

Written Exercise Sheet 1

(20 Punkte)

Hints: These assignments will be discussed at E23, from 10:15 am - 11:45 am on 03, Nov., 2015. You are not obligated to turn in the solutions. Exception! If you would like to turn in your answers regarding to Question 3, please hand in your solutions to Prof. Jian-Jia Chen after the lecture session on 03, Nov. 2015. we will organize to give you feedbacks.

1 Worst-Case Execution Time (2 Punkte)

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?

2 Leistung-Energie Verbrauch (3 Punkte)

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

Nehmen Sie an, dass sie ein CMOS-System betreiben, dass dynamische Spannungs- und Frequenzskalierung (dynamic voltage frequency scaling (DVFS)) und dynamisches Leistungs-Management (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.

Suppose that you need to operate this system continuously with a constant frequency f^* until the battery gets empty. What is the maximum frequency f^* that can be chosen to ensure that the system can be operated consecutively for 360 hours?

3 Kaffe- oder Teeautomat (6 Punkte)

Sie werden gebeten, einen Kaffee- oder Teeautomaten zu entwerfen, den Sie später selbst verwenden. Für diese Ausgabe betrachten wir nur das grundlegende Design, ohne auf Details der Hardwarekomponenten einzugehen. Bitte formulieren Sie schriftlich:

- Ausformulierte Anforderungen und Spezifikationen für Ihren Kaffee-/Teeautomaten,
- Anwendungsfälle für den von Ihnen entworfenen Automaten und
- Sequenzdiagramme zu dem von Ihnen entworfenen Automaten für das Brauen einer sowie zwei Tassen Kaffee oder Tee

4 H-Bahn-Utopien (2 Punkte)

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 2015 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

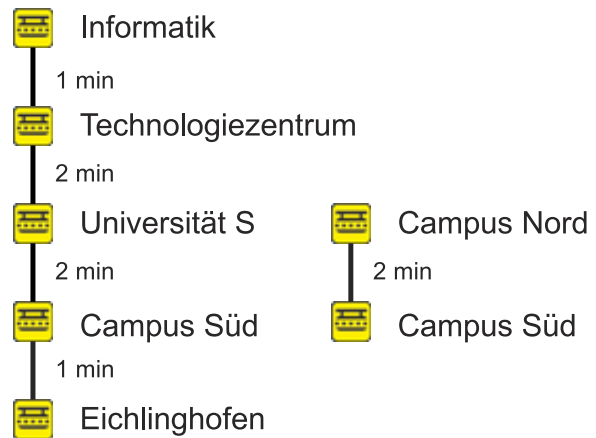
zur Mensa laufen müssen. Die beiden Bahnen der Linie 1 (eine startet bei der Informatik, eine in Eichlinghofen) fahren um 10:11 los und haben in jeder Station eine Minute Aufenthalt. Die Linie 2 fährt zur selben Zeit am Nordcampus los und hat jeweils eine *halbe* Minute Aufenthalt.

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, ...

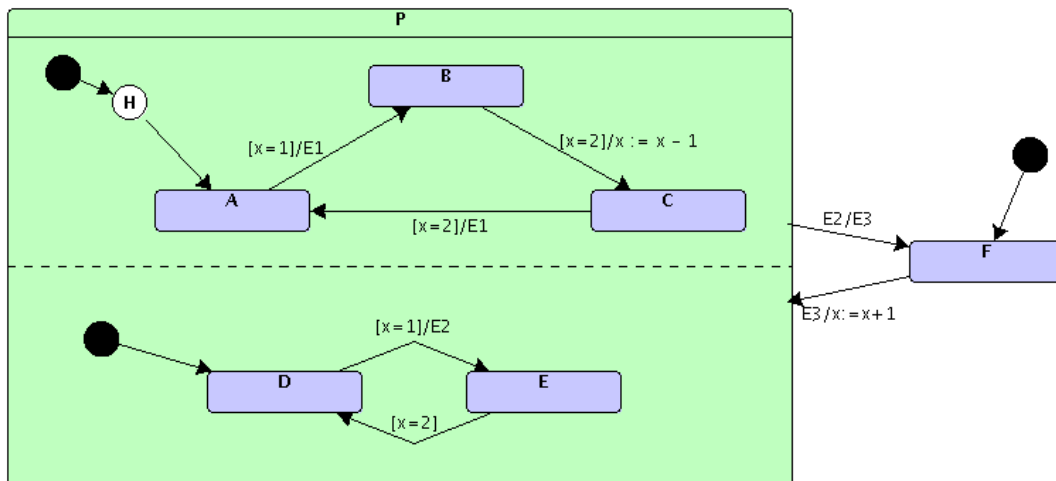
Linie 1 Süd-Nord 10:11 Eichlinghofen, 10:13 Campus Süd, 10:16 Universität S, 10:19 Technologiezentrum, 10:21 Informatik, 10:23 Technologiezentrum, ...

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



5 StateCharts (2 Punkte)

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.

6 Multithreading (3 Punkte)

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
};
```

- Jeder Thread soll nur einmal ausgeführt werden an ein Mikorprocessorsystem.
- Threads können in beliebiger Reihenfolge ausgeführt werden und die Ausführung kann jederzeit zwischen ihnen wechseln
- Zur Vereinfachung wird angenommen, dass jede Operation atomar ausgeführt wird, d.h. während der Ausführung einer Zeile a_x bzw. b_y kann kein Wechsel (genannt Kontextwechsel) auftreten.
- 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.

7 Semaphores (2 Punkte)

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

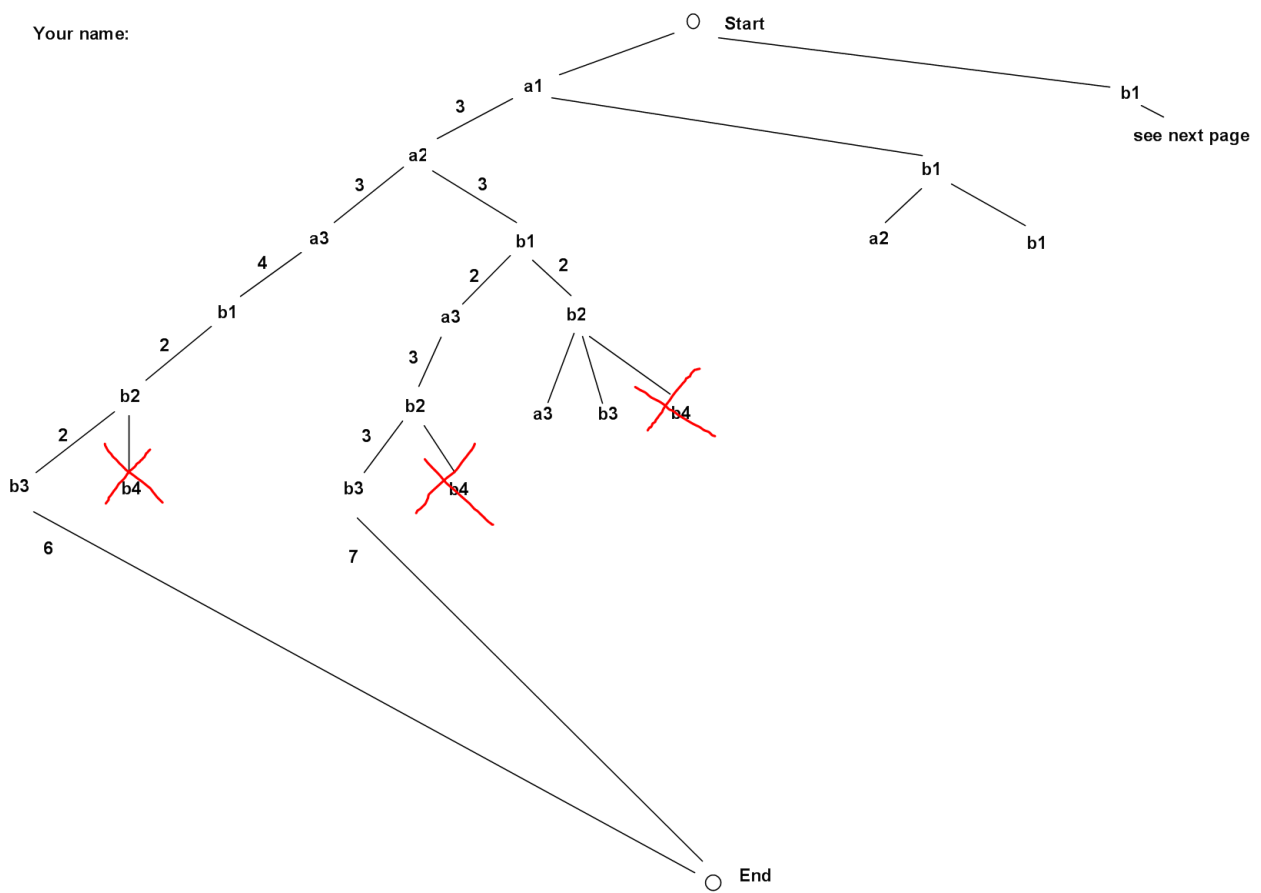
```
semaphore S;
integer u;

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);
};

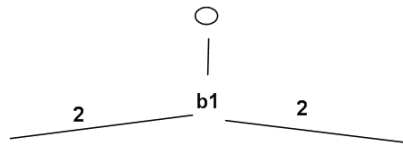
thread b {
    u = 2;           //op b1
    P(S);
    if u<4          //op b2
        {u = u + 4; ..} //op b3
    else u=10;      //op b4
    V(S);
};
```

Beide Threads können wieder in beliebiger Reihenfolge abgearbeitet werden, so lange 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.

Your name:



Your name



General Hints: