Prof. Dr. Christian Steger
Prof. Dr. Reinhard Wilhelm

Sebastian Altmeyer, M.Sc.
Michael Gerke, B.Sc.
Dipl.-Inf. Hans-Jörg Peter

Embedded Systems 2010/2011 – Assignment Sheet 5

Due: Tuesday, 30th November 2010, *before* the lecture (i.e., 10:10)
Please indicate your **name**, **matr. number**, **email address**, and which **tutorial** you are planning to attend on your submission. We encourage you to collaborate in **groups** of up to **three** students. Only one submission per group is necessary. However, in the tutorials every group member must be capable to present each solution.

Exercise 1: SyncCharts with Scade (35 pts.)

Use Scade to create a SyncChart model of a lift controller having the following properties:

- The lift moves between two levels.
- There are two buttons within the lift (one for each level) and one button at each level.
- If the lift is not in use, the door is closed.
- The door closes automatically after five ticks, unless somebody enters/leaves the lift or the button for the current level is pressed.
- The door opens, if the lift enters another level or if the button for the current level is pressed (only in case the lift is not moving).
- The lift can be in-between the two levels; the door must be closed in this case.

Input signals: *LiftButton1*, *LiftButton2* (buttons within the lift), *LevelButton1*, *LevelButton2* (buttons at the two levels) and *LightSignal* (indicates if someone enters/leaves the lift).

Create a new ScadeSuite workspace "EmbSys" and within this workspace a new project "Lift". Add a file "group.txt" to this directory containing the name and matr. number of each group member. Compress the whole workspace directory (to .rar or .zip) and send it to:

`altmeyer@cs.uni-saarland.de`

In addition, provide a print-out of the graphical representation of the model as well as a short explanation. Only submissions available on paper **and** via mail will be graded.

Please turn the page over...

Exercise 2: Aperiodic Uniprocessor Scheduling (35 pts.)

Prove that the problem of scheduling a set of aperiodic, asynchronous tasks on a non-preemptive, uniprocessor architecture is NP-complete by solving the following two subtasks:

(a) Consider the problem *Partition* which is known to be NP-hard:

> For a given finite set of positive integers S, where the sum of its elements is even, find a subset S' of S such that
>
> $$\sum_{i \in S'} i = \frac{1}{2} \sum_{j \in S} j.$$

Prove that the scheduling problem stated above is NP-hard by showing that there is a polynomial reduction from Partition. That is, formally describe how an arbitrary instance of Partition can be transformed into an instance of the scheduling problem such that there is only a polynomial blow-up in the size of the transformed instance. Furthermore, you have to describe how to interpret the scheduling result: What means (un)schedulability for the original Partition instance? (25 pts.)

(b) Prove that the scheduling problem stated above is in NP by providing a polynomial-time algorithm that checks if a given schedule is feasible. (10 pts.)

Exercise 3: EDD Scheduling (30 pts.)

Consider the EDD (Earliest-Due-Date) algorithm for scheduling n synchronous, aperiodic, and independent tasks on a uniprocessor architecture. For each of the following statements, decide whether EDD minimizes it. Justify your answer either by giving a formal proof or a counter-example.

(a) The average response time (10 pts.)

$$\overline{R} = \frac{1}{n} \sum_{i=1}^{n} f_i.$$

(b) The total completion time (10 pts.)

$$T_c = \max_{1 \leq i \leq n} (f_i).$$

(c) The number of late tasks (10 pts.)

$$N_{late} = \left| \{1 \leq i \leq n : d_i < f_i\} \right|.$$

Here, for each $1 \leq i \leq n$, d_i is the deadline and f_i is the response (or finishing) time of task i, respectively.