Grundlagen zuverlässiger fehlertoleranter Systeme

Sicherheit von Flugzeugtechnik und Autotechnik (USA)

Einheiten	10 k	100 Mio
Betriebsstunden/Jahr	55 Mio	30000 Mio
Kosten/Einheit	65 Mio	20 k
Todesfälle/Jahr	350	42 k
Unfälle/Jahr	170	21 k
Todesfälle/10^6 Betr.Std.	6,4	0,71
Bediener-Training	hoch	niedrig
Redundante Komponenten	alle flug-kritischen Systeme	Bremsen

Zuverlässigkeitsanfordererungen für sicherheitskritische Systeme der Flugzeugtechnik: 10-9 Fehler/h für eine Missionszeit von 10 h.

Phil Koopman CMU, Lecture 10/98

Kann man die Ansätze der Flugzeugtechnik übernehmen?

Zu teuer.

Unterschiedliche Betriebsbedingung.

Schwer durchzusetzende Wartungsintervalle.

Schlecht ausgebildete Benutzer.

Anforderungen an die Autoelektronik

Anfangszuverlässigkeit:	**(0 *km* / 0 *h*) Fehler: < 500 • 10^{-9}**
	im 1. Jahr Fehler: < 1000 • 10^{-9}
System-Lebenszeit:	**3500 *h* (ca. 5Jahre bei 2*h*/Tag)**
Garantie:	**≥ 1 Jahr, Ersatzteile ≥ 10 Jahre**
Umgebungsbedingungen:	**- 40 bis +85 ^oC**
Vibration:	**10 *Hz* bis 1 *kHz*, zufällig 5*g*, Sinus 2-5*g***
Shock:	**30 *g***
Versorgungsspannung:	**8-16*V***
	Motorstart mit 6*V* (-40 bis +85 ^oC), 18*V* für 2*h*, 24*V*
	für 1 *min* umgekehrte Polarität 13,5*V* für 1 *min*

Streßtest für Autoelektronik

• Funktionstest:	**8, 13.5, 16 *V* bei -40, 25, 85 ^oC**
• Hitzetest:	**85 ± 2 ^oC für 16*h* bei 16 *V* und 6000 *upm***
• Kältetest:	**- 40 ± 3 ^oC für 2*h***
• Lagerung:	**85 ± 2 ^oC für 504*h***
• Temperaturschock:	**- 40 bis 85 ^oC Übergang in 30 *sek.* 25 mal**
• Temperaturänderung:	**- 40 bis 85 ^oC , 3 ± 0.6 ^oC */min* für 2 Zyklen**

Stefan Poledna: "Lecture on Fault-Tolerant Systems", Vorlesungsfolien, Institut für Technische Informatik, TU Wien, SoSe 1996

Definitionen:

Verläßlichkeit (Dependability): "Doing the right thing at the right time!"

Die Verläßlichkeit (Dependability) eines Systems ist die Qualität einer vom System erbrachten Funktion (Service), in die begründbar und berechtigterweise Vertrauen (reliance) gesetzt werden kann.

Die Funktion (Service) ist das an der Schnittstelle zu anderen Systemen, die mit dem betrachteten System interagieren, beobachtbare Systemverhalten. Die Qualität bezieht sich auf die Übereinstimmung der erbrachten mit der spezifizierten Systemfunktion.

Fragestellungen:

- **Fehler: Welche Klassen von Fehlern werden berücksichtigt?**
- **Attribute: Welche Aspekte der Verläßlichkeit werden besonders hervorgehoben?**
- **Maße: Wie läßt sich die Verläßlichkeit quantitativ erfassen?**

Laprie, J.-C. : Dependability: A unifying concept for reliable, safe, secure computing.
In IFIP Congress, volume 1, (1992)pages 585-593.

Aspekte der Verläßlichkeit

Überlebensfähigkeit (Reliability) bedeutet Zuverlässigkeit in Hinblick auf ununterbrochenes korrektes Systemverhalten. Es ist als die Wahrscheinlichkeit definiert, daß ein zu Beginn fehlerfreies System bis zu einem bestimmten Zeitpunkt fehlerfrei bleibt.

der Ansatz in der Flugzeugindustrie

Verfügbarkeit (Availability) bedeutet Zuverlässigkeit in Hinblick auf die momentane Bereitschaft eines Systems zur Erbringung eines Service. Verfügbarkeit wird als quantitaves Maß definiert, das die Ausfalldauer zu der Dauer korrekten Systemverhaltens in Beziehung setzt, d.h. die Wahrscheinlichkeit, das System zu einem beliebigen Zeitpunkt fehlerfrei anzutreffen.

der Ansatz in der Autoindustrie

Prozeßsicherheit (Safety) bedeutet Zuverlässigkeit in Hinblick auf das Verhindern katastrophaler Auswirkungen eines Systemverhaltens auf seine Umgebung, wobei mit Umgebung meist die physikalische, reale Umgebung gemeint ist, wie z.B. industrielle Prozeßsteuerungsanlagen, Kraftwerke, Verkehrslenkungssysteme, u.s.w.

der Ansatz in der industriellen Automatisierung

Informationssicherheit (Security) bedeutet Zuverlässigkeit in Hinblick auf die Erhaltung der Vertraulichkeit (Confidentiality) und Integrität (Integrity) von Information in einem Computersystem.

Wie häufig fallen Komponenten aus?

λ: Ausfälle/10^6 Betriebsstunden (~115 Jahre)

Militärischer Microprozessor	0,022
Automotiver Microprozessor	0,12
Elektromotor	2,17
Bleibatterie	16,9
Ölpumpe	37,3

zum Vergleich:

einzelner menschlicher Operateur:	$100/10^6$ Aktionen
Mensch in der Krisenbewältigung:	$300000/10^6$ Aktionen

10^{-9} Ausfälle/h in der Flugtechnik
~10^{-9} Ausfälle/h in der Autotechnik

Basiszuverlässigkeit unzureichend

Mechanismen der Verläßlichkeit

Fehlervermeidung
Fehlertoleranz

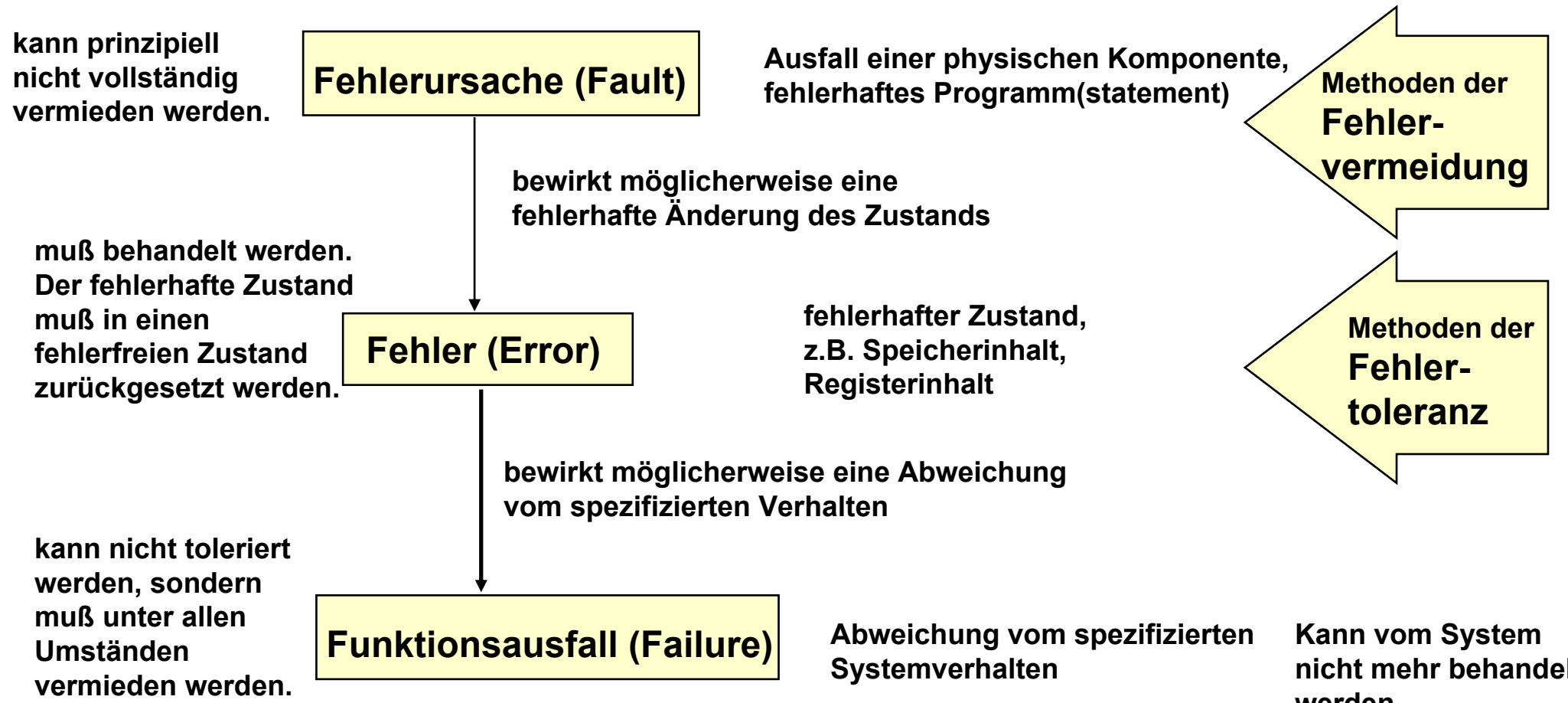
Alle Mechanismen der Fehlertoleranz beruhen auf Redundanz

- **Informationsredundanz**
- **Komponentenredundanz**
- **Zeitredundanz**

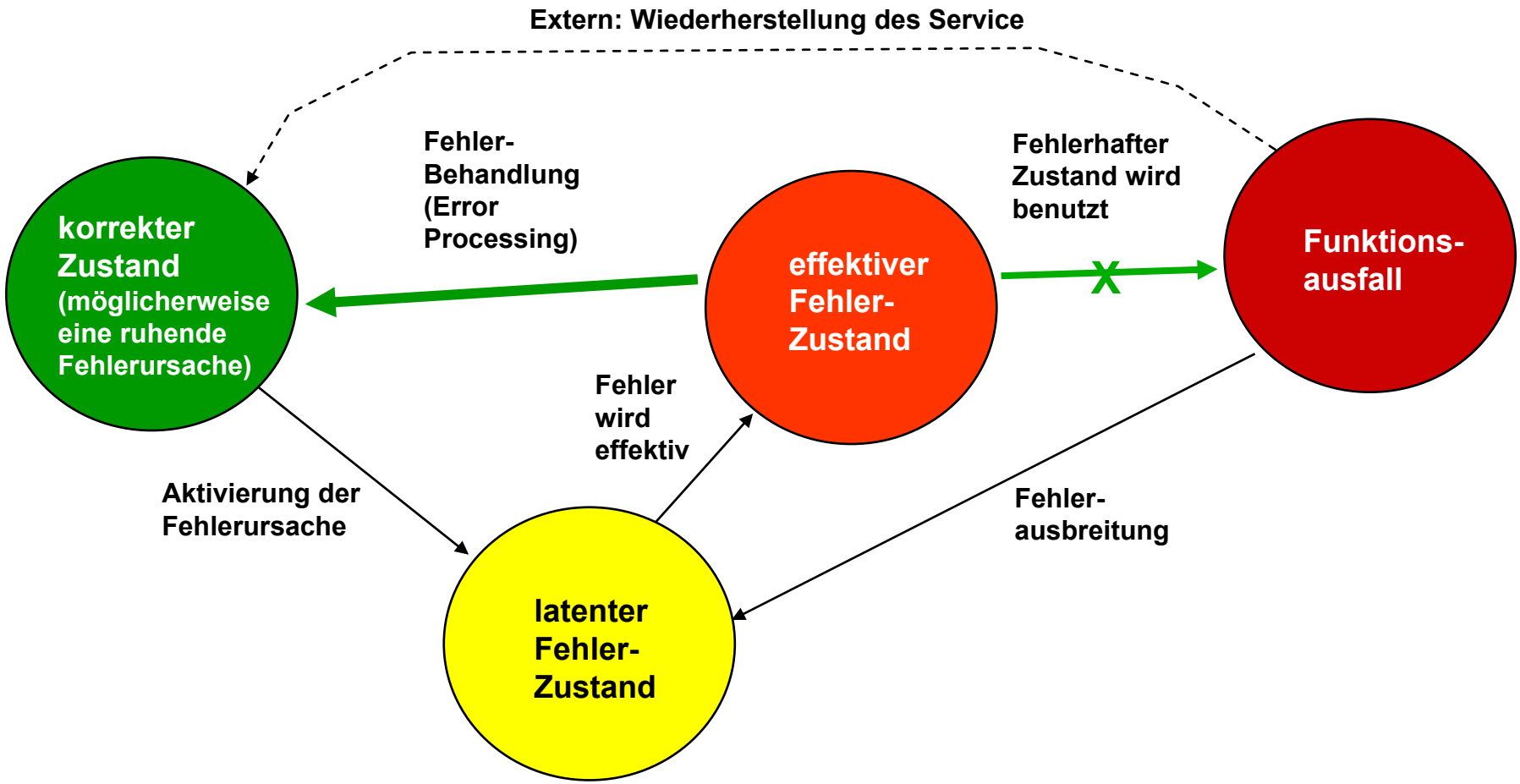
Die Wahl der Redundanzmethode ist davon abhängig, welche Fehlerklasse berücksichtigt wird!

Mechanismen der Fehlertoleranz

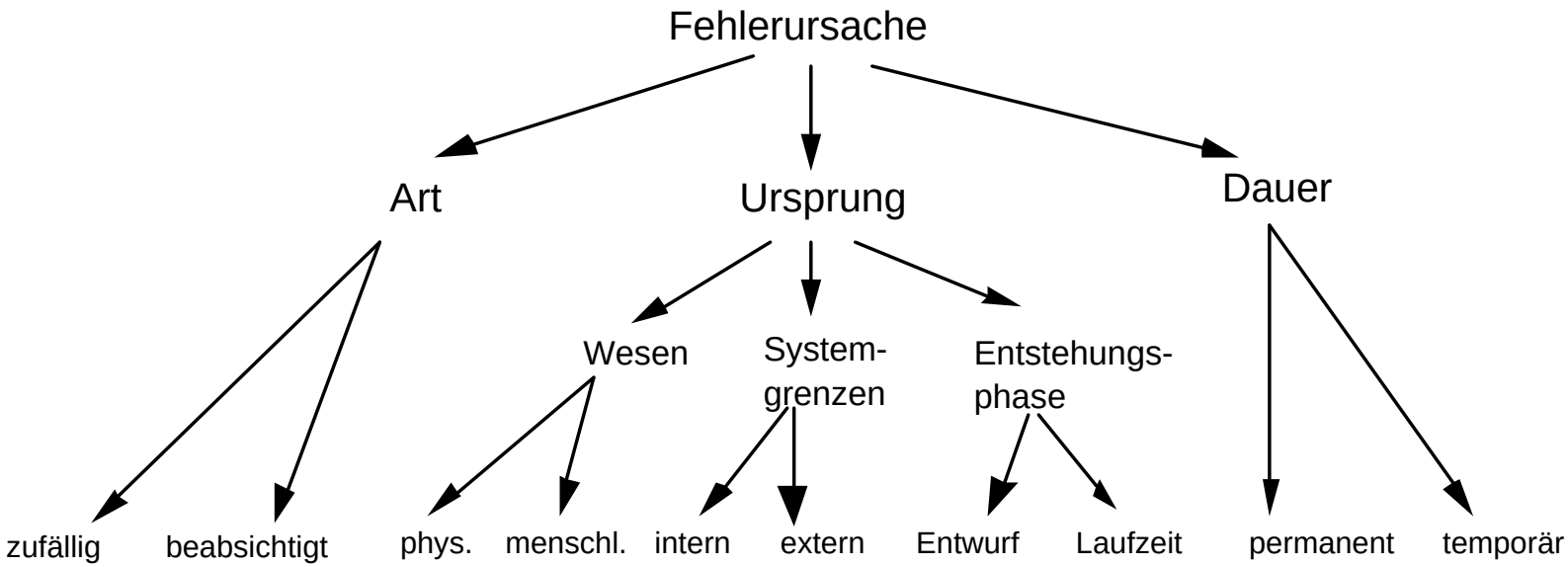
Fehlerklassifizierung



Zustandsdiagramm der *Fehlerursache-Fehler-Funktionsausfall* -Kette



Fehlerklassifizierung nach der Fehlerursache



Beispiel: Physische Fehlerursachen

Ursprung des Fehlers: physikalisches Ereignis								
Art		System-grenzen		Entstehungs-phase		Dauer		Bezeichnung
zufällig	beabsichtigt	intern	extern	Entwurf	Laufzeit	permanent	temporär	
•		•			•	•		permanenter phys. Fehler
•		•			•		•	intermittierender Fehler
•			•		•		•	transienter Fehler

Typische permanente Fehler: Stuck-at- {0,1}, Stuck-together.
Typische intermittierende Fehler: Muster-abhängige Fehler, Temperatur- und Zeitfehler.
Typische transisente Fehler: α,β,γ-Teilchen, temperaturabhängige Fehler.

Beispiel: Unbeabsichtigte menschliche Fehlerursachen

Ursprung des Fehlers: menschliches Fehlverhalten								
Art		System-grenzen		Entstehungs-phase		Dauer		Bezeichnung
zufällig	beabsichtigt	intern	extern	Entwurf	Laufzeit	permanent	temporär	
•		•		•		•		Entwurfsfehler
•			•		•		•	Bedienungsfehler

Beispiel: Beabsichtigte menschliche Fehlerursachen

Ursprung des Fehlers: menschliches Fehlverhalten								
Art		System-grenzen		Entstehungs-phase		Dauer		Bezeichnung
zufällig	beabsichtigt	intern	extern	Entwurf	Laufzeit	permanent	temporär	
	•		•		•	•		Intrusion
	•		•		•		•	
	•	•			•	•		Virus
	•	•		•		•		Trojanisches Pferd
	•	•		•		•		maliziöse Logik

Problem der Informationssicherheit !

Möglichkeiten der Fehlererkennung

Diagnostische Tests: Überprüfen einzelne Komponenten des Systems, indem sie aus der Struktur der Komponenten Eingabewerte ableiten, die bestimmte Fehler in eine Komponente aktivieren und zum Ausgang propagieren.

Code Tests: Basieren auf fehlererkennenden Codes.

Timing Tests: Überprüfen der bekannten Zeitbedingungen.

Replikations-Tests: Mehrfache Ausführung einer Berechnung und Vergleich der Ergebnisse.

Reversive-Tests: Umkehrung der Berechnung. Aus den Ergebnisses werden die Eingaben abgeleitet verglichen.

Plausibilitäts-Tests: überprüfen Ergebnisse auf ihre Plausibilität hinsichtlich der beabsichtigten Nutzung.

Strukturelle Tests: Überprüfen die Struktur von Datanstrukturen.

Testen permanenter Hardwarefehler

Was testen? Chips auf einem Board, chipinterne Komponenten.

Wie testen?
Generelle Probleme: Komplexe Schaltungen erfordern ausführliche Tests. Große Mengen von Testdaten, Probleme mit sequentiellen Schaltungen, Auswertung schwierig.

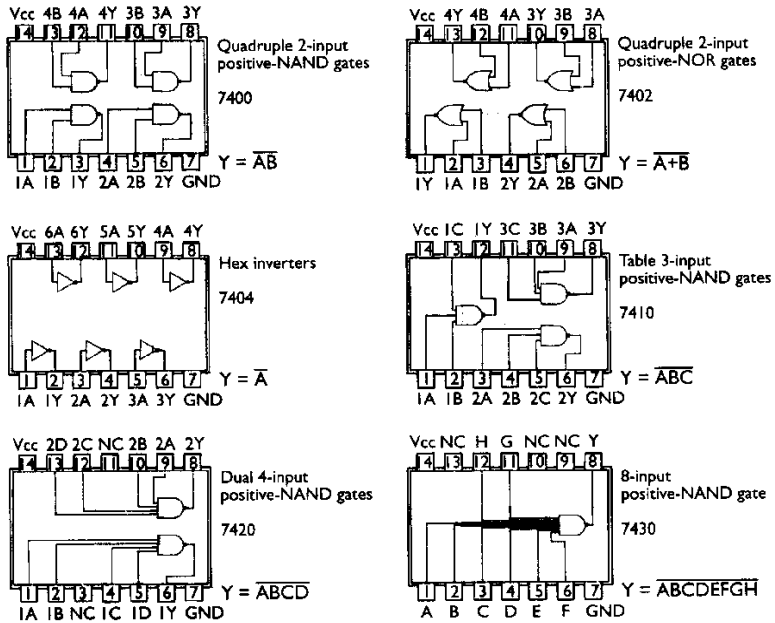
Probleme auf Board-Ebene: Isolation von Chips vom Rest der elektronischen Schaltungen, Anlegen von Testmustern an die Pins der Testkandidaten.

Probleme auf Chip-Ebene: Isolation von Komponenten, Anlegen von Testmustern an die internen Strukturen, eklatant schlechtes Verhältnis von verfügbaren Pins zu Anzahl der internen Komponenten.

Problem verschärft sich durch "System-on-a-Chip" Technik.

Problem: Komplexe interne Struktur, wenige externe Verbindungen (PINS)

1980



3 Pins/Gatter

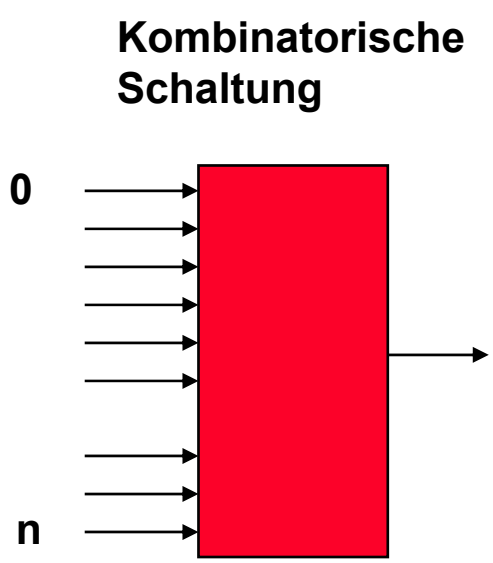
2002

> 0,0001 Pins/Gatter

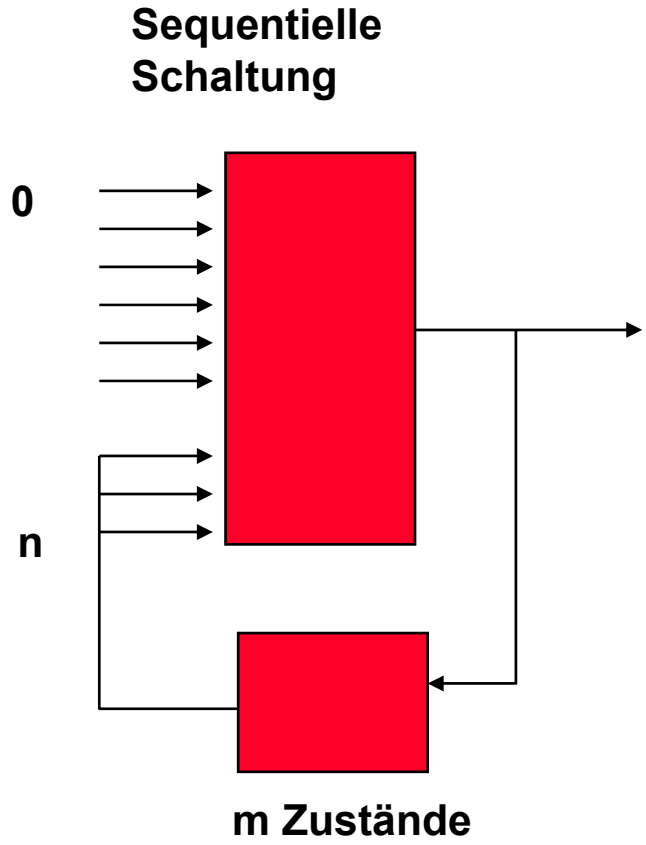
Kontrollierbarkeit (Controllability): Wie kann man einen bestimmten Zustand herbeiführen, damit der Fehler aktiviert wird?

Beobachtbarkeit (Observability): Wie kann man einen Fehler zum Ausgang propagieren?

Problem des ausführlichen Testens

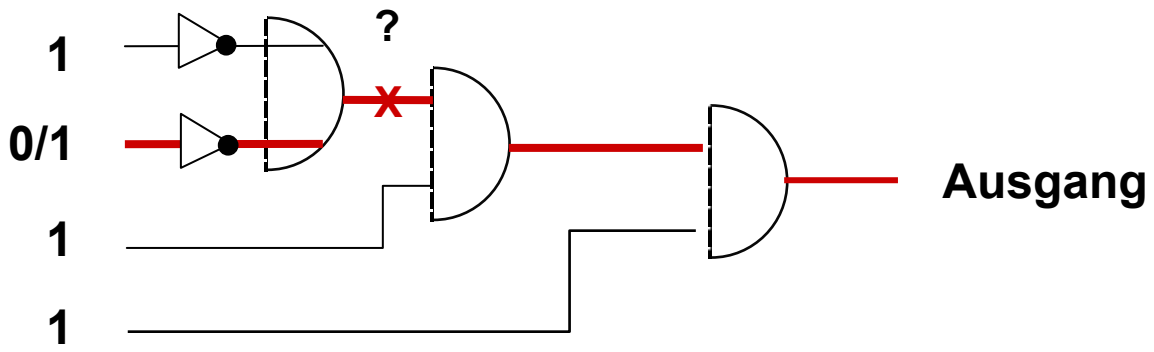


2^n Testmuster



2^{n+m} Testmuster

Aktivierung eines Fehlers



Pfadsensitivierung: **Anlegen eines Musters, das es erlaubt, einen bestimmten Fehler zu aktivieren und auf einem Pfad zum Ausgang zu propagieren.**

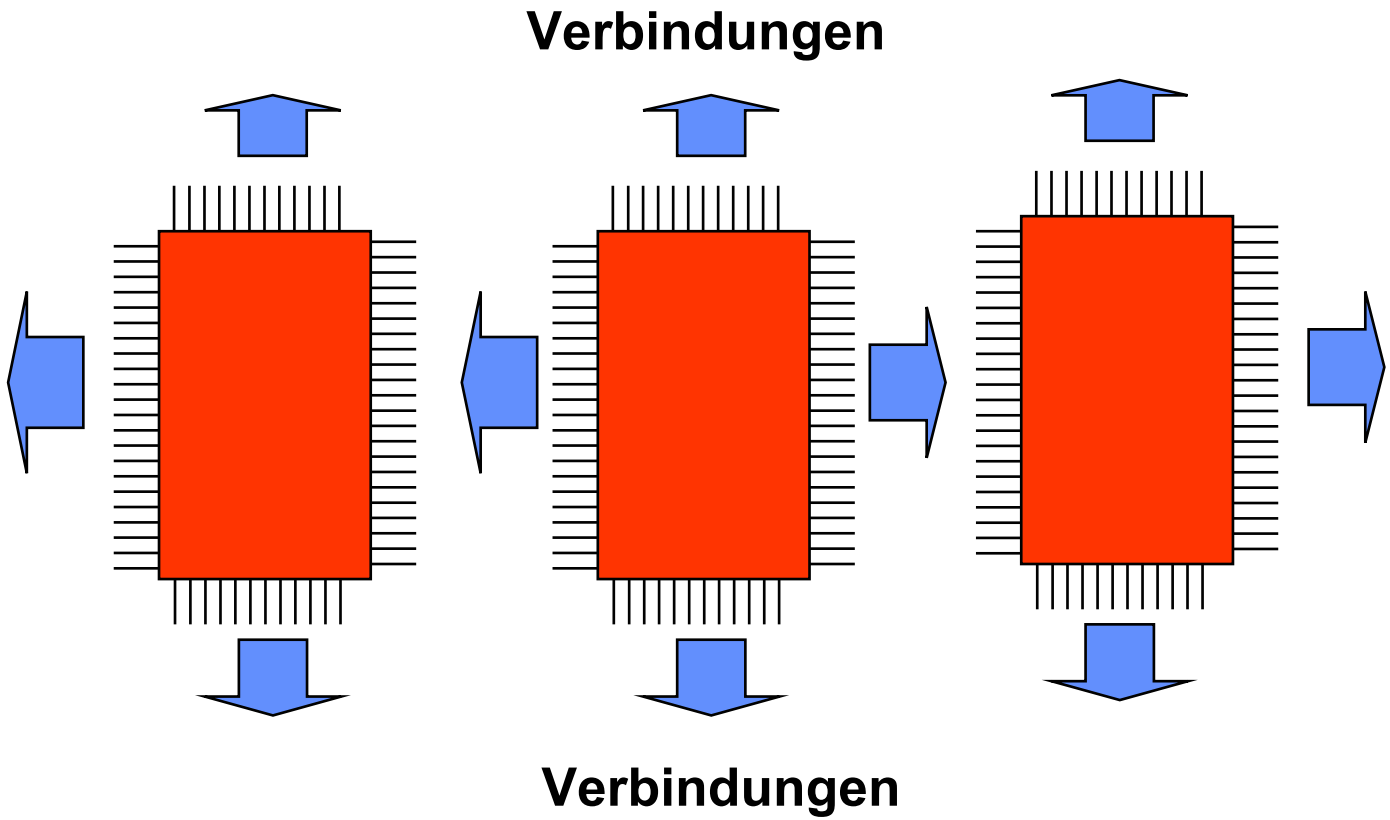
Verfahren zur Testunterstützung

Boundary Scan: Isolation von Chips auf dem Board, Heranführen beliebiger Testmuster an den Chip, Auslesen der Ergebnisse

Scan Path Design: Heranführen beliebiger Testmuster an interne Komponenten auf dem Chip, Auslesen der Ergebnisse

Boundary Scan: Der JTAG Standard

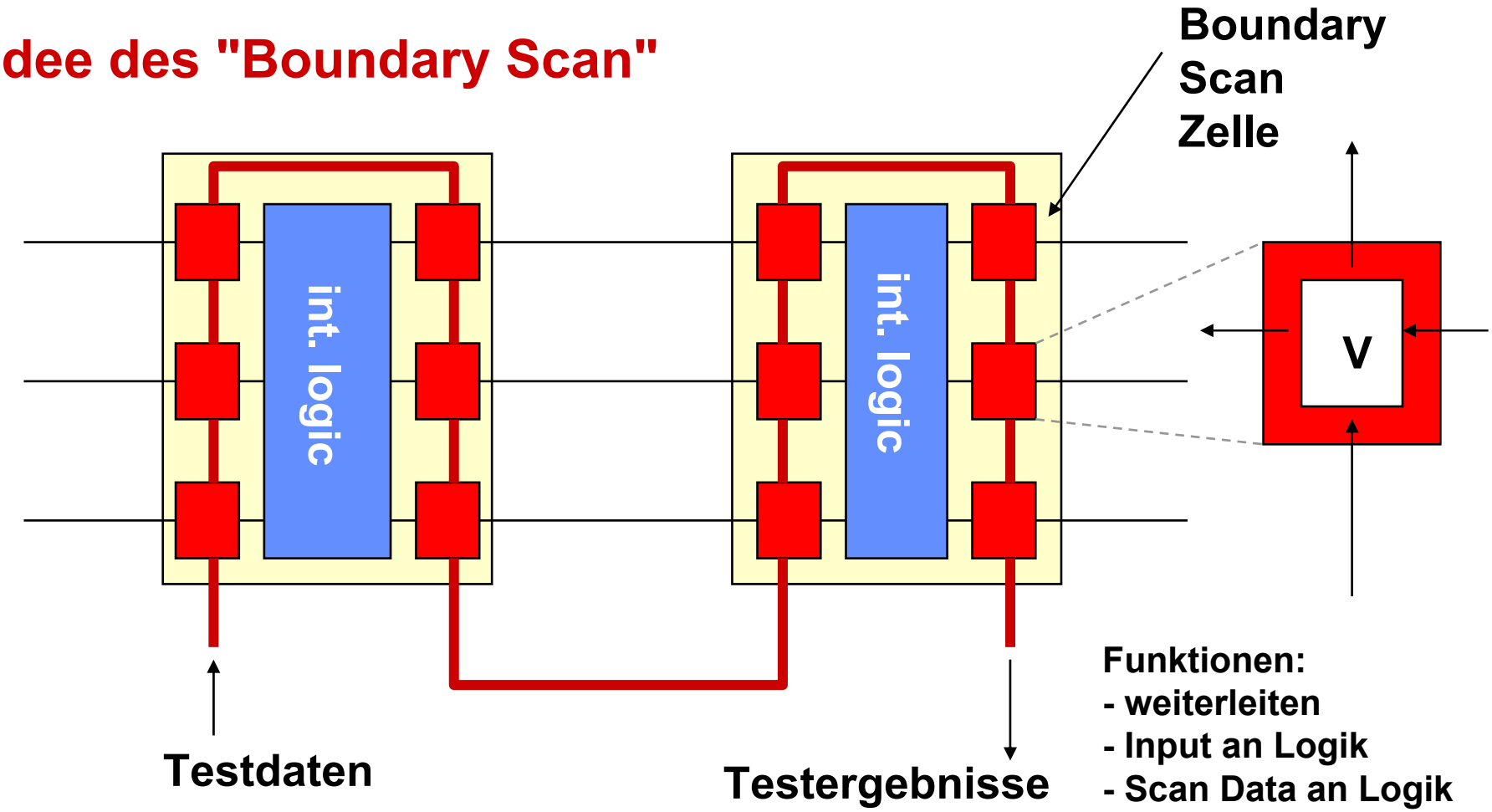
(IEEE 1149)



Wie isoliert man einen Chip für den Test?
Wie bringt man Testmuster an die Eingänge?

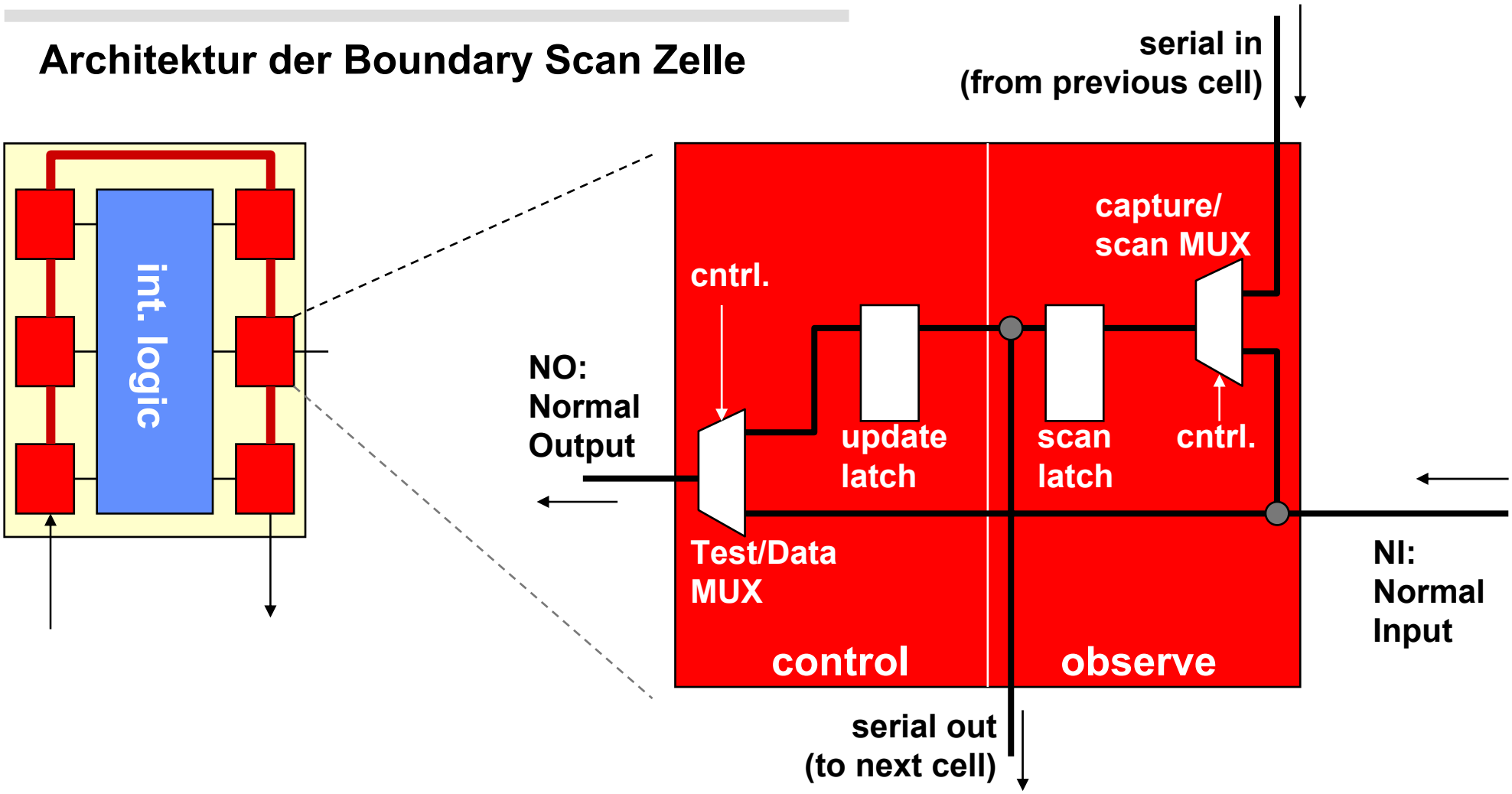
Der JTAG Standard (IEEE 1149)

Die Idee des "Boundary Scan"



Der JTAG Standard (IEEE 1149)

Architektur der Boundary Scan Zelle

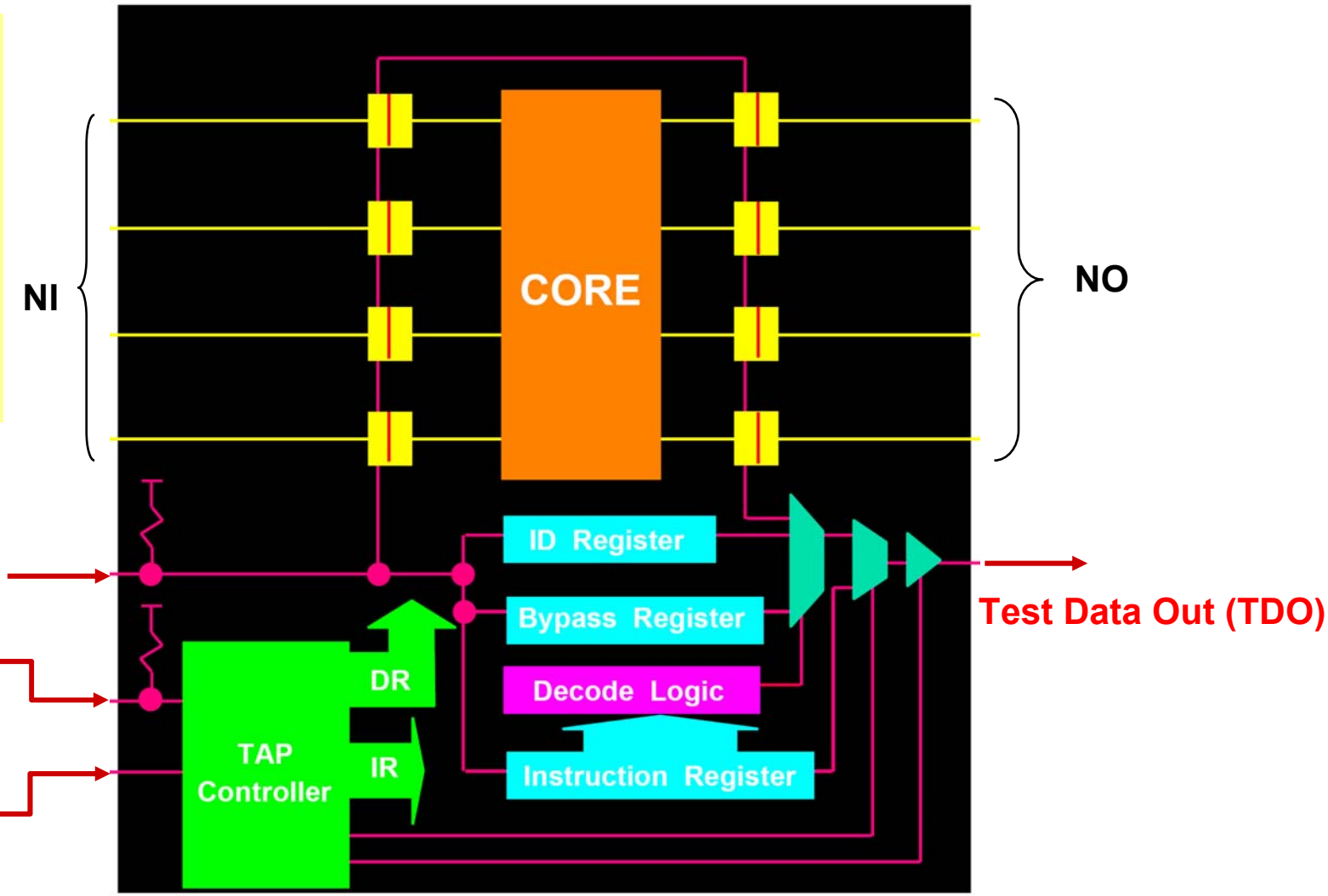


Die Architektur der JTAG Kontrolleinheit

Laden der Test-muster

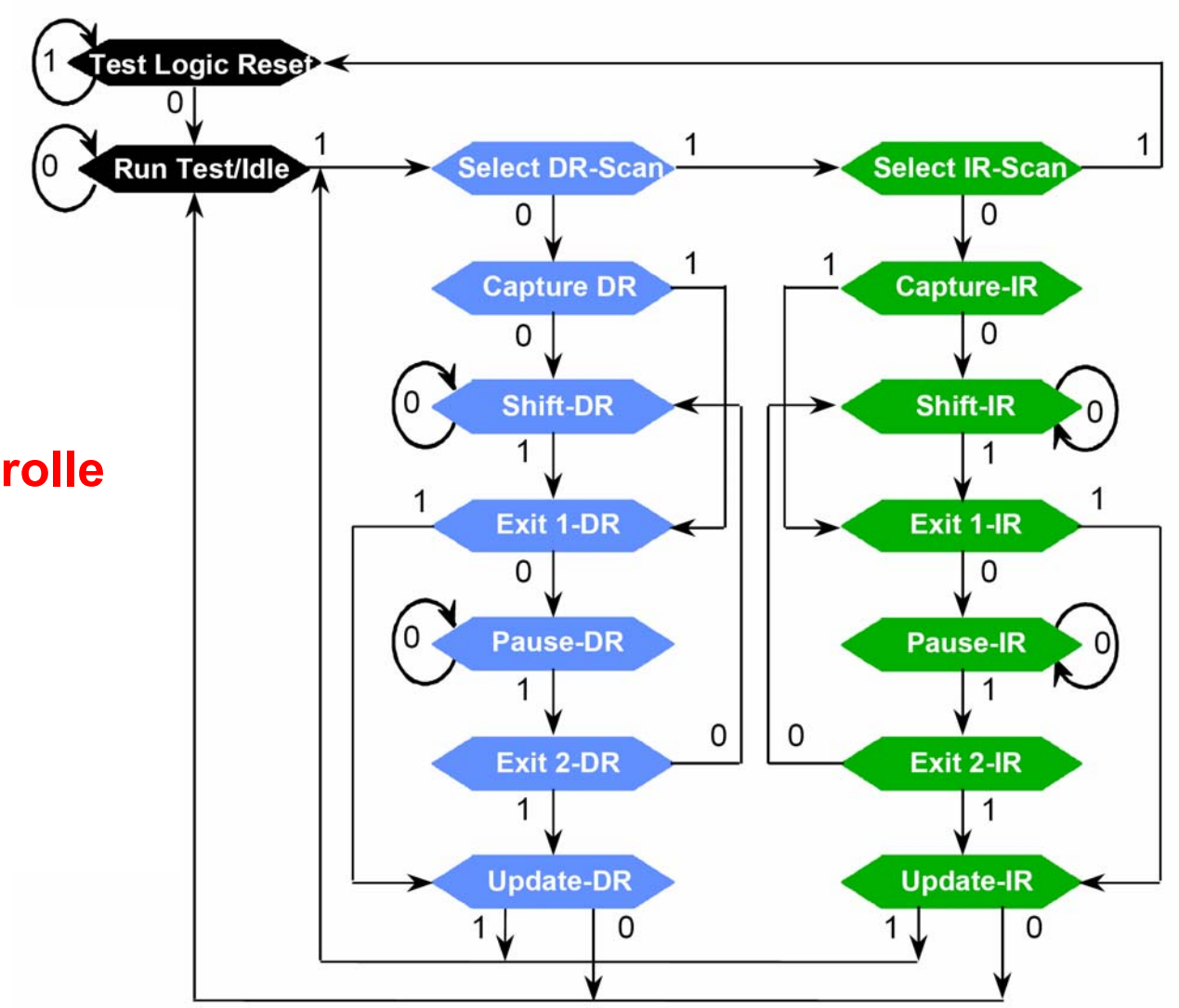
Ausführen der Tests

Verfügbar machen der Resultate

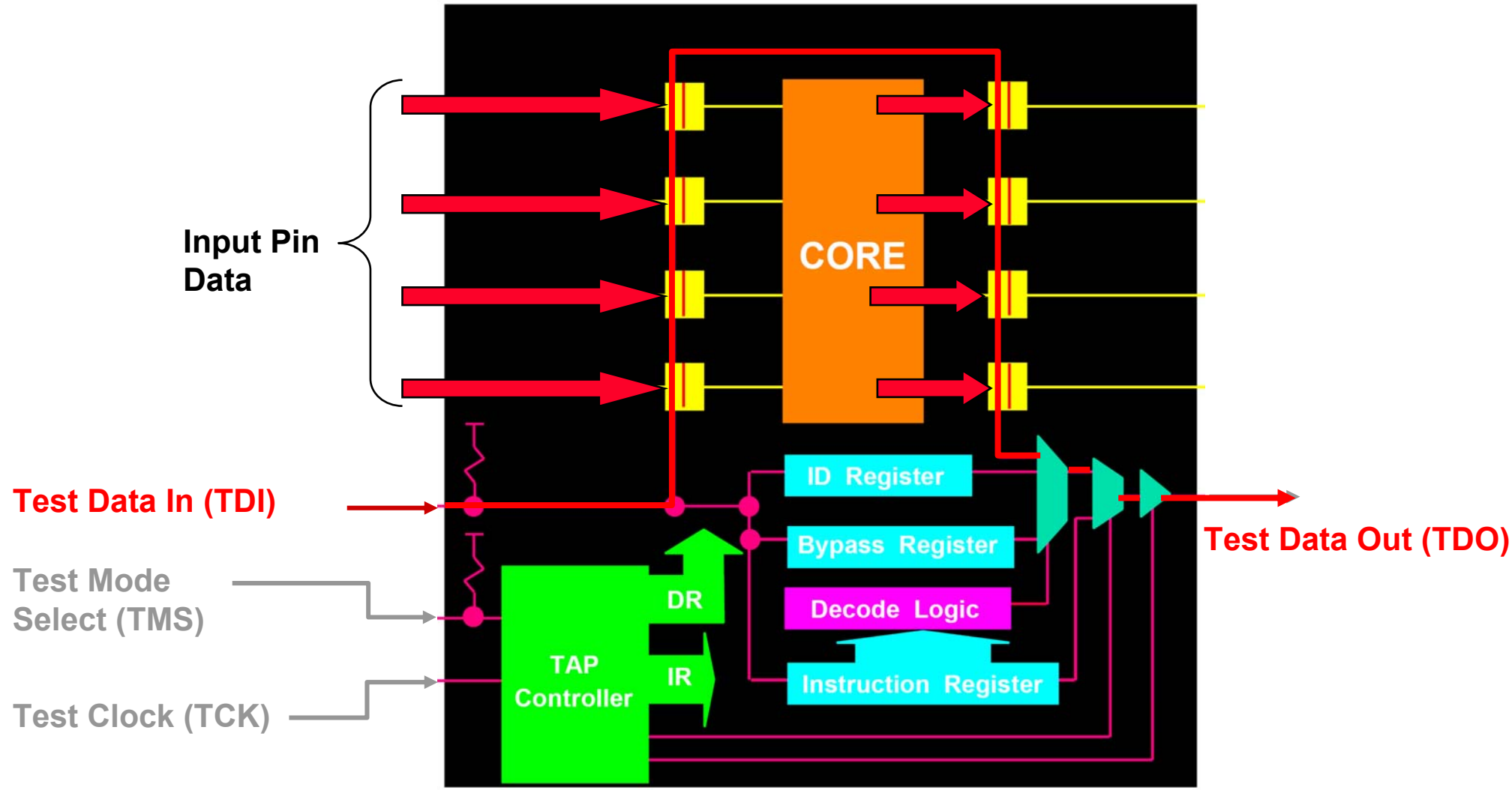


Test Access Port (TAP) Controller

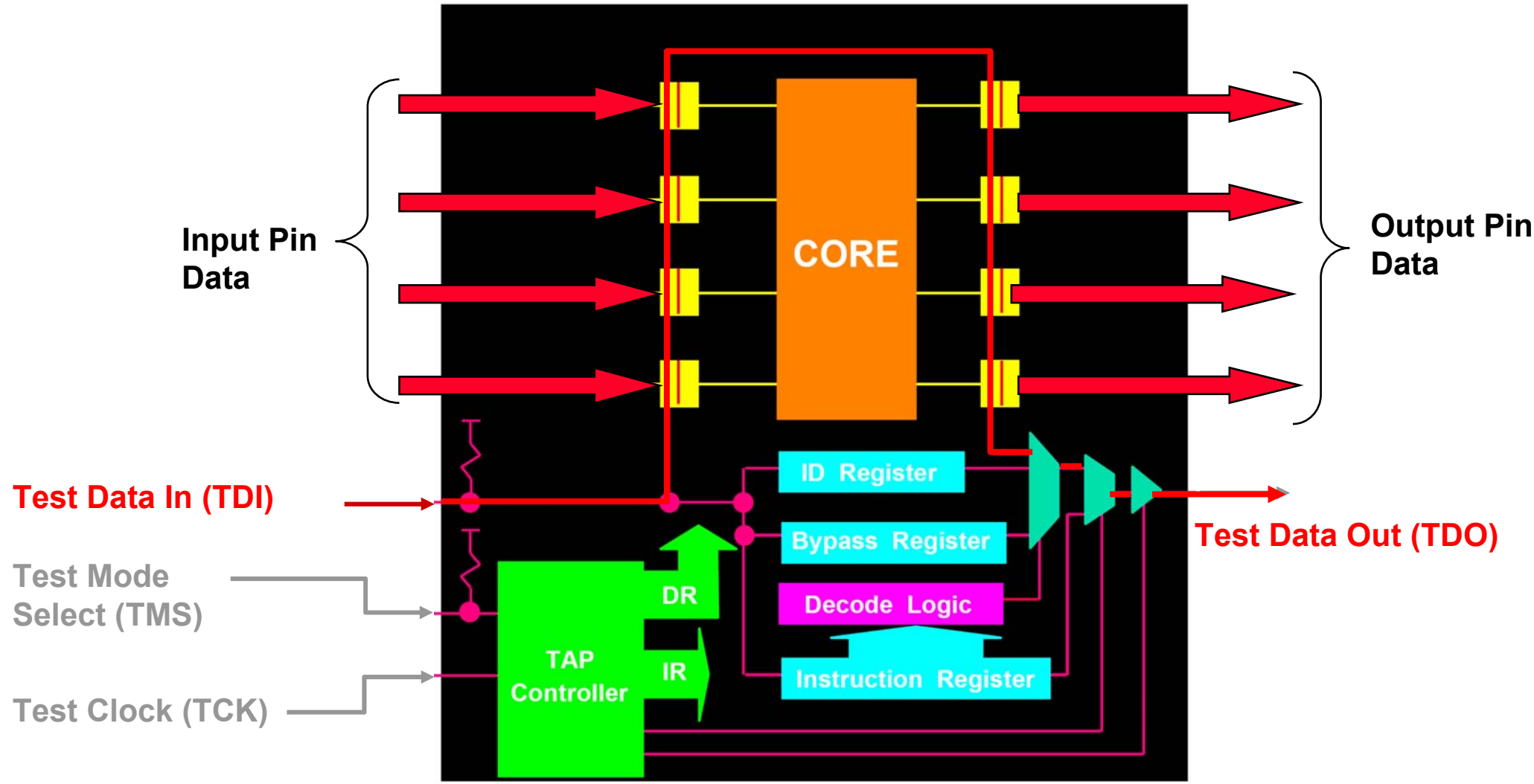
Zustandsautomat zur Kontrolle der Testfunktionen



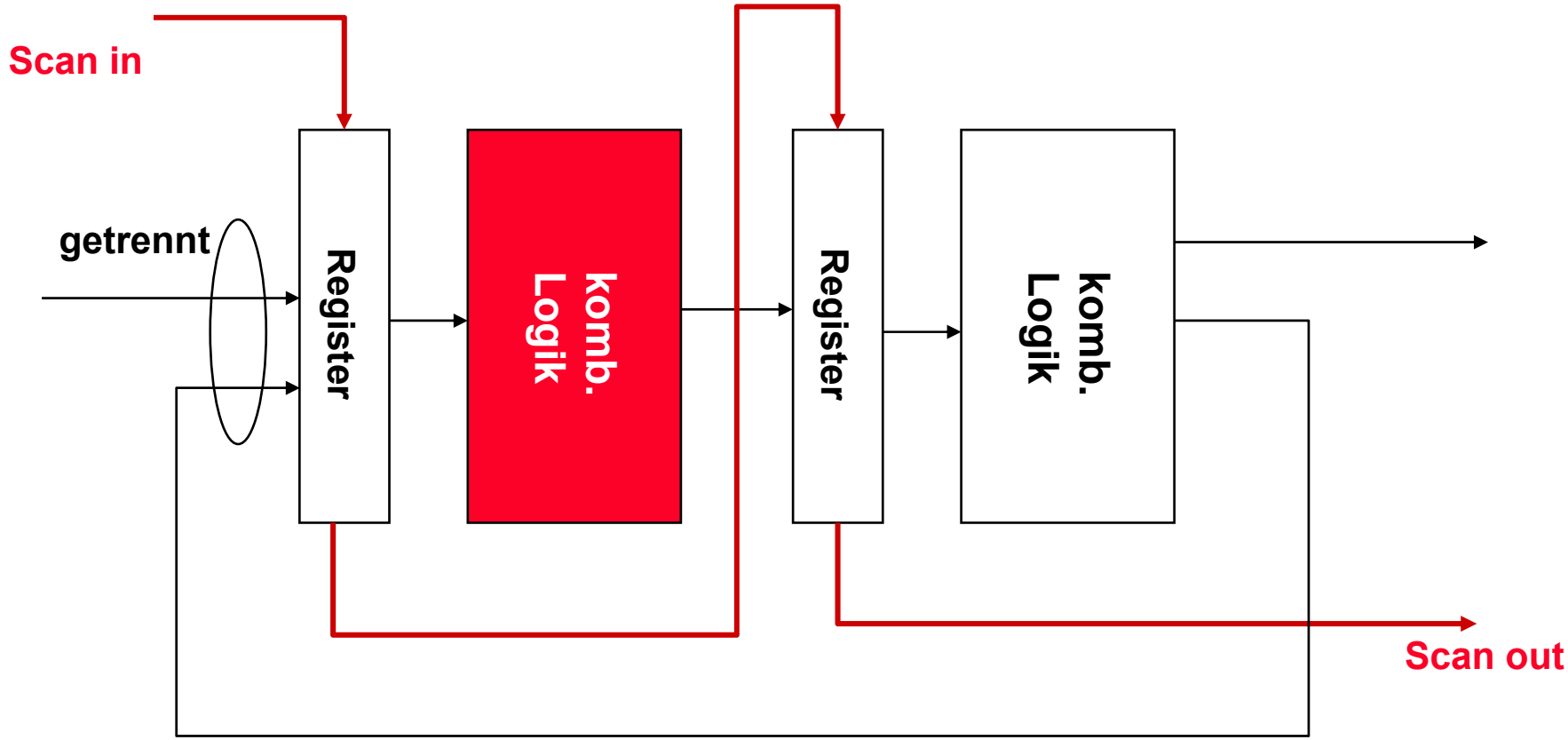
Die "Sample/Preload" Instruktion



Die "Extest" Instruktion



Test der internen Logik: Die Scan Path Idee



HAL: My Fault-Prediction Unit tell me that I will fail within the next 36 hours.....

Wie werden die Ausfallwahrscheinlichkeiten über die Zeit ermittelt ?

Maße der Fehlertoleranz

Lebensdauer T

Zeit vom Beanspruchungsbeginn (DIN 40 042) bis zum Totalausfall (nicht mehr reparierbar)

Ausfallwahrscheinlichkeit F(t)

ist die Wahrscheinlichkeit für eine Komponente bis zum Zeitpunkt $T < t_i$ auszufallen.

Überlebenswahrscheinlichkeit R(t) (Reliability)

Wahrscheinlichkeit, daß eine Komponente zum Zeitpunkt t_i noch nicht ausgefallen ist. F(t) ist das Komplement zu R(t).

$$R(t) = 1 - F(t)$$

Für nicht reparierbare Systeme ist R(t) eine monoton fallende Funktion. $R(0) \leq 1$, $R(\infty) = 0$

Ausfallwahrscheinlichkeitsdichte f(t)

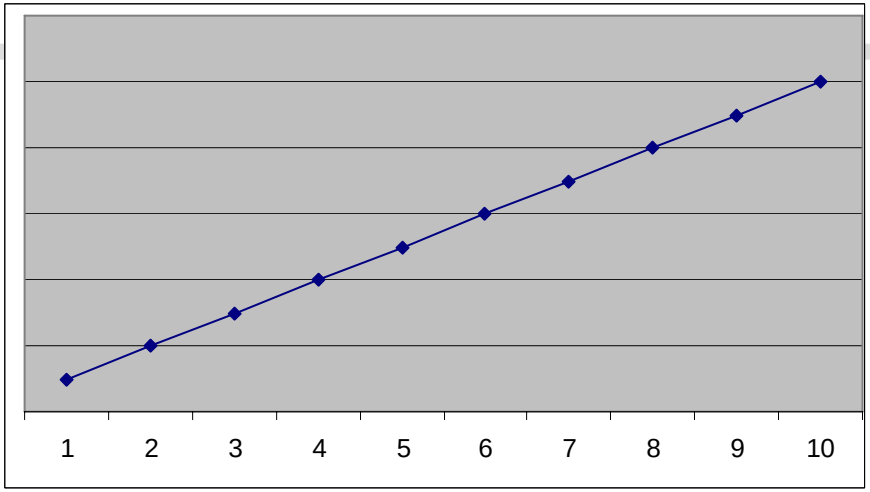
f(t) • dt ist die Wahrscheinlichkeit, daß der Ausfall einer Komponente im Zeitintervall (t, t+dt) auftritt.

f(t) ist dann die Wahrscheinlichkeit, mit der in diesem Zeitintervall Ausfälle erwartet werden können.

$$f(t) = \frac{dF(t)}{dt} = - \frac{dR(t)}{dt}$$

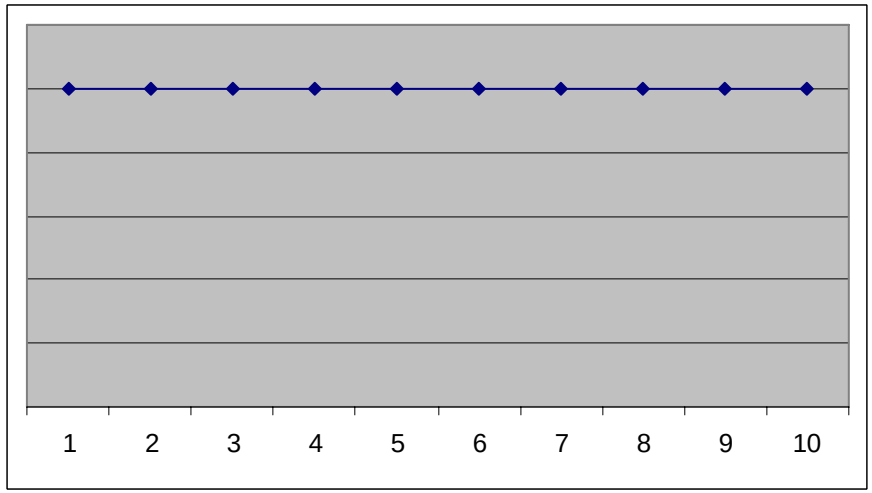
Konstante Ausfallwahrscheinlichkeitsdichte

F(t)

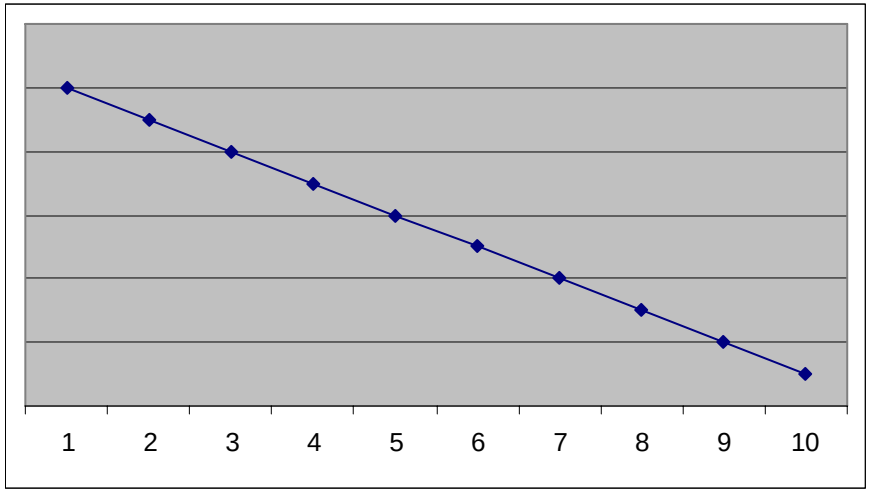


$$f(t) = \frac{dF(t)}{dt} = - \frac{dR(t)}{dt}$$

f(t)

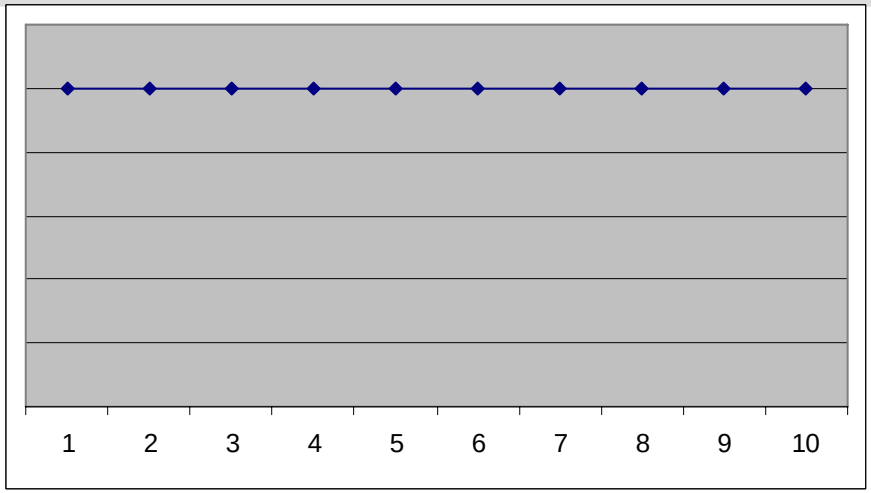


R(t)

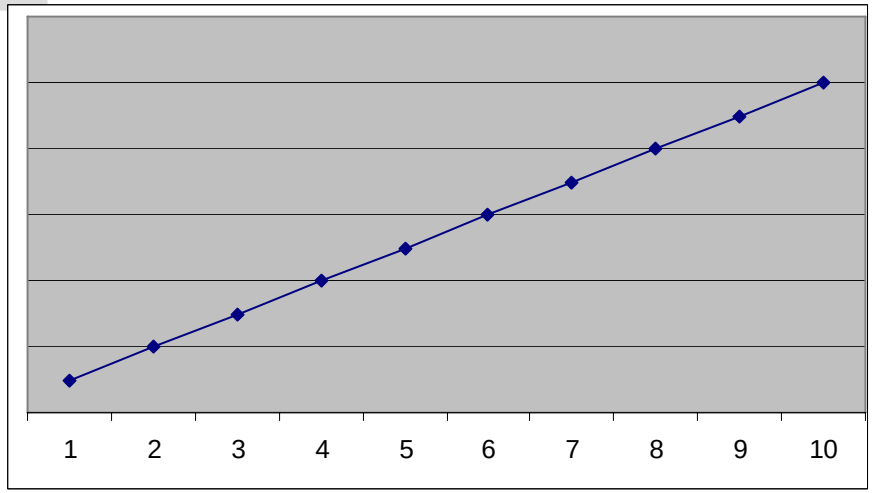


Konstante Ausfallwahrscheinlichkeitsdichte

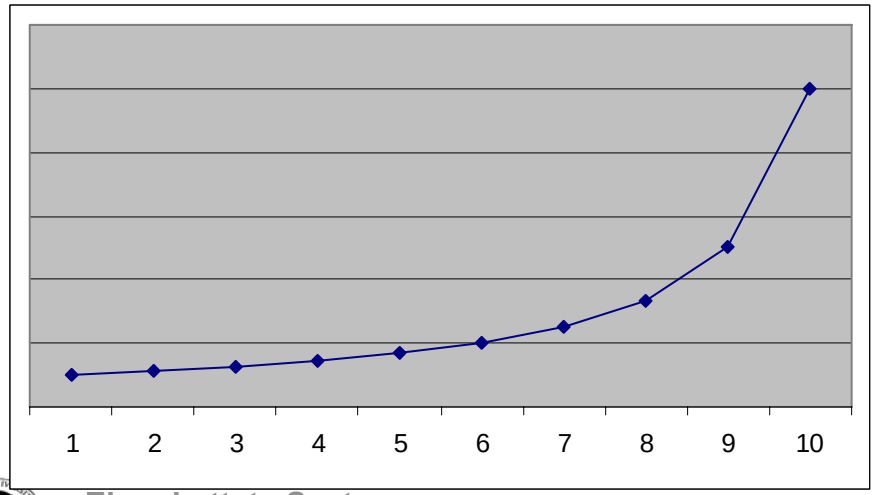
f(t)



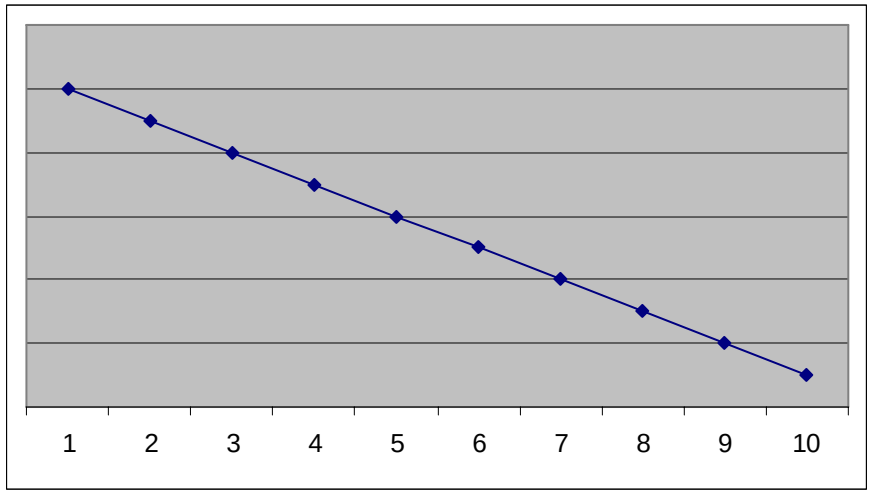
F(t)



Ausfallrate: λ

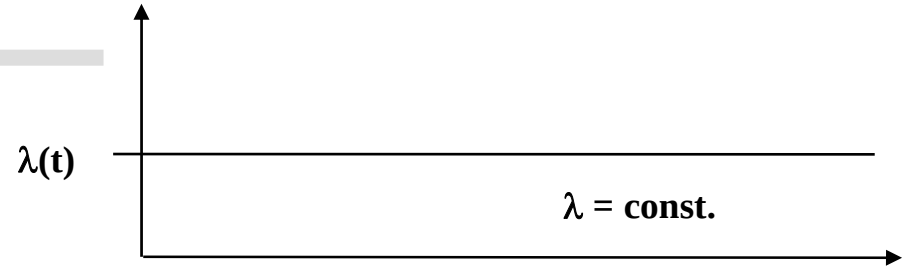


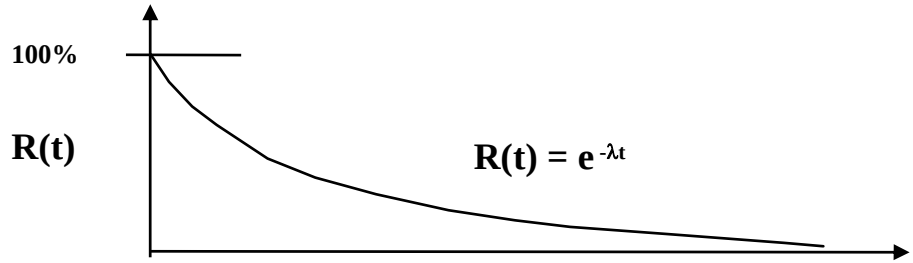
R(t)



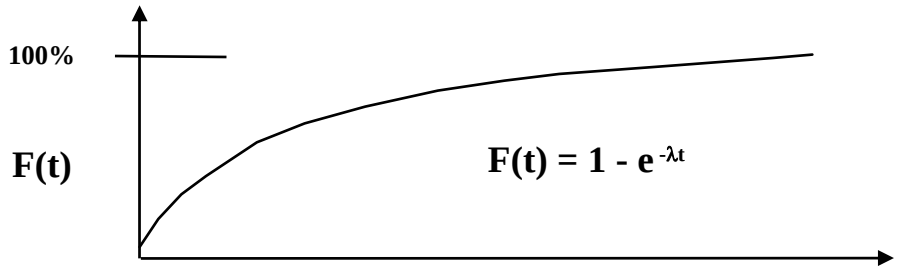
Maße der Fehlertoleranz

Ausfallrate λ (t)
Anzahl der Ausfälle pro Zeiteinheit

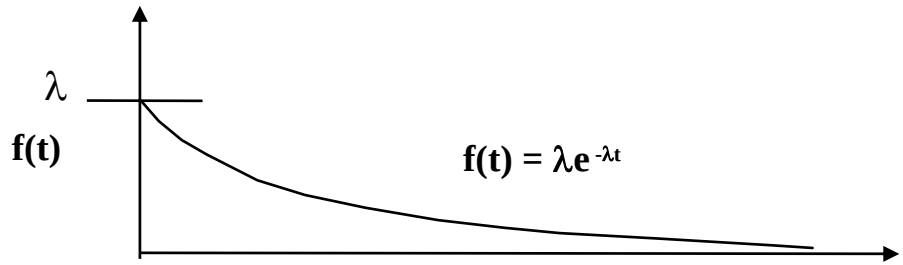




Bemerkung: Die Ausfallrate ist relativ zum Bestand definiert. Fallen pro Zeiteinheit immer gleich viele Komponenten aus, steigt die Ausfallrate relativ zum Bestand an, der ja immer kleiner wird.

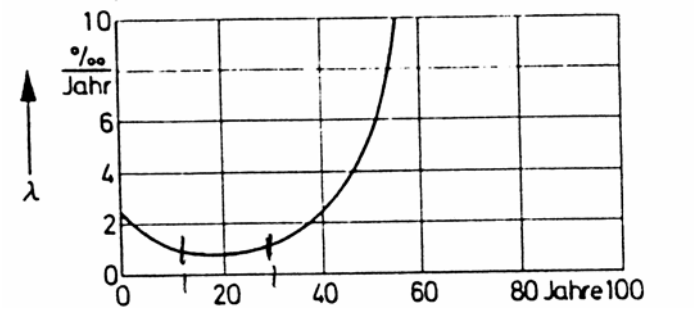


Bleibt die Ausfallrate relativ zum Bestand konstant, ergibt sich daraus eine Exponentialverteilung für die Überlebenswahrscheinlichkeit R(t).

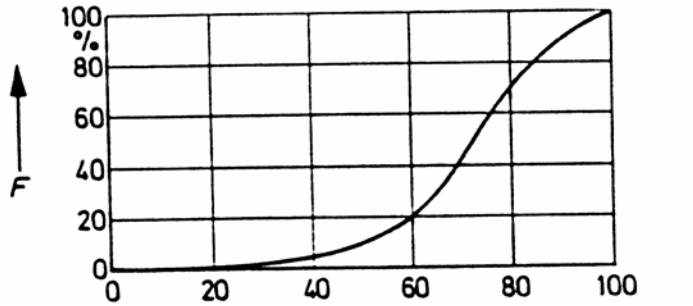


Lebensdauerverteilung beim Menschen

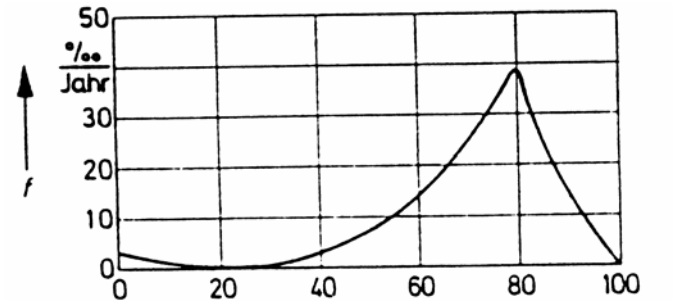
Ausfallrate λ (t)



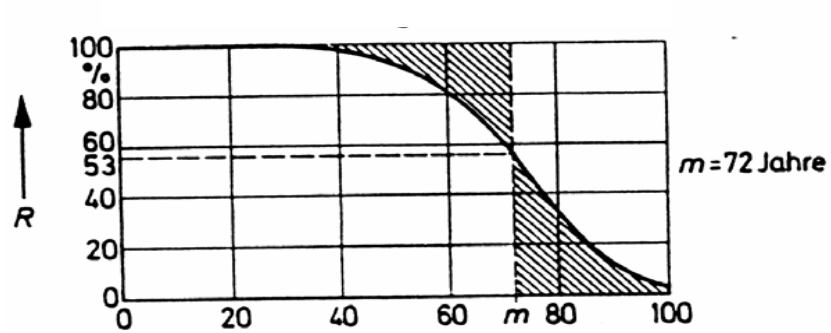
Ausfallwahrscheinlichkeit F(t)



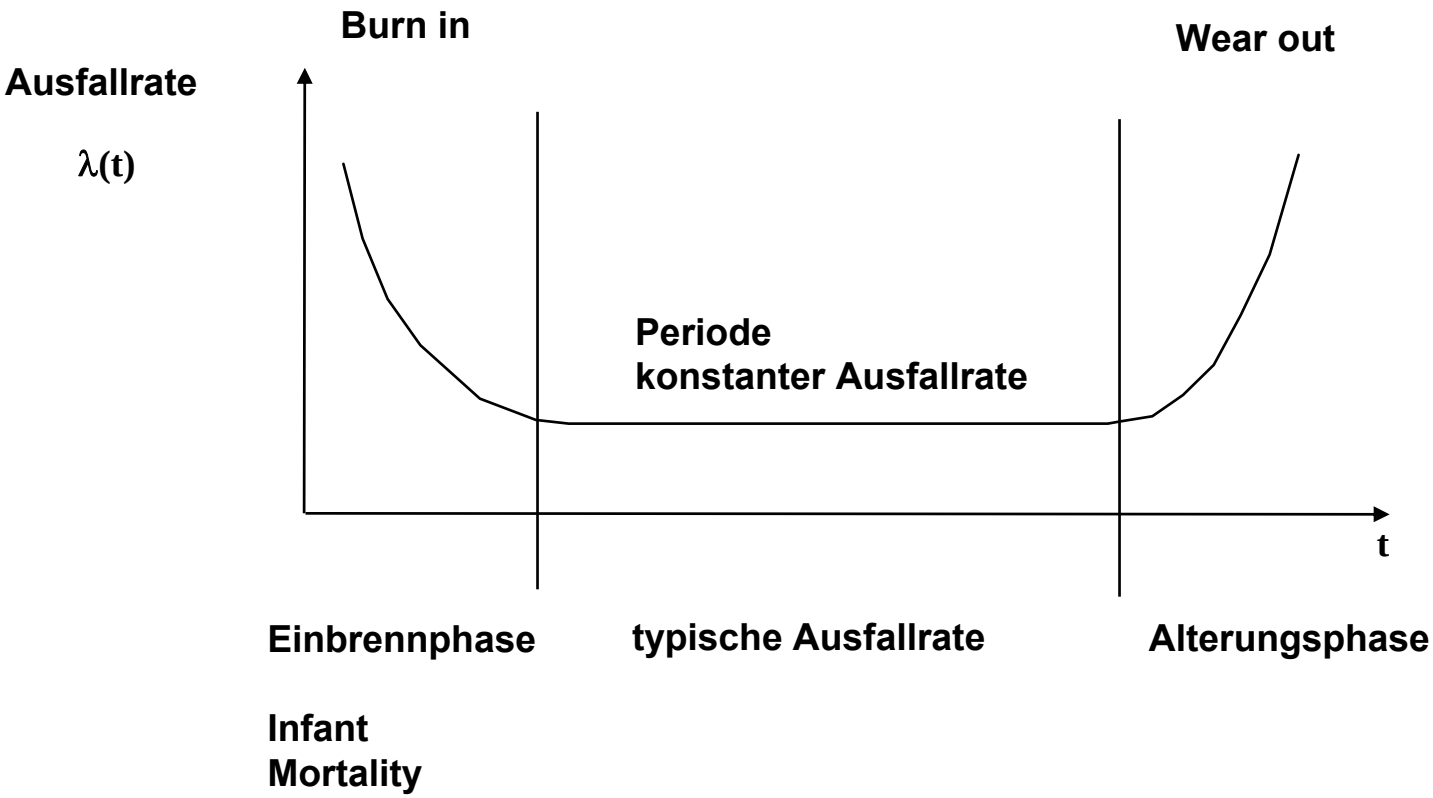
Ausfallwahrscheinlichkeitsdichte f(t)



Überlebenswahrscheinlichkeit R(t) (Reliability)



Ausfallrate über die Zeit



Typische Ausfallrate:
VLSI-Chip: 10^{-8} Ausfälle/h = 1 Ausfall in 115000 Jahren

Kenngrößen (Zusammenfassung)

Kenngröße	Symbol	Einheit
Lebensdauer	T	h
Ausfallwahrscheinlichkeit	F	%
Überlebenswahrscheinlichkeit	R	%
Ausfallwahrscheinlichkeitsdichte	f	%/h
Ausfallrate	λ	1/h

Da die Ausfallrate über die Zeit als konstant angenommen wird, lohnt sich häufiges Testen. Je kürzer das Intervall zwischen den Tests, desto geringer die Wahrscheinlichkeit, dass das System in diesem Intervall ausfällt!

Maße der Fehlertoleranz

Unter der Annahme von $\lambda(t) = \text{const.}$ gilt:

$$\frac{1}{\lambda} = \text{MTBF} = \text{MTTFF} = \text{MTTF}$$

MTBF : Mean Time Between Failures

MTTFF: Mean Time To First Failure

MTTF : Mean Time To Failure

Verfügbarkeit (Availability)

Zeit in der ein System intakt ist bezogen auf die gesamte Missionszeit

$$A = \frac{\text{U (Up time)}}{\text{M (Mission time)}}$$

$$\text{M} = \text{U} + \text{TR (Repair time)}$$

$$A = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

Maße der Fehlertoleranz

Beispiele:

- **Telefonvermittlungssysteme** — **Klasse 5**
  Nichtverfügbarkeit 2h/a bis 3 Min /a

  Aber: in den letzten Jahren waren mehrere große Ausfälle:
  1 USA-nationsweiter Fehler: 8h
  1 Midwest: 4 Tage

- **Starkstromüberwachung** — **Klasse 4**
  Nichtverfügbarkeit typ.: 2h/a

- **AAS (Advanced Automation System) IBM**
  3 sek/a für kritische Dienste (A = 0,9999999) — **Klasse 7++**
  156 sek/a für weniger kritische Dienste (A = 0,9999950) — **Klasse 5**

Prozeßsicherheit (Betriebssicherheit, Safety)

- **ist die Überlebenswahrscheinlichkeit in Beziehung zu kritischen (Funktions-) Ausfällen**

- **ist die Realisierung eines bestimmten Verhaltens beim Auftreten bestimmter Fehler (Ausschuß Steuerungstechnik VDE/VDI). Grundlage dafür ist eine Sicherheitsvereinbarung, die einen Fehlerkatalog und einen Verhaltenskatalog spezifiziert.**

- **ist die Wahrscheinlichkeit, dass das System ein bestimmtes antizipiertes Verhalten zeigt, d.h. wenn es von einem kritischen Funktionsausfall betroffen ist, in einen Zustand überführt werden kann, in dem der kritische Funktionsausfall sich nicht katastrophal auswirken kann.**

Der Sicherheitsgrad kann angegeben werden mit:

$$S = 1 - \frac{k}{u+k}$$

S : Sicherheitsgrad
k: Anzahl der kritischen Funktionsausfälle
u: Anzahl der unkritischen Funktionsausfälle

Ein Funktionsausfall ist kritisch, wenn seine Konsequenzen die normalen Installations- und Betriebskosten eines Systems bei weitem (mehrere Größenordnungen) übersteigen.

Maße der Fehlertoleranz

Verfügbarkeit: Einteilung in Systemklassen

Klasse: $\lfloor \log_{10} (1/(1-A)) \rfloor$

1 Jahr = 525600 Minuten = 8760 h

Systemtyp	Nicht-Verfügbarkeit Minuten/Jahr	Verfügbarkeit %	Klasse
Nicht verwaltete Systeme	50 000	~ 90	1
Verwaltete Systeme	5 000	99	2
Gut verwaltete Syst.	500	99,9	3
Fehlertolerante Syst.	50	99,99	4
Hochverfügbare Syst.	5	99,999	5
Sehr hochverf. Syst.	0,5	99,9999	6
Ultra-hochverf. Syst.	0,05	99,99999	7

Organisation der redundanten Komponenten

Arten der Redundanz:

- **Komponentenredundanz**
- **Zeitredundanz**

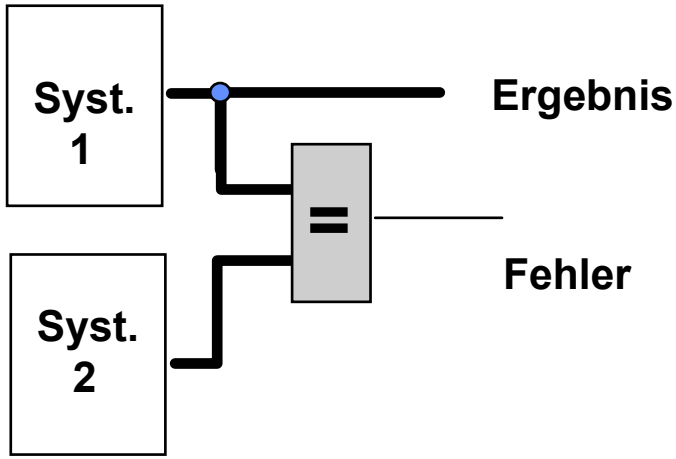
- **Aktive Redundanz:** **Mehrere Komponenten erbringen dieselbe Dienstleistung nebenläufig.**

- **Passive Redundanz:** **Redundante Komponenten sind nicht an der Erbringung der Dienstleistung beteiligt.**

- **Cold Standby:** **die redundante(n) Komponente(n) werden erst aktiviert, wenn eine aktive Komponente ausgefallen ist. Der Zustand der Berechnung zum Zeitpunkt des Ausfalls der aktiven Komponente muss auf der redundanten Komponente rekonstruiert werden.**

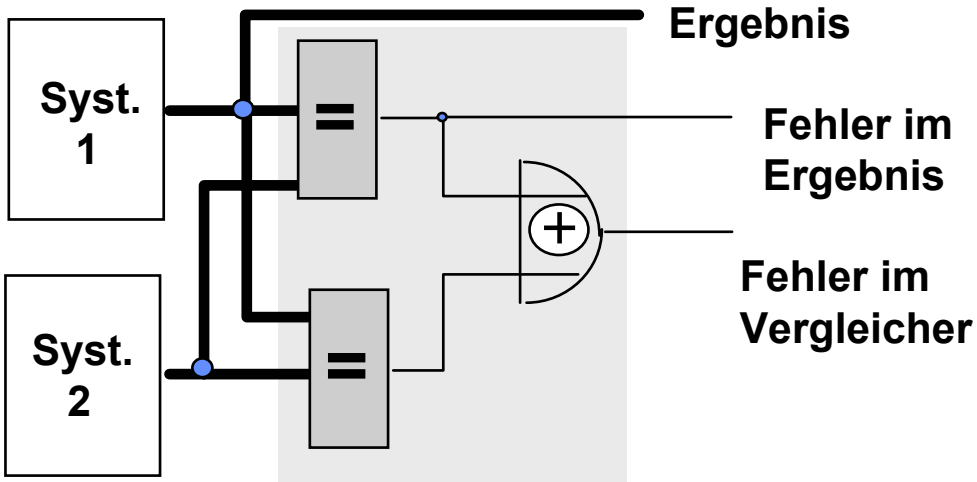
- **Hot Standby:** **die redundante(n) Komponente(n) ist aktiv, erzeugt aber keine Ausgaben. Die redundante Komponente enthält beim Ausfall der aktiven Komponente bereits deren Zustand und kann sie sofort ersetzen.**

Organisation der redundanten Komponenten

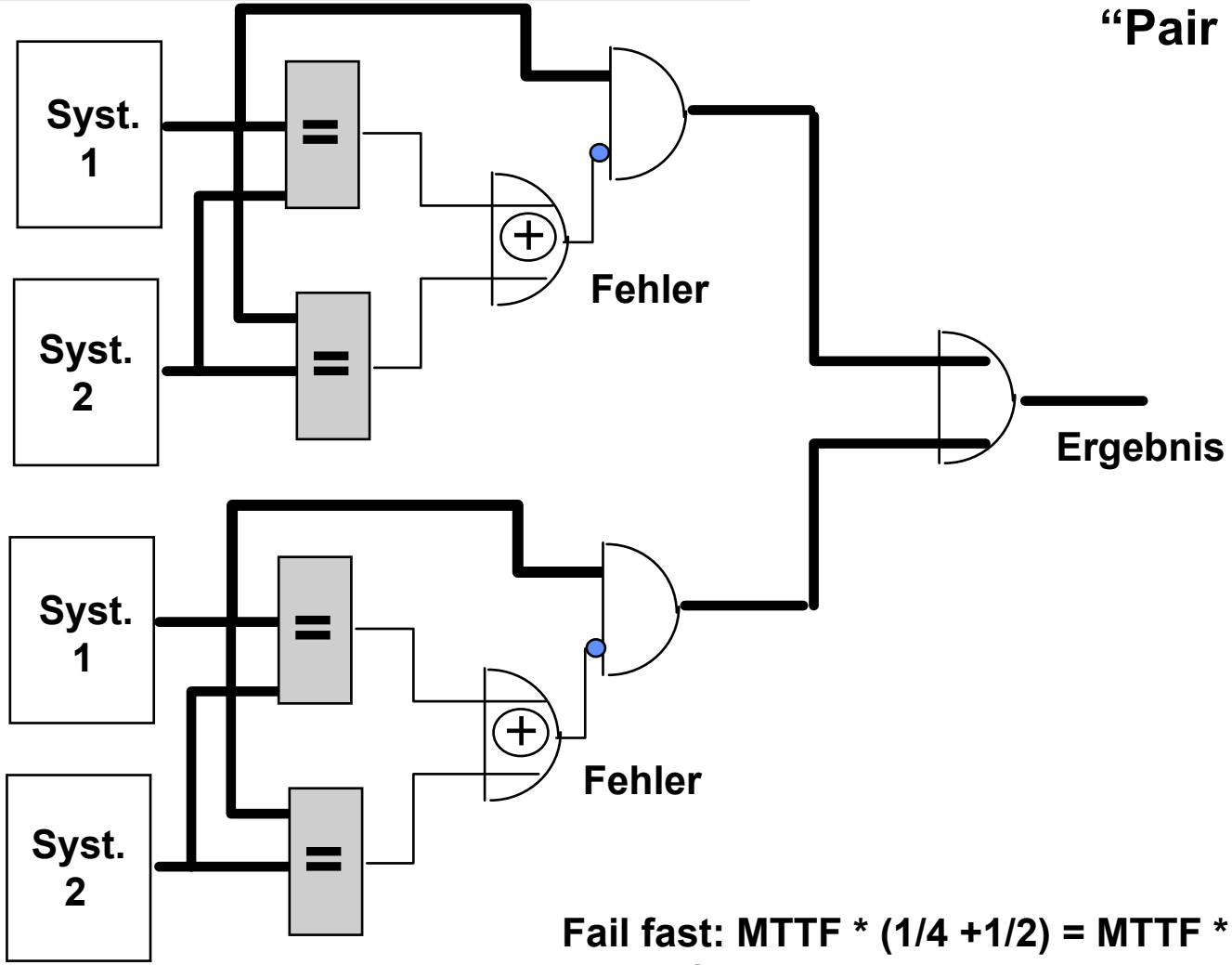


MTTF für ein Modul ist: 1* MTTF
Für ein System mit 2 Moduln gilt dann:
MTTF / 2.

Fail Fast: MTTF/2
Fail Soft: MTTF *(1+1/2) = MTTF * 3/2



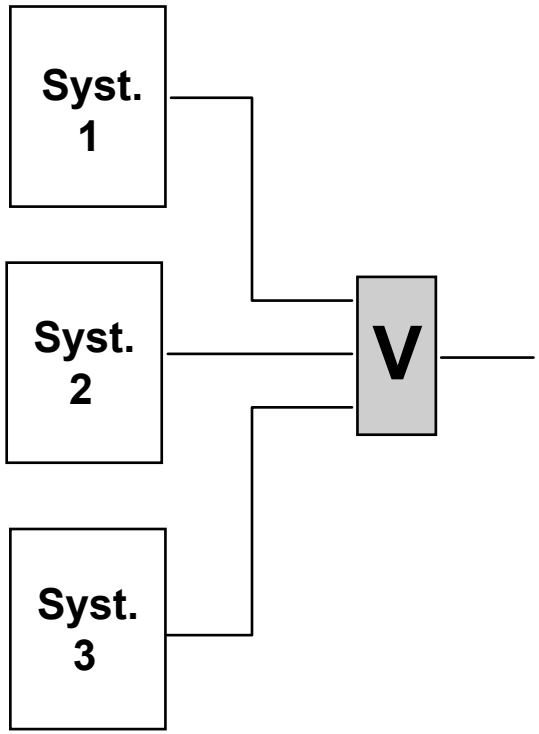
Organisation der redundanten Komponenten

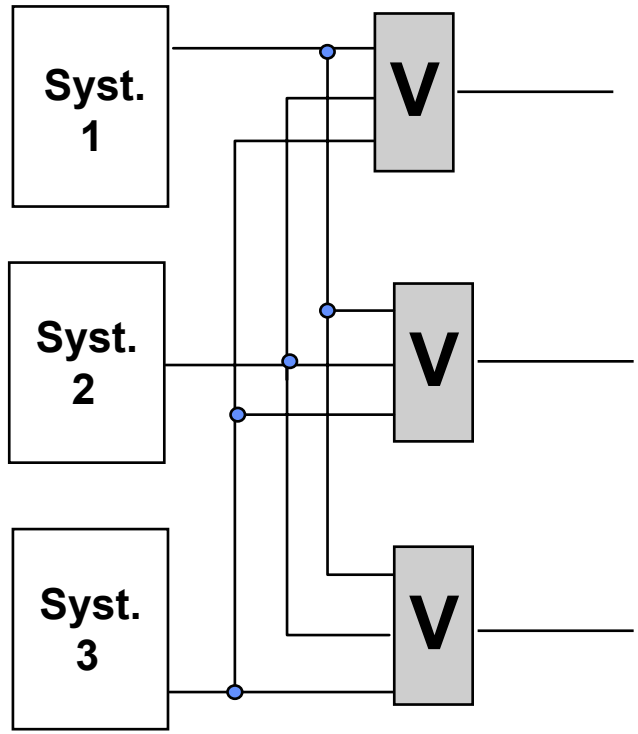


Fail fast: MTTF * (1/4 +1/2) = MTTF * (3/4)
Fail soft: MTTF* ((3/4) + (2/4) + 1) = MTTF * 2,25

Organisation der redundanten Komponenten

Triple Modular Redundancy (TMR)





Fail Fast: MTTF/3 + MTTF/2 = MTTF (2/6) + MTTF (3/6) = MTTF(5/6)
Fail Soft: MTTF (1 + 5/6)

Organisation der redundanten Komponenten

Annahme: MTTF = 1 Jahr

Organisation	MTTF (Jahre)	Klasse	Gleichung	Kosten
Simplex	1	3	MTTF * 1	1
Duplex (FF)	~ 0,5	3	MTTF/2	2 + ε
Duplex (FS)	~ 1,5	3	MTTF(3/2)	2 + ε
TMR (FF)	~ 0,8	3	MTTF(5/6)	3 + ε
TMR (FS)	~ 1,8	3	MTTF(1+5/6)	3 + ε
Pair&Spare (FF)	~ 0,75	3	MTTF(3/4)	4 + ε
Pair&Spare (FS)	~ 2,25	3	MTTF((3/4) + (2/4) + 1)	4 + ε

FF: Fail fast
FS: Fail soft

Quelle: J.Gray