

# 10. Aufgabenblatt

Abgabe 04.07.14

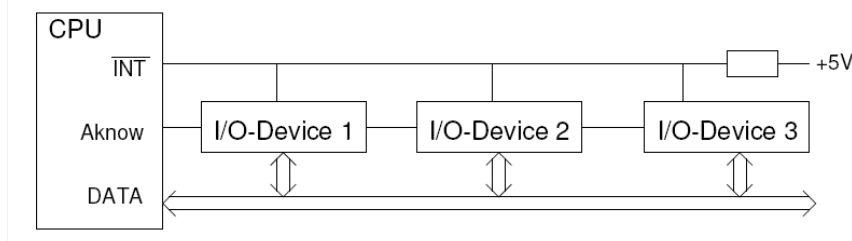
*Hinweis zu Programmieraufgaben:* Bitte machen Sie zu jeder Programmieraufgabe Testläufe, die die Funktionalität Ihrer Programme ausreichend dokumentieren. Die Testläufe und den Quellcode drucken Sie bitte aus, und geben beides mit der jeweiligen Übung auf Papier ab. Der Quellcode muss zusätzlich in Form einer assemblierbaren Datei in das Repository <https://svn.imp.fu-berlin.de/cst-teaching-schmitt/SS14/TI-2/Tut-<NUM>/grp<NUM>> eingchecked werden. Ihr jeweiliger Tutor wird Ihnen Näheres mitteilen.

## Problem 1: Interrupts

- Kann ein Interrupt einen anderen unterbrechen?
- Welche Varianten gibt es beim Interruptpolling und wie unterscheiden sich diese?

## Problem 2: Interrupt-Behandlung

- Welches Problem des Interrupt-Polling wird durch das Daisy-Chain-Verfahren gelöst?
- Beschreiben Sie den Ablauf einer Interrupt-Behandlung mit dem Daisy-Chain-Verfahren?
- Erlaubt das Daisy-Chain-Verfahren eine „faire“ Prozessorzuteilung?
- Beispiel: Jedes I/O-Device hat eine Durchlaufzeit der Daisy-Chain von 133ns. Legt das Device die Kennung auf den Datenbus, so benötigt es dazu 100ns. Die CPU übernimmt dann weitere 100ns später die Kennung und schlägt die Sprungadresse in der Vektortabelle nach.  
Wie lange dauert es maximal vom Interrupt bis die Interrupt Service Routine (ISR) angesprungen wird, wenn eine Befehlsausführung 333ns dauert und I/O-Device 3 einen Interrupt auslöst (Nachschlagen in der Interrupt-Vektortabelle dauert 50ns)?



## Problem 3: BubbleSort

Implementieren Sie den BubbleSort-Algorithmus in x86 Assembler.

Hinweis: Für diese Aufgabe sollten Sie sich mit den verschiedenen Speicherzugriffsmöglichkeiten/Adressierungsmöglichkeiten des x86 auseinandersetzen.