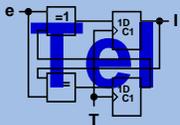


# Konzeption eines „Immunsystems“ für zukünftige OC-Systeme

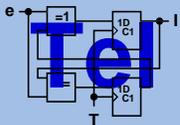
Frank Sakowski

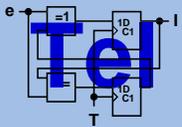
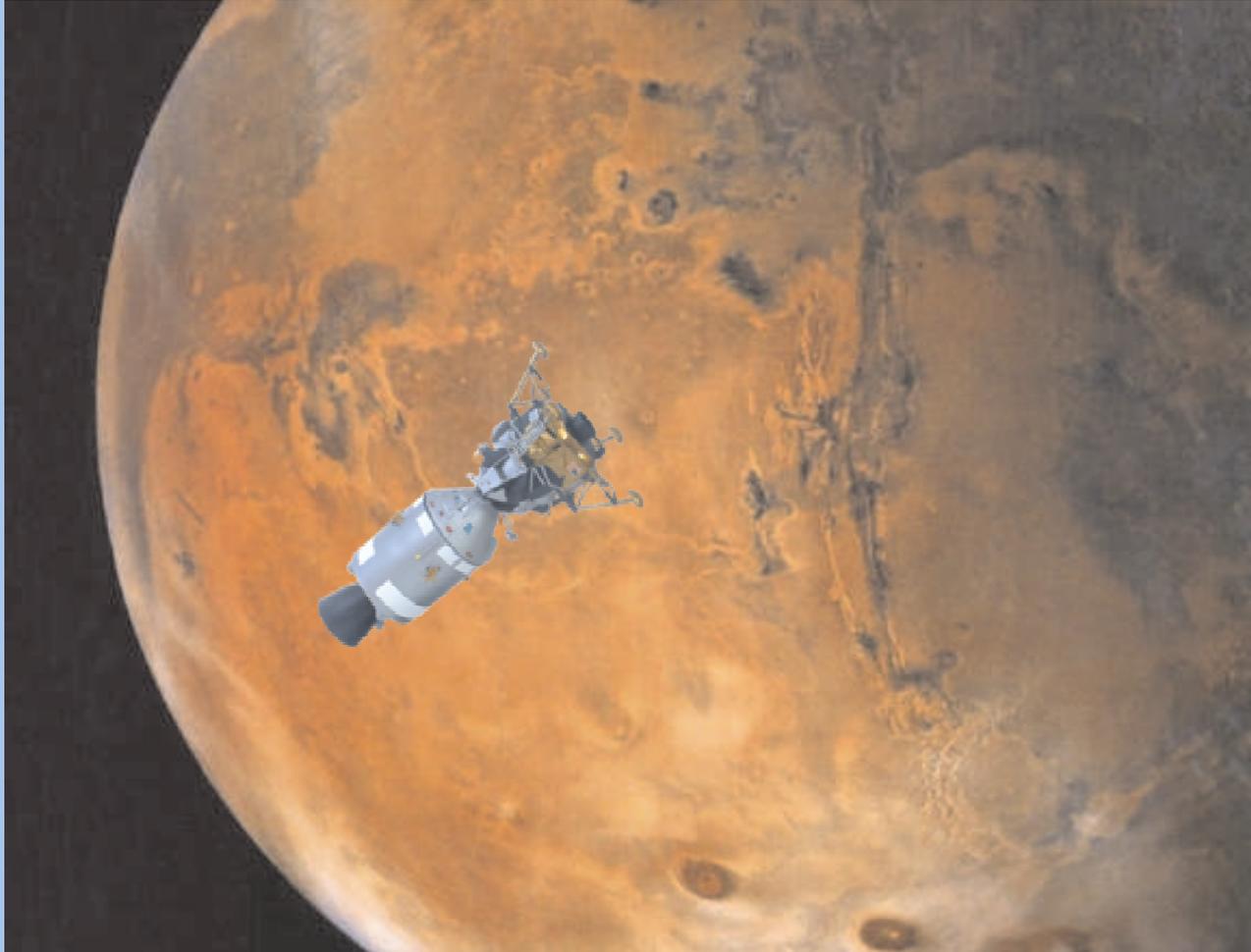
`Frank.Sakowski@mail.inf.tu-dresden.de`

Technische Universität Dresden  
Fakultät Informatik  
Institut für Technische Informatik

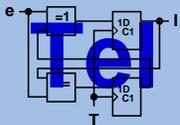


- ❑ Einführung
  - Motivation, Aufgabe
- ❑ Vorbetrachtung
  - Gefahren, Methoden zur Erhöhung der Zuverlässigkeit, Der Nutzen der Methoden
- ❑ Umsetzung von Störungstoleranz
  - Klassische Techniken, POE-Modell, Immunotronics, Embryonics, Embryonics+Immunotronics, EHW
- ❑ Die Bedeutung von FPGA
- ❑ Anwendung in einem „Immunsystem“
- ❑ Zusammenfassung
- ❑ Ausblick
- ❑ Diskussion





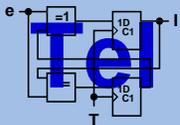
- ❑ Identifizierung von Möglichkeiten zur Umsetzung eines „Immunsystems“
- ❑ Untersuchung klassischer Techniken und vor allem naturinspirierter OC-Techniken
- ❑ Empfehlung für die Umsetzung eines „Immunsystems“
- ❑ Beschränkung auf Hardware



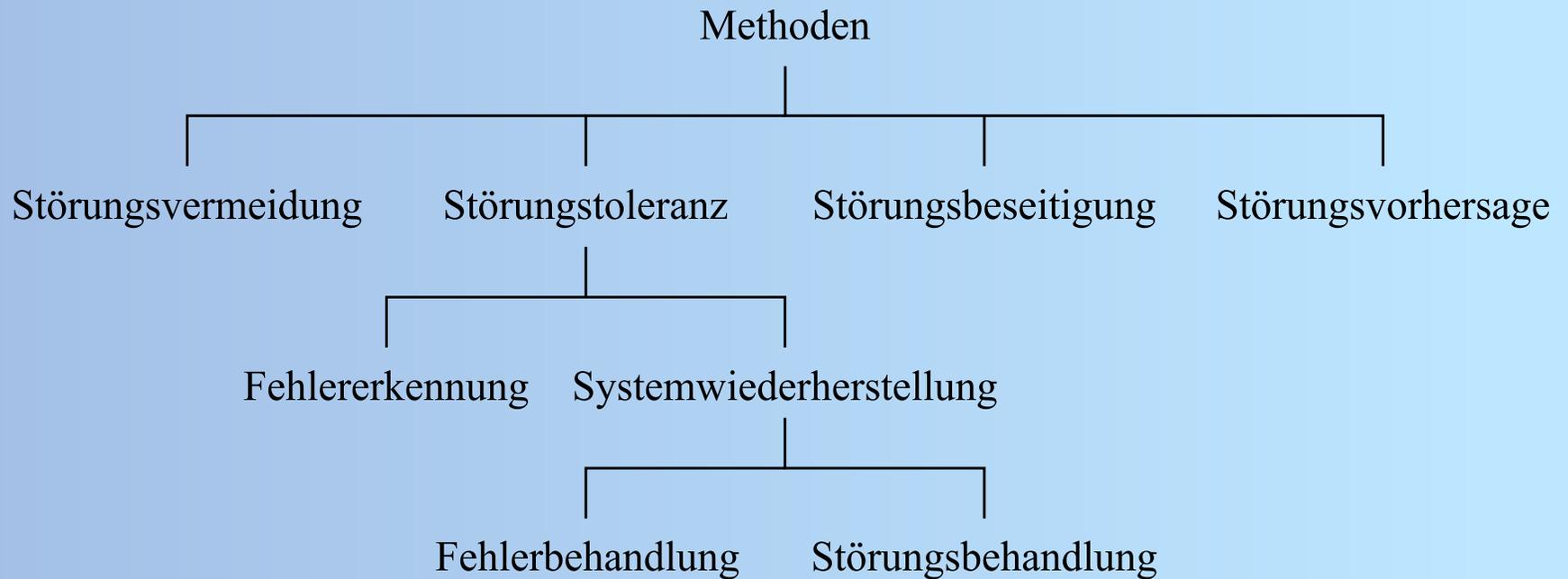


nach Avizienis u.a.: *Basic concepts and taxonomy of dependable and secure computing*. 2004

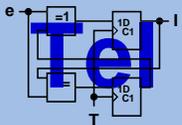
- Bsp.: Leitungsschluss zur Masse
  - „0“ auf der Leitung
  - UND-Gatter versagt

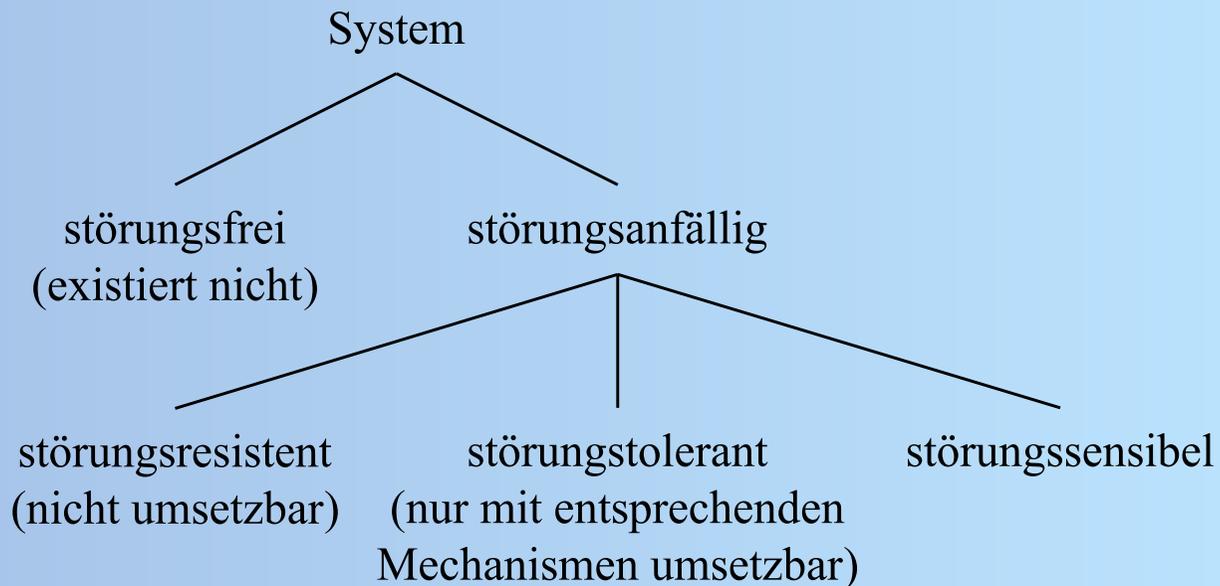


# Methoden zur Erhöhung der Zuverlässigkeit

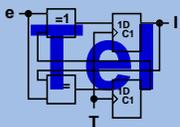


nach Avizienis u.a.: *Basic concepts and taxonomy of dependable and secure computing*. 2004



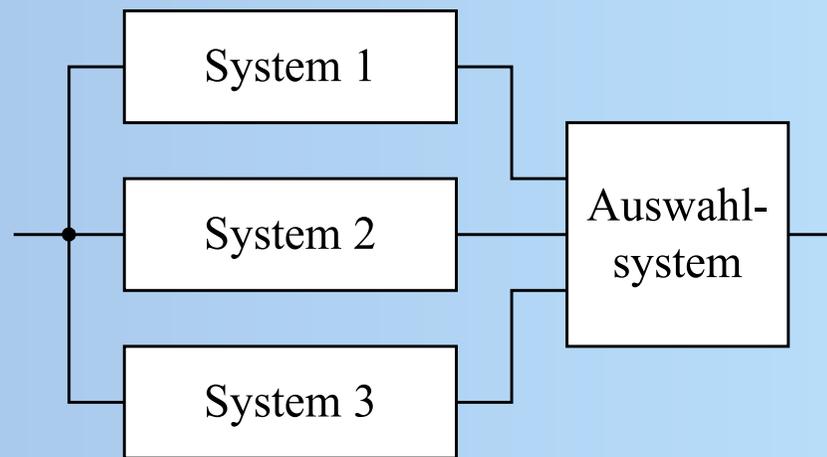


⇒ „Immunsystem“ ist Störungstoleranzsystem

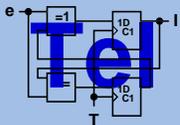


# Umsetzung von Störungstoleranz

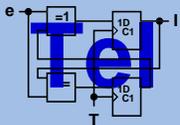
- inwiefern erfüllen klassische und naturinspirierte Techniken die Methoden der Störungstoleranz
- auf welcher Abstraktionsebene arbeiten die Techniken
- welche Besonderheiten weisen sie auf

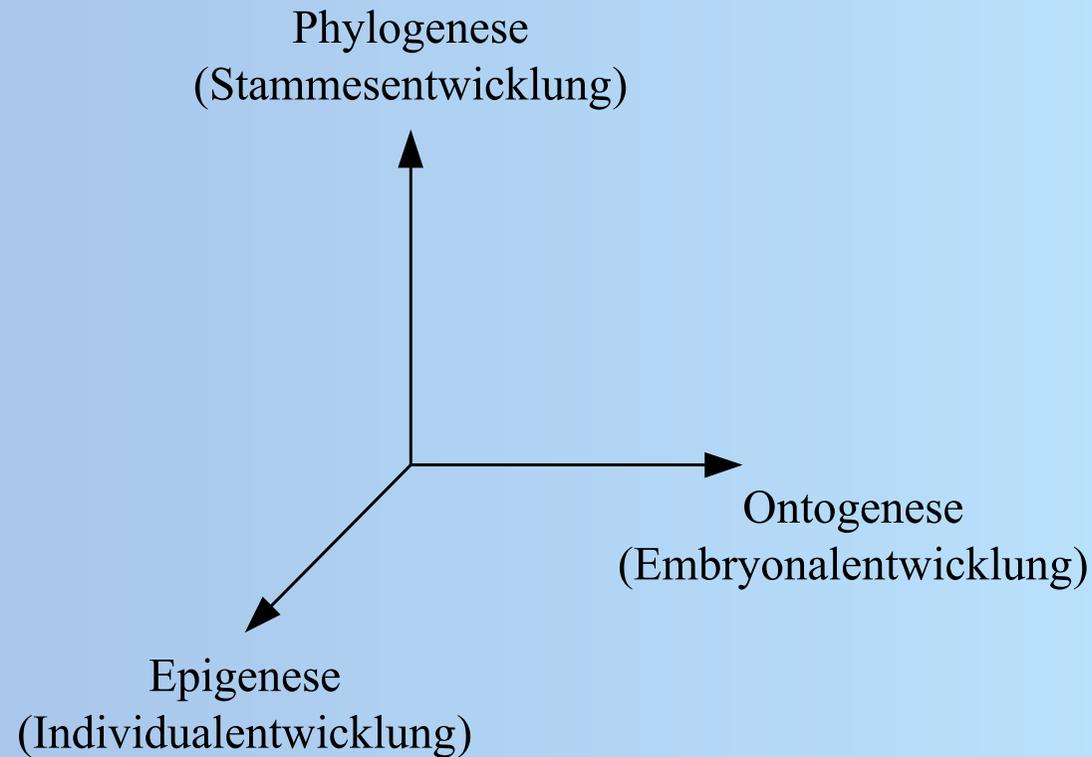


klassische dreifach-modulare Redundanz

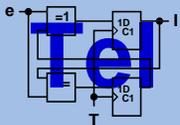


- ❑ Fehlererkennende/Fehlerkorrigierende Codes
  - nebenläufige Fehlererkennung, Logikebene
- ❑ Watchdog
  - nebenläufige Fehlererkennung, Architekturebene
- ❑ Pair-and-Spare
  - vollständige Störungstoleranz, Architekturebene
- ❑  $N$ -modulare Redundanz, Entwurfsvielfalt
  - systematische Kompensation (Maskierung), Logik- und Architekturebene
- ❑ Selbstüberprüfende Komponenten
  - vorsorgende Fehlererkennung, Logikebene

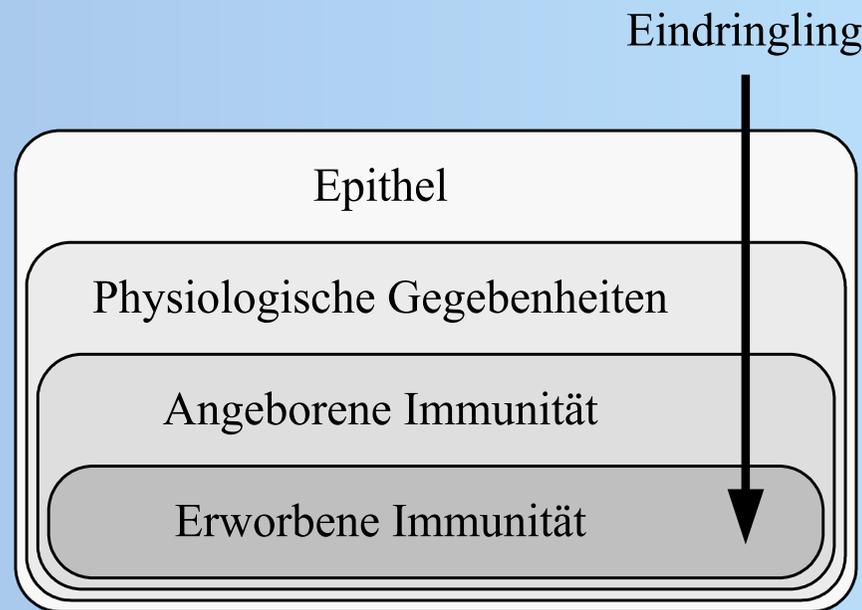




nach Sipper u.a.: *A phylogenetic, ontogenetic, and epigenetic view of bio-inspired hardware systems.* 1997

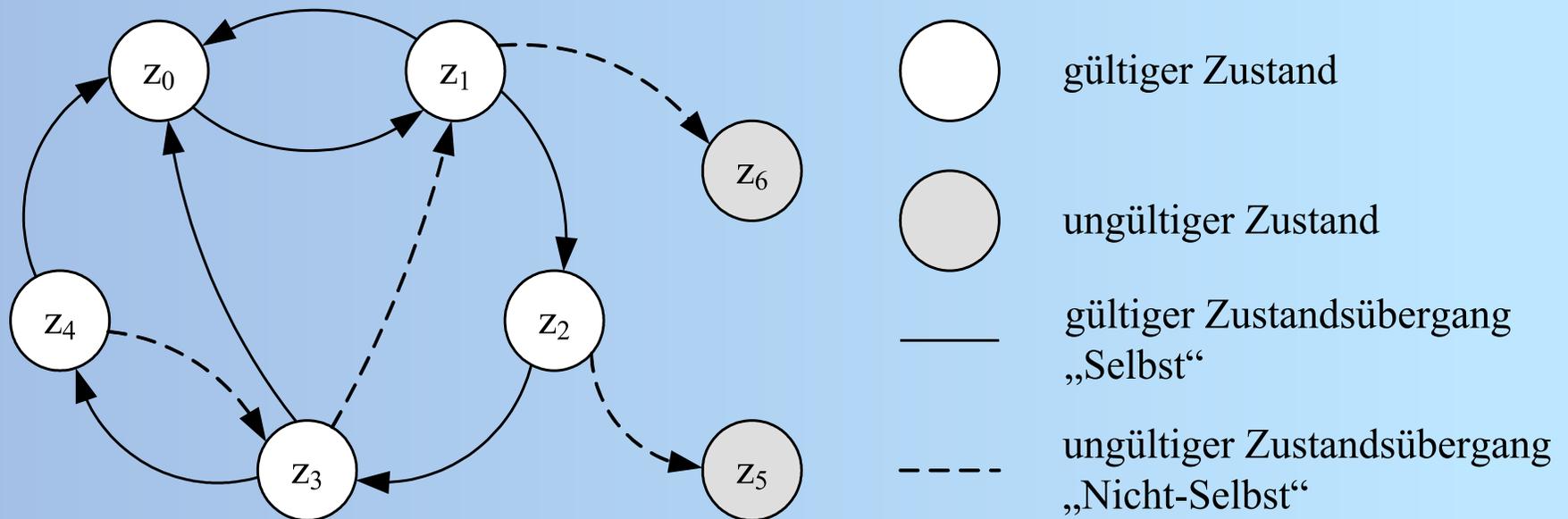


- inspiriert durch das menschliche Immunsystem
- Immunsystem ist mehrschichtiges, verteiltes System mit exzellenter Lernfähigkeit  $\Rightarrow$  Epigenese

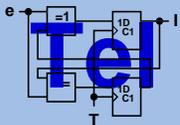


nach Bradley und Tyrrell: *Multi-layered Defence Mechanisms: Architecture, Implementation and ....* 2001

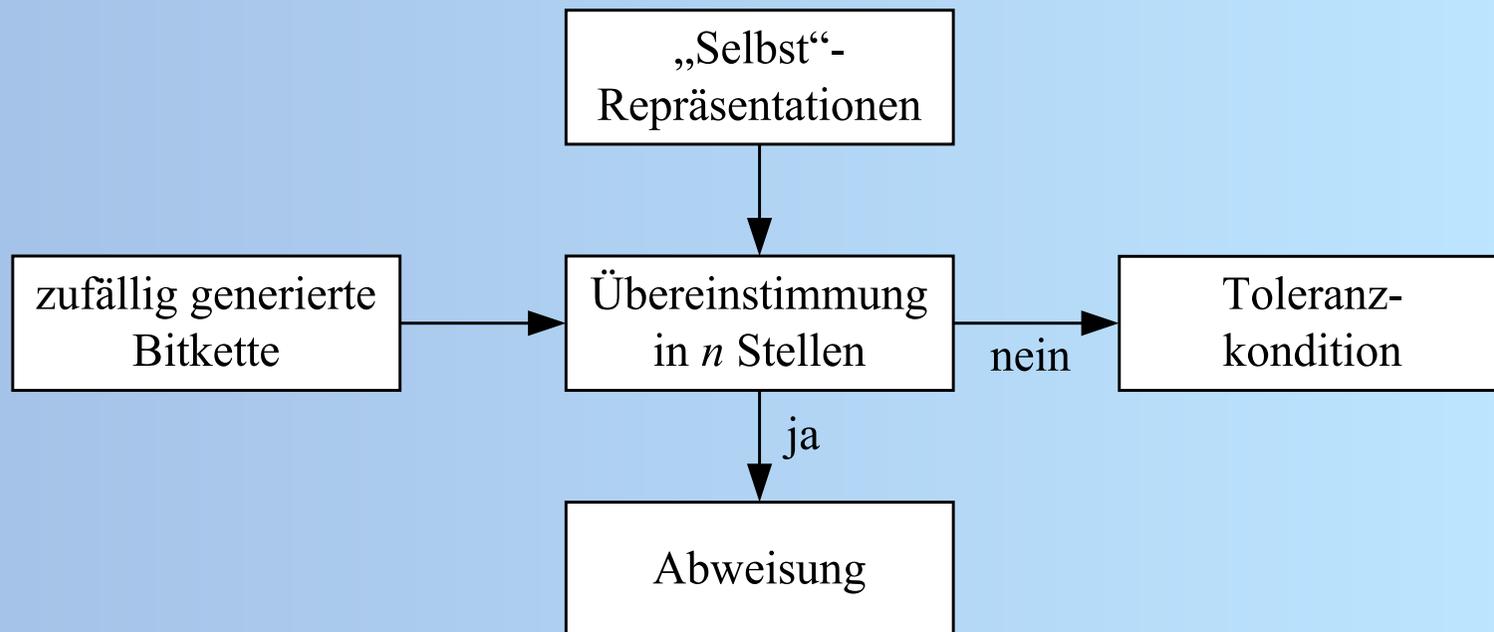
- ❑ „Selbst“-„Nicht-Selbst“-Unterscheidung
- ❑ Darstellung des Systems als endlicher Automat
- ❑ Schritt 1: *Datensammlung*



„Nicht-Selbst“-Zustandsübergänge nicht vollständig eingezeichnet

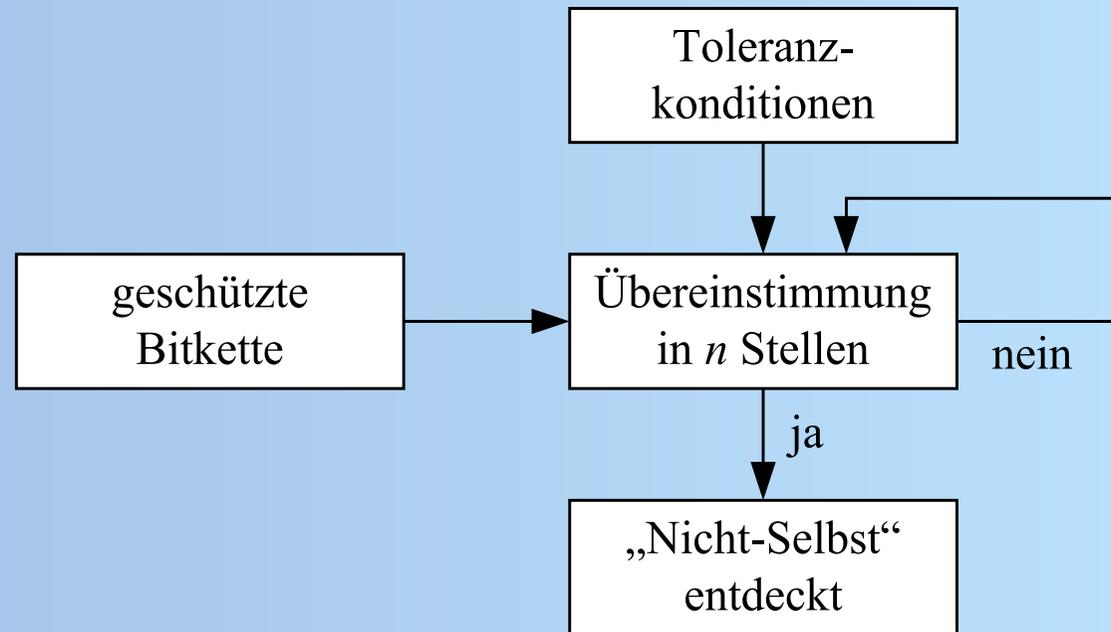


- ❑ „Negative Auswahl“-Algorithmus
- ❑ Schritt 2: *Toleranzkonditionerstellung*

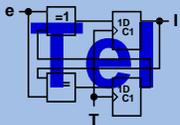


nach Forrest u.a.: *Self-Nonself Discrimination in a Computer*. 1994

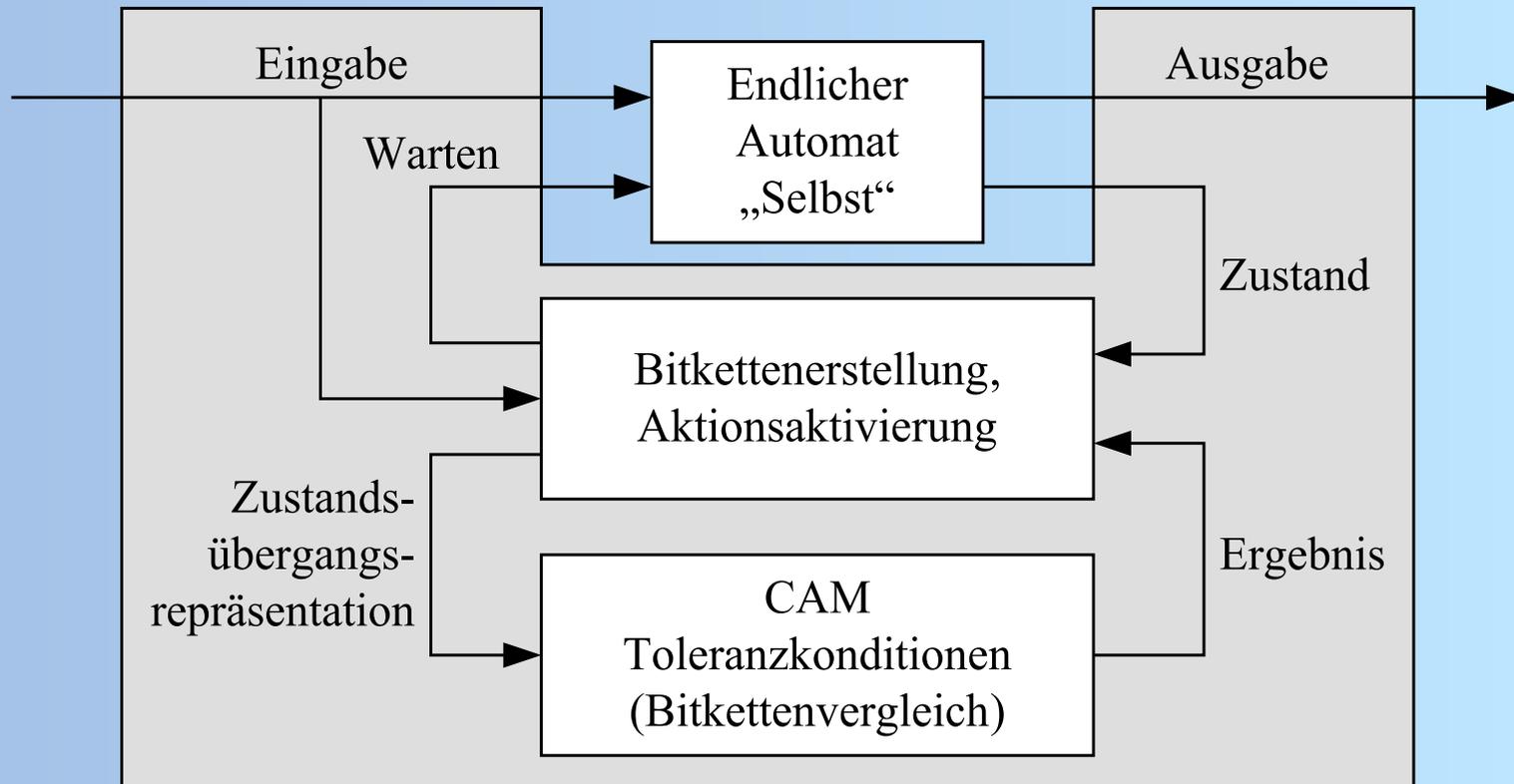
- ❑ Toleranzkonditionen in einem ausschnittsweise vergleichenden, inhaltsadressierbaren Speicher
- ❑ Schritt 3: *Fehlererkennung*



nach Forrest u.a.: *Self-Nonself Discrimination in a Computer*. 1994

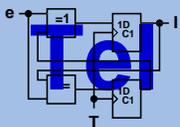


# Umsetzung von Störungstoleranz Immunotronics – Gesamtaufbau

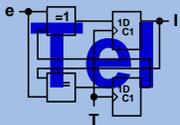


nach Bradley und Tyrrell: *Immunotronics - novel finite-state-machine architectures with built-in self-test ...* 2002

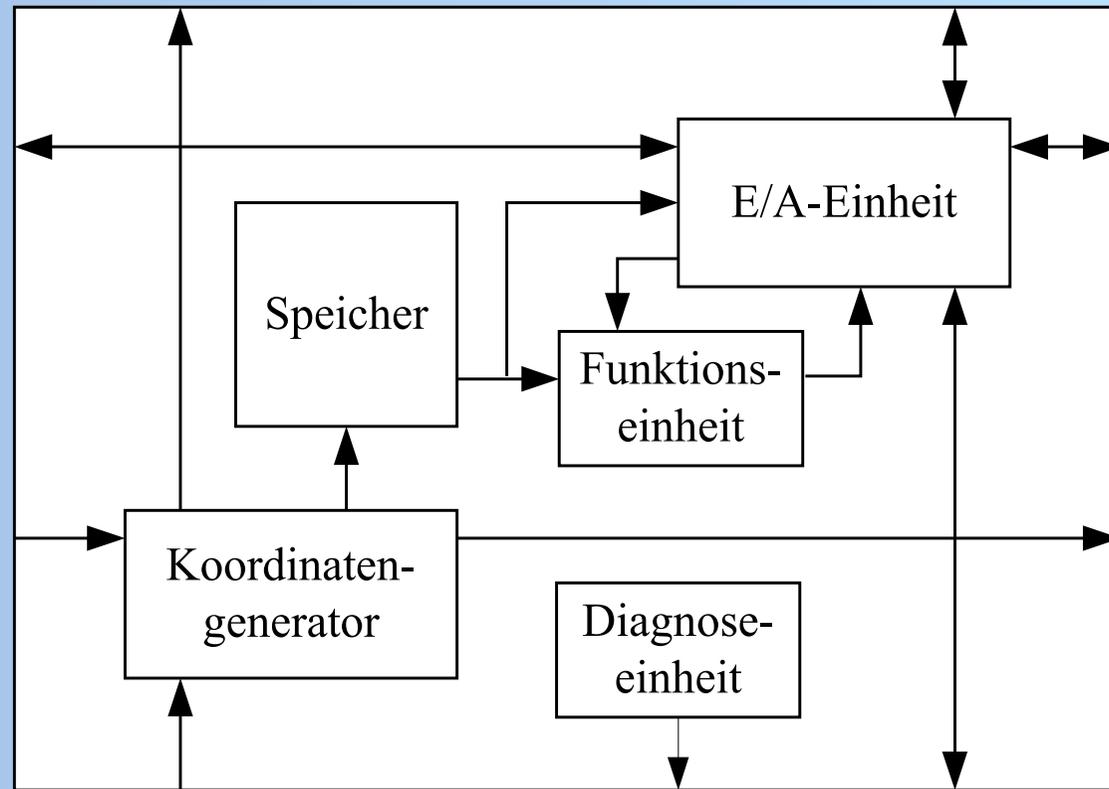
⇒ nebenläufige Fehlererkennung



- ❑ inspiriert durch embryologische Entwicklung und zellularen Aufbau aller Lebewesen  $\Rightarrow$  Ontogenese
- ❑ Übertragung der Eigenschaften Selbstreplikation und Selbstreparatur
- ❑ FPGA zur Umsetzung hervorragend geeignet
- ❑ zwei separate Ansätze: Zürich – vierstufiger Ansatz, York – zweistufiger Ansatz

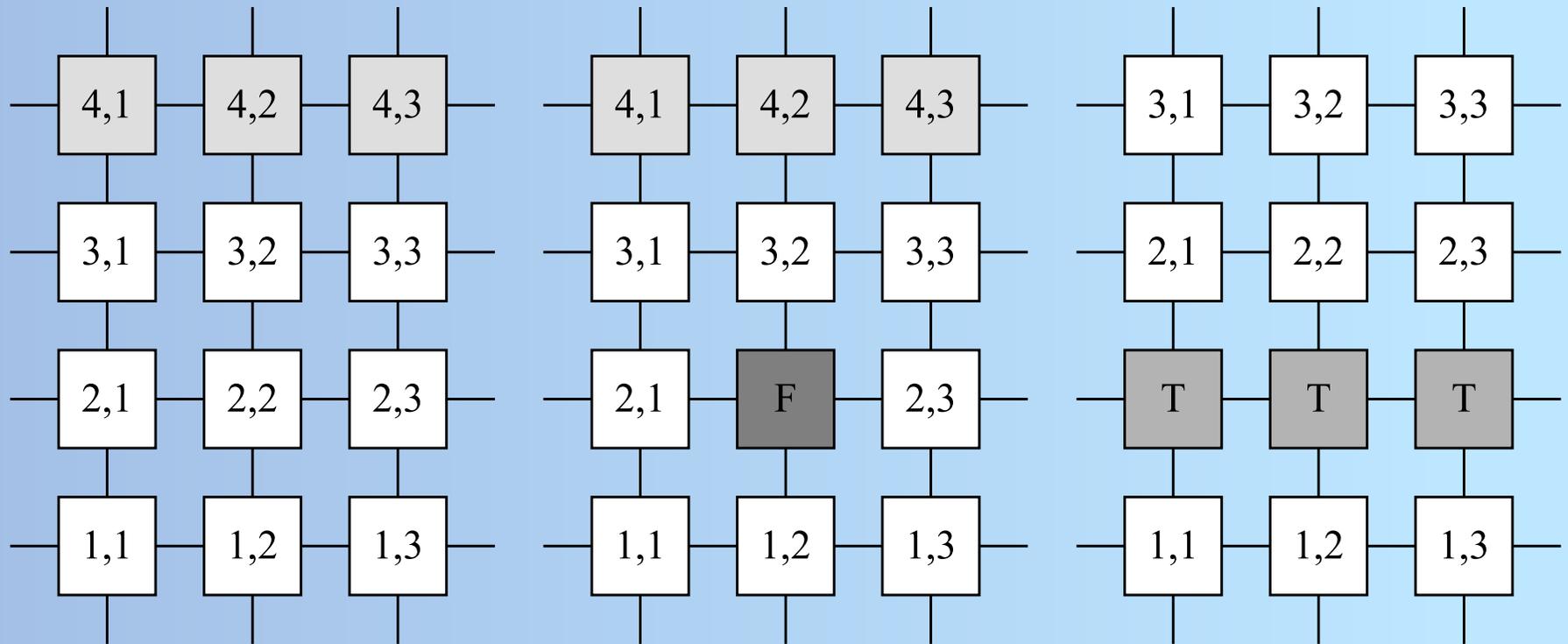


- alle Zellen im Embryonics-Feld sind identisch und universell



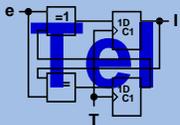
nach Ortega und Tyrrell: *A Hardware Implementation of an Embryonic Architecture Using Virtex FPGAs*. 2000

# Umsetzung von Störungstoleranz Embryonics – Rekonfiguration

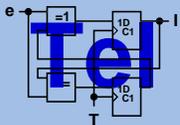


nach Ortega und Tyrrell: *A Hardware Implementation of an Embryonic Architecture Using Virtex FPGAs*. 2000

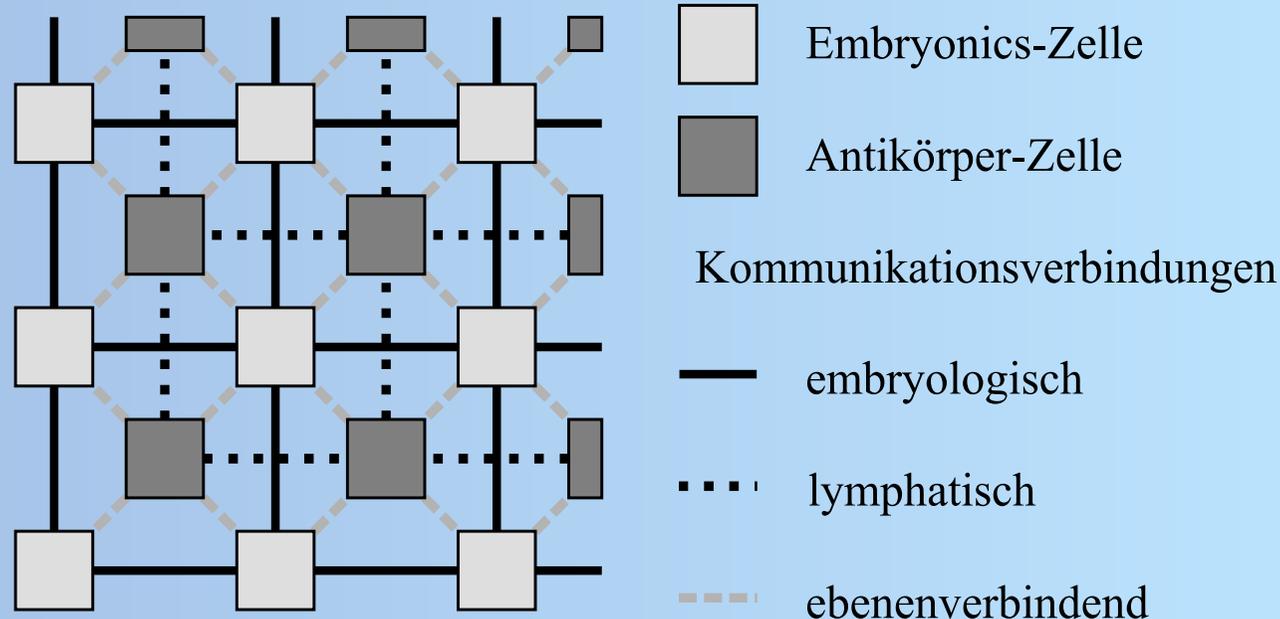
⇒ „vollständige“ Störungstoleranz



- ❑ Kombination beider Systeme um Stärken zu vereinen
- ❑ überwachende Immunotronics-Schicht über Embryonics-Schicht
- ❑ zwei Ansätze: York – mehrfache Überwachung,  
Bristol – einfache Überwachung



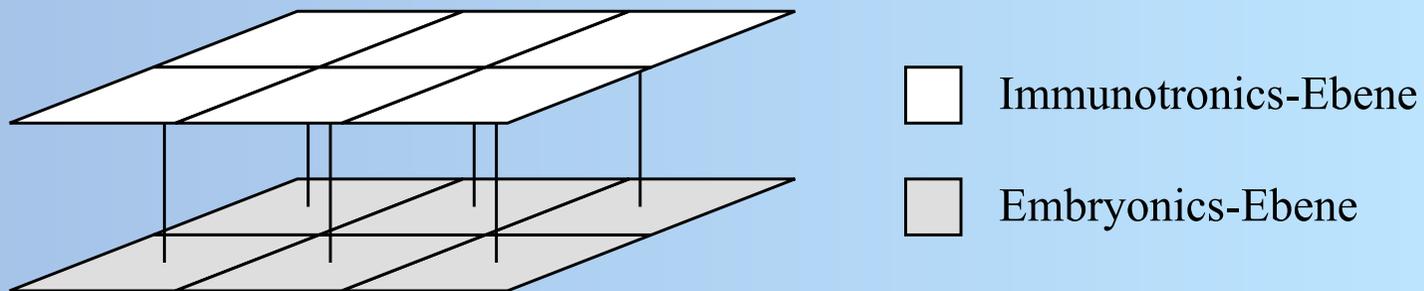
- ❑ Antikörper-Zelle überwacht nur die Konfiguration der Embryonics-Zelle
- ❑ Konzept der eigentlichen Immunotronics sehr vereinfacht



nach Bradley u.a.: *Embryonics+Immunotronics: A Bio-Inspired Approach to Fault Tolerance*. 2000

# Embryonics+Immunotronics – Bristol'sche Ansatz

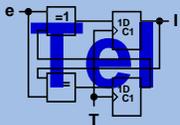
- ❑ Immunotronics-Zelle ähnlich der Embryonics-Zelle  
⇒ zweifach-modulare Redundanz
- ❑ Konzept der eigentlichen Immunotronics vollständig verworfen



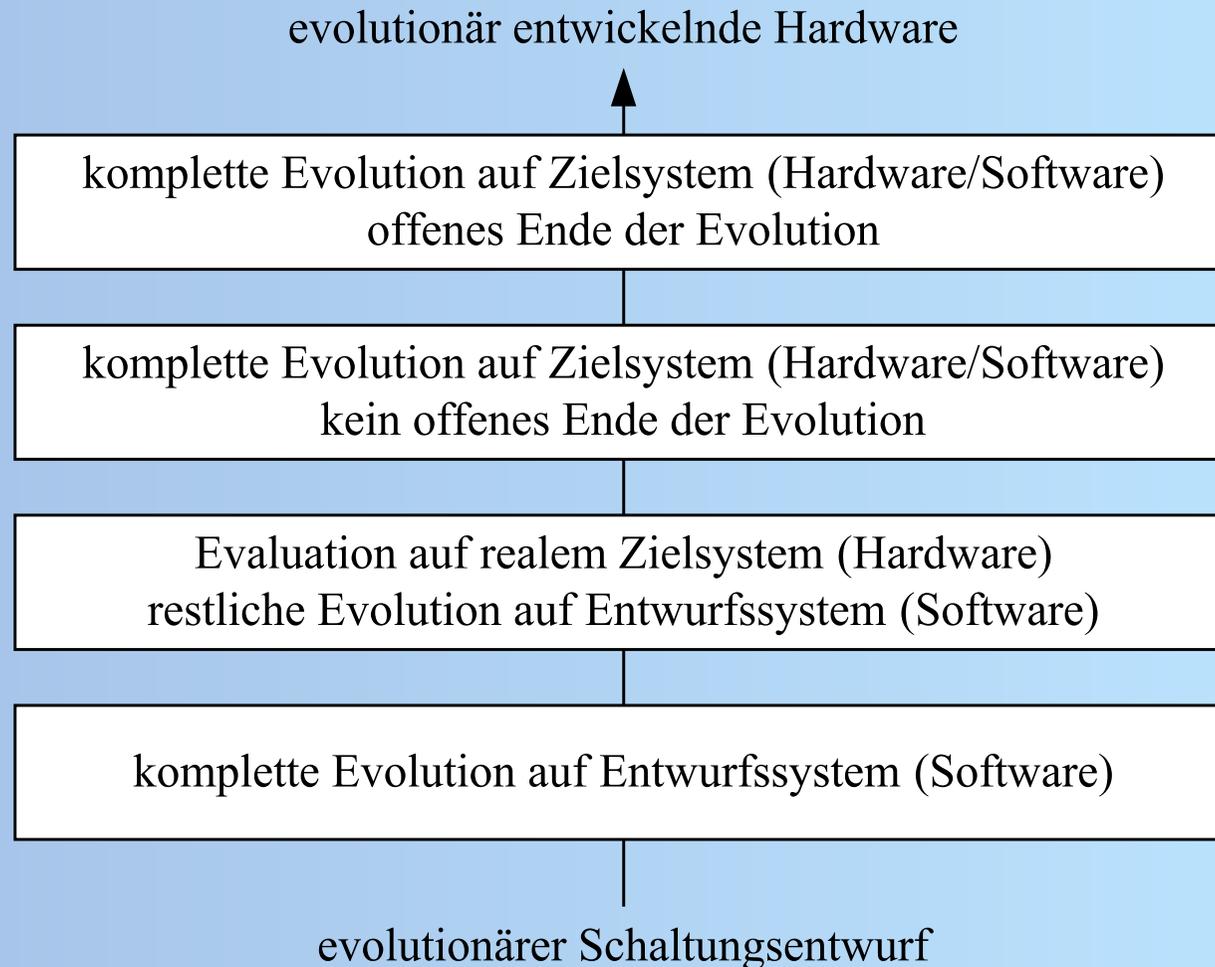
nach Zhang u.a.: *Artificial Innate Immune System: an Instant Defence Layer of Embryonics*. 2005

- ❑ baut auf Evolutionäre Algorithmen  $\Rightarrow$  Phylogenese
- ❑ verarbeitbare Präsentation (Genotyp) des Individuums (Phänotyp) muss gefunden werden
- ❑ gute Bewertungsfunktion ist entscheidend für späteres Ergebnis

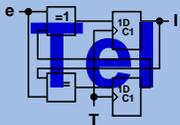
- I) Generiere anfängliche Population (Generation)
- II) Solange das Zielkriterium nicht erreicht ist wiederhole:
  - i) Bestimme die Eignung der Individuen der Population
  - ii) Selektiere die geeignetsten Individuen
  - iii) Generiere Nachkommen der selektierten Individuen unter Zuhilfenahme von Variationsmethoden
  - iv) Bilde eine neue Population aus den selektierten Individuen und deren Nachkommen



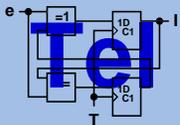
# Evolutionär entwickelbare Hardware – Varianten



nach Sipper u.a.: *A phylogenetic, ontogenetic, and epigenetic view of bio-inspired hardware systems*. 1997



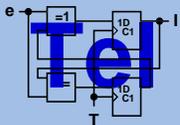
- ❑ zellulärer Aufbau → zellulares System
  - ❑ Konfiguration über Bitkette → einfach verarbeitbare Repräsentation
  - ❑ rekonfigurierbar → Selbstreparatur, Selbstanpassung
  - ❑ teilweise rekonfigurierbar → steuerndes und veränderbares bzw. aktuelles und zu testendes System auf einem Chip
- ⇒ für die Umsetzung naturinspirierter Techniken unverzichtbar



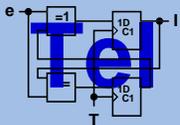
# Anwendung in einem „Immunsystem“

Kriterien:

- vollständige Störungstoleranz gegeben
- im System verteilbar/integrierbar
- skalierbar
- keine anderen Probleme, die sich negativ auswirken



- ❑ Fehlererkennende/Fehlerkorrigierende Codes
    - zu beschränktes Anwendungsgebiet
  - ❑ Watchdog, Pair-and-Spare
    - auf Logikebene nicht sinnvoll
  - ❑  $N$ -modulare Redundanz
    - + auf Logikebene gut einsetzbar
    - benötigt zusätzlichen Aufwand
  - ❑ Entwurfsvielfalt
    - + beseitigt das Problem der Entwicklungsstörungen bei NMR
  - ❑ Selbstüberprüfende Komponenten
    - + sehr gut integrierbar, elementare Fehlererkennung
- ⇒ für Fehlererkennungsaufgaben unerlässlich, aber es fehlt wirkungsvolle Systemwiederherstellung um als „Immunsystem“ zu wirken

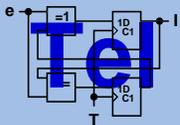


## □ Immunotronics

- + sehr leistungsfähige Fehlererkennung
- nur Fehlererkennung
- exponentielle Zeitkomplexität der Toleranzkonditionenerstellung
- enormer Ressourcenbedarf bei Verteilung im System
- keine Selbstüberprüfung

## □ Embryonics

- + hervorragende Störungsbehandlung
- keine vollständige Fehlererkennung
- Anwendung muss funktionell auf Zellen aufteilbar sein
- Speichermangel

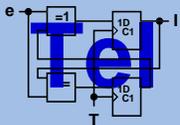


❑ Embryonics + Immunotronics

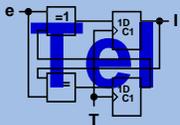
- trotz Kombination keine vollständige Fehlererkennung
- Speichermangel
- erhöhter Ressourcenbedarf durch Kombination

❑ EHW

- + vollständig integrierte Störungstoleranz
- viel Probieren und Erfahrung notwendig
- Synthese großer Systeme unmöglich
- sehr eingeschränkte Portabilität
- Korrektheit schlecht prüfbar
- Evolution bringt auch kontraproduktive Individuen hervor
- online anpassbare EHW nicht möglich

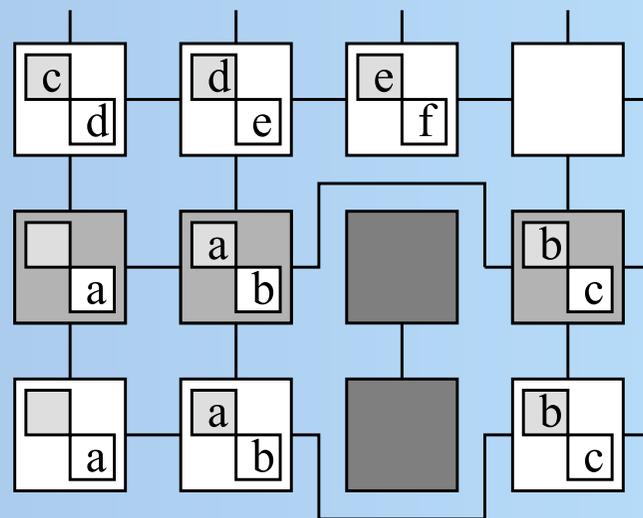


- ❑ innovative Techniken durch Inspiration bei der Natur möglich
- ❑ Natur bietet bedeutend mehr Ressourcen als die Technik  
→ umfangreiche Anpassungen bei der Übertragung auf technische Systeme notwendig
- ❑ klassischen Techniken nicht geeignet
- ❑ mit heutigem Stand der Technik ist keine der naturinspirierten Techniken für ein „Immunsystem“ geeignet
- ❑ Embryonics bietet größtes Potential
- ❑ OC-Systeme nur für einige Spezialanwendungen sinnvoll
- ❑ ausführliche Spezifikation und gewissenhafte Anwendung der anderen Methoden minimieren die Notwendigkeit komplexer Störungstoleranzsysteme



## Ansätze zur Weiterentwicklung der Embryonics:

- ❑ zwei sich gegenseitig überprüfende Zellen zur Fehlererkennung
- ❑ Funktionsbestimmung nicht anhand von Koordinaten →  
Speicherung der aktuellen und einer alternativen Funktion,  
Schieben der Funktionen bei Rekonfiguration
- ❑ anstatt Transparentschaltung, komplette Isolation der Zelle bei einem Fehler



- ❑ Avizienis u.a.: *Basic concepts and taxonomy of dependable and secure computing*. 2004
- ❑ Bradley u.a.: *Embryonics+Immunotronics: A Bio-Inspired Approach to Fault Tolerance*. 2000
- ❑ Bradley und Tyrrell: *Immunotronics - novel finite-state-machine architectures with built-in self-test using self-nonsel self differentiation*. 2002
- ❑ Ortega und Tyrrell: *A Hardware Implementation of an Embryonic Architecture Using Virtex FPGAs*. 2000
- ❑ Mange u.a.: *Towards robust integrated circuits: The embryonics approach*. 2000
- ❑ Yao und Higuchi: *Promises and challenges of evolvable Hardware*. 1999

