

## 5. Aufgabenblatt

Ausgabe    Abgabe  
14.06.10   25.06.10

Kontakt bei Fragen

Matthias Dräger, Viet Do, Marco Jeschke, Uwe Kuehn, Markus Rudolph  
{mdraeger/do/kuehn/mjeschke/rudolph}@mi.fu-berlin.de

### Problem 1: Speedup durch Parallelverarbeitung (2+2=4)

Ein Programm besitze einen seriellen Anteil von 15% und einen parallelisierbaren Anteil von 85%. Wie groß wird der durch Parallelisierung erzielbare maximale Speedup

- bei 32 Prozessoren, wenn kein Overhead berücksichtigt werden muss?
- wenn auf einem homogenen Multiprozessorsystem die Zahl der Prozessoren gegen unendlich geht?

Anmerkung: der Kommunikationsaufwand kann vernachlässigt werden.

### Problem 2: Adressierungsarten (4)

Nennen Sie drei Adressierungsarten bei Assemblerbefehlen Verwendung finden. Welche davon finden Sie bei MMIX vor?

### Problem 3: Adressierungsarten (7)

Erläutern Sie die folgenden Adressierungsarten:

- implizite Register-Adressierung mit Flag
- unmittelbare Speicheradressierung
- absolut-direkte Speicheradressierung
- Register-indirekte Speicheradressierung
- Register-relativ-indizierte Speicheradressierung
- Register-relativ-indirekt-indizierte, zweistufige Speicheradressierung
- Programmzähler-relativ-indirekte, zweistufige Speicheradressierung

### Problem 4: Interrupt-Behandlung (1+3+1+2=7)

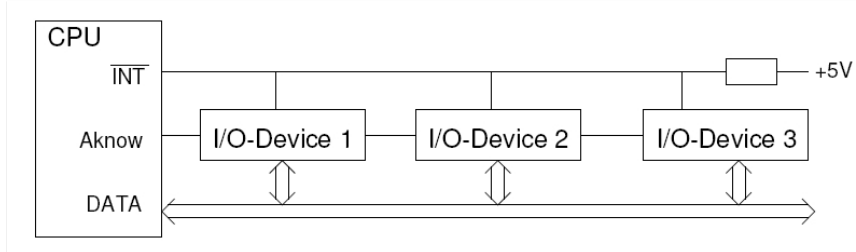
- Welches Problem des Interrupt-Polling wird durch das Daisy-Chain-Verfahren gelöst?
- Beschreiben Sie den Ablauf einer Interrupt-Behandlung mit dem Daisy-Chain-Verfahren?
- Erlaubt das Daisy-Chain-Verfahren eine „faire“ Prozessorzuteilung?

# TI II

Sommersemester 2010  
PD Dr. Katinka Wolter

---

- d) Beispiel: Jedes I/O-Device hat eine Durchlaufzeit der Daisy-Chain von 160ns. Legt das Device die Kennung auf den Datenbus, so benötigt es dazu 80ns. 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) angesprochen wird, wenn eine Befehlsausführung 320ns dauert und I/O-Device 4 einen Interrupt auslöst (Nachschlagen in der Interrupt-Vektortabelle dauert 60ns)?



## Problem 5: Interrupts (1+3=4)

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

## Problem 6: Caching (1+3+2+2+2=10)

- Welchen Vorteil bringt die Aufteilung in Tags und Index zur Adressierung eines Datums im Cache?
- An wie vielen Stellen muss ein Block bei voll-assoziativ, direct-mapped und n-way-set Caches gesucht werden?
- Was versteht man unter Cache-Kohärenz? Warum strebt man nicht Konsistenz an?
- Welche generellen Vor- und Nachteile weisen virtuell bzw. physikalisch adressierte Cache-Speicher auf?
- Wie wirken sich Prozesswechsel auf den jeweiligen Cache aus? Beachten Sie die Position der MMU!

## Problem 7: Stackprogrammierung in MMIX (12)

Programmieren Sie für einen OCTA-Stack in MMIX die folgenden Unterprogrammaufrufe

- push - Schiebt ein OCTA auf den Stack.
- pop - Liefert das zuletzt auf den Stack gelegte OCTA und löscht dieses.
- top - Liefert das zuletzt auf den Stack gelegte OCTA.
- isEmpty - prüft ob der Stack leer ist.

Überlegen Sie sich hierzu geeignete Übergabe- und Returnparameter. Die Unterprogramme sollen mit mehreren Instanzen des Stacks funktionieren können.

Hinweis: Verwenden Sie PUSHJ und POP.