Name:
Neptun-code:

Embedded Systems test paper #1

December 19, 2006, 14:15–15:45
Room I.E.225

The exam is closed book. Communication is strictly prohibited. You have 90 minutes to work on the questions. You can solve the problems in any order. Please, do not forget to put your name and neptun-code on each paper you consider valuable. Grading:

~~40 ≥ x > 34p: **5**~~ 36 ≥ x > 30p: **5**
~~34 ≥ x > 28p: **4**~~ 30 ≥ x > 24p: **4**
~~28 ≥ x > 22p: **3**~~ 24 ≥ x > 18p: **3**
~~22 ≥ x > 16p: **2**~~ 18 ≥ x > 12p: **2**
~~16 ≥ x ≥ 0p: **1**~~ 12 ≥ x ≥ 0p: **1**

1. Define failure rate, MTTF, MTTR, MTBF and availability *(2p)*. How can we define hard real-time systems (i.e., safety critical systems) using reliability measures *(2p)*?

2. Define UTC and TAI, and describe the difference between them *(2p)*. Why UTC might be dangerous for HRT systems *(1p)*?

3. Given a clock synchronization system with a precision of 900 microsec. What is a reasonable granularity for the global time *(1p)*? What are the limits for the observed values for a time interval of 11 msec *(2p)*?

4. Describe the Fault-Tolerant Averaging (FTA) clock-synchronization algorithm *(3p)*. What kind of faults can it tolerate and how many clocks might have this type of errors *(2p)*?

5. ~~Consider a break pedal in a car. Its temporal accuracy is 4 msec. Transaction from the sensor node to the processing node takes 2 msec. Define the concept of temporal accuracy *(1p)*. Define parametric/phase-insensitive RT image (draw a figure) *(1p)*. How frequently shall we measure the pedal position to make its image parametric/phase-insensitive *(2p)*?~~

6. Define simple and complex tasks *(1p)*. Draw a finite state machine of the possible states of a complex task *(2p)*. What is the difference between aperiodic and sporadic tasks *(1p)*?

7. Given 3 tasks with the following properties (times are in ms):

Task ID	C_i, comp. time (WCET)	P_i, period	D_i, deadline	S_1 sem. locking time	S_2 sem. locking time
T_1	5	250	10	2	-
T_2	2	20	15	-	1
T_3	25	330	50	4	5

- Ignore semaphore usage, and calculate the necessity test for independent, periodic tasks scheduled on one processor. Is it schedulable? *(3p)*
- Calculate the interference and the response time of T_3 (i.e., I_3 and R_3) using the iterative solution of deadline monotonic analysis. *(3p)*
- Calculate the blocking time B_1 and the response time R_1 for T_1, if priority ceiling protocol is used. *(2p)*

Ignore operating system overhead in any of the calculations.

8. Describe how stack watermarking can be carried out (draw figures) *(2p)*. How can we use it for real-time monitoring of the stack *(2p)*?

9. Describe CAN bus operation, arbitration mechanism and the bit-fields *(3p)*. How long can the network be if the bus bandwidth is 100 kbit/sec and propagation speed is $2x10^{-8}$ m/s? Why? (Hint: calculate the bit cell of communication) *(2p)*