

# Rekonfiguration in Echtzeitsystemen

Master-Seminar AKSS

Adrian Meßner  
04. Juli 2014



LEHRSTUHL FÜR VERTEILTE SYSTEME  
UND BETRIEBSSYSTEME



FRIEDRICH-ALEXANDER  
UNIVERSITÄT  
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

- **Grundlagen**
  - Motivation
  - Grundlagen
  - Moduswechsel mit Anforderungen
- **Synchrone Protokolle**
  - Idle Time
  - Protokolle mit Offset
- **Asynchrone Protokolle**
  - Periodisches Protokoll
  - Nicht-periodisches Protokoll
  - Protokoll mit zwei Offsets
- **Zusammenfassung**



# Agenda

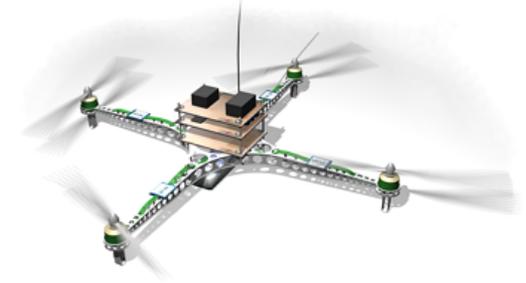
---

- **Grundlagen**
  - Motivation
  - Grundlagen
  - Moduswechsel mit Anforderungen
  
- **Synchrone Protokolle**
  - Idle Time
  - Protokolle mit Offset
  
- **Asynchrone Protokolle**
  - Periodisches Protokoll
  - Nicht-periodisches Protokoll
  - Protokoll mit zwei Offsets
  
- **Zusammenfassung**



# Motivation – Rekonfiguration in Echtzeitsystemen

- **Technische Systeme unter verändernden Umwelteinflüssen**  
→ Betriebsmodus der Anwendung



→ Betriebsmodus der Systems

- Ein Echtzeitsystem muss in der Lage sein, sich diesen Bedingungen anzupassen!
- Ziel: **Moduswechsel** (Anpassung des Aufgabensystems)

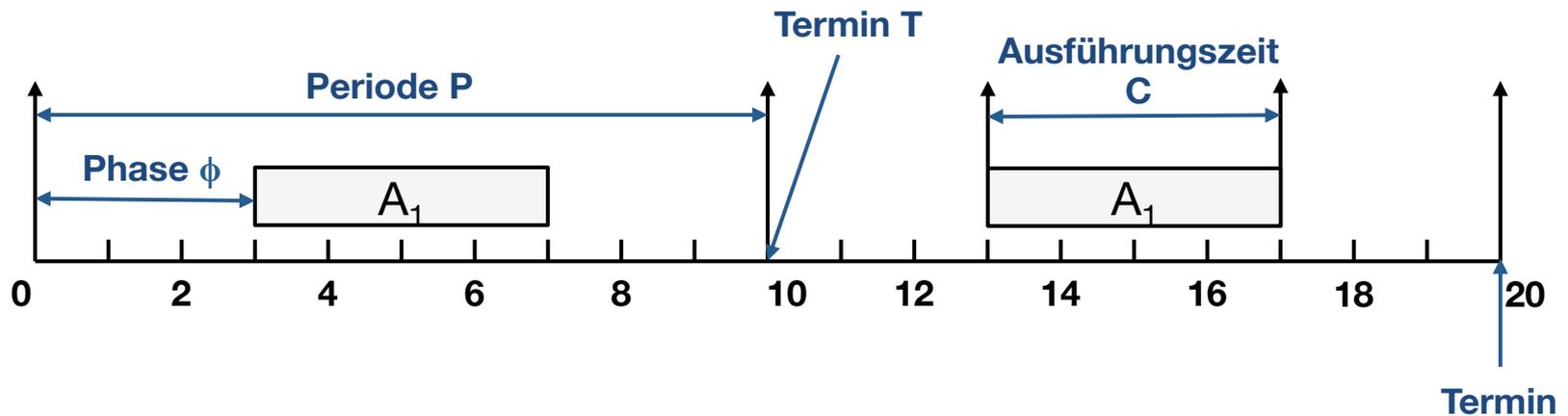


# Wiederholung: Grundlagen der Echtzeitsysteme (1)

- zeitgesteuerte und ereignisgesteuerte Echtzeitsysteme

- **Aufgaben**

- Maximale Ausführungszeit (C, bzw. WCET)
- Periode (T)
- Termin (D)
- Priorität (P)
- Phase ( $\phi$ )



# Wiederholung: Grundlagen der Echtzeitsysteme (2)

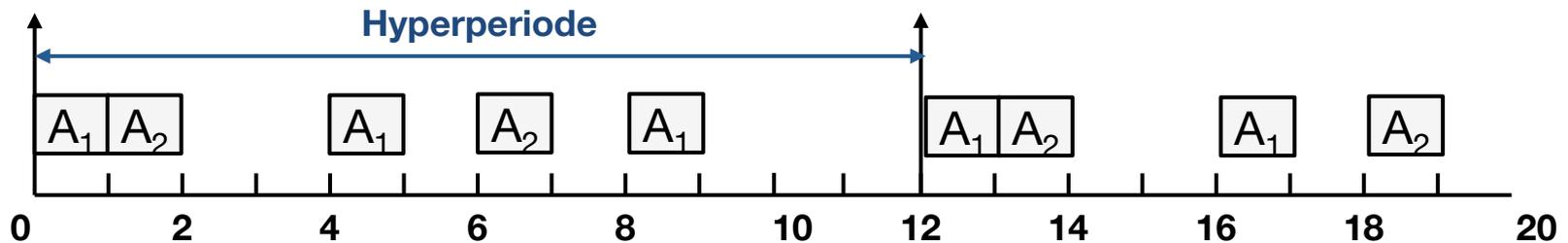
- Weiche, feste, harte Termine

- Hyperperiode

- Bsp.:

$$A_1 : C_1 = 1, T_1 = 4, P_1 = 2$$

$$A_2 : C_2 = 1, T_2 = 6, P_2 = 1$$

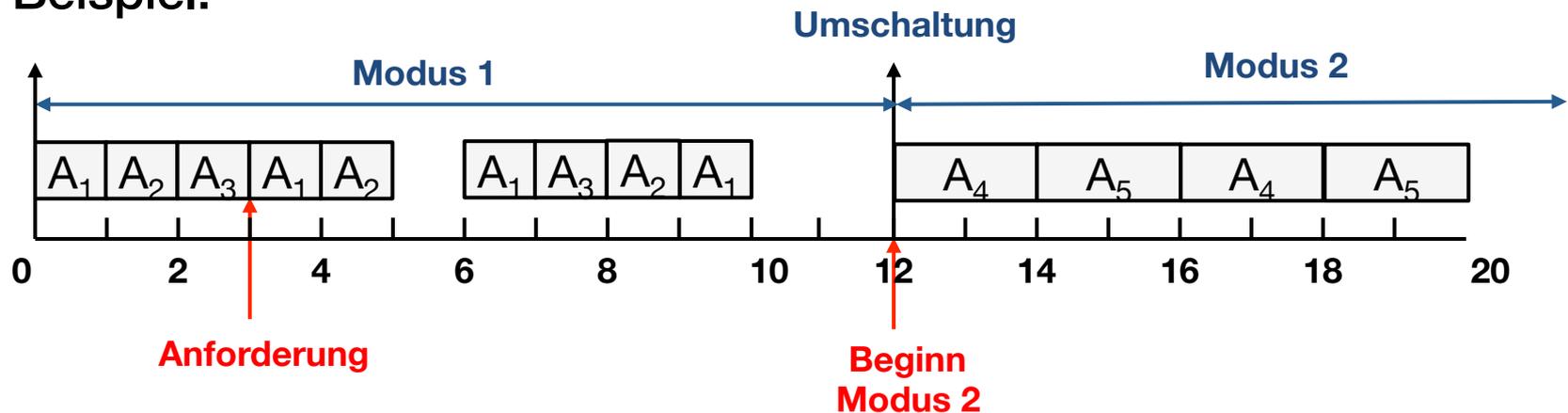


# Rückblick: „one-shot“

## ■ Vorgehen:

- Tausch der Ablaufpläne / Ablauftabellen durch Umschaltung
- Zeitpunkt der Umschaltung nach Ende der Hyperperiode aller Aufgaben

## ■ Beispiel:



## ■ Vorteil:

- keine Abhängigkeiten werden verletzt
- Ressourcenkonflikte durch allokierte Betriebsmittel werden vermieden

## ■ Nachteil:

- Umstellungszeit steigt mit wachsender Anzahl an Aufgaben
- Umstellung kann u.U. sehr lange dauern



# Anforderungen an einen Moduswechsel

---

- **Planbarkeit**

- Es müssen alle Termine eingehalten werden

- **Periodizität**

- Manche Aufgaben müssen immer periodisch ausgeführt werden

- **Unverzögerlichkeit**

- Der Moduswechsel soll schnell vollzogen werden

- **Konsistenz**

- Korrekte Verteilung von gemeinsamen Betriebsmitteln



# Moduswechsel durch explizites Ereignis

---

- **Vermeidung der Probleme des one-shot Ansatzes**
- **Änderung des Betriebsmodus durch einen mode change request (MCR)**
  - Auslösung jederzeit durch eine Aufgabe möglich
  - Übergang zwischen alten und neuen Modus
- **Unterscheidung von Aufgaben im Moduswechsel:**
  - Alte, noch fertigzustellende Aufgaben
  - Alte, abgebrochene Aufgaben
  - Unveränderte Aufgaben
  - Veränderte Aufgaben
  - Vollständig neue Aufgaben
- **Offene Frage: Erstellung eines gültigen Ablaufplanes für den Übergang**



# Beispielsystem (1) - MCR

## ■ Bsp: Aufgabensystem

$A \downarrow a: C \downarrow a = 2, T \downarrow a = 6, P \downarrow a = 1$

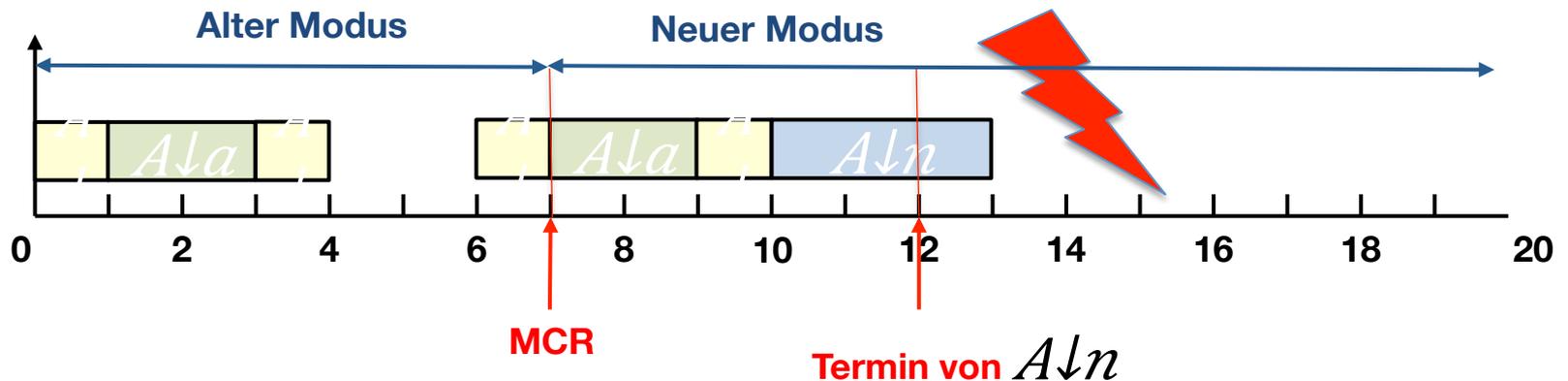
(Aufgabe des alten Modus)

$A \downarrow u: C \downarrow u = 1, T \downarrow u = 3, P \downarrow u = 2$

(unveränderte Aufgabe)

$A \downarrow n: C \downarrow n = 3, T \downarrow n = 6, P \downarrow n = 1$

(Aufgabe des neuen Modus)



- Starten von  $A \downarrow a$  nach  $A \downarrow u$
- Auslösen von MCR nach Beginn von  $A \downarrow a$
- Überlast!  $A \downarrow n$  verpasst seinen Termin!



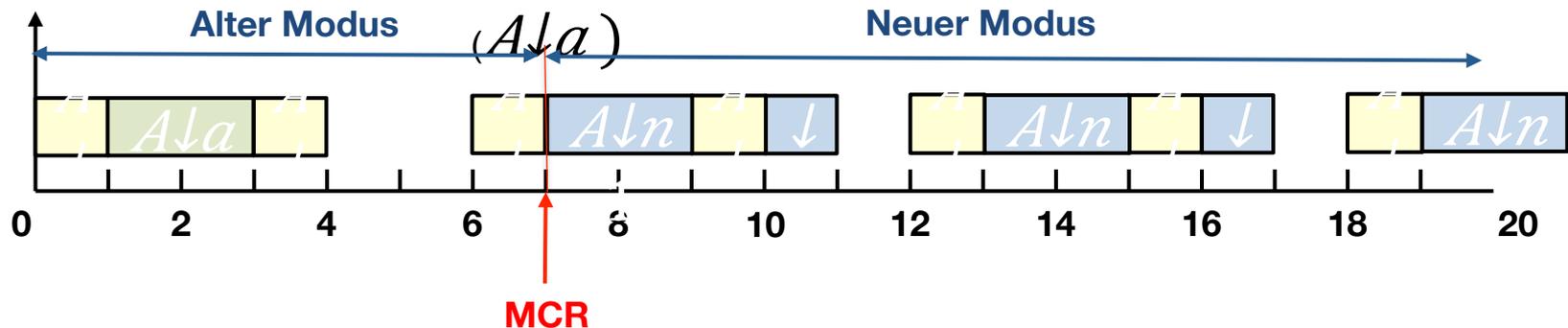
# Beispielsystem (2) - MCR

## ■ Bsp: Aufgabensystem

$A \downarrow a$ :  $C \downarrow a = 2$ ,  $T \downarrow a = 6$ ,  $P \downarrow a = 1$

$A \downarrow u$ :  $C \downarrow u = 1$ ,  $T \downarrow u = 3$ ,  $P \downarrow u = 2$

$A \downarrow n$ :  $C \downarrow n = 3$ ,  $T \downarrow n = 6$ ,  $P \downarrow n = 1$



## ■ Neuer Ansatz: Abbrechen von $A \downarrow a$

- Alle Termine können eingehalten werden

## ■ Aber: Abbruch von alten Aufgaben nicht immer erlaubt!

- Ergebnis von  $A \downarrow a$  darf nicht verworfen werden
- Modifizierung der Parameter von  $A \downarrow a$



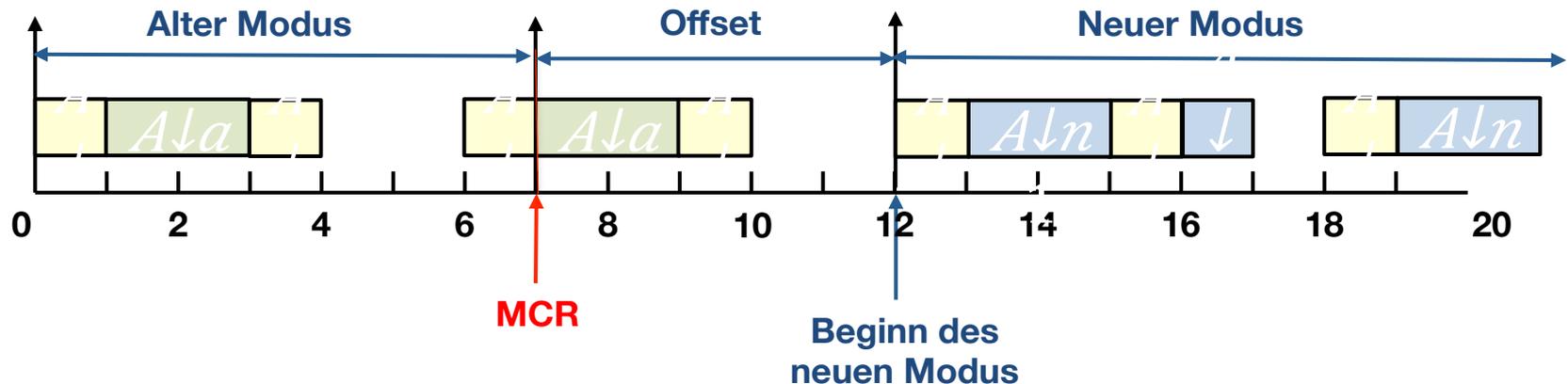
# Beispielsystem (3) - MCR

## ■ Bsp: Aufgabensystem

$A \downarrow a$ :  $C \downarrow a = 2$ ,  $T \downarrow a = 6$ ,  $P \downarrow a = 1$

$A \downarrow u$ :  $C \downarrow u = 1$ ,  $T \downarrow u = 3$ ,  $P \downarrow u = 2$

$A \downarrow n$ :  $C \downarrow n = 3$ ,  $T \downarrow n = 6$ ,  $P \downarrow n = 1$



- **Abhilfe: Einsatz von Offsets für bessere Planbarkeit und Konsistenz**
  - Aber: Verschlechterung von Unverzöglichkeit und Periodizität
- **Anforderungen sind widersprüchlich!**
- **Ziel: Protokoll mit bestmöglichem Kompromiss aus Anforderungen**



- **Daten, die von mehreren Aufgaben gemeinsam genutzt werden**
- **Zugriffskontrolle über Prioritätsobergrenzen**
  - Prioritätsvergabe an Betriebsmittel
  - Prioritätsobergrenze = Priorität der höchstpriorären Aufgabe, die das Betriebsmittel verwendet
  - Betriebsmittel vererbt seine Priorität an die Aufgabe
  - Vermeidung von:
    - unkontrollierte Prioritätsumkehr
    - transitive Blockierungen
    - Deadlocks



# Zugriffskontrolle in Echtzeitsystemen

---

- **Synchrone Protokolle: Umstellung nach Fertigstellung aller alten Aufgaben**
- **Asynchrone Protokolle: Konflikt von Aufgaben aus beiden Modi**
  - Blockieren von Aufgaben des neuen Modus
  - Verletzung der Prioritätsobergrenzen durch neue Aufgaben
  - Prioritäten der Aufgaben sind fest vorgegeben
  - Fest vorgegebene Prioritätsobergrenzen
- **Abhilfe: globale Prioritätsobergrenzen (*ceiling of ceilings*)**
  - Prioritätsobergrenze aller Modi
  - Aber: Blockieren von neuen Aufgaben
  - Lösung:
    - Ändern der Prioritäten von Aufgaben
    - Modifizierte Interfaces für Betriebsmittel



- **Umgang mit Aufgaben aus dem alten Modus:**
  - Fertigstellung
  - Abbruch
  - Gemischt
- **Behandlung unveränderter Aufgaben:**
  - Periodische Protokolle
  - Nicht-periodische Protokolle
- **Kombination von Aufgaben aus beiden Modi**
  - Synchrone Protokolle
  - Asynchrone Protokolle



## ■ **Synchrone Protokolle**

- Idle Time Protokoll
- Single Offset Protokolle
  - Maximum-period offset
  - Minimum single offset (nichtperiodisch)
  - Minimum single offset (periodisch)

## ■ **Asynchrone Protokolle**

- Asynchrones Protokoll mit Periodizität
- Nicht-periodisches asynchrones Protokoll
- Asynchrones Protokoll mit zwei Offsets



# Agenda

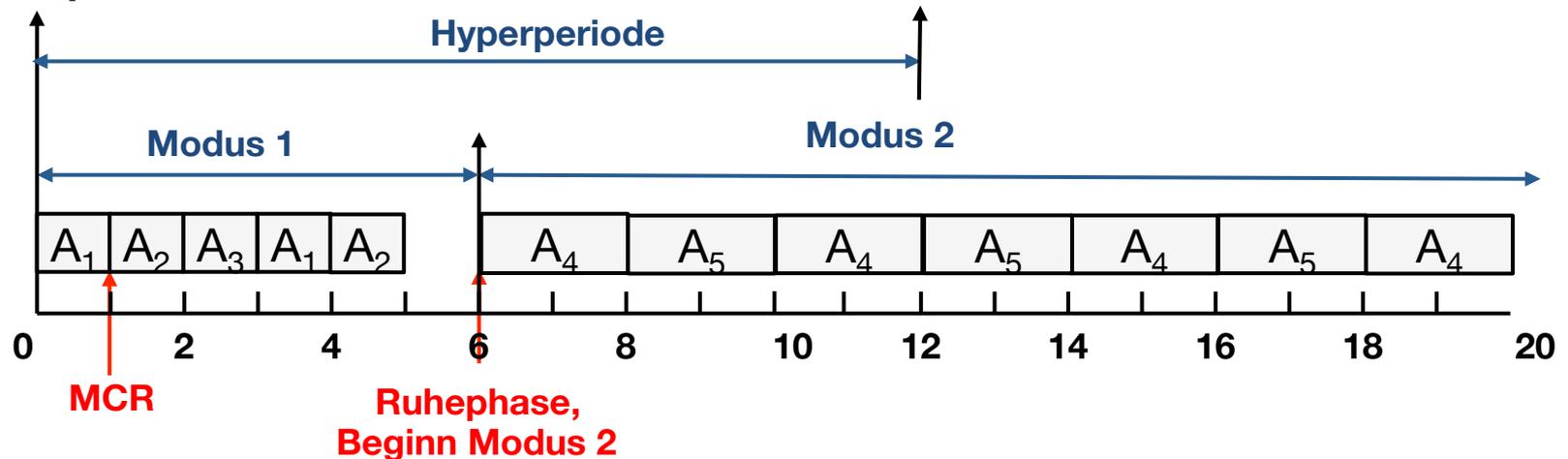
---

- **Grundlagen**
  - Motivation
  - Grundlagen
  - Moduswechsel mit Anforderungen
- **Synchrone Protokolle**
  - Idle Time
  - Protokolle mit Offset
- **Asynchrone Protokolle**
  - Periodisches Protokoll
  - Nicht-periodisches Protokoll
  - Protokoll mit zwei Offsets
- **Zusammenfassung**



# Idle Time Protokoll

- Auslösung alter Aufgaben wird nach einem MCR nicht beeinflusst
- Neuer Modus wird in der Ruhephase eingeleitet
- Bsp.:

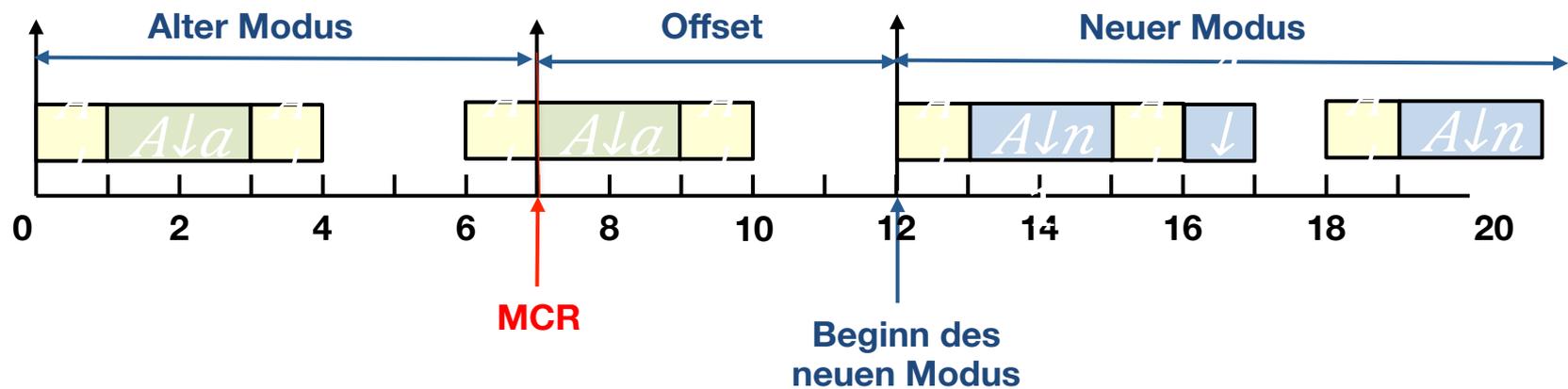


- Vorteile:
  - Sehr einfach
  - Keine spezielle Planbarkeitsanalyse erforderlich
  - Keine Konflikte mit Betriebsmittel
- Nachteile:
  - Verbesserte, aber immer noch lange Umstellungsdauer



# Maximum-period offset Protokoll

- Nach MCR: Verschieben aller neuen Aufgaben um ein Offset
  - größter Periode aller Aufgabe aus beiden Modi
- Keinen Einfluss auf unveränderte Aufgaben

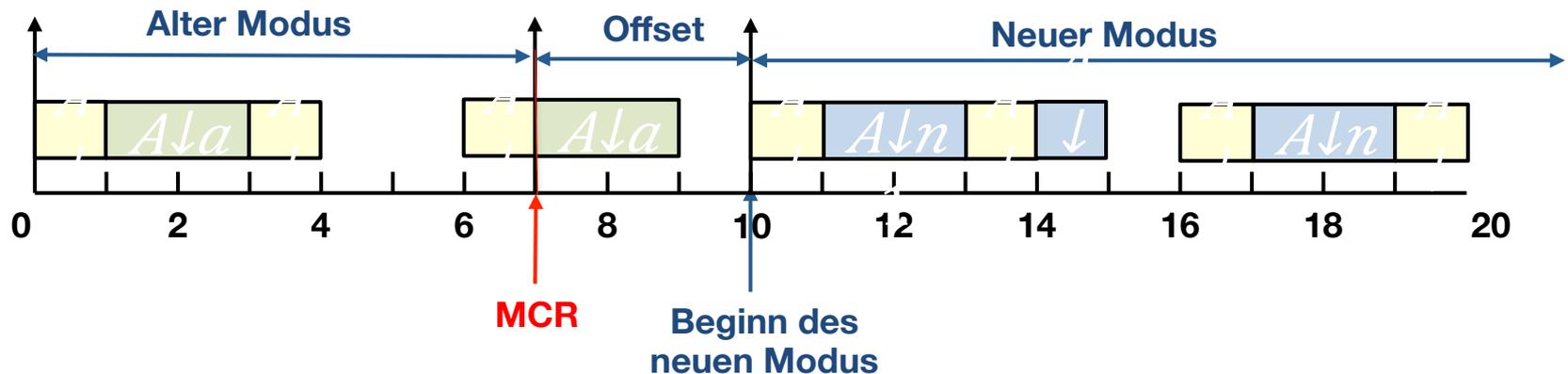


- Vorteile:
  - Keine spezielle Planbarkeitsanalyse erforderlich
- Nachteile:
  - Umstellung dauert lange
  - Blockierung durch Prioritätsobergrenzen



# Minimum single offset (nicht-periodisch)

- Basis: Maximum-Period Offset Protokoll
- Fertigstellung begonnener Aufgaben aus altem Modus
- Verzögerung von Aufgaben aus dem neuen Modus durch Offset

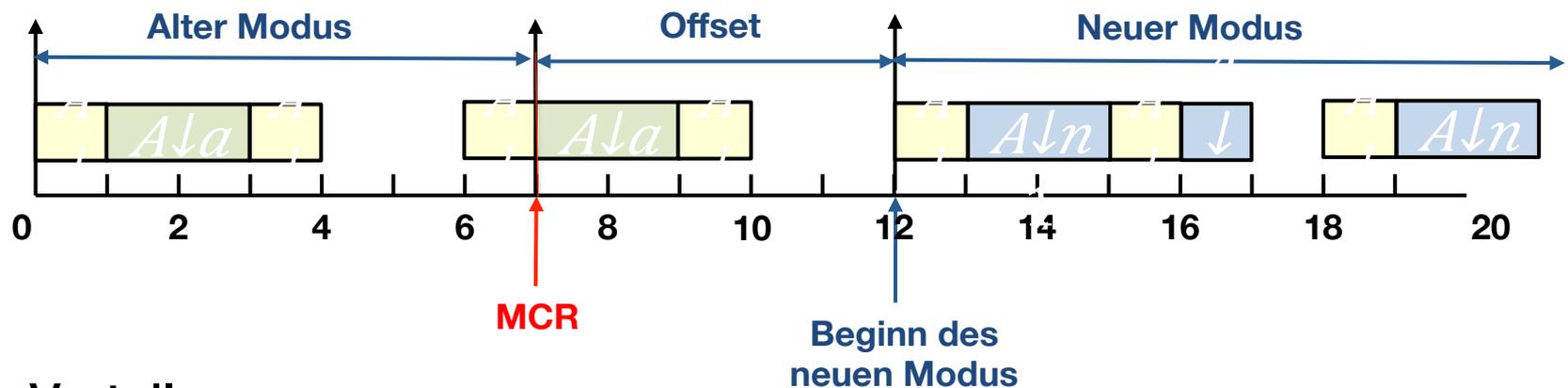


- Vorteile:
  - Schnellere Umstellung
  - Keine Konsistenzprobleme
- Nachteile:
  - Nicht Einhalten von Periodizität



# Minimum single offset (periodisch)

- Basis: Minimum single offset Protokoll (nichtperiodisch)
- Weiterhin periodische Ausführung von unveränderten Aufgaben



- **Vorteil:**
  - Gewährleistung von Periodizität
- **Nachteil:**
  - Längere Umstellungsdauer
  - Gefahr von Konsistenzproblemen



# Agenda

---

- **Grundlagen**
  - Motivation
  - Grundlagen
  - Moduswechsel mit Anforderungen
  
- **Synchrone Protokolle**
  - Idle Time
  - Protokolle mit Offset
  
- **Asynchrone Protokolle**
  - Periodisches Protokoll
  - Nicht-periodisches Protokoll
  - Protokoll mit zwei Offsets
  
- **Zusammenfassung**



# Asynchrones Protokoll mit Periodizität (1)

---

## ■ Verfahren:

- Alte Aufgaben werden fertiggestellt
- Unabhängige Ausführung von unveränderten Aufgaben
- Veränderte Aufgaben werden nach ihrer alten Periode fortgesetzt
- Neue Aufgaben werden nach einem Offset gestartet

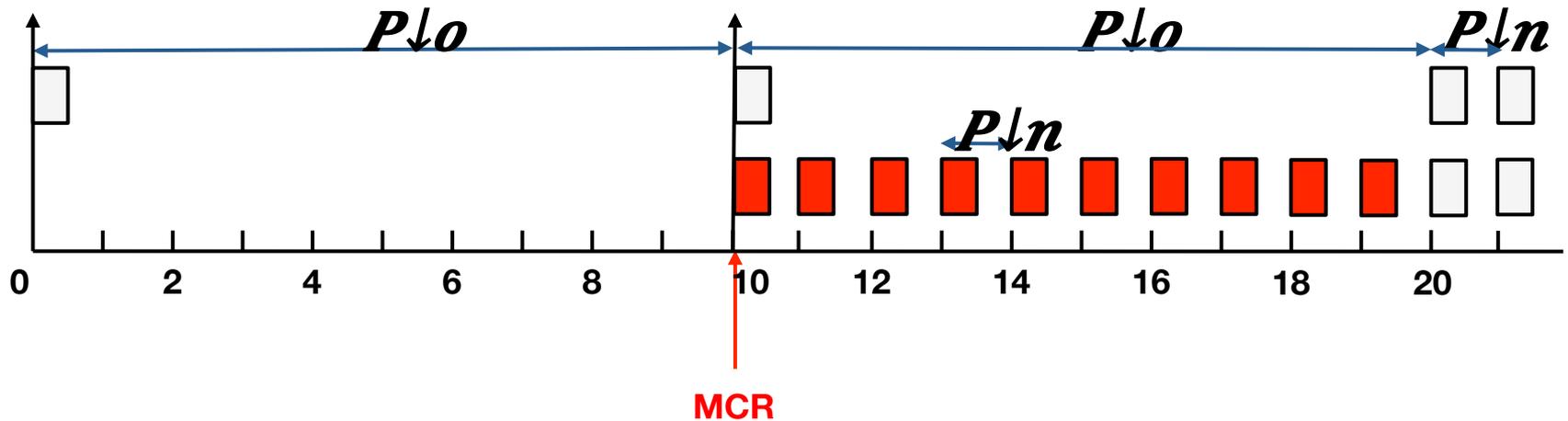
## ■ Analyse:

- Unterteilung der Aufgaben in Gruppen
- Bilden von Zeitfenstern
- Aber: Analyse ist pessimistisch



# Asynchrones Protokoll mit Periodizität (2)

- Beispiel: starke Verkürzung der Periode



- **Vorteil:**
  - Geeignet für unveränderte Aufgaben
- **Nachteil:**
  - Problem bei Verkürzung der Periode

# Nichtperiodisches Asynchrones Protokoll (1)

---

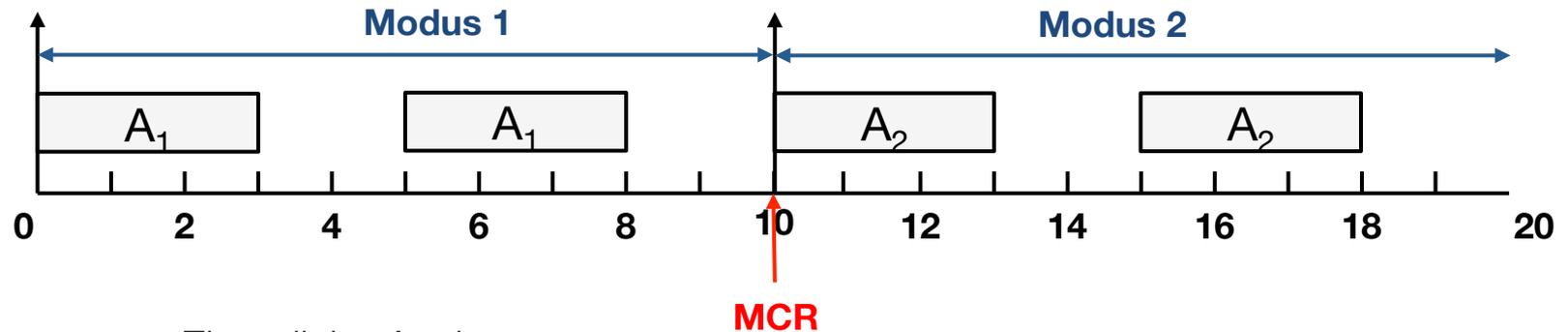
- **Alle Aufgaben im neuen Modus können durch ein Offset verzögert werden**
- **Vorteile:**
  - Immer korrekter Ablaufplan möglich
  - Vereinfachte Planbarkeitsanalyse
  - Keine Konflikte mit Betriebsmitteln
  - Option alte Aufgaben anzubrechen



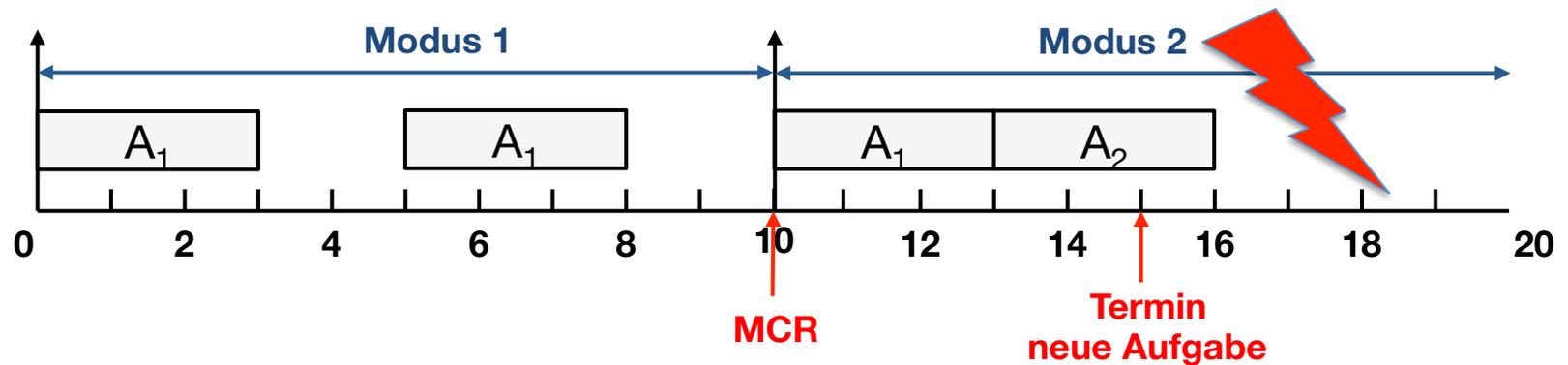
# Nichtperiodisches Asynchrones Protokoll (2)

## ■ Nachteile:

- Pessimistische Analyse
  - Erwartete Analyse:



- Eigentliche Analyse:



- Keine Berücksichtigung von Blockierungszeiten
- Keine exakten Offsets

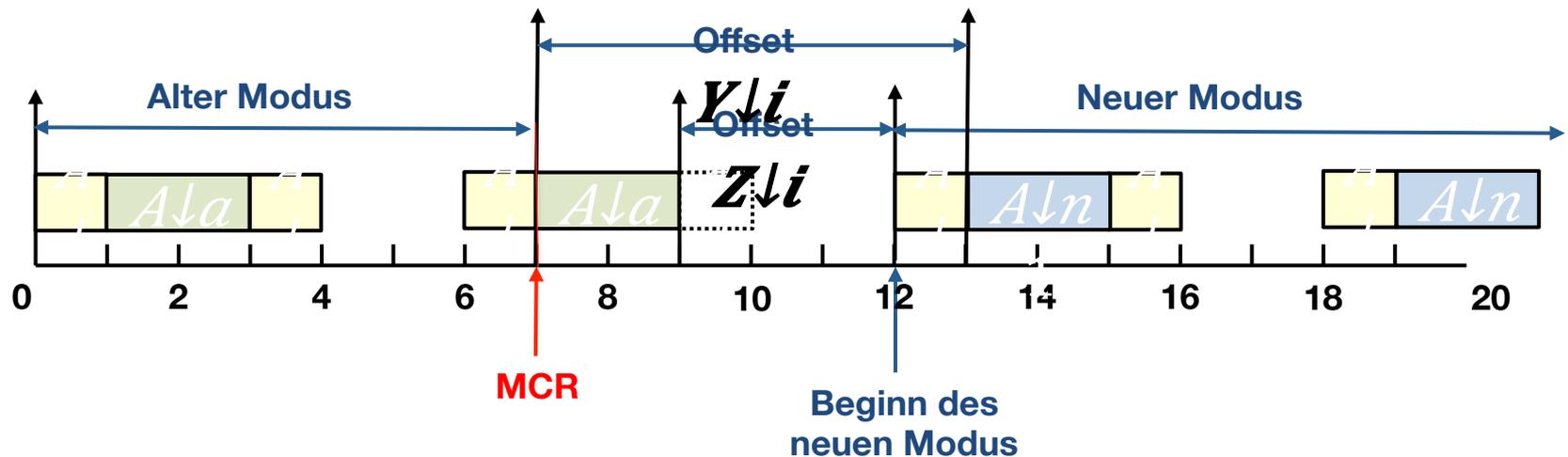


# Asynchrones Protokoll mit zwei Offsets (1)

## ■ Vorgehen:

- Optionale Fertigstellung von alten Aufgaben
- Veränderte Aufgaben werden nach Offset  $Y\downarrow i$  gestartet
- Neue Aufgaben werden nach Offset  $Z\downarrow i$  gestartet
- Unveränderte Aufgaben werden nach Offset  $Z\downarrow i$  gestartet
  - Falls Offset nicht benötigt:  $Z\downarrow i = 0$

## ■ Beispielsystem:



# Asynchrones Protokoll mit zwei Offsets (2)

---

## ■ Vorteile:

- Immer gültiger Ablaufplan möglich
- Gewährung von Konsistenz
- Flexibler und schneller Übergang

## ■ Nachteile:

- Evtl. nicht Einhalten von Periodizität



# Agenda

---

- **Grundlagen**
  - Motivation
  - Grundlagen
  - Moduswechsel mit Anforderungen
- **Synchrone Protokolle**
  - Idle Time
  - Protokolle mit Offset
- **Asynchrone Protokolle**
  - Periodisches Protokoll
  - Nicht-periodisches Protokoll
  - Protokoll mit zwei Offsets
- **Zusammenfassung**



- **Moduswechsel in Echtzeitsystemen benötigen Protokolle zur Durchführung**
- **Betriebsmittel müssen bei der Übergangsphase kontrolliert verteilt werden**
  
- **Synchrone Protokolle:**
  - Einfach, keine spezielle Planbarkeitsanalyse
  - Keine Garantie von Periodizität und Schnelligkeit
  
- **Asynchrone Protokolle:**
  - Schnell, periodisch / nicht-periodisch
  - Spezielle Planbarkeitsanalyse erforderlich
  - Asynchrones Protokoll mit zwei Offsets als bisher effektivstes Protokoll

# Offene Fragen

---

- Antwortzeiten der Protokolle?
- Bewertung der Protokolle?
- Optimierung von Offsets?
- ...



