

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

Prof. G. Kemnitz

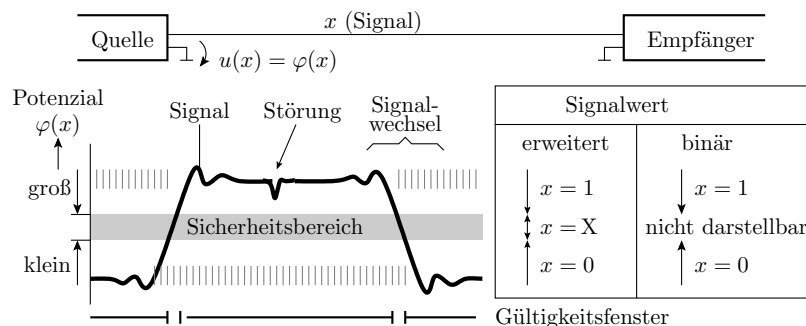
17. April 2020

Contents

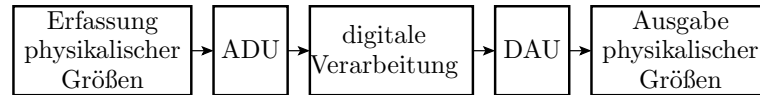
1	Schaltungsstrukt. & Test	1	3	Testberechnung	22
1.1	Gatterschaltungen	2	3.1	Fehlersimulation	22
1.2	Speicher	3	3.2	D-Algorithmus	23
1.3	Mixed-Signal	6	3.3	Implikationstest	25
1.4	Rechnerstrukturen	7	3.4	Suchraumstrukturierung	25
1.5	Baugruppen	8	3.5	Komplexe Funktionsbausteine	26
1.6	Testbus	10	3.6	Sequentielle Schaltungen	27
2	Fehlermodellierung	12	4	Selbsttest	28
2.1	Verbind.- & Trans.-Fehler	14	4.1	Pseudo-Zufallsregister	29
2.2	Haftfehler	16	4.2	Signaturregister	30
2.3	FC fehlerorientierte Suche	17	4.3	Selbsttest mit LFSR	32
2.4	FC Zufallstest	18	4.4	Fehlerorientierte Wichtung	33
2.5	Verzögerungsfehler	20			
2.6	IDDQ-Test	21			

1 Schaltungsstrukt. & Test

Digitalisierung



- Wertediskretisierung: groß, klein und Zwischenbereich für ungültig. Abtastung im Gültigkeitsfenster.
- Macht robust gegen Verfälschungen durch Rauschen, induktives und kapazitives Übersprechen, Fertigungsstreuungen, Alterung, ...



Verarbeitung physikalischer Größen:

- Analoge Verarbeitung (stetiger Wertebereich der dargestellten Größen): Erfassung, Vorverarbeitung und Ausgabe.
- Digital Verarbeitung, oft Rechnerstrukturen, Festlegung der Funktion durch Software.

Im Vergleich zur früheren ausschließlich analogen Verarbeitung:

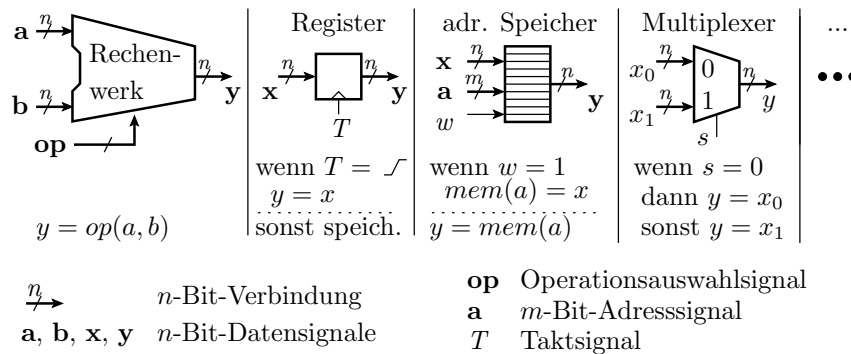
- wesentlich mehr Bauteile, aber kleinere Fertigungsmaße,
- schneller, so dass Berechnungen sequentielle durch Programme statt parallel durch dedizierte Schaltungen, ...
- Entwurf, Test, Fehlersuche, Anpassung an neue Aufgaben, ... einfacher.

Test, Fehler, ...:

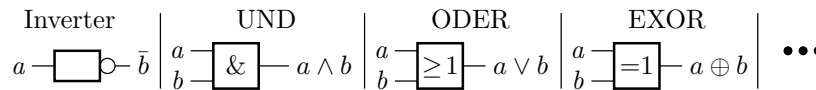
- Schwer zu findende Fehler hauptsächlich in Digitalteil und SW.

1.1 Gatterschaltungen

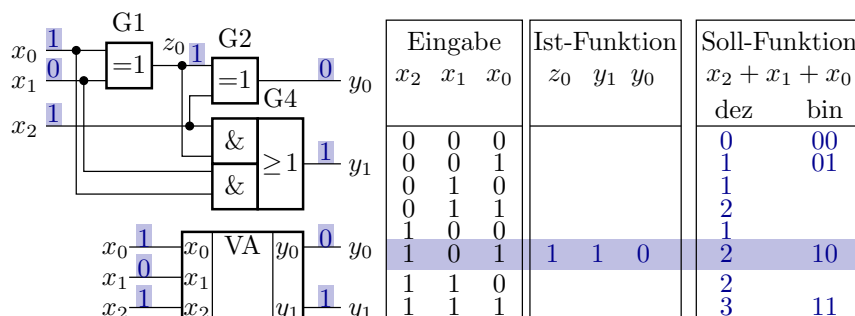
Grundbausteine von Rechnern



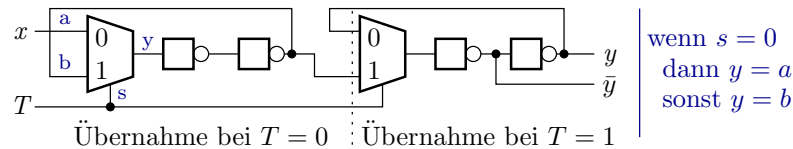
- Rechenwerke, Register, ... bestehen aus Logikgattern:



- 1-Bit-Addierer (Volladdierer):

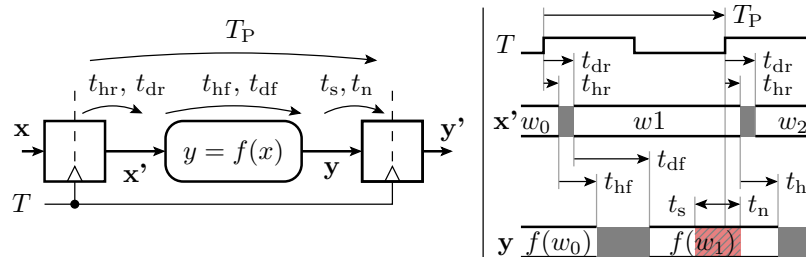


- 1-Bit-Registerzelle: Übernahme bei $T : 0 \rightarrow 1$, sonst speichern



Diskretisierung der Zeitabläufe

Einrahmen der Verarbeitungsfunktion mit Registern. Die Verarbeitung beginnt mit der aktiven Taktflanke der Eingaberegister und endet mit (der nächsten) aktiven Taktflanke der Ergebnisregister.



Hinreichende Bedingung für die Abtastung garantiert gültiger Werte:

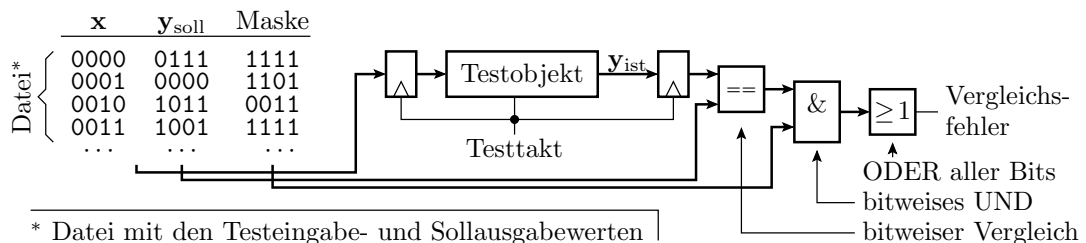
$$T_P \geq T_{P.min} = t_{dr} + t_{df} + t_s$$

$$t_n \leq t_{n.max} = t_{hr} + t_{hf}$$

(T_P – Taktperiode; t_{hr} , t_{dr} , t_s , t_n – Halte-, Verzögerungs-, Vorhalte- und Nachhaltezeit der Register; t_{hf} , t_{df} , t_s , t_n – Halte- und Verzögerungszeit der Verarbeitungsfunktion).

Robust gegen Glitches außerhalb der Abtastfenster.

Test digitaler Schaltungen



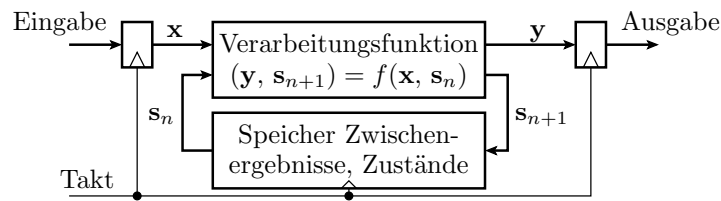
Eingabebereitstellung und Ausgabeüberwachung / -aufzeichnung.

- Manuell, nur Logikverhalten mit Schaltern und LEDs.
- Automatisch einschließlich Laufzeitkontrolle mit Signalgenerator und Logikanalysator¹.
- Tester: Bereitstellung Eingabe-, Soll- und Maskenwerte, Soll/Ist-Vergleich. Ausgabe OK / Protokoll der Abweichungen.

1.2 Speicher

Systeme mit Gedächtnis

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



Rechnerstrukturen, Ablaufsteuerungen, ... bilden aus Eingaben \mathbf{x} und gespeicherten Werten \mathbf{s} Ausgaben \mathbf{y} und zu speichernde Werte \mathbf{s} :

$$(\mathbf{y}, \mathbf{s}_{n+1}) = f(\mathbf{x}, \mathbf{s}_n)$$

Ablauf eines Testschritts für $f(\dots)$:

1. (viele) Schritte zur Einstellung der Testeingabe für \mathbf{s} .
2. ein Schritt : $(\mathbf{y}, \mathbf{s}_{n+1}) = f(\mathbf{x}, \mathbf{s}_n)$ für den Test von $f(\dots)$.
3. (viele) Schritte zur Kontrolle/Ausgabe von \mathbf{s} .

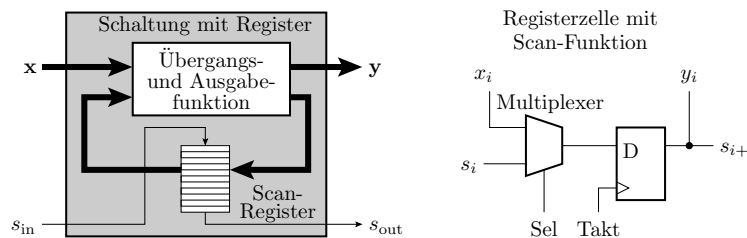
Wünschenswert/Voraussetzung: Testlese- und -Schreibzugriff für Zwischenergebnisse und Zustände ohne, dass \mathbf{s} durch $f(\dots)$ verarbeitet wird, damit:

- Testsuche,
- Fehlersimulation, ...

wie bei einem isolierten Test der Verarbeitungsfunktion.

Die (Test-) Lese-, die (Test-) Schreib- und die Speicherfunktion des Zusatzspeichers werden beim Test von $f(\dots)$ mit getestet. Extra-Betrachtung nicht erforderlich.

Scan-Verfahren

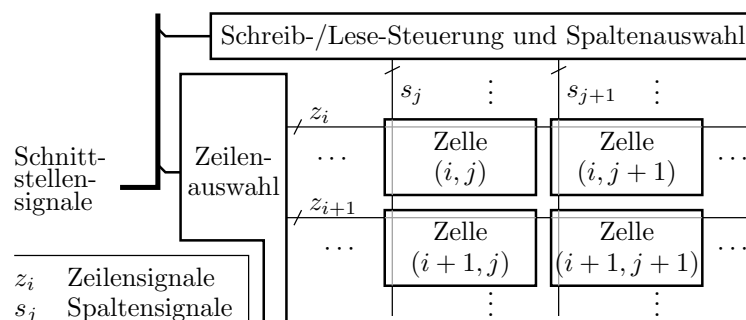


Lese- und Schreibzugriff während des Tests durch Umschalten des Zustandsspeicher in ein Schieberegister.

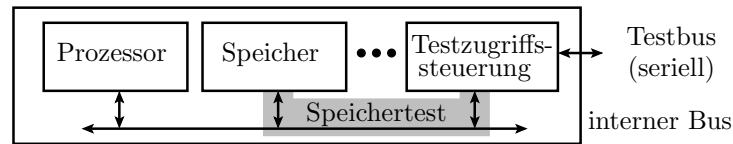
- Mindestschaltungsaufwand ein Multiplexer je Speicherzelle.
- Ablauf eines Testschritts: r Schiebeschritte zum Beschreiben des Zustandsspeichers, Testschritt, r Schiebeschritte zum Lesen (und Überschreiben) des Zustandsspeichers.

Blockspeicher

Große Speicher besteht im Kern aus einer regelmäßigen 2D-Anordnung von flächenminimierten Speicherzellen. Die Grundfunktionen (nur lesbar, beschreib- und lesbar, ...) hängen von der Zellenfunktion ab. Die Zeilen- und Spaltenauswahl legt die Zugriffsmöglichkeiten fest (Zugriffsbreite, Portanzahl, ...) .



Test von Blockspeichern



Abhängig von den zu erwartenden Fehlern in den in den Zellen:

- Haftfehler (Lesewert ist ständig null oder ständig eins).
- Übergangsfehler (Wert nur in einer Richtung änderbar).
- Zerstörendes Lesen (Löschen des Inhalts beim Lesen).
- Gegenseitige Beeinflussung unterschiedlicher Zellen,
- ...

gibt es skalierbare Testabläufe, anpassbar auf Speichergröße und -organisation, die dafür aber einen isolierten Test verlangen, z.B. über einen Testbus, der im Testmodus die Kontrolle interner Busse übernimmt.

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

Mehrfaches Durchwandern des Speichers in unterschiedlicher Reihenfolge mit der Operationsfolge Zelle Lesen, Wert kontrollieren und inversen Wert zurückschreiben. Zielfehler 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 $1 \rightarrow 0$ $0 \rightarrow 1$	$W(i)1, R(i)1, W(i)0, R(i)0$ $W(i)0, R(i)0, W(i)1, R(i)1$
	Stuck-open-Fehler	kein Zugriff auf Speicherzelle i (Ausgabe des Wertes der vorherigen Leseoperation)		$W(i)0, R_1(j), R(i)0, W(i)1,$ $R_0(j), R(i)1$
	zerstören-des 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

$C(\dots)$ Inhalt Zelle ...

$R(i)0$ Lese Inhalt Zelle i und vergleiche mit Sollwert 0

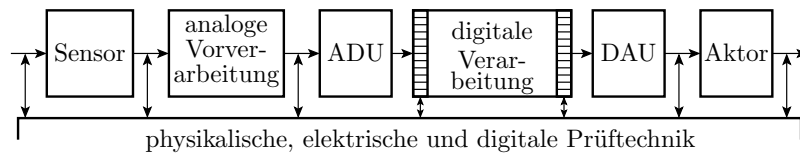
$R(i)1$ Lese Inhalt Zelle i und vergleiche mit Sollwert 1

$R_0(j)$ Lese eine andere Zelle, in der 0 steht

$R_1(j)$ Lese eine andere Zelle, in der 1 steht

1.3 Mixed-Signal

Test analoger und Mixed-Signal-Bausteine



- Sensoren: Vorgabe physikalischer Eingaben, Kontrolle elektrischer Ausgaben.
- Analoge Vorverarbeitung und ADU: Vorgabe elektrischer Eingaben (Spannungen, Ströme, Signalverläufe), Kontrolle digitalisierter Werte.
- DAU und Aktoren: Vorgabe digitaler Werte, Kontrolle elektrischer oder physikalischer Größen.

Modularer Funktionstester

Der komplette Test von Systemen mit physikalischer Anbindung verlangt die Bereitstellung von Testverläufen für die Sensorsignale und Kontrollen der Ausgabesignale. Typische Lösung ist ein Rechner mit einem modular zusammensetzbaren System aus



- Logikgenerator- und Logikanalysatorbaugruppen,

- DAU- und ADU-Baugruppen,
- programmierbaren Spannungsversorgungen,
- Baugruppen für Busschnittstellen (RS232, SPI, CAN, ...),
- Lastschaltungen, Adapter, ...

HIL- (Hardware in the Loop) Tester

Nachbildung der Systemumgebung physikalisch, als Simulationsmodell oder gemischt. Maschinen und Anlagenbau:

- Physikalische Simulation der gesteuerten Maschine oder Anlage,
- 3D-Visualisierung des physikalischen Verhaltens,
- Untersuchung von Grenzwert- und Gefahrensituationen.



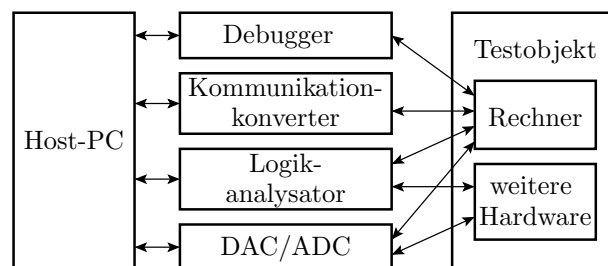
Fahrzeugbau, Luft- und Raumfahrt

- physikalische Simulationen von Motoren, Lenksystemen bis hin zu kompletten Flugzeugen,
- Nachstellung komplizierter Testsituationen im Labor (fahrendes Auto, Flugzeug in der Luft, ...)

Jedes Simulationsmodell hat Genauigkeitsgrenzen. Kein vollständiger Ersatz für den Test in der Anwendungsumgebung.

1.4 Rechnerstrukturen

Prüftechnik für Mikrorechners



Kompliziertere Algorithmen, insbesondere auch Funktionen zur Selbstüberwachung und Fehlerbehandlung werden in Software realisiert. Systemstruktur: Ein oder mehrere Rechner, Speicher, anwendungsspezifische Hardware, ...

Erforderliche Prüftechnik zusätzlich zu der für die digitalen, analogen und MixedSignal-Schaltungsteile:

- Logikgenerator- und Logikanalysatorbaugruppen,
- DAU- und ADU-Baugruppen,
- programmierbare Spannungsversorgungen,
- Busschnittstellen (RS232, SPI, CAN, ...), ...

Software-Debugger mit den Funktionen:

- Programme laden und mit Testausgaben starten,
- Schrittbetrieb, im Haltzustand Lesen/Schreiben von Variablen und Hardware-Spezialregister,
- Trace-Aufzeichnung, ...

1.5 Baugruppen

Baugruppentest

Elektronische Baugruppen: Leiterplatte (Verdrahtungsträger), bestückt mit Schaltkreisen, Widerständen, Steckverbindern, ...

Leiterplatte und Bauteile sollten vom jeweiligen Hersteller gründlich geprüft und hinreichend zuverlässig sein:

1. Test auf Bestückungsfehler am spannungsfreien Objekt (statische Tests):
 - optische Inspektion,
 - elektrische Zweipunktmessungen (Widerstand, Kapazität, ...),
 - In-Circuit-Test.
2. Test unter Betriebsspannung (dynamische Tests):
 - Spannung anlegen, Kontrolle der Stromaufnahme, auf Erwärmung und Rauchbildung,
 - digitaler In-Circuit-Test,
 - Test der Gesamtfunktion.

Die Reihenfolge:

1. Tests ohne Versorgungsspannung,
2. Beseitigung aller Kurzschlüsse und Unterbrechungen
3. Rauchttest,
4. Test mit Betriebsspannung

vermeidet, dass Bauteile durch Bestückungs- und Verdrahtungsfehler zerstört werden.

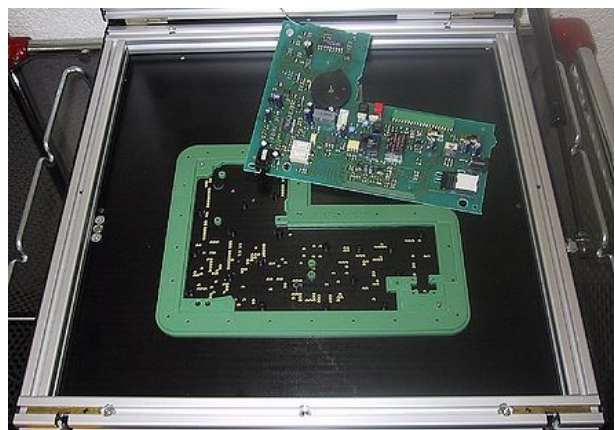
Auf ganzheitliche Funktionstests wird verzichtet

- bei normalen Anforderungen an die Verlässlichkeit,
- Serienfertigung und hinreichend zuverlässigen Komponenten,

weil

- Baugruppenhersteller nicht die Informationen für eine fehlerorientierte Testauswahl für bauteilinterne Fehler haben,
- isolierte Zufallstests für Bauteile die $c \gg 1$ -fache effektive Testsatzlänge gegenüber ganzheitlichen Zufallstests haben,
- Verbindungsfehler mit den spannungsfreien Verfahren besser zu erkennen und zu lokalisieren sind.

MDA (Manufacturing Defect Analyzer)



Prüfsysteme, die im spannungsfreien Zustand nach:

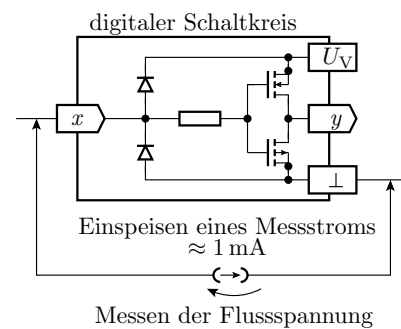
- Unterbrechungen,
- Kurzschlüssen und
- Fehlbestückungen

mit elektrischen Zweipunktmessungen (Widerstand Kapazität, Diodenkennlinie, ...) suchen.

In der Serienfertigung erfolgt die Kontaktierung mit einem mit Unterdruck angesaugten Nadeladapter. Die Nadeln sind mit reine Relais²-Matrix zur Verbindung mit den Prüfgeräten angeschlossen.

Suche potentieller Bestückungs- und Verdrahtungsfehler mit elektrischen Zweipunktmessungen:

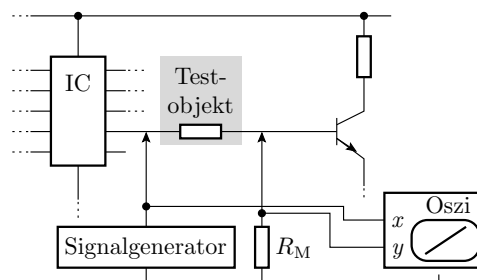
- Stromeinspeisung und Messung der Spannung oder
- Spannungseinspeisung und Strommessung.



Bauteiltypische Strom-Spannungsbeziehungen für Sinuseingabe:

- Widerstand: Gerade,
- Kondensator: Elypse,
- Diode: Kennlinie mit Knick, ...

Schaltkreise haben an ihren Eingängen i.Allg Schutzdioden zur Versorgungsspannung und Masse



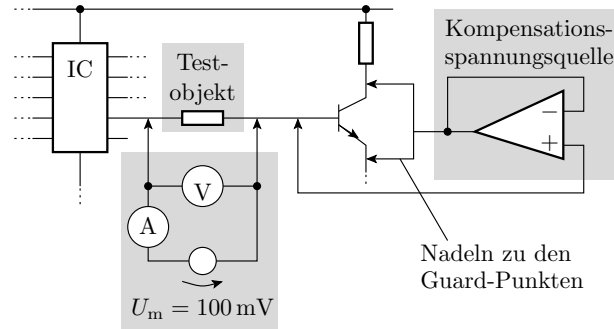
- Die Strom-Spannungs-Beziehung zwischen zwei Punkten hängt nicht nur vom Bauteil zwischen den Nadeln, sondern von allen Strompfaden, im Beispiel durch Transistor und Schaltkreis ab.
- Bestimmbar durch Ausprobieren an einem »Golden Device«.

Problematisch können sein

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

²Relais: Mit Elektromagneten gesteuerte Schalter.

Analoger In-Circuit Test

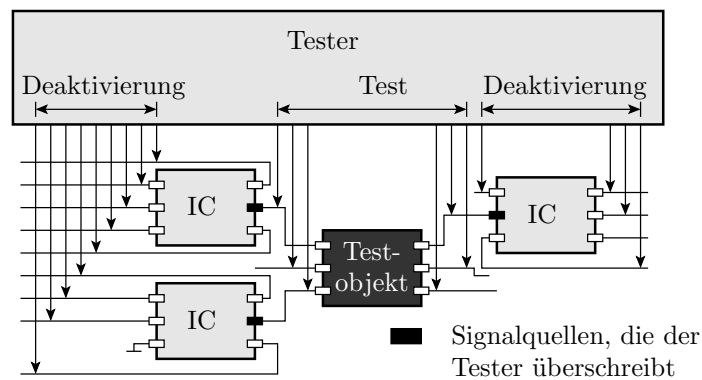


Unterdrückung von Parallelströmen zum Testobjekt durch Kompensation der Spannungsabfälle über den wegführenden Bauteilen auf einer Testobjektseite auf null über »Guard-Punkte«.

... Erlaubt einen isolierten Zweipoltest.

- Vereinfacht die Testauswahl, Sollwertfestlegung, ...
- Mindert die Häufigkeit von Fehlklassifikationen.
- Für digitale Schaltkreise ungeeignet.

Digitaler In-Circuit-Test



- 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).

1.6 Testbus

Testbus – JTAG (IEEE 1149.1)

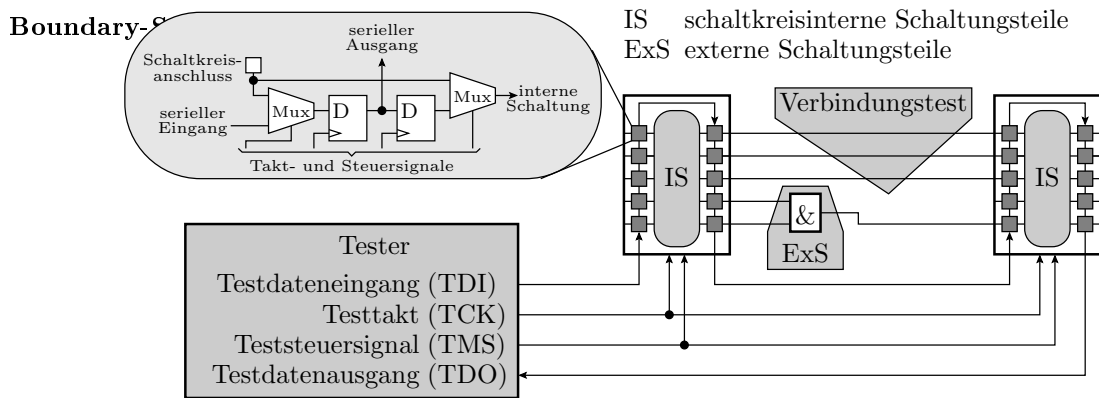
Steuerung integrierter Testhilfen in Schaltkreisen und auf Baugruppen über einen seriellen Bus (minimale Anzahl von Verbindungsleitungen):

- Übernahme der Funktion der Nadeln für den Baugruppentest (Steuerung 0, 1, hochohmig und Lesen der Logikwerte der IC-Anschlüsse),
- statt optischer Bestückungskontrolle, Lesen der Hersteller und Identifikationsnummern aus den Bauteilen.

Weitere typische Testbusfunktionen:

- Selbst-, Scan- und isolierter Test interner Schaltungsteile,
- Programmierschnittstelle für Mikrorechner und FPGA,
- Schnittstelle zum In-Circuit-Debugger, ...

JTAG: Testbus-Standard IEEE 1149.1

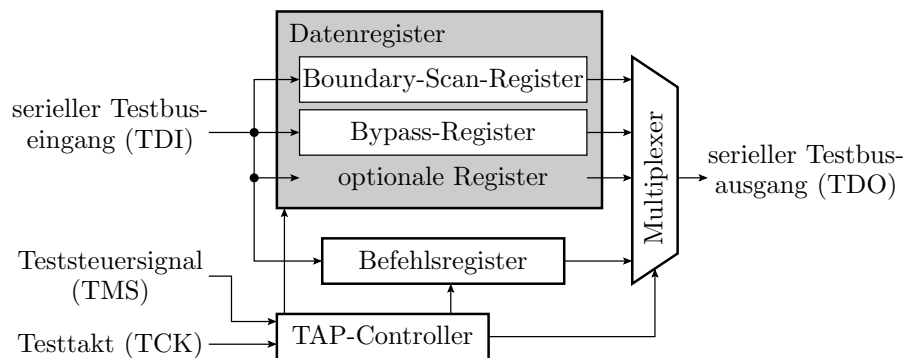


Ersatz der mechanischen Nadeln durch »silicon nails« (seriell beschreibbare Register an den Schaltkreisanschlüssen, im Normalbetrieb überbrückt). Alternative zu den teuren, für jede Baugruppe speziell anzufertigenden Nadeladaptern.

Ablauf eines Testschritts für den Baugruppentest:

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

JTAG-Testbusarchitektur der Schaltkreise



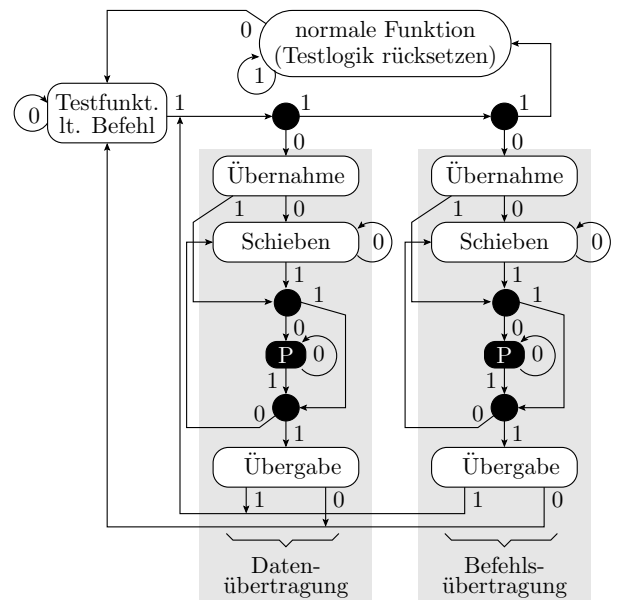
Eine Boundary-Scan-Implementierung umfasst:

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

- Automat mit 16 Zuständen
- Kantenauswahl über TMS-(Test Mode Select) Signal

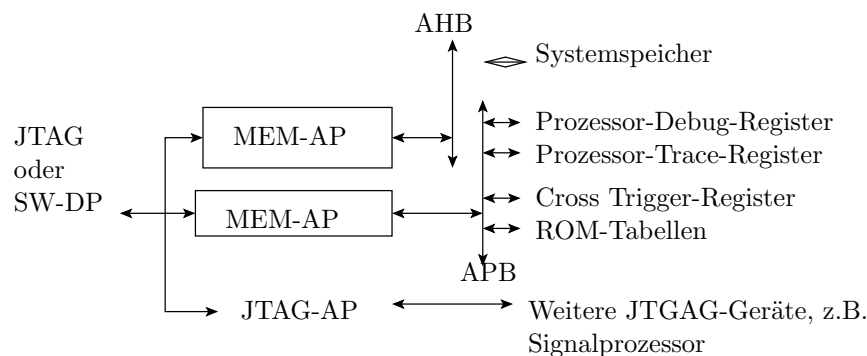
Typischer Testablauf:

- Befehlsregister lesen (Bestückungskontrolle),
- Bauteilnummern lesen (Bestückungskontrolle),
- Bauteilnummern lesen(Bestückungskontrolle),
- Einen Teil der Schaltkreise auf Bypass setzen. Für die anderen Datenregister auswählen.
- Verbindungstest. ...



Zur Bestückungskontrolle übergibt das Befehlsregister beim Lesen ein Muster zur Erkennung von Unterbrechungen der Schieberegisterkette auf der Baugruppe.

ARM-Testbusarchitektur



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

2 Fehlermodellierung

Fehlermodellierung und Testauswahl

Fehlermodellierung und fehlerorientierte Testauswahl bzw. Testbewertung ist nur für Systemtypen erforderlich, für die ein ausreichend sicheres Erkennen, Lokalisieren und Beseitigen der auftretenden Fehler Probleme bereitet:

Frei strukturierte Schaltkreisteile mit $\gg 20$ Eingabebits:

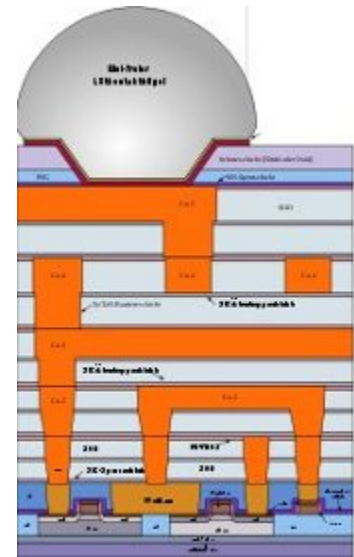
- kein Test mit allen Eingaben möglich,
- kein isolierter Test in kleinen Teilen möglich, ...

Die Fehlermodellierung und Testberechnung wurde ursprünglich für diesen Schaltungstyp entwickelt.

Unproblematisch:

- Baugruppen: Zweipunktmessungen.
- Digitalschaltung bis max 10...20 Eingabebits: Test mit allen Eingabemöglichkeiten, ...

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 weggeätztes) und
- überflüssiges (zu viel aufgetragenes, zu wenige weggeätztes)

Material.

Einteilung in lokale und globale Fehler

Globale Fehler:

- Fehlerhafte Schichteigenschaften durch Prozesssteuerfehler. Betroffen sind alle Strukturelemente derselben Halbleiter-, Leitungs- oder Isolationsschicht.
- Großflächig überflüssiges oder fehlenden 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 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 elektrische Eigenschaften an den Anschlüssen incl. Versorgungsstrom.

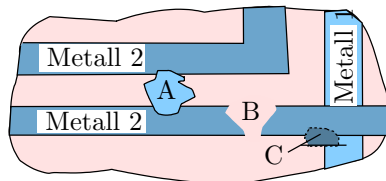
Großflächig überflüssiges oder fehlenden Material:

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

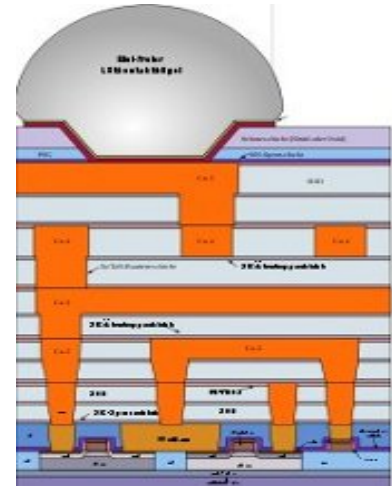
Anspruchsvoll ist die Suche nach kleinen Defekten, die überall sein können und nur selten FF verursachen.

2.1 Verbind.- & Trans.-Fehler

Verbindungs- und Transistorfehler



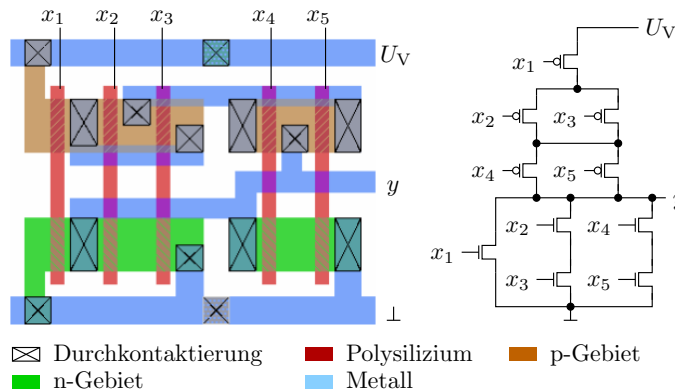
- A zusätzliches Metall
- B fehlendes Metall
- C fehlende Isolation



Einzelfehler durch fehlendes und überflüssiges Material:

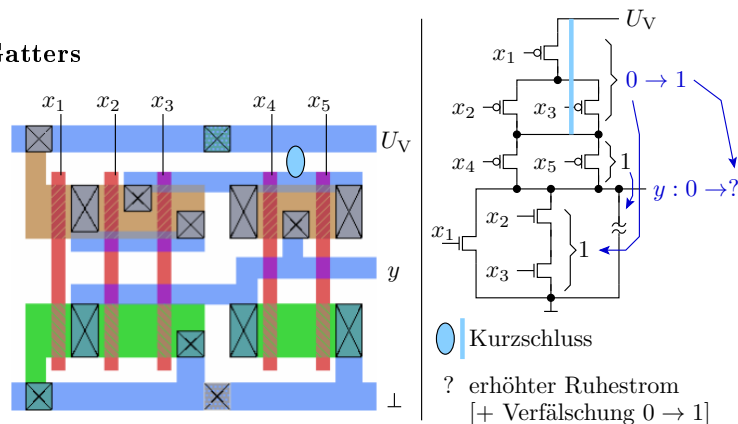
- kurzgeschlossene und unterbrochene Verbindung,
- nicht richtig ein- oder ausschaltende Transistoren,
- Leckströme ohne Beeinträchtigung der logischen Funktion, ...
- überhöhte Stromdichten oder Feldstärken, die zu Frühausfällen führen.

Transistorebene



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

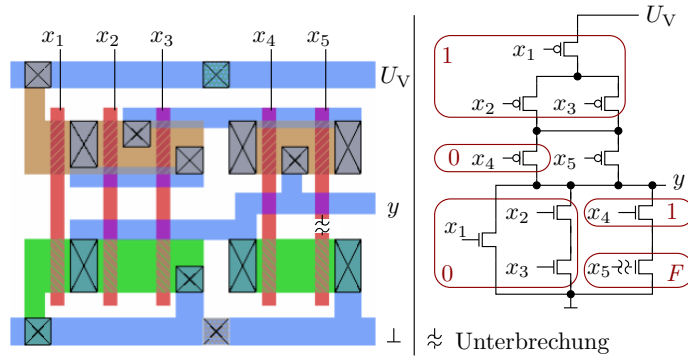
Kurzschluss im Gatters



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.

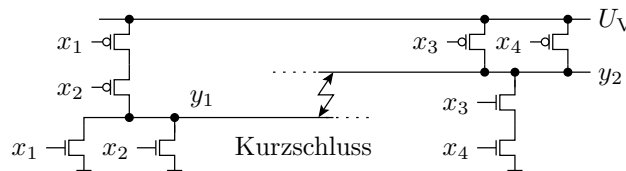
Offenes Gate



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

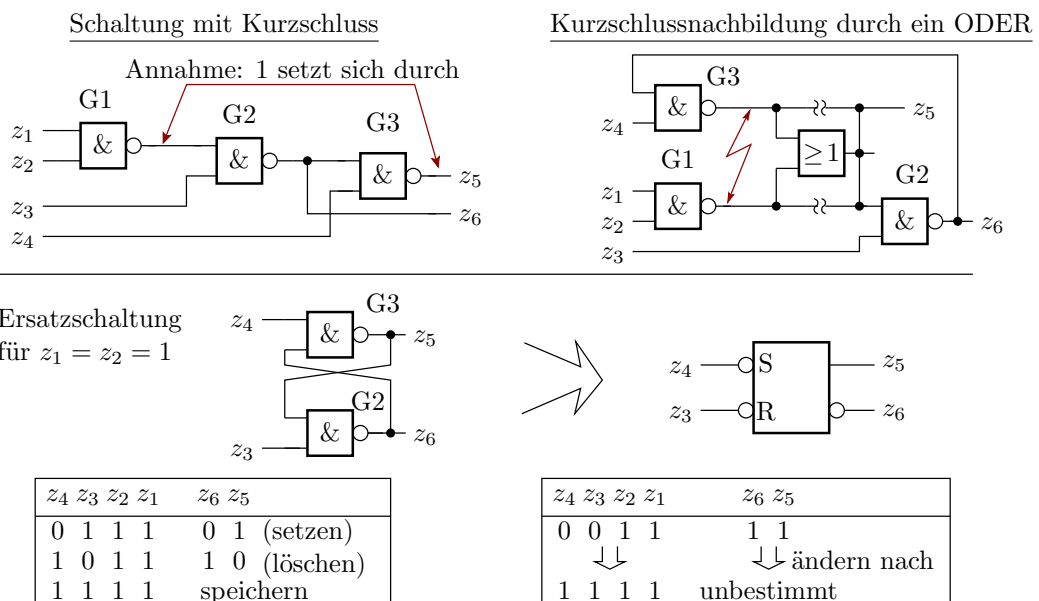


Mögliche Nachweisbedingungen:

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 $y_{2\text{Soll}} = 1$)

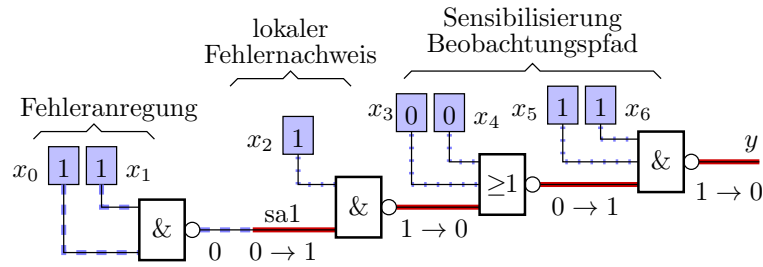
Ob sich dabei $y_1 = y_2 = 0$ oder $y_1 = y_2 = 1$ durchsetzt, hängt von den Transistorbreiten, bzw. Transistorsteiheiten ab.

Zusätzliches Speicherverhalten durch Kurzschluss



2.2 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:

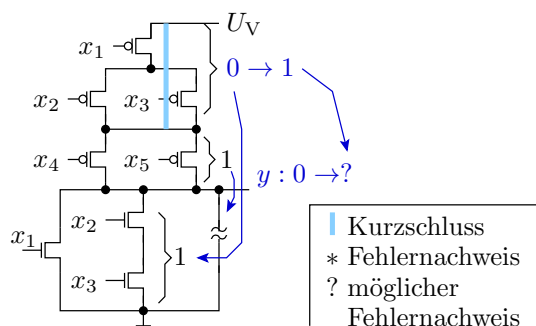
- Gatter mit Kurzschlüssen und Unterbrechungen.
- Unterbrochene Verbindungen.
- Teilschaltungen mit Kurzschlüssen, ...

Vereinfachte Modellfehler:

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

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

Kurzschluss im Gatters



$x_3 x_2 x_1$		sa0(x3)	sa0(x2)	sa0(x1)
0 0 0				*
0 0 1		?		*
0 1 0			*	
0 1 1				?
1 0 0				*
1 0 1				?
1 1 0			*	*
1 1 1				?
p_i		5/32	2/32	2/32

Statischer Ruhestrom und eventuell $y : 0 \rightarrow 1$ für:

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

Haftfehler mit Nachweisbedingung $a = (\overline{x_4} \vee \overline{x_5}) \wedge B(y : 0 \rightarrow 1)$:

- $sa0(x_1) : \overline{(x_1 \rightarrow 0) \vee (x_2 \wedge x_3) \vee (x_4 \wedge x_5)} = (\overline{x_2} \vee \overline{x_3}) \wedge a$
- $sa0(x_2) : \overline{x_1 \vee ((x_2 \rightarrow 0) \wedge x_3) \vee (x_4 \wedge x_5)} = x_3 \wedge \overline{x_1} \wedge a$
- $sa0(x_3) : \overline{x_1 \vee (x_2 \wedge (x_3 \rightarrow 0)) \vee (x_4 \wedge x_5)} = x_2 \wedge \overline{x_1} \wedge a$

Offenes Gate

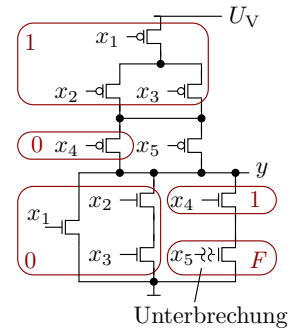
Wenn $a = \bar{x}_1 \wedge (\bar{x}_2 \vee \bar{x}_3) \wedge x_4$ und

- $a_{n-1} \wedge x_5 = \uparrow \wedge B(y : 1 \rightarrow 0) ?L$
- $x_5 = 0 \wedge B(y : 0 \rightarrow 1) ?L$
- $x_5 = 0 \wedge ?I$

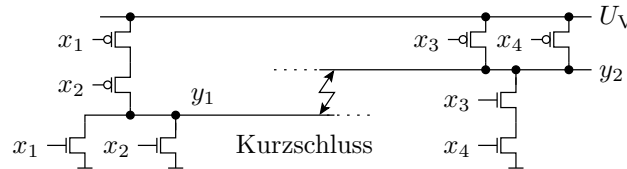
Ähnlich nachweisbare Haftfehler:

- $sa0(x_5) : a \wedge x_5 \wedge B(y : 1 \rightarrow 0)$
- $sa1(x_5) : a \wedge \bar{x}_5 \wedge B(y : 0 \rightarrow 1);$

(a – gemeinsame Anregungsbedingung; $B(y : 0 \rightarrow 1)$, $B(y : 1 \rightarrow 0)$ – Veränderung $0 \rightarrow 1$ bzw. $0 \rightarrow 1$ von y beobachtbar; $?L$ – Fehler möglicherweise logisch nachweisbar; $?I$ – Fehler möglicherweise über Ruhestromüberwachung nachweisbar).



Kurzschluss zweier Gatterausgänge



Nachweis über statische Stromaufnahme: $y_1 \neq y_2$

Bedingungen für einen möglichen logischen Nachweis:

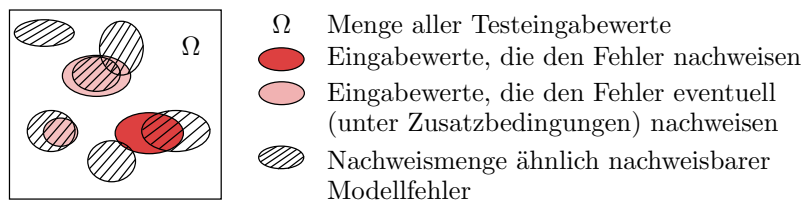
- $\bar{x}_1 \wedge \bar{x}_2 \wedge x_3 \wedge x_4 \wedge (B(y_1 : 1 \rightarrow 0) \vee B(y_2 : 0 \rightarrow 1))$
- $x_1 \vee x_2 \wedge \bar{x}_3 \vee \bar{x}_4 \wedge (B(y_1 : 0 \rightarrow 1) \vee B(y_2 : 1 \rightarrow 0))$

Ähnlich nachweisbare Haftfehler:

- $sa0(x_1)$, $sa0(x_2)$ wenn $y_1 \wedge (B(y_1 : 1 \rightarrow 0) \vee B(y_2 : 0 \rightarrow 1))$
- $sa1(x_1)$, $sa1(x_2)$ wenn $\bar{y}_1 \wedge (B(y_1 : 0 \rightarrow 1) \vee B(y_2 : 1 \rightarrow 0))$
- $sa0(x_3)$, $sa0(x_4)$ wenn $y_2 \wedge (B(y_1 : 0 \rightarrow 1) \vee B(y_2 : 1 \rightarrow 0))$
- $sa1(x_3)$, $sa3(x_4)$ wenn $\bar{y}_2 \wedge (B(y_1 : 1 \rightarrow 0) \vee B(y_2 : 0 \rightarrow 1))$

2.3 FC fehlerorientierte Suche

Fehler und 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 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:

$$p_i = 1 - \prod_{\forall j | MF_j \sim F_i} (1 - p_{ij})^{a_j}$$

($\forall j | MF_j \sim F_i$ – alle j für die Modellfehler j ähnlich wie Fehler i nachweisbar ist). Fehlerüberdeckung

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

Zwei Modellrechnungen mit $p_{ij} = 50\%$, Verteilung $\mathbb{P}[k = n_i \cdot a_j]$ (n_i – Anzahl ähnlich nachweisbare Modellfehler für Fehler i ; a_j – Anzahl der Tests je Haftfehler j): ...:

$k = n_i \cdot a_j$	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%

$$FC_M = 1 - \mathbb{P}[k = 0]; \quad FC = 1 - \sum_{k=1}^{k_{\max}} p_k \cdot 0,5^k$$

A:	$FC_M = 99\%$	$FC = 90\%$
B:	$FC_M = 97\%$	$FC = 95\%$

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

2.4 FC Zufallstest

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:

$$h(\zeta) \approx h(c \cdot \zeta)$$

Wenn die ähnlich nachweisbaren Modellfehler tendentiell mehr FF verursachen als die lokalen 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:

$$FC(n) \approx FC_M(c \cdot n)$$

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

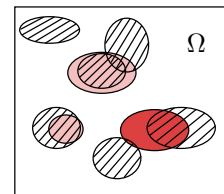
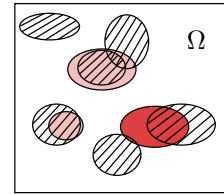
Wiederholung: Schätzung Erwartungswert

Nach Foliensatz 3, Abschnitt 2.5 »Bereichsschätzung Normalverteilung« ist der aus einem Zählwert x_{ist} abschätzbare Bereich für den Erwartungswert:

$$\mathbb{E}[X] = x_{\text{ist}} \mp \sqrt{\kappa \cdot x_{\text{ist}}} \cdot \Phi^{-1}\left(1 - \frac{\alpha}{2}\right)$$

(α – Irrtumswahrscheinlichkeit; Φ^{-1} – inverse standardisierten Normalverteilung; κ – Varianzerhöhung durch Anhängigkeiten).

α	4,54%	0,26%	0	4%	2%	1%	0,4%	0,2%
$\Phi^{-1}\left(1 - \frac{\alpha}{2}\right)$	2	3	4	2,05	2,33	2,57	2,88	3,10



Obergrenze relativer Schätzfehler:

$$\varepsilon = \max \left(\frac{|\mathbb{E}[X] - x_{\text{ist}}|}{x_{\text{ist}}} \right) = \frac{\text{sd}[X]}{\mathbb{E}[X]} \cdot \Phi^{-1} \left(1 - \frac{\alpha}{2} \right) \approx \dots$$

$$\varepsilon \approx \sqrt{\frac{\kappa}{x_{\text{ist}}}} \cdot \Phi^{-1} \left(1 - \frac{\alpha}{2} \right)$$

Effektiver Mindestzählwert für eine Erwartungswertschätzung mit der relativen Genauigkeit ε und Irrtumswahrscheinlichkeit α :

$$x_{\text{eff}} = \frac{x_{\text{ist}}}{\kappa} \geq \left(\frac{\Phi^{-1} \left(1 - \frac{\alpha}{2} \right)}{\varepsilon} \right)^2$$

mit x_{ist} gleich »Anzahl der nicht nachweisbaren Modellfehler« erfordert die Abschätzung einer zu erwartenden Modellfehlerüberdeckung FC_M eine Gesamtmodellfehleranzahl von:

$$\#F_M = \kappa \cdot \frac{x_{\text{eff}}}{1 - FC_M} \geq \kappa \cdot \left(\frac{\Phi^{-1} \left(1 - \frac{\alpha}{2} \right)}{\varepsilon} \right)^2$$

Beispiel 1. Relative Genauigkeit $\varepsilon = 10\%$, zulässige Irrtumswahrscheinlichkeit $\alpha = 2\%$, angestrebte Modellfehlerüberdeckung $FC_M = 99\%$, Varianzerhöhung durch Anhängigkeiten $\kappa = 5$.

- Erforderlicher effektiver Mindestzählwert:

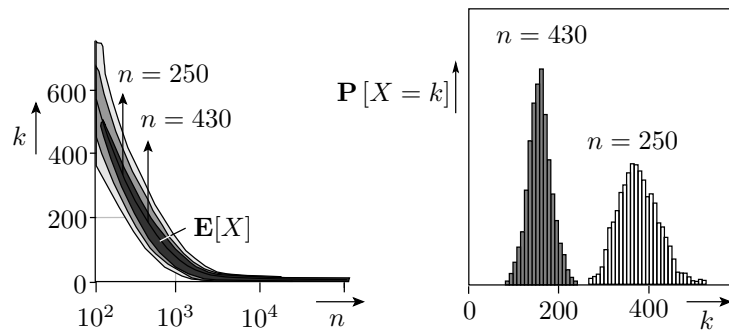
$$x_{\text{eff}} \geq \left(\frac{\Phi^{-1} \left(1 - \frac{\alpha}{2} \right)}{\varepsilon} \right)^2 = 497$$

- Gesamtmodellfehleranzahl:

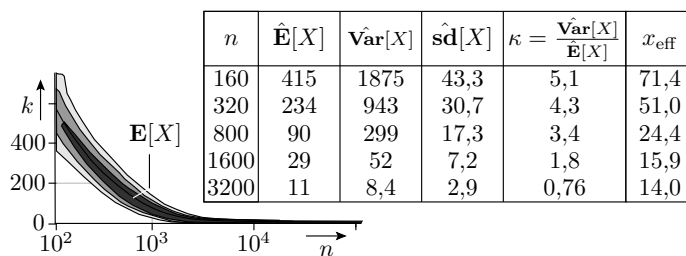
$$\#F_M \geq \kappa \cdot \frac{x_{\text{eff}}}{1 - FC_M} = 248.500$$

Experiment mit Haftfehlern

Kombinatorische Beispielschaltung (Benchmark c3540). 3606 simulierte, unterschiedlich nachweisbare Haftfehler. Zählwert X ist die Anzahl der nicht nachweisbaren Haftfehler. Abschätzung von $\mathbb{P}[X = k]$ aus einer Stichprobe von $\#w = 1000$ Zählwerten für verschiedenen Zufallstestsätze der Länge n .



Varianzerhöhung im Experiment



Die effektiven Zählwerte $x_{\text{eff}} \approx 15 \dots 50$ sind über eine Zehnerpotenz kleiner als in der Beispielrechnung zuvor. Da sich in die Schaltung keine weiteren unabhängig nachweisbaren Modellfehler einbauen lassen, gibt es zur Erhöhung der Schätzgenauigkeit für FC_M nur wie im Experiment den Weg, die Anzahl der nicht nachweisbaren Fehler für unterschiedliche Zufallstestsätze zu bestimmen und zu mitteln.

Effektive Zählwerte für Modellfehlerstichprobe

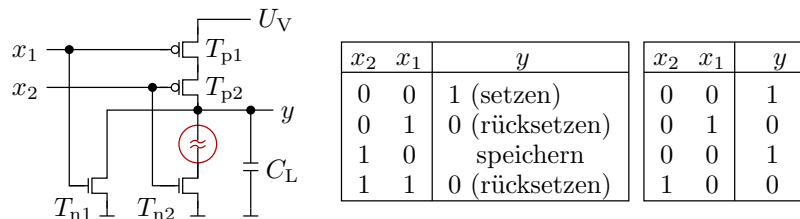
n	1000 Modellfehler			x_{eff}	300 Modellfehler			x_{eff}
	$\hat{E}[X]$	$\hat{sd}[X]$	κ		$\hat{E}[X]$	$\hat{sd}[X]$	κ	
160	118	14,1	1,91	61,6	31,2	5,61	1,13	27,6
320	68	10,4	1,71	39,8	16,2	4,26	1,18	13,7
800	25	6,3	1,63	15,4	4,8	2,28	1,10	4,4
1600	8	2,8	0,99	8,1	0,9	1,08	1,30	0,7

Das Verhältnis aus der Anzahl der simulierten Fehler und den effektiven Zählwerten verbessert sich bei Beschränkung auf eine Modellfehlerstichprobe.

Im Vergleich zur gezielten Suche, bei der für jeden zu erwartenden Fehler ähnlich nachweisbare Fehler benötigt werden, genügt für die Bewertung von Zufallstests eine ausreichend große Fehlerstichprobe.

2.5 Verzögerungsfehler

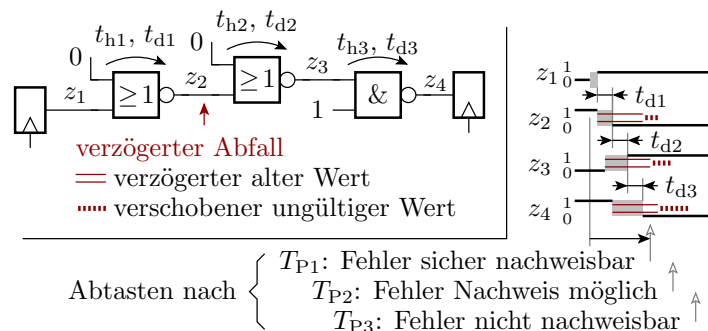
Verzögerungsfehler



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.

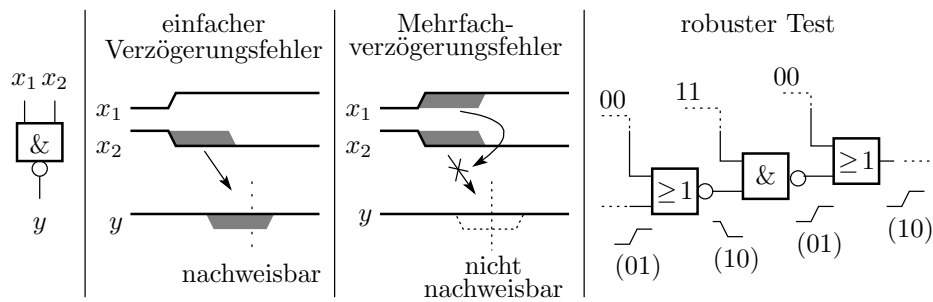
(Robuster) Zwei-Pattern-Test

Der Fehlernachweis erfordert zwei Eingaben:

- Einstellung 0 für \uparrow und 1 für \downarrow am Fehlerort.

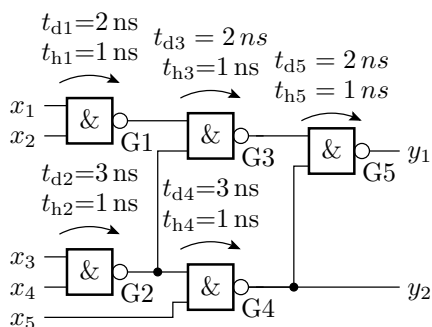
- Haftfehlertest sa0 für \uparrow , sa1 für \downarrow am Fehlerort.

(\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

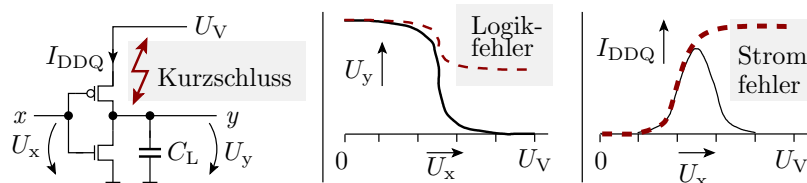


Pfade	$\sum t_{h,i}$	$\sum t_{d,i}$
G1-G3-G5	3 ns	6 ns
G2-G3-G5	3 ns	7 ns
G2-G4-G5	3 ns	8 ns
G2-G4	2 ns	6 ns
G4-G5	2 ns	5 ns
G4	1 ns	3 ns

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

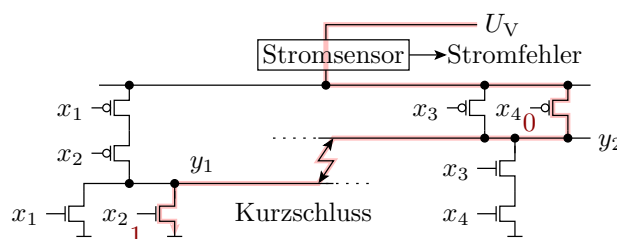
2.6 IDDQ-Test

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 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 Zufallstest bei gleicher Fehlerüberdeckung.
- Probleme I_{DDQ} -Test: Unterscheidung von zulässigem und überhöhtem Ruhestrom funktioniert nur bis einige tausend Gatter. Integrierte Stromsensoren, ...

3 Testberechnung

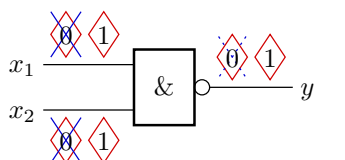
3.1 Fehlersimulation

Testsuche durch Fehlersimulation

Wiederhole, bis genügend Modellfehler nachgewiesen sind:

- Geziele, manuelle oder zufällige Auswahl weiterer Testbeispiele
- Fehlersimulation und Anhängen der nachweisbaren Modellfehler

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



0 sa0-Modellfehler
1 sa1-Modellfehler
X identisch nachweisbar
X implizit nachweisbar

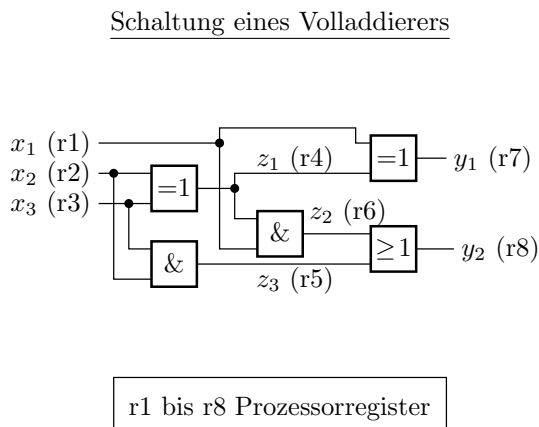
x_2	x_1	$\overline{x_2 \wedge x_1}$	sa0(x_1)	sa1(x_1)	sa0(x_2)	sa1(x_2)	sa0(y)	sa1(y)
0	0	1	1	1	1	1	0	1
0	1	1	1	1	1	0	0	1
1	0	1	1	0	1	1	0	1
1	1	0	1	0	1	0	0	1

Nachweisidentität (gleiche Nachweismenge)

-----> Nachweisimplikation

■ zugehörige Eingabe ist Element der Nachweismenge

Haftfehler sind einfach zu simulieren



Programm für die Gutsimulation

```

lade x1 in Register r1
lade x2 in Register r2
lade x3 in Register r3
r4 = r2 xor r3
speichere Inhalt r4 in z1
r5 = r2 and r3
speichere Inhalt r5 in z3
r6 = r1 and r4
speichere Inhalt r6 in z2
r7 = r1 xor r4
speichere Inhalt r7 in y1
r8 = r5 or r6
speichere Inhalt r8 in y2
    
```

- 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.

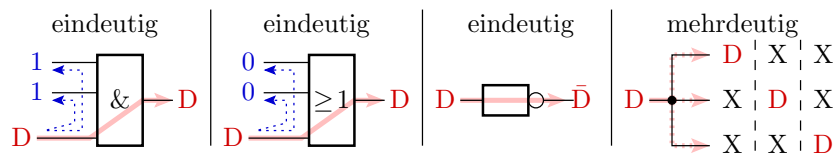
3.2 D-Algorithmus

D (Discrepancy)-Kalkül von Roth

Erweiterung der Logikwerte um 3 Pseudo-Werte³:

- 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:



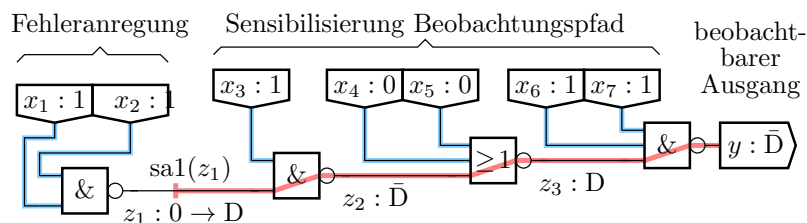
Testsuche für Haftfehler

Ein Haftfehler unterstellt für den Fehlerort, dass der Wert

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

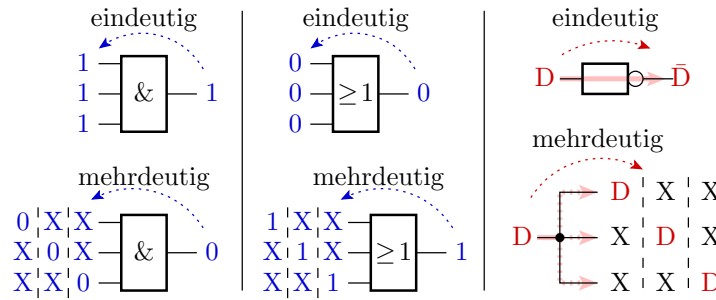
Ausgehend vom Fehlerort werden Eingaben gesucht,

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



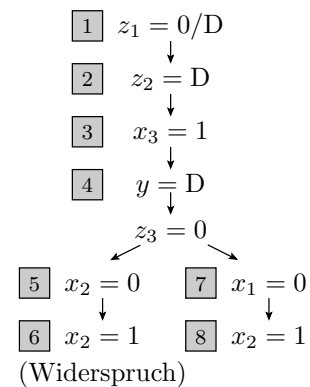
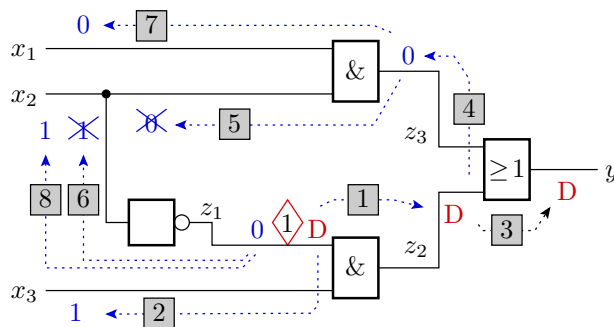
³W. Daehn: Testverfahren in der Mikroelektronik: Methoden und Werkzeuge. Springer 1997.

Ein- und mehrdeutige Pfade



Ausgehend vom Fehlerort:

- Festlegen von Werten zur Weiterführung des Beobachtungs- oder eines Steuerpfads.
- Bei Widersprüchen zurück zu letzten Möglichkeit einer Alternativenentscheidung, ... => Baumsuche



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
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	X	0	D	0D	D
7	1	X	0	0	D	0D	D
8	1	1	0	0	D	0D	D

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 $> 2^{30...40}$ nicht mehr vollständig durchsuchbar.
- Abbruch der Suche nach einer bestimmten Rechenzeit.

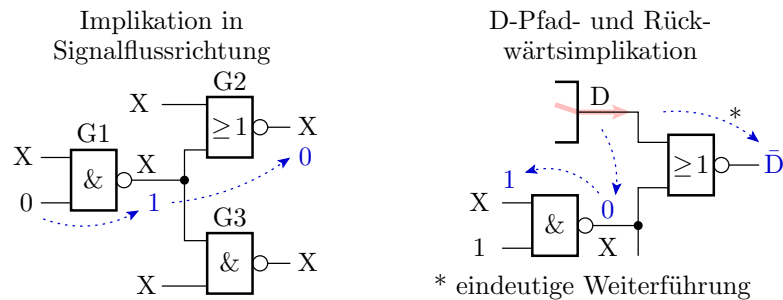
Heuristiken:

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

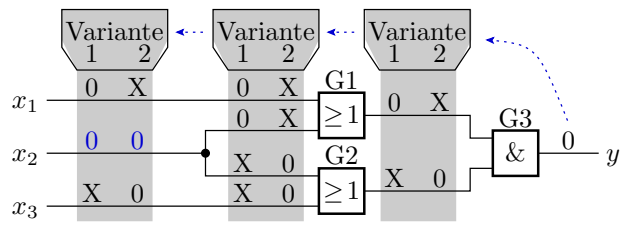
3.3 Implikationstest

Implikationstest (Widerspruchsfrüherkennung)

- Aus den berechneten Wertefestlegungen alle eindeutig folgenden Werte berechnen.



- Mindert die Entscheidungsbaumtiefe.
- Rückwärtsimplikation über mehrere Gatterebenen:

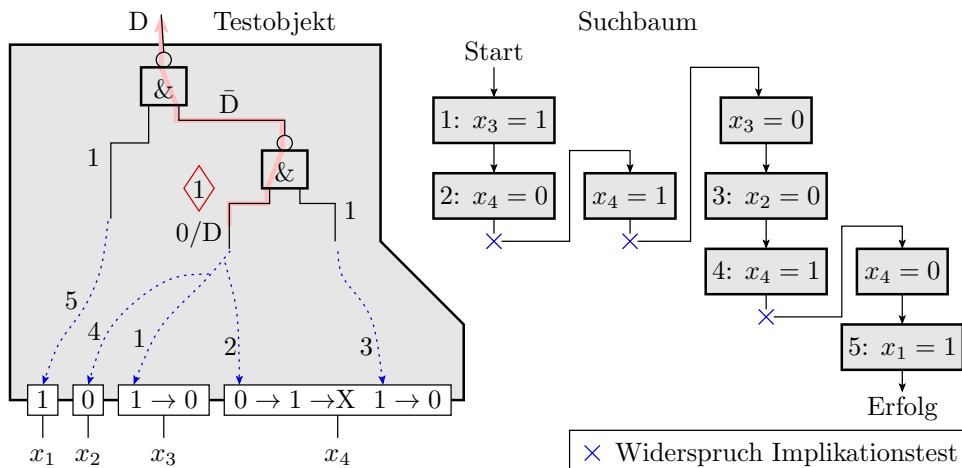


- Für $y = 0$ gibt es zwei Einstellmöglichkeiten.
- Für beide Möglichkeiten muss $x_2 = 0$ sein.
- Das Erkennen von Implikationen dieser Art mindert die Backtracking-Häufigkeit um bis zu 80%.

3.4 Suchraumstrukturierung

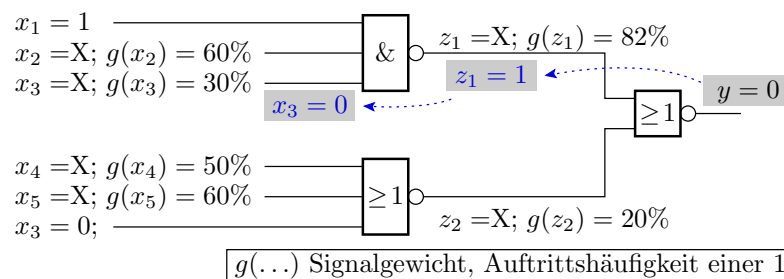
Suchraumbegrenzung

- 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 Signalwichtungen⁴ über eine kurze Simulation mit Zufallswerten oder analytisch.
- Wahl der Steuerwerte / Beobachtungspfade, die mit größerer Wahrscheinlichkeit aktivierbar / sensibler sind.

3.5 Komplexe Funktionsbausteine

Komplexe Funktionsbausteine

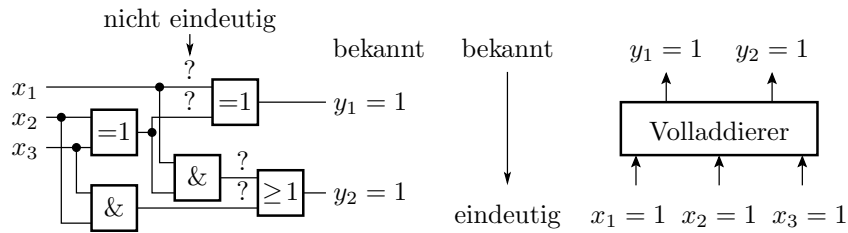
- Beschreibung durch Tabellenfunktion (Bsp. Volladdierer):

x_2	x_1	x_0	s	c	gegeben	Lösungsmenge
0	0	0	0	0	XXX00	\Rightarrow 00000
0	0	1	1	0	01DXX	\Rightarrow 01D \bar{D}
0	1	0	1	0		
0	1	1	0	1		
1	0	0	1	0	1XXXD	\Rightarrow 10D \bar{D} , 1D0 \bar{D}
1	0	1	0	1		
1	1	0	0	1	11XX1	\Rightarrow 11111, 11101
1	1	1	1	1		

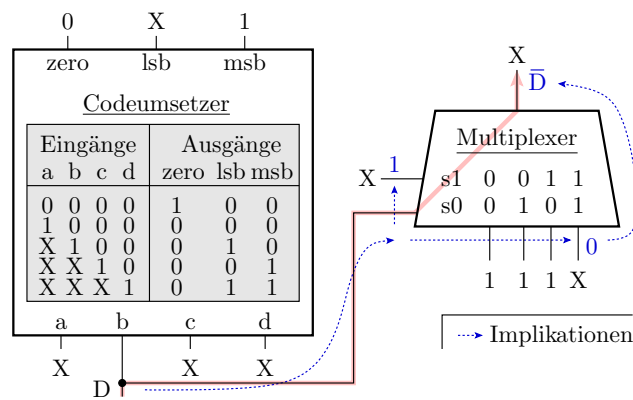
- Vervollständigung des Vektors der gegebenen Anschlusswerte durch Vergleich mit allen Tabellenzeilen:
 - »1« und »0« passen nur auf »1« und »0«.
 - »X« passt immer.
 - »D« muss für »D=0« und für »D=1« passen.

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

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 Anschlusswertvervollständigung erkannt.

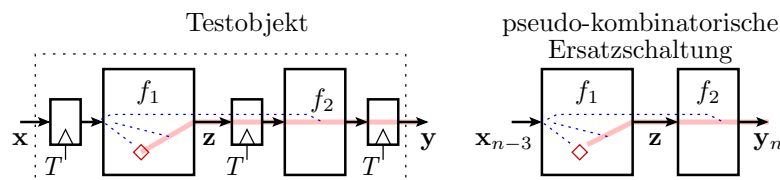


- »lsb« hängt bei »zero=0« und »msb=1« nicht von »b« ab. Eindeutiger D-Pfad über Multiplexer.
- Tabelleneingabewerte »X« (Eingang beeinflusst nicht die Ausgabe) führt zu Tabellen mit $\ll 2^{N_E}$ Tabellenzeilen (N_E – Anzahl der Eingänge).

3.6 Sequentielle Schaltungen

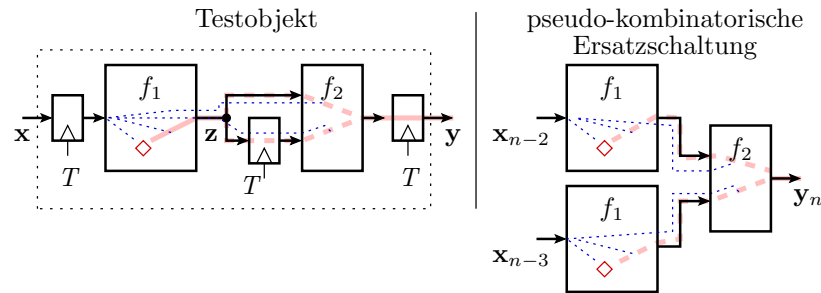
Pfadsensibilisierung für Systeme mit Gedächtnis

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



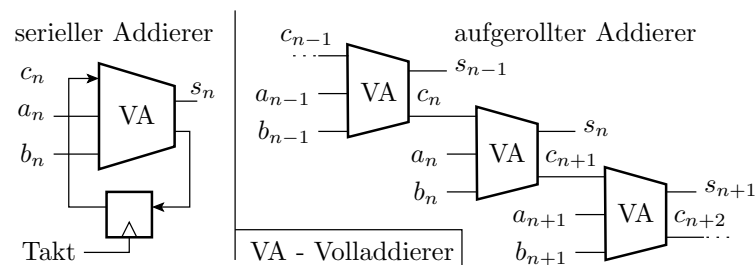
- Testberechnung wie für eine kombinatorische Schaltung.
- Die Verzögerung der Ausgabe gegenüber der Eingabe wird erst bei der Testdurchführung berücksichtigt.

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. Seite 4).

4 Selbsttest

Selbsttest

Einbau der Testfunktionseinheiten mit in den Schaltkreis:

- Testmustergenerator: Pseudo-Zufallsgenerator, Zähler, Schieberegister.
- Testablaufsteuerung, in der Regel über Testbus (vergl. Abschn. 1.1.6).
- Ausgabekontrolle, vorzugsweise durch Bildung eines Prüfkennzeichens mit LFSR (Linear Feedback Shift Register).

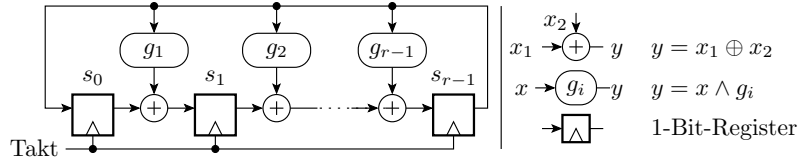
Vorteile:

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

4.1 Pseudo-Zufallsregister

Linear rückgekoppelte Schieberegister

Ein linear rückgekoppelte Schieberegister (LFSR Linear Feedback Shift Register) 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 $\mathbf{g} = (g_{r-1}, g_{r-2}, \dots, g_1, 1)$ zum Zustand \mathbf{s} :



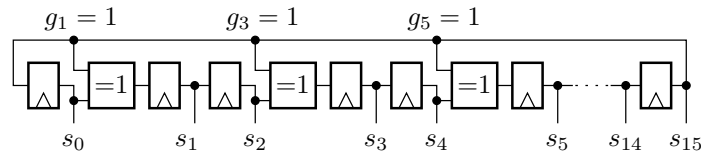
Für jede Bitanzahl r des Zustandsvektors gibt es Konstanten \mathbf{g} , sog. »primitive Polynome«, bei denen alle Zustände außer 000...0 ineinander übergehen. Nur solche Konstanten \mathbf{g} werden verwendet.

Primitiven Polynome und die Konstante \mathbf{g}

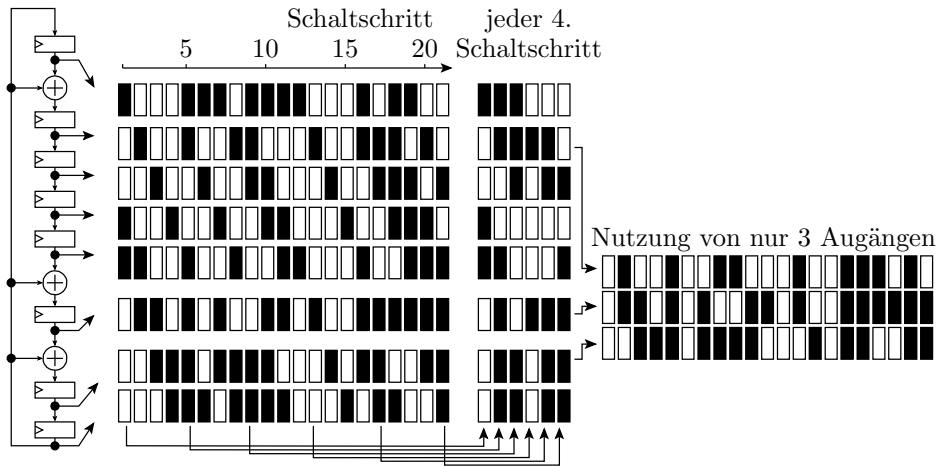
Mit dem Internet-Suchbegriff »Primitive Polynome« findet man z.B. für 16-Bit LFSR:

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

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.

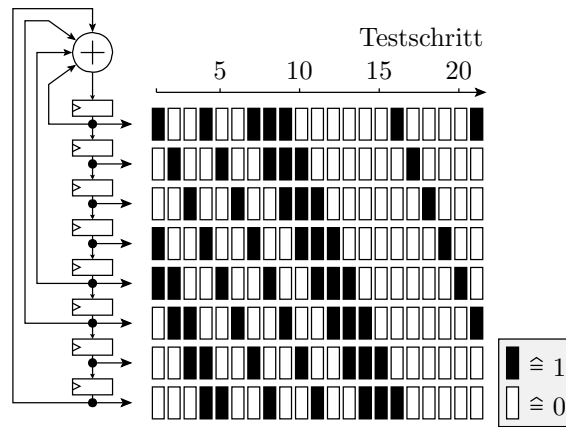


Pseudo-Zufallsfolge eines 8-Bit-LFSR



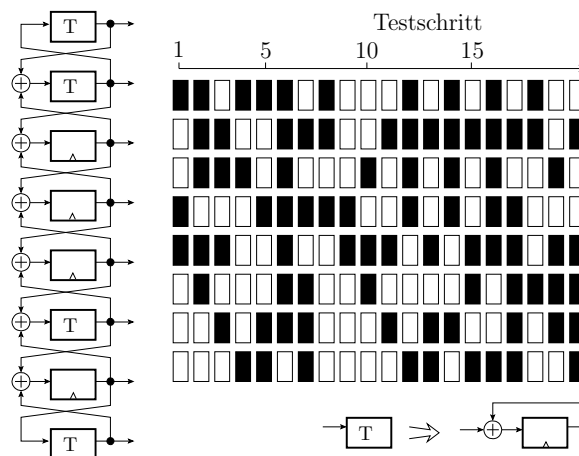
Falls die »Streifenmuster« durch die Schiebeoperationen stören, nur einen Teil der Ausgaben nutzen.

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.

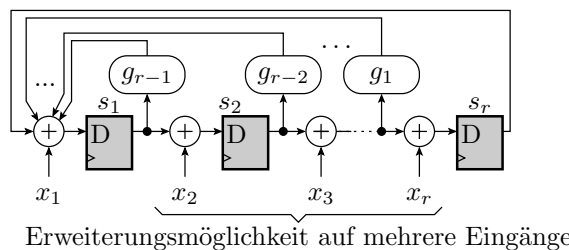
Es gibt viele weitere lineare Automaten, die auch zyklisch Bitfolgen in zufälliger Reihenfolge erzeugen. Beispiel Zellenautomaten, bei denen jedes Folgebitt aus dem eigenen und den Zuständen der Nachbarbits gebildet wird:



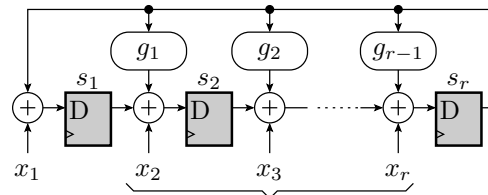
4.2 Signaturregister

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 Zyklusstruktur⁵. 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.

Experiment Fehlererkennungssicherheit von LFSR

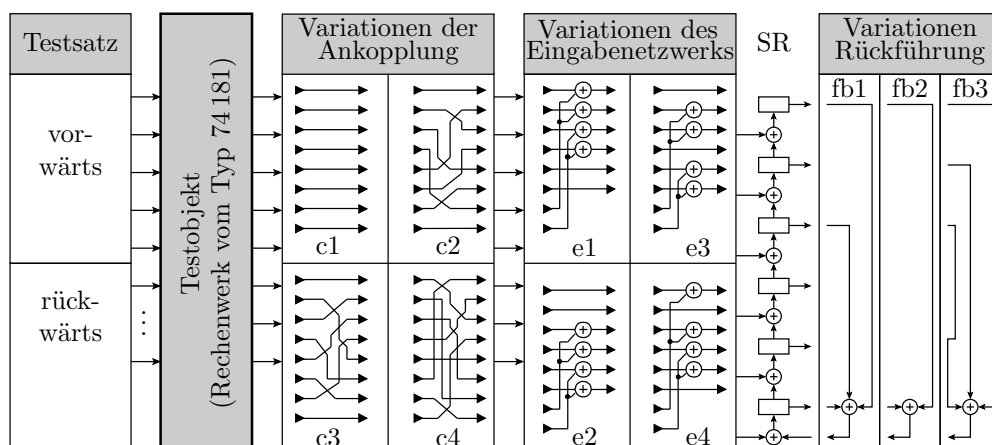
Es ist schwer zu glauben, dass

- mit r -Bit Prüfkennzeichen beliebige Verfälschung mit einer Wahrscheinlichkeit $p_E = 1 - 2^{-r}$ 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 $p_E = 1 - 2^{-6} = 98,44\%$ erkenn- und mit einer Wahrscheinlichkeit $p_F = 2^{-6} = 1,36\%$ nicht erkennbar sein müsste. Definition einer Zufallsgröße X_i zum Zählen der nicht erkennbaren Fehler:

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

⁵Zyklusstruktur ohne Eingaben.

Wenn die Theorie stimmt, müsste die Anzahl der maskierten Fehler

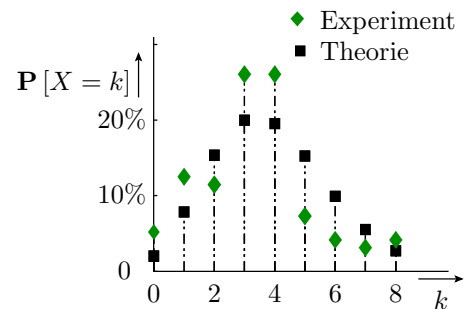
$$X = \sum_{i=1}^{250} X_i$$

binomialverteilt mit $p_F = 2^{-6}$ sein (siehe Foliensatz 3):

$$\mathbb{P}[X = k] = \binom{N}{k} \cdot p_F^k \cdot (1 - p_F)^{N-k}$$

Anzahl der maskierten Fehler

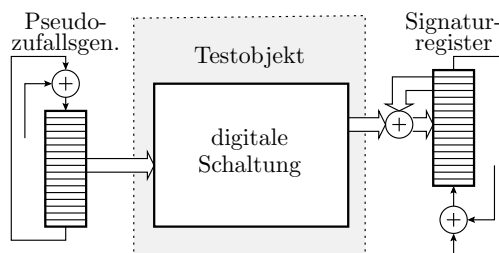
		e1				e2				e3				e4			
		c1	c2	c3	c4	c1	c2	c3	c4	c1	c2	c3	c4	c1	c2	c3	c4
vorwärts	fb1	3	4	1	2	3	4	3	3	4	2	4	3	4	3	4	6
	fb2	3	4	1	7	2	2	1	4	2	1	1	3	2	5	3	7
	fb3	5	2	2	8	4	5	3	4	3	6	3	7	5	3	3	4
rückwärts	fb1	6	4	4	2	3	4	3	4	3	4	3	4	4	8	4	5
	fb2	2	0	0	1	4	1	4	1	0	0	0	1	1	1	4	1
	fb3	2	4	3	4	4	8	5	8	3	3	3	6	3	3	4	3



Abweichungen zwischen Vorhersage und Experiment nicht signifikant. Hypothese $p_F = 2^{-6}$ bestätigt.

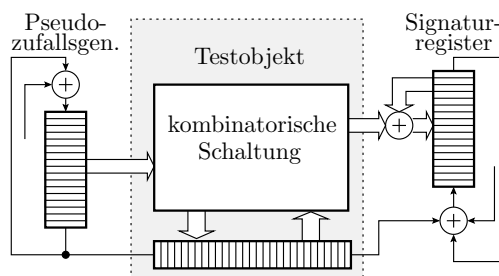
4.3 Selbsttest mit LFSR

Selbsttest (BIST Built-in Self Test)



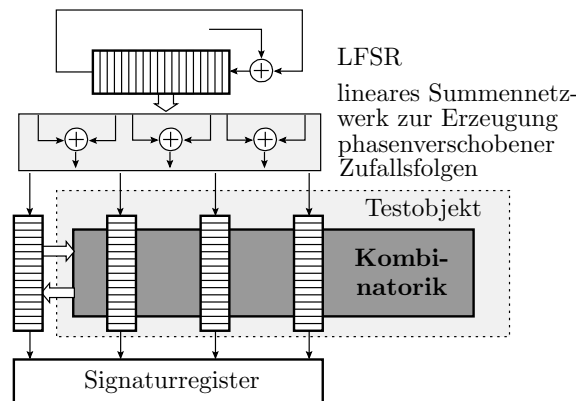
- Einrahmen der Schaltung mit Schieberegistern und Ergänzung einiger EXOR-Gatter.
- Wenn als Schieberegister vorhandene Ein- und Ausgaberegister verwendet werden, besonders niedriger Zusatzaufwand.
- Test mit voller Schaltungsgeschwindigkeit von Millionen bis Milliarden Test pro Sekunde.

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.]

4.4 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 Auftretswahrscheinlichkeit 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

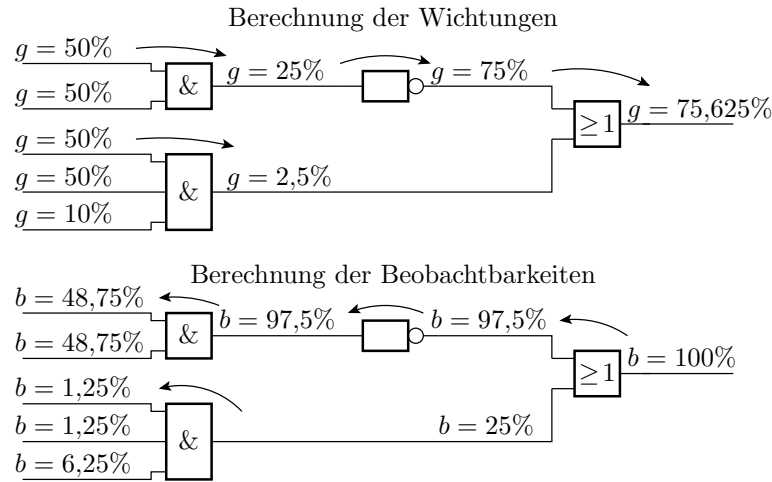
Funktionen der Wichtungen an den Eingängen und damit über Eingabewichtungen einstellbar.

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. ...

$\begin{array}{l} x_1 \\ x_2 \end{array} \begin{array}{l} \frac{g(x_1), b(x_1)}{g(x_2), b(x_2)} \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array} \boxed{\&} \begin{array}{l} \text{---} \\ \text{---} \end{array} \begin{array}{l} g(y), b(y) \\ y \end{array}$	$\begin{array}{l} b(x_2) = b(y) \cdot g(x_1) \\ b(x_1) = b(y) \cdot g(x_2) \\ g(y) = g(x_1) \cdot g(x_2) \end{array}$
$\begin{array}{l} x_1 \\ x_2 \end{array} \begin{array}{l} \frac{g(x_1), b(x_1)}{g(x_2), b(x_2)} \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array} \boxed{\geq 1} \begin{array}{l} \text{---} \\ \text{---} \end{array} \begin{array}{l} g(y), b(y) \\ y \end{array}$	$\begin{array}{l} b(x_1) = b(y) \cdot (1 - g(x_2)) \\ b(x_2) = b(y) \cdot (1 - g(x_1)) \\ g(y) = 1 - (1 - g(x_1)) \cdot (1 - g(x_2)) \end{array}$
$x \text{---} \frac{g(x), b(x)}{\quad} \begin{array}{c} \text{---} \\ \text{---} \end{array} \boxed{\quad} \begin{array}{l} \text{---} \\ \text{---} \end{array} \begin{array}{l} g(y), b(y) \\ y \end{array}$	$\begin{array}{l} b(x) = b(y) \\ g(y) = (1 - g(x)) \end{array}$

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

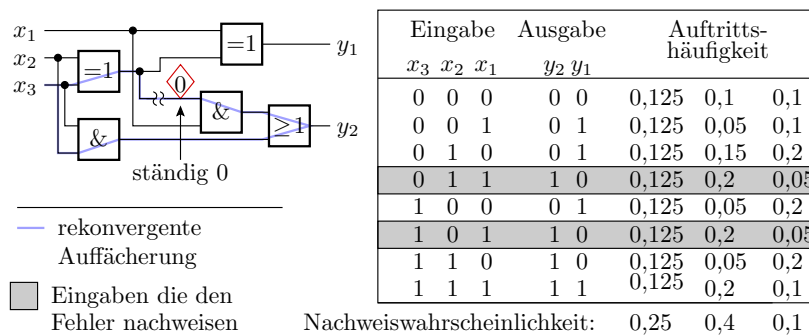


Die Anregungswahrscheinlichkeit eines sa0-Fehlers ist die Wichtung $g(\dots)$ und die eines sa1-Fehler die Gegenwahrscheinlichkeit $1 - g(\dots)$ am Fehlerort.

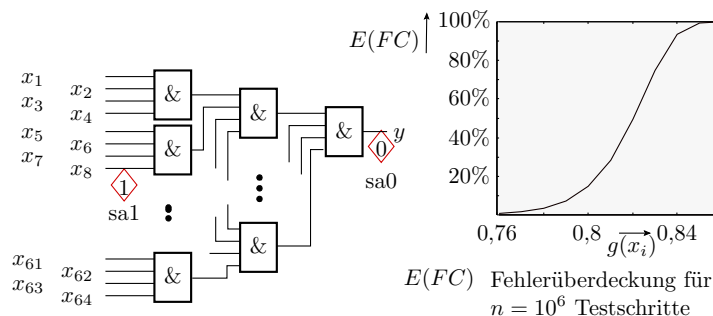
Die Nachweiswahrscheinlichkeit ist das Produkt aus Anregungs- und Beobachtungswahrscheinlichkeit.

Rekonvergente Auffächerung

Bei rekonvergenter Auffächerung werden gleiche Zufallswerte nach unterschiedlicher Verknüpfung mit anderen Werten logisch verknüpft. Für verknüpfte Werte, die von derselben Zufallsgröße abhängen, gelten die einfachen Regeln auf Seite 33 nicht. Berechnung aus den Auftrittshäufigkeiten der Eingaben für den Fehlernachweis:



Fehlerüberdeckung und Wichtung

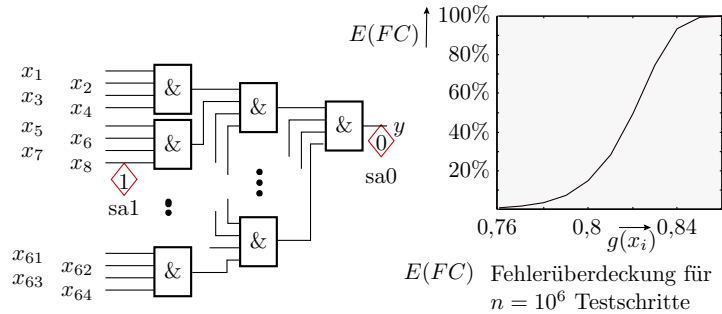


Angenommene Fehler: Für je einen der 64 Eingänge ständig 1:

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

Für den Ausgang ständig 0:

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



Zu erwartende Fehlerüberdeckung als Mittelwert der Nachweiswahrscheinlichkeit:

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

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 aus⁶:

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.

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:

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

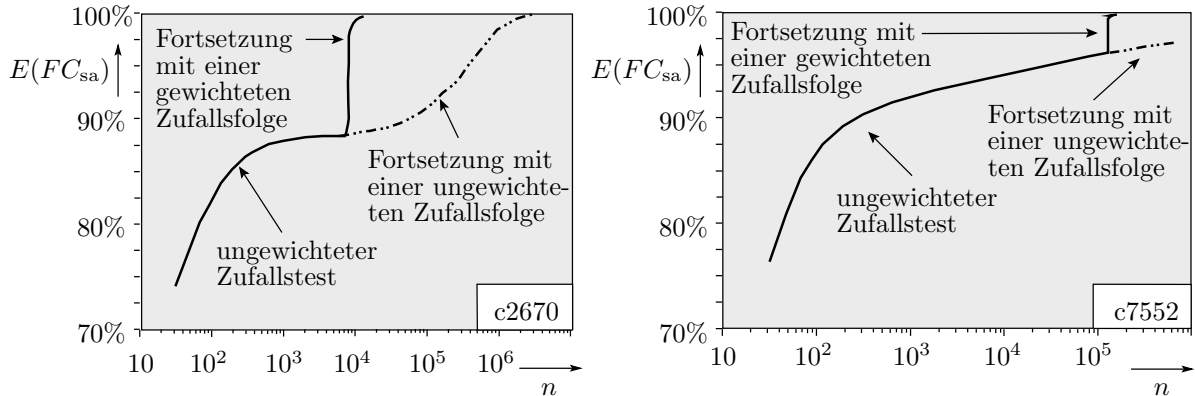
($\#N$ – Anzahl der Nullen; $\#E$ – Anzahl der Einsen für Eingang x_i in den Testeingaben; $\#E_{AND} \in \{2, 3, 4, \dots\}$ Anzahl der $[N]$ AND-verknüpften Zufallsfolgen zur Wichtungserzeugung).

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

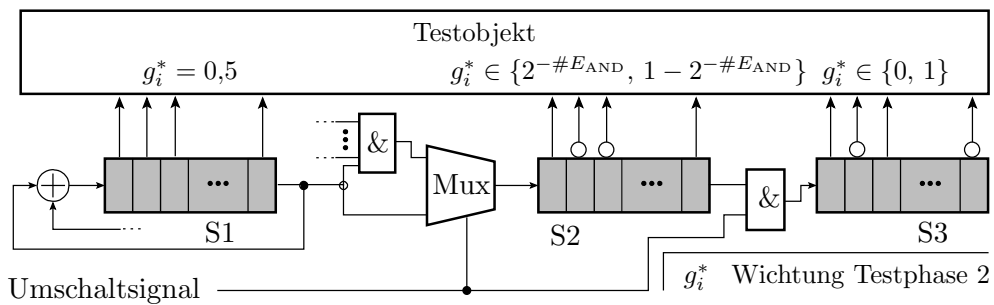
⁶J. Hartmann, G. Kemnitz: How to do weighted random testing for BIST? ICCAD 1993.

Experiment mit den Schaltungen c2670 und c7552⁷

- 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.



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

Nicht nennenswert aufwändiger als ohne Wichtung.

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