



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

Klausur Rechnerstrukturen
Sommersemester 2013 – 7. Aug. 2013
Musterlösung

Aushang der Ergebnisse: ab Mitte September 2013

Musterlösung 1: Fehlertoleranz & Ausfallverhalten

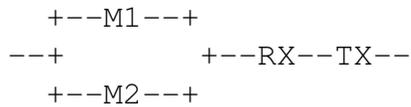
11P

Fehlertoleranz

8P

a) Zuverlässigkeitsblockdiagramm:

1P



b) $S = (M1 \vee M2) \wedge RX \wedge TX$

1P

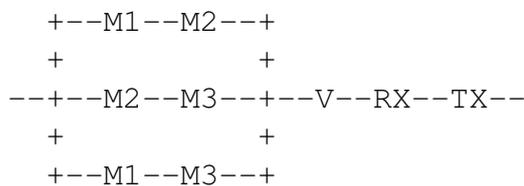
c) SP parallel, d.h. $\varphi(M_p) = 1 - (1 - \varphi(M))^2$

1P

Somit gilt für das Gesamtsystem: $\varphi(S) = \varphi(RX) * \varphi(TX) * (1 - (1 - \varphi(M))^2)$

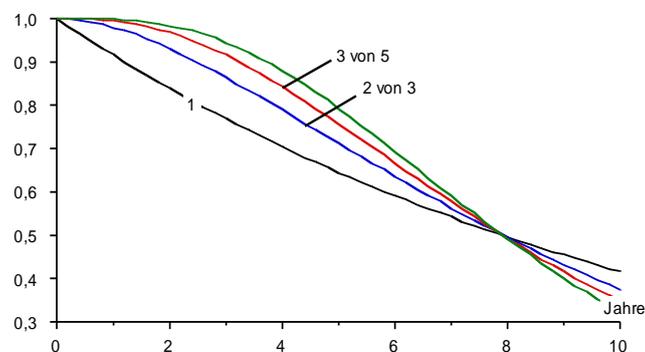
d) Zuverlässigkeitsblockdiagramm:

2P



e)

2P



Begründung: Das 3-aus-5-System zeigt in der Anfangszeit eine höhere Funktionswahrscheinlichkeit als das 2-aus-3-System. Nach einiger Zeit (Schnittpunkt der beiden Graphen im Bild) kehrt sich dieser Effekt jedoch um. Anfangs sind die einzelnen Funktionswahrscheinlichkeiten noch hoch und bei einem 3-aus-5-System können bis zu 2 Komponenten ausfallen. Wenn die Funktionswahrscheinlichkeit einen bestimmten Punkt unterschreitet, ist ein 2-aus-3-System sicherer, da nur 2 Komponenten zum Betrieb notwendig sind.

f) Das 3-aus-5-System, da ein 2-aus-3-System erst nach ca. 8 Jahren sicherer ist.

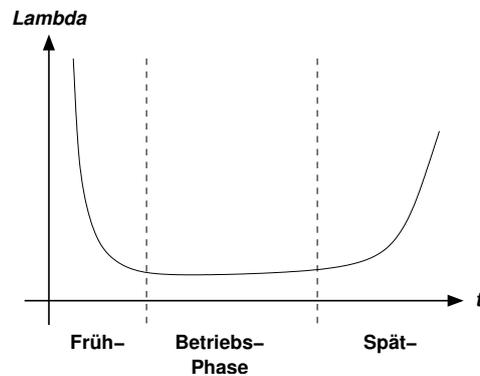
1P

Ausfallverhalten

3P

g) Badewannenkurve

1P



h) Badewannenkurve

2P

- Hohe Ausfallwahrscheinlichkeit in der Frühphase durch initiale Ausfälle verursacht durch Fertigungsfehler bzw. defekte Bauteile. (1P)
- Alterungseffekte bzw. Verschleiß führen zu hoher Ausfallwahrscheinlichkeit in der Spätphase. (1P)

Musterlösung 2: Low-Power-Entwurf & Rechnerbewertung

10P

Low-Power-Entwurf

5P

- a) Veränderung der Leistungsaufnahme:

1P

Spannungsabsenkung: $3,2V \rightarrow 0,8V$

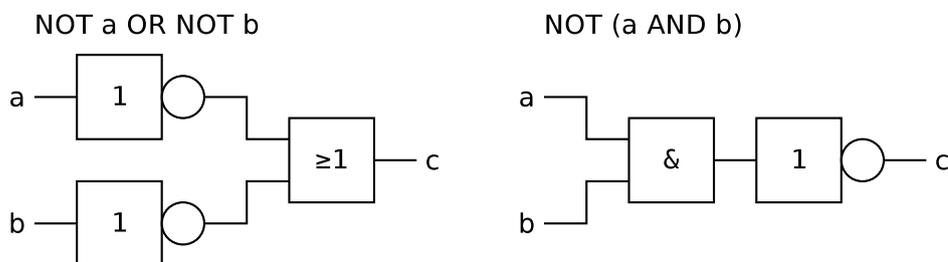
U^2 : mit Faktor $(1/4)^2 = 1/16$ verändert.

Frequenzerhöhung: $100MHz \rightarrow 4GHz$: Faktor 40

Aus $P \sim U^2 \cdot f$ resultiert eine Zunahme der elektrischen Leistung um den Faktor $40/16 = 2,5$.

- b) Schaltbilder:

1P



- c) Signalwahrscheinlichkeiten:

1P

Variante 1 (0,5P):

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

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

$$\mathbb{P}_C(1) = 1 - (1 - \mathbb{P}_{\neg A}(1)) * (1 - \mathbb{P}_{\neg B}(1)) = 1 - (1 - 2/3) * (1 - 1/2) = 1 - 1/6 = 5/6$$

Variante 2 (0,5P):

$$\mathbb{P}_{A \wedge B}(1) = 1/3 * 1/2 = 1/6$$

$$\mathbb{P}_{\neg(A \wedge B)}(1) = 1 - 1/6 = 5/6$$

- d) Formel für Schaltwahrscheinlichkeit (0,5P):

2P

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

Berechnung der Schaltwahrscheinlichkeiten:

Variante 1 (0,5P):

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

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

$$\mathbb{P}_{Schalt \neg A \vee \neg B} = 2 * 5/6 * 1/6 = 5/18$$

Variante 2 (0,5P):

$$\mathbb{P}_{Schalt\ A\wedge B} = 2 * 1/6 * 5/6 = 5/18$$

$$\mathbb{P}_{Schalt\ \neg(A\wedge B)} = 2 * 5/6 * 1/6 = 5/18$$

$$\text{Gesamtschaltwahrscheinlichkeit (0,5P): } \mathbb{P}_{Schalt_2\ gesamt} = 5/18 + 5/18 - 25/324 = (180 - 25)/324 = 155/324$$

$$(\mathbb{P}_{Schalt_1\ gesamt} = 8/18 + 9/18 + 5/18 - 8*9/18^2 - 9*5/18^2 - 8*5/18^2 + 8*9*5/18^3 = 22/18 - (72 + 45 + 40)/324 + 360/(324 * 18) = 18 * 22/324 - 157/324 + 20/324 = (396 - 137)/324 = 259/324 \text{ ist auch gültig})$$

Leistungsbewertung

5P

e) Prozessorwahl:

1P

$$f = \frac{\#Instruktionen * CPI}{Zeit}$$

$$f_A = \frac{3,5 * 10^6 * 12/7}{2 * 10^{-3} s} = \frac{6}{2s} * 10^9 = 3GHZ$$

$$f_B = \frac{0,5 * 10^6 * 9 + 1 * 10^6 * 2}{2 * 10^{-3} s} = \frac{6,5}{2s} * 10^9 = 3,25GHZ$$

Zwar benötigt Prozessor B weniger Instruktionen, dennoch ist Prozessor A zu wählen, da die niedrigere Frequenz weniger Leistungsaufnahme verspricht, denn letztere ist proportional zur Taktfrequenz.

f) Durchsätze: $D_i\ max = 1/\text{Bedienzeit } X_i$; $X_i = t_{\text{Zugriff}} + t_{\text{Übertragung}}$

1P

$$X_1 = 8ms + \frac{100kB}{25.000kB/s} = 8ms + 1/250s = 12ms \rightarrow D_1 = \frac{1}{12ms}$$

$$X_2 = 12ms + \frac{100kB}{50.000kB/s} = 12ms + 1/500s = 14ms \rightarrow D_2 = \frac{1}{14ms}$$

g) Einsetzbare Festplatten:

1P

Nur Festplatten mit $D_i > A$ können eingesetzt werden, da sonst die Festplatte nicht genügend Zeit hat, alle Aufträge rechtzeitig zu bedienen.

$$A = 60/s = \frac{720}{12s} < \frac{1000}{12s} = D_1. \text{ Festplatte 1 kann verwendet werden.}$$

$$A = 60/s = \frac{840}{14s} < \frac{1000}{14s} = D_2. \text{ Festplatte 2 kann auch verwendet werden.}$$

h) Name des Gesetzes: Gesetz von Little

1P

Formel: $Q = W * D$; Q: Anzahl von Aufträgen in der Warteschlange; W: Wartezeit; D: Durchsatz.

Zeit W: $W_i = Q_i/D_i$, wobei hier $D=A$ zu setzen ist, da die tatsächlich angefragte bzw. genutzte Rate zu berücksichtigen ist.

$$W_1 = \frac{3}{60/s} = 50ms$$

$$W_2 = \frac{6}{60/s} = 100ms$$

i) Reaktionszeit:

1P

Reaktionszeit des Gesamtsystems aus Warteschlange und Festplatte:

Reaktionszeit_{*i*} = Wartezeit_{*i*} + Bedienzeit_{*i*}.

Reaktionszeit₁ = 50ms + 12ms = 62ms

Reaktionszeit₂ = 100ms + 14ms = 114ms

Musterlösung 3: Speicherhierarchie

10P

Cache-Kohärenzprotokoll MESI

7P

a)

4P

Prozessor	Aktion	Prozessor 1			Prozessor 2		
		Line 1	Line 2	Line 3	Line 1	Line 2	Line 3
1	rd 6	6/E					
2	rd 2				2/E		
1	wr 6	6/M					
1	wr 4		4/M				
2	rd 3					3/E	
1	rd 5			5/E			
2	rd 6	6/S					6/S
2	wr 5			5/I	5/M		

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

1P

c) Mögliche Antworten:

2P

- 2 Zustandsbit in jeder Cache-Zeile zur Speicherung des MESI-Zustandes
- Bus-Monitor zur Überwachung des Speicherbusses
- Retry, Abort und Invalidate Signale zu weiteren Prozessoren
- Erweiterung des Cache-Controllers um den MESI-Zustandsautomat

Cache-Leistung

3P

d) Formel:

1P

$$t_a = r_{Hit-L1} * t_{L1} + r_{Miss-L1} * (r_{Hit-L2} * t_{L2} + (r_{Miss-L2} * t_{Mem})) = r_{Hit-L1} * t_{L1} + (1 - r_{Hit-L1}) * (r_{Hit-L2} * t_{L2} + (1 - r_{Hit-L2}) * t_{Mem})$$

Oder für sequentielle Abfrage:

$$t_a = r_{Hit-L1} * t_{L1} + r_{Miss-L1} * (r_{Hit-L2} * (t_{L1} + t_{L2}) + (r_{Miss-L2} * (t_{L1} + t_{L2} + t_{Mem})))$$

e) Durchschnittliche Zugriffszeit:

2P

Bei dem System ist zu berücksichtigen, dass die Anfragen an die übergeordneten Hierarchieebenen mitgezählt werden müssen.

$$t_a = 0,7 * 4ns + 0,3 * (4ns + 7/8 * 16ns + 1/8 * (16ns + 120ns)) = 2,8ns + 0,3 * (18ns + 1/8 * 136ns) = 2,8ns + 0,3 * (18ns + 17ns) = 2,8ns + 105ns/10 = 13,3ns$$

Musterlösung 4: Fertigung und Hardwareentwurf

11P

Fertigungskosten

5P

a)

3P

- Formel: $cost_{die} = \frac{cost_{wafer}}{dpw * yield_{die}}$

- Rechnung und Antwort:

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

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

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

$$cost_{w,r} = \frac{1000 * 600 * 0,2}{400 * 0,5} = \frac{200 * 600}{200} = 600$$

→ Der rechteckige Wafer darf maximal 600 Euro kosten.

- Begründung: Für das alte Verfahren, da es trotz der geringeren Anzahl an Dies pro Wafer geringere Kosten pro nutzbarem Die verursacht.

b) Formel für kreisförmige Wafer: $dpw = A - B = \frac{\pi * (d_{wafer}/2)^2}{a_{die}} - \frac{\pi * d_{wafer}}{\sqrt{2} * a_{die}}$ 1,5P

A: theoretisches Maximum

B: Verschnitt

(0,5P Abzug jeweils für Fehler, fehlende Beschreibung oder ungenaue Angaben)

- c) Antwort: Nein, auch bei rechteckigen Wafern kann es zu Verschnitt kommen, wenn die Seitenlänge nicht ein Vielfaches der Die-Seitenlänge ist. 0,5P

Hardwareentwurf

6P

- d) Mittel zur Verbindung: Signale

1P

- e) Andere Version: `c <= a nor b;`

1P

- f) VHDL-Beschreibung:

4P

```
library ieee;
use ieee.std_logic_misc.all;

entity fulladder is
  port (
    a_i, b_i, c_im1 : in std_logic;
    s_i, c_i : out std_logic
  );
end entity;

architecture structure of fulladder is
```

```
component halfadder is
  port (
    in1, in2 : in std_logic;
    sum, carry : out std_logic
  );
end component;

signal sum1, carry1, carry2 : std_logic;

begin

  ha1 : halfadder port map (c_im1, a_i, sum1, carry1);

  ha2 : halfadder port map (sum1, b_i, s_i, carry2);

  c_i <= carry1 or carry2;

end architecture;
```

Punktevergaben (je 0,5P):

- Entity-Name „Volladdierer“ o.ä.
- Ports prinzipiell richtig eingetragen
- Einwertiger Typ „bit“ oder „std_logic“
- Architekturname „structure“ oder „mixed“ o.ä.
- Komponentename „Halbaddierer“ o.ä.
- Hilfssignal zur Verdrahtung der Instanzen
- Zweifache Instantiierung der Komponente „Halbaddierer“
- Korrekte Verdrahtung der Instanzen mitsamt Hilfssignal

Musterlösung 5: Parallelverarbeitung und Architekturen

9P

Quantitative Maßzahlen

5P

a) $S(n) = T(1)/T(n)$

2P

$$E(n) = S(n)/n$$

$$I(n) = P(n)/T(n)$$

$$U(n) = I(n)/n = R(n) * E(n)$$

b) Nach Amdahls Gesetz $T(n) = \frac{T(1)}{n} * (1 - a) + T(1) * a$ ist die maximal erzielbare Beschleunigung durch den nur sequentiell ausführbaren Anteil a beschränkt zu $S(n) \leq \frac{1}{\frac{1-a}{n} + a}$.

c) Architekturauswahl:

1P

$$T_A(n) = 0,8 * 120s/16 + 0,2 * 120s = 6s + 24s = 30s.$$

$$T_B(n) = 0,72 * 100s/24 + 0,28 * 100s = 3s + 28s = 31s.$$

Wir wählen Architektur A aus, da sie um eine Sekunde schneller ist.

Alternativer Rechnungsweg:

$$S_A(n) \leq 1/(0,8/16 + 0,2) = 1/0,25 = 4 \text{ auf zwei Minuten anzuwenden} \Rightarrow 30 \text{ Sekunden.}$$

$$S_B(n) \leq 1/(0,72/24 + 0,28) = 1/0,31 \text{ auf 100 Sekunden anzuwenden} \Rightarrow 31 \text{ Sekunden.}$$

Parallele Architekturen und Parallelisierung

4P

d)

1,5P

- NORMA: No Remote Memory Access
- NUMA: Non-Uniform Memory Access
- UMA: Uniform Memory Access

e)

1P

- OpenMP für gemeinsamen Speicher
- MPI für verteilten Speicher

f) Vier Klassen von Rechnerarchitekturen:

1P

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

g) MIMD: Multiple Instruction – Multiple Data

0,5P

Musterlösung 6: Parallelismus auf Befehlsebene

9P

a) VLIW-Prozessoren

4P

Slot 1	Slot 2
1) add r1, r2, r3	2) ld f1, [r2]
3) sub r5, r3, r2	6) ld r6, [r3]
4) fpdiv f3, f1, r5	5) st [r5], r1
7) mul r4, r1, r6	8) fpadd f5, f3, f1

System A

Slot 1	Slot 2
3) sub r5, r3, r2	2) ld f1, [r2]
1) add r1, r2, r3	4) fpdiv f3, f1, r5
6) ld r6, [r3]	5) st [r5], r1
7) mul r4, r1, r6	8) fpadd f5, f3, f1

System A

Integer	Gleitkomma	Load/Store
3) sub r5, r3, r2		2) ld f1, [r2]
1) add r1, r2, r3	4) fpdiv f3, f1, r5	6) ld r6, [r3]
7) mul r4, r1, r6	8) fpadd f5, f3, f1	5) st [r5], r1

System B

Entscheidung:

Wir entscheiden uns für System A, da dieses zwar 33% langsamer ist, aber weniger Kosten verursacht und wesentlich flexibler ist, weshalb es für weitere Anwendungen vielversprechender ist.

Oder: Wir entscheiden uns für System B, da dieses 25% schneller ist bei nur 10% mehr Kosten.

b) Superskalare Befehlsausführung nach dem Algorithmus von Tomasulo

5P

#	Befehlsfolge	Issue	Execute	Write Result
1	mul r2, r1, r1	1	2	6
2	div r4, r4, r2	2	7	15
3	add r1, r4, r4	3	16	18
4	add r2, r4, r3	4	19	21
5	div r1, r2, r3	7	22	30
6	sub r4, r4, r2	8	22	24
7	add r3, r1, r2	9	31	33
8	mul r1, r2, r3	16	34	
9	add r3, r3, r3	19		
10	sub r4, r4, r1	22		

Listing 1

(Format: Opcode Ziel, Quelle 1, Quelle 2)

Feld	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
Value	(R4+R4)	(R1*R1)	(R3)	–	(R4+R3)	–	(R4-R5)	–	–	–
Valid	1	1	0	0	1	0	1	0	0	–
RS	–	–	AS2	AS2	–	MD1	–	MD1	AS1	–

Registerdatei nach Takt 25

Unit	Empty	InFU	Op	Dest	Src1	Vld1	RS1	Src2	Vld2	RS2
Add/Sub 1	0	0	add9	R9	R3	0	AS2	R3	0	AS2
	1	0	sub6	R7	(R4)	1	–	(R5)	1	–
Add/Sub 2	0	0	sub10	R4	R7	1	(AS1)	(R8)	0	MD1
	0	0	add7	R3	R6	0	MD1	(R5)	1	–
Mul/Div 1	0	1	div5	R6	(R5)	1	–	(R3)	1	–
	0	0	mul8	R8	(R5)	1	–	R3	0	AS2

Reservierungstabelle nach Takt 25