

KNr.

MNr.

Zuname, Vorname

Ges.)(100)

1.)(35)

2.)(25)

3.)(15)

4.)(25)

Zusatzblätter:

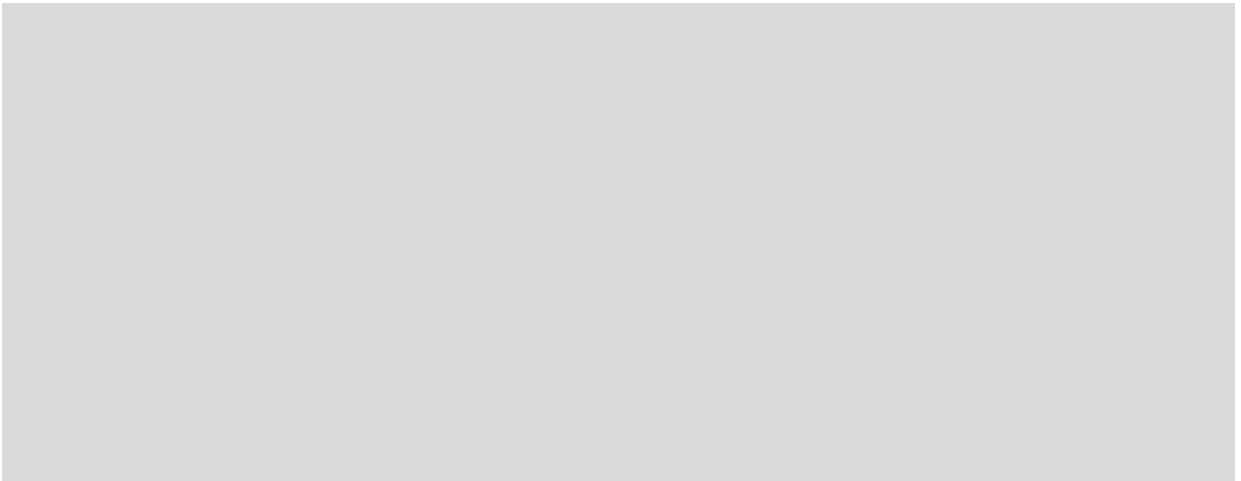
**Bitte verwenden Sie nur dokumentenechtes Schreibmaterial!**

## 1 Grundlagen (35)

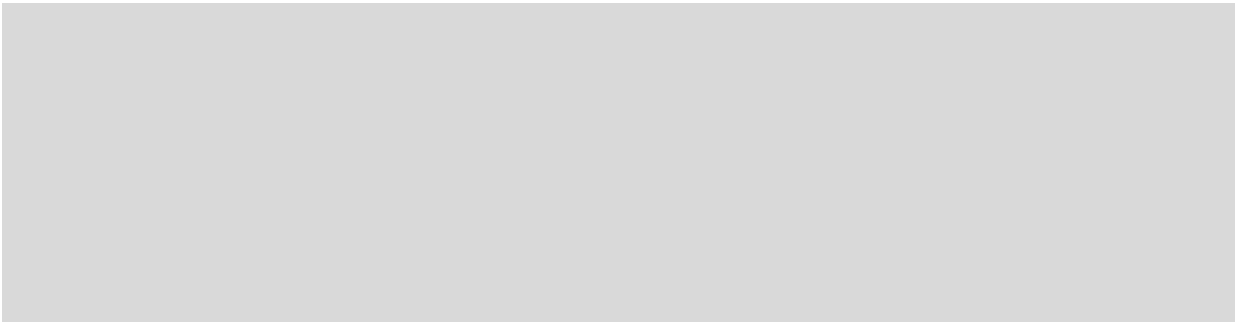
Was versteht man unter einer *Lamport Clock*? Erklären Sie deren Funktionsweise und beschreiben Sie, in welcher Hinsicht *Vector Clocks* den *Lamport Clocks* überlegen sind. (5)

Zeigen Sie anhand eines Beispiels, dass zwei oder mehrere Cluster, die jeweils eine  *$g/3g$ -sparse* Zeitbasis verwenden, nicht so kommunizieren können, dass die Ordnung aller Events auf der Empfängerseite immer eindeutig rekonstruiert werden kann. (4)

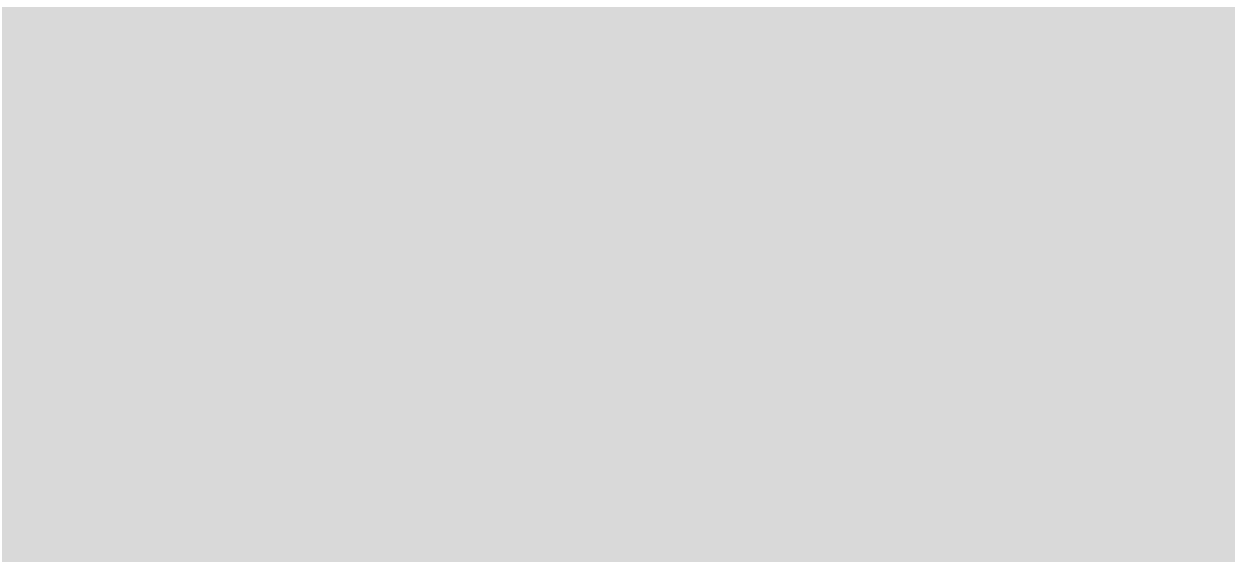
Welche Präzision ist bei der Uhrensynchronisation mittels *Fault Tolerant Average Algorithmus* zu erreichen? Geben Sie die Formel an und beschreiben sie diese. (4)



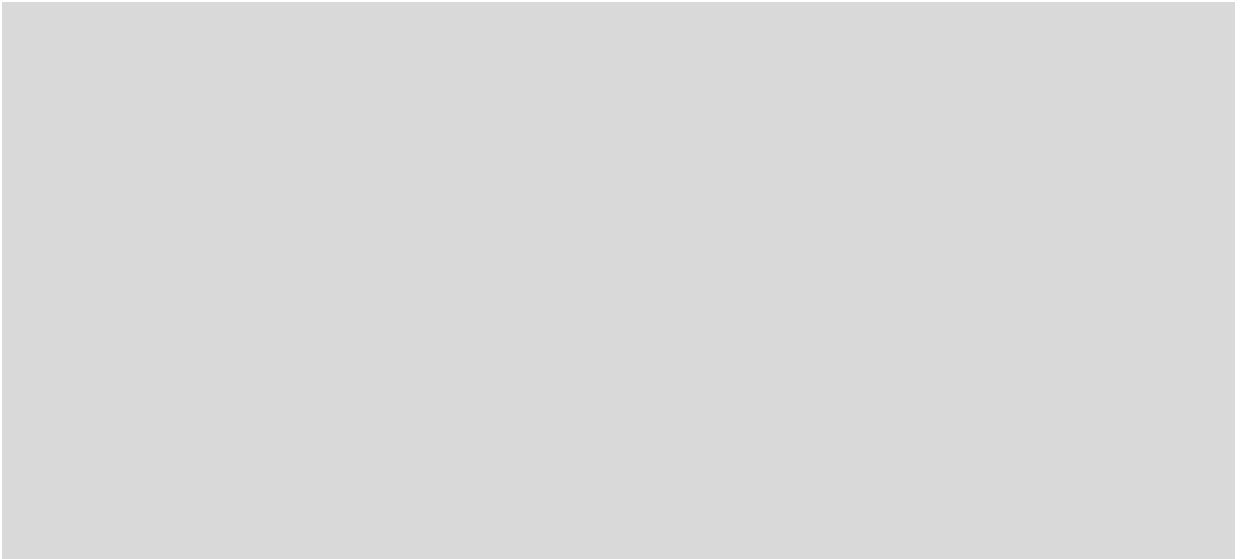
Wie funktioniert *Cristian's Algorithmus* zur Uhrensynchronisation? (3)



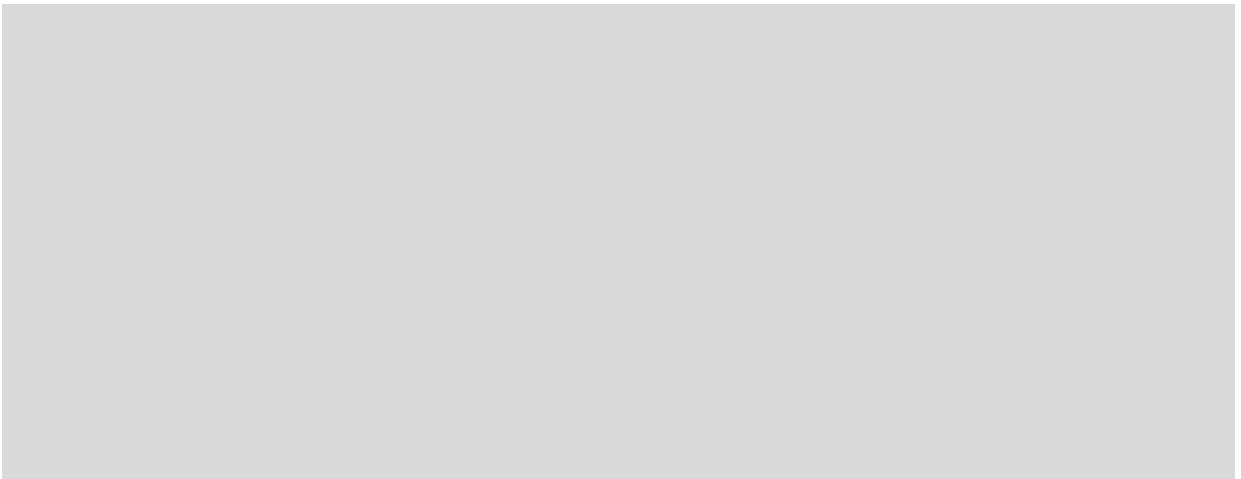
Was versteht man unter einem *H-State* und einem *G-State*? Beschreiben Sie die Bedeutung dieser beiden Konzepte. (4)



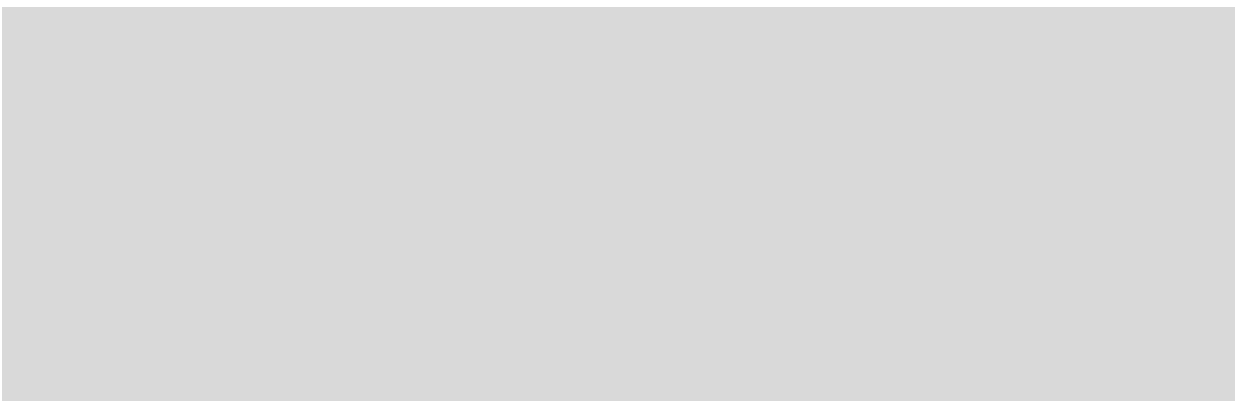
Charakterisieren Sie die folgenden Interface-Typen: *Elementary Interface*, *Composite Interface* und *Temporal Firewall Interface*. (6)



Was versteht man unter dem *Action Delay*? Wie groß ist das Action Delay in einem verteilten Echtzeitsystem (a) ohne globale Zeitbasis bzw. (b) mit globaler Zeitbasis? (5)



Charakterisieren Sie das Ressourcenmanagement in einem zeitgesteuerten Echtzeitbetriebssystem. (4)



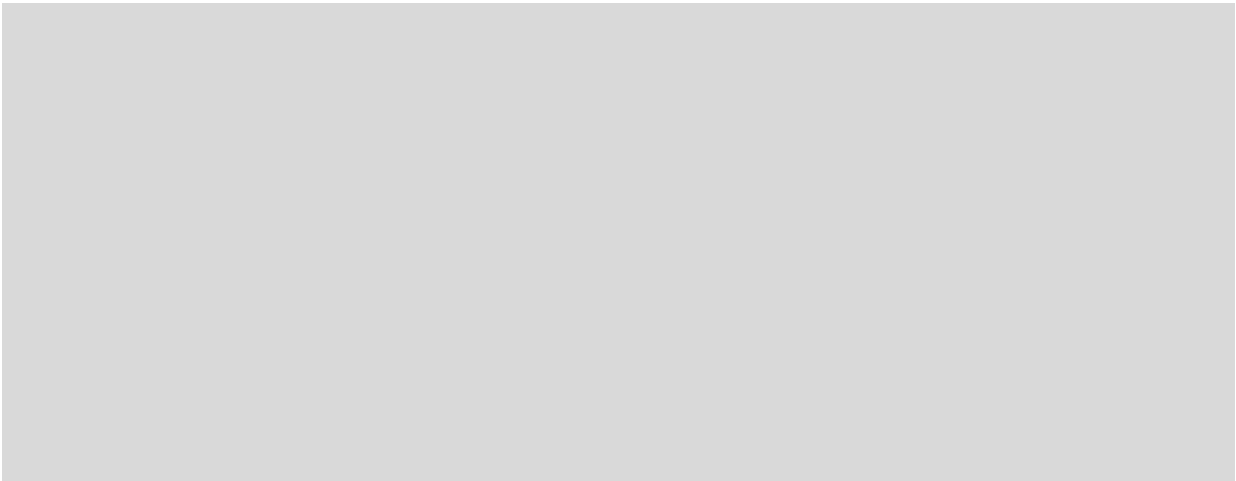
## 2 Scheduling (30)

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.

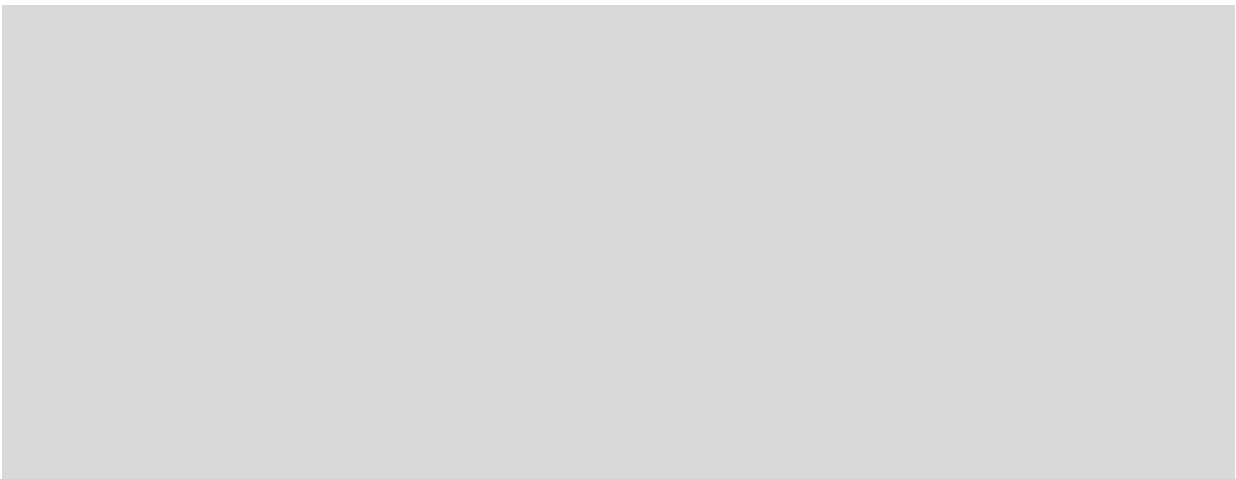
	<u>Periode <math>T_i</math></u>	<u>Ausführungszeit <math>C_i</math></u>
Task 1:	4	1
Task 2:	16	2
Task 3:	8	5

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:

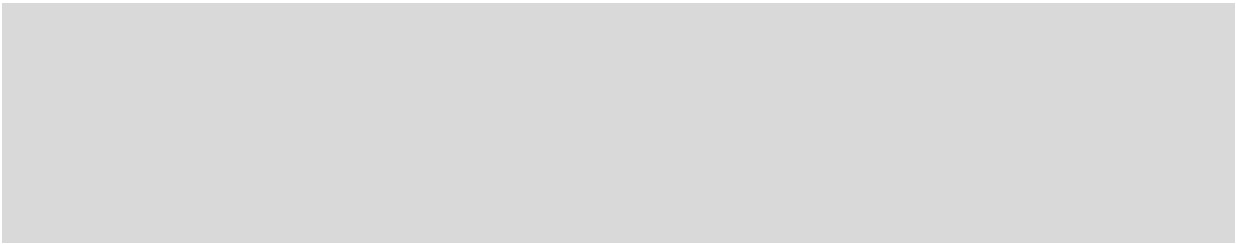


Earliest Deadline First Scheduling:

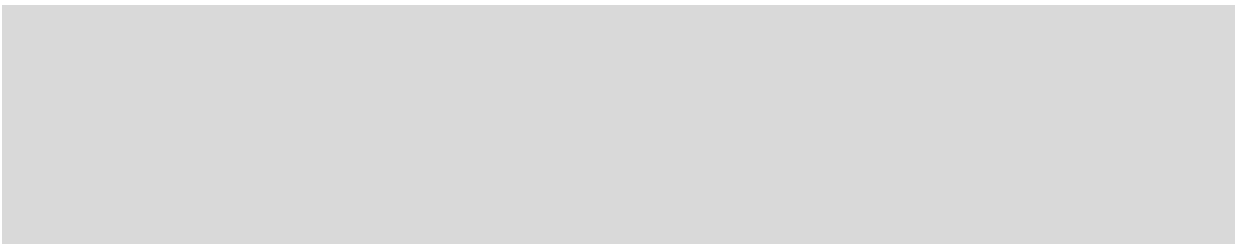


Welche Aussagen können Sie aufgrund der Ergebnisse der Schedulability Tests über das gegebene Task Set machen?

Rate Monotonic Scheduling:

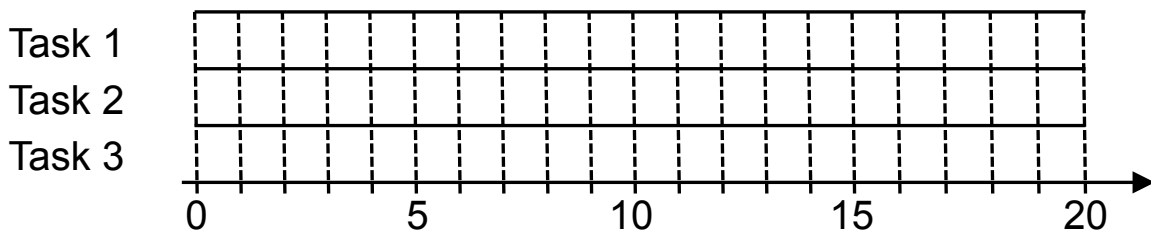


Earliest Deadline First Scheduling:

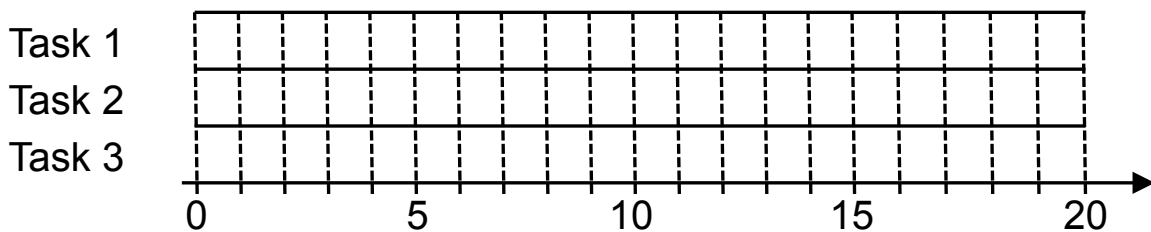


Tragen Sie in den folgenden Diagrammen ein, wie das gegebene Task Set nach den beiden Schedulingstrategien abgearbeitet wird. Nehmen Sie dabei an, dass die erste Aktivierung aller drei Tasks zum Zeitpunkt Null erfolgt. Tragen Sie in jedem Diagramm für jede Zeiteinheit den jeweils laufenden Task ein. 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:

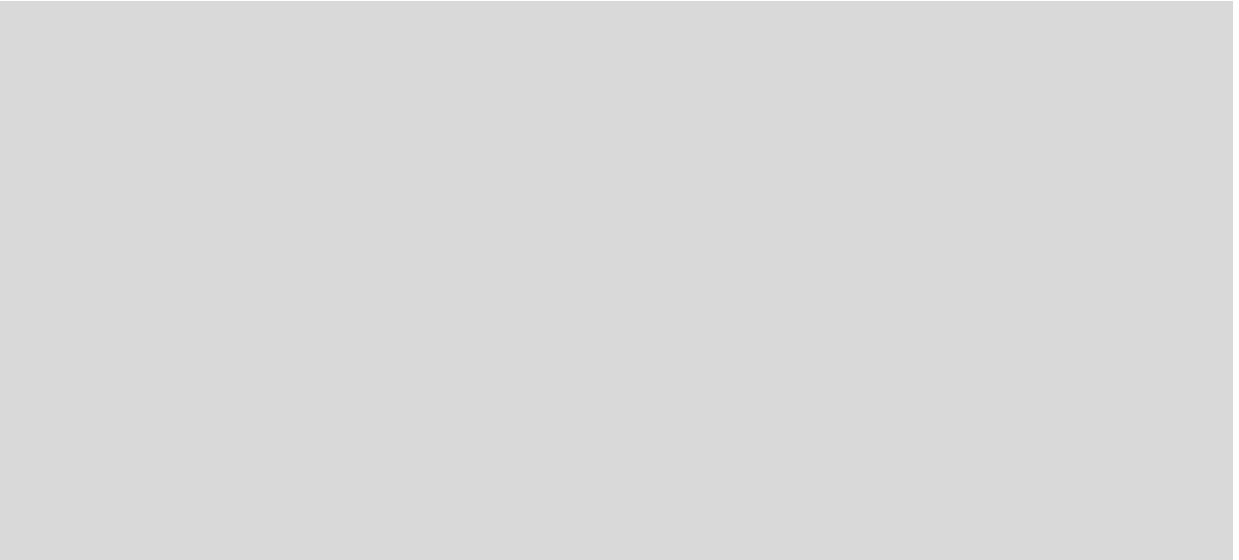


Earliest Deadline First Scheduling:



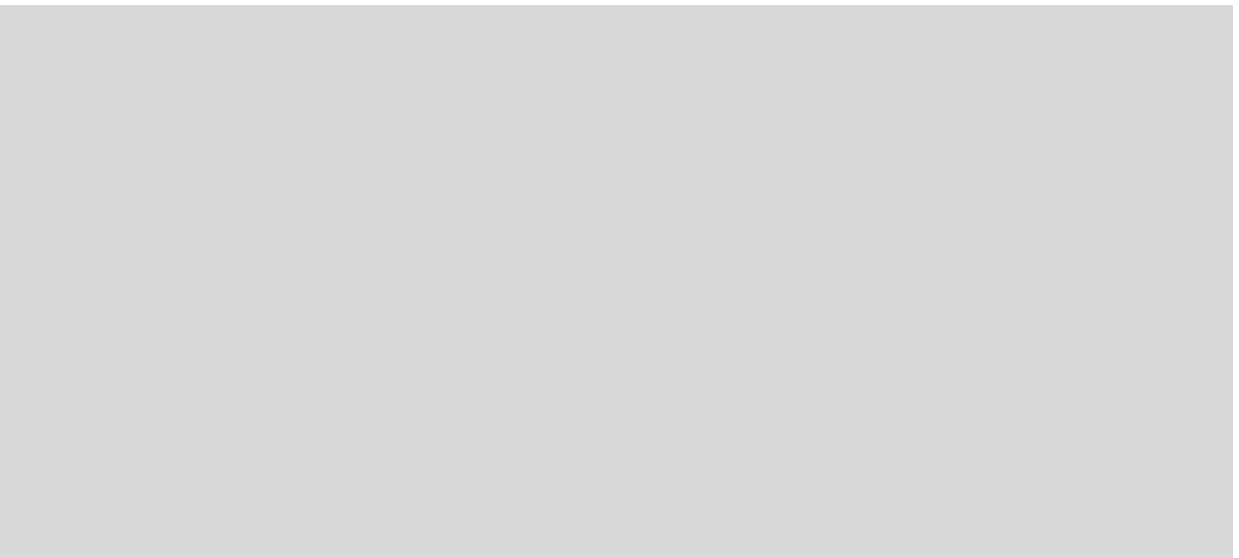
### 3 Rechenaufgaben (15)

Ein Sensor liest ein Signal, dessen Verlauf durch eine Dreiecksschwingung mit Frequenz  $f$  und Amplitude  $A$  beschrieben wird. Schätzen Sie den maximalen Fehler des Real-Time Images unter der Annahme ab, dass  $d_{acc}$  gleich 2% der Periode des Eingangssignals ist. (7)



Gegeben sind 6 Knoten mit perfekt synchronisierten Uhren, die über ein Netzwerk von  $100m$  Länge verbunden sind. Die Knoten kommunizieren mittels TDMA Protokoll, wobei jeweils Nachrichten von 1000 Bit mit einer Übertragungsrate von  $10 \text{ Mbit/s}$  übertragen werden. Nachrichten werden mit einem Abstand (Interframe Gap) von  $1\mu s$  übertragen.

Mit welcher Frequenz (= Nachrichten pro Sekunde) kann jeder der Knoten Nachrichten im Netzwerk senden? (8)



## 4 WCET Analyse (25)

Für das folgende Programmstück, das *bubble sort* implementiert, soll eine baumbasierte WCET-Analyse (tree based WCET analysis) durchgeführt werden. Für jedes Statement bzw. für jede Bedingung ist neben dem Codestück die Ausführungszeit angegeben. Die maximale Iterationsanzahl der Schleife leiten Sie aus dem Code ab.

```
i = SIZE-1;           tsimple = 12
while (i > 0) {       tcond = 16
  j = 1;              tsimple = 10
  while (j <= i) {    tcond = 18
    if (a[j-1] > a[j]) tcond = 26
    {
      swap(a, i, j);  tsimple = 64
    }
    j++;              tsimple = 10
  }
  i--;                tsimple = 10
}
```

Geben Sie eine Formel an, die zeigt, wie die baumbasierte Analyse eine WCET-Schranke für das angegebene Codestück ermittelt. Berechnen Sie die WCET-Schranke.

