

# Theoretisches Aufgabenblatt 2

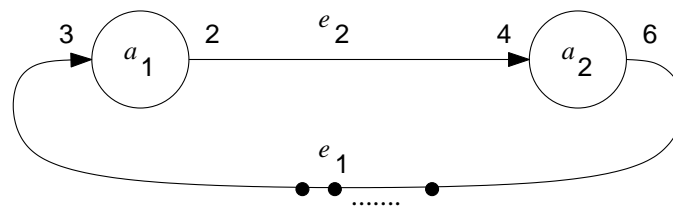
**Hinweis:** Dieses Aufgabenblatt wird am 14.11.2017 von 10.15 Uhr bis 11.45 Uhr in Raum OH14/E23 besprochen. Sie sind nicht verpflichtet, Ihre Lösungen abzugeben.

## 1 Imperative Programmiersprachen

Mr. Smart möchte die Programmiersprache Java nutzen, um ein eingebettetes Echtzeitsystem zu designen. Welche Probleme und Schwierigkeiten könnten dabei auftreten? Erklären Sie, welche Probleme durch imperative Programmiersprachen und Shared Memories verursacht werden können. Was schlussfolgern Sie in Bezug auf derartige Modelle?

## 2 SDF

Das folgende SDF-Modell sei gegeben:



Nehmen Sie an, dass die SDF-Akteure eine Zeiteinheit zur Ausführung benötigen. Erzeugen Sie einen möglichen Ausführungsschedule unter der Annahme, dass Kante  $e_1$  initial 6 Tokens besitzt und nur ein Akteur gleichzeitig aktiv sein kann (keine parallele Ausführung möglich) für sie benötigt wird:

Zeit	Tokens auf Kanten		Nächste Akteur-Ausführung
	$e_1$	$e_2$	
0	6	0	
1			
2			
3			
4			

Nehmen Sie nun an, dass Kante  $e_1$  initial 9 Tokens besitzt. Geben Sie ein Beispiel für parallele Ausführung, d. h., ein Beispiel, in dem beide Akteure aktiv sind. Notieren Sie die Namen der gleichzeitig aktiven Akteure in der rechten Spalte!

Zeit	Tokens auf Kanten		Akteur-Ausführung(en)
	$e_1$	$e_2$	$a_1, a_2$ or $(a_1 \text{ and } a_2)$
0	9	0	
1			
2			
3			
4			

### 3 KPN

Die Fibonacci-Zahlen sind folgendermaßen definiert:  $F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-2) + F(n-1) & \text{if } n > 1 \end{cases}$

Erstellen Sie die folgenden Process Primitives:

- **Process Init1(input A, output B):** Sendet zu Beginn einmal den ganzzahligen Wert "1" auf seinem Outputchannel und führt danach eine unendliche Schleife aus: Lese einen Wert vom Inputchannel und schreibe ihn in den Outputchannel.
- **Process Init0(input A, output B):** Sendet zu Beginn einmal den ganzzahligen Wert "0" auf seinem Outputchannel und verhält sich danach wie Prozess Init1.
- **Process Dup(input A, output B, C):** Führt eine unendliche Schleife aus: Lese einen Wert vom Inputchannel und schreibe ihn in beide Outputchannel.
- **Process Add(input A, B, output C):** Führt eine unendliche Schleife aus: Lese einen Wert von jedem Inputchannel, addiere die beiden Werte, schreibe das Ergebnis in den Outputchannel.
- **Process Sink(input A):** Führt eine unendliche Schleife aus: Lese einen Wert pro Zyklus vom Inputchannel. (Könnte/sollte nichts tun...).

Entwickeln Sie ein Prozessnetzwerk, das ausschließlich diese Komponenten verwendet, um die Reihe der Fibonacci-zahlen zu berechnen.

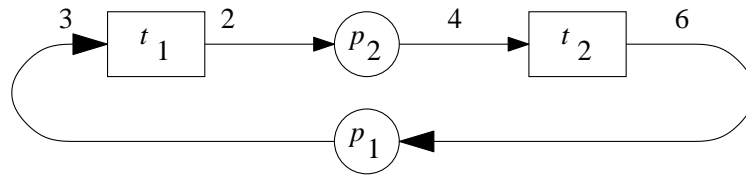
### 4 Turing-vollständig oder nicht?

"KPNs sind Turing-vollständig, wohingegen SDFs nicht Turing-vollständig sind." Was bedeutet dieser Satz? Turing-Vollständigkeit kann sowohl etwas Positives als auch etwas Negatives für ein Berechnungsmodell sein. Erklären Sie die positiven und negativen Aspekte. Ist es möglich, ein SDF-Modell in ein Kahn-Prozess-Netzwerk umzuwandeln? Falls nicht, was ist das Kernproblem?

### 5 Petrinetz (1)

Betrachten Sie das einfache Stellen/Transitions-Netz mit  $M_0(p_1) = 9, M_0(p_2) = 0$ :

Welche Kombinationen von Markierungen von  $p_1$  und  $p_2$  sind möglich? Zeichnen Sie einen Graphen mit den Knoten  $(M(p_1), M(p_2))$  und kennzeichnen Sie mögliche Transitions. Starten Sie mit einem Knoten (9,0).



## 6 Petrinetz (2)

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

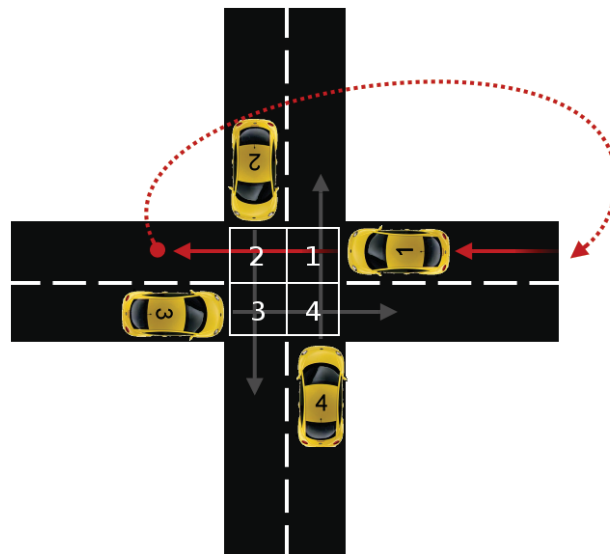


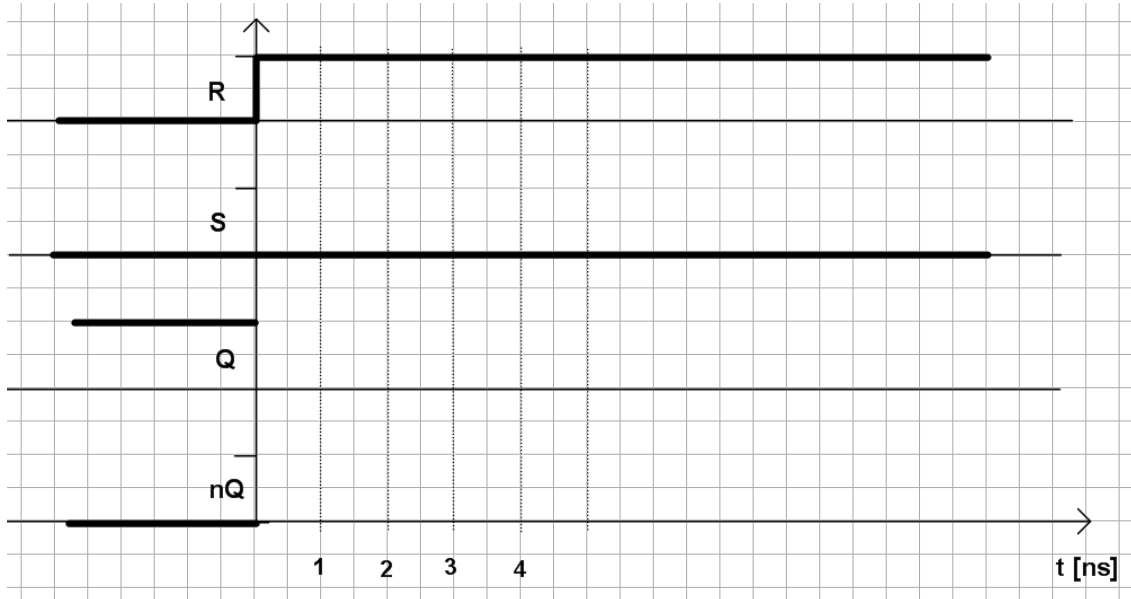
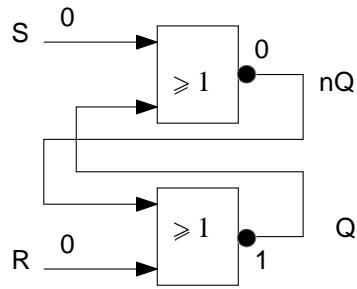
Abbildung 1: Kreuzung

Wir gehen dazu von einer vierarmigen Kreuzung aus (siehe Abbildung 1), an der vier Autos stehen können, keine Ampel den Verkehr regelt, *Rechts-vor-Links* gilt und die Autos *nur geradeaus* 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 er die Kreuzung überqueren (im Zustand *Fahren*), woraufhin die beiden Quadranten wieder freigegeben werden und sich das Auto erneut der Kreuzung nähern kann.

## 7 D/E Simulationen

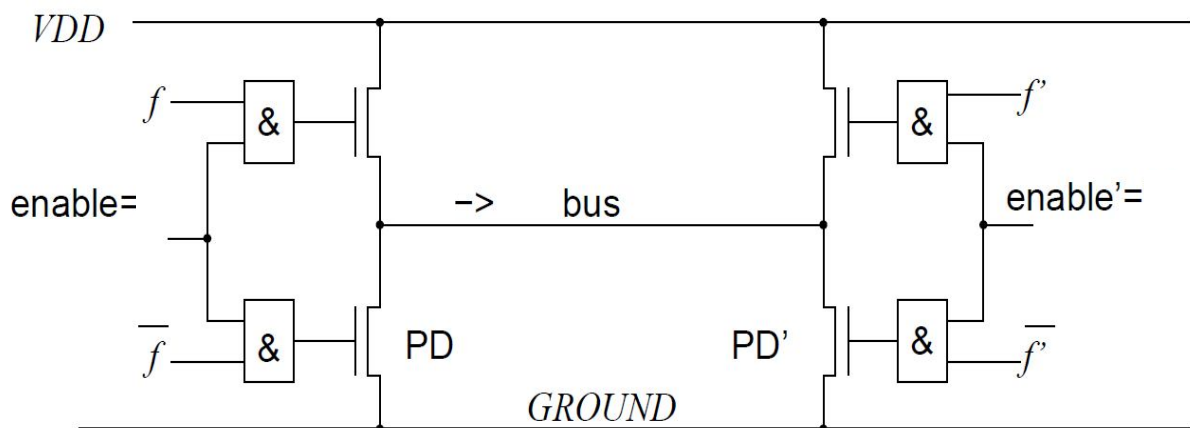
Betrachten Sie die folgenden miteinander verbundenen NOR-Gatter.

Nehmen Sie an, dass das obere NOR-Gatter alle Veränderungen um 2 ns verzögert, das untere NOR-Gatter um 3 ns. Nehmen Sie an, dass zum Zeitpunkt  $t < 0$  alle Signalwerte denjenigen im unten abgebildeten Diagramm entsprechen. Nehmen Sie ferner an, dass zum Zeitpunkt  $t = 0$  R zu '1' wechselt. Zeichnen Sie die Wellenform an den Ausgängen nQ und Q der beiden Gatter entsprechend den Berechnungen einer diskreten Ereignissimulation!



## 8 IEEE 1164 Anwendung

Betrachten Sie den folgenden einfachen Bus, der mit den Werte des IEEE-Standards 1164 simuliert wurde.



(a) Welche der IEEE 1164 Werte werden sich auf dem Bus befinden, wenn beide Inputs auf '0' gesetzt sind? (enable=enable'='0') ?

(b) Welche der IEEE 1164 Werte werden sich auf dem Bus befinden, wenn enable='0', enable'='1' und f'='1'?