



**Karlsruher Institut für Technologie**  
Institut für Technische Informatik  
Prof. Dr. Wolfgang Karl

**Klausur Rechnerstrukturen**  
**Sommersemester 2010**  
**Musterlösung**

Aushang der Ergebnisse: ab Mitte Oktober 2010

## Musterlösung 1: Quantifizierung

10P

### Kriterien zur Leistungsbewertung

3 P

Anwendersicht( $\frac{1}{2}$ P): Reduzierung von Antwortzeit( $\frac{1}{2}$ P), und Ausführungszeit( $\frac{1}{2}$ P)

Betriebssicht( $\frac{1}{2}$ P): Anzahl durchgeführter Jobs( $\frac{1}{2}$ P) (pro Zeiteinheit) und Energieeffizienz( $\frac{1}{2}$ P)

### Leistungsbewertung

3P

$$a) i_{fpu} = \sum i_{typ} = (2.250 + 250 + 200 + 300) * 10^3 = 3.000.000$$

2P

$$i_{crypto} = (500 + 2.000 + 1.000 + 200 + 300) * 10^3 = 4.000.000$$

$$cyc_{fpu} = (2.250 * 2 + 250 * 6 + 200 * 5 + 300 * 10) * 10^3 = 10 * 10^6$$

$$\rightarrow t_{fpu} = t_{cyc} * cyc_{fpu} = 2ns * 10 * 10^6 = 20ms (\frac{1}{2}P)$$

$$cyc_{crypto} = (1.000 + 4.000 + 1.000 + 200 * 5 + 300 * 10) * 10^3 = 10 * 10^6$$

$$\rightarrow t_{crypto} = t_{cyc} * cyc_{crypto} = 2ns * 10 * 10^6 = 20ms (\frac{1}{2}P)$$

$$MIPS_{fpu} = \frac{i}{t * 10^6} = \frac{3.000.000}{20 * 10^{-3}s * 10^6} = \frac{3.000}{20s} = 150MIPS(\frac{1}{2}P)$$

$$MIPS_{crypto} = \frac{4.000.000}{20 * 10^{-3}s * 10^6} = \frac{4.000}{20s} = 200MIPS(\frac{1}{2}P)$$

- b)  $Speedup_{crypto} = \frac{CPUtime_{fpu}}{CPUtime_{crypto}} = \frac{20ms}{20ms} = 1$  ( $\frac{1}{2}$ P) beide Lösungen sind gleich schnell, weshalb Sie die ursprüngliche Lösung empfehlen, da diese weniger Instruktionen zur Formulierung des Gesamtproblems braucht. ( $\frac{1}{2}$ P) 1P

### Benchmarks zur Leistungsbewertung

4 P

- c) CPU2006 besteht aus CINT2006 für die Bewertung rechenintensiver Ganzzahlleistung ( $\frac{1}{2}$ P) und CFP2006 zur Bewertung rechenintensiver Fließkommaleistung ( $\frac{1}{2}$ P). 1P

- d) Base und Peak. Diese ermöglichen sowohl den Vergleich des konservatives Setups zweier Rechensysteme als auch den Vergleich nach aggressiven, herstellerepezifischen Optimierungen. 1P

- e) Man bildet das geometrische Mittel über die bezgl. der Referenzmaschine normalisierten Ausführungszeiten der einzelnen Programme  $\sqrt[n]{\prod_{i=1}^n \text{Execution time ratio}_i}$  wenn  $i$  ein Benchmark der Gruppe ist ( $\frac{1}{2}$ P). Für das geometrische Mittel gilt:  $\text{geometrisches Mittel}(\frac{X_i}{Y_i}) = \frac{\text{geometrisches Mittel}(X_i)}{\text{geometrisches Mittel}(Y_i)}$  damit ergibt sich trotz der Relation zur Referenzmaschine ein konsistenter Wert beim Vergleich zweier Systeme. Dies wäre beim arithmetischen Mittel nicht gegeben. ( $\frac{1}{2}$ P) 1P

- f) Kernels sind die rechenintensiven Teile realer Programme. ( $\frac{1}{2}$ P) Beispiele hierfür sind: Lawrence Livermore Loops, BLAS, LINPACK ( $\frac{1}{2}$ P) 1P

## Musterlösung 2: Fragen des Rechnerentwurfs

10P

### Low-Power-Entwurf

3P

a)  $P_{\text{Switching}} = C_{\text{eff}} * V_{dd}^2 * f$

1,5P

Die Spannung trägt quadratisch zur Leistungsaufnahme beim Schalten bei. Desweiteren ist die Aufnahme proportional zur Frequenz sowie zur effektiven Kapazität. Damit  $C_{\text{eff}}$  schneller geladen werden kann, müssen die Taktflanken steiler sein. Dazu wiederum muss die Spannung höher werden, was aber den o.g. Nachteil des quadratischen Beitrags hat.

b)  $\mathbb{P}_{\text{Schalt}} = 2 * \mathbb{P}_{\text{Ausgangssignal}}(1) * (1 - \mathbb{P}_{\text{Ausgangssignal}}(1))$

1,5P

$$\mathbb{P}_{\text{Ausgangssignal ODER}}(1) = 1 - \mathbb{P}_{\text{Ausgangssignal}}(0) = 1 - \frac{1}{4} * \frac{3}{4} = \frac{13}{16}$$

$$\mathbb{P}_{\text{Schalt}} = 2 * \frac{13}{16} * (1 - \frac{13}{16}) = \frac{2*13*3}{16*16} = \frac{39}{128}$$

### Schaltungsentwurf

7P

c)

3P

- Datenverfeinerung: Abstrakte Streams müssen in konkrete Signale gewandelt werden, abstrakte Datentypen wie Vektoren werden zu Bit-Array umgesetzt.
- Strukturverfeinerung: Implementierung einer Architektur über die Angabe und Verdrahtung von Unterkomponenten
- Verhaltensverfeinerung: Eine Blackbox wird bezüglich ihres Verhaltens beschrieben

d)

2P

- Entity: Beschreibung der äußeren Schnittstelle und ggf. der Parametrisierung
- Architecture: Implementierung zu einer Entity; kann strukturell, Verhalten oder Mischung aus beidem sein
- Configuration: Zuweisung von Implementierung zu verwendeten Komponenten und Entities
- Process: Gebündelte Beschreibung des (Teil-)Verhaltens einer Implementierung als Reaktion auf Ereignisse oder Auslöser von Ereignissen

e) Die Beschreibung implementiert einen einfachen Zähler. Alle 64 Takte wird das Signal flag auf 1 gesetzt, wobei im ersten Durchlauf 64 Takte durchlaufen werden vor dem Setzen, und anschließend jeweils 63 Takte lang flag auf 0 gesetzt ist und dann für die Dauer eines Taktes auf 1 jeweils.

2P

Das Verschieben ändert nichts am Ergebnis, da alle Zuweisungen in Hardware parallel ausgeführt werden und somit der Vergleich in beiden Varianten zeitgleich mit der Erhöhung stattfindet. Die Verschiebung hilft dennoch, die Sensibilität für den Zeitpunkt der Abfrage und des aktuellen Werts zu erhöhen und kann somit aus Verständnisgründen sinnvoll scheinen.

## Musterlösung 3: Prozessorarchitektur

10P

### Pipelining und Leistungsbewertung

5P

a) je 0.5P

1,5P

Sequentielle Ausführung:  $T_{\text{seq}} = n * k$ Ausführung in Pipeline:  $T_{\text{pipe}} = n + k - 1$ Speedup  $S = \frac{T_{\text{seq}}}{T_{\text{pipe}}} = \frac{n*k}{n+k-1}$ 

b)

2,5P

$$T_{\text{Takte}} = n + \text{Verzögerungen} + k - 1 = (1 + 0,03 * 2 + 0,08 * 3) * 10.000.000 + 6 - 1 = 1,3 * 10.000.000 + 5 = 13.000.005$$

$$T_{\text{Sekunden}} = \frac{13.000.005}{2,6\text{GHz}} \approx \frac{13 * 10^6}{2,6 * 10^9} s = 5 * 10^{-3} s = 0,005 s = 5ms$$

$$S = \frac{10.000.000 * 6}{13.000.005} \approx \frac{60}{13} \approx 4,6$$
CPI-Wert:  $\frac{13.000.005}{10.000.000} \approx 1,3$ 

c)

1P

CPI-Wert von 1,1  $\Rightarrow$  #Takte = 1,1 \* #Instruktionen = 11.000.000Frequenz = # Instruktionen / Ausführungszeit =  $11 * 10^6 / 0,007 s = \frac{11}{7} * 10^9 Hz \approx 1,5GHz$ Speedup von A gegenüber B:  $S = \frac{7ms}{5ms} = 1,4$ 

Wir wählen Prozessorarchitektur A nicht, obwohl sie die Ausführungszeit stark verbessern kann. Stattdessen wählen wir B, weil die Leistungsaufnahme zum Schalten nur  $\frac{1,5}{2,6} \approx 0,6$  der von Architektur A beträgt bei vergleichsweise geringem Slowdown um  $1 - \frac{5}{7} \approx 0,3$  (Faktor  $5/7 \approx 0,714$ ) und somit effizienter ist, denn das Verhältnis von Slowdown zu Leistungsaufnahme beträgt  $0,7 : 0,6 \approx 1,167 > 1$  (Architektur B hat eine deutlich geringere Leistungsaufnahme bei nicht-proportionaler Verringerung der Ausführungsgeschwindigkeit).

### Sprungvorhersage

3P

d) Der Korrelationsprädiktor ist in Abb. 1 dargestellt.

1,5P

e) Sprungverlaufstabelle:

1,5P

| Verlauf | Prädiktor | Vorhersage | Sprung | Akt. Verlauf | Aktualisierter Prädiktor |
|---------|-----------|------------|--------|--------------|--------------------------|
| T       | (NT,NT)   | NT         | T      | T            | (NT, T)                  |
| T       | (NT,T)    | T          | NT     | NT           | (NT,NT)                  |
| NT      | (NT, NT)  | NT         | T      | T            | (T,NT)                   |

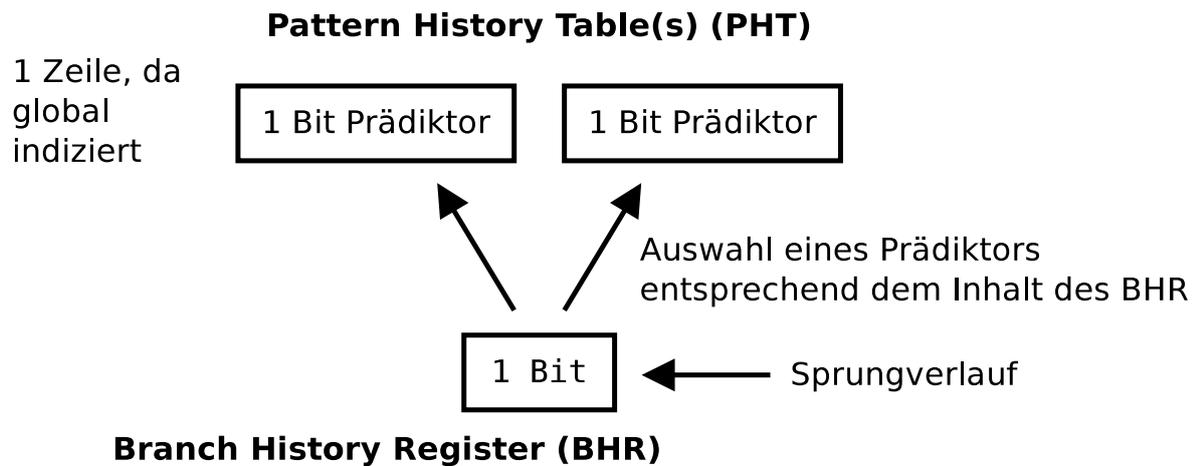


Abbildung 1: (1,1)-Korrelationsprädiktor mit global indizierter PHT

**Superskalartechnik****2P**

f) Registerstatustabelle:

|                           |   |           |   |   |      |   |
|---------------------------|---|-----------|---|---|------|---|
| Register                  | 1 | 2         | 3 | 4 | 5    | 6 |
| Wert                      | - | (R4)-(R3) | - | - | (R5) | - |
| Gültig?                   | 0 | 1         | 0 | 0 | 1    | 0 |
| Reservation Station (RS#) | 3 | 0         | 1 | 2 | 0    | 4 |

Reservation Stations:

| RS#              | Leer | In FU | Opcode | Ziel | Quelle1   | Gültig1 | RS1 | Quelle2 | Gültig2 | RS2 |
|------------------|------|-------|--------|------|-----------|---------|-----|---------|---------|-----|
| 1 Addierer       | 0    | 0     | sub    | 3    | (R1)      | 0       | 3   | (R5)    | 1       | 0   |
| 2 Addierer       | 0    | 1     | add    | 4    | (R4)-(R3) | 1       | 0   | (R3)    | 1       | 0   |
| 3 Multiplizierer | 0    | 1     | mul    | 1    | (R3)      | 1       | 0   | (R5)    | 1       | 0   |
| 4 Dividierer     | 0    | 0     | div    | 6    |           | 0       | 3   | (R4)    | 1       | 0   |

## Musterlösung 4: Parallelverarbeitung

10P

### Verbindungsstrukturen

4P

- a) • (0,5P) Verbindungsgrad: 6 2P  
 • (0,5P) Diameter:  $\frac{3*(n-1)}{2}$   
 • (0,5P) Minimale Bisektionsbreite:  $2 * n^2$   
 • (0,5P) Skalierbarkeit: gut!  
 Eine Erweiterung ist einfach durch Hinzufügen einer Ebene möglich. Der Verbindungsgrad der Knoten ändert sich nicht.

- b) (1P bei Nennung einen der beiden Punkte) 1P

- Unterstützung von One-to-all-Broadcasts
- Unterstützung von Reduktionsoperationen

- c) Das Grundmuster der einzelnen Stufen entspricht der Mischpermutation. 1P

### Quantitative Maßzahlen

3P

- d) • (0,5P)  $T = T_{comp} + T_{comm} + T_{idle}$  1P  
 • (0,5P) Berechnungszeit (Computation Time),  
 Kommunikationszeit (Communication Time),  
 Untätigkeitszeit (Idle Time)
- e) Antwort (0,5P): 2P  
 Die erreichbare Beschleunigung wird durch den sequentiellen Programmteil begrenzt.

Amdahls Gesetz (0,5P):

$$T(n) = \frac{1}{n} * T(1) * (1 - a) + T(1) * a$$

Herleitung (0,5P) und einfache Formel (0,5P):

$$\begin{aligned} S(n) &= \frac{T(1)}{T(n)} \\ &= \frac{T(1)}{\frac{1}{n} * T(1) * (1 - a) + T(1) * a} \\ &= \frac{1}{\frac{1}{n} * (1 - a) + a} \\ S(n) &\leq \frac{1}{a} \end{aligned}$$

**Vektorverarbeitung****3P**

f) Unter einem Vektor wird in der Vektorverarbeitung ein Array (Feld) von Gleitpunktzahlen verstanden. *1P*

g) (Pro Begriff 0,5P) *2P*

1. Vektor-Pipeline-Parallelität
2. Mehrere Vektor-Pipelines in einer Vektoreinheit
3. Vervielfachung der Pipelines
4. Mehrere Vektoreinheiten

## Musterlösung 5: Speicherhierarchie

10P

a)

1P

- Überlappende Zugriffe auf Instruktionen und Daten
- keine gegenseitig Verdrängung von Daten und Instruktionen

$$b) r_{H-L1} = \frac{\#Hits}{\#Zugriffe} = \frac{\#Inst * r_{H-I} + \#Daten * r_{H-D}}{\#Inst + \#Daten} = \frac{10000 * 0,9 + 5000 * 0,6}{10000 + 5000} = \frac{12000}{15000} = 80\%$$

1P

$$c) t_a = r_{H-L1} * t_{H-L1} + (1 - r_{H-L1}) * t_{Mem}$$

0.5P

$$d) t_a = 80\% * 2 ns + 20\% * 100 ns = 1,6 ns + 20 ns = 21.6 ns$$

0.5P

e) Bus-Snooping ( $\frac{1}{2}$ P), welches einen gemeinsamen Datenbus voraussetzt ( $\frac{1}{2}$ P).

1P

f)

2P

- Write-Hit - M
- Shared Read-Miss m. E. - S
- Snoop-Hit on a Write - I
- Snoop-Hit on a Read - S

g) ( $\frac{1}{2}$ P Abzug pro Fehler)

4P

| Prozessor | Aktion | Prozessor 1 |        | Prozessor 2 |        | Prozessor 3 |        |
|-----------|--------|-------------|--------|-------------|--------|-------------|--------|
|           |        | Line 1      | Line 2 | Line 1      | Line 2 | Line 1      | Line 2 |
|           | init   | -           | -      | -           | -      | -           | -      |
| 1         | rd 1   | 1/E         |        |             |        |             |        |
| 3         | rd 2   |             |        |             |        | 2/E         |        |
| 3         | rd 4   |             |        |             |        |             | 4/E    |
| 2         | rd 2   |             |        | 2/S         |        | 2/S         |        |
| 1         | wr 3   |             | 3/M    |             |        |             |        |
| 2         | rd 3   |             | 3/S    |             | 3/S    |             |        |
| 3         | wr 2   |             |        | 2/I         |        | 2/M         |        |
| 2         | wr 2   |             |        | 2/M         |        | 2/I         |        |
| 3         | rd 1   | 1/S         |        |             |        | 1/S         |        |
| 1         | rd 1   | 1/S         |        |             |        | 1/S         |        |
| 2         | rd 4   |             |        |             | 4/S    |             | 4/S    |
| 1         | wr 2   |             | 2/M    | 2/I         |        |             |        |

## Musterlösung 6: Fehlertoleranz

10P

- a) Temporäre und permanente Fehler (je  $\frac{1}{2}$ P) 1P
- b) 3P
- Fail-stop-System: Ausfälle sind nur Anhalteausfälle
  - Fail-silent-System: Ausfälle sind nur Unterlassungsausfälle
  - Fail-safe-System: Ausfälle sind nur unkritische Ausfälle
- c) Zuverlässigkeitsblockdiagramm: 1P
- ```

+---B1---+   +---T1---+
---+         +---+         +---S---H---
+---B2---+   +---T2---+

```
- d) Systemfunktion:  $S = (B_1 \vee B_2) \wedge (T_1 \vee T_2) \wedge S \wedge H$  1P
- e) Funktionswahrscheinlichkeit 1P
- $$\varphi(SF) = (1 - (1 - \varphi(B))^2) * (1 - (1 - \varphi(T))^2) * \varphi(S) * \varphi(H)$$
- f) Gegenseitige Redundanz ( $\frac{1}{2}$ P), ermöglicht abgestuften Leistungsabfall bzw. Graceful Degredation ( $\frac{1}{2}$ P). 1P
- g) Zuverlässigkeitsblockdiagramm 1P
- ```

+---T1---T2---+
+               +
--+---T1---T3---+
+               +
+---T2---T3---+

```
- h) Funktionswahrscheinlichkeit für ein n-aus-m System: 1P
- $$\varphi_m^n = \sum_{k=n}^m \binom{m}{k} * \varphi(K)^k * (1 - \varphi(K))^{(m-k)}$$