Betriebsfehler: Fehlertoleranz und Redundanz

Vortrag von Dirk Spöri am
Donnerstag, 11. Juli 2002

Gliederung

1. Einleitung: Katastrophe Lauda Air
2. Grundbegriffe
   - Fehlerkorrektur
   - Fehlertoleranz
   - Fehlerredundanz
3. Fehlerbehandlung bei modularen Systemen
   - Fehlertoleranz in Speicherbausteinen
   - Fehlertoleranz bei Speichersystemen, RAID
   - Redundante Designs
4. Beispiel Airbus A320

Absturz der Lauda Air Boeing 767

Sonntag, 26. Mai 1991
23:05 Bangkok Zeit: Lauda Air Boeing 767-300ER, aus Hongkong kommend, hebt in Bangkok ab. Alles normal, Flugzeug auf 7000 m gestiegen und steigt weiter

23:16 Eine beratende Warnleuchte (niedrigste von drei Warnstufen) begann zu blinken. Copilot schaute auf Checkliste: „ein anderer Fehler kann die Schubumkehr auslösen. Die Schubumkehr schaltet sich normalerweise nach der Landung ein. Keine sofortige Reaktion notwendig“

23:18 Copilot Turner: „Sie ist eingeschaltet“ (Schubumkehr aktiviert während Steiflug bei hoher Motorleistung)
Einige Momente später hört man eine akkustische Warnung, daß die Fluglage instabil ist. Zwei Sekunden später endet das Band, das Flugzeug stürtzte ab.

Absturz der Lauda Air Boeing 767

- Die Schubumkehr der Boeing 767 schaltete sich ohne Zutun der Piloten automatisch ein, was zum Auseinanderbrechen und Absturz des Flugzeuges über Thailand und zu 223 Toten führte.
- Nach Boeing-Angaben hätte eine 767 trotz Schubumkehr weiterfliegen können
- Die Einschaltung der Schubumkehr sollte während des Fluges unmöglich sein:
  - mechanisches System verhindert während Flug Schubumkehr
  - möglicherweise ausgefallen, daraufhin löste ein Computerfehler die Schubumkehr aus

Anforderungen an sicherheitskritische Komponenten (Flugsteuersystem)

Wahrscheinlichkeit eines Ausfalls: $< 10^{-9}$ / Flugstunde

d.h. eine mittlere Überlebensdauer von über 1 Milliarde Stunden.

Beispielrechnung:
- Bei 2000 Flugstunden im Jahr pro Flugzeug,
- 25 Jahre Dienstzeit pro Flugzeug,
- 1000 ausgelieferte Flugzeuge:

= 50 Millionen Stunden Flugzeit für die gesamte Flotte.

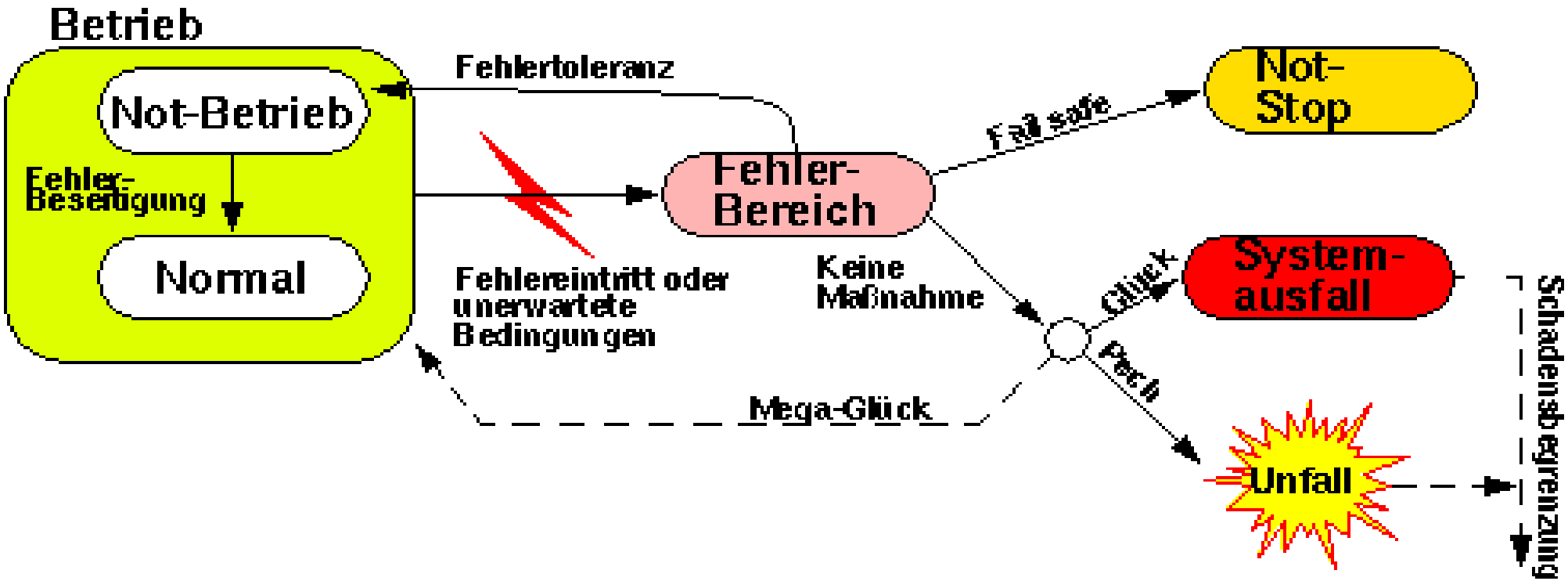
Ein Ausfall tritt dann mit weniger als 5% Wahrscheinlichkeit ein.

Was sind Betriebsfehler?

- zufällige, physikalische Fehler
- Verschleißfehler, z.B. durch Alterung elektrischer Bauteile
- Störungsbedingte Fehler aufgrund äußerer physikalischer Einflüsse
- Bedienungsfehler
- Wartungsfehler: fehlerhafte Systemeingriffe während Wartungsintervall
- Sabotage, Vandalismus

Schema

Auswirkungen von Fehlern



Definitionen

Fehlertoleranz:
Fähigkeit eines Systems, auch mit einer begrenzten Zahl fehlerhafter Subsysteme seine spezifizierte Funktion zu erfüllen.

Redundanz:
Vorhandensein von mehr als für die Ausführung der vorgesehenen Aufgaben an sich notwendigen Mittel.
(nach DIN 40041, Teil 4)

Stufen der Fehlertoleranz

Typ	Verhalten des Systems
(go)	System- und Anwendungsprogramm sicher und korrekt
(fail-operational)	System fehlertolerant ohne Leistungsverminderung
(fail-soft)	Systembetrieb sicher, aber Leistung vermindert
(fail-save)	Nur Systemsicherheit gewährleistet
(fail-unsave)	unverhersehbares Systemverhalten

Fehlerkorrektur

Vorwärtsfehlerkorrektur:
Fehler wird festgellt, aber das System versucht trotz des Fehlers oder durch Korrektur weiterzumachen.
Der Fehler wird oft gar nicht sichtbar, da Ergebnisse erst nach Korrektur verfügbar sind.

Rückwärtsfehlerkorrektur:
Fehler wird festgestellt, daraufhin kehrt das System zu einem früheren Zustand vor Auftreten des Fehlers zurück. Der fehlerhafte Prozeß wird wiederholt oder die entsprechende Leistung ist nicht mehr verfügbar.

Der Fehler ist möglicherweise direkt sichtbar oder indirekt, durch Leistungseinbußen oder eingeschränkte Funktionalität.

Redundanz

Aktive-Redundanz (funktionsbeteiligte, heiße Redundanz): Mehrere Komponenten eines Systems führen dieselbe Funktion simultan aus. Fällt eine Komponente aus, wird dieser Fehler durch die verbleibenden Komponenten direkt kompensiert und führt daher nicht unmittelbar zu einer von außen erkennbaren Reaktion.

Standby-Redundanz (passive Redundanz): Zusätzliche Mittel sind eingeschaltet/bereitgestellt, werden aber erst bei Ausfall/Störung an der Ausführung der vorgesehenen Aufgabe beteiligt.

Kalte Redundanz: Zusätzliche Mittel werden erst bei Ausfall/Störung eingeschaltet/bereitgestellt.

Entspricht:
Statische vs. **dynamische** Redundanz

Redundanz - Maßnahmen:

Strukturelle Redundanz: Erweiterung des Systems um Objekte wie zusätzliche - gleichartige Rechner, - Baugruppen, - Speicher Komponenten

Funktionelle Redundanz: Erweiterung des Systems um zusätzliche Funktionen

Informationsredundanz: z.B. durch zusätzliche Bitpositionen wie Prüfbits, Polynombildung usw.

Zeitredundanz: z.B. Wiederholung der fehlgeschlagenen Operation

Fehlerbehandlung modularer Systeme

(Module sind: Prozessoren, I/O-Devices, Sensoren, Aktuatoren...)

- Mögliche Fehler: Steuerungsstillstand oder unkontrolliertes Verhalten von Prozessormodulen
- Anomalien in aktiven Modulen (z.B. Prozessoren) verbreiten sich durch Buszugriffe, Schreibzugriffe usw. schneller im System
- solche Anomalien sind (neben mechanischen Ausfällen) aufgrund der Komplexität der Module auch die wahrscheinlichsten
- zudem sind dort mehr systematische Fehler (z.B. Entwicklungsfehler), welche sich schwerer durch Redundanz beheben lassen

Modulinterne Fehlerbehandlung

... muß abdecken

1. den Prozessor, meistens in der Form eines Microcontrollers,

2. den Speicher und

3. den Busanschluß

Fehlertoleranz in Speicherbausteinen

Redundante Codes

Das Konzept redundanter Codes basiert darauf, daß die Menge der gültigen Codeworte nur eine Teilmenge T der Menge M aller möglichen Bitkombinationen bildet.

Je größer die Redundanz des Codes ist, desto kleiner ist die Mächtigkeit der Teilmenge T im Verhältnis zur Mächtigkeit der Menge M.

Entsprechend steigt die Wahrscheinlichkeit dafür, dass ein fehlerhaftes Codewort nicht wieder Element der Teilmenge T ist, so dass bei einer Prüfung der Fehler entdeckt werden kann.

Fehlertoleranz in Speicherbausteinen

Error Correcting Codes

- hinzufügen von Prüfbits zur Erkennung und Korrektur von Fehlern
- einfachste Form: Parity-Bit
  (nur Erkennung eines einzelnen Bitfehlers)
- fortgeschrittene Methode: Hamming-Code
- weitere Stichworte (wird hier nicht behandelt):
  zyklischeCodes, arithmetische Codes, Prüfsummen

Fehlertoleranz in Speicherbausteinen

Hamming Code

Die kleinste Distanz zwischen zwei beliebigen (gültigen) Wörtern eines Codes wird **Hamming-Abstand d** genannt.
Diese kleinste Distanz ergibt sich aus der Anzahl der Binärstellen, die mindestens verfälscht werden müssen, um ein gültiges Codewort in ein anderes gültiges Codewort zu überführen.

Anzahl erkennbarer Fehler: d-1

Anzahl korrigierbarer Fehler: (d-1)/2 für ungerades d

Jedem Codewert werden Prüfbits hinzugefügt.
Mehr Prüfbits = bessere Fehlerkorrektur.

Fehlertoleranz in Speicherbausteinen

Einschränkungen und Lösungen

Ein Speicher habe W Worte mit insgesamt N Bitfehler. Der IC ist dann fehlerfrei, wenn die N Bitfehler alle auf unterschiedliche Worte fallen (für einen ECC der 1 Bitfehler pro Wort korrigiert), das hat die Wahrscheinlichkeit $P_{Korrigierbar}$

$$P_K(N) = \prod_{i=0}^{N-1}\left(\frac{W-i}{W}\right) = \frac{W!}{W^N(W-N)!}$$

Die Fehlerwahrscheinlichkeit

$$\mathbf{P_{Error}\,(N) = 1 - P_K(N)}$$

steigt rasch mit zunehmender Anzahl der Fehler.

Fehlertoleranz in Speicherbausteinen

Einschränkungen und Lösungen

Beispiel: 16 MBit DRAM mit Wörtern zu je 137 Bit (128 Bit, dazu 9(?) Prüfbits), also W = 131072; Annahme N = 200 Bitfehler

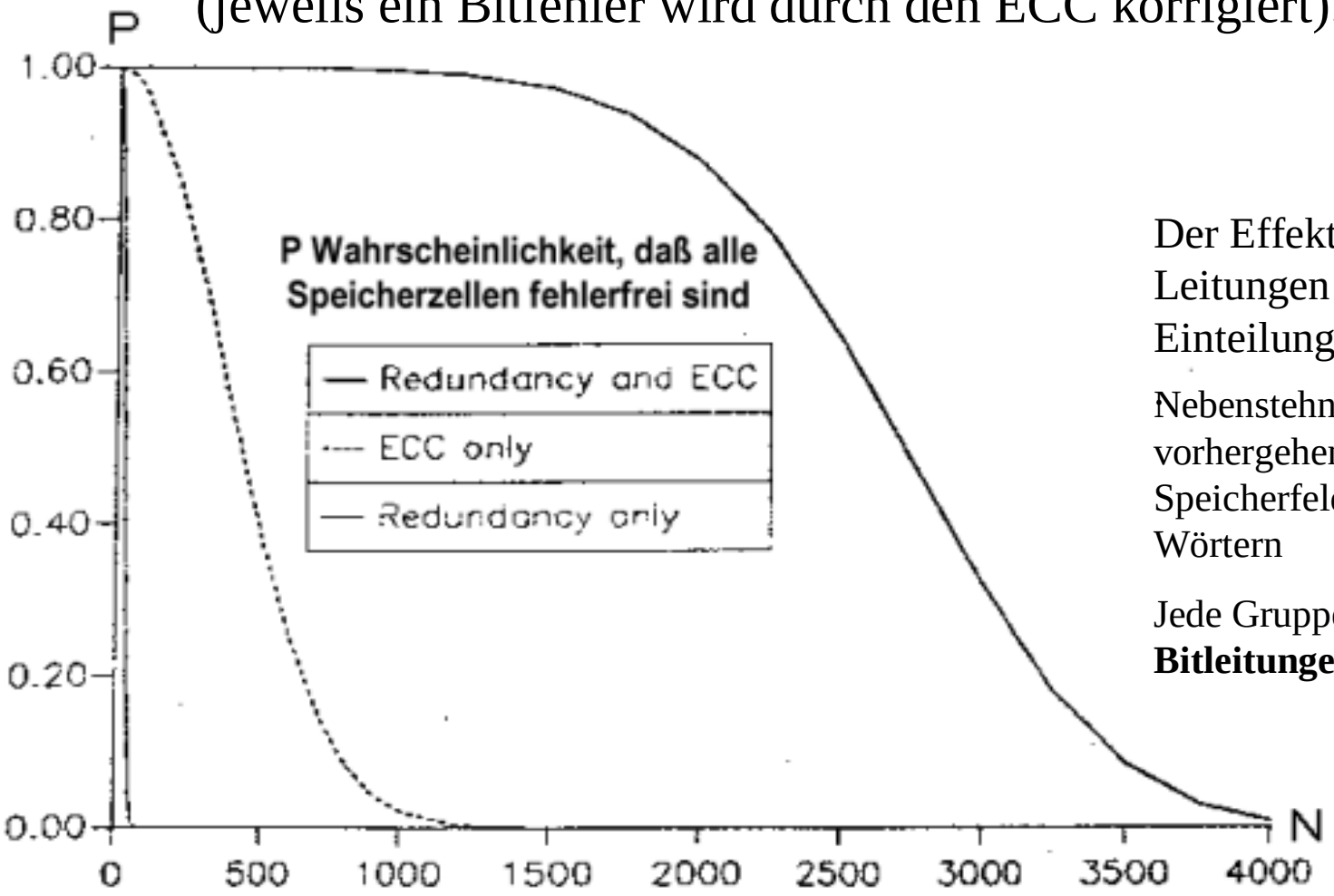
$$P_K(N) = \frac{W!}{W^N (W-N)!}$$

➡ Wahrscheinlichkeit P_{Error} für zumindest eine fehlerhafte Speicherzelle von etwas keiner als 0,2.

Fehlertoleranz in Speicherbausteinen

Verbesserung: redundante Bitleitungen

Zwei redundante Bitleitungen ermöglichen nun zwei Wörter mit je zwei Bitfehlern oder ein Wort mit drei Bitfehlern zu korrigieren (jeweils ein Bitfehler wird durch den ECC korrigiert).



Der Effekt der redundanten Leitungen ist abhängig von Einteilung des ICs.

Nebenstehndes Schaubild gilt für das vorhergehende Beispiel; dabei besteht das Speicherfeld aus 64 Gruppen zu je 2048 Wörtern

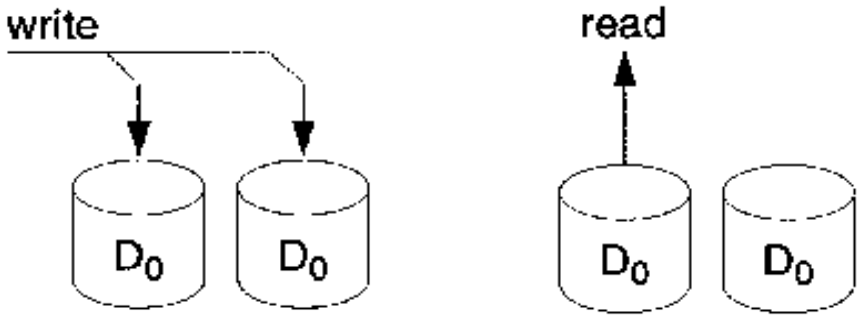
Jede Gruppe enthält **zwei redundante Bitleitungen**

Fehler in Speichersystemen

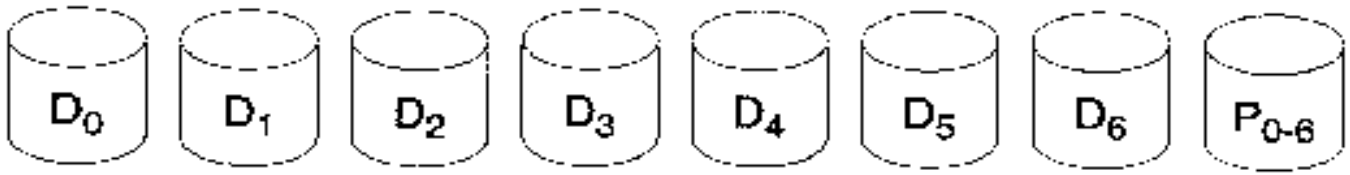
- Festplattenfehler
- Kontrollerausfall
- I/O Busfehler
- Fehler der Cache-Komponente
- Ausfall Externe Stromversorgung

Fehlertoleranz bei Speichersystemen

Mirroring



Parity



$$P_{0\text{-}6} = D_0 \oplus D_1 \oplus D_2 \oplus D_3 \oplus D_4 \oplus D_5 \oplus D_6$$

D = Drive, P = Parity

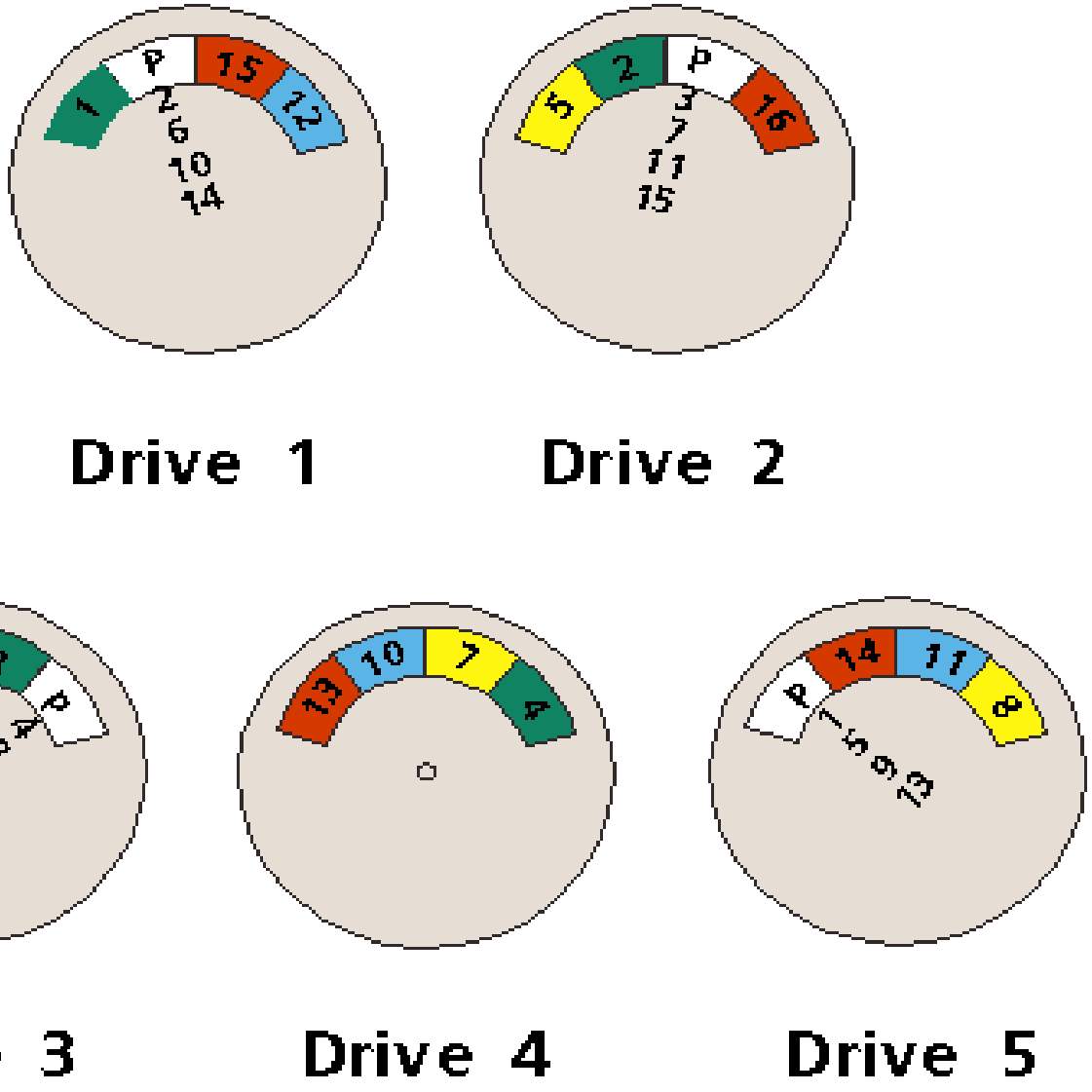
Beispiel RAID

RAID steht für **Redundant Array of Inexpensive Disks**. Verschiedene “RAID-Level” beschreiben dabei Systeme mit mehreren Massenspeichern, die sich einerseits in **Geschwindigkeit**, andererseits in **Sicherheit**, oder in beidem auszeichnen.

Sicherheit wird durch die beiden Prinzipien **Mirroring** und Error Correcting Codes (**Parity)** erreicht.

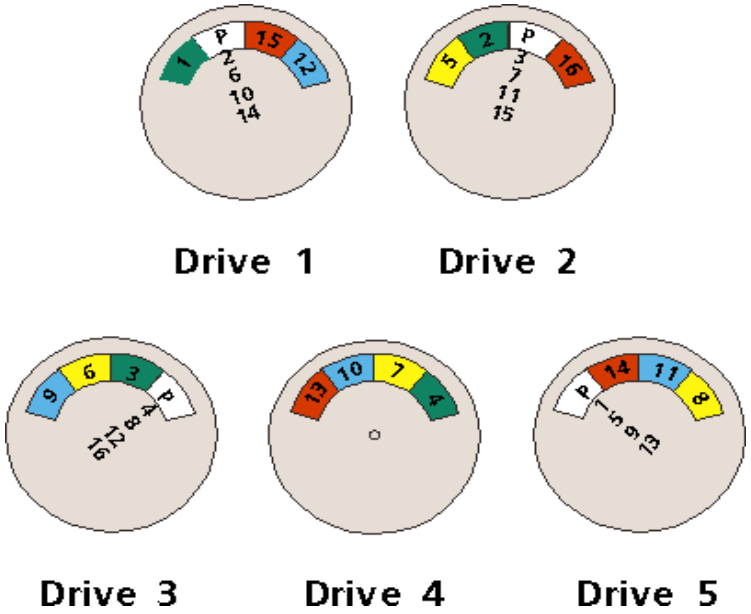
Eines der verbreitetsten RAID-Systeme ist RAID5.

RAID Level 5



(P = Paritätsstreifen)

RAID Level 5



Ausfall einer Platte wird kompensiert aus Paritätsstreifen der anderen Platten.

Bei Ausfall von Platte 2 errechnen sich die fehlenden Daten durch die folgende Gleichung:

$$D_{i.2} = D_{i.1} \oplus D_{i.3} \oplus D_{i.4} \oplus P_i$$

($D_{i.j}$ = Sektor i von Laufwerk j ; P = Paritätsstreifen)

Beispiel RAID

Rekonstruktion der Daten:
Austausch der Platte (= Hot Plug, falls im laufenden Betrieb) oder Nutzung einer freien Platte im Array (Hot Spare)

Rekonstruktion **als Hintergrundprozeß**:
Abarbeitung normaler Anwenderzugriffe möglich

Aber:
Umschaltung/Austausch von Platten einfach,
aber aufwendig bei defekten Kontrollern

Modulexterne Fehlerbehandlung

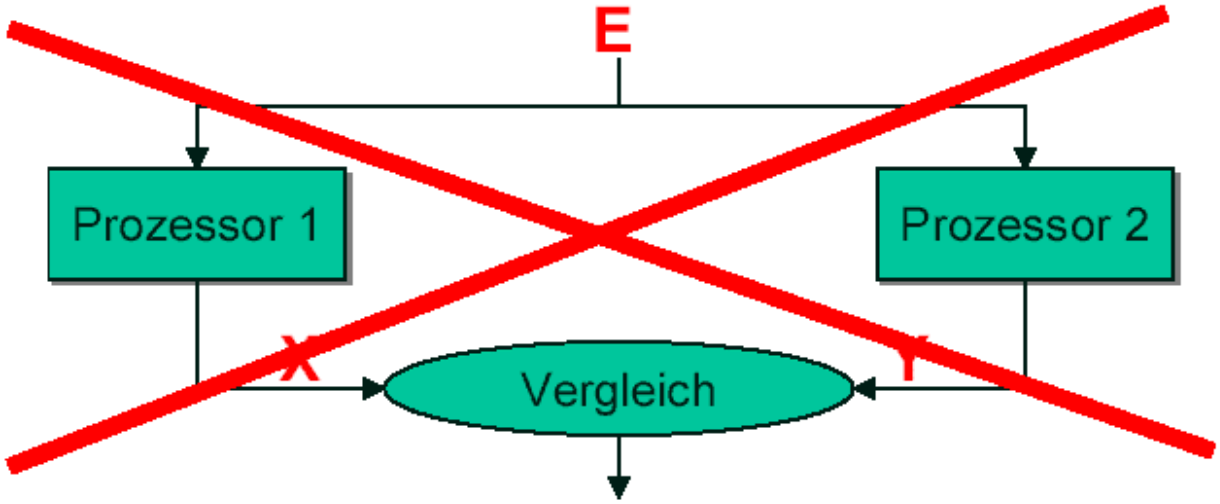
1. Erkennung des Fehlers

2. Korrektur des Fehlers durch Redundanz

Erkennung von Ausfällen

Problem:
Ein Ausfall liegt auch dann vor, wenn eine Komponente (z.B. Prozessor) fehlerhafte Ergebnisse produziert

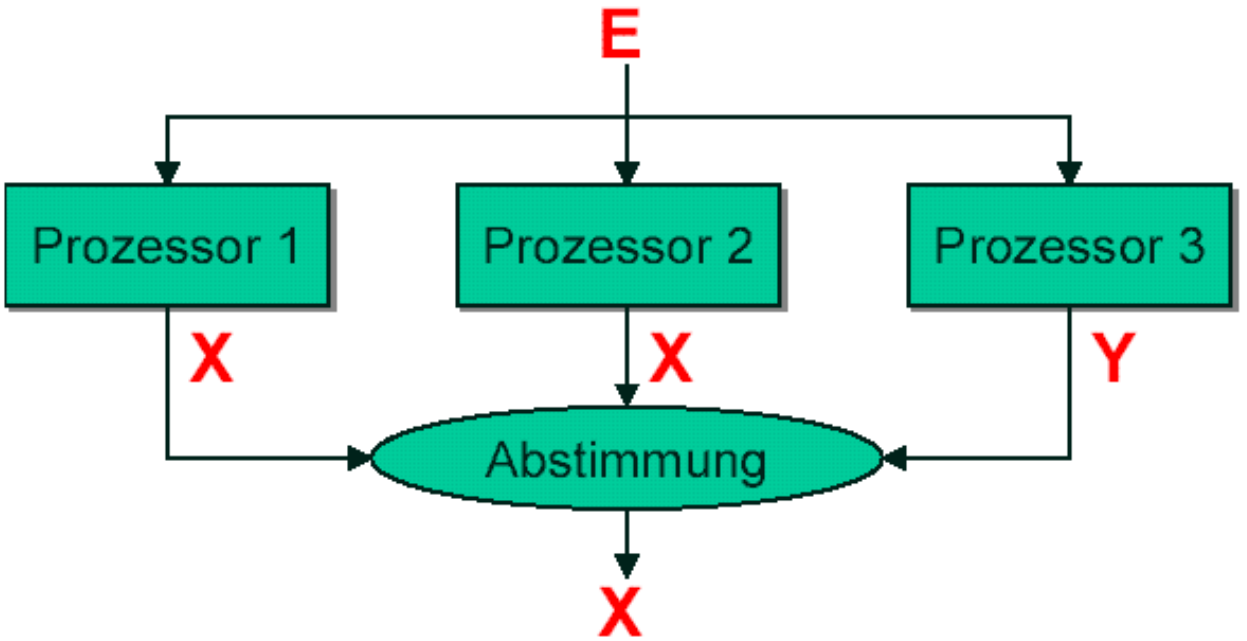
Abhilfe:
Bei 2-fach-Redundanz:
Ergebnisse der verschiedenen Prozessoren werden verglichen, bei Abweichung gilt die gesamte Komponente als ausgefallen.



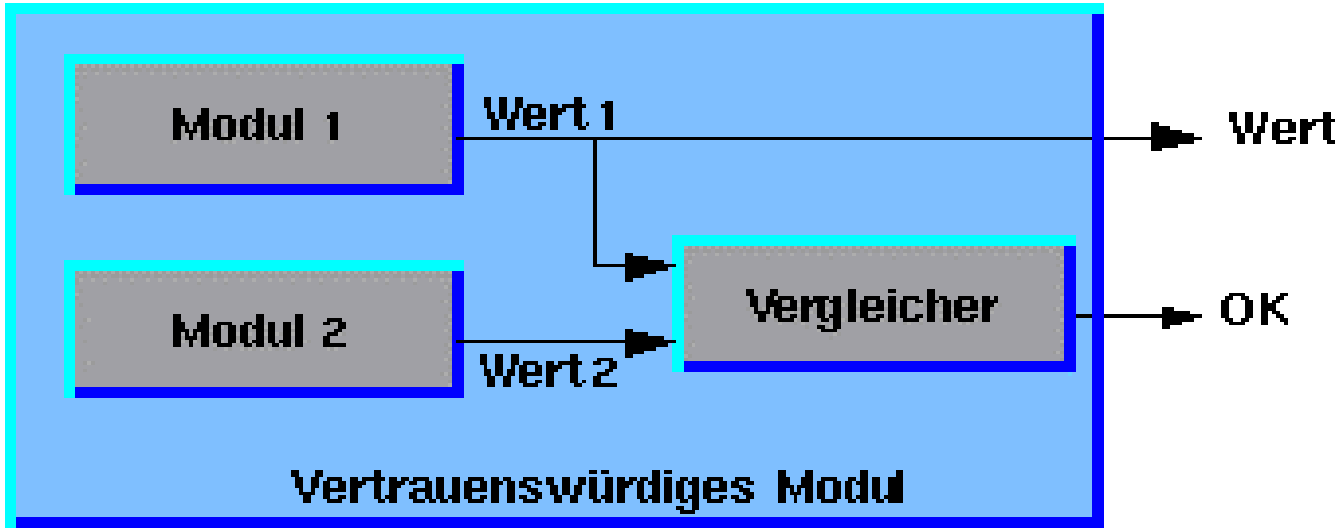
Erkennung von Ausfällen

Problem:
Ein Ausfall liegt auch dann vor, wenn eine Komponente (z.B. Prozessor) fehlerhafte Ergebnisse produziert

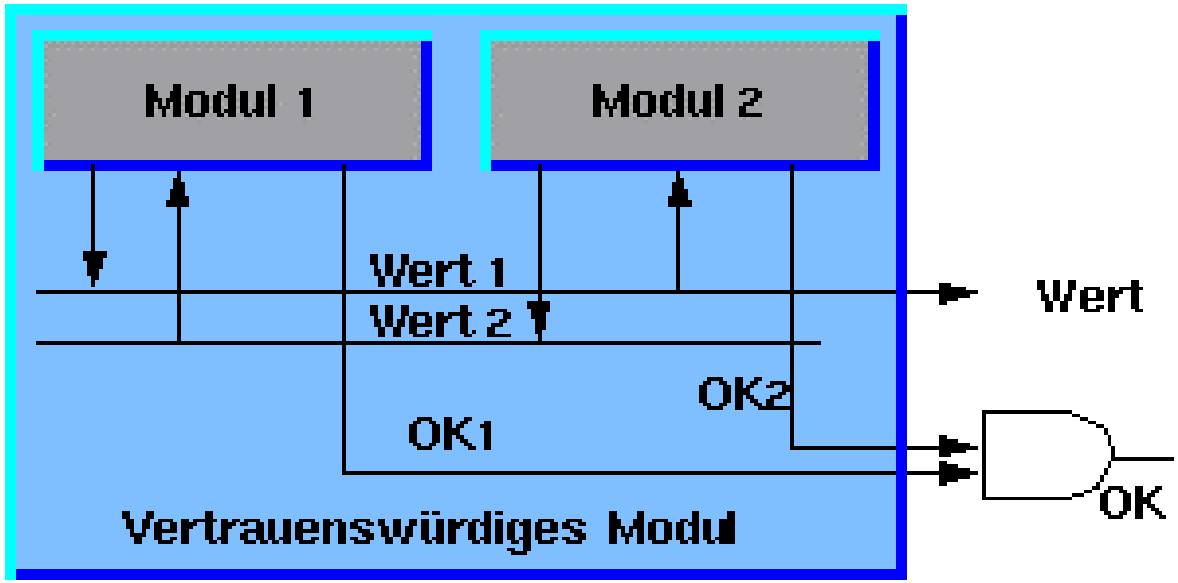
Abhilfe:
Bei 3-fach-Redundanz:
Eine Mehrheitsabstimmung wird durchgeführt



Vertrauenswürdige Module

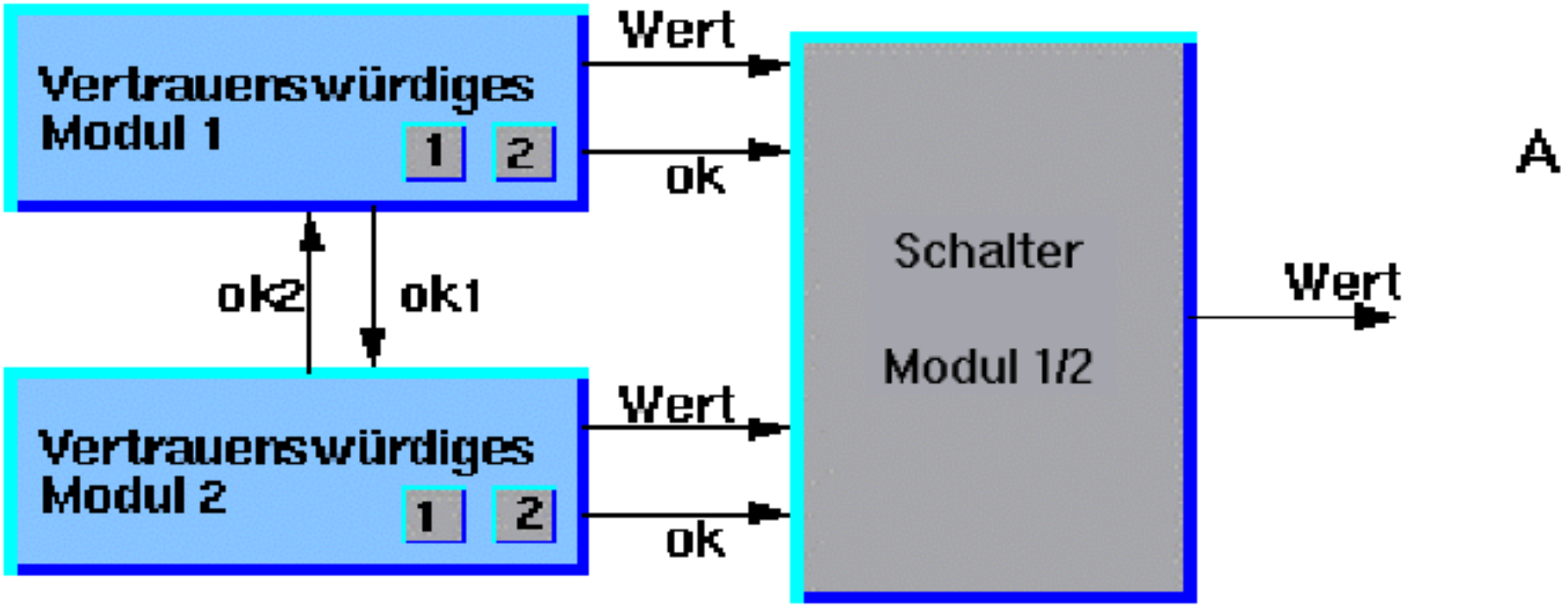


A

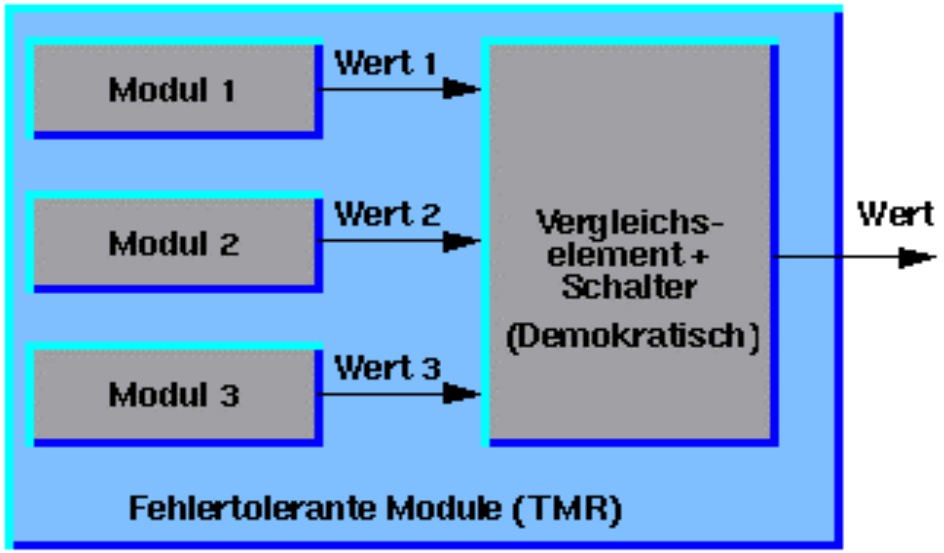


B

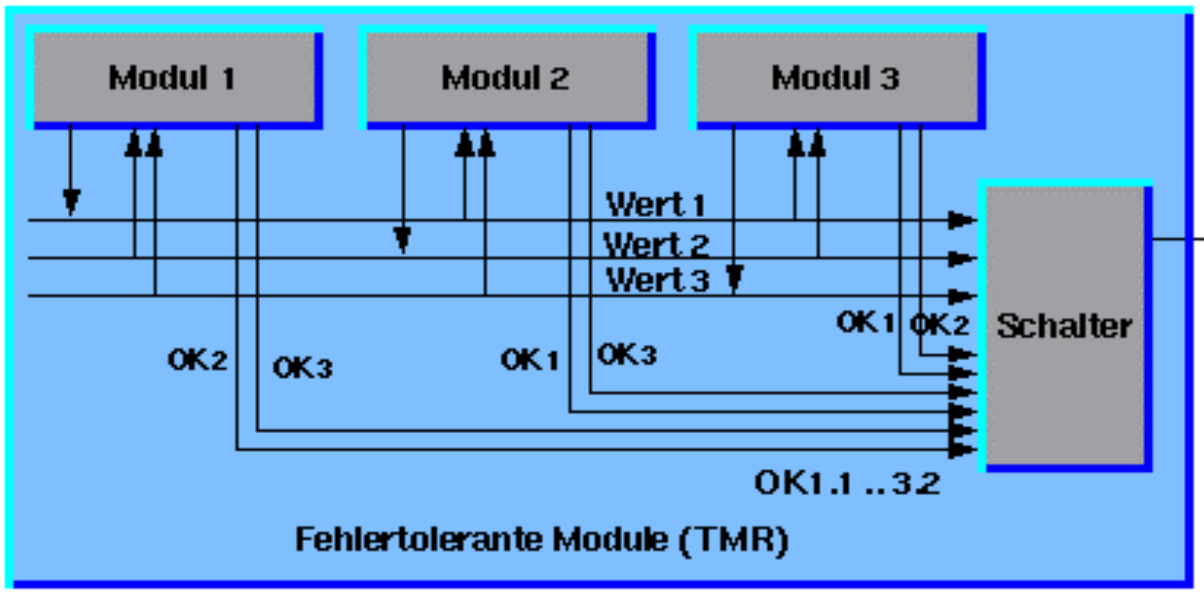
Fehlertolerante Konstruktionen



Triple Modular Redundancy (TMR)



A



B

Asynchrones Design

Jeder Computer...

• sampelt Sensoren unabhängig

• evaluiert die Rahmenbedingungen und Regeln unabhängig

• sendet die Regelkommandos an eine Mittler- oder Auswahlkomponente (“Voter”)

Probleme asynchroner Systeme

• Sensormessungen der einzelnen Module zu leicht unterschiedlichen Zeitpunkten können zu unterschiedlichen Bewertungen führen, die durch Verarbeitungsregeln nochmal verstärkt werden

⇒ Auswahlkomponente muß ein breites Intervall an Werten akzeptieren; die Erkennung einer fehlerhaften Komponenten wird erschwert und verzögert

⇒ sobald fehlerhafte Inputs ausgeschlossen werden, kann sich ein Mittelwert sprunghaft ändern, (z.B. ungünstig für Steuerung eines Flugzeuges)

• die Bewertungen der einzelnen Kanäle können noch stärker divergieren, wenn deren Regeln Entscheidungspunkte beinhalten.
Lösung: “Mehrheitsentscheidung” der Module an den entsprechenden Stellen

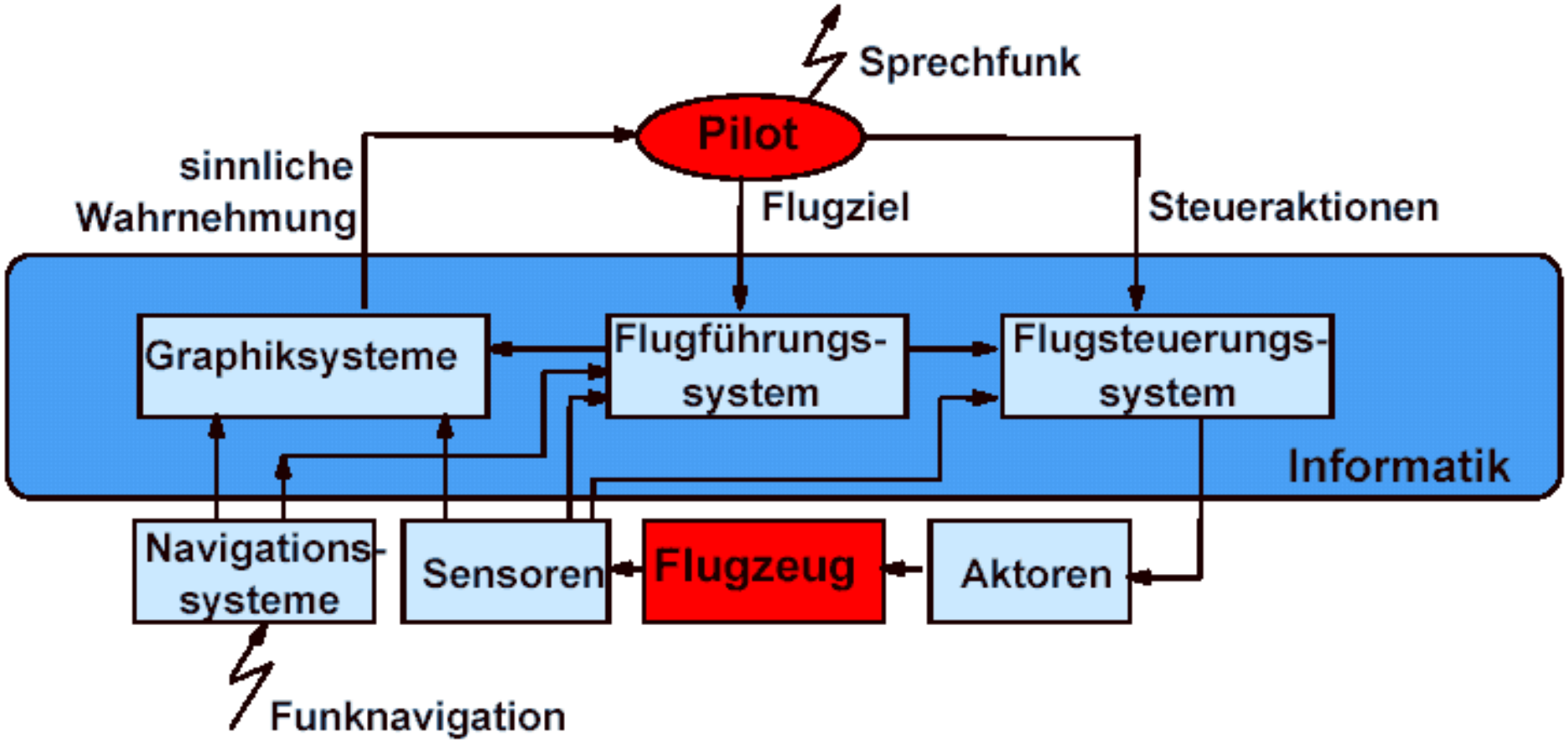
Beispiel Airbus A320

• Airbus A320 ist das erste Flugzeug, daß vollkommen Computergesteuert fliegt und nur noch eine mechanische Steuerung im Backup-System vorsieht
Technologie: “Fly-by-wire”

• Computersystem soll den Piloten entlasten und übermittelt nur noch Probleme an Piloten, die kritisch sind: z.B. Feuer im Triebwerk, im Gepäckraum oder in den Toiletten

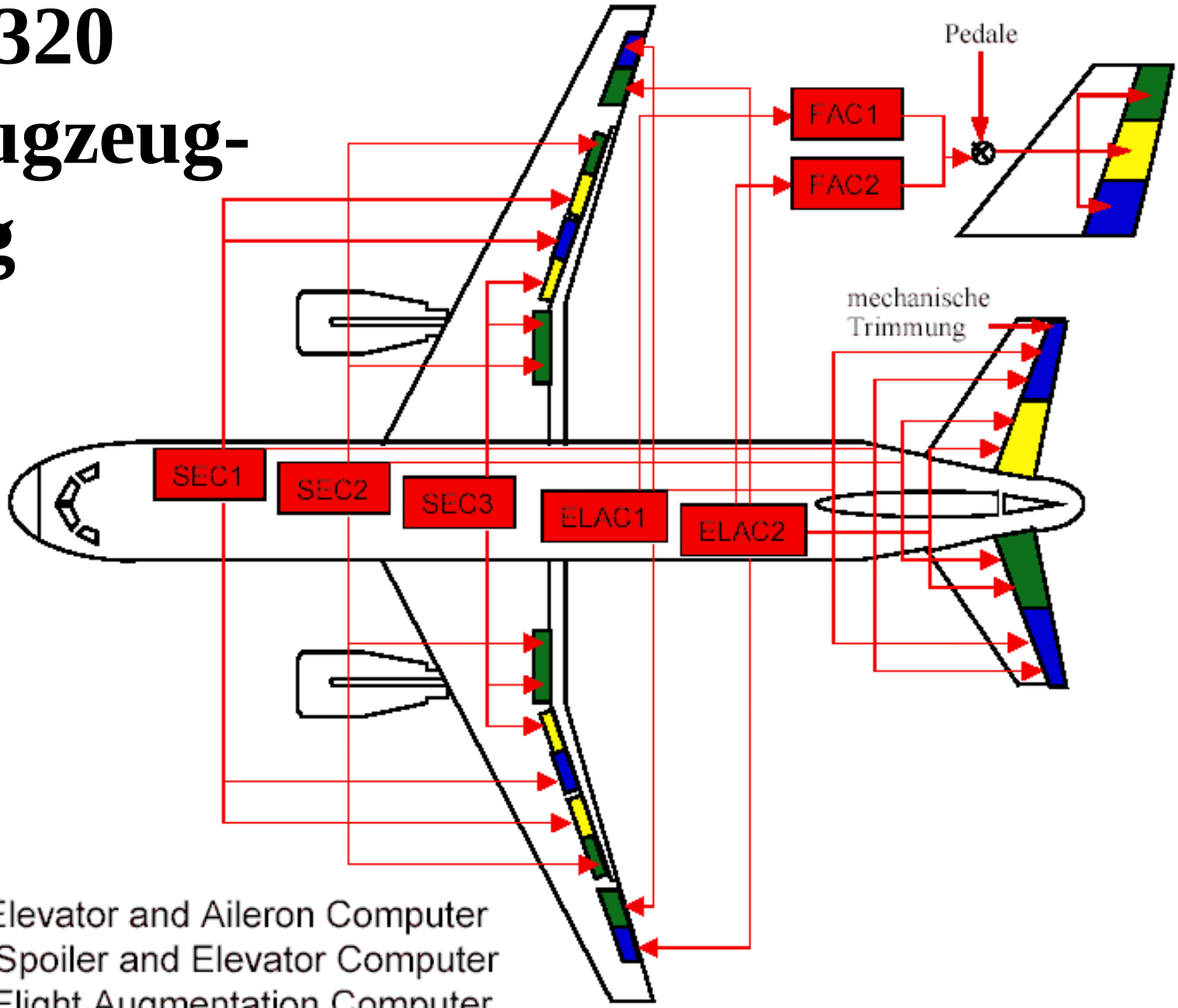
• die Piloten “sehen” also nicht mehr die Realität, sondern eine Computerabbildung und alle Steuerbefehle der Piloten werden zuerst im Computer überprüft

Computersystem Airbus A320



Kann man sich auf die eingesetzte Hardware und Software verlassen?

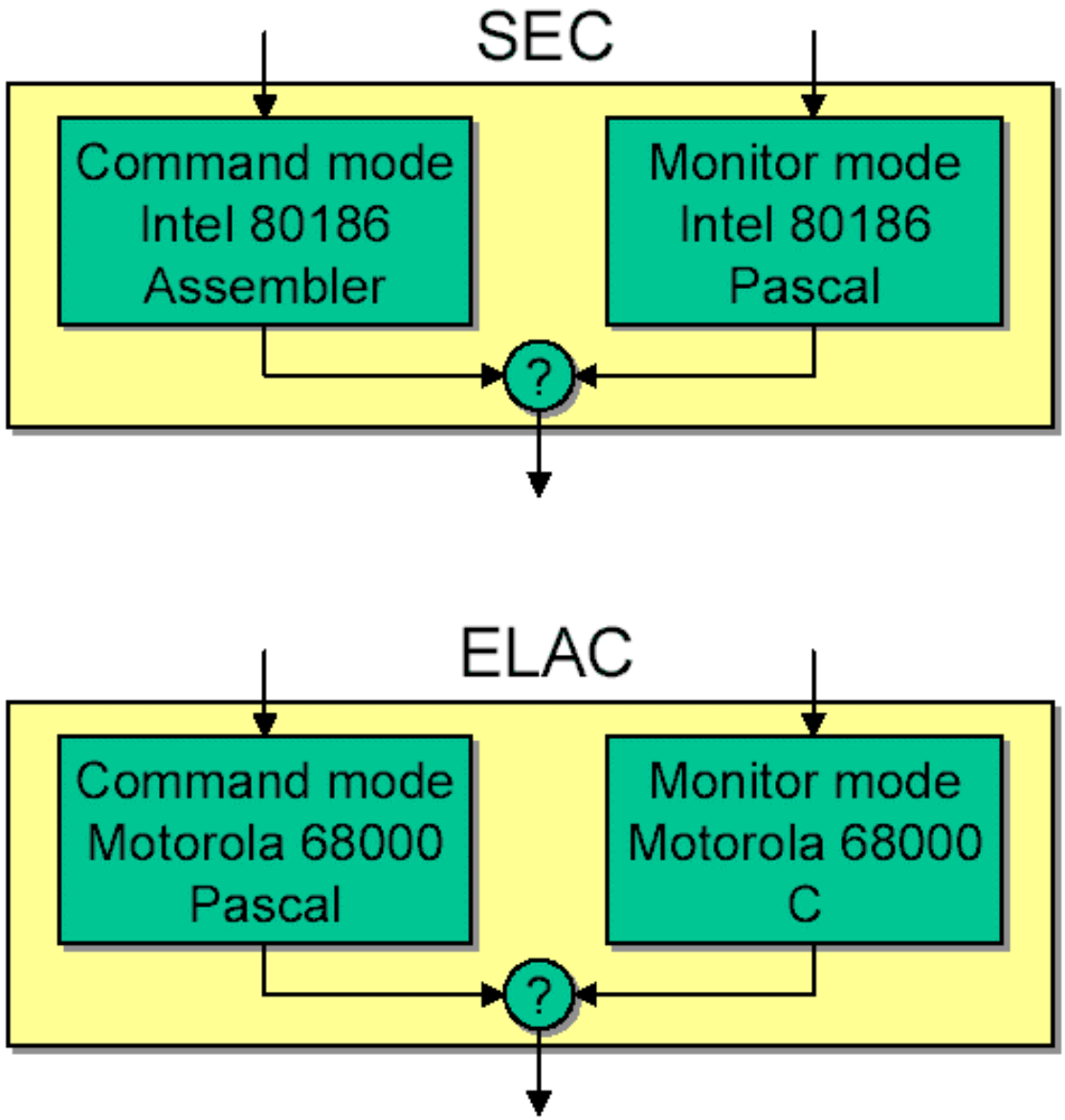
Airbus A320 Primärflugzeug-steuerung



2 ELAC Elevator and Aileron Computer
3 SEC Spoiler and Elevator Computer
2 FAC Flight Augmentation Computer

Airbus A320

Fehlererkennung durch diversitäre Hard- und Software



Literaturhinweise

- Klaus Eppele: Erhöhung der Zuverlässigkeit von Rechnersystemen (Datacom 09/91)
  http://www.improve-mtc.de/Veroffentlichungen/Zuverlassigkeit1/zuverlassigkeit1.html
- Praktische Beispiele fehlertoleranter Systeme
  http://www.morawek.at/Arbeiten/Fehlertoleranz/Fehlertoleranz.html
- S. Montenegro: Prinzipien der Fehlertoleranz
  S. Montenegro: Fehlertoleranz und Industrie Computer
  beide über http://www.first.gmd.de
- Forum On Risks To The Public In Computers And Related Systems
  http://catless.ncl.ac.uk/Risks
- Informatik Informatik im Cockpit: Pilot contra Computer
  http://kbs.cs.tu-berlin.de/publications/presentations/He260399.pdf