Test und Verlässlichkeit
Grosse Übung zu Foliensatz 5

Prof. G. Kemnitz

Institut für Informatik, TU Clausthal (TV_GUeF5)

24. Juni 2021

Schaltungsstrukt. & Test

Statische Tests

Aufgabe 5.1: MDA und ICT

1. Wodurch unterscheidet sich der analoge In-Circuit-Test von einer Zweipunktmessung zur Kontrolle auf Fertigungsfehler?
2. Ersetzt ein digitaler In-Circuit-Test Zweipunktmessungen zur Kontrolle auf Fertigungsfehler vollständig?
3. Was bedeutet bei Boundry-Scan »Ersatz der Nadelbettadapters durch Silicon Nails«?

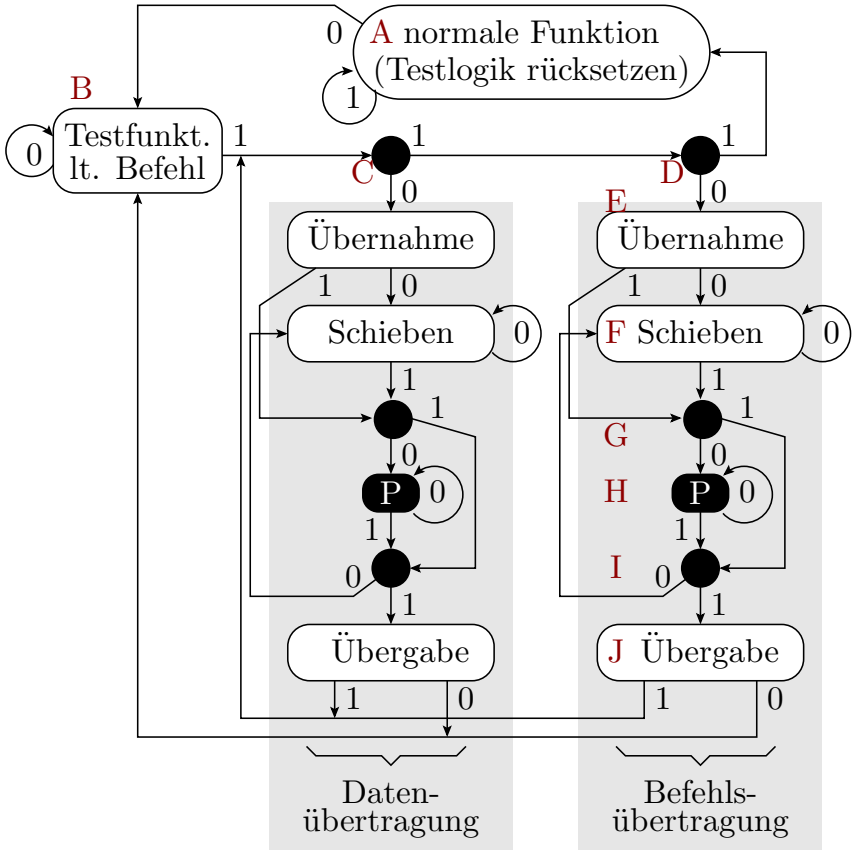
Zur Kontrolle

1. Die Strom-Spannungs-Beziehung einer Zweipunktmessung zur Kontrolle auf Fertigungsfehler hängt außer vom Bauteil zwischen den Punkten auch von der umgebenden Schaltung ab. Beim analogen In-Circuit-Test werden die wegfließenden Ströme von einem der Punkte unterdrückt und so die Abhängigkeit der gemessenen Strom-Spannungs-Beziehung von anderen Bauteilen unterbunden. Vereinfacht die Testerstellung.
2. Kein vollständiger Ersatz. Digitaler ICT ist ein Test unter Spannung. Mindestens zur Kontrolle und Beseitigung von Kurzschlüssen vor Anlegen der Spannung sind Zweipunktmessung zwischen den Leitungen erforderlich.
3. Bei Boundary-Scan erfolgt der Lese- und Schreibzugriff der logischen Pegel der Leitungen auf einer Baugruppe mit integrierten Teststrukturen (Schieberegisterringen) statt der Kontaktierung mit einem Nadeladapter.

Testbus (JTAG)

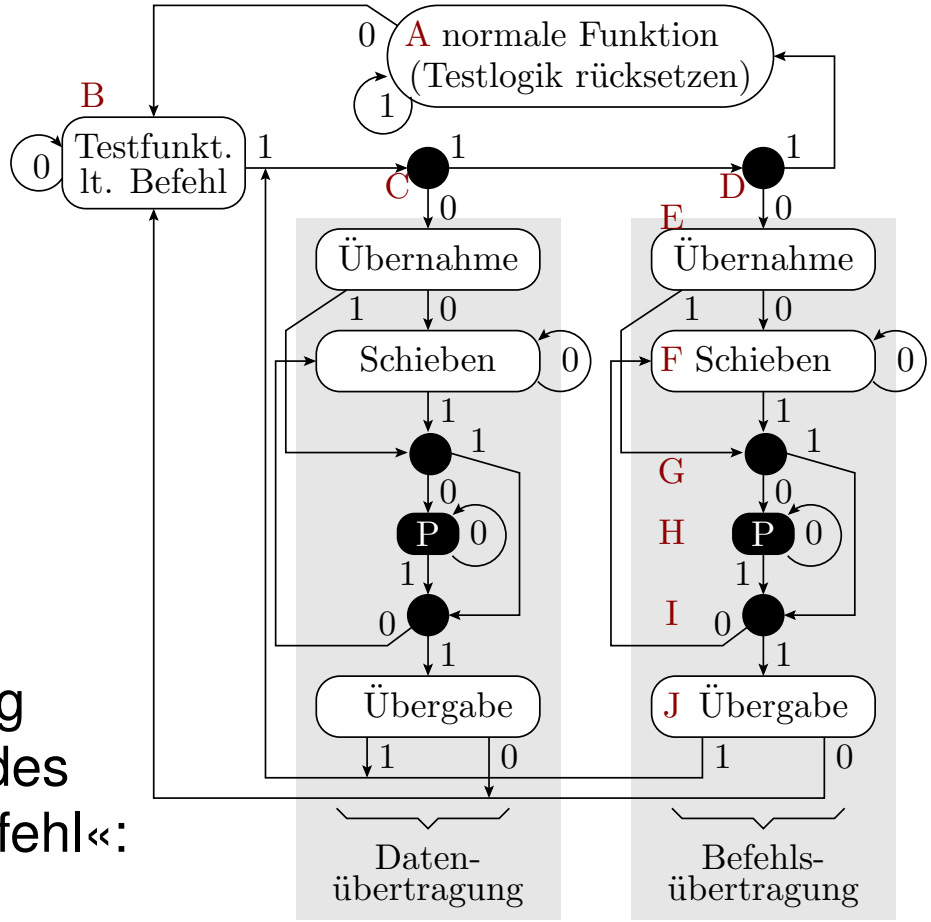
Aufgabe 5.2: Boundary-Scan

1. Mit welcher Bitfolge an TMS lässt sich der TAP-Controller aus einem beliebigen Zustand in den Zustand »normale Funktion« versetzen?
2. Mit welcher Zustands- und Bitfolge an TMS und TDI lässt sich ausgehend vom Zustand »normale Funktion« das 8-Bit-Befehlswort 0x4D und anschließend der Zustand »Testfunktion laut Befehl« einstellen?



Zur Kontrolle

1. TMS-Folge, um den TAP-Controller aus einem beliebigen Zustand in den Zustand »normale Funktion« zu versetzen: 5-mal ’1’.
2. Zustands- und Bitfolge an TMS und TDI zur Einstellung des Befehlswort 0x4D und des Zustand »Testfunktion lt. Befehl«:

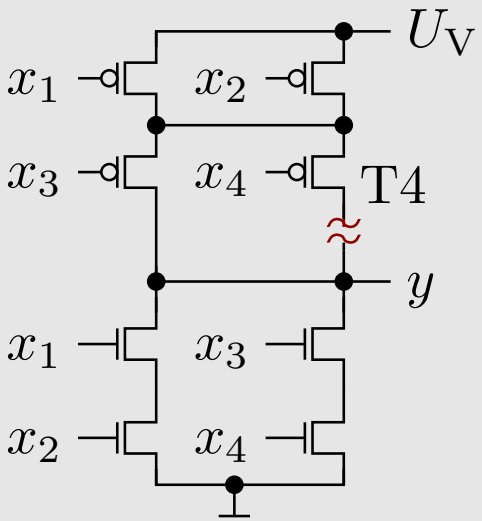


Takt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Zustand	A	B	C	D	E	F	F	F	F	F	F	F	F	G	I	J	B
TMS	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	
TDI	–	–	–	–	–	1	0	1	1	0	0	1	0	–	–	–	

Fehlermodellierung

Reale und Haftfehler

Aufgabe 5.3: Stuck-open-Fehler



x_4	x_3	x_2	x_1	A	B
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		

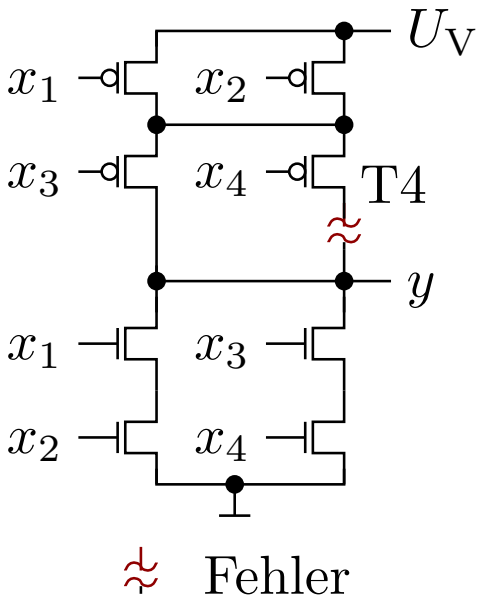
x_4	x_3	x_2	x_1	A	B
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

≈ sop-Fehler, Nachweis: Entladen von y und aufladen über T4

a) Kennzeichen Sie in Spalte A die Eingaben, mit denen y entladen und in Spalte B die, mit denen y über T4 aufgeladen wird.

b) Wie groß ist die Nachweiswahrscheinlichkeit des sop-Fehlers mit einer Folge von zwei zufälligen ungewichteten Eingabevektoren?

a) Kennzeichen Sie in Spalte A die Eingaben, mit denen y entladen und in Spalte B die, mit denen y über T4 aufgeladen wird.

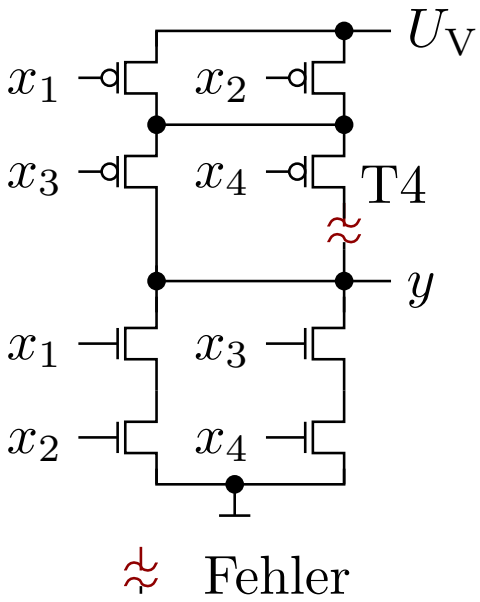


x_4	x_3	x_2	x_1	A	B
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1	x	
0	1	0	0		x
0	1	0	1		x
0	1	1	0		x
0	1	1	1	x	

x_4	x_3	x_2	x_1	A	B
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1	x	
1	1	0	0	x	
1	1	0	1	x	
1	1	1	0	x	
1	1	1	1	x	

- Wahrscheinlichkeit, dass der erste Eingabevektor y entlädt: $\frac{7}{16}$
- Wahrscheinlichkeit, dass der zweite Eingabevektor y über T4 auflädt: $\frac{3}{16}$
- Wahrscheinlichkeit sop-Fehlernachweis: $\frac{21}{256} = 8{,}2\%$

b) Wie groß ist die Nachweiswahrscheinlichkeit des sop-Fehlers mit einer Folge von zwei zufälligen ungewichteten Eingabevektoren?



x_4	x_3	x_2	x_1	A	B
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1	x	
0	1	0	0		x
0	1	0	1		x
0	1	1	0		x
0	1	1	1	x	

x_4	x_3	x_2	x_1	A	B
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1	x	
1	1	0	0	x	
1	1	0	1	x	
1	1	1	0	x	
1	1	1	1	x	

- Wahrscheinlichkeit, dass der erste Eingabevektor y entlädt: $\frac{7}{16}$
- Wahrscheinlichkeit, dass der zweite Eingabevektor y über T4 auflädt: $\frac{3}{16}$
- Wahrscheinlichkeit sop-Fehlernachweis: $\frac{21}{256} = 8{,}2\%$

FC und Modell-FC

Aufgabe 5.4: Testzeitskalierung

Die zu erwartende Fehlerüberdeckung eines Zufallstests stehe in folgendem Zusammenhang mit der Testsatzlänge n:

$$FC = 1 - \frac{\#F\left(n\right)}{\#F\left(n_0\right)} = 1 - \left(\frac{n}{n_0}\right)^{-0,4}$$

Wie groß ist die tatsächliche Fehlerüberdeckung bei einer Modellfehlerüberdeckung von $FC_{\mathrm{M}} = 90\%$, wenn die Nachweiswahrscheinlichkeiten je Testschritt für tatsächlicher Fehler tendentiell doppelt so groß ist wie für Modellfehler?

Zur Kontrolle

Aus dem gegebenen Zusammenhang zwischen Fehlerüberdeckung und Testsatzlänge und der gegebenen Modellfehlerüberdeckung ergibt sich für das Verhältnis aus Testsatzlänge und Bezugstestsatzlänge:

$$\frac{n}{n_0} = (1 - FC_{\mathrm{M}})^{-\frac{1}{0,4}} = 0,1^{-\frac{1}{0,4}} = 316$$

Für die realen Fehler mit tendentiell der doppelten Nachweiswahrscheinlichkeit beträgt die Fehlerüberdeckung:

$$FC = 1 - \left(2 \cdot \frac{n}{n_0}\right)^{-0,4} = 92,4\%$$

Testberechnung

Aufgabe 5.5: Vollständiger Test

Wie lange dauert ein vollständiger Test einer Funktion ohne Gedächtnis mit 32 Eingabebits und einer Funktionsausführungszeit von 1 ms, wenn die Funktion

1. genau einmal mit jeder Eingabemöglichkeit und
2. genau einmal mit jede Folge von zwei möglichen Eingaben[1]

getestet wird?

[1] Zur Kontrolle, dass die Funktion tatsächlich kein Gedächtnis hat.

Zur Kontrolle

1. Testzeit für den Test mit allen 2^{32} Eingabevarianten genau einmal:

$$t_{\text{Test}} = 2^{32} \cdot 1\,\text{ms} = 49{,}7\,\text{Tage}$$

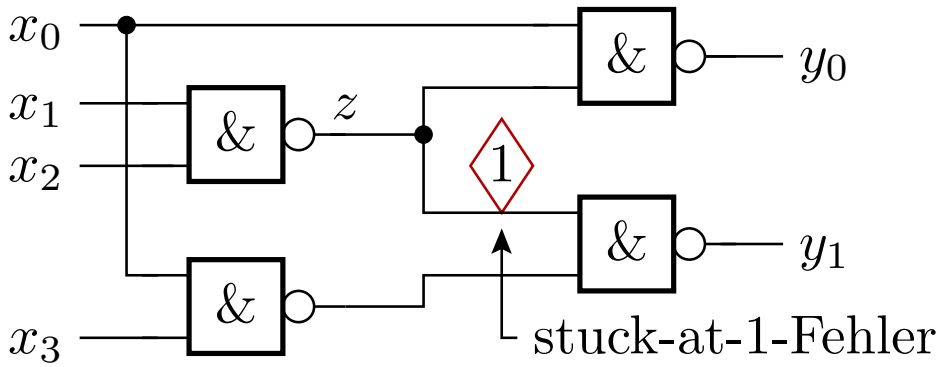
2. Testzeit, wenn alle Folgen von zwei möglichen Eingaben genau einmal abgearbeitet werden:

$$t_{\text{Test}} = 2^{32} \cdot 2^{32} \cdot 1\,\text{ms} = 5{,}8 \cdot 10^{8}\,\text{Jahre}$$

Aufgabe 5.6: Nachweismengen

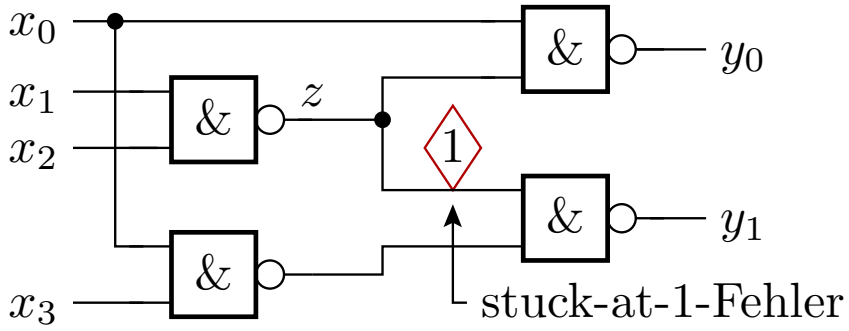
Bestimmen Sie für den eingezeichneten Haftfehler die Menge der Eingaben

1. M_{A} mit denen der Fehler angeregt wird,
   1. M_{B} mit denen der Fehler beobachtbar ist und
   2. M_{N} mit denen der Fehler nachweisbar ist.



Hinweis: Notation der Eingabemengen als Kreuze in der Wertetabelle auf der nächsten Folie.

3. Testberechnung

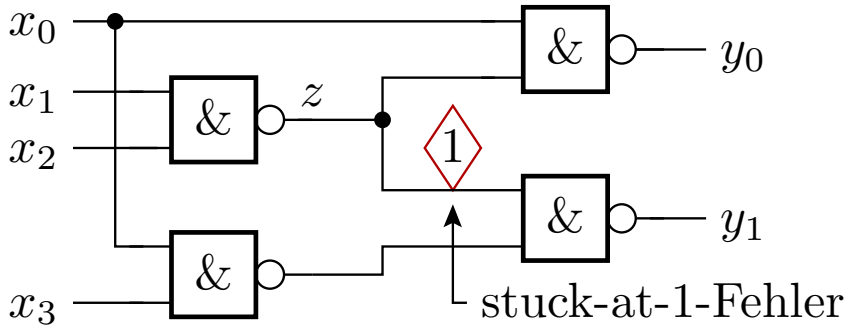


x_0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
x_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
M_A																
M_B																
M_N																

Menge der Eingaben $x_3x_2x_1x_0$:

1. M_A mit denen der Fehler angeregt wird:
2. M_B mit denen der Fehler beobachtbar ist:
3. M_N mit denen der Fehler nachweisbar ist:

Zur Kontrolle



x_0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
x_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
M_A							x	x							x	x
M_B	x	x	x	x	x	x	x	x	x			x	x		x	
M_N							x	x							x	

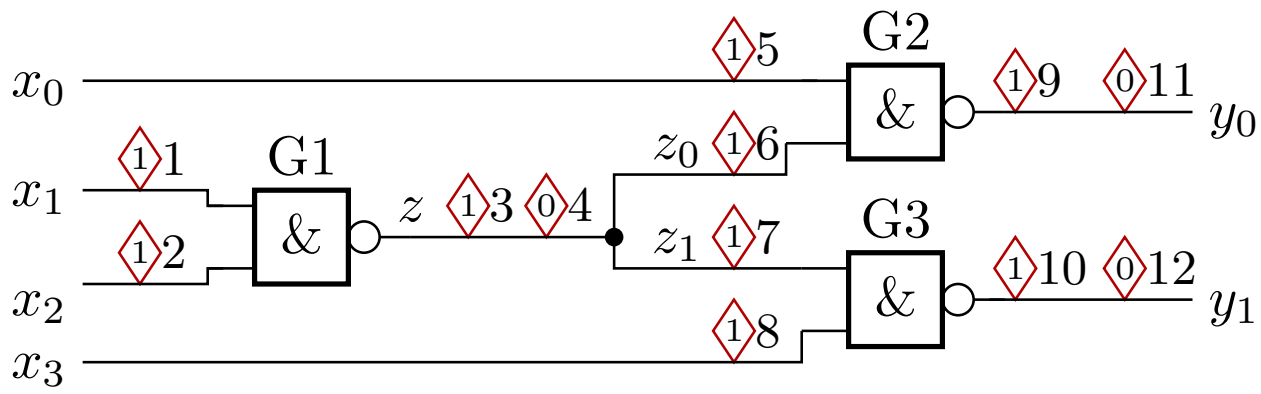
Menge der Eingaben $x_3x_2x_1x_0$, mit denen der Fehler:

1. angeregt wird: $M_A \in \{0110, 0111, 1110, 1111\}$
2. beobachtbar ist: $M_B \in \{0{*}{*}{*}, 1{*}{*}0\}$ (* – beliebiger Bitwert)
3. nachweisbar ist: $M_N \in \{0110, 0111, 1110\}$

Fehlersimulation

Aufgabe 5.7: Fehlersimulation

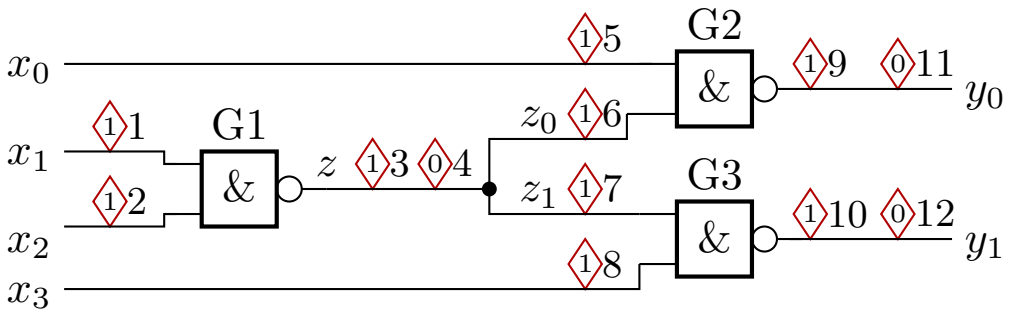
Schreiben Sie ein C-Programm zur fehlerparallelen Simulation der nachfolgenden Schaltung. Gutsimulation in Bit 0, Simulation der Fehler in den den Fehlern zugeordneten Bits 1 bis 12:



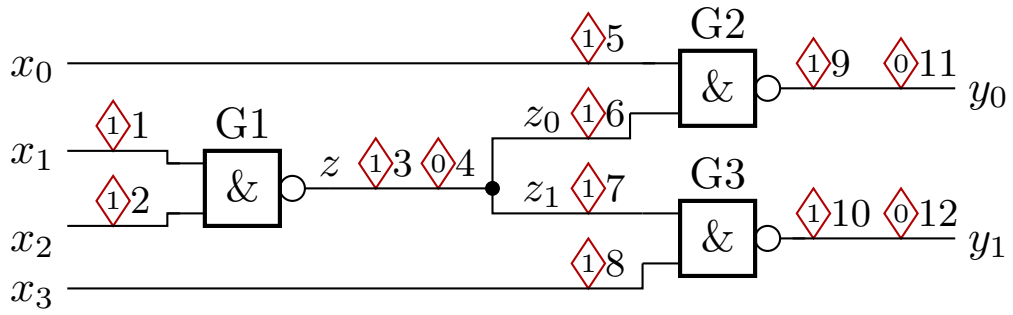
Programmrahmen:

```
uint16_t x0, x1, x2, x3, z, z0, z1, y0, y1;
<wiederhole für alle 16 Eingabemöglichkeiten>{
 <Simulation der Gatter und Fehler>
}
```

Zur Kontrolle




```
uint16_t x0, x1, x2, x3, z, z0, z1, y0, y1;
for (x3=0;x3==0xFF;~x3){
 for (x2=0;x2==0xFF;~x2){
  for (x1=0;x1==0xFF;~x1){
   for (x0=0;x0==0xFF;~x0){
    x0 = x0 | 1<<5;      // F5:  sa1(x0)
    x1 = x1 | 1<<1;      // F1:  sa1(x1)
    x2 = x2 | 1<<2;      // F2:  sa1(x2)
    x3 = x3 | 1<<8;      // F8:  sa1(x3)
    z  = ~(x1 & x2);     // Gatter G1
    z  = z | 1<<3;       // F3:  sa1(z)
    z  = z & ~(1<<4);    // F4:  sa0(z)
    z0 = z  | 1<<6;      // F6:  sa1(z0)
    z1 = z  | 1<<7;      // F8:  sa1(z1)
```




```
for (x3=0;x3==0xFF;~x3){
 for (x2=0;x2==0xFF;~x2){
  for (x1=0;x1==0xFF;~x1){
   for (x0=0;x0==0xFF;~x0){
     ...
     y0 = ~(x0 & z0);      // Gatter G2
     y0 = y0 | 1<<9;       // F9:  sa1(y0)
     y0 = y0 & ~(1<<11);   // F11: sa0(y0)
     y1 = ~(z1 & x3);      // Gatter G3
     y1 = y1 | 1<<10;      // F10: sa1(y1)
     y1 = y1 & ~(1<<12);   // F12: sa0(y1)
   }
  }
 }
```

Zur Kontrolle

```
uint16_t x0, x1, x2, x3, z, z0, z1, y0, y1;
for (x3=0;x3==0xFF;~x3){
 for (x2=0;x2==0xFF;~x2){
  for (x1=0;x1==0xFF;~x1){
   for (x0=0;x0==0xFF;~x0){
    x0 = x0 | 1<<5;      // F5:  sa1(x0)
    x1 = x1 | 1<<1;      // F1:  sa1(x1)
    x2 = x2 | 1<<2;      // F2:  sa1(x2)
    x3 = x3 | 1<<8;      // F8:  sa1(x3)
    z  = ~(x1 & x2);     // Gatter G1
    z  = z | 1<<3;       // F3:  sa1(z)
    z  = z & ~(1<<4);    // F4:  sa0(z)
    z0 = z  | 1<<6;      // F6:  sa1(z0)
    z1 = z  | 1<<7;      // F8:  sa1(z1)
    y0 = ~(x0 & z0);     // Gatter G2
    y0 = y0 | 1<<9;      // F9:  sa1(y0)
    y0 = y0 & ~(1<<11);  // F11: sa0(y0)
    y1 = ~(z1 & x3);     // Gatter G3
    y1 = y1 | 1<<10;     // F10: sa1(y1)
    y1 = y1 & ~(1<<12);  // F12: sa0(y1)
   }
}}
```

Aufgabe 5.8: Erforderliche Modellfehleranzahl

Um für eine Modellfehlerüberdeckung von $FC_{\mathrm{M}} = 99\%$ zu garantieren, soll das Simulationsabbruchkriterium eine ganzzahlige Restanzahl von nicht nachweisbaren Modellfehlern

$$\#F_{\mathrm{NErk}} \in \{0, 1, 2\}$$

sein. Wie groß muss die Anzahl der nicht redundanten Modellfehler $\#F_{\mathrm{M}}$ bei einer Irrtumswahrscheinlichkeit $\alpha_1 = \alpha_2 = 2\%$ sein?

Annahmen:

- Vernachlässigbare Abhängigkeiten im Fehlernachweis ($\kappa = 1$).
- Anzahl der nicht nachweisbaren Fehler $\#F_{\mathrm{NErk}}$ poisson-verteilt.

Vorschlag für den Lösungsweg:

- Abschätzung der Unter- und Obergrenze des Erwartungswertes.
- Für den ungünstigsten Fall des Erwartungswertes (Ober- oder Untergrenze?) muss die Modellfehleranzahl so groß sein, dass die zu erwartende Fehlerüberdeckung 99% ist.

Zur Kontrolle

Tabelle der Unter- und Obergrenzen des Erwartungswertes:

	$\alpha_1 = \alpha_2 = 0{,}5\%$		$\alpha_1 = \alpha_2 = 1\%$		$\alpha_1 = \alpha_2 = 2\%$	
X_{ist}	$E(X)_{\text{min}}$	$E(X)_{\text{max}}$	$E(X)_{\text{min}}$	$E(X)_{\text{max}}$	$E(X)_{\text{min}}$	$E(X)_{\text{max}}$
0	0,005	5,299	0,01	4,606	0,02	3,912
1	0,103	7,430	0,148	6,639	0,215	5,835
2	0,338	9,274	0,436	8,405	0,567	7,517

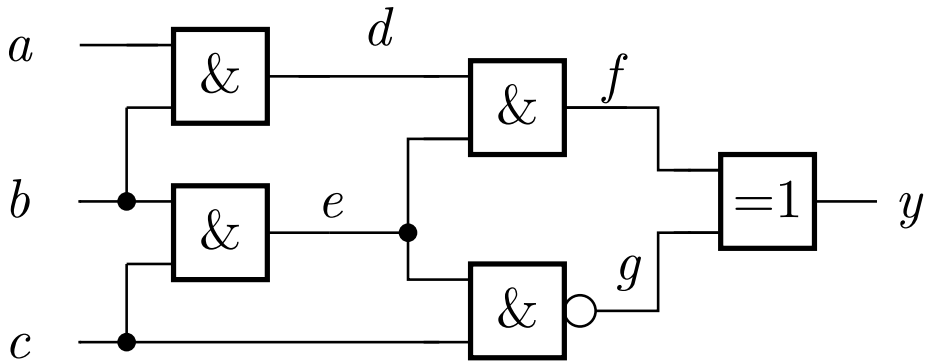
Erforderliche Modellfehleranzahl:

$$\varphi_{\text{M}} = \frac{E(\varphi_{\text{NErk}})_{\text{max}}}{1 - FC_{\text{M}}} = 100 \cdot E(\varphi_{\text{NErk}})_{\text{max}}$$

$\varphi_{\text{NErk.ist}}$	0	1	2
φ_{M}	392	584	751

D-Algorithmus

Aufgabe 5.9: D-Algorithmus[2]

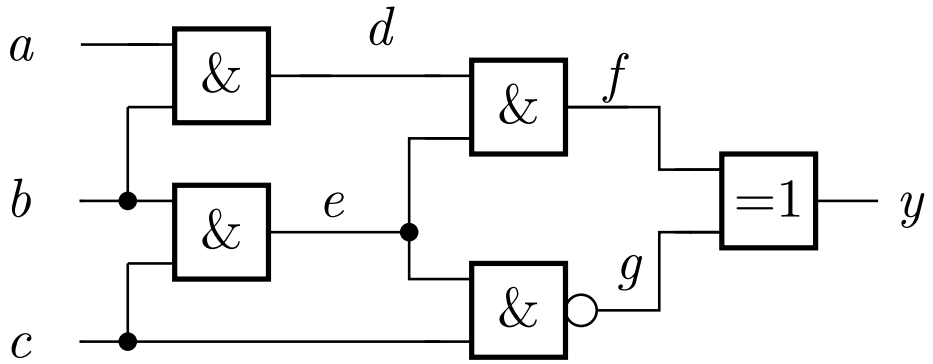


1. Geben Sie alle Möglichkeiten für die Sensibilisierung eines D-Pfads von Eingang c zum Ausgang y an.
2. Suchen Sie für den Haftfehler sa1(a) einen Test mit dem D-Pfad $a \rightarrow d \rightarrow f \rightarrow y$.

Kennzeichnung der Wertefestlegungen: F – lokale Fehlernachweisbedingung; I – implizite Festlegung; E – Entscheidung; $\bar{\text{E}}$ – invertierte Entscheidung; W – Widerspruch.

[2] Aus http://www.eda.ei.tum.de/fileadmin/tueieda/www/EI-BSc/EDS/tutorium/Tutorial_Dalgorithmus.pdf

Zur Kontrolle



Aufgabenteil 1: alle Pfade

$c \to g \to y$

$c \to e \to g \to y$

$c \to e \to f \to y$

$c \to e \to g \to y$

$c \to g \to y$ (mit $e \to f$)

$c \to e \to g \to y$ (mit f)

$c \to g \to y$ (mit e)

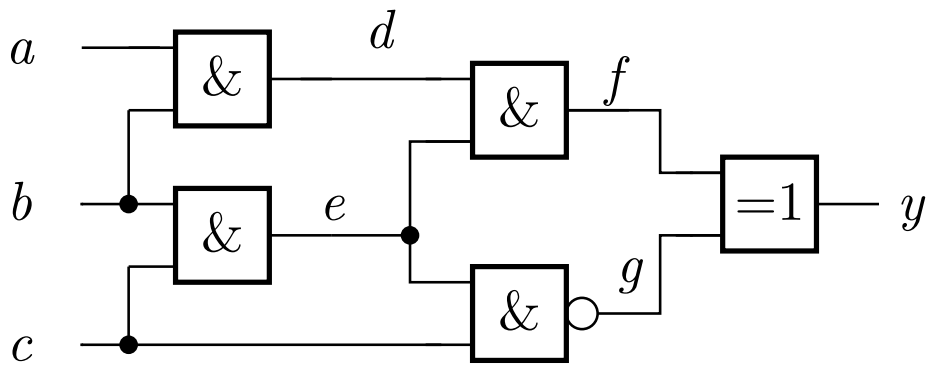
$c \to g \to y$ (mit $e \to f$)

Aufgabenteil 2:
Testsuche für
sa1(a), D-Pfad
$a - d - f - y$

$a =$D	F
$b = 1$	I
$d =$D	I
$e = 1$	I
$f =$D	I
$c = 1$	I
$g = 0$	I
$y =$D	I

$(a, b, c) = (1, 1, 1)$

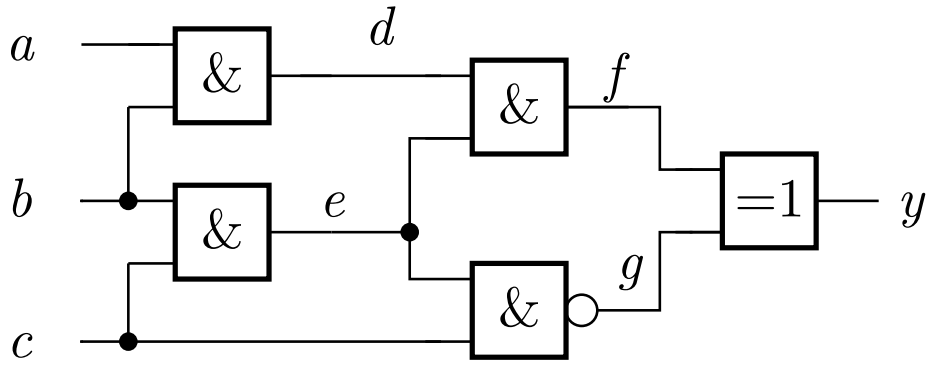
Aufgabe 5.10: D-Algorithmus Fortsetzung



3. Suchen Sie für den Haftfehler sa0(e) einen Test mit dem D-Pfad $e \to g \to y$.
4. Suchen Sie für den Haftfehler sa0(c) einen Test mit dem D-Pfad $c \to e \to f \to y$.

Kennzeichnung der Wertefestlegungen: F – lokale Fehlernachweisbedingung; I – implizite Festlegung; E – Entscheidung; $\bar{\mathrm{E}}$ – invertierte Entscheidung; W – Widerspruch.

Zur Kontrolle



Aufgabenteil 3: sa1(e)
über D-Pfad $e - g - y$

$e = \mathrm{D}$	F
$g = \bar{\mathrm{D}}$	I
$c = 1$	I
$b = 0$	I

$f = 1$	E	$f = 0$	$\bar{\mathrm{E}}$
$d = 1$	I	$y = \bar{\mathrm{D}}$	I
$e = 1$	W	$d = 0$	I

$(a, b, c) = (*, 1, 1)$

Aufgabenteil 4: sa0(c)
über D-Pfad $c - e - f - y$

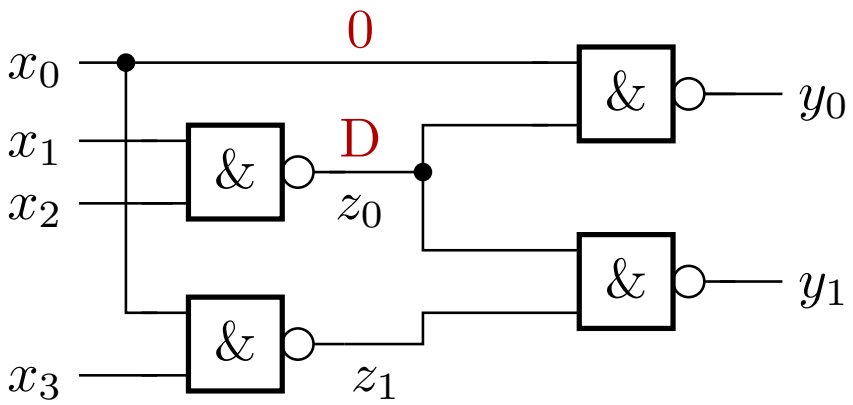
$c = \bar{\mathrm{D}}$	F
$b = 1$	I
$e = \bar{\mathrm{D}}$	I
$g = \mathrm{D}$	I
$d = 1$	I
$f = \bar{\mathrm{D}}$	I
$y = 1$	W

keine Lsg.

Implikationstest

Aufgabe 5.11: Implikationstest

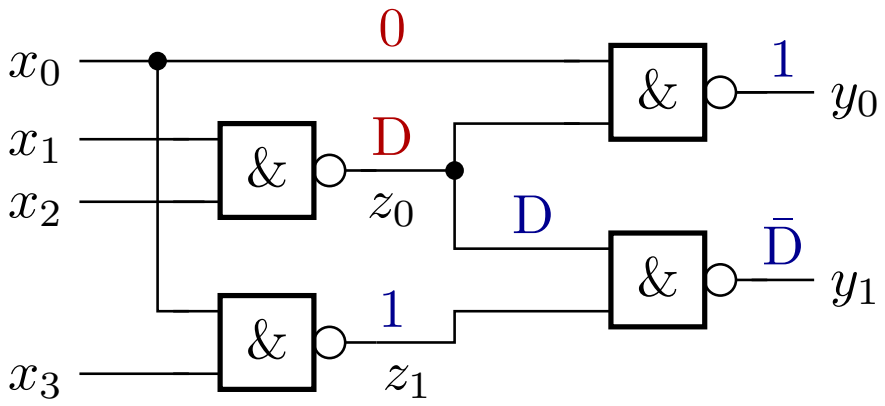
Bestimmen Sie für die nachfolgende Schaltung mit den beiden Signalfestlegungen (einmal »0« und einmal »D«) alle damit implizit festgelegten Signalwerte.



Zur Kontrolle

D	1	D
D	D	1

mehrere Mögl.

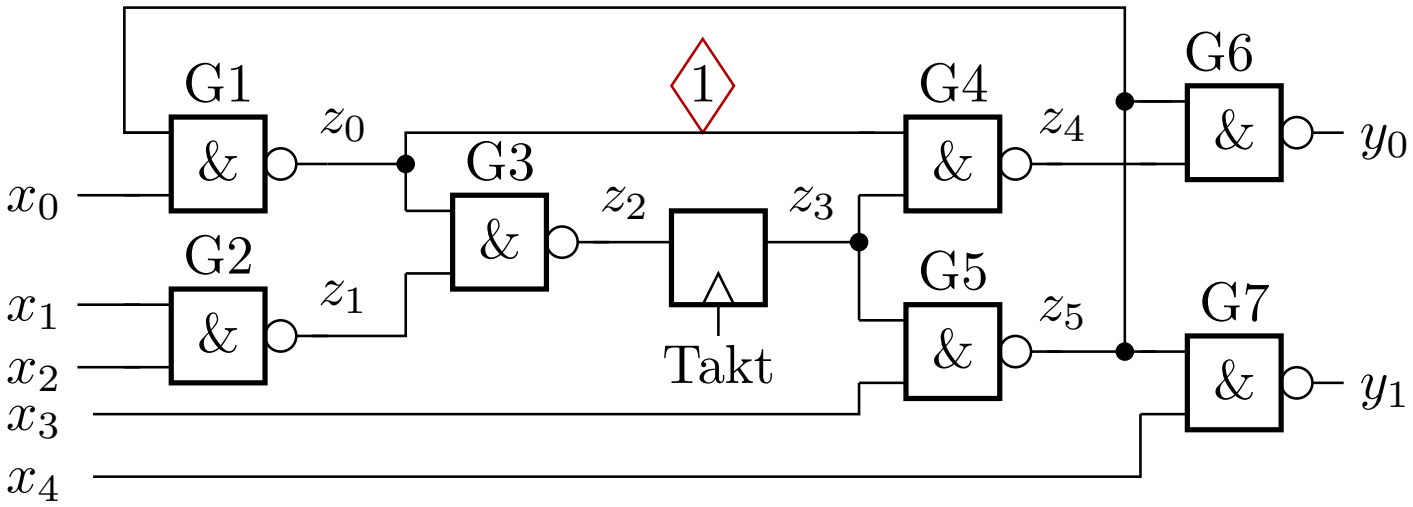


- $x_0 = 0$ impliziert $y_0 = 1$, $z_1 = 1$ und $y_1 = \bar{D}$.
- Unter der Annahme, dass der D-Pfad nicht zurückzutreiben ist, wäre $x_1 = 1$ und $x_2 = 1$ auch implizit festgelegt.

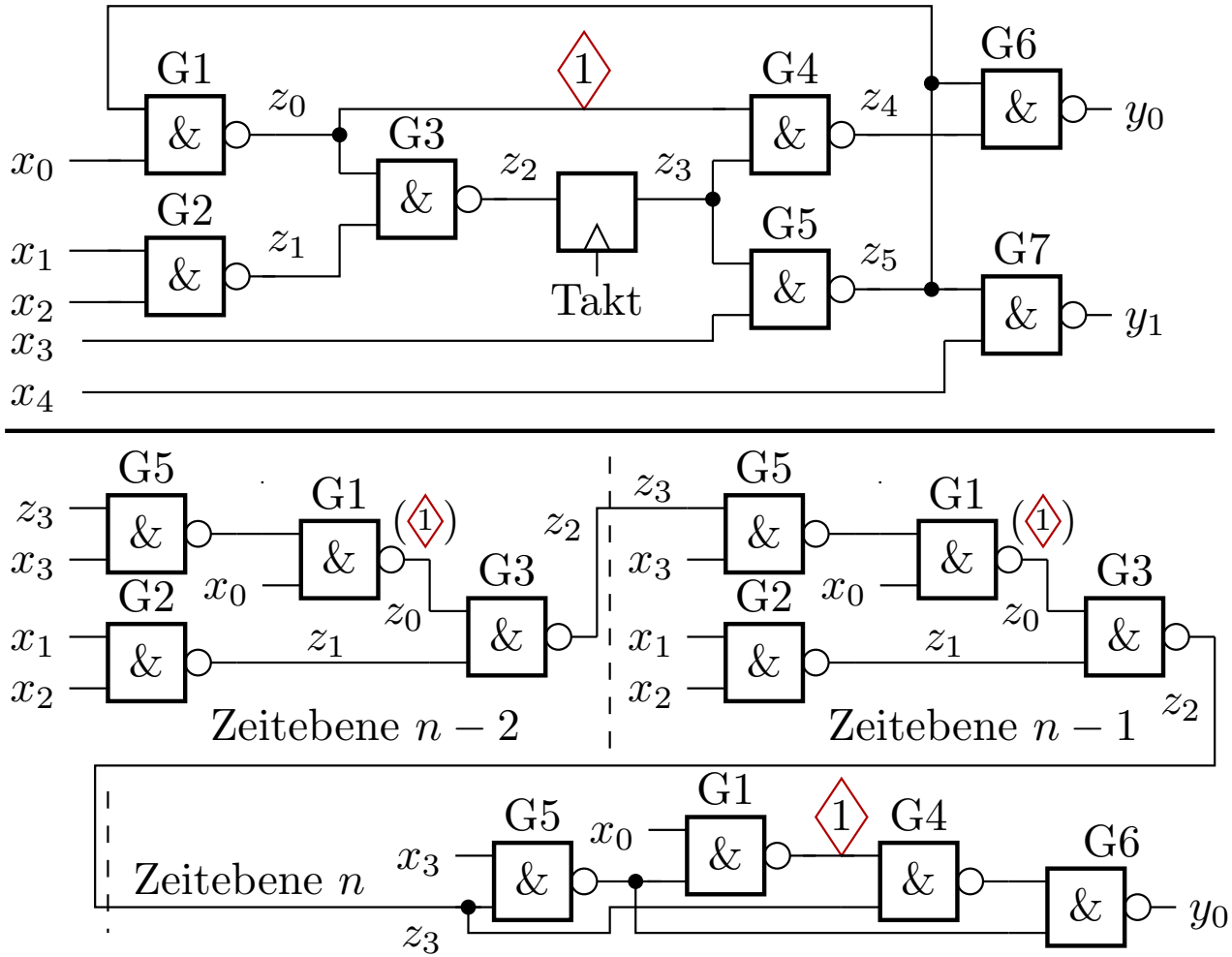
Sequentielle Schaltungen

Aufgabe 5.12: Kombinatorische Ersatzschaltung

Rollen Sie die nachfolgende Schaltung zu einer kombinatorischen Ersatzschaltung für die Testberechnung des eingezeichneten Haftfehlers auf mit einer Begrenzung der Länge der Steuerspfade auf max. drei Zeitebenen (max. 3 Schaltungskopien).



Zur Kontrolle



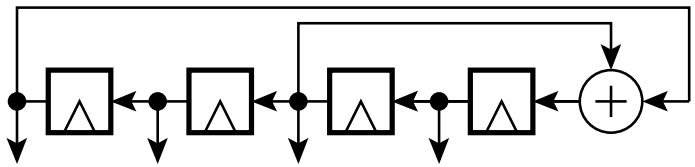
(1) wenn Haftfehler in allen Kopien

Selbsttest

Pseudo-Zufallsregister

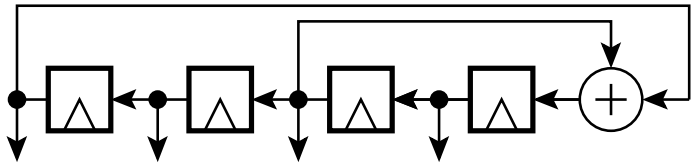
Aufgabe 5.13: Pseudo-Zufallszahlengenerator

Welche Zustandsfolgen durchläuft der nachfolgende Pseudo-Zufallszahlengenerator zyklisch ab dem Startzustand 1001?



Schritt	y_3	y_2	y_1	y_0
0	1	0	0	1
1				
2				
3				
4				
5				
6				

Lösung



Schritt	y_3	y_2	y_1	y_0
0	1	0	0	1
1	0	0	1	1
2	0	1	1	1
3	1	1	1	1
4	1	1	1	0
5	1	1	0	0
6	1	0	0	1

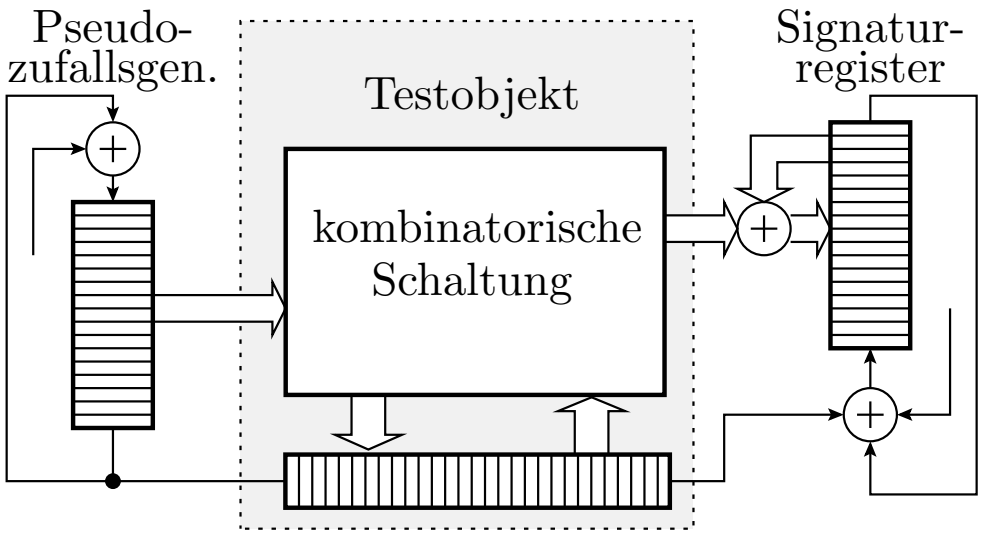
Vom gewählten Startwert werden nur fünf, d.h. nicht wie bei einer primitiven Rückführung alle 15 Zustände ungleich null zyklisch durchlaufen.

Selbsttest mit LFSR

Aufgabe 5.14: Selbsttest

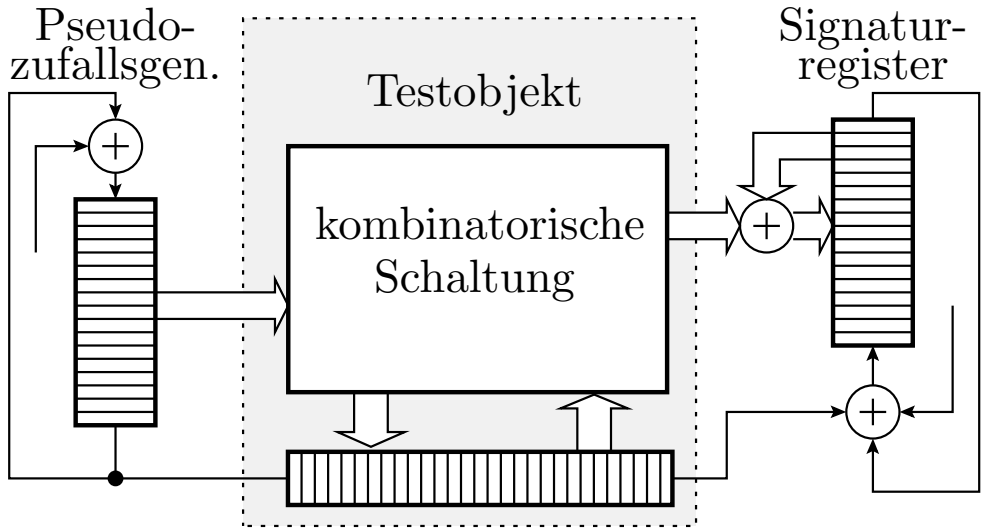
Die nachfolgende Selbsttestanordnung hat ein LFSR[3] als Pseudozufallsgenerator an den Eingängen, einem Scan-Register zum Lesen und Beschreiben der internen Speicherzellen und ein LFSR als Signaturregister an den Ausgängen.

- Beschreiben Sie den Ablauf des Selbsttests.
- Wie viele Taktschritte benötigt der Selbsttest?



[3] LFSR – Linear Feedback Shift Register.

Lösung

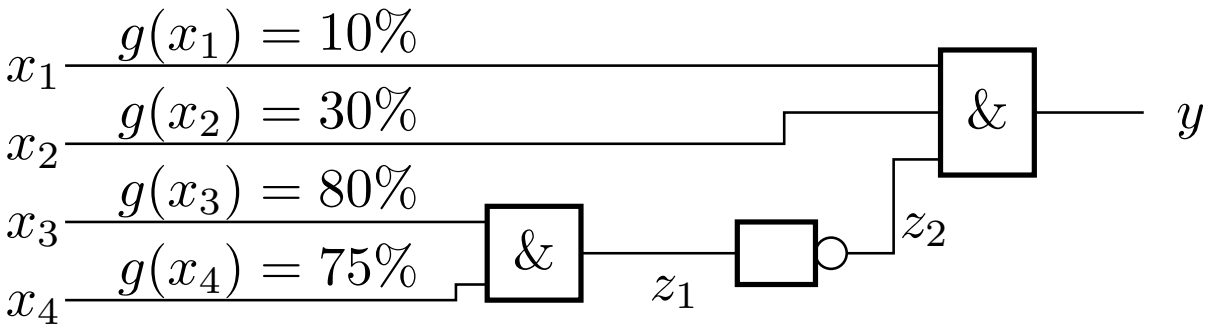


1. Testablauf:
   - Initialisiere Testmustergenerator
   - l_r Schiebeschritte zur Initialisierung des Scan-Registers
   - Übergabe Scan-Register; Initialisiere Signaturregister
   - Wiederhole für alle n Testschritte
     - Übernahme Scan-Register
     - l_r Schiebeschritte
     - Übergabe Scan-Register
   - Vergleich der Ist- mit der Sollsignatur
2. Testdauer in Taktschritten: $1 + l_\mathrm{r} + 1 + n \cdot (l_\mathrm{r} + 2) + 1$

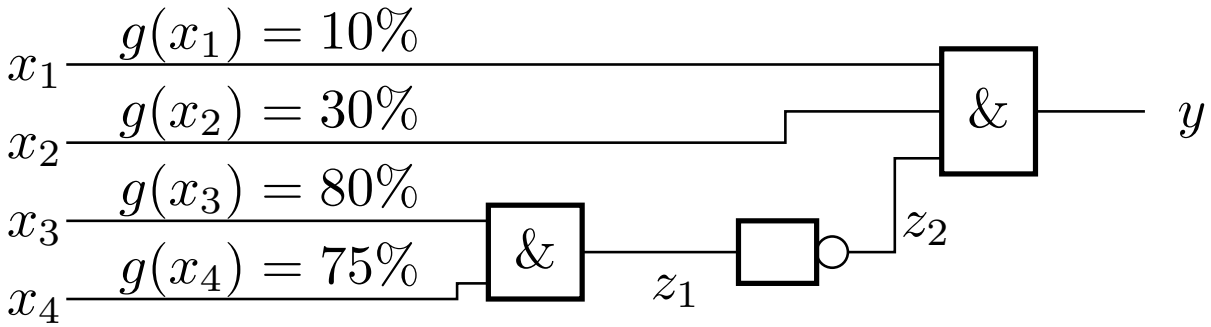
Fehlerorientierte Wichtung

Aufgabe 5.15: Wichtung und Beobachtbarkeit

Welche Beobachtbarkeit hat Eingang x_1 mit den vorgegebenen Wichtungen der Bitsignale an den Eingängen?



Lösung



- x_1 ist beobachtbar, wenn $x_2 = 1$ und $z_2 = 1$:

$$b\left(x_1\right) = g\left(x_2\right) \cdot g\left(z_2\right)$$

- z_2 ist eins, wenn $z_1 = 0$:

$$g\left(z_2\right) = 1 - g\left(z_1\right)$$

- z_1 ist eins, wenn $x_3 = 1$ und $x_4 = 1$:

$$\begin{aligned} g\left(z_1\right) &= g\left(x_3\right) \cdot g\left(x_4\right) \\ b\left(x_1\right) &= g\left(x_2\right) \cdot \left(1 - g\left(x_3\right) \cdot g\left(x_4\right)\right) \\ &= 30\% \cdot \left(1 - 80\% \cdot 75\%\right) = 12\% \end{aligned}$$