jian-jia.chen [☺] tu-dortmund.de
lea.schoenberger [☺] tu-dortmund.de

Übungen zur Vorlesung
Eingebettete Systeme
Wintersemester 17/18

Theoretisches Aufgabenblatt 1

Hinweis: Dieses Aufgabenblatt wird am 24.10.2017 von 10.15 Uhr bis 11.45 Uhr in Raum OH14/E23 besprochen. Sie sind nicht verpflichtet, Ihre Lösungen abzugeben.

1 Worst-Case Execution Time

Für Eingebettete Systeme ist es wichtig, die maximale (worst-case) Ausführungszeit eines Programmes zu kennen. Wie bestimmen Sie eine solche obere Schranke? Gibt es dabei Schwierigkeiten?

Lösung: Es ist im Allgemeinen nicht möglich, die exakte WCET eines Problems zu bestimmen (nicht-berechenbar durch eine Turingmaschine), da zu diesem Zweck alle etwaigen Input Sets sowie alle initialen und zwischenzeitlichen Systemzustände berücksichtigt werden müssen. Dennoch existieren verschiedene Techniken, mittels derer eine obere Schranke berechnet werden kann wie beispielsweise Archictecture Analysis, Program Analysis etc. Die WCET ist von verschiedenen Faktoren wie Eingabeparametern (Parameter eines Algorithmus, Problemgröße etc.), der Cache-Konfiguration und der Replacement Policies, dem Pipelining, dem Scheduling, Einflüssen durch die Umgebung etc. abhängig.

2 Leistung-Energie Verbrauch

Sie verfügen über eine Batterie mit einer Kapazität von 3600 mAh (Milliamperestunden). Nehmen Sie an, die gelieferte Spannung **konstant** bei 1.1 Volt liegt, bis die Batterie leer ist. Nehmen Sie weiterhin an, dass der Leistungsverbrauch anderer elektronischer Komponenten ignoriert/vernachlässigt werden kann.

Nehmen Sie an, dass Sie ein CMOS-System betreiben, das dynamische Spannungs- und Frequenzskalierung (dynamic voltage frequency scaling (DVFS)) und dynamisches Leistungsmanagment (dynamic power management (DPM)) unterstützt. Nehmen Sie an, dass Sie jede Frequenz zwischen 10 MHz und 200 MHz wählen können. Der Leistungsverbrauch für Berechnungen mit der Frequenz f MHz ist $10^{-5} f^3 + 1$ mWatt.

Nehmen Sie ferner an, dass Sie dieses System solange mit einer Konstanten Frequenz f^* betreiben, bis die Batterie leer ist. Welche ist die maximale Frequenz f^*, durch deren Auswahl Sie sicherstellen können, dass das System für 360 Stunden in Betrieb ist?

Lösung: 100 MHz

3 H-Bahn-Utopien

Erstellen Sie ein Weg-/Zeit-Diagramm, das den Zyklus der H-Bahn-Linien 1 und 2 im Taktbetrieb darstellt. Nutzen Sie dazu die Informationen aus der Abbildung, die die Fahrzeiten für das Jahr 2018 darstellt, in dem die Bahn bereits von der Emil-Figge-Str. zu den Informatik-Gebäuden verlängert wurde, damit die Informatiker nicht immer den langen Weg zur Mensa laufen müssen. Die beiden Bahnen der Linie 1 (eine startet bei der Informatik, eine in Eichlinghofen) fahren um 10:11 los und haben in jeder Station eine Minute Aufenthalt. Die Linie 2 fährt zur selben Zeit am Nordcampus los und hat jeweils eine *halbe* Minute Aufenthalt.

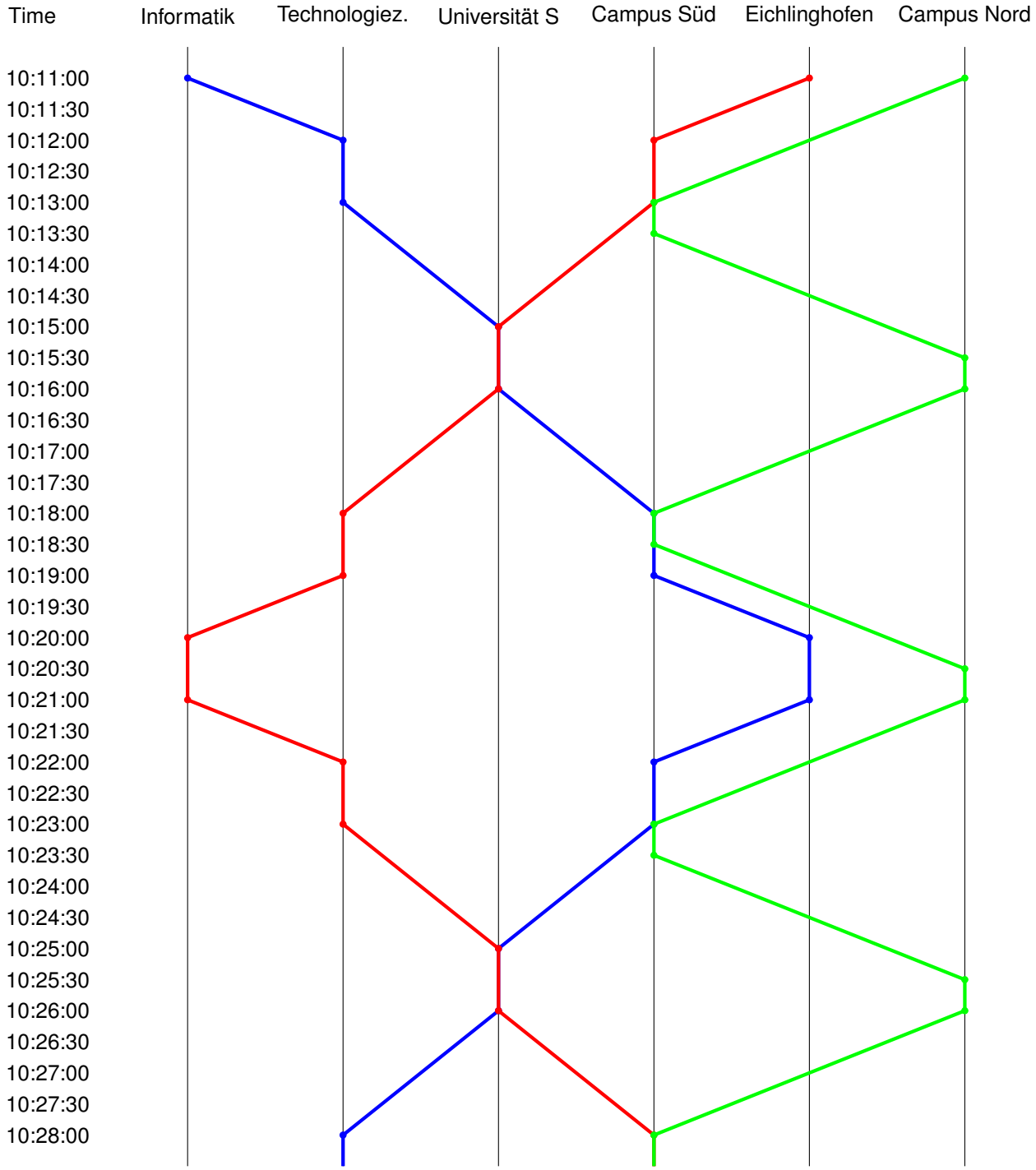
Die Abfahrtszeiten sind also:

Linie 1 Nord-Süd **10:11** Informatik, **10:13** Technologiezentrum, **10:16** Universität S, **10:19** Campus Süd, **10:21** Eichlinghofen, **10:23** Campus Süd, ...

Linie 1 Süd-Nord **10:11** Eichlinghofen, **10:13** Campus Süd, **10:16** Universität S, **10:19** Technologiezentrum, **10:21** Informatik, **10:23** Technologiezentrum, ...

Linie 2 **10:11:00** Campus Nord, **10:13:30** Campus Süd, **10:16:00** Campus Nord, **10:18:30** Campus Süd, ...

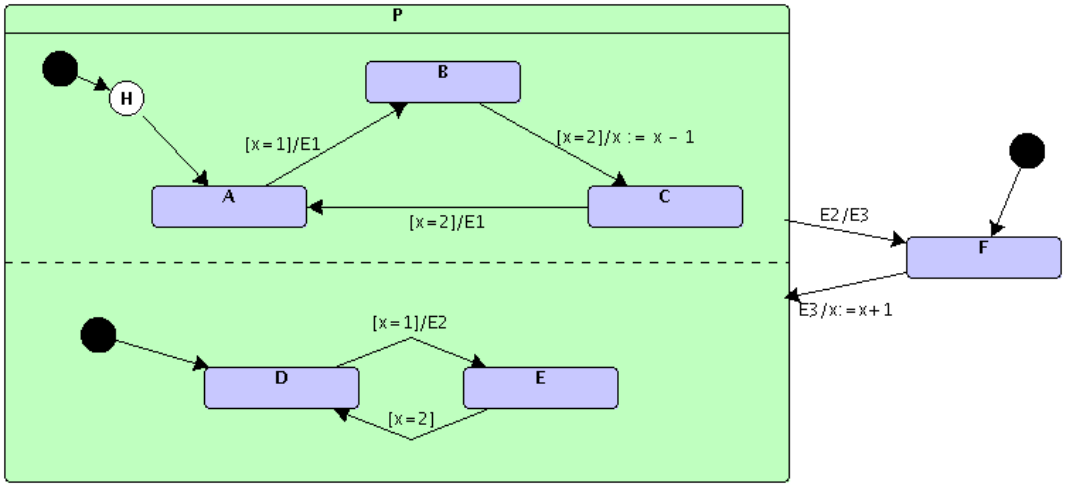
Lösung:



4 StateCharts

Gegeben sei folgendes StateChart-Diagramm:

Interpretieren Sie die Funktionsweise des StateCharts, wenn das Ereignis E3 im Zustand F mit $x = 0$ auftritt. Geben Sie zu jedem Schritt die jeweilige Konfiguration (Zustände, Events, Variablenwerte) an und begründen Sie, warum die Transition von E nach D nicht schaltet.



Lösung:

	A	B	C	D	E	F	P	x
(reset)						x		0
E3	x			x		x	x	1
$[x = 1]$		x			x		x	1
E1, E2						x		1
E3		x		x			x	2
	...	...	...	...	...	...	...	...

5 Multithreading

Gegeben sei folgender Code:

```
semaphore S;
integer u;
```

```
thread a {
u = 3;              //op a1
if u<10             //op a2
   {u = u + 1; ..} //op a3
else u=5;           //op a4
};
```

```
thread b {
u = 2;              //op b1
if u<4              //op b2
  {u = u + 4; ..}  //op b3
else u=10;          //op b4
};
```

- In einem Mikroprozessorsystem soll jeder Thread nur einmal ausgeführt werden.
- Threads können in beliebiger Reihenfolge ausgeführt werden und der ausgeführte Thread kann jederzeit Wechseln.
- Zur Vereinfachung wird angenommen, dass jede Operation atomar ausgeführt wird, d.h., während der Ausführung einer Zeile ax bzw. by kann kein Wechsel (genannt Kontextwechsel) stattfinden.
- Welche Werte kann u nach vollständiger Ausführung beider Threads annehmen?

Das Diagramm am Ende dieses Übungsblatts zeigt eine partielle Darstellung der möglichen Ausführungssequenzen. Jede Sequenz entspricht einem Pfad von oben nach unten durch den Graphen. Vervollständigen Sie das Diagramm und notieren Sie die Werte für u an den Kanten. Sequenzen, die aufgrund des Werts von u nicht erreicht werden können, sollen durchgestrichen oder ausgelassen werden.

Lösung:

- `Start` $\rightarrow$ `Thread a` $\rightarrow \ldots \rightarrow 6,7,7,7,7,7,7,7,7,7$
- `Start` $\rightarrow$ `Thread b` $\rightarrow \ldots \rightarrow 10,8,8,8,8,8,8,8,8,4$

6 Semaphores

Nun wird der Code aus der vorherigen Aufgabe so modifiziert, dass die if-Abfragen durch Semaphoren geschützt werden:

```
semaphore S;
integer u;
```

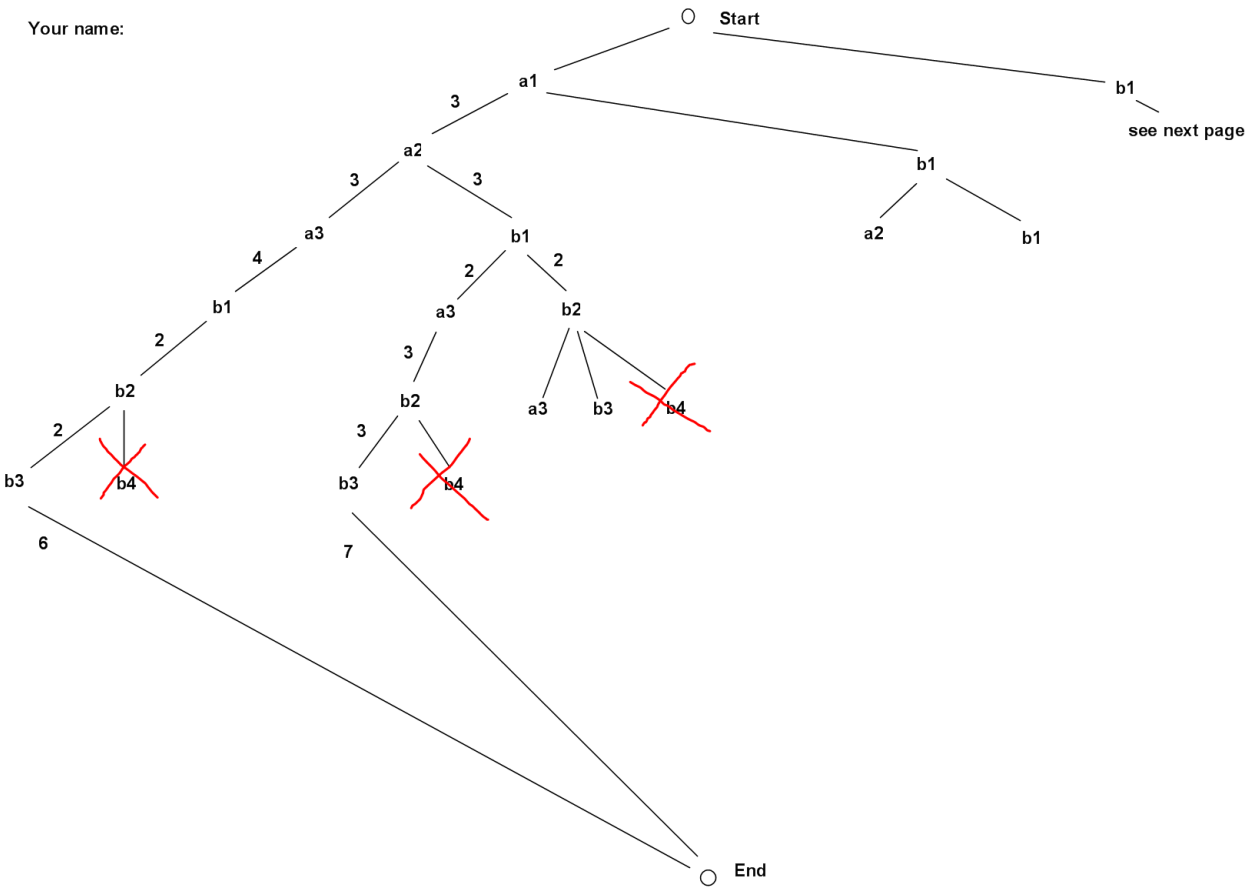
```
thread a {
u = 3;              //op a1
P(S);
if u<10             //op a2
   {u = u + 1; ..} //op a3
else u=5;           //op a4
V(S);
};
```

```
thread b {
u = 2;              //op b1
P(S);
if u<4              //op b2
  {u = u + 4; ..}  //op b3
else u=10;          //op b4
V(S);
};
```

Beide Threads können wieder in beliebiger Reihenfolge abgearbeitet werden, solange die durch die Semaphoren gegebenen Ausschlüsse beachtet werden, und die Ausführung kann wieder jederzeit zwischen beiden Threads wechseln. Zeichnen Sie einen Graphen, der ähnlich wie in der vorherigen Aufgabe die möglichen Ausführungspfade der Operationen zeigt. Ergänzen Sie wieder die Werte von u an den Kanten des Graphen. Wie zuvor können nicht erreichbare Sequenzen durchgestrichen oder ausgelassen werden.

Lösung:

- `Start` $\rightarrow$ `Thread a` $\rightarrow \ldots \rightarrow 6,7,7,7$
- `Start` $\rightarrow$ `Thread b` $\rightarrow \ldots \rightarrow 4,8,10,8$


Your name:
Start
a1
b1
see next page
3
a2
b1
3
3
a3
b1
a2
b1
4
2
2
b1
a3
b2
2
3
b2
a3
b3
b4
2
3
b3
b4
b3
b4
6
7
End

Your name