UNIVERSITÄT DES SAARLANDES
FR 6.2 – Informatik
Prof. Bernd Finkbeiner
Dr. Calogero Zarba
Moritz Hahn

Final Exam
Embedded Systems
SS 2007

Name: .....

Matr.Nr.: .....

- The exam consists of **13 problems** and is worth **180 points**. You have **180 minutes** to complete it.
- Print your full name on every sheet of paper you use.
- In the table below, mark each problem for which you turn in a solution.
- Attach all pages of your solution to this problem set.

Sign here:

.....

Problem	1	2	3	4	5	6	7	8	9	10	11	12	13	Σ
Submitted														
Score														
Max	25	5	15	5	20	5	10	10	15	10	10	20	30	180

1 Problem (Statecharts) [25 points]

Draw a statechart modeling a heart rate monitor. Your model should handle the following events and variables:

- an internal event tick, periodically generated every 0.1 seconds with the help of a timer;
- an input event on-off, generated when the user presses the on/off button;
- an input event start-stop, generated when the user presses the start/stop button;
- an input integer variable age;
- an input boolean variable signal, externally set to *true* whenever a heart beat is detected;
- an output floating-point variable rate;
- an output boolean variable blink, internally set to *true* when the display is blinking.

When the heart monitor is turned on then:

1. If the start/stop button has never been pressed then the value of the variable rate should be 0. The display should not blink.
2. If the start/stop button has been pressed an odd number of times then the value of the variable rate should be 0 for the first minute. Afterwards, it should be updated every full minute to the number of beats detected in the last minute. The display should blink if and only if the button start/stop was pressed more than a minute ago and the value of the variable rate falls outside the range

$$[\ 0.6 \times (220 - \mathsf{age}) \quad \ldots \quad 0.8 \times (220 - \mathsf{age})\ ]\ .$$

3. If the start/stop button has been pressed a positive even number of times then let T be the time, expressed in minutes, between the second-last time and last time the start/stop button was pressed, and let B be the number of beats detected during that time interval. If $T < 1\text{min}$ then the value of the variable rate should be 0; otherwise it should be equal to B divided by T. The display should not blink.

2 Problem (Petri nets) [5 points]

(a) Define place/transition Petri nets.

(b) Draw a net that is not simple.

(c) Draw a net that is not pure.

3 Problem (Petri nets) [15 points]

Draw a colored Petri net implementing the function

$$midpoint(X)\,,$$

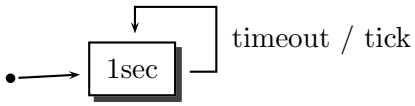
where

- $X = \langle x_1, \ldots, x_n \rangle$ is a nonempty multiset of integers;
- $midpoint(X) = (\max(X) + \min(X))/2$.

Your net should contain an "input" place A, initially containing n tokens labeled by the elements of X. All other places should initially contain no tokens. After a finite number of transitions, your net should enter a deadlock, and a given "output" place R should contain one token labeled by $midpoint(X)$. If necessary, you can assign priorities to the transitions of your net. When more than one transition is activated, assume that only a transition with the highest priority can fire.

4 Problem (SDL) [5 points]

Construct a graphical SDL representation equivalent to the following statechart timer.



5 Problem (VHDL) [20 points]

Consider the following VHDL declaration of a shifter.

```
entity shifter is
   port (left    : in Bit;  -- '1' for shifting one bit to the left, '0' for right
         logical : in Bit;  -- '1' for logical, '0' for arithmetic
         in0     : in Bit;
         in1     : in Bit;
         in2     : in Bit;
         in3     : in Bit;
         out0    : out Bit;
         out1    : out Bit;
         out2    : out Bit;
         out3    : out Bit);
end entity shifter;
```

Assume that the most significant bit is the leftmost bit, that is, the input and output vectors look like this:

in3	in2	in1	in0
out3	out2	out1	out0

Remember that the left arithmetic shift and the logical shifts fill the vacant bit with 0, whereas the right arithmetic shift fills the vacant bit with the value of `in3`.

(a) Write a *behavioral* VHDL architecture for the shifter.

(b) Draw a hardware schematic, *which uses only multiplexers as basic components.*

(c) Write a *structural* VHDL architecture based on your schematic.

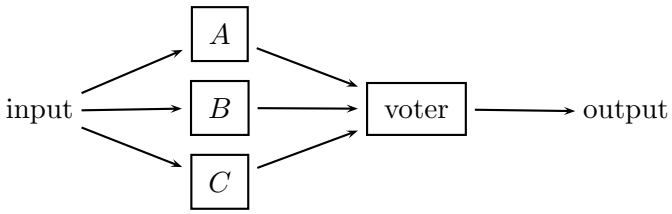
(d) Assuming that each multiplexer has a WCET of 10ns, what is the WCET of your shifter's structural architecture?

6 Problem (Reliability) [5 points]

1. Define RELIABILITY $R(t)$.
2. Define MAINTAINABILITY $M(t)$.
3. Define AVAILABILITY $A(t)$.
4. Define FAILURE RATE $r(t)$.

7 Problem (TMR-arrangement) [10 points]

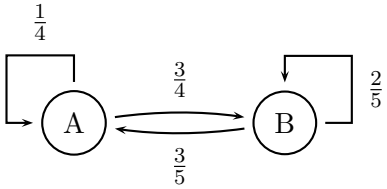
Consider the following triple-module-redundancy (TMR) arrangement.



Assume that all components deliver a bit (0 or 1) as their output. The correct output of the voter is 1 if and only if at least two of the voter's inputs are equal to 1; it is 0 otherwise. Denote with R_X the probability that component X outputs the correct bit, and let $R_A = R_B = R_C = \frac{3}{4}$ and $R_{\text{voter}} = \frac{4}{5}$. What is the probability R_{TMR} that the TMR-arrangement outputs the correct bit?

8 Problem (Markov processes) [10 points]

Given the following Markov process, compute $s_{\lim}(A)$ and $s_{\lim}(B)$.



9 Problem (A/D conversion) [15 points]

Consider the problem of converting an analog value ranging between $U_{\min} = 2V$ and $U_{\max} = 9.5V$ into a corresponding 4-bit digital value ranging between 0000_2 and 1111_2.

(a) Using a flash A/D converter, how many comparators are necessary? What should be the reference voltage values of the comparators?

(b) Using the successive approximation method, carry out the conversion of the input voltages $U_{\text{in}} = 5.2V$ and $9.4V$ into the corresponding binary values. In each case, and for each step of the conversion, show the arranged comparison voltage U_{ref} as well as the binary value after each comparison.

(c) Discuss the advantages and disadvantages of the flash and successive approximation methods.

10 Problem (Scheduling) [10 points]

Let $\mathcal{J} = \{J_1, \ldots, J_n\}$ be a set of tasks, and let $\mathcal{P} = \{P_1, \ldots, P_m\}$ be a set of processors. In class we formally defined what is a SCHEDULE of $\mathcal{J}$ with respect to $\mathcal{P}$ in the special case of uniprocessor machines, i.e., when $m = 1$.

(a) Assume $m = 1$. Formally define what is a schedule of $\mathcal{J}$ with respect to $\mathcal{P}$.

(b) For arbitrary $m \geq 1$, formally define what is a schedule for $\mathcal{J}$ with respect to $\mathcal{P}$. Your definition should prevent a task from being executed at the same time in two distinct processors.

(c) For arbitrary $m \geq 1$, formally define what is a *nonpreemptive* schedule of $\mathcal{J}$ with respect to $\mathcal{P}$.

11 Problem (Periodic scheduling) [10 points]

Let $\Gamma = \{\tau_1, \tau_2\}$ be a task set with

$$\begin{array}{ll} T_1 = D_1 = 5 & C_1 = 2 \\ T_2 = D_2 = 7 & C_2 = 4 \end{array}$$

Exhibit for Γ:

(a) an EDF-schedule;

(b) an RM-schedule.

12 Problem (Aperiodic scheduling) [20 points]

Consider the problem of scheduling a set of synchronous tasks on a uniprocessor machine. It was shown in class that Jackson's EDD algorithm minimizes the maximum lateness

$$L_{max} = \max_i (f_i - d_i) \, .$$

For each of the following criteria, determine whether the EDD algorithm minimizes it:

(a) average response time $\overline{R} = \frac{1}{n} \sum_{i=1}^{n} f_i$;

(b) total completion time $T_c = \max_i(f_i)$;

(c) weighted sum of completion times $T_w = \sum_{i=1}^{n} w_i f_i$;

(d) number of late tasks $N_{late} = \sum_{i=1}^{n} (\mathit{if}\ d_i > f_i\ \mathit{then}\ 1\ \mathit{else}\ 0)$.

For each criterion, if EDD minimizes it, give a formal proof; otherwise give a counterexample.

13 Problem (Aperiodic scheduling) [30 points]

Consider the following scheduling problem $1 \mid sync \mid T_w$:

> Using a uniprocessor machine, find a schedule for a set $\mathcal{J} = \{J_1, \ldots, J_n\}$ of n synchronous tasks with computation times $C_1, \ldots, C_n$ that minimizes the weighted sum of the completion times
>
> $$T_w = \sum_{i=1}^{n} (w_i f_i) \,,$$
>
> where $w_i > 0$ is a weight, and f_i is the time at which task i finishes its execution. (*Note:* The schedule is not required to respect the deadlines. We are only interested in minimizing T_w.)

(a) Let $\mathcal{J}$ be a task set, and let σ be a schedule for $\mathcal{J}$ that is optimal with respect to the problem $1 \mid sync \mid T_w$. Formally prove that there exists a nonpreemptive schedule σ^* for $\mathcal{J}$ with the same T_w of σ.

(b) Devise a polynomial-time algorithm that, given a task set $\mathcal{J} = \{J_1, \ldots, J_n\}$, computes a schedule σ for $\mathcal{J}$ that is optimal with respect to the scheduling problem $1 \mid sync \mid T_w$.

(c) Formally prove that your algorithm computes an optimal schedule.