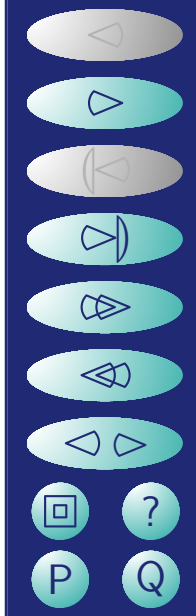
Neuaufsetzen im laufenden Betrieb nach Fehlereintritt in redundanten Echtzeitsystemen

Martin Skambraks

Fachbereich Elektrotechnik und Informationstechnik

FernUniversität, 58084 Hagen

martin.skambraks@fernuni-hagen.de



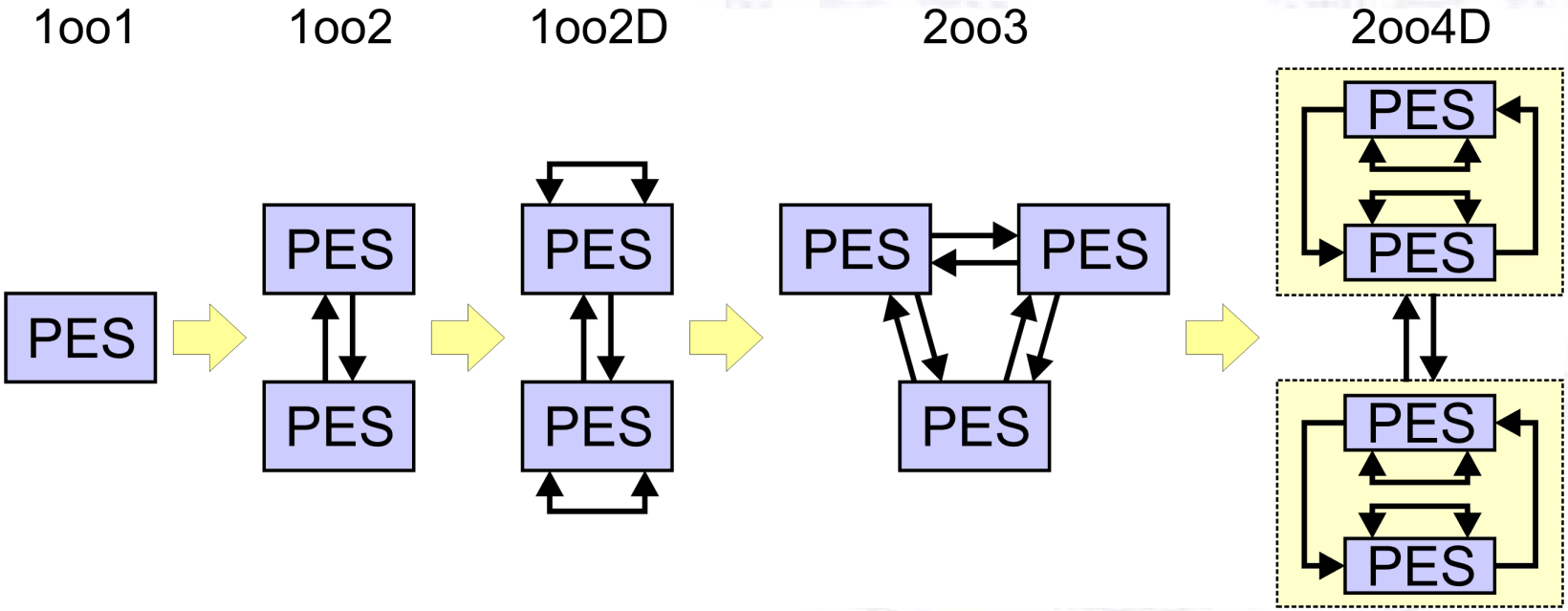
Inhalt

1. Fehlertoleranzarchitekturen zur Kompensation von Hardwareausfällen bei *Programmierbaren Elektronischen Systemen (PES)*
   - industriell eingesetzte Architekturen
   - Hauptprobleme
2. Bekannte Verfahren zum Neuaufsetzen im laufenden Betrieb
   - schaltungstechnisch realisiertes Verfahren entwickelt am *Charles Stark Draper Laboratory*
   - softwaretechnisch realisiertes Verfahren von Bondavali et al.
3. Vorstellung eines neuartigen Konzeptes

Industriell eingesetzte Fehlertoleranzarchitekturen

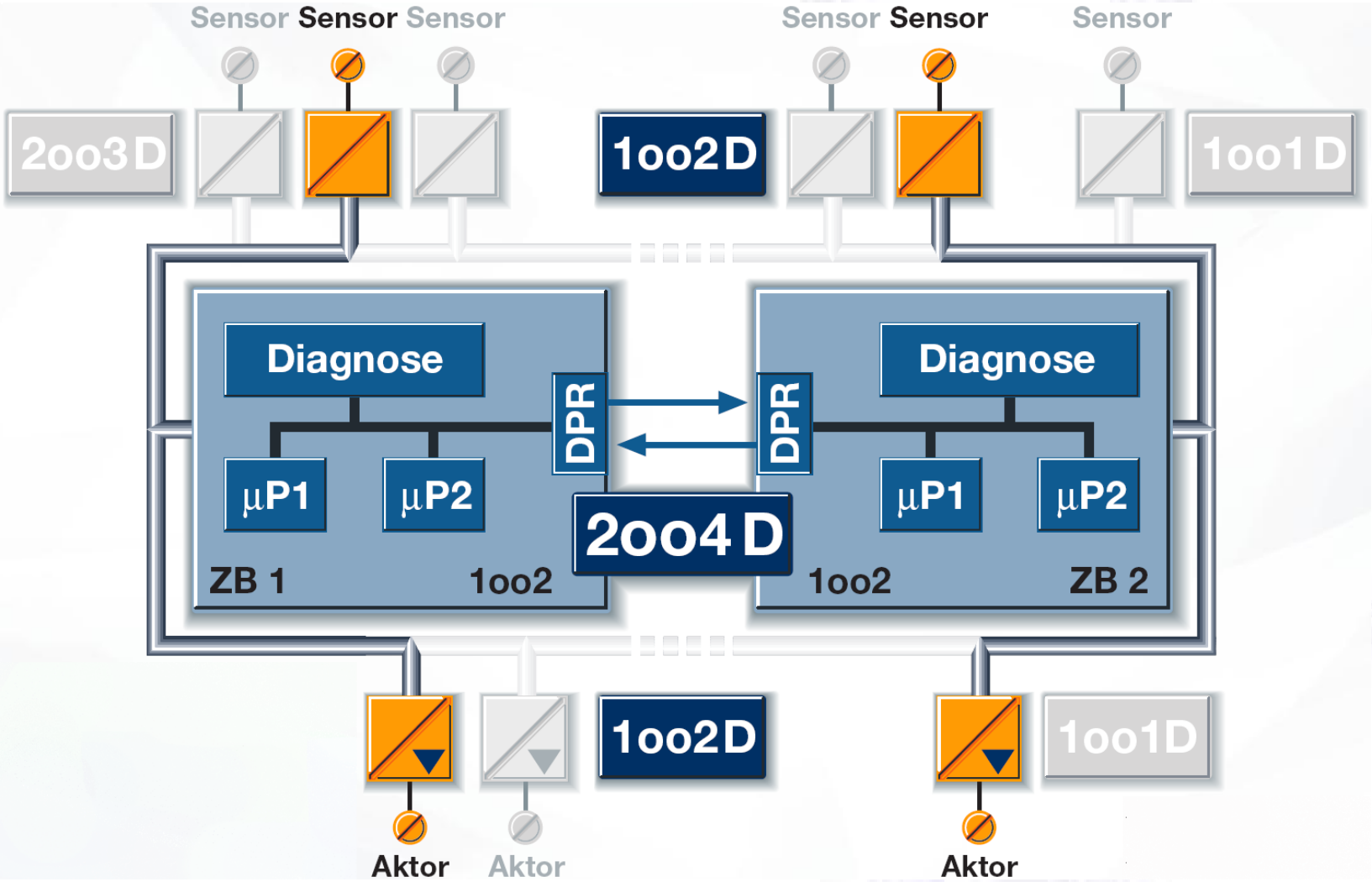
Architekturbezeichnung gemäß IEC 61508

- *M*oo*N* (M out of N)
- *M*oo*N*D (D steht für 'diagnostics')



Ein Vergleich sicherheitsgerichteter SPSen diverser Hersteller wird auf der Internetseite `www.spazint.ru/eng/faq04.htm` geboten.

Die HiQuad-Architektur von HIMA



Neuaufsetzen im laufenden Betrieb

Hauptproblem

- Um systematische Mehrfachausfälle (Common Cause Failures) zu vermeiden, müssen redundante Komponenten räumlich verteilt installiert werden

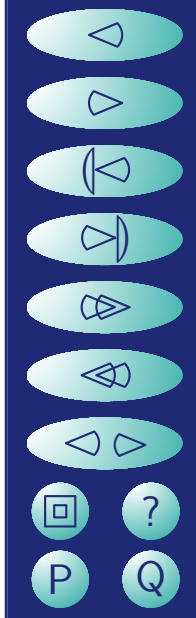
  $\Rightarrow$ realisierbare Übertragungsbandbreite zwischen redundanten Einheiten begrenzt.

- Der Neustart bzw. Austausch einer PES-Einheit im laufenden Betrieb verlangt die Angleichung des internen Zustands zur Laufzeit.

- Während eine neu gestartete PES-Einheit den internen Zustand der redundanten PESe kopiert, ändern diese ihren internen Zustand fortlaufend

  $\Rightarrow$ hoher Kommunikationsaufwand, da geänderte Daten gegebenenfalls wiederholt übertragen werden müssen.

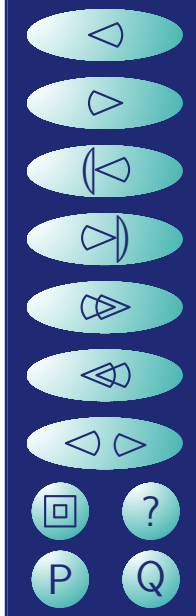
$\Rightarrow$ Kompromiss aus erreichbarer Rechenleistung und notwendiger Übertragungsbandbreite notwendig



Neuaufsetzen im laufenden Betrieb

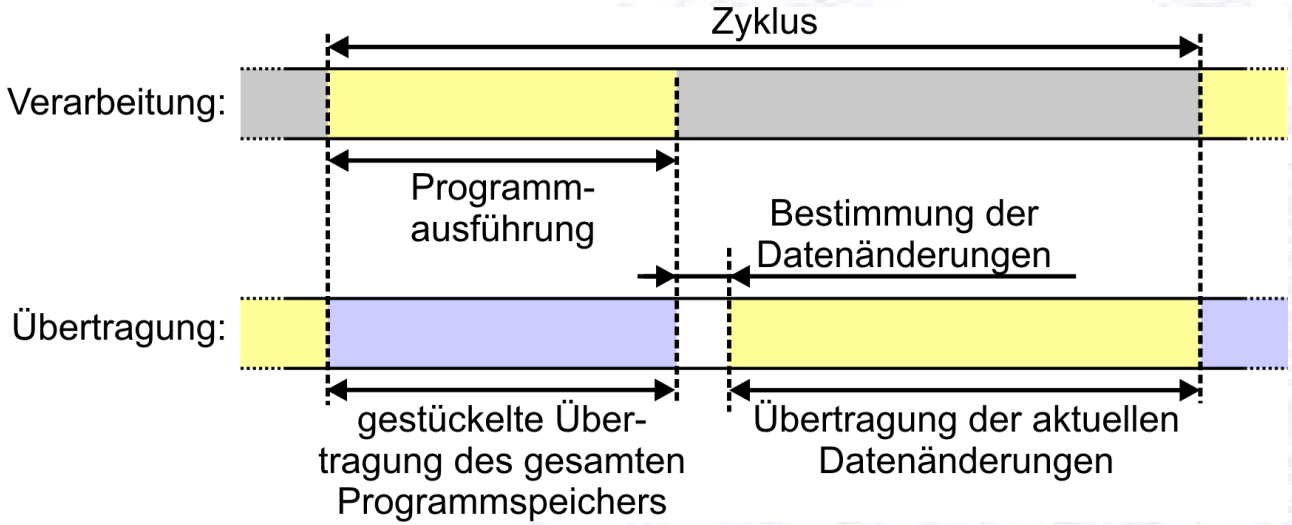
Minimierung des Übertragungsaufwandes:

- Bei den im Arbeitsspeicher abgelegten Werten handelt es sich vielfach nur um Zwischenwerte:
  - $\Rightarrow$ sie sind zum Kopieren des Zustands irrelevant, sofern während des Kopiervorgangs auch die Endergebnisse übertragen werden,
  - $\Rightarrow$ Übertragungsaufwand kann durch zyklische Arbeitsweise minimiert werden
    - Zwischenwerte innerhalb eines Verarbeitungszyklusses zum Neuaufsetzen irrelevant
    - nur Endergebnisse entscheidend
- Nachteil:
  - $\Rightarrow$ Datenänderungen können erst im Anschluss an die Verarbeitung übertragen werden



Neuaufsetzen im laufenden Betrieb

Minimierung des Übertragungsaufwandes:

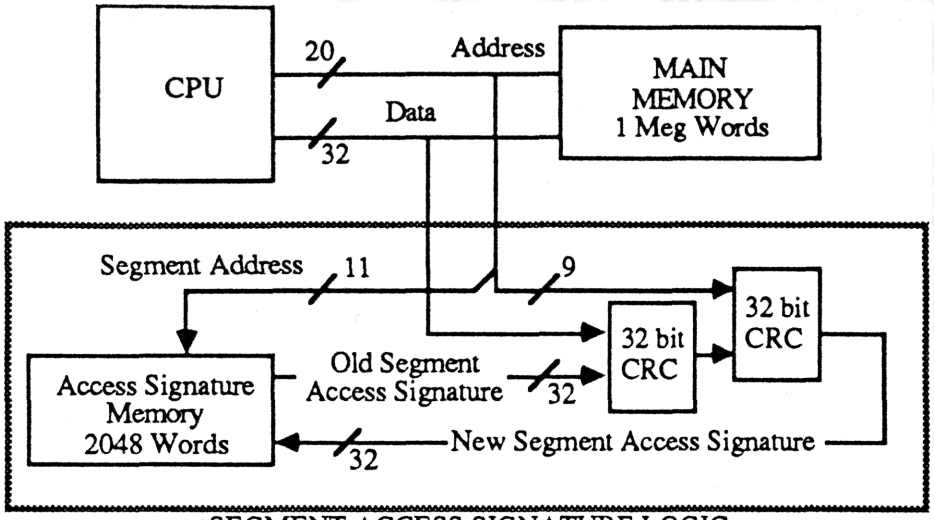


- Übertragungsmedium nutzbar während der Ausführung des Programmcodes zum gestückelten Kopieren des gesamten Speichers in aufeinander folgenden Zyklen

Bekannte Verfahren

Schaltungstechnisch realisiertes Verfahren entwickelt
am *Charles Stark Draper Laboratory*:

- Arbeitsspeicher in Segmente (Speicherseiten) eingeteilt
- Spezielle Zugriffslogik bestimmt bei Speicherzugriffen individuelle *Zugriffssignaturen*
- Beruht auf zyklischer Arbeitsweise
- Durch Signaturvergleich zweier aufeinanderfolgender Zyklen können die veränderten Speichersegmente identifiziert werden.

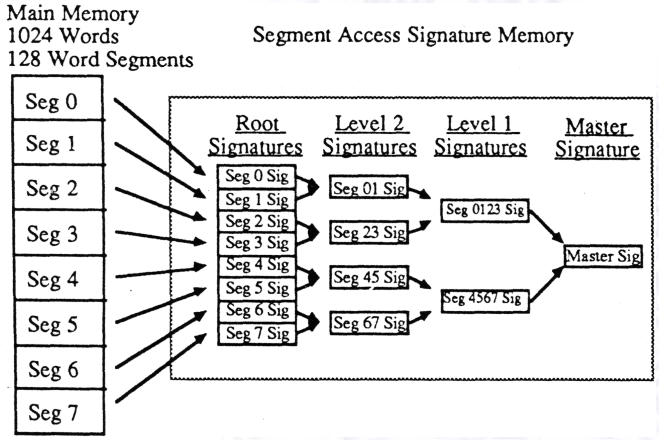


SEGMENT ACCESS SIGNATURE LOGIC

Bekannte Verfahren

Schaltungstechnisch realisiertes Verfahren entwickelt am *Charles Stark Draper Laboratory*:

- Hierarchische Ordnung der Signaturen verringert den Rechenaufwand zur Bestimmung der veränderten Speichersegmente

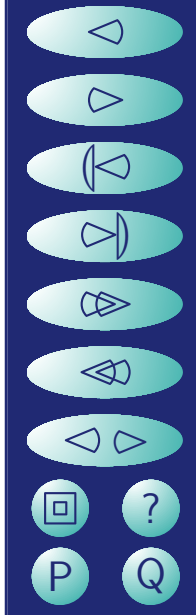


- Kopieren des Systemzustandes:
  - Bestimmung der veränderten Speichersegmente und Übertragung an redundante Einheiten gegen Ende eines jeden Zyklus
  - Restliche Zykluszeit wird genutzt, um gesamten Speicher gestückelt in aufeinanderfolgenden Zyklen zu übertragen.

Bekannte Vefahren

Softwaretechnisch realisiertes Verfahren von Bondavalli et al.:

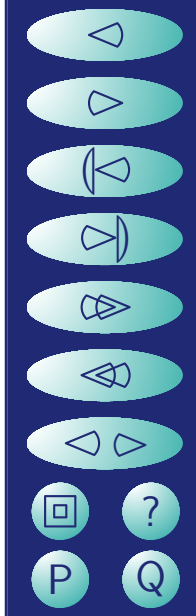
- Es wird ebenfalls von einem zyklisch arbeitenden System ausgegangen.
- Jedem Datenwort ist ein Kennzeichnungsbit zugeordnet.
- Bei einer Datenänderung wird das Kennzeichnungsbit gesetzt; bei Übertragung des Datenwortes an die redundanten Einheiten wird es zurückgesetzt.
- Tritt in einer Einheit ein Verarbeitungsfehler auf, so wird diese neu gestartet und in den anderen Einheiten werden alle Kennzeichnungsbits gesetzt.
- Gegen Ende eines jeden Zyklus übergibt jede Einheit eine feste Anzahl an Datenwörtern mit gesetztem Kennzeichnungsbit an die redundanten Einheiten.
- Der Wiederherstellungsvorgang ist abgeschlossen, wenn alle Kennzeichnungsbits zurückgesetzt sind.



Neuaufsetzen im laufenden Betrieb bei task-orientierten Echtzeitsystemen

- Prinzip der Minimierung des Übertragungsaufwandes durch zyklische Verarbeitung anwendbar:
  - Programmcode der Tasks in Segmente eingeteilt, deren WCET kleiner als die Zykluszeit ist,
  - Task-Verwaltungsfunktionen werden in jedem Zyklus ausgeführt,
  - um kurze Reaktionszeiten zu erzielen, sollte die Zyklusdauer möglichst kurz sein.
- Problem: Minimaler Grenzwert für Anzahl auftretender Datenänderungen innerhalb eines Verarbeitungsintervalls, denn
  - theoretisch können alle Tasks in einem Zyklus durch ein einziges Ereignis gleichzeitig aktiviert werden $\rightarrow$ große Anzahl an Datenänderungen

  $\Rightarrow$ Untere Grenze für die Zyklusdauer



Zugrundeliegende PES-Architektur

- Trennung von Task-Verwaltung und Task-Ausführung

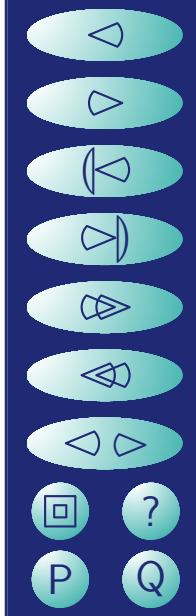
  $\Rightarrow$ unterschiedliche Handhabung von Task-Verwaltungsdaten und Task-Ausführungsdaten (Programmdaten) möglich

- Task-orientierte Verarbeitung in diskreten Verarbeitungsintervallen

  $\Rightarrow$ zu Beginn eines Verarbeitungsintervalls Systemzustand nur durch Task-Verwaltungsdaten und Programmdaten gegeben; Inhalt der Prozessorregister irrelevant

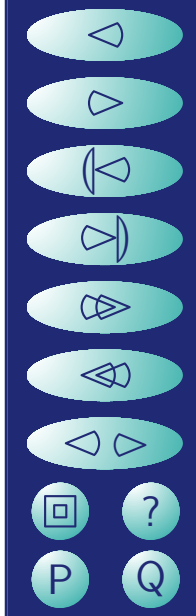
- Task-Verwaltungsfunktionen in Form digitaler Logikschaltung realisiert

  $\Rightarrow$ direkter Zugriff zu Task-Verwaltungsdaten auf Schaltungsebene

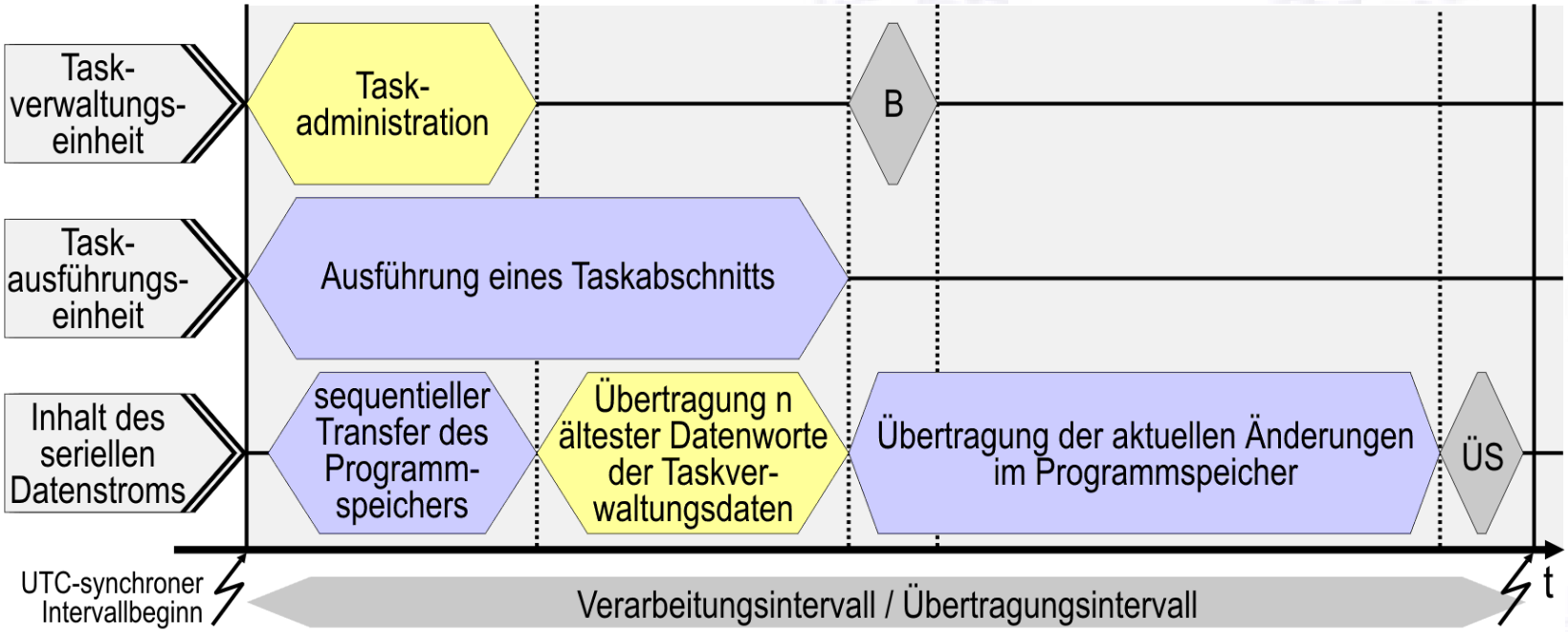


Wiederherstellungskonzept

- Task-Ausführungsdaten (Programmdaten)
  - Begrenzung der Häufigkeit von Datenänderungen beschränkt Rechenleistung
  - Anzahl der in einem Task-Abschnitt zulässigen Datenänderungen begrenzt
  - Im Anschluss an die Verarbeitung werden die Datenänderungen zu den redundanten Einheiten übertragen.
- Task-Verwaltungsdaten
  - Begrenzung der Häufigkeit von Datenänderungen beeinflusst Echtzeitverhalten negativ
  - Übertragung nach dem Prinzip "Älteste Datenwörter zuerst":
    - in jedem Zyklus wird eine feste Anzahl ältester Datenworte übertragen
      $\longrightarrow$ wiederholte Übertragung häufig veränderter Datenworte wird weitgehend vermieden,
    - nur durch schaltungstechnische Integration effizient realisierbar



Aufbau der seriellen Datenströme



B: Bestimmung des im nächsten Intervall auszuführenden Taskabschnitts

ÜS: Übertragung zusätzlicher Statusinformationen

Zusammenfassung

- Neuaufsetzen im laufenden Betrieb verlangt Kompromiss zwischen Rechenleistung und notwendiger Übertragungsbandbreite.
- Durch zyklische Verarbeitung kann eine höhere Rechenleistung bei gleicher Übertragungsbandbreite erreicht werden.
- Problem bei task-orientierten Echtzeitsystemen:
  - einerseits sollte die Zykluszeit möglichst kurz sein
  - andererseits ist die Begrenzung der in einem Zyklus zulässigen Datenänderungen schwierig
    - theoretisch können alle Tasks in einem Zyklus durch ein einziges Ereignis gleichzeitig aktiviert werden

    $\rightarrow$ die damit verbundene Anzahl an Datenänderungen ist nicht reduzierbar
- Lösung:
  unterschiedliche Handhabung von Task-Verwaltungsdaten und Programmdaten
- Übertragung der Task-Verwaltungsdaten nach dem Prinzip "Älteste Datenworte zuerst"