



Institut für Technische Informatik

Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung

Prof. Dr. rer. nat. Wolfgang Karl

Klausur Rechnerstrukturen Sommersemester 2014 Musterlösung

Aushang der Ergebnisse: ab Anfang/Mitte September 2014

Musterlösung 1: Fertigungskosten, HW-Entwurf und Fehlertoleranz

12 P

Fertigungskosten

3 P

a) $dpw = A - B = \frac{A_{wafer}}{A_{die}} - B = \frac{\pi * (d_{wafer} * \frac{1}{2})^2}{a_{die}} - \frac{\pi * d_{wafer}}{\sqrt{2 * a_{die}}}$ 1 P
wobei A das theoretische Maximum und B der Verschnitt ist.

b) Die erzielbare Anzahl von Dies pro Wafer (dpw) wird überproportional ansteigen. 1 P
Dies liegt daran, dass der Beitrag der Wafer-Größe quadratisch in die dpw -Berechnung eingeht, der Verschnitt jedoch nur linear.

c) Kosten des Dies, Kosten für das Testen des Dies, Kosten für das Packaging und den abschließenden Test. 1 P

$$\text{Formel: } cost_{IC} = \frac{cost_{die} + cost_{ie-test} + cost_{packaging}}{Y_{final}}$$

Hardwareentwurf

3 P

d) • Entity: Schnittstelle des Moduls mit Definition der Ein- und Ausgänge. 2 P

• Architecture: Beschreibung des Verhaltens oder der Struktur

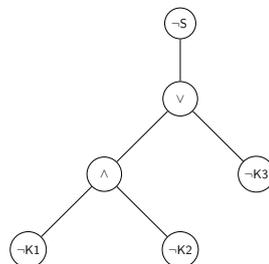
e) $c \leq a \text{ XOR } b;$ 1 P

Fehlertoleranz

6 P

f) Systemfunktion: $S = (K1 \vee K2) \wedge K3$ 0,5 P

g) Fehlerbaum: $S = (K1 \vee K2) \wedge K3$ d.h. $\neg S = (\neg K1 \wedge \neg K2) \vee \neg K3$ 1 P



h) • Fail-silent-System \Leftrightarrow Unterlassungsausfall 1,5 P

• Fail-stop-System \Leftrightarrow Anhalteausfall

• Fail-safe-System \Leftrightarrow unkritische Ausfälle zuordnen

i) Der Knoten sendet entweder korrekte Nachrichten oder gar keine Nachrichten. Er sendet jedoch nie falsche Nachrichten. Bei einem Fail-Stop-System werden aufgrund eines aufgetretenen Fehlers gar keine Nachrichten mehr versandt. 1 P

j) Bezeichnet das Vorhandensein von redundanten Mitteln, die während des gesamten Einsatzzeitraums die gleiche Nutzfunktion erbringen. Beispiel: n-von-m-System 1 P

k) Bezeichnet das Vorhandensein von redundanten Mitteln, die erst nach Auftreten eines Fehlers aktiviert werden, um eine ausgefallene Nutzfunktion zu erbringen. 1 P

Musterlösung 2: Parallelismus auf Befehlsebene und Sprungvorhersage

10 P

Parallelismus auf Befehlsebene

5 P

- a) 1 P
- Bearbeitung eines Objekts wird in Teilschritte zerlegt
 - Abarbeitung in einer sequentiellen Folge (Phasen der Pipeline)
 - Unterschiedliche Phasen der Pipeline können verschiedene Objekte überlappend abarbeiten
- b) $S = n * k / (k + n - 1)$ 1 P
- c) Datenkonflikt: Eine Instruktion benötigt das Ergebnis einer vorangehenden und noch nicht abgeschlossenen Instruktion in der Pipeline. 1 P
- d) Ein Datenkonflikt tritt nur dann auf, wenn ein Operand in der Pipeline (noch) nicht verfügbar ist. Ist der Abstand zwischen zwei Befehlen, zwischen denen ein Datenabhängigkeit besteht, groß genug, so wirkt sich diese Abhängigkeit nicht in der Ausführung aus. 1 P
- e) Durch Out-of-Order-Ausführung kann sich der Abstand oder sogar die Reihenfolge der Befehle ändern. Es können also durch die superskalare Ausführung wieder Datenkonflikte auftreten. 1 P

Sprungvorhersage

5 P

- f) 4 P

Befehl	Inhalt		Prädiktor loop1			Prädiktor loop2		
	r0	r1	alt	Vorhersage	neu	alt	Vorhersage	neu
Init	1	3	-	-	SNT	-	-	SNT
BNEQZ r1, loop2	1	2	-	-	-	SNT	falsch	WNT
BNEQZ r1, loop2	1	1	-	-	-	WNT	falsch	ST
BNEQZ r1, loop2	1	0	-	-	-	ST	falsch	WT
BEQZ r0, loop1	0	0	SNT	falsch	WNT	-	-	-
BNEQZ r1, loop2	0	2	-	-	-	WT	richtig	ST
BNEQZ r1, loop2	0	1	-	-	-	ST	richtig	ST
BNEQZ r1, loop2	0	0	-	-	-	ST	falsch	WT
BEQZ r0, loop1	-1	0	WNT	richtig	SNT	-	-	-

- g) Nein! Dann wäre zwar der erste Sprung korrekt vorher gesagt worden, der zweite jedoch nicht! 1 P

Musterlösung 3: Parallelverarbeitung und Architekturen

10 P

Quantitative Maßzahlen

5 P

- a) Amdahls Gesetz (Je $\frac{1}{2}$ P für Formel und Erklärung):

1 P

$$T(n) = \underbrace{\frac{1}{n} * T(1) * (1 - a)}_1 + \underbrace{T(1) * a}_2$$

a mit $0 < a < 1$ ist der Anteil eines Programms, der nur sequentiell ausgeführt werden kann. Die Formel zerfällt damit in die Ausführungszeiten des parallel ausführbaren Programmteils 1 und des rein sequentiell ausführbaren Programmteils 2.

- b) $S(n) = \frac{T(1)}{T(n)} = \dots \leq \frac{1}{a}$ für $n \rightarrow \infty$

1 P

Die maximal erreichbare Beschleunigung wird durch den Anteil a des Programms begrenzt, der nur sequentiell ausgeführt werden kann.

- c) Die absolute Beschleunigung und absolute Effizienz erhält man, indem der beste sequentielle Algorithmus mit dem besten parallelen Algorithmus verglichen wird.

1 P

- d) Der Zusatzaufwand für die Parallelisierung durch Kommunikation und Synchronisation „verfälscht“ das Ergebnis bei der sequentiellen Ausführung.

1 P

- e) Das Verhalten heißt superlineare Beschleunigung bzw. superlinearer Speedup und widerspricht der Abschätzung $1 \leq S(n) \leq n$.

1 P

Parallele Architekturen

5 P

- f) Vier Klassen von Rechnerarchitekturen:

1 P

- SISD: Single Instruction - Single Data
- SIMD: Single Instruction - Multiple Data
- MISD: Multiple Instruction - Single Data
- MIMD: Multiple Instruction - Multiple Data

- g) • Nachrichtenorientierte Multiprozessoren: MIMD, unterschiedliche Prozessoren mit jeweils eigenständigen Programmen und Daten

1 P

- Vektorrechner: SIMD, Einzelne Befehle auf einem Datenvektor

- h) Multiprozessor mit:

1,5 P

verteiltmem Speicher	NORMA	No Remote Memory Access
gemeinsamem Speicher	UMA	Uniform Memory Access
verteiltmem gemeinsamen Speicher	NUMA	Non-Uniform Memory Access

- i) Herausforderungen: Verlustleistung (Energie), Hauptspeicher (Kapazität, Zugriffsgeschwindigkeit, Skalierbarkeit, Mithalten mit Rechengeschwindigkeit), Zuverlässigkeit/Verfügbarkeit, Parallelität und Lokalität

1,5 P

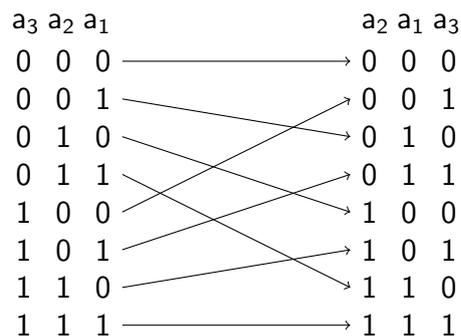
Musterlösung 4: Verbindungsstrukturen und Vektorverarbeitung

8 P

Verbindungsstrukturen

5 P

- a) • Fähigkeit, die wesentlichen Eigenschaften des Verbindungsnetzes auch bei beliebiger Erhöhung der Knotenzahl beizubehalten. 1 P
- Vergrößerung möglich ohne die wesentlichen Eigenschaften des Netzwerks zu verlieren.
- b) • Erweiterbarkeit sehr schlecht! 1 P
- Bei jeder Erweiterung muss die Anzahl an Knoten verdoppelt werden
- Der Knotengrad aller Knoten steigt mit jeder Erweiterung. Alle schon existierenden Knoten müssen somit erweitert werden.
- c) Bei einem fehlertoleranten Netz muss zwischen jedem Paar von Knoten (0,5P) mindestens ein weiterer, redundanter Weg vorhanden sein (0,5P) 1 P
- d) Mischpermutation: Kreisverschiebung der Adressbits 1 P



- e) Bei „Big Data“ wird eine große Menge unterschiedlicher Daten aus verschiedenen, oft sehr großen Datenmengen verarbeitet und miteinander verknüpft. Diese Daten liegen dabei oft nicht in zusammenhängenden Speicherbereichen vor. Deshalb ist hierfür eine sehr große Anzahl an unterschiedlichen Speicherzugriffen notwendig. Eine geringe Latenz ist speziell beim Austausch vieler kleinerer Datenpakete wichtig. 1 P

Vektorverarbeitung

3 P

- f) • Verkettung mehrerer (spezialisierter) Pipelines 1 P
- Die Ergebnisse einer Pipeline werden sofort der nächsten Pipeline zur Verfügung gestellt.
- g) • Vektor-Pipeline-Parallelität 2 P
- Mehrere Vektor-Pipelines in einer Vektoreinheit
- Vervielfachung der Pipelines
- Mehrere Vektoreinheiten

Musterlösung 5: Low-Power-Entwurf & Rechnerbewertung

10 P

Low-Power-Entwurf

4 P

a) $g = \neg A \wedge \neg B$

1 P

b) Funktion f :

1 P

$$\mathbb{P}_{A \vee B}(1) = 1 - \mathbb{P}_{A \vee B}(0) = 1 - \left(\frac{2}{3} * \frac{1}{2}\right) = 1 - \frac{1}{3} = \frac{2}{3}$$

$$\mathbb{P}_{\neg(A \vee B)}(1) = 1 - \mathbb{P}_{A \vee B}(1) = 1 - \frac{2}{3} = \frac{1}{3}$$

Funktion g :

$$\mathbb{P}_{\neg A}(1) = \frac{2}{3}$$

$$\mathbb{P}_{\neg B}(1) = \frac{1}{2}$$

$$\mathbb{P}_{\neg A \wedge \neg B}(1) = \frac{2}{3} * \frac{1}{2} = \frac{1}{3}$$

c) Funktion f :

2 P

$$\mathbb{P}_{A \vee B} = 2 * \frac{2}{3} * \frac{1}{3} = \frac{4}{9}$$

$$\mathbb{P}_{\neg(A \vee B)} = 2 * \frac{1}{3} * \frac{2}{3} = \frac{4}{9}$$

Funktion g :

$$\mathbb{P}_{\neg A} = 2 * \frac{2}{3} * \frac{1}{3} = \frac{4}{9}$$

$$\mathbb{P}_{\neg B} = 2 * \frac{1}{2} * \frac{1}{2} = \frac{1}{2}$$

$$\mathbb{P}_{\neg A \wedge \neg B} = 2 * \frac{1}{3} * \frac{2}{3} = \frac{4}{9}$$

Leistungsbewertung

6 P

d) Anzahl der Instruktionen:

3 P P

$$IC = 750 * 10^3 + 500 * 10^3 + 150 * 10^3 + 200 * 10^3 = 1,6 * 10^6$$

Anzahl der Zyklen mit ursprünglicher Funktionseinheit:

$$C_{alt} = 750 * 10^3 + 500 * 10^3 * 4 + 150 * 10^3 * 5 + 200 * 10^3 * 10 = 0,75 * 10^6 + 2 * 10^6 + 0,75 * 10^6 + 2 * 10^6 = 5,5 * 10^6$$

Anzahl der Zyklen mit neuer Funktionseinheit:

$$C_{neu} = 5,5 * 10^6 - 500 * 10^3 * 2 = 4,5 * 10^6$$

Ursprüngliche Ausführungszeit:

$$T_{exe,alt} = 5,5 * 10^6 * \frac{1}{4 * 10^9 * s^{-1}} = \frac{11}{2 * 4 * 10^3} s = \frac{11}{8} ms$$

Neue Ausführungszeit::

$$T_{exe,neu} = 4,5 * 10^6 * \frac{1}{4 * 10^9 * s^{-1}} = \frac{9}{8} ms$$

Speed Up:

$$S = \frac{11/8}{9/8} = \frac{9 * 8}{8 * 7} = \frac{11}{9}$$

MIPS mit ursprünglicher Funktionseinheit:

$$MIPS_{alt} = \frac{1,6 * 10^6}{\frac{11}{8} * 10^{-3} * 10^6} = \frac{16 * 8 * 10^3}{11 * 10} = \frac{12800}{11}$$

MIPS mit neuer Funktionseinheit:

$$MIPS_{neu} = \frac{1,6 \cdot 10^6}{\frac{9}{8} \cdot 10^{-3} \cdot 10^6} = \frac{16 \cdot 8 \cdot 10^3}{9 \cdot 10} = \frac{12800}{9}$$

- e) Die SPECrate quantifiziert den Durchsatz, die SPECratio die Geschwindigkeit eines Rechensystems. *1 P*
- f) Es wird das geometrische Mittel verwendet, da das geometrische Mittel im Gegensatz zum arithmetischen Mittel konsistent, unabhängig von der Referenzmaschine ist. *1 P*
- g) $SPECratio_x = \frac{Referenzzeit_x}{Laufzeitzeit_x \text{ auf Testsystem}}$ *1 P*

Musterlösung 6: Speicherhierarchie

11 P

Cache-Kohärenzprotokoll MESI

7 P

a)

4 P

Prozessor	Aktion	Prozessor 1		Prozessor 2	
		Zeile 1	Zeile 2	Zeile 1	Zeile 2
	init	-	-	-	-
1	rd 3	3/E			
2	rd 1			1/E	
2	rd 3	3/S			3/S
1	rd 4		4/E		
1	wr 3	3/M			3/I
1	rd 5		5/E		
2	wr 5		5/I		5/M
1	rd 5		5/S		5/S
1	rd 1	1/S		1/S	
2	wr 2				2/M

- b) Um die Speicherkonsistenz zu erhalten wird der Exclusive-Zustand des MESI-Protokolls nicht benötigt. Dieser gibt lediglich an, dass eine Cache-Zeile ohne das Aussenden des Invalidate-Signals geschrieben werden kann. Wird also anstelle des Exclusive-Zustands der Shared-Zustand verwendet, muss bei jeder Schreiboperation das Invalidate-Signal gesendet werden. Die Konsistenz ist aber dadurch trotzdem sichergestellt. 2 P
- c) Für die Darstellung von nur drei Zuständen werden wie bei der Darstellung von vier Zuständen jeweils zwei zusätzliche Bits benötigt. Der vierte Zustand, in diesem Fall der Exclusive-Zustand, verursacht also keinen zusätzlichen Overhead, bringt aber den Vorteil mit sich, eine Cache-Zeile schreiben zu können, ohne ein Invalidate-Signal versenden zu müssen. 1 P

Cache-Leistung

4 P

- d) Variante A: $0,75 * 8 ns + 0,25 * (0,8 * 20 ns + 0,2 * 100 ns) = 6 ns + 0,25 * (16 ns + 20 ns) = 6 ns + 0,25 * 36 ns = 6 ns + 9 ns = 15 ns$ 3 P
- Variante B: $0,75 * 6 ns + 0,25 * (0,8 * 30 ns + 0,2 * 130 ns) = 4,5 ns + 0,25 * (24 ns + 26 ns) = 4,5 ns + 0,25 * 50 ns = 4,5 ns + 12,5 ns = 17 ns$
- Variante C: $0,9 * 6 ns + 0,1 * 100 ns = 5,4 ns + 10 ns = 15,4 ns$

Somit sind nur die Varianten A und C für einen Einsatz im System geeignet.

- e) Die 90/10 Regel besagt, dass bei 90% der Speicherzugriffe innerhalb eines Programms nur etwa 10% der Daten benötigt werden, die das Programm insgesamt während seiner Laufzeit benötigt. Somit ist also ein deutlich kleinerer Speicher ausreichend, um die meisten Anfragen beantworten zu können. 1 P