



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

Klausur Rechnerstrukturen
Wintersemester 2012/13 – 5. März 2013
Musterlösung

Aushang der Ergebnisse: ab Mitte April 2013

Musterlösung 1: Parallelverarbeitung und Architekturen

10P

- a) Amdahls Gesetz (Je 0,5P für Formel und Erklärung):

1P

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

Die Formel zerfällt in die Ausführungszeiten des parallel ausführbaren Programmteils 1 und des rein sequentiell ausführbaren Programmteils 2.

- b) • $S(n) = \frac{T(1)}{T(n)}$
 • $E(n) = \frac{S(n)}{n}$

1P

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

1P

$$S(n) = \frac{1}{a} \text{ für } n \rightarrow \infty$$

Hier ist $a = \frac{1}{64}$ und damit $S_{max}(n) = 64$

- d) Bei fester Problemgröße und steigender Anzahl der Prozessoren, wird ab einer bestimmten Anzahl an Prozessoren eine Sättigung eintreten.

1P

- e) • Superlinearer Speedup: $S > n, E > 1$
 • Beispiele:
 – paralleles Backtracking (depth-first-search)
 – Cache- und Hauptspeichernutzung

1P

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

1P

- g) 1. Dekomposition
 2. Zuweisung
 3. Festlegung
 4. Abbildung

2P

Die Schritte 1-3 werden auch als Partitionierung zusammengefasst.

- h) • OpenMP: Shared-Memory-Programmiermodell
 • MPI: Nachrichtenorientiertes Programmiermodell

1P

- i) Projektion auf Exascale mit heutiger Technologie im Bereich von mehreren GW. Verlustleistung für Systeme im ExaFlops-Bereich kann höchstens im Bereich 20-40 MW liegen

1P

Musterlösung 2: Low-Power-Entwurf & Rechnerbewertung

10P

Low-Power-Entwurf

3P

a) Signalwahrscheinlichkeiten:

1P

Variante 1:

$$\mathbb{P}_{\neg A \text{ Ausgang}=1} = 1 - \frac{1}{4} = \frac{3}{4}$$

$$\mathbb{P}_{\neg A \vee B \text{ Ausgang}=1} = 1 - \left(1 - \frac{3}{4}\right) * \left(1 - \frac{2}{3}\right) = 1 - \frac{1}{4} * \frac{1}{3} = \frac{11}{12}$$

Variante 2:

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

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

b) Schaltwahrscheinlichkeiten:

1P

Für beide Schaltungsvarianten ergibt sich mit der bekannten Formel:

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

$$\mathbb{P}_{Schalt 1} = 2 * \frac{11}{12} * \frac{1}{12} = \frac{22}{144} = \frac{11}{72}$$

$$\mathbb{P}_{Schalt 2} = 2 * \frac{1}{3} * \frac{2}{3} = \frac{4}{9} = \frac{32}{72}$$

c) Bewertung:

1P

Sie empfehlen Variante 1 aufgrund der geringeren Schaltwahrscheinlichkeit ($\mathbb{P}_{Schalt 1} = \frac{11}{72} < \frac{32}{72} = \mathbb{P}_{Schalt 2}$), da somit eine geringere Verlustleistung durch die Schaltvorgänge zu erwarten ist.

Leistungsbewertung

3P

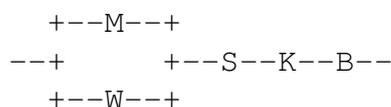
- d) Falsch: eine geringere Zykluszeit bedeutet eine höhere Frequenz und mit $P \sim f$ eine höhere Leistungsaufnahme 1P
- e) Falsch: Ein Übertakten des Prozessors führt dazu, dass pro Zeiteinheit auch mehr Instruktionen ausgeführt werden und somit der CPI Wert gleich bleibt. 1P
- f) Richtig: da die Rechnerorganisation einen Einfluss auf die Länge des kritischen Pfades hat, beeinflusst diese auch die mögliche Taktfrequenz. 1P

Fehlertoleranz

4P

g) Zuverlässigkeitsblockdiagramm:

1P



h) Strukturformel/Systemfunktion:

1P

$$F = (M \vee W) \wedge S \wedge K \wedge B$$

i) Funktionswahrscheinlichkeit:

1P

$$\varphi(F) = (1 - (1 - \varphi(M))(1 - \varphi(W))) * \varphi(S) * \varphi(K) * \varphi(B)$$

j) Klassifikation: Fail-safe-System, da der Ausfall der Brennstoffzelle unkritisch ist und als einziger Ausfall zugelassen wurde. 1P

Musterlösung 3: Speicherhierarchie

10P

Cache-Kohärenzprotokoll MESI

6P

a)

Zeile	Prozessor	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
			Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
0.		init	-	-	-	-	-	-
1.	1	wr 1	1/M					
2.	2	rd 2			2/E			
3.	3	wr 1	1/I				1/M	
4.	1	rd 3	3/E					
5.	3	rd 3	3/S					3/S
6.	2	rd 1				1/S	1/S	
7.	3	rd 2			2/S			2/S
8.	2	wr 4				4/M		
9.	3	rd 1					1/S	
10.	2	rd 1			1/S			

4P

- b) Das MOESI-Protokoll würde zu einer beschleunigten Abarbeitung führen: in Zeile 6 werden würden die Daten direkt durch einen Cache-zu-Cache-Transfer übertragen. Dies würde zwei Speicherzugriffe sparen (1 lesend/1 schreibend). 1P
- c) In diesem Beispiel gleichen sich die Vorteile durch den Einfluss der LRU-Strategie auf MESI-Zustandsänderungen aus: während die Ersetzung in Zeile 7 von Prozessor 3 einen Hit in Zeile 9 bedingt, verhindert die Ersetzung der Zeile 8 in Prozessor 2 einen Hit in Zeile 10. Somit wiegen sich positive wie negative Effekte dieser Berücksichtigung auf. 1P

Cache-Leistung

4P

- d) Alternative A: $0,8 * 6 ns + 0,2 * (0,8 * 20 ns + 0,2 * 100 ns) = 4,8 ns + 0,2 * (16 ns + 20 ns) = 4,8 ns + 0,2 * 36 ns = 4,8 ns + 7,2 ns = 12 ns$, 3P
- Alternative B: $0,8 * 4 ns + 0,2 * (23 ns * 0,75 + 0,25 * 123 ns) = 3,2 ns + 0,2 * (17,25 ns + 30,75 ns) = 3,2 ns + 0,2 * 48 ns = 3,2 ns + 9,6 ns = 12,8 ns$
- Damit erreicht Alternative B die langsamere Antwortzeit, weshalb ein Wechsel der Cachehierarchie nicht zu empfehlen ist, da dies eine Verschlechterung bedeuten würde.
- e) Für eine Multiprozessorsystem mit gemeinsamen Speicher lohnt es sich diese Bewertung zu überdenken, da bei der Variante A im Fall eines Cache-Hits in den höheren Ebenen Anfragen an den Hauptspeicher gestartet werden, die dann später wieder abgebrochen werden, blockieren diese den Bus und verhindern so, dass notwendige Daten zu anderen Prozessoren transferiert werden. Somit kann man argumentieren, dass die längere Antwortzeit der Variante B toleriert werden kann, da hier keine unnötigen Anfragen an den L2 Cache und den Hauptspeicher auftreten. 1P

Musterlösung 4: Fertigung und Hardwareentwurf

10P

Fertigungskosten

5P

a) • Formel: $cost_{die} = \frac{cost_{wafer}}{dpw * yield_{die}}$

0.5P

- Rechnung und Antwort:

2P

$$cost_{die, rechteck} = cost_{die, kreis}$$

$$\frac{cost_{w,r}}{dpw_r * yield_{die,r}} = \frac{cost_{w,k}}{dpw_k * yield_{die,k}}$$

$$cost_{w,r} = \frac{cost_{w,k} * dpw_r * yield_{die,r}}{dpw_k * yield_{die,k}}$$

$$cost_{w,r} = \frac{1000 * 600 * 0,4}{400 * 0,5} = \frac{600}{0,5} = 1200$$

→ Der rechteckige Wafer darf maximal 1200 Euro kosten.

- Begründung: Das neue Verfahren, da es trotz kleinerer Ausbeute Dies mit geringeren Herstellungskosten produziert. 0.5P

- b) • Erklärung: Auf kreisförmigen Wafern kann die Grundfläche nicht vollständig genutzt werden, da durch die Kreisform die Fläche am Rand des Wafers nicht genutzt werden kann. 0.5P

- Formel für kreisförmige Wafer: $dpw = A - B = \frac{\pi * (d_{wafer}/2)^2}{a_{die}} - \frac{\pi * d_{wafer}}{\sqrt{2} * a_{die}}$ 1P

A: theoretisches Maximum

B: Verschnitt

(0.5P Abzug für Fehler oder fehlende Beschreibung)

- Antwort: Nein, auch bei rechteckigen Wafern kann es zu Verschnitt kommen, wenn die Seitenlänge nicht ein Vielfaches der Die-Seitenlänge ist. 0.5P

Schaltungsentwurf in VHDL

5P

- c) • Schnittstelle (Entity)

1,5P

- Verhaltens- oder Strukturrealisierung (Architecture)
- Zuordnung (Configuration)

- d) • Bekanntmachung der Komponenten (component declaration)

1,5P

- Erzeugung von Kopien der Komponente (component instantiation)
- event. Konfiguration (component configuration)

- e) Lösung mit Bibliotheksaufruf:

1P

```
c <= a nand b;
```

f) Ergänzung um die Instantiierung:

IP

```
And_Gate1 : And_Gate PORT MAP (X1, X2, S1);  
Inverter1 : Inverter PORT MAP (S1, Y);
```

Vollständiger Quelltext:

```
1 ARCHITECTURE Structure of Nand2 IS  
2   COMPONENT Inverter  
3     PORT (  
4       In1 : IN Std_Logic;  
5       Out1 : OUT Std_Logic  
6     );  
7   END COMPONENT  
8   COMPONENT And_Gate  
9     PORT (  
10      In1, In2: IN Std_Logic;  
11      Out1 : OUT Std_Logic  
12    );  
13   END COMPONENT  
14   SIGNAL S1: Std_Logic;  
15 BEGIN  
16   And_Gate1 : And_Gate PORT MAP (X1, X2, S1);  
17   Inverter1 : Inverter PORT MAP (S1, Y);  
18 END Structure;
```

Musterlösung 5: Verbindungsstrukturen & Vektorrechner

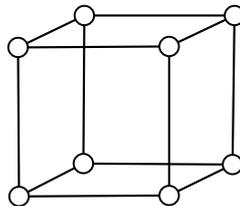
10P

Verbindungsstrukturen

5P

- a) • Fähigkeit, die wesentlichen Eigenschaften des Verbindungsnetzes auch bei beliebiger Erhöhung der Knotenzahl beizubehalten. *1P*
- Vergrößerung möglich ohne die wesentlichen Eigenschaften des Netzwerks zu verlieren.

- b) Radius $K = 2$, Dimension $n = 3$ *1P*



- c) • Knotenzahl: $N = K^n = 3^3 = 27$ *1P*

- Knotengrad $2 * n = 2 * 3 = 6$

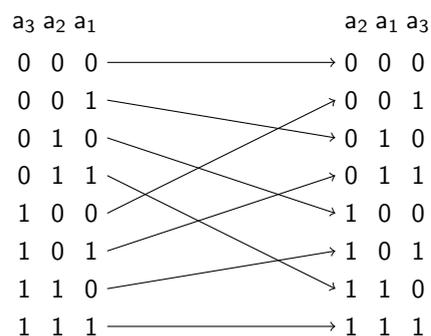
- d) • Erweiterbarkeit schwierig/nicht gut! *1P*

- Bei einer Erweiterung durch Vergrößern des Radius K steigt die Knotenanzahl stark an, der Knotengrad bleibt gleich.

- Bei einer Erweiterung durch Erhöhen der Dimension n wird der Knotengrad aller Knoten mit jeder Erweiterung erhöht. Alle schon existierenden Knoten müssten somit erweitert werden.

⇒ Ein 3D-Torus mit unterschiedlichen Kantenlängen wäre besser erweiterbar.

- e) Mischpermutation: *1P*



Vektorverarbeitung:**5P**

f)

3P

```

MOV  R1, 64          # R1 mit 64 initialisieren
MTC1 VLR, R1        # vector-length register := 64
LV   V1, Ra         # int a[n] in V1 laden
LV   V2, Rb         # int b[n] in V2 laden
LV   V3, Ra         # V3 enthaelt c[n], welches mit a[n] initialisiert wird
MOV  R1, 0xff       # R1 mit 0xff initialisieren
SEQV.I V1, V2       # Vergleich der beiden Vektoren a[n] und b[n]
SUBV.I V3, V3, V3   # Komponenten von V3 mit 1 im VMR werden auf 0 gesetzt
ADDVS.I V3, V3, R1  # diese Komponenten werden auf 0xff gesetzt
CVM                                     # Clear Vektor Mask -- VMR auf 1 setzen
SV   Rc, V3        # Schreiben von c[n]

```

g) Es handelt sich um eine abgewandelte Form des Blue-Box-Verfahrens, wenn die Pixel in Vektor $a[n]$ mit denen des Vektors $b[n]$ übereinstimmen, werden diese durch Pixel einer bestimmten Farbe (0xff) ersetzt. *1P*

h)

1P

- Stride von 1:

Latenz von 12 Zyklen für erstes Element und 63 Zyklen für alle weiteren zu holenden Elemente, da bei 16 Speicherbänken jeder nächste Zugriff auf die jeweils nächste Bank geht und somit die Latenz von 12 Takten versteckt werden kann. Folglich benötigt man 75 Zyklen.

- Stride von 8:

Da 8 die Anzahl der Bänke (16) teilt, geht jeder zweite Zugriff auf die gleiche Memory Bank und kollidiert mit dem vorhergehenden Zugriff. Somit lassen sich nur 2 Anfragen überlappt ausführen. Damit benötigt jeder zweite Speicherzugriff die Latenz von 12 Takten während sich immer ein Zugriff überlappt ausführen lässt und man benötigt insgesamt: $12 * 32 + 1 = 385$ Takte

Musterlösung 6: Rechnerarchitektur

10P

Sprungvorhersage

3P

a) Tabelle:

3P

Sprung	Vorhersage	Sprungausgang	Aktualisierter Prädiktor
1	SNT	NT	SNT
2	SNT	T	WNT
1	SNT	T	WNT
2	WNT	NT	SNT
1	WNT	T	ST
2	SNT	T	WNT

(0.5P pro richtige Zeile)

Parallelismus auf Befehlsebene

7P

b) Superskalartechnik: Parallelisierung durch Hardware (0.5P)

2P

- geringere Komplexität des Compilers
- i.A. bessere Hardware-Auslastung

VLIW: Parallelisierung durch Compiler (0.5P)

- geringere Komplexität der Hardware
- Performance besser vorhersagbar

Jeweils für einen Vorteil: 0.5P

c)

5P

Feld	R1	R2	R3	R4
Value				(R4)
Valid	0	0	0	1
RS	Div 1	Mul 1	Int 2	

Unit	Empty	InFU	Op	Dest	Src1	Valid1	RS1	Src2	Valid2	RS2
Int 1	1									
Int 2	0	0	add	R3	(R3)	1		(R1)	1	
Mul 1	0	1	mul	R2	(R1)	1		(R4)	1	
Div 1	0	0	div	R1		0	Mul 1		0	Int 2

(0.5P Abzug pro Fehler)