



TECHNISCHE
UNIVERSITÄT
DRESDEN

Fakultät Informatik, Institut für Technische Informatik, Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur

Test und Diagnose digitaler Systeme, prüffreundlicher Entwurf

Fabian Pilz



Dresden, 23.05.2012



DRESDEN
concept
Exzellenz aus
Wissenschaft
und Kultur

Gliederung

1. Motivation
2. Black-Box-Test
3. Fehlermodelle
4. Testmuster für kombinatorische Schaltungen
5. Sequentielle Schaltungen

Motivation

- Fehlerursachen:
 - Entwurfsfehler
 - Fertigungsfehler
 - physikalische Fehler
- Fertigungsfehler
 - Verunreinigungen
 - Kurzschlüsse
- Wahrscheinlichkeit eines Defektes steigt mit Chipfläche

Black-Box-Test

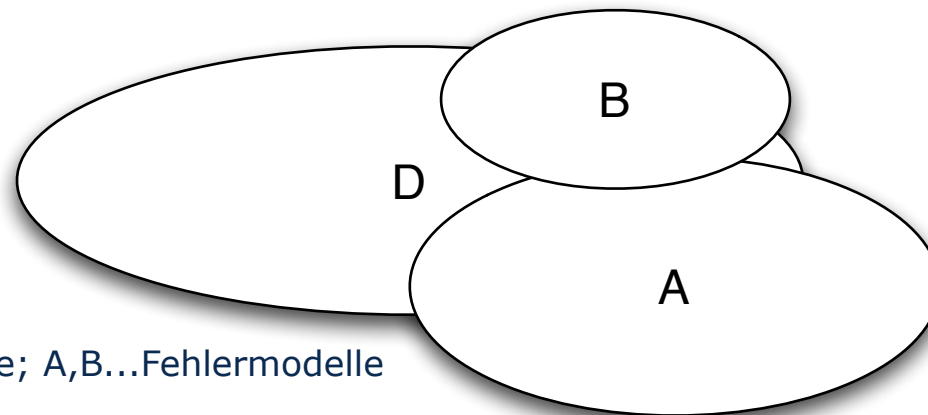
- Prüfung des Systems ohne Kenntnisse der Schaltungsstruktur
- Beispiel: Fulladder

a	b	cin	cout	sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

- Problem: Eingangsbelegungen wachsen exponentiell mit Anzahl der Eingänge

Fehlermodelle

- Allgemein
 - legen fest, welche Fehler unbedingt erkannt werden müssen
 - Kompromiss zwischen Aufwand und Qualität
 - hohe Fehlerüberdeckung \neq hohe Defektüberdeckung

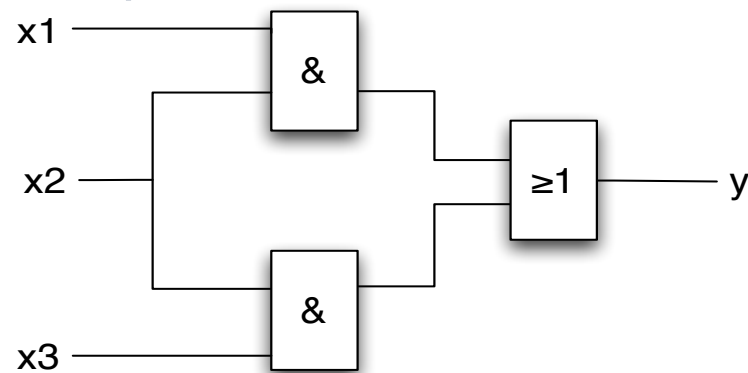


D...Defektmenge; A,B...Fehlermodelle

Entnommen aus [1]

Fehlermodelle

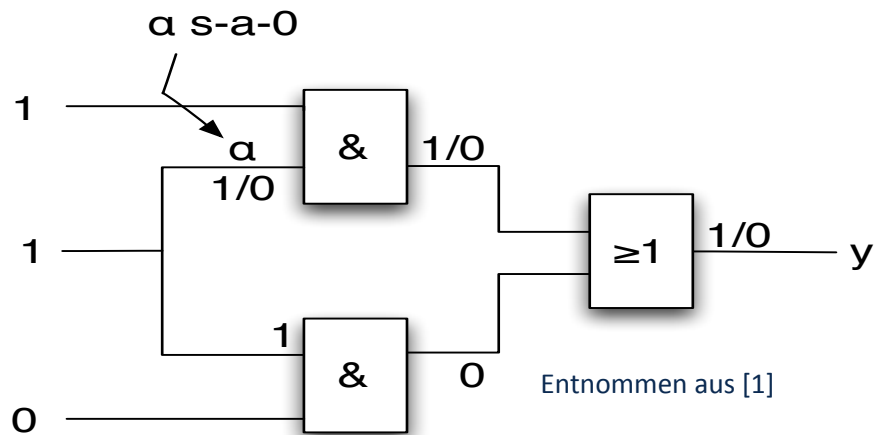
- Stuck-at-Fehlermodell
 - Festhängen einer Leitung auf einen logischen Wert
 - α -Stuck-at-0-Fehler
 - α -Stuck-at-1-Fehler
 - Beispiel:



Entnommen aus [1]

Fehlermodelle

- Beispiel Forts.



- Leitung a permanent auf 0
 - Erkennbar, wenn der Fehler bis zum Ausgang propagiert wird
 - entsprechende Anfangsbelegung muss gewählt werden

Fehlermodelle

- Bridging-Fehlermodell
 - Betrachtung von möglichen Kurzschlüssen zweier Leitungen
 - Ergebnis für beide Leitungen 0: AND-Typ
 - Ergebnis für beide Leitungen 1: Or-Typ
 - $m \times (m-1)$ mögliche Paarbildungen
 - schwache Bridgingfehler: Widerstand zwischen Leitungen mit $R \neq 0$

Fehlermodelle

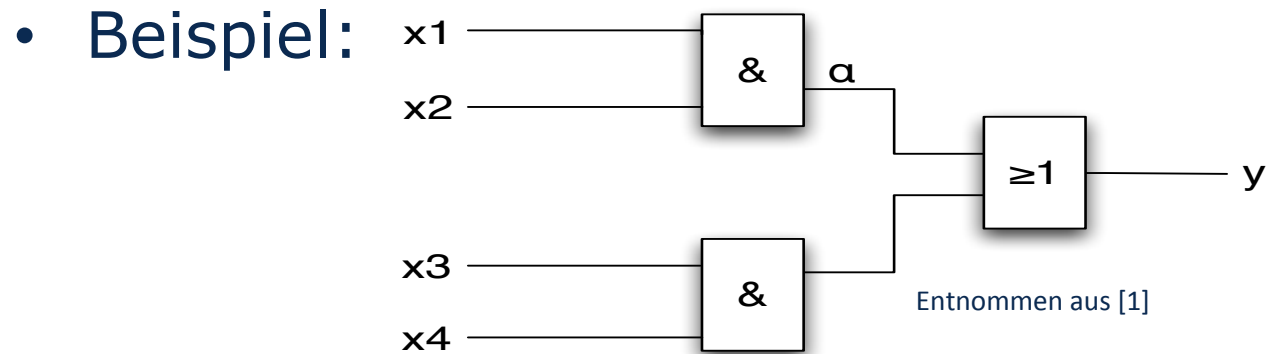
- Transistorfehler
 - Von Gatterebene auf Transistorebene:
zusätzliche Leitungen, die fehlerhaft sein können
 - Beispiel: Transistor leitet nie → Verhalten des gesamten Gatters kann beeinflusst sein

Testmuster für kombinatorische Schaltungen

- Ziel: Testmuster automatisch generieren
- Boolesche Differenz

$$\frac{\partial y}{\partial x_i} = y(x_i = 1) \oplus (x_i = 0)$$

Testmuster für kombinatorische Schaltungen



$$y = x_1x_2 + x_3x_4$$

$$\frac{\partial y}{\partial x_1} = (0x_2 + x_3x_4) \oplus (1x_2 + x_3x_4)$$

$$= (x_3x_4) \oplus (x_2 + x_3x_4)$$

$$= x_2 \overline{x_2x_4} = x_2 (\overline{x_3} + \overline{x_4})$$

Testmuster für kombinatorische Schaltungen

- Anwendung für stuck-at-Fehlermodell
 - Für a-s-a-0: $a=1$ und $dy/da=1$

$$\alpha \cdot \frac{\partial y}{\partial \alpha} = 1$$

$$\alpha \cdot \frac{\partial y}{\partial \alpha} = (x_1 x_2) \cdot \left[(0 + x_3 x_4) \oplus (1 + x_3 x_4) \right]$$

$$= x_1 x_2 \overline{x_3} + x_1 x_2 \overline{x_4} = 1$$

- Testmuster: $(1, 1, 0, 0); (1, 1, 0, 1); (1, 1, 1, 0)$

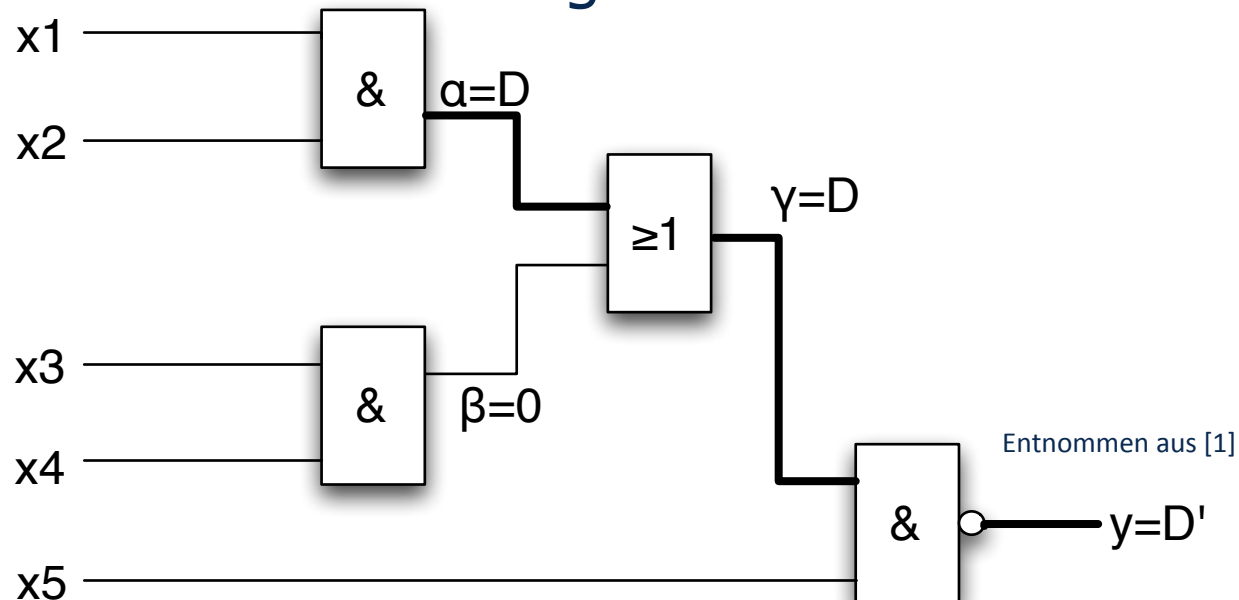
Testmuster für kombinatorische Schaltungen

- Pfadsensibilisierung

Signal korrekte Schaltung	Signal fehlerhafte Schaltung	Symbol
0	0	0
1	1	1
1	0	D
0	1	D'

Testmuster für kombinatorische Schaltungen

- **Pfadsensibilisierung**



- es soll eine D-Kette von Fehlerstelle bis zum Ausgang erzeugt werden
- kommt zu Stande, wenn $\bar{\beta} \cdot x_5 = 1 = \overline{x_3 x_4} \cdot x_5$

Testmuster für kombinatorische Schaltungen

- Pfadsensibilisierung
 - D-Algorithmus zur Testmuster generierung
 1. Eingabe: Kombinatorik, Fehler an Leitung a
 2. Ausgabe: Testmuster zur Fehlererkennung, andernfalls Fehlermeldung
 3. Methode
 - Schaltungseingänge: $\{0,1\}$
 - Schaltungsausgang: $\{D,D'\}$
 - Leitung a entfernen
 - Gattereingang von Leitung a auf D ($a-s-a-0$)
 - Gatterausgang von Leitung a auf 1 ($a-s-a-0$)
 - durch Backtracking wird versucht, einen stabilen Zustand zu finden

Testmuster für kombinatorische Schaltungen

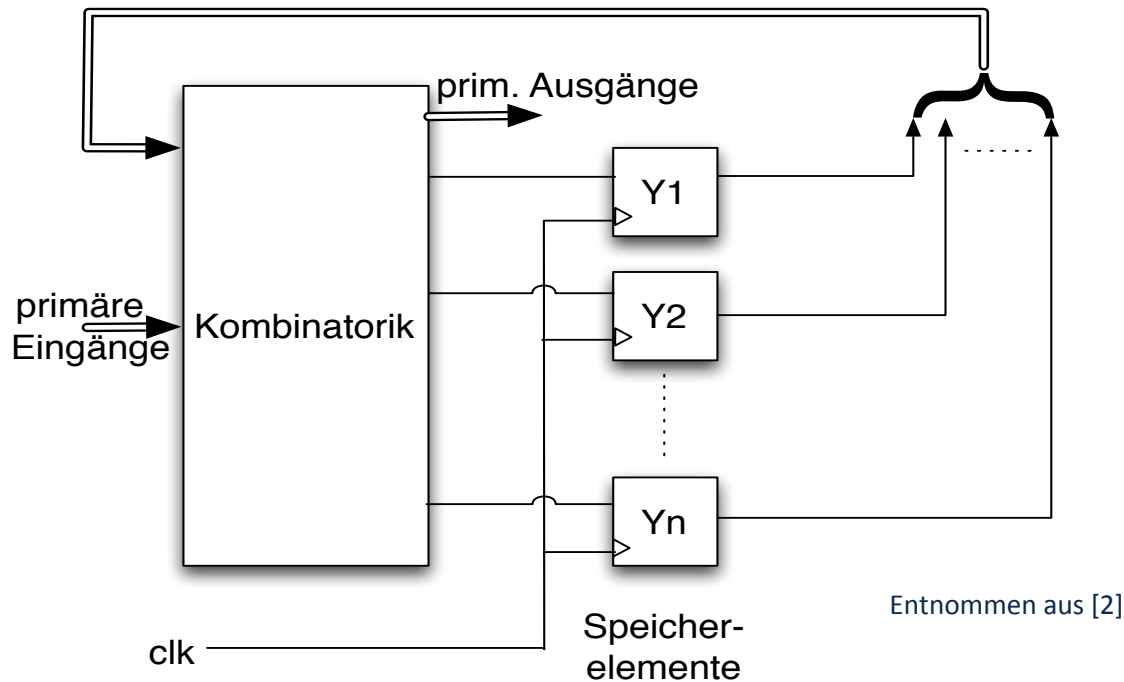
- **Fehlersimulation**

- bei gegebener Eingabe sollen Fehler entdeckt werden
- Algorithmus:

```
while (Fehlermenge  $\neq$   $\emptyset$ )  
{  
  Wähle eine Eingabe x;  
  if (x entdeckt Fehler aus Fehlermenge)  
  {  
    Gib x als Testmuster aus;  
    Entferne alle von x entdeckten Fehler aus Fehlermenge;  
  }  
}
```


Testmuster für sequentielle Schaltungen

- Allgemein
 - Speicherelemente: testen von außen schwierig

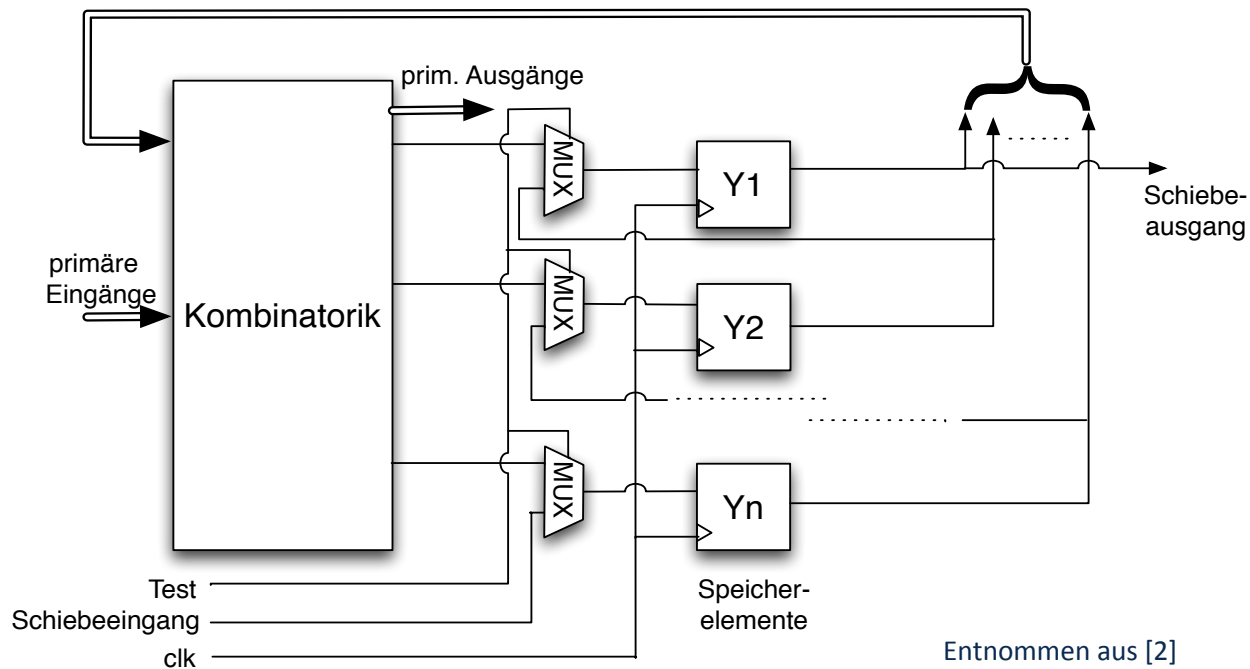


Testmuster für sequentielle Schaltungen

- Scan-Path
 - Speicherelemente können getrennt getestet werden
 - Speicherelemente können in beliebigen anderen Zustand gebracht werden
 - Ausgabe der Kombinatorik in Speicherelemente kann beobachtet werden

Testmuster für sequentielle Schaltungen

- Scan-Path



- im Testmodus Schieberegisterkette
- Test=0: Schaltung im Normalbetrieb

Testmuster für sequentielle Schaltungen

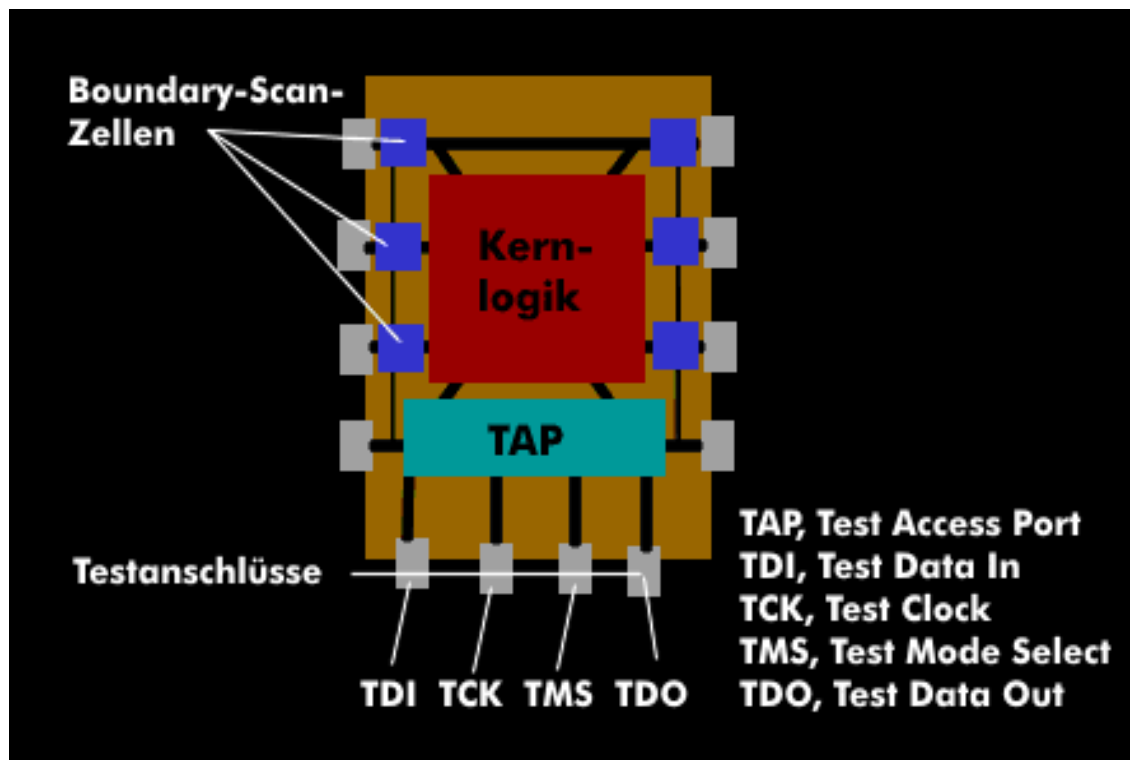
- Scan-Path
 - zunächst separate Überprüfung der Kombinatorik
 - danach Schiebetrieb mit geeignetem Registertest
 - Flush-Test
 - Schiebetest
 - asynchrone Speicherelemente nicht geeignet

Testmuster für sequentielle Schaltungen

- Boundary Scan
 - In-Circuit-Test
 - entwickelt durch „Joint Test Action Group“
 - Ziel: Standardisierung, da auf Leiterplatte Schaltungen verschiedener Hersteller sein können
 - Chip kann getestet werden, obwohl er schon in einem System verbaut ist

Testmuster für sequentielle Schaltungen

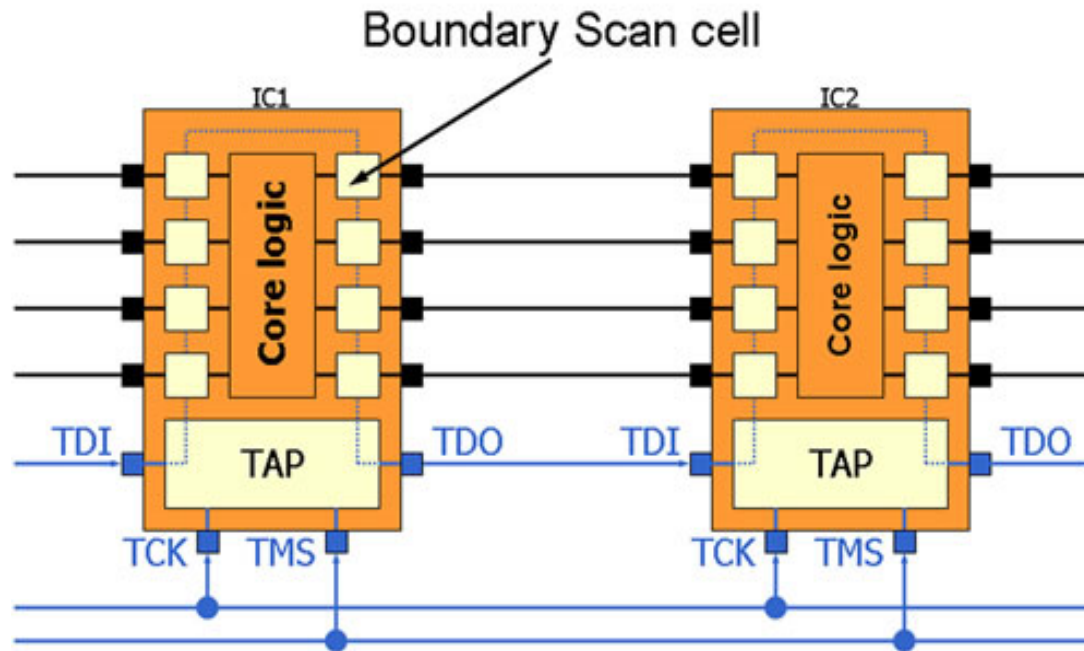
- Boundary Scan



Entnommen aus [4]

Testmuster für sequentielle Schaltungen

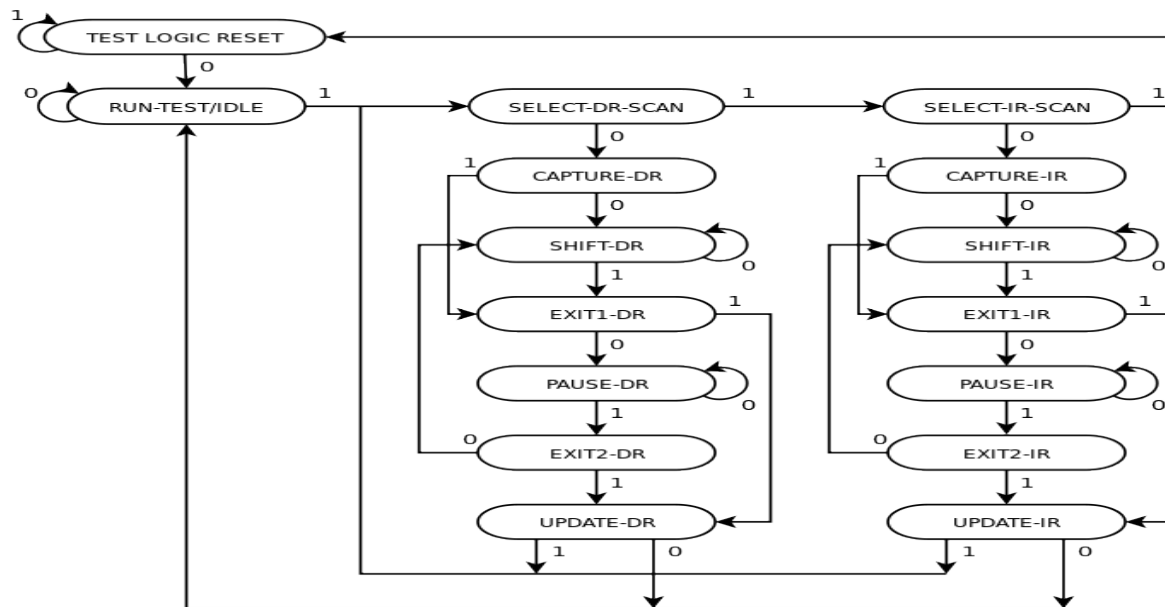
- Boundary Scan
 - Scankette



Entnommen aus [5]

Testmuster für sequentielle Schaltungen

- Boundary Scan
 - TAP-Controller
 - Steuerung der Boundary-Scan-Logik



Entnommen aus [6]

Testmuster für sequentielle Schaltungen

- **Boundary Scan**
 - **Befehlsregister**
 - bestimmt Betriebsmode des Boundary-Scan-Bausteins
 - wichtigste Befehle: Bypass, Sample/Preload, Extest
 - **Datenregister**
 - Informationsspeicher
 - bypass-Register
 - Boundary-scan-Register
 - **Boundary Scan Description Language (BSDL)**
 - Schnittstelle zwischen Chiphersteller und Testingenieur
 - Informationen über Testbussignale, Struktur der Boundaryzelle etc.

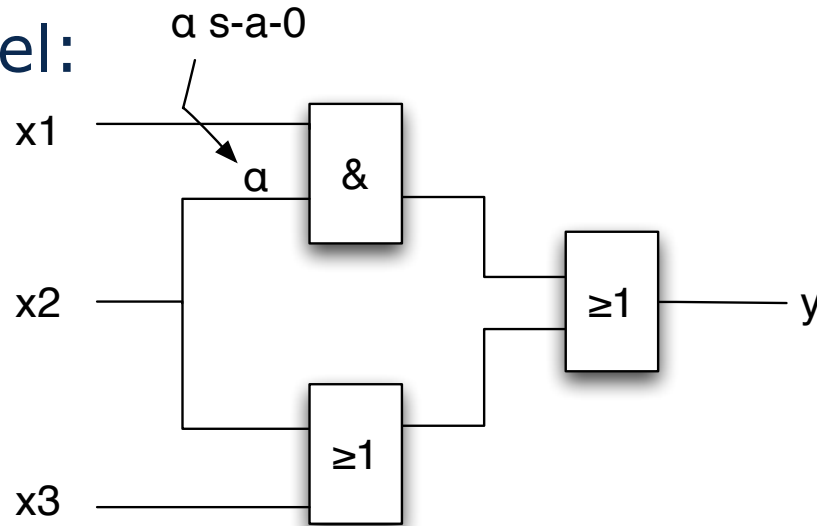
Quellen

- [1] Dirk Jansen: Handbuch der Electronic Design Automation; Carl Hanser Verlag: 2001
- [2] Hans-Joachim Wunderlich: Hochintegrierte Schaltungen; Springer-Verlag: 1991
- [3] Reinhard Kärger: Diagnose von Computern; Teubner: 1996
- [4] <http://www.itwissen.info/definition/lexikon/Boundary-Scan-BScan-boundary-scan.html>, 23.05.2012
- [5] http://www.goepel.com/fileadmin/downloads/einfrg_bscan.pdf, 23.05.2012
- [6] http://upload.wikimedia.org/wikipedia/de/1/1a/JTAG_TAP_Controller_State_Diagram.svg, 23.05.2012

Zusätzliche Folien

Redundante Schaltungen

- **Beispiel:**



$$\alpha \cdot \frac{\partial y}{\partial \alpha} = 1$$

$$x_2 \cdot \overline{x_1} \overline{x_2} \overline{x_3} = 1$$

$$0 = 1 \longrightarrow \textit{Widerspruch!}$$

Verbesserung D-Algorithmus

Algorithmus	geschätzte Beschleunigung	Jahr
D-Algorithmus	1	1966
PODEM	7	1981
FAN	23	1983
TOPS	292	1987
SOCRATES	1574	1988
Waicukauski	2189	1990
EST	8765	1991
TRAN	3005	1993
Rekursives Lernen	485	1995
Tafertshofer	25057	1997

aus Chiptest, Tobias Ruf, 10.7.2002

IDDQ-Test

- Untersuchung der Stromaufnahme eines Chips
 - Anlegen verschiedener Testmuster
 - bei erhöhter Stromaufnahme lässt sich auf einen Fertigungsfehler schließen
- Defekt-orientierte Methode