

KNr.

MNr.

Zuname, Vorname

Ges.)(100)

1.)(25)

2.)(30)

3.)(15)


4.)(30)

Zusatzblätter:

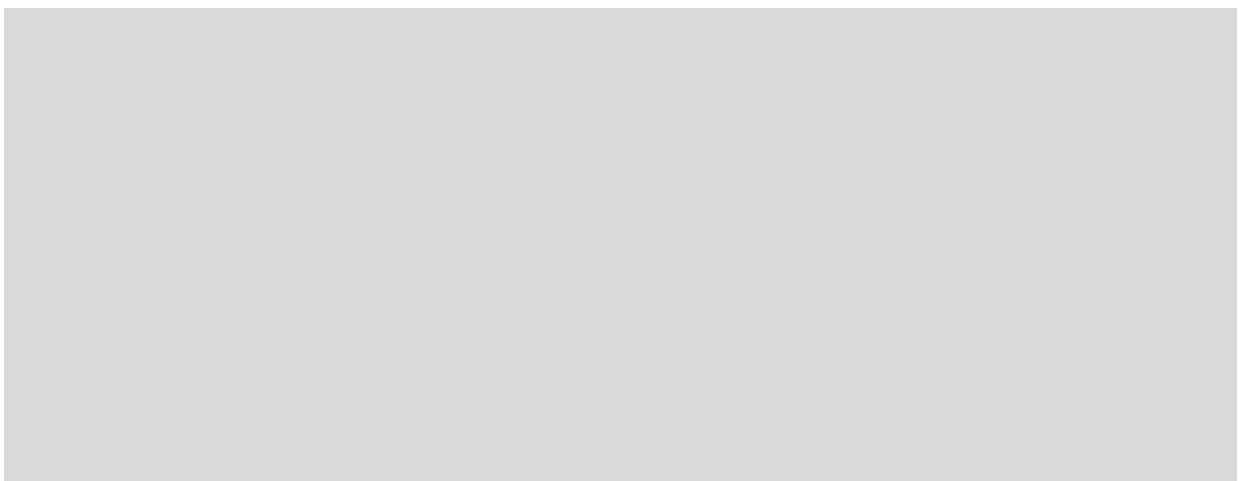
**Bitte verwenden Sie nur dokumentenechtes Schreibmaterial!**

## 1 Grundlagen (25)

Was versteht man unter einem *H-State* und einem *G-State*? Beschreiben Sie die Bedeutung dieser beiden Konzepte. (4)



Welche Eigenschaften müssen beschrieben werden, um das Echtzeit-Interface (Linking Interface) zwischen Subsystemen eines Echtzeitsystems vollständig zu characterisieren? (4)



Wie lautet die *Reasonableness Condition* für eine globale Zeitbasis? Welche Konsequenzen hat eine Verletzung der Reasonableness Condition? (4)

A large, solid gray rectangular area intended for the student to write their answer to the first question.

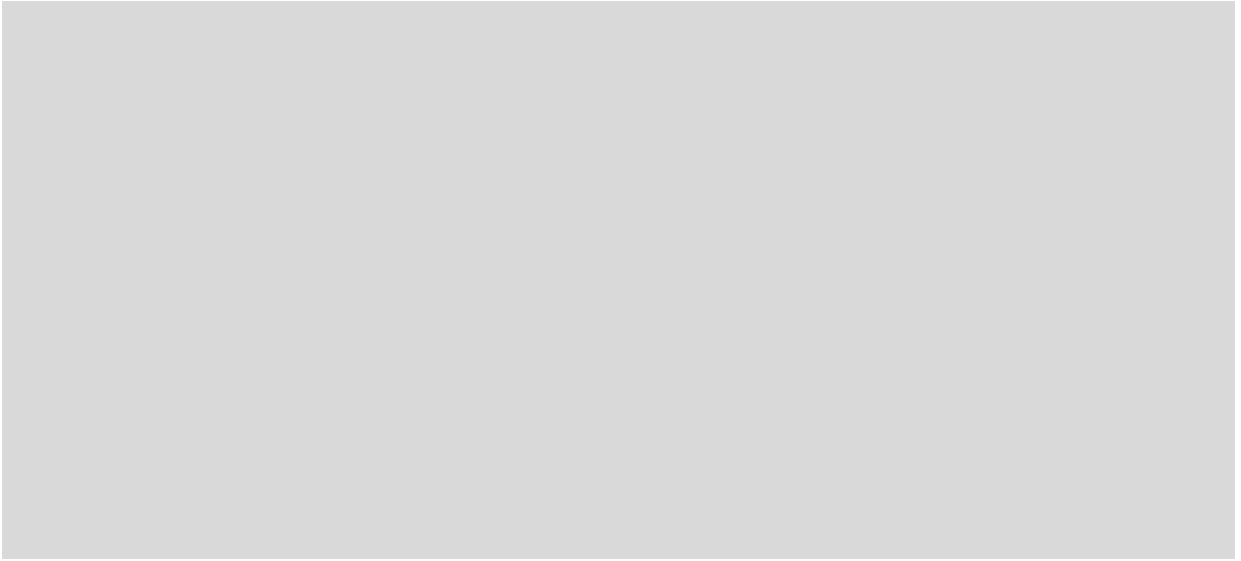
In welchem Verhältnis stehen *kausale* und *temporale Ordnung*? (3)

A large, solid gray rectangular area intended for the student to write their answer to the second question.

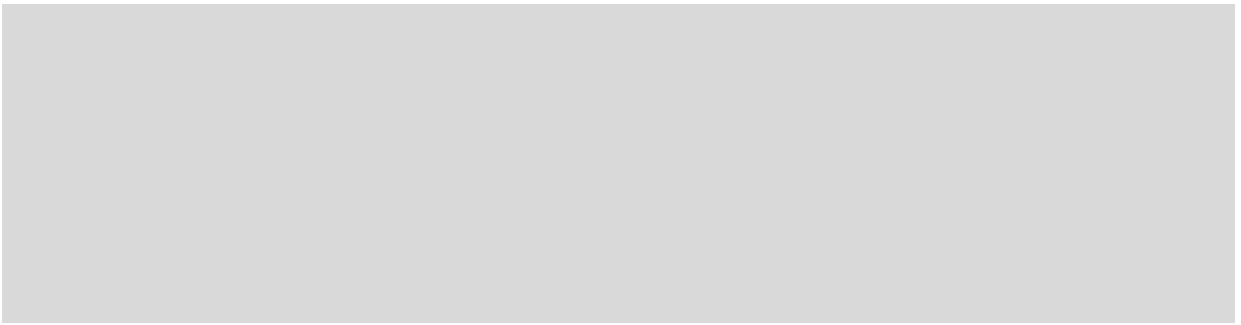
Erklären Sie die Funktionsweise des *Central Master Algorithm* zur Uhrensynchronisation. Welche *Precision* kann mit diesem Algorithmus erreicht werden? (5)

A large, solid gray rectangular area intended for the student to write their answer to the third question.

Erklären Sie die Funktionsweise eines zeitgesteuerten Kommunikationsprotokolls. (3)



Was versteht man unter der Idempotenz einer Nachricht? (2)



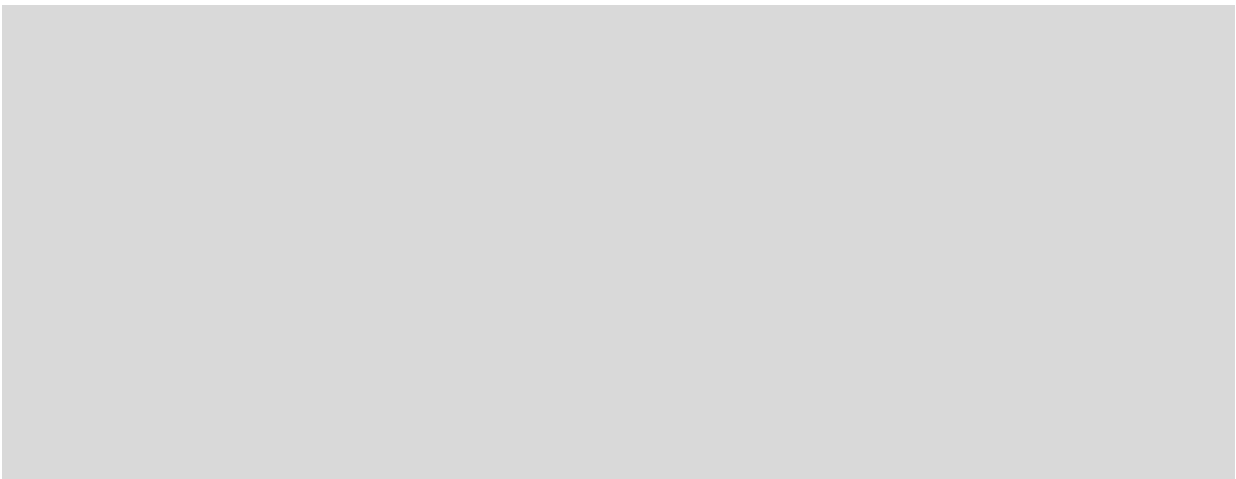
## 2 Scheduling (30)

Gegeben sind drei Echtzeittasks, die nach zwei verschiedenen Schedulingverfahren (Rate Monotonic Scheduling bzw. Earliest Deadline First Scheduling) abgearbeitet werden sollen. Die Perioden und Ausführungszeiten der drei Tasks sind in der folgenden Tabelle angegeben.

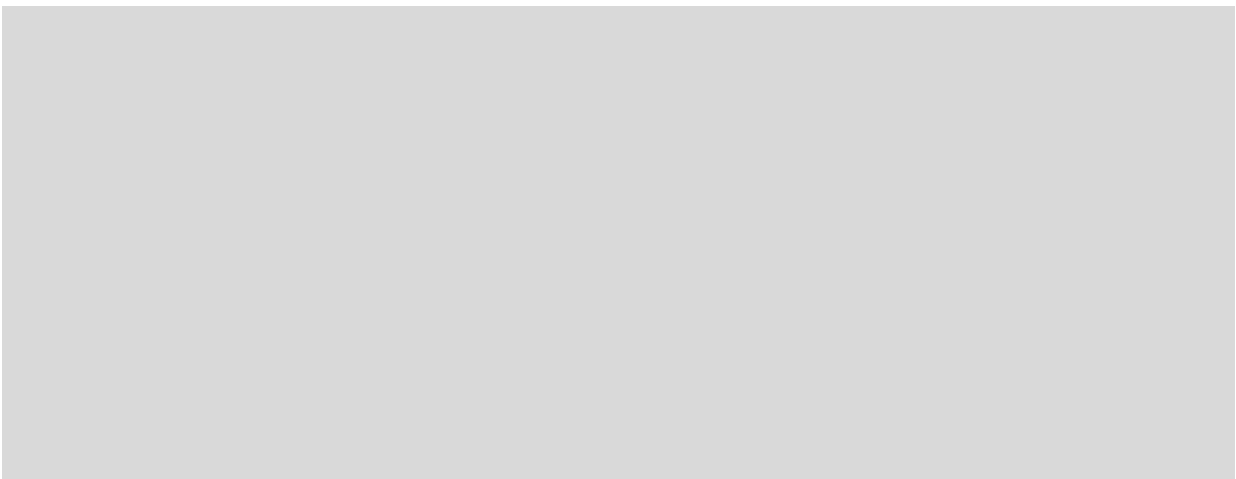
|         | <u>Periode <math>T_i</math></u> | <u>Ausführungszeit <math>C_i</math></u> |
|---------|---------------------------------|-----------------------------------------|
| Task 1: | 3                               | 1                                       |
| Task 2: | 4                               | 1                                       |
| Task 3: | 5                               | 2                                       |

Führen Sie für das Task Set die notwendigen und hinreichenden *Utilization-Based Schedulability Tests* für (a) *Rate Monotonic Scheduling* und (b) *Earliest Deadline First Scheduling* durch.

Rate Monotonic Scheduling:



Earliest Deadline First Scheduling:



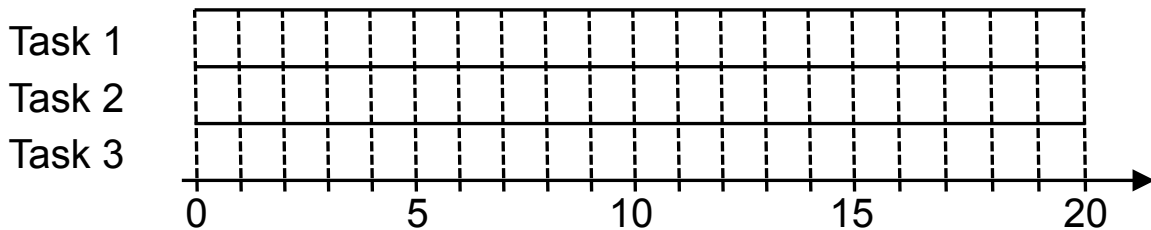
Welche Aussagen können Sie aufgrund der Ergebnisse der Schedulability Tests über das gegebene Task Set machen?

Rate Monotonic Scheduling:

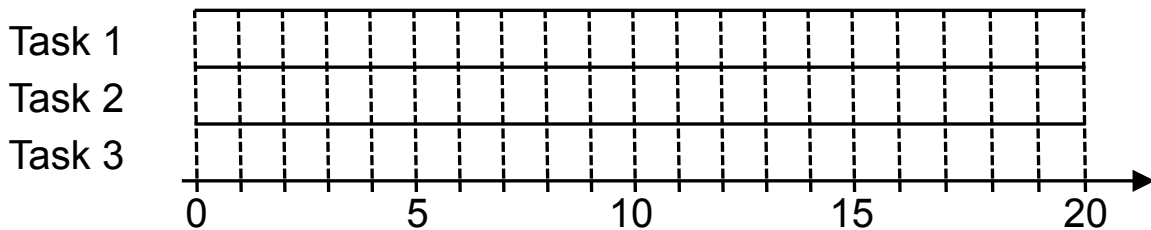
Earliest Deadline First Scheduling:

Tragen Sie in den folgenden Diagrammen ein, wie das gegebene Task Set nach den beiden Schedulingstrategien abgearbeitet wird. Nehmen Sie dabei an, dass die erste Aktivierung aller drei Tasks zum Zeitpunkt Null erfolgt. Tragen Sie in jedem Diagramm für jede Zeiteinheit den jeweils laufenden Task ein. Sollte es zur Verletzung einer Task-Deadline kommen, dann zeichnen Sie dies deutlich ein. Sie brauchen in diesem Fall das Schedule nicht fortzusetzen.

Rate Monotonic Scheduling:

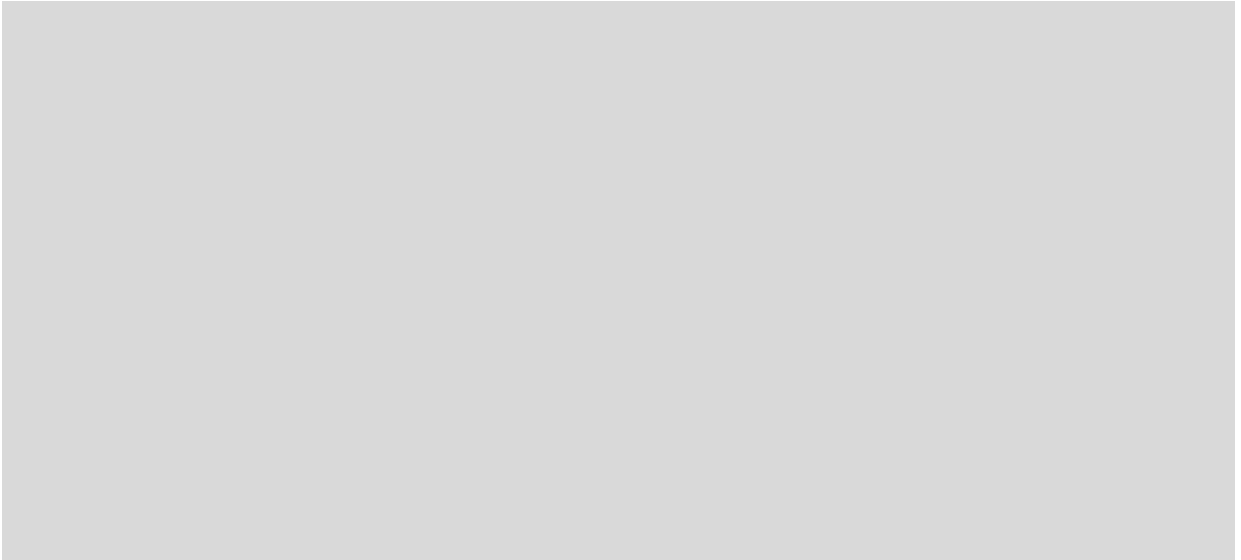


Earliest Deadline First Scheduling:

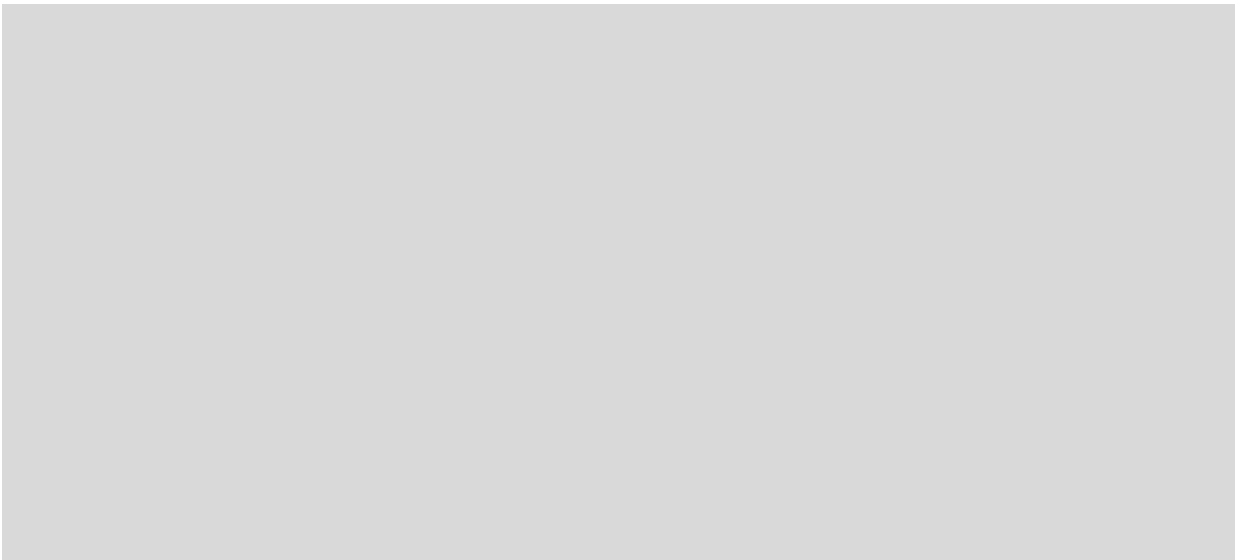


### 3 Rechenaufgaben (15)

In einem Echtzeitnetzwerk mit 500 Metern Länge werden die Daten mit 10Mbit/s übertragen. Die Nachrichtenlänge beträgt 50 Bit. Berechnen Sie die maximale Auslastung (Protocol Efficiency) des gegebenen Netzwerks. (8)



Gegeben sind die Parameter eines sporadischen Tasks:  $mint_s = 50ms$ ,  $D_s = 40ms$  und  $C_s = 2ms$ . Der sporadische Task soll in einen quasiperiodischen Task umgewandelt werden. Berechnen Sie Deadline  $D_P$  und Periode  $T_P$  für den quasiperiodischen Task. (7)



## 4 WCET Analyse (30)

Die Abbildung zeigt ein Programmstück, für das die WCET Analyse mittels *Implicit Path Enumeration* durchgeführt werden soll. Für jedes Statement bzw. für jede Bedingung ist neben dem Code die Ausführungszeit angegeben. Die maximale Iterationsanzahl der Schleife und eventuelle andere Constraints leiten Sie aus dem Code ab.

|                      |                          |
|----------------------|--------------------------|
| sum1 = 0;            | $t_{\text{simple}} = 10$ |
| sum2 = 0;            | $t_{\text{simple}} = 10$ |
| count = 0;           | $t_{\text{simple}} = 10$ |
| while (count < 5) {  | $t_{\text{cond}} = 16$   |
| if (count < 3) {     | $t_{\text{cond}} = 12$   |
| sum1 = sum1 + 1;     | $t_{\text{simple}} = 20$ |
| } else {             |                          |
| sum2 = sum2 + count; | $t_{\text{simple}} = 22$ |
| }                    |                          |
| count = count + 1;   | $t_{\text{simple}} = 18$ |
| }                    |                          |
| count = 0;           | $t_{\text{simple}} = 10$ |

Geben Sie einen Flussgraphen für das WCET-Problem an und ermitteln Sie daraus die Zielfunktion sowie die Flow Constraints zur WCET Berechnung mittels *Implicit Path Enumeration*. Die Zielfunktion für den von ihnen konstruierten Graphen soll möglichst konkret angegeben werden.

