


<p style="text-align: center;">Ostfalia Hochschule für angewandte Wissenschaften</p>  <p>Fakultät Fahrzeugtechnik Prof. Dr.-Ing. V. von Holt Institut für Fahrzeugsystem- und Servicetechnologien</p>	<p>Modulprüfung Embedded Systems BPO 2011/BPO 2008</p> <p style="text-align: center;">WS 2015/2016 06.01.2016</p>	<p>Name:.....</p> <p>Vorname.....</p> <p>Matr.Nr.:.....</p> <p>Unterschrift.....</p>
---	---	--

Zugelassene Hilfsmittel: **Einfacher Taschenrechner**
Zeit: 60 Minuten

1 (16)	2 (32)	3 (12)	Summe (60)	Note

Aufgabe 1 (16 Punkte) – Kurzfragen

- a) (2 P) Welche 2 wesentlichen Aufgaben erfüllt ein (allgemeines) „**Betriebssystem**“?
- b) (2 P) Was unterscheidet Systeme mit **preemptiven** bzw. **nicht-preemptivem** Multitasking voneinander?
- c) (2 P) Wann bezeichnet man ein **Schedulingverfahren** als „**optimal**“?
- d) (2 P) Was versteht man unter einer „**Task**“ und was unter dem „**Taskkontext**“?

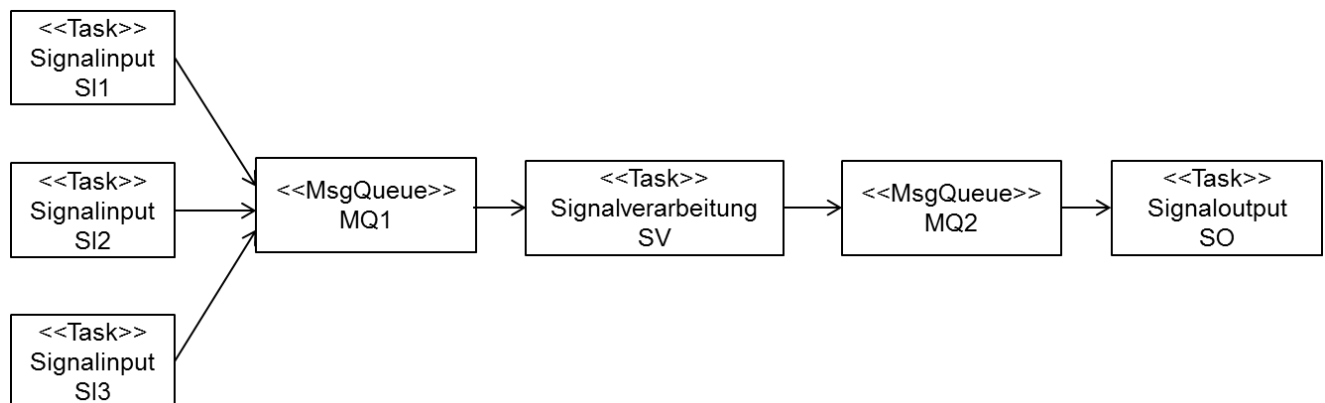
- e) (8 P) Erläutern Sie anhand eines Aktivitätsdiagramms das Prinzip der Reihenfolgesynchronisation mit Semaphoren beim Zugriff auf geteilte Ressourcen!

Aufgabe 2 (32 Punkte) – Scheduling

Ein System zur Signalverarbeitung empfängt von **3 Quellen** unterschiedliche Eingangssignale. Diese werden von **3 Eingangs-Tasks SI1, SI2 und SI3** empfangen. Die Eingangssignale treffen mit **unterschiedlichen** Perioden, die jeweils **gleichverteilt innerhalb eines Intervalls variieren** als **ereignisgesteuerte** Signale ein. Die Periodenzeiten sowie die jeweiligen Bearbeitungszeiten der 3 Eingangs-Tasks sind in der u.a. Tabelle aufgeführt.

Die Eingangs-Tasks leiten die Signale in eine **Message Queue MQ1** weiter. Aus dieser werden sie durch die **Signalverarbeitungs-Task SV** in der **Reihenfolge ihres Eintreffens** entnommen und verarbeitet. Je nach **Signalquelle unterscheiden** sich die **Bearbeitungszeiten** der Eingangssignale durch die SV-Task (s. Tabelle).

Nach der Verarbeitung werden die Signale in der **Message Queue MQ2** abgelegt, aus der sie durch die **Ausgabe-Task SO** entnommen und an weiterverarbeitende Prozessoren über eine Schnittstelle ausgegeben werden.



Die folgende Tabelle enthält die Zykluszeiten sowie die Laufzeiten der einzelnen Tasks.

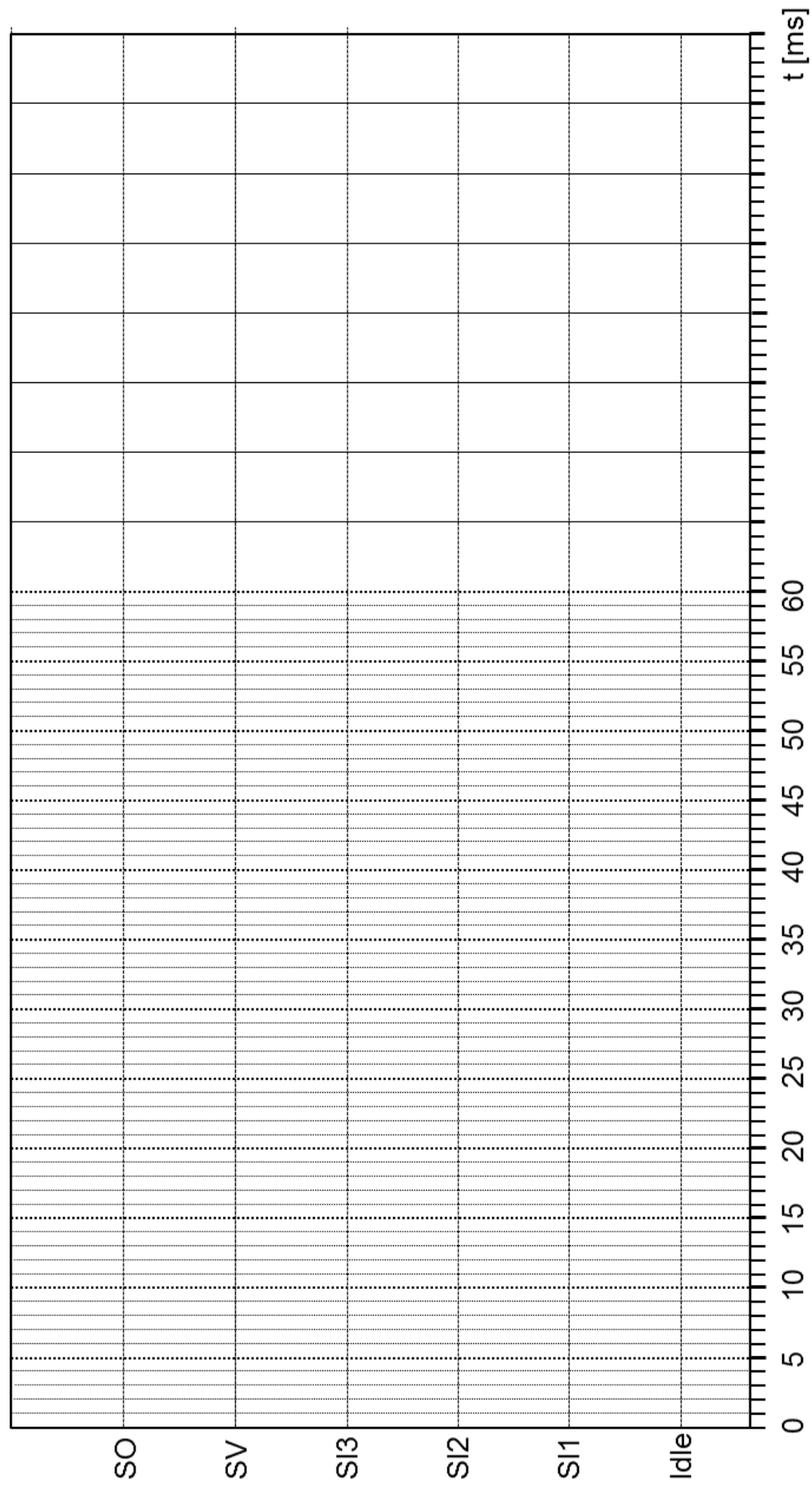
Tasks	Zykluszeit [ms]	Laufzeit[ms]
SI1	10..30	1
SI2	20..40	1
SI3	40..60	2
SV	?	SI1: 2 SI2: 3 SI3: 7
SO	?	1

- a) (3 P) Welche Annahmen nach o.a. Tabelle charakterisieren den **Worst-Case** bzw. den **Best-Case**?
- b) (4 P) Bestimmen Sie die **mittlere Zykluszeit** der SV-Task sowie der SO-Task für den Worst-Case und den Best-Case!

- c) (6 P) Berechnen Sie die minimale, maximale sowie die mittlere **Prozessorlast**, die durch die einzelnen Tasks sowie durch das gesamte **Taskset** verursacht wird!

Tasks	Minimal [%]	Maximal [%]	Gemittelt [%]
SI1			
SI2			
SI3			
SV			
SO			
Gesamt			

- d) (1 P) Ist das gegebene Taskset nach den Berechnungen unter c) prinzipiell realisierbar?
- e) (6 P) Das Taskset soll durch ein **Rate-Monotonic-Scheduling** realisiert werden soll. Welche **Prioritäten** müssen den **Tasks** jeweils zugewiesen werden? (**Höchste Priorität : 0**)
Nach welcher **Regel** werden die **Prioritäten vergeben**?
Ist das **Taskset** in jedem Fall mit RMS-Scheduling **umsetzbar** (Begründung)?
- f) (12 P) Weisen Sie anhand des u.a. Schedulediagramms für den **Worst-Case** nach, ob das Taskset mittels RMS realisierbar ist!
(Die Message Queues sind zum Startzeitpunkt als leer anzunehmen!)



Aufgabe 3 (12 Punkte) – Synchronisation/Kommunikation

Ein Nachteil der in Aufgabe 2 verwendeten Taskanordnung besteht darin, dass unter ungünstigen Umständen eine der Message Queues überlaufen könnte, d.h. es könnten Informationen verloren gehen. Um dem vorzubeugen soll die Anordnung um eine sogenannte Flusskontrolle erweitert werden, bei der die in die Message Queues schreibenden Tasks nur dann eine Information weitergeben, wenn in den Message Queues noch Platz vorhanden ist.

Ergänzen Sie den u.a. Ausschnitt aus der Taskanordnung von Aufgabe 2 um eine solche Flusskontrolle und erläutern Sie deren Funktionsweise anhand von Pseudocode oder mittels eines Aktivitätsdiagramms!