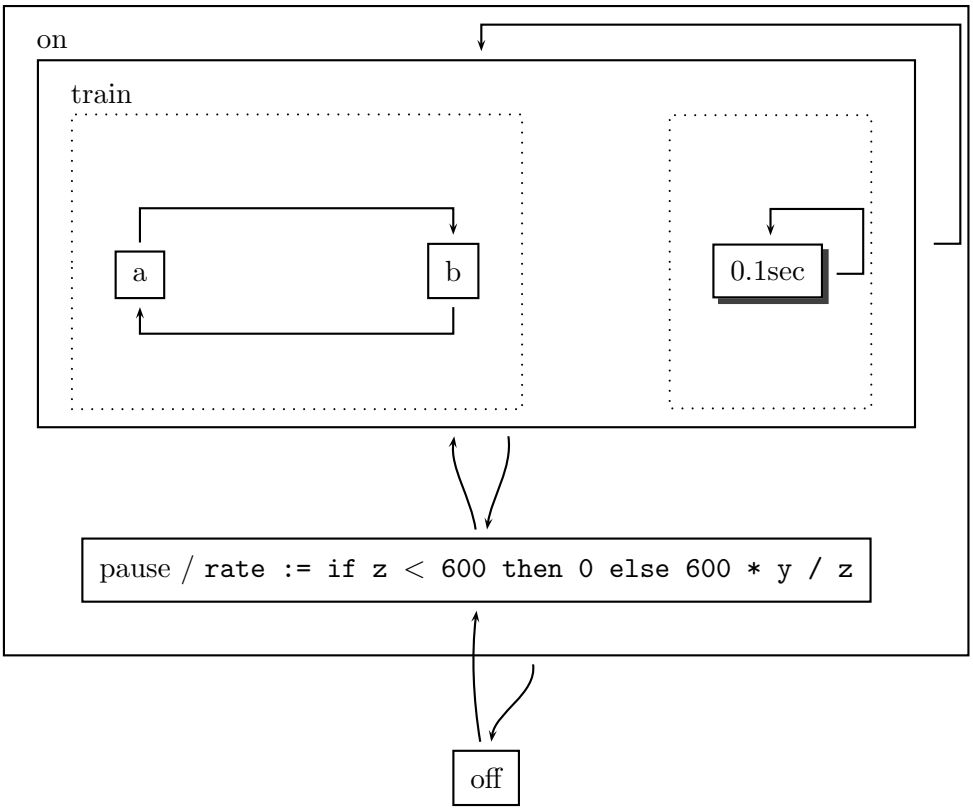
Prof. Bernd Finkbeiner
Dr. Calogero Zarba
Moritz Hahn

Final Solutions

Embedded Systems

---

1 Problem (Statecharts)



Transitions

- train → train: `after(600, tick) / rate := x; blink := x < 0.6 * (220 - age) or x > 0.8 * 220 - age; x := 0;`
- a → b: `[signal] / x := x + 1; y := y + 1`
- b → a: `[!signal]`

- 0.1sec → 0.1sec: `timeout/tick, z := z + 1`
- pause → train: `start-stop / x:= 0; y := 0; z := 0; rate := 0; blink := 0`
- train → pause: `start-stop`
- on → off: `on-off / x := 0; y := 0; z := 0; rate := 0; blink := 0`
- off → pause: `on-off / x := 0; y := 0; z := 0; rate := 0; blink := 0`

2 Problem (Petri nets)

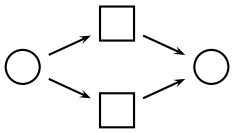
(a) A PLACE/TRANSITION PETRI NET is a tuple

$$(P, T, F, K, W, M)$$

where

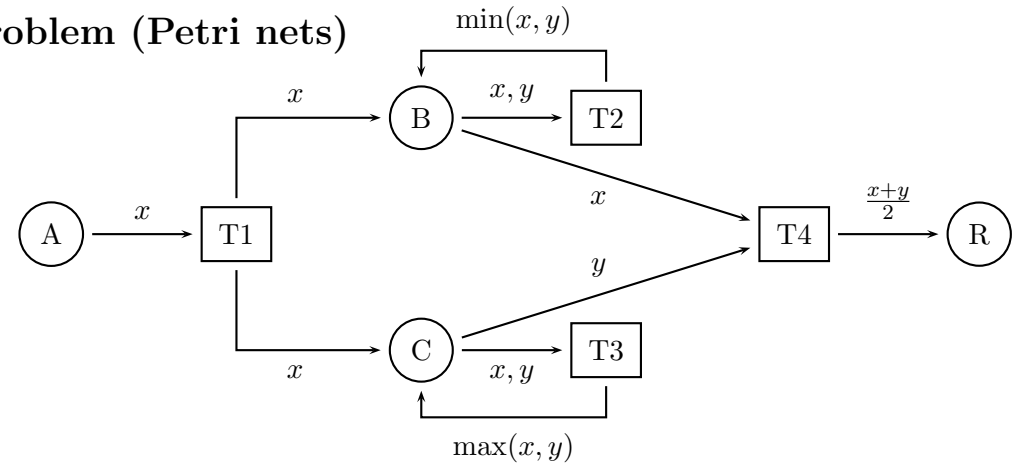
- P is a nonempty set of *places*;
- T is a nonempty set of *transitions*;
- $P \cap T = \emptyset$;
- $F \subseteq (P \times T) \cup (T \times P)$ is a *flow relation*;
- $K : P \to \mathbb{N}^+ \cup \{\omega\}$ is a *capacity function*
- $W : F \to \mathbb{N}^+$ is a *weight function*;
- $M : P \to \mathbb{N}$ is the *initial marking*;
- $M(p) \leq K(p)$, for all places $p \in P$.

(b)



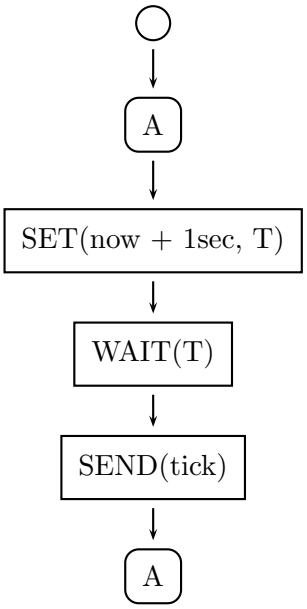
(c)

3 Problem (Petri nets)



Priorities: T1 > T2, T3 > T4

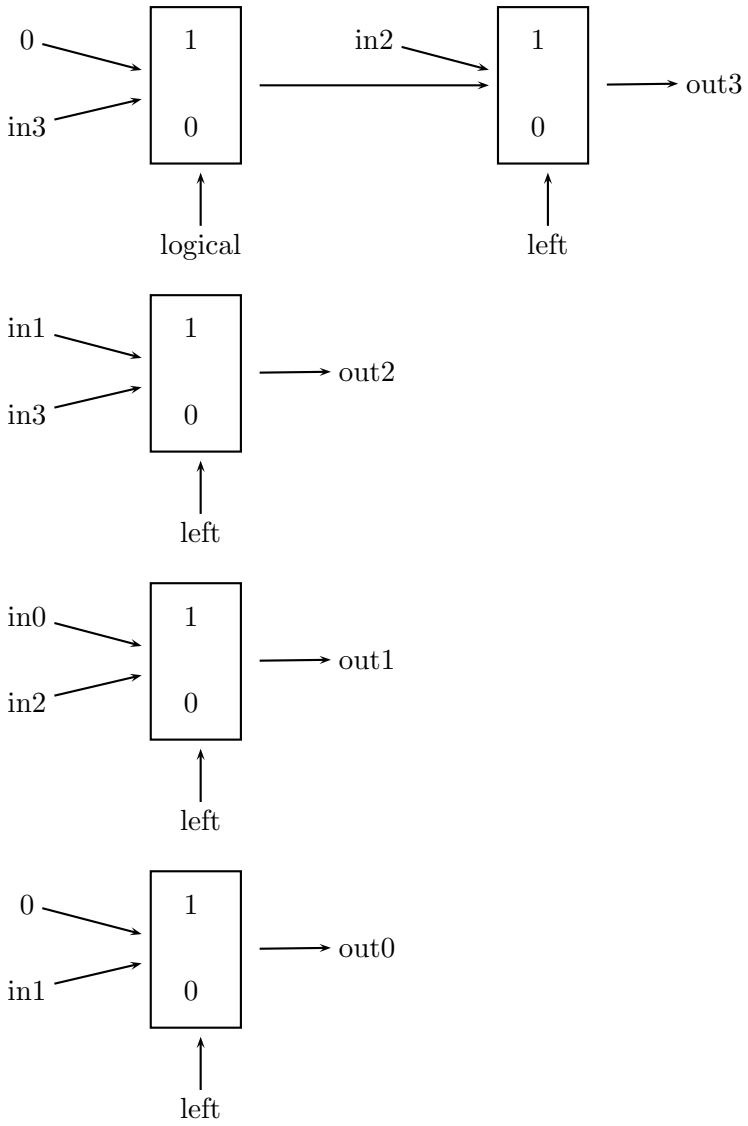
4 Problem (SDL)



5 Problem (VHDL)

(a)
```
architecture b of shifter is
begin
   out3 <= if left then in2 else if logical then 0 else in3;
   out2 <= if left then in1 else in3;
   out1 <= if left then in0 else in2;
   out0 <= if left then 0 else in1;
end b
```

(b)



(c)
```
architecture s of shifter is
signal x: Bit;
begin
   m3a: multiplexer port map(logical, 0, in3, x);
   m3b: multiplexer port map(left, in2, x, out3);
   m2: multiplexer port map(left, in1, in3, out2);
   m1: multiplexer port map(left, in0, in2, out1);
   m0: multiplexer port map(left, 0, in1, out0);
end s;
```

(d) 20ms

6 Problem (Reliability)

(a) $R(t)$ is the probability that the system works at time t, provided that it was working at time 0. More precisely, $R(t)$ is therefore the probability that the system works during the interval of time $[0 \ldots t]$.

(b) $M(t)$ is the probability that the system is repaired at time t, provided it stopped working at time 0.

(c) $A(t)$ is the probability that the system works at time t.

(d) $r(t) = \frac{-R'(t)}{R(t)}$.

7 Problem (TMR-arrangement

The probability that the correct bit is computed before the voter is

$$3(3/4)^2 - 2(3/4)^3 = 27/32\,.$$

Thus, the probability that the correct bit is computed after the voter is

$$(27/32)(4/5) + (5/32)(1/5) = 27/40 + 1/32 = 113/160\,.$$

8 Problem (Markov processes)

Solve

$$\begin{aligned} a &= (1/4)a + (3/5)b \\ b &= (3/4)a + (2/5)b \\ a + b &= 1 \end{aligned}$$

obtaining

$$\begin{aligned} a &= 4/9 \\ b &= 5/9 \end{aligned}$$

9 Problem (A/D conversion)

(a) $2^4 - 1 = 15$ comparators with reference values $2 + 0.5k$, for $k = 1, \ldots, 15$.

(b)

$$\begin{aligned} 1000 &\implies 6.0 > 5.2 && \implies 0 \\ 0100 &\implies 4.0 < 5.2 && \implies 1 \\ 0110 &\implies 5.0 < 5.2 && \implies 1 \\ 0111 &\implies 5.5 > 5.2 && \implies 0 \implies 0110 \end{aligned}$$

$$
\begin{aligned}
1000 &\implies 6.0 < 9.4 &&\implies 1\\
1100 &\implies 8.0 < 9.4 &&\implies 1\\
1110 &\implies 9.0 < 9.4 &&\implies 1\\
1111 &\implies 9.5 > 9.4 &&\implies 0 \implies 1110
\end{aligned}
$$

(c) Flash method uses constant time and linear space; successive approximation method uses logarithm time and logarithm space.

10 Problem (Scheduling)

For each processor p, associate a function

$$\sigma_p : \mathbb{R}_0^+ \to \{0, \ldots, n\}$$

with the following constraints:

(a) $(\forall t \geq 0)(\exists t_1, t_2)$ such that $t \in [t_1, t_2)$ and $\forall t' \in [t_1, t_2)$ we have $\sigma_p(t') = \sigma_p(t)$

(b) Add to (a) the following condition: $(\forall t \geq 0)(\sigma_p(t) \neq 0 \wedge \sigma_q(t) \neq 0 \implies \sigma_p(t) \neq \sigma_q(t))$

(c) Add to (a) and (b) the following condition:
$(\forall k \in \{1 \ldots, n\}(\sigma_p(t) = k \implies (\exists t_1, t_2 \geq 0)(t \in [t_1, t_2) \wedge (\forall t' \geq 0)(\sigma_p(t') = k \iff t' \in [t_1, t_2))))$

11 Problem (Periodic scheduling)

(a) An EDF schedule is

112222112222112112221122221122221122

(b) An RM schedule is

112221122211222112221122211222112122

Notice the time overflow for process 1.

12 Problem (Aperiodic scheduling)

(a) no, consider $C_1 = 2$, $d_1 = 1$, $C_2 = 1$, $d_2 = 2$.

(b) yes, because there are no idle times in EDD.

(c) no, consider $w_1 = 1$, $C_1 = 1$, $d_1 = 1$, $w_2 = 2$, $C_2 = 1$, $d_2 = 2$.

(d) no, consider $C_1 = 100$, $d_1 = 1$, $C_2 = 1$, $d_2 = 2$, $C_3 = 1$, $d_3 = 2$.

13 Problem (Aperiodic scheduling)

(a) A preemptive schedule can be turned into a nonpreemptive schedule using the construction explained in class. This construction can only improve the f_i. Therefore, it can only improve T_w.

(b) Execute tasks in nondecreasing order of w/C priority.

(c) Use swapping argument explained in class. Without loss of generality, we can assume we are swapping two consecutive tasks. Before the swap we have

xxx aaa bbb yyy

and T_w is

$$x + w_a(c_x + c_a) + w_b(c_x + c_a + c_b) + y\,.$$

After the swap we have

xxx bbb aaa yyy

and T_w is

$$x + w_b(c_x + c_b) + w_a(c_x + c_b + c_a) + y$$

Moreover, we have $w_b/c_b > w_a/c_a$ which implies that T_w becomes smaller after the swap.