jian-jia.chen [☺] tu-dortmund.de
ingo.korb [☺] tu-dortmund.de

Exercises for Lecture
Embedded Systems
Winter Semster 16/17

Written Exercise Sheet 2

(19 Punkte)

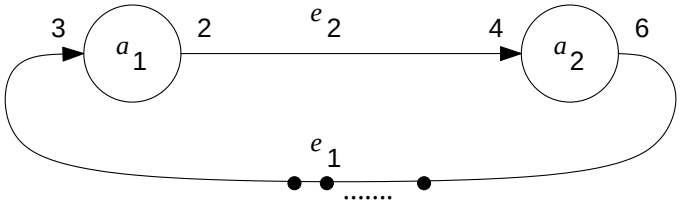
Hints: These assignments will be discussed at E23, from 10:15 am - 11:45 am on 16, Nov., 2016. You are not obligated to turn in the solutions.

1 Imperative Programming Languages (2 Punkte)

Mr. Smart wants to use Java as the programming language for his design of real-time embedded systems. What are the problems and issues that he may face? Explain the problems with imperative languages and shared memories. What is your conclusion about such models?

2 SDF (2 Punkte)

Suppose that the following SDF model is given:



Suppose that SDF actors execute in 1 unit of time. Generate a possible execution schedule, assuming a single execution unit (no parallel execution) and an initial set of 6 tokens for edge e_1:

Time	Tokens on edges		Next actor execution
	e_1	e_2	a_1 or a_2
0	6	0	
1			
2			
3			
4			

Now, assume an initial set of 9 tokens for edge e_1. Provide an example of parallel execution, i.e. an example for which both actors are active. Enter the names of the concurrently executing actors into the right column!

Time	Tokens on edges		Next actor execution(s)
	e_1	e_2	a_1, a_2 or (a_1 and a_2)
0	9	0	
1			
2			
3			
4			

3 KPN (4 Punkte)

Fibonacci numbers are defined by: $F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-2) + F(n-1) & \text{if } n > 1 \end{cases}$

Create the following processes primitives:

- **Process Init1(input A, output B):** At the start, it sends just once the integer value "1" on its output channel. Afterwards, it executes in an infinite loop: Read one value from the input channel and put it on the output channel.
- **Process Init0(input A, output B):** At the start, it sends just once the integer value "0" on its output channel. Afterwards, it has the same behavior like process init1.
- **2 x Process Dup(input A, output B, C):** It executes in an infinite loop: Read one value from the input channel and put the value on both output channels (remove "2x").
- **Process Add(input A, B, output C):** It executes in an infinite loop: Read one value from each input channel. Add the two values. Put the result on its output channel.
- **Process Sink(input A):** In an infinite loop, this process reads one value from the input channel per cycle (Could/should do nothing. . .).

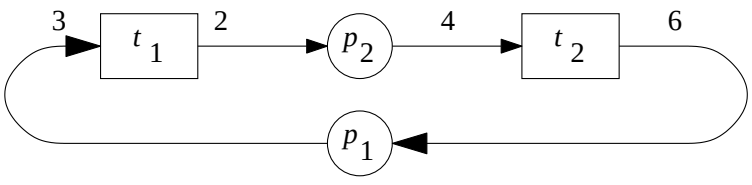
Develop a process network with only these components to produce the sequence of Fibonacci numbers.

4 Turing-complete or not? (2 Punkte)

"KPNs are Turing-complete, whereas SDFs are not Turing-complete." What does the above sentence actually mean? Being Turing-complete can be good or bad for a model of computation. Explain the good sides and the bad sides. Would it be possible to convert an SDF model into a Kahn process network? If not, what be the esential difficulty?

5 Petri Net (1) (2 Punkte)

Consider the following simple place/transition net with $M_0(p_1) = 9, M_0(p_2) = 0$:



Which combinations of markings of p_1 and p_2 are feasible? Create a graph with $(M(p_1), M(p_2))$ as nodes and indicate possible transitions. Start with a node (9,0).

6 Petri Net (2) (3 Punkte)

In der folgenden Aufgabe soll ein Petrinetz erstellt werden, welches das Problem einer unklaren Vorfahrssituation auf zwei sich kreuzenden Strassen in einem Wohngebiet modelliert.

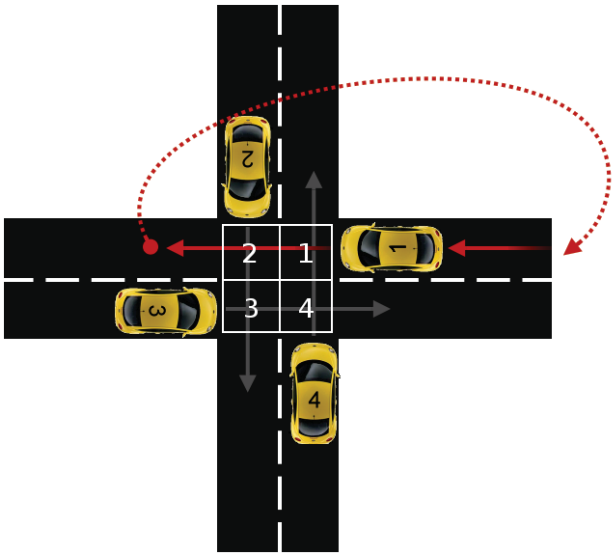
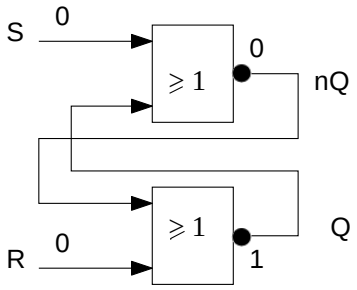


Abbildung 1: Kreuzung

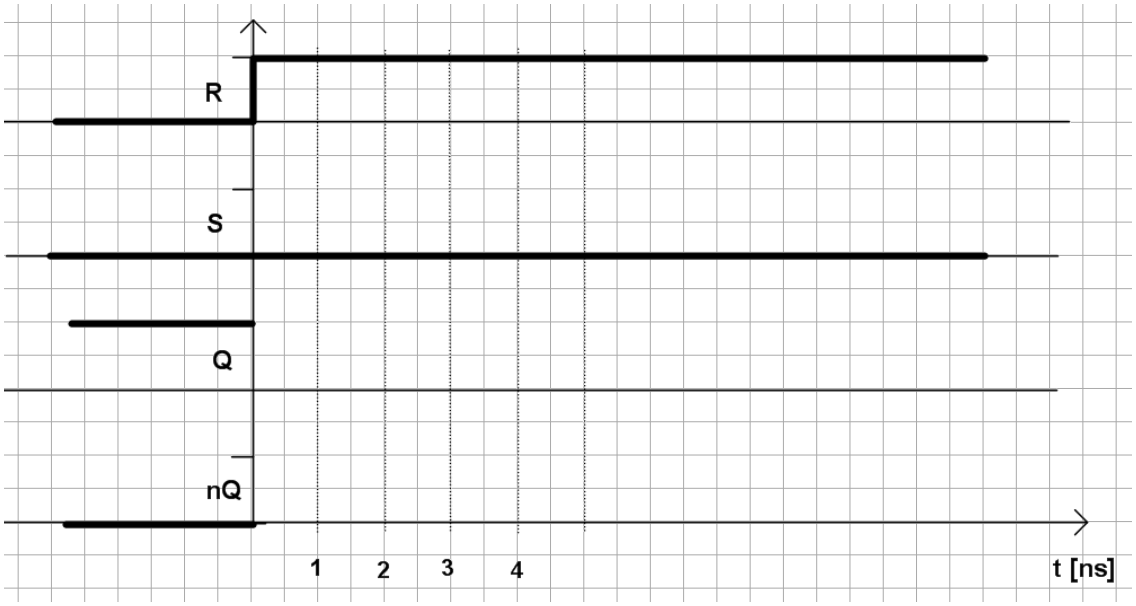
Wir gehen dazu von einer vier-armigen Kreuzung aus (siehe Abbildung 1), an der *vier* Autos stehen können, keine Ampel den Verkehr regelt, *Rechts-vor-Links* gilt und die Autos *nur geradaus* über die Kreuzung fahren können. Für eine vereinfachte Modellierung ist die Kreuzung in *vier belegbare Quadranten* eingeteilt. Das Petrinetz soll so modelliert werden, dass die Autos immer wieder an die Kreuzung zurückkehren, nachdem sie diese überquert haben. Zu Beginn befindet sich jedes Auto im Zustand *Heranfahren*. Anschließend kann ein Auto in den Zustand *Warten* wechseln und belegt den Kreuzungs-Quadranten direkt vor seiner Motorhaube. Beachten Sie, dass der Fahrer sich natürlich vergewissern muss, ob sich rechts von ihm ein Auto befindet, d.h. dass der Quadrant des rechten Nachbarn frei ist. Danach kann man die Kreuzung überqueren (im Zustand *Fahren*) und anschließend werden die beide Quadranten wieder freigegeben. Das Auto kann erneut an die Kreuzung fahren.

7 D/E Simulations (2 Punkte)

Consider the following simple cross-connected NOR-gates.

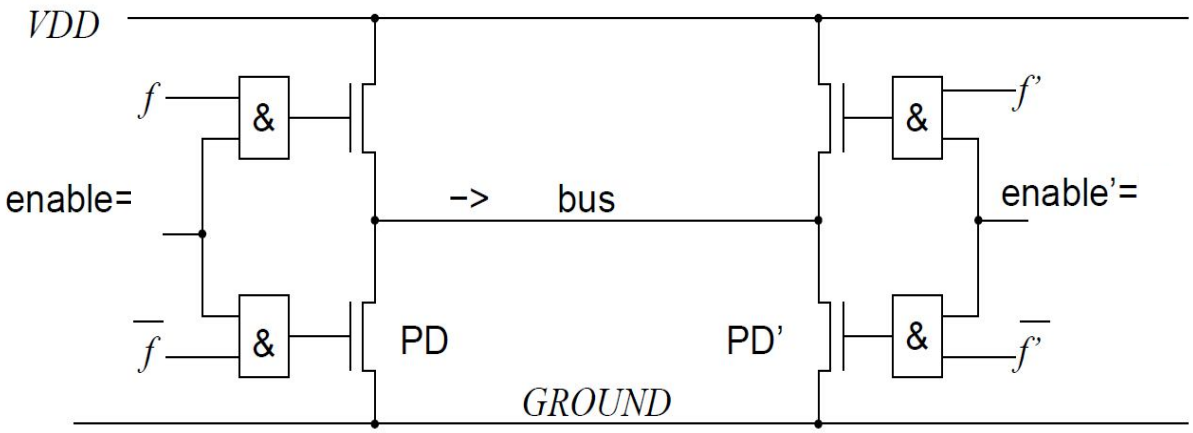


Assume that the top NOR-gate delays all changes by 2 ns, the NOR-gate at the bottom delays all changes by 3 ns. Assume that for $t < 0$, signal values are as indicated in the timing diagram below. At $t = 0$, R is assumed to change to '1'. Depict the waveforms at the outputs nQ and Q of the two gates, as computed by discrete event simulation!



8 IEEE 1164 Application (2 Punkte)

Consider the following simple bus to be modelled with the values of IEEE standard 1164



(a) Which of the IEEE 1164 values will be on the bus if both enable inputs are set to ’0’ (enable=enable’=’0’) ?

(b) Which of the IEEE 1164 values will be on the bus if enable=’0’, enable’=’1’ and f’=’1’?

General Hints: