

Universität Karlsruhe
Institut für Technische Informatik
Prof. Dr. Wolfgang Karl

Klausur Rechnerstrukturen
Sommersemester 2006
Musterlösung

Aushang der Ergebnisse: ab Mitte/Ende September 2006

Musterlösung 1: Quantifizierung

15P

a) $(\frac{1}{2}P) \text{ cost}_{ic} = \frac{\text{cost}_{die} + \text{cost}_{pkg} + \text{cost}_{test}}{\text{yield}_{final}}$ 1P

Da die Verbesserung des Preises keine fertigungstechnischen Ursachen hat (gleiche Fertigungstechnologie, gleiches Packaging, gleiche Testprozedur), kann die Ursache somit nur die Gesamtausbeute sein ($\frac{1}{2}P$).

- b) Offenbar wurden das Packaging optimiert, welches zuvor zu einer Erhöhung der Ausschussquote beigetragen haben. 1P

Beispiele für Schäden in den nachfolgenden Schritten sind Zerstörung durch statische Elektrizität durch mangelnde ESD-Sicherheit oder Beschädigung beim Packaging.

c) $(\frac{1}{2}P) \text{ Formel: } \text{cost}_{die} = \frac{\text{cost}_{wafer}}{dpw * \text{yield}_{die}}$ 1P

$(\frac{1}{2}P)$ Da die Wafer-Kosten identisch geblieben sind, kann sich nur die Ausbeute verschlechtern haben.

- d) Es gilt wegen gleicher Die- und Wafer-Herstellungskosten: 1P

- $(\frac{1}{2}P) cd = \frac{cw}{1 * dpw * yd_1} = \frac{cw}{2.5 * dpw * yd_2}$
- $(\frac{1}{2}P) \rightarrow \frac{yd_2}{\frac{1}{2}} = yd_1 \rightarrow yd_2 = \frac{2}{5} yd_1$

Der Die-Yield ist mit den größeren Wafern auf $\frac{2}{5} = 40\%$ des ursprünglichen Wertes eingebrochen.

- e) $(\frac{1}{2}P) \text{ yield}_{die} = \text{yield}_{wafer} * (1 + \frac{dpua * a_{die}}{\alpha})^{-\alpha}$ $(\frac{1}{2}P)$ Die Anzahl von Defekten pro Unit-Area hat sich erhöht. 1P

- f) (je $\frac{1}{2}P$ für Formel und Berechnung) 5P

- Anzahl Instruktionen: $i = \sum i_{typ}$
 $= (300 + 100 + 150 + 50) * 10^3 = 600.000$
- Taktzyklen: $c = \sum i_{typ} * c_{typ}$
 $= (300 * 1 + 100 * 1 + 150 * 5 + 50 * 12) * 10^3 = 1.750.000$
- Zykluszeit: $t_{cyc} = \frac{t_{exec}}{c}$
 $= \frac{500 * 10^{-6}}{1750 * 10^3} = \frac{500}{1750} * 10^{-9} = \frac{2}{7} * 10^{-9} = \frac{1}{3,5} * 10^{-9}$
- Taktfrequenz: $f = \frac{1}{t_{cyc}} = \frac{7}{2} * 10^9 = 3,5 * 10^9$
- MIPS = $\frac{i}{t * 10^6} = \frac{600.000}{500} = 1200$

-
- g) Der Zeitaufwand für Kontrollflusstransfer ist sehr hoch ($\frac{1}{2}$ P). Da Allzweck-Anwendungen hiervon dominiert werden, ist der Einsatz der Architektur hierfür nicht sinnvoll ($\frac{1}{2}$ P). *1P*
- h) ($\frac{1}{2}$ P) $f = \frac{i \cdot CPI}{t} \rightarrow CPI = \frac{f \cdot t}{i}$ *1P*
($\frac{1}{2}$ P) $CPI = \frac{2,5 \cdot 10^9 \cdot 5 \cdot 10^{-3}}{1 \cdot 10^6} = \frac{2,5 \cdot 5}{1} = 12,5$
- i) ($\frac{1}{2}$ P) $MIPS = \frac{f}{CPI \cdot 10^6} = \frac{f \cdot i}{f \cdot t \cdot 10^6} = \frac{i}{t \cdot 10^6}$ *1P*
($\frac{1}{2}$ P) $MIPS = \frac{i}{t \cdot 10^6} = \frac{1 \cdot 10^6}{5 \cdot 10^{-3} \cdot 10^6} = \frac{1}{5} \cdot 10^3 = 200$
- j) Konservative Optimierung ($SPEC_{base}$, fairer Systemvergleich), aggressive Optimierung (Maximalleistung) *1P*
- k) Betriebssystem und Compiler. *1P*

Musterlösung 2: Hardwareentwurf

10P

- a) ($\frac{1}{2}$ P) Fall 1: Aufgrund der Zuweisung mittels \leq ist a ein Signal. 1P
 ($\frac{1}{2}$ P) Fall 2: Aufgrund der Zuweisung mittels $:=$ ist a eine Variable.
- b) Voraussetzung: $a = 4$, Zuweisung in einem getakteten Prozess 2P
 Variante 1: $x = 4 + 1 = 5$ ($\frac{1}{2}$ P), weil $a \leq a+1$ erst mit dem Prozess-Ende gültig gemacht wird und bis dahin der alte Wert gilt ($\frac{1}{2}$ P).
 Variante 2: $x = (4+1)+1 = 5+1 = 6$ ($\frac{1}{2}$ P), weil die Zuweisung $a := a+1$ unmittelbar erfolgt ($\frac{1}{2}$ P).
- c) Es ist kein Initialzustand für counter definiert, d.h. der Signalzustand ist undefiniert 1P
 (U, $\frac{1}{2}$ P). Undefiniert+1 ist immer noch undefiniert ($\frac{1}{2}$ P).
- d) 1P

```

process (clk)
begin
  if clk'event and clk='1' then
    counter<=counter+1;
    if counter>190 then
      counter<=0;      end if;
  end process;

```

Hinweis: Eine Abfrage auf counter=191 ist nicht ausreichend, da aufgrund der fehlenden Initialisierung der Startzustand counter=192 auftreten kann. Somit würden die Zustände 192-255 nicht abgefangen.

e) Register-Transfer-Level (RTL), weil zielunabhängige Hardwarebeschreibung 1P

f) je fehlerhafte Zeile $\frac{1}{2}$ P Abzug 1,5P

Typ	UND-Matrix	ODER-Matrix
PLA	p	p
PAL	p	f
PROM	f	p

g) Beim PROM beschreibt die UND-Matrix die Adresse, die ODER-Matrix die Daten. 1P

h) Es werden 9 Aus- und 5 Eingänge benötigt ($\frac{1}{2}$ P), weiterhin sind Speicherglieder 1,5P
 erforderlich ($\frac{1}{2}$ P): Aus diesem Grund wird der 22V10 Verwendung finden ($\frac{1}{2}$ P).

(Keine Wertung von Baustein alleine; $\frac{1}{2}$ P Abzug je fehlendem/falschen Grund.)

Musterlösung 3: Prozessorarchitektur

12P

- a) 500MHz entsprechen 2ns Zykluszeit, also ergibt sich: 2P

$$\left(\frac{1}{2}P\right) t_{NP} = t_{cyc} * CPI$$

$$\left(\frac{1}{2}P\right) t_{NP} = 2ns * (2 * 0,6 + 5 * 0,3 + 13 * 0,1) = 2 * 4ns = 8ns$$

Für den Geschwindigkeitszuwachs (Speedup) gilt:

$$\left(\frac{1}{2}P\right) s = \frac{t_{NP}}{t_P}$$

$$\left(\frac{1}{2}P\right) s = \frac{8}{2+3} = \frac{8}{5} = 1,6$$

- b) Unbedingte Sprünge ($\frac{1}{2}P$), werden (soweit im Programmkontext möglich) entsprechend der Pipeline-Länge nach vorne verschoben (delayed branch, $\frac{1}{2}P$). 1P
- c) Abhängigkeiten sind potentielle Konfliktsituationen; der Konflikt kann, muss aber nicht eintreten. 1P
- d) Beides sind sprungzentrische (lokale) Prädiktoren. 1P
- e) Sie eignen sich aufgrund der zwangsläufig auftretenden Interferenz ($\frac{1}{2}P$) nicht für die Auflösung von abhängigen Sprüngen wie sie beispielsweise bei if/then-Konstrukten auftreten ($\frac{1}{2}P$). 1P
- f) ($\frac{1}{2}P$) gselect3/3: Es werden die unteren 3 Bit von A mit den unteren 3 Bit von H konkateniert. 1P
 ($\frac{1}{2}P$) gshare6/6: Es werden die unteren 6 Bit von A mit den unteren 6 Bit von H exklusiv-oder verknüpft.
- g) je Fehler $\frac{1}{2}P$ Abzug 4P

Ein-sprung	Sprung 1				Sprung 2			
	Prädiktor	Vorhersage	Sprung	P. neu	Prädiktor	Vorhersage	Sprung	P. neu
NT	(NT,NT)	NT	T	(T,NT)	(NT,T)	T	NT	(NT,NT)
-	(T,NT)	T	NT	(NT,NT)	(NT,NT)	NT	T	(T,NT)
-	(NT, NT)	NT	T	(NT, T)	(T,NT)	NT	NT	(T,NT)
-	(NT, T)	NT	NT	(NT, T)	(T, NT)	T	T	(T,NT)

- h) Der Hybridprädiktor verwendet einen sogenannten Selektionsprädiktor ($\frac{1}{2}P$) zur Auswahl des eigentlichen Sprungvorhersagemechanismus ($\frac{1}{2}P$). 1P

-
- Musterlösung 4: Speicherhierarchie und Parallelrechner** 15P
- a) Architekturen mit einem gemeinsamen Speicher 0,5P
- b) $S(n) = \frac{T(1)}{T(n)}$, d.h. $S(8) = \frac{1,6}{0,256} = 6,25$ 1P
 $E(n) = \frac{S(n)}{n}$, d.h. $E(8) = \frac{6,25}{8} = 0,78$
- c) Amdahls Gesetz: $T(n) = T(1) * (\frac{1-a}{n} + a)$ 1P
 $\rightarrow 0,256 = 1,6 * (\frac{1-a}{8} + a) = 1,6 * \frac{1+7a}{8} = 0,2 * (1 + 7a)$
 $\rightarrow 7a = 0,28 \rightarrow a = \frac{0,28}{7} = 4\%$ des Programmcodes sind nur sequentiell ausführbar
- d) $T(16) = T(1) * (\frac{1-a}{n} + a)$, d.h. $T(16) = 1,6 * (\frac{1-0,04}{16} + 0,04) = 0,16$ 1P
 $S(16) = \frac{1,6}{0,16} = 10$
- e) Dimension eines Kanals = Bitposition von (Start-Knoten# XOR End-Knoten#) 0,5P
- f) Wege vom Knoten 5=0101 zum Knoten 15=1111: 1,5P
A = 0101 und B = 1111 $\rightarrow W = A \text{ XOR } B = 1010$, somit 2 Wege:
Weg 1: 0101 \rightarrow 1101 \rightarrow 1111
Weg 2: 0101 \rightarrow 0111 \rightarrow 1111
- g) Erste Berechnungsphase: Jedem Prozessor werden N/P Elemente von jedem Array $1,5P$ zugeteilt; jeder Prozessor berechnet die Teilsumme von diesen Elementen.
Zeitschritte: N/P
Zweite Berechnungsphase: Addieren der Teilsummen von allen P Prozessoren in eine globale Variable.
Zeitschritte: P
- h) Es gilt: $r_{M2} = \frac{\#L2\text{-Misses}}{\#L2\text{-Zugriffe}}$ 2P
#L2-Misses = 50 (Zugriffe auf Hauptspeicher)
#L2-Zugriffe = #L1-Misses = $(1 - r_{H1}) * 500 = 100$
 $\rightarrow r_{M2} = \frac{50}{100} = 50\%$
 $t_a = r_{H1} * t_{L1} + r_{M1} * (r_{H2} * t_{L2} + r_{M2} * t_{mem})$
 $t_a = 0,8 * 1 + (1 - 0,8) * (0,5 * 5 + (1 - 0,5) * 150) = 16,3$ Zyklen
- i) (Linke Tabelle $\frac{1}{2}P$, rechte Tabelle 1P; je Fehler $\frac{1}{2}P$ Abzug) 2P

Adresse	Treffer
4415C01	-
001D3C0	-
24AD411	-
009F950	-
24AD47F	X
4415C19	-
3D972C0	-
001D36E	-

Satz	Block	Tags
0	0	4415C, 009F9, 4415C
	1	24AD4, 001D3
1	0	001D3,
	1	3D972

($\frac{1}{2}$ P): Unter 8 Zugriffen findet nur 1 Hit statt, daher ergibt sich die Treffer-Rate zu $\frac{1}{8} = 0.125$.

j) Erklärung der MESI-Zustände:

4P

Invalid: Die betrachte Zeile ist ungültig.

Exclusive (exclusive unmodified): Die betrachtete Zeile enthält eine exklusive, unveränderte Kopie des entsprechenden Speicherblocks.

Shared: Speicherblock existiert als unmodifizierte Kopie in der lokalen Cache-Zeile sowie in anderen Caches.

Modified (exclusive modified): Die betrachtete Cache-Zeile enthält eine exklusive, veränderte Kopie des entsprechenden Speicherblocks.

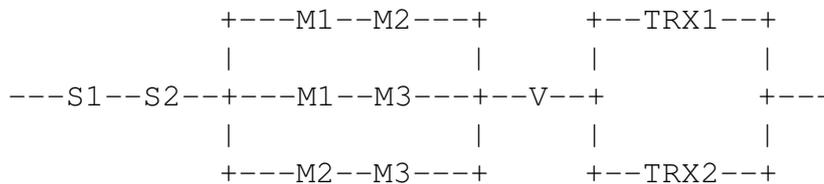
Zustand	Zustand Wechsel		Steuersignal
	nach	Ereignis	
Invalid	E	exclusive read miss	
	S	shared read miss (Block bereits in anderen Caches)	Shared-Signal
	M	write-miss	Invalid-Signal
Exclusive	I	snoop-hit on write (Schreibzugriff durch anderen Prozessor)	Invalid-Signal
	S	snoop-hit on read (Lesezugriff durch anderen Prozessor)	Shared-Signal
	M	write-hit	
Shared	I	snoop-hit on write (Schreibzugriff durch anderen Prozessor)	Invalid-Signal
	M	write-hit	Invalid-Signal
Modified	I	snoop-hit on write (Schreibzugriff durch anderen Prozessor)	Invalid-Signal
	S	snoop-hit on read (Lesezugriff durch anderen Prozessor)	Shared-Signal

Musterlösung 5: Fehlertoleranz

8P

a)

1P

b) (je Formelbestandteil $\frac{1}{2}P$)

1,5P

$$S = (S_1 \wedge S_2) \wedge ((M_1 \wedge M_2) \vee (M_1 \wedge M_3) \vee (M_2 \wedge M_3)) \wedge V \wedge (TRX_1 \vee TRX_2)$$

c)

1P

$$\Phi_m^{(n)} = \Phi(V) * \sum_{k=n}^m \binom{m}{k} * \Phi(K)^k * (1 - \Phi(M))^{(m-k)}$$

d) Der Entscheider ist ein single-point-of-failure.

0,5P

e) ($\frac{1}{2}P$ Abzug für falsche TRX-Berechnung, sofern restliche Formel korrekt)

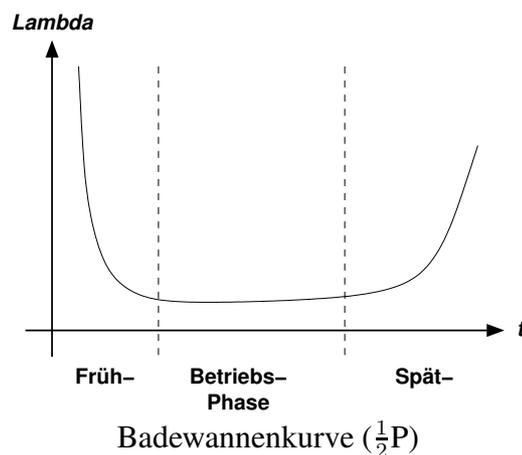
1P

$$\text{TRX parallel, d.h. } \Phi(TRX_p) = 1 - (1 - \Phi(TRX))^2$$

$$\text{Somit gilt für das Gesamtsystem: } \Phi(S) = \Phi(SP)^2 * \Phi(TMR) * (1 - (1 - \Phi(TRX))^2)$$

f)

2P



Frühphase: initiale Ausfälle / Fertigungsfehler,

Betriebsphase: quasi-lineare Ausfallrate,

Spätphase: altersbedingte Ausfälle (je $\frac{1}{2}P$).

g) Die Wahrscheinlichkeit eines Ausfalls in der Betriebsphase ist signifikant geringer als die initiale Ausfallwahrscheinlichkeit. 1P