Prof. Bernd Finkbeiner
Dr. Calogero Zarba
Moritz Hahn

Problem Set 8
Due: July 3, 2007, 9:14 am

Embedded Systems

Problem 1 (EDD-scheduling) [10 points]

Consider the following set of aperiodic tasks

	J_1	J_2	J_3	J_4
a_i	0	0	0	0
d_i	9	16	5	10
C_i	4	5	2	3

where task J_i has arrival time a_i, absolute deadline d_i, and computation time C_i.

Compute a scheduling for the given set of tasks using Jackson's EDD scheduling algorithm. What is the maximum latency $L_{\max}$ of the given set of tasks? Is the given set of tasks schedulable with respect to the criteria $1 \mid \mathit{sync} \mid L_{\max}$?

Problem 2 (Scheduling) [50 points]

Consider the following scheduling problem $1 \mid \mathit{sync} \mid \overline{R}$:

> Using a uniprocessor machine, find a schedule for n aperiodic tasks $J_1, \ldots, J_n$ with computation times $C_1, \ldots, C_n$ that minimizes the average response time
>
> $$\overline{R} = \frac{1}{n} \sum_{i=1}^{n} (f_i - a_i)\,,$$
>
> where f_i is the time at which task i finishes its execution. Assume that all tasks arrive at time 0, that is, $a_i = 0$, for all $i = 1, \ldots, n$.

(a) Devise a scheduling algorithm that is optimal for the scheduling problem $1 \mid \mathit{sync} \mid \overline{R}$, and whose complexity is no worse than $\mathcal{O}(n \log n)$. **[20 points]**

(b) Formally prove that your algorithm is optimal. *Hint:* If your algorithm is deterministic, then you can start by proving the following lemma: Let S be a task-set, let $\sigma(S)$ be *the* schedule obtained by your algorithm, and let τ be any schedule. Then there exists a sequence of schedules $\tau_0, \ldots, \tau_k$ such that (i) $\tau_0 = \tau$, (ii) $\tau_k = \sigma(S)$, and (iii) the average response time of τ_{i+1} is not greater than the average response time of τ_i, for $i = 0, \ldots, k-1$. **[30 points]**

Problem 3 (WCET/VHDL) [10 points]

Consider the 4-bits multiplier whose schematic can be found in Solution Set 5. What is its WCET?

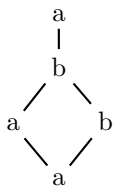
Assume that:

- All gates are implemented using only the NAND-gate.
- The NAND-gate has execution time 10ns.

- If two VHDL entities e_1 and e_2 are composed sequentially, obtaining a new entity e, then the WCET of e is the sum of the WCET of e_1 and the WCET of e_2.
- If two VHDL entities e_1 and e_2 are composed in parallel, obtaining a new entity e, then the WCET of e is the maximum of the WCET of e_1 and the WCET of e_2.

Problem 4 (WCET/Cache analysis) [30 points]

FIFO is an alternative cache-replacement policy. In case of a cache-miss, the oldest cache entry is removed. Consider the following sequence of accesses:



(a) Assume an LRU-cache with 2 entries. Apply a must-cache and a may-cache analysis of the program. Can you predict the cache state at the last access to a? **[10 points]**

(b) Assume a FIF0-cache with 2 entries and an empty cache at the beginning. Can you predict a cache-hit or a cache-miss at the last access to a? **[10 points]**

(c) Repeat (b) with an unknown cache-state at the beginning of the program. What can you say about the predictability of an LRU-cache compared to a FIFO-cache? **[10 points]**