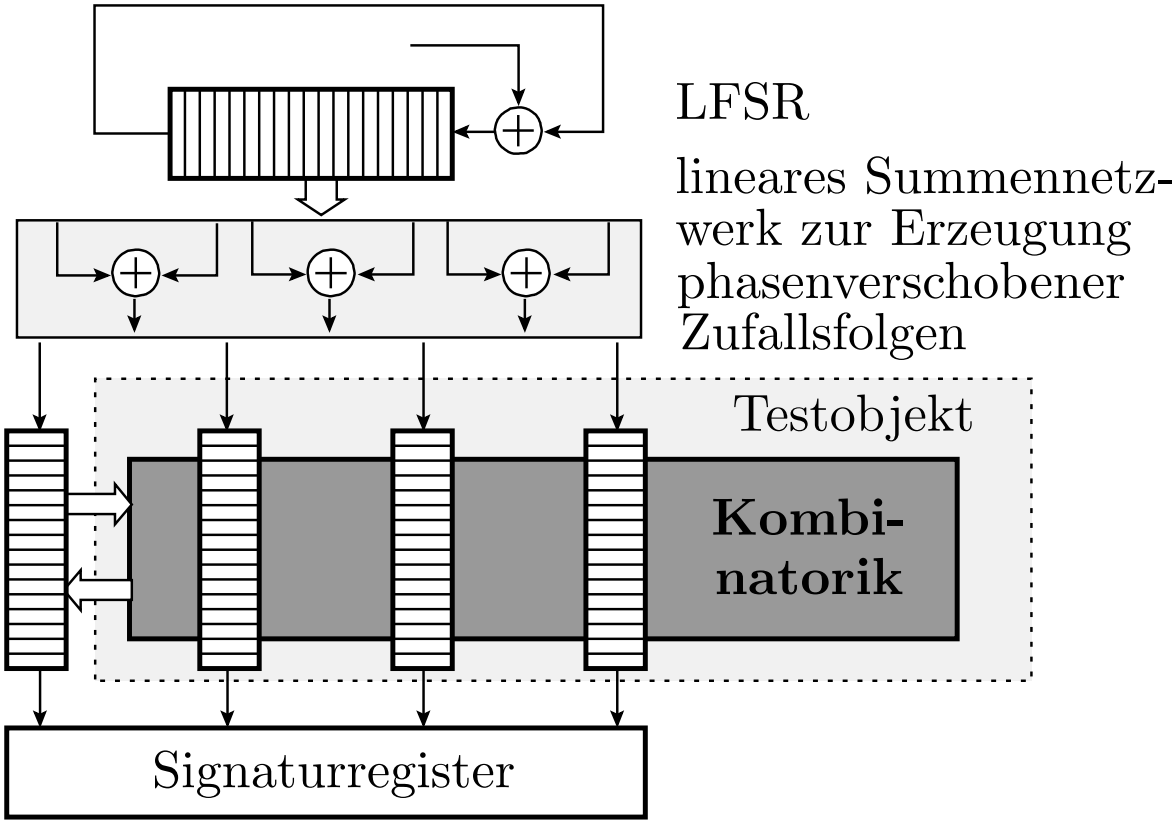
- Einrahmen der Schaltung mit Schieberegistern und Ergänzung einiger EXOR-Gatter.
- Wenn als Schieberegister vorhandene Ein- und Ausgaberegister verwendet werden, besonders niedriger Zusatzaufwand.
- Test mit voller Schaltungsgeschwindigkeit von Millonen bis Milliarden Test pro Sekunde.

BIST plus Scan



- Zwischen den Testschritten Zustandsregister 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 mit mehreren Scan-Registern, die zwischen den Testschritten parallel gelesen und mit neuen Zufallswerten beschrieben werden.



Weiterführende Literatur [G. Kemnitz: Test und Verlässlichkeit von Rechnern. Springer 2007.]

Fehlerorientierte Wichtung

Fehlerorientierte Wichtung

Fehlerorientiert gesuchte Tests verlangen einen entsprechend großen Speicher, für Selbsttest ungeeignet. Alternative fehlerorientierte Wichtung.

Die Wichtung $g(x)$ eines binären Signal x ist die Auftrittswahrscheinlichkeit einer eins:

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

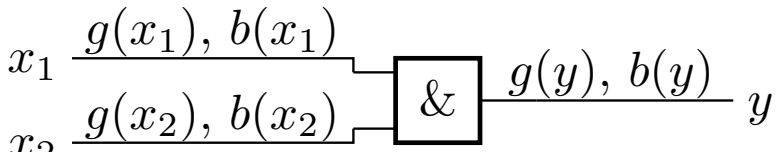
In einem System, in dem logische Werte UND, ODER, ... verknüpft werden, sind die Wahrscheinlichkeiten

- der Steuer- und Beobachtbarkeit und
- des Fehlernachweises

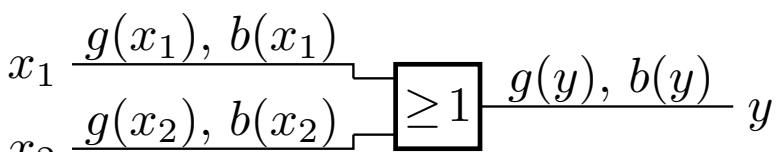
Funktionen der Wichtungen an den Eingängen und damit über Eingabewichtungen einstellbar.

Berechnung der Wichtungen und Beobachtbarkeiten

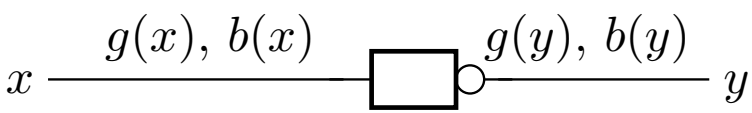
Die Wichtung einer UND-Verknüpfung ist das Produkt der Wichtungen der Operanden. ... Die Eingabe einer UND-Operation ist beobachtbar, wenn die andere Eingabe eins, bei einer ODER-Operation, wenn die andere Eingabe null ist. ...



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

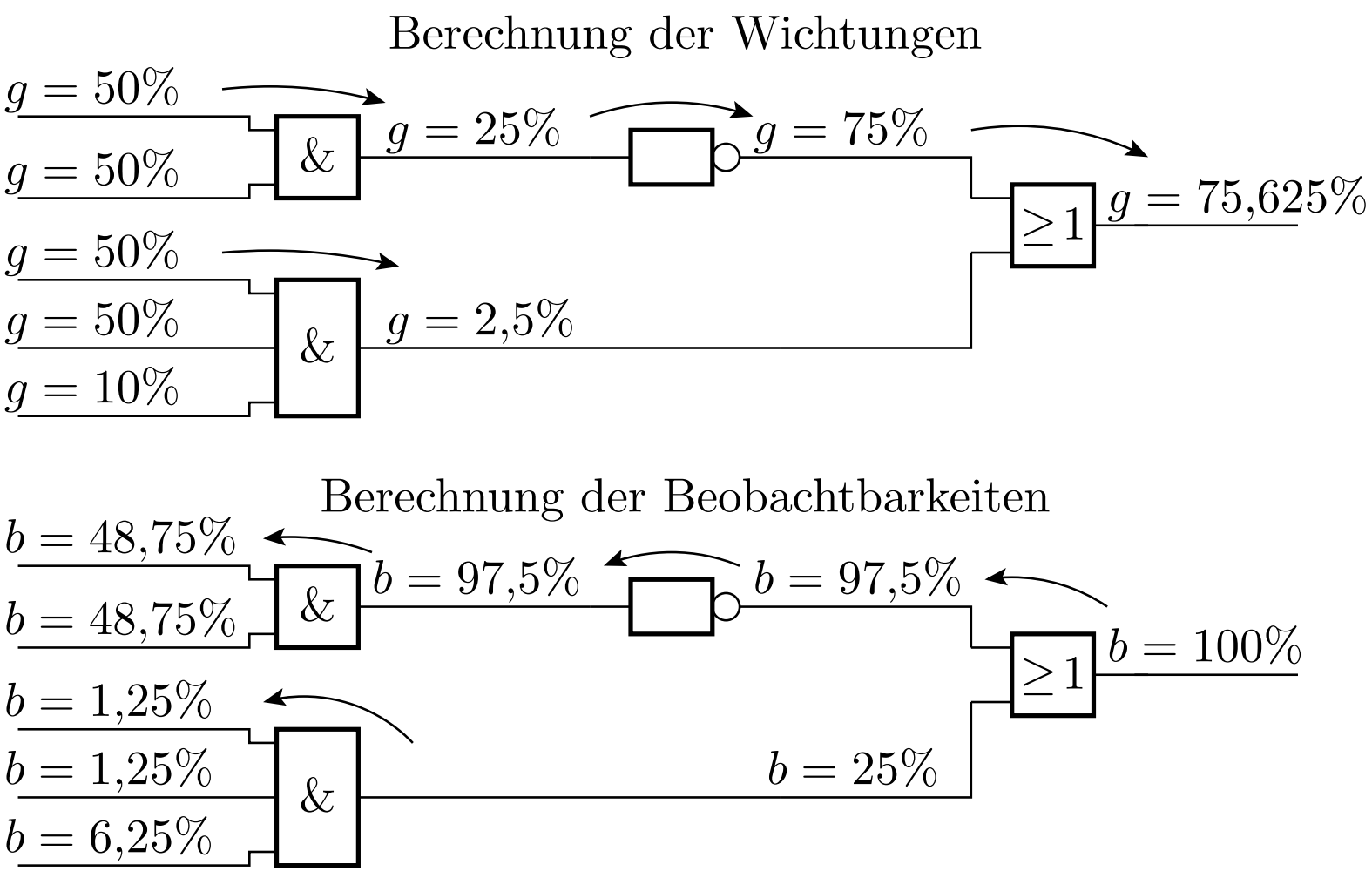


$$b(x_1) = b(y) \cdot (1 - g(x_2))$$
$$b(x_2) = b(y) \cdot (1 - g(x_1))$$
$$g(y) = 1 - (1 - g(x_1)) \cdot (1 - g(x_2))$$



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

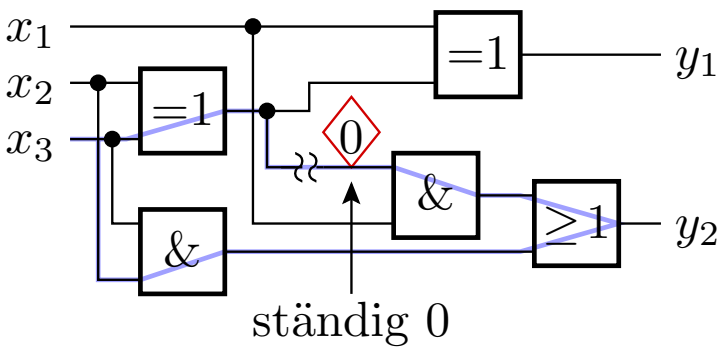
Wichtungen werden in Richtung und Beobachtbarkeiten entgegen der Richtung des Berechnungsflusses bestimmt.



Die Anregungswahrscheinlichkeit eines sa0-Fehlers ist die Wichtung $g(...)$ und die eines sa1-Fehler die Gegenwahrscheinlichkeit $1 - g(...)$ am Fehlerort. Die Nachweiswahrscheinlichkeit ist das Produkt aus Anregungs- und Beobachtungswahrscheinlichkeit.

Rekonvergente Auffächerung

Bei rekonvergenter Auffächerung werden gleiche Zufallswerte nach unterschiedlicher Verknüpfung mit anderen Werte logisch verknüpft. Für verknüpfte Werte, die von derselben Zufallsgröße abhängen, gelten die einfachen Regeln auf Folie 117 nicht. Berechnung aus den Auftrittshäufigkeiten der Eingaben für den Fehlernachweis:

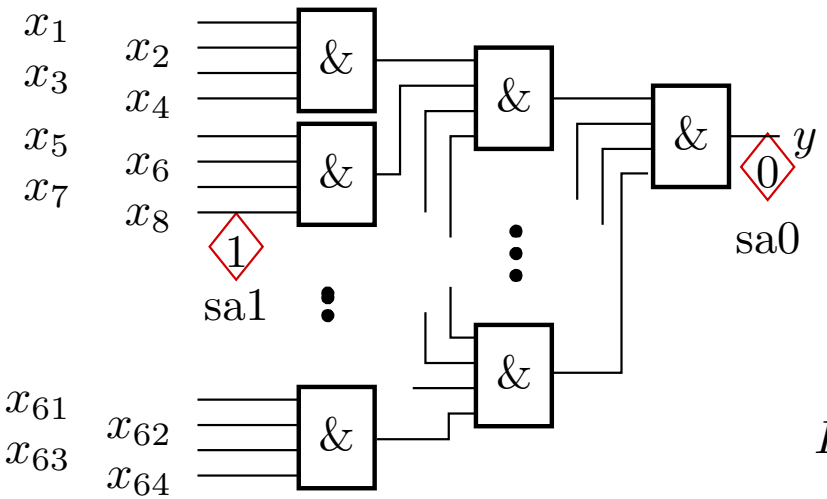


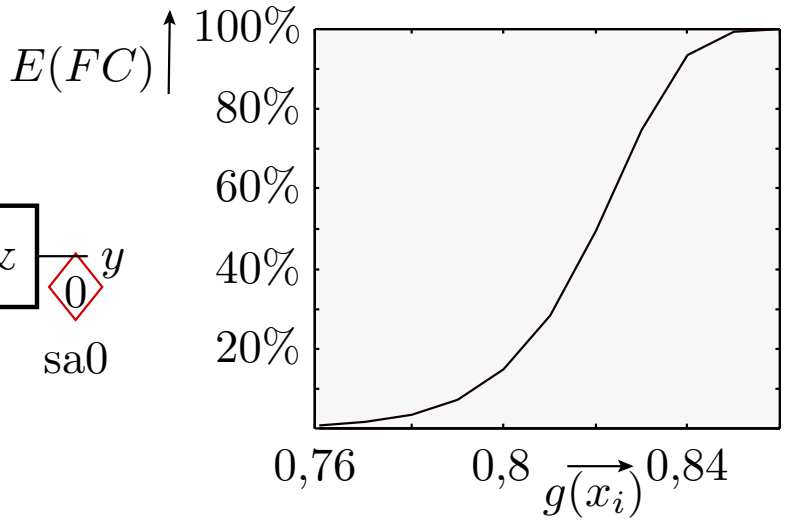
rekonvergente Auffächerung

Eingaben die den Fehler nachweisen

Eingabe x_3	x_2	x_1	Ausgabe y_2	y_1	Auftrittshäufigkeit		
0	0	0	0	0	0,125	0,1	0,1
0	0	1	0	1	0,125	0,05	0,1
0	1	0	0	1	0,125	0,15	0,2
0	1	1	1	0	0,125	0,2	0,05
1	0	0	0	1	0,125	0,05	0,2
1	0	1	1	0	0,125	0,2	0,05
1	1	0	1	0	0,125	0,05	0,2
1	1	1	1	1	0,125	0,2	0,1
Nachweiswahrscheinlichkeit:					0,25	0,4	0,1

Fehlerüberdeckung und Wichtung





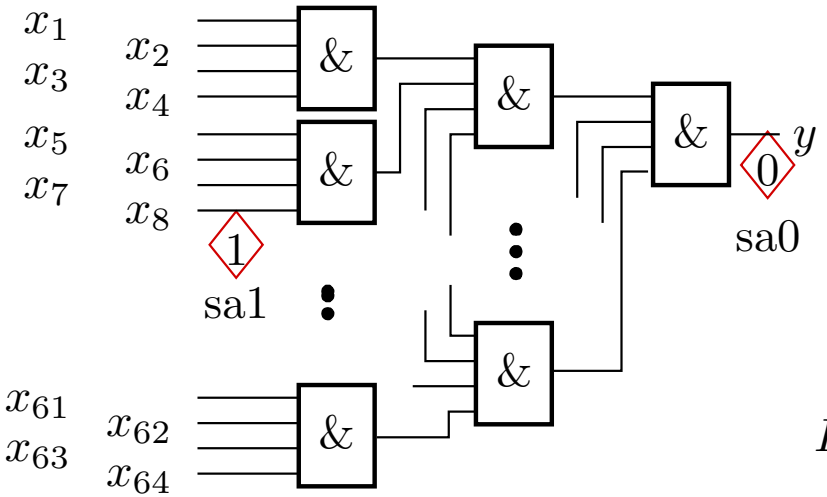
$E(FC)$ Fehlerüberdeckung für $n = 10^6$ Testschritte

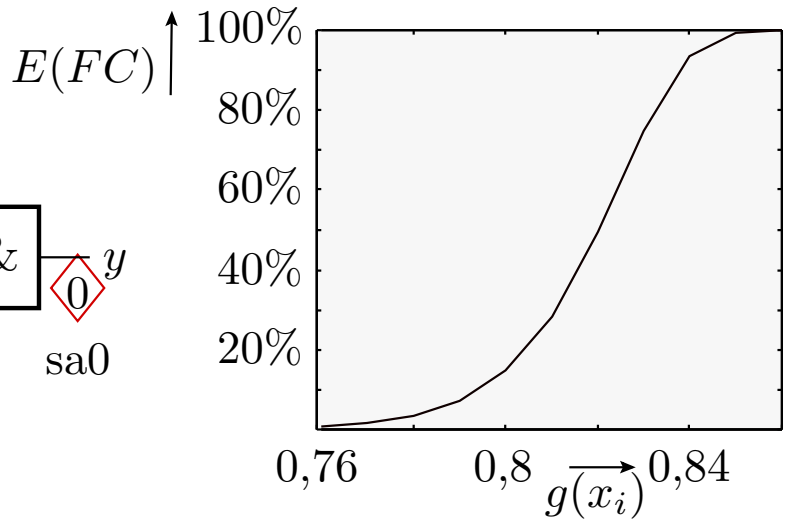
Angenommene Fehler: Für je einen der 64 Eingänge ständig 1:

$$p_{\text{sa1}} = g^{63} \cdot (1 - g)$$

Für den Ausgang ständig 0:

$$p_{\text{sa0}} = g^{64}$$





$E(FC)$ Fehlerüberdeckung für $n = 10^6$ Testschritte

Zu erwartende Fehlerüberdeckung als Mittelwert der Nachweiswahrscheinlichkeit:

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

Eine Wichtung von $g = 86\%$ verringert die erforderliche Testsatzlänge für $FC \geq 99\%$ von $n \gg 2^{64}$ auf $n \approx 10^6$.

Fehlerorientierte Wichtungsauswahl

Statt einheitlicher Wichtung aller Eingabesignale

- Festlegung einer individuellen Wichtung für jeden Eingang,
- Wichtungsumschaltung jeweils nach einem längeren Zufallstest.

Ein pragmatischer Ansatz dazu aus[6]:

1. Festlegung einer größeren Menge von Modellfehlern.
2. Längerer Test mit ungewichteten Zufallswerten und Abhaken aller damit nachweisbaren Modellfehler.
3. Suche für die restlichen Modellfehler eine Eingabewichtung, die deren Nachweiswahrscheinlichkeiten erheblich erhöht.
4. Längerer Test mit den so gewichteten Zufallswerten und Abhaken aller damit nachweisbaren Modellfehler.
5. Wenn erforderlich, Wiederholung von Schritt 3 und 4.

[6] J. Hartmann, G. Kemnitz: How to do weighted random testing for BIST? ICCAD 1993.

Auswahl der Wichtungswerte

- Für alle Modellfehler, die der ungewichtete Zufallstest nicht nachweist, gezielte Berechnung einer Eingabe (D-Algorithmus) mit möglichst vielen »X« (Don't Care) Eingabewerten.
- Begrenzung der Wichtung auf 5 Werte:

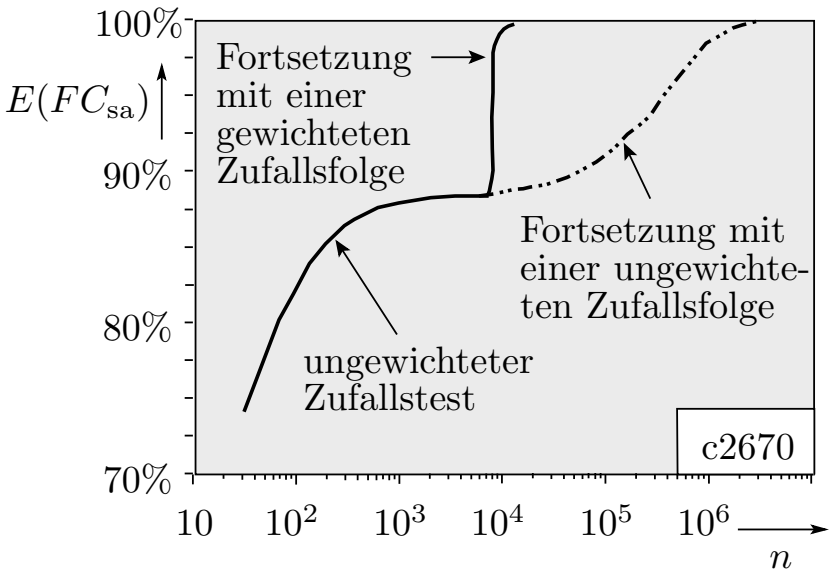
$$g(x_i) = \begin{cases} 0 & x_i \in \{0, \mathrm{X}\} \\ 2^{-\#E_{\mathrm{AND}}} & \#N \gg \#E \\ 0{,}5 & \text{sonst} \\ 1 - 2^{-\#E_{\mathrm{AND}}} & \#N \ll \#E \\ 1 & x_i \in \{1, \mathrm{X}\} \end{cases}$$

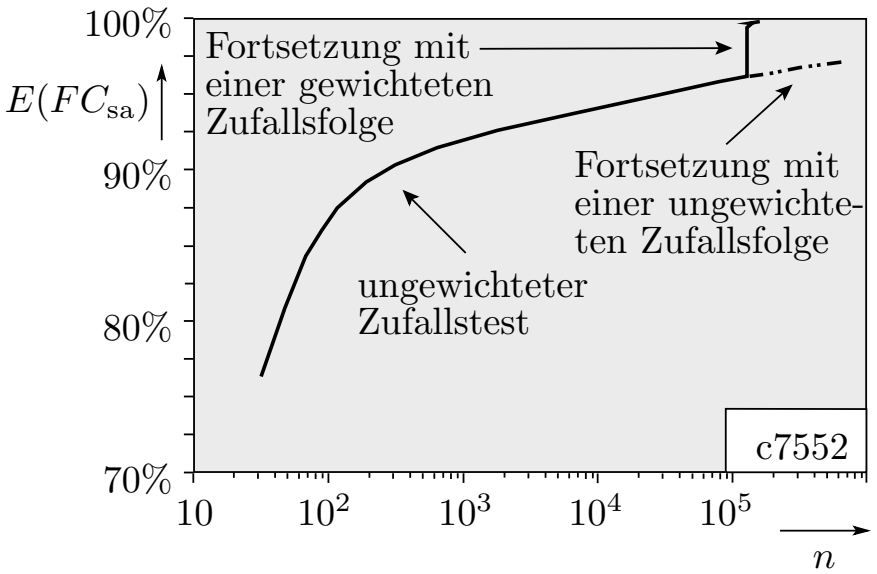
($\#N$ – Anzahl der Nullen; $\#E$ – Anzahl der Einsen für Eingang x_i in den Testeingaben; $\#E_{\mathrm{AND}} \in \{2, 3, 4, \ldots\}$ Anzahl der [N]AND-verknüpften Zufallsfolgen zur Wichtungserzeugung).

- Die zweite und weitere Wichtungsberechnungen berücksichtigen nur noch die Testeingaben bis dahin nicht nachgewieser Fehler.

Experiment mit den Schaltungen c2670 und c7552[7]

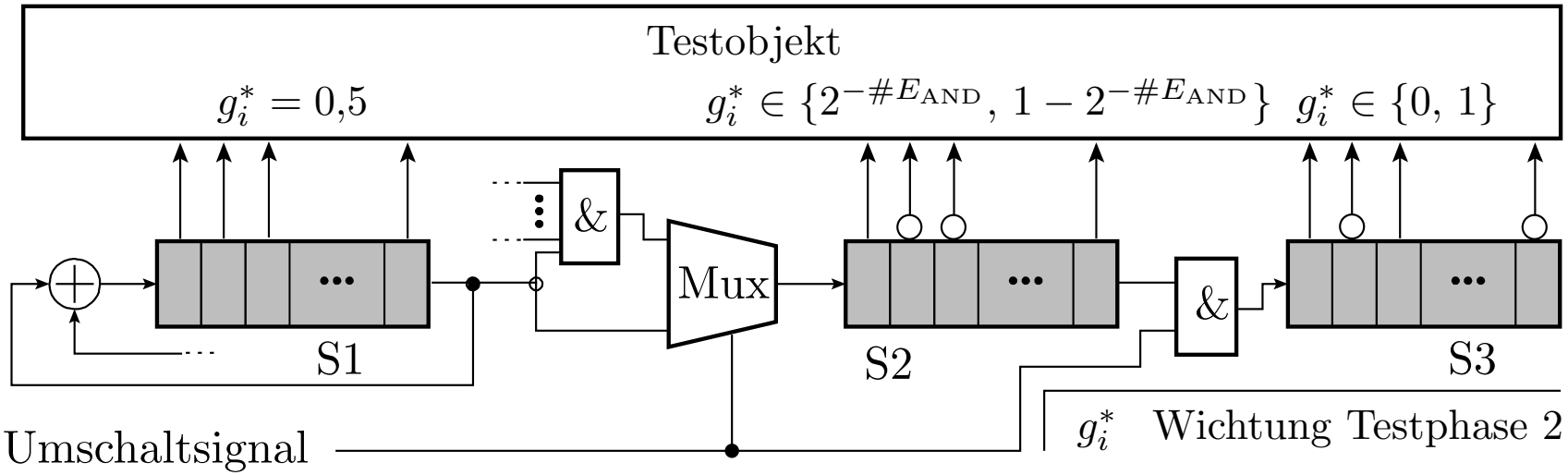
- Test mit 10^4 bzw. 10^5 ungewichteten Zufallsmustern, die 90% bzw. 95% der Haftfehler nachweisen.
- Gezielte Testberechnung für die restlichen Haftfehler.
- Individuelle Wichtung aller Eingabebits zur Maximierung der mittleren Auftrittshäufigkeit der berechneten Testeingaben.





[7] Kombinatorische Benchmarkschaltungen zum Vergleich von Testlösungen. Die Zahl hinter dem »c« ist die Anzahl der Signalleitungen.

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 $\#E_{\text{AND}}$ Ausgabefolgen von S1 und für S2 durch »UND 0«. Erzeugung der Wichtungen $1 - 2^{-\#E_{\text{AND}}}$ und »1« durch Inverierung.

Nicht nennenswert aufwändiger als ohne Wichtung.