



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

**Klausur Rechnerstrukturen**  
**Wintersemester 2009/10**  
**Musterlösung**

Aushang der Ergebnisse: ab Mitte April 2010

## Musterlösung 1: Quantifizierung

10P

a)

2P

( $\frac{1}{2}$ P) Gesetz von Little:

( $\frac{1}{2}$ P)  $Q = W * D \Leftrightarrow D = \frac{Q}{W}$ , für  $W \neq 0$

Q: Anzahl von Aufträgen in der Warteschlange,

W: Wartezeit,

D: Durchsatz

(1P) Setze  $Q = 50$ :

$$\Rightarrow D_A = \frac{50}{10 \text{Min}} = 5 \frac{\text{Jobs}}{\text{Minute}}$$

$$\Rightarrow D_B = \frac{50}{15 \text{Min}} = 3,3 \frac{\text{Jobs}}{\text{Minute}}$$

b)  $240 \frac{\text{Jobs}}{\text{Stunde}} \simeq 4 \frac{\text{Jobs}}{\text{Minute}}$ , womit sich aufgrund des höheren Durchsatzes System A qualifiziert.

0,5P

$D_B < 4$ : System B erfüllt die Anforderungen nicht.

c)

2P

( $\frac{1}{2}$ P) Hardwaremonitore

( $\frac{1}{2}$ P) Vorteil: unabhängige physikalische Geräte, dadurch keine Beeinflussung des zu beobachtenden Systems und damit exakte Ergebnisse

oder Nachteil: aufwändigere Implementierung und Installation

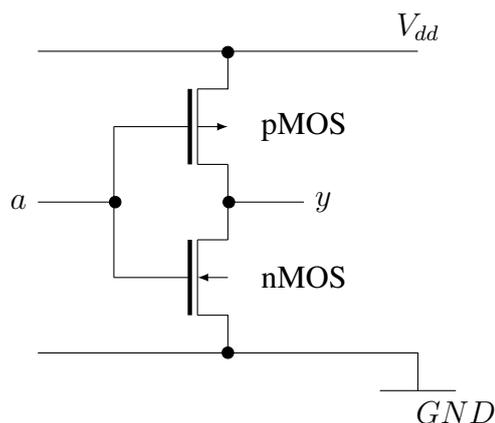
( $\frac{1}{2}$ P) Softwaremonitore

( $\frac{1}{2}$ P) Vorteil: einfachere Implementierung

oder Nachteil: Beeinträchtigung der normalen Betriebsverhältnisse, da typischerweise im Betriebssystem installiert - damit Ergebnisse weniger exakt

d) Inverter:

0,5P



e) Durch die Anordnung der pMOS und nMOS-Netze ist gewährleistet, dass bei konstanten Eingangsvariablen kein Strom von  $V_{dd}$  nach  $GND$  fließt.

0,5P

f) Da sich beim Umschalten ein kurzzeitiger Stromfluß von  $V_{dd}$  nach  $GND$  nicht ver-

0,5P

meiden lässt, versucht man die Funktion  $f$  so zu implementieren, dass die Schaltwahrscheinlichkeit der einzelnen Gatter minimiert ist.

- g) statischer und dynamischer Teil von  $P_{total}$  **0,5P**
- h)  $P_{leakage}$  ständiger Stromfluß durch Leckströme ( $\frac{1}{2}$ P) **1,5P**  
( $\frac{1}{2}$ P) Widerstände zwischen den Leiterbahnen sind nicht unendlich hoch  
( $\frac{1}{2}$ P) Leckströme wachsen mit zunehmender Miniaturisierung
- i) ( $\frac{1}{2}$ P)  $dpw = \frac{\pi * (d_{wafer} * \frac{1}{2})^2}{a_{die}} - \frac{\pi * d_{wafer}}{\sqrt{2} * a_{die}} = A - B$  **2P**  
( $\frac{1}{2}$ P) A: Gesamtfläche, B: Verschnitt  
(1P) Die bisherige Grundfläche des Wafers ist rund und damit fällt der Verschnitt  $B$  an. Durch Verwendung eines rechteckigen Wafers lässt sich der Verschnitt stark reduzieren ( $B \rightarrow 0$ ).

## Musterlösung 2: Hardwareentwurf

10P

a) (je  $\frac{1}{2}P$  für Einsatzgebiet und Erklärung)

2P

- Simulation: Überprüfung der im Entwurf befindlichen Schaltung auf theoretische Korrektheit
- Synthese: Überführung einer abstrahierten Schaltungsbeschreibung in eine detaillierte topologische Struktur und damit auf Netzliste und Zieltechnologie
- Verifikation: Überprüfung einer Schaltung bezüglich ihrer Funktionsweise gegenüber der Spezifikation

b) (1P für eine korrekte Antwort)

1P

- Verwendung von alternativer, kompatibler Einheit
- Festlegung verwendeter Architecture
- Angabe von Generics wie etwa Signalbreiten

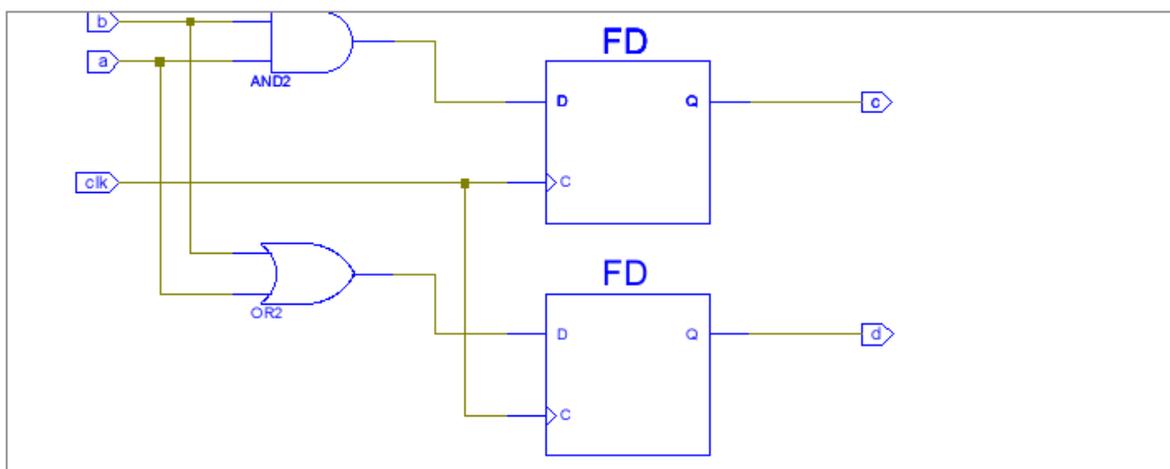
c) (je  $\frac{1}{2}P$  für korrekte Zeile; korrekte Syntax nicht erforderlich)

3P

```
entity schaltung is
  Port ( clk : in bit;
        a   : in bit;
        b   : in bit;
        c   : out bit;
        d   : out bit
  );
end entity;
```

d) (je  $\frac{1}{2}P$  für D-Latches bzw. Gatter)

2P



**Signale und Variablen****2P**

- e) Signale werden innerhalb der Architektur angelegt ( $\frac{1}{2}P$ ), während Variablen innerhalb eines Prozesses ( $\frac{1}{2}P$ ) definiert werden. *1P*
- f) Der grundsätzliche Unterschied liegt in der Wertzuweisung: Bei einer Variable erfolgt diese sequentiell und **unmittelbar** ( $\frac{1}{2}P$ ), während sie bei Signalen taktsynchron ( $\frac{1}{2}P$ ) geschieht. *1P*

Ein weiterer Unterschied liegt im Gültigkeitsbereich: Das Signal gilt über die ganze Architektur hinweg und der Zugriff darauf muss entsprechend abgestimmt werden; die Variable gilt lediglich innerhalb des Prozesses.

## Musterlösung 3: Prozessorarchitektur

10P

a) Formeln:

1P

- $T_{seq} = n * k$
- $T_{pipe} = n + k - 1$

0,5P

- $S = \frac{T_{seq}}{T_{pipe}} = \frac{n*k}{n+k-1}$

0,5P

b) Berechnung:

1P

- $T_{eff\_takte} = 10 * 10^6 + 0.02 * 10 * 10^6 + 0.06 * 10 * 10^6 * 3 + 5 - 1 = 12.000.004$
- $T_{eff\_sekunden} = 12.000.004 * \frac{1}{2GHz} \approx 0,006s = 6ms$

0,5P

- $S = \frac{T_{seq}}{T_{pipe}} = \frac{5*10^7}{12.000.004} \approx 4$

0,5P

c) Architekturvergleich:

1,5P

- $CPI_A = \frac{\#Takte_A}{Instruktion} = \frac{12.000.004}{10.000.000} \approx 1.2 (\frac{1}{2}P)$
- $\#Takte_B = 6ms * 2.5GHz = 6 * 10^{-3}s * \frac{2.5*10^9}{s} = 15 * 10^6$
- $CPI_B = \frac{\#Takte_B}{12.500.000} = 1.2 (\frac{1}{2}P)$

1P

- Wahl und Begründung: Die CPI-Werte sind für beide Architekturen gleich; der CPI-Wert eignet sich aber ohnehin nicht zur Bewertung von Rechensystemen. Wir wählen Architektur A, da die Anzahl benötigter Takte bei gleicher absoluter Ausführungszeit geringer ist und sie somit weniger Gatter schalten muss in der gleichen absoluten Zeit, was die gesamte Energieaufnahme reduziert.

0,5P

d) Dynamische Verfahren (je  $\frac{1}{2}P$ ):

1P

- Stalling/Interlocking
- Forwarding
- Forwarding mit Interlocking

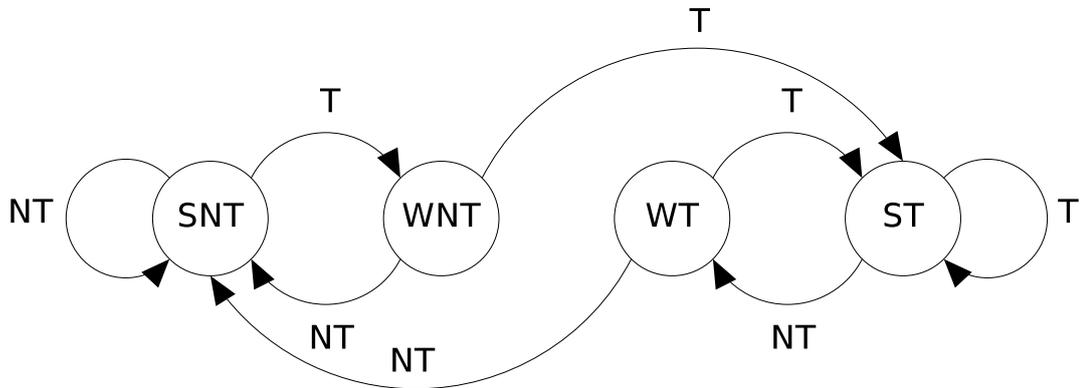
e) Skizze BTAC

1P

Adresse der Verzweigung	Sprungziel-adresse	Vorhersagebits / Sprungverlaufs-anzeige

f) Zeichnung 2-Bit-Hysteresezähler

1P



Kodierung: 00                      01                      10                      11

g) Sprungverläufe (je  $\frac{1}{2}$ P pro korrektem Sprung):

1P

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

h) Tomasulo-Algorithmus:

2,5P

Registerstatustabelle:

1P

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

Reservation Stations:

1,5P

---

RS#	Leer	In FU	Opcode	Ziel	Quelle1	Gültig1	RS1	Quelle2	Gültig2	RS2
1 Addierer	0	1	sub	2	(R4)	1	0	(R3)	1	0
2 Addierer	1									
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

### Quantitative Maßzahlen

3P

a) Amdahls Gesetz:

1P

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

$$80 = 2560 * \left( \frac{1-a}{64} + a \right) = 2560 * \frac{1+63a}{64} = 40 * (1 + 63a)$$

$$\frac{80}{40} = 2 = 1 + 63a \Rightarrow a = \frac{1}{63}$$

b) • Algorithmenunabhängige Definition (0,5P):

1P

Bester sequentieller mit bestem parallelen Algorithmus

• Nachteil der algorithmenabhängigen Definition (0,5P):

Der für die Parallelisierung erforderliche Zusatzaufwand an Kommunikation und Synchronisation kommt „ungerechterweise“ auch für den sequentiellen Algorithmus zum Tragen.

c) • (0,5P) Der Parallelindex gibt eine obere Schranke für die Beschleunigung.

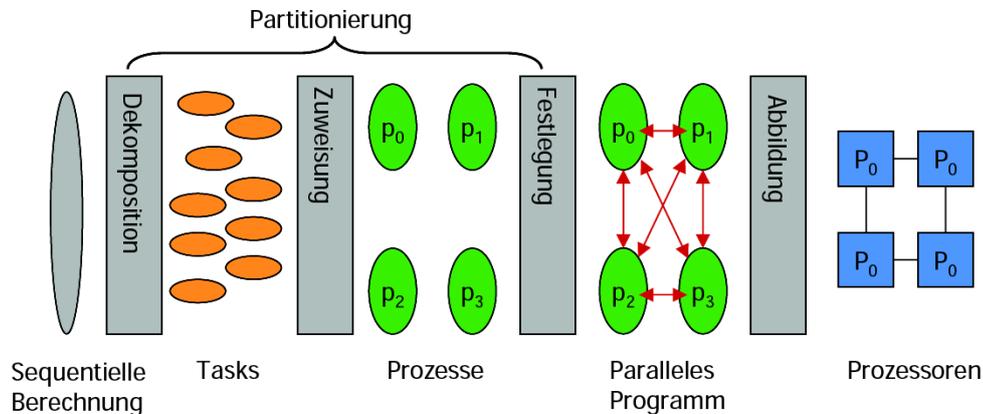
1P

• (0,5P)  $1 \leq S(n) \leq I(n) \leq n$

### Parallelisierung

2P

d) (Pro Begriff 0,5P)



**Verbindungsnetze und -strukturen****2P**

- e)   • (0,5P) Jede Erweiterung benötigt mindestens die Verdopplung der Prozessorenanzahl. 1P
- (0,5P) Der Verbindungsgrad der Knoten steigt bei jeder Erweiterung um 1. Rechner sind deshalb aus räumlichen Anordnungsgründen begrenzt.
- f)   • (0,5P) Knotenzahl:  $N = 2^n$  1P
- (0,5P) Diameter:  $\log_2 N$

**Vektorverarbeitung****3P**

- g)   • (0,5P) Vektorprozessoren (Vektorrechner): SIMD 1P
- (0,5P) Nachrichtengekoppelte Parallelrechner: MIMD
- h) Jeder Vektorrechner besitzt in seinem Rechenwerk einen Satz von Vektorpipelines. 1P
- i) Die bei den Gleitpunkteinheiten skalarer Prozessoren nötigen Adressrechnungen entfallen. 1P

## Musterlösung 5: Speicherhierarchie

10P

a)  $t_a = r_{H-L1} * t_{H-L1} + (1 - r_{H-L1}) * t_{Mem}$  **0.5P**

b)  $t_a = r_{H-L1} * t_{H-L1} + (1 - r_{H-L1}) * t_{Mem}$  **0.5P**  
 $t_a = 80\% * 2.5 ns + 20\% * 40 ns = 10 ns$

c) Taktfrequenz  $f = 2 GHz \Rightarrow$  Zykluszeit  $= 0.5 ns \Rightarrow t_a = 20$  Zyklen **1P**  
 Dauer  $= 1000000 * 1$  Zyklus  $+ 250000 * 20$  Zyklen  $= 6000000$  Zyklen

d) Write-Through-Cache  $\Rightarrow$  Schreiboperationen werden direkt in den Hauptspeicher geschrieben. **1P**

e)  $\Rightarrow$  Zugriffszeit immer  $t_{Mem} = 40 ns$  für Store-Operationen. ( $\frac{1}{2}$ P) **1P**

$$Dauer = \sum Anz_{OP} * Dauer_{OP}$$

$$Dauer = Dauer_{Integer} + Dauer_{Cache-Hit} + Dauer_{Cache-Miss}$$

$$Dauer_{Integer} = 1000000 * 1 \text{ Zyklus} = 1000000 \text{ Zyklen}$$

$$Dauer_{Cache-Hit} = 160000 * 5 \text{ Zyklen} = 800000 \text{ Zyklen}$$

$$Dauer_{Cache-Miss} = 90000 * 80 \text{ Zyklen} = 7200000 \text{ Zyklen}$$

$$Dauer = 1000000 \text{ Zyklen} + 4000000 \text{ Zyklen} + 4000000 \text{ Zyklen} = 9000000 \text{ Zyklen} (\frac{1}{2}P)$$

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

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	-	-	-	-	-	-
3	rd 4					4/E	
2	rd 1			1/E			
1	rd 4	4/S				4/S	
3	rd 2						2/E
3	wr 3					3/M	
1	rd 1		1/S	1/S			
1	wr 3	3/M				-/I	
2	rd 2				2/S		2/S
3	rd 1		1/S	1/S		1/S	
1	wr 1		1/M	-/I		-/I	
2	rd 4			4/E			
1	wr 2	2/M			-/I		-/I

h) Nein, bei obiger Speicherzugriffsfolge würde MOESI zu keiner Leistungssteigerung führen ( $\frac{1}{2}$ P). In obiger Speicherzugriffsfolge existiert kein Übergang von **M** nach **O**, **1P**

deshalb gibt es auch keinen Cache zu Cache-Transfer, der sich leistungssteigernd auswirken könnte( $\frac{1}{2}P$ ).

## Musterlösung 6: Fehlertoleranz

10P

a)

3P

- ungenutzte Redundanz: Ersatzkomponenten führen keine sonstigen Funktionen aus und bleiben bis zur fehlerbedingten Aktivierung passiv. 1P
- fremdgenutzte Redundanz: Ersatzkomponenten erbringen nur Funktionen die nicht zum betreffenden Subsystem gehören und im Fehlerfall verdrängt werden. 1P
- gegenseitige Redundanz: Ersatzkomponenten erbringen die von einer anderen Komponente zu unterstützenden Funktionen. Die Komponenten stehen sich gegenseitig als Reserve zur Verfügung. 1P

b)

1P

$$F_L(t) = \int_0^t f_L(s) ds$$

c) ( $\frac{1}{2}$ P) Fail-stop-System bezeichnet ein System, dessen Ausfälle nur Anhalteausfälle sind. 1,5P

( $\frac{1}{2}$ P) Fail-silent-System bezeichnet ein System, dessen Ausfälle nur Unterlassungsausfälle sind.

( $\frac{1}{2}$ P) Antwort: für Höchstverfügbarkeit nimmt man Fail-safe-Systeme, da bei diesen Systemen nur unkritische Ausfälle auftreten.

d) Blockdiagramm( $\frac{1}{2}$ P):

1,5P

```

+--M1--+
--+      +--V--
+--M2--+

```

Die Schwachstelle der gewählten Architektur ist, dass Fehler zwar erkannt, jedoch keine Mehrheitsentscheidung getroffen werden kann, da nur 2 Ergebnisse vorliegen. ( $\frac{1}{2}$ P)  
Um dem System diese Art der Entscheidung zu ermöglichen ist die Erweiterung um einen weiteren Microcontroller zwingend, so dass sich ein Triple-Modular-Redundancy-System (TMR) ergibt. ( $\frac{1}{2}$ P)

e) (je Fehler  $\frac{1}{2}$ P Abzug)

1P

```

+--SF1--+
+      +
+--SF2--+
--+      +--M--B--
+--SF3--+
+      +
+---A---+

```

f) (je Fehler  $\frac{1}{2}$ P Abzug)

1P

$$S = (SF1 \vee SF2 \vee SF3 \vee A) \wedge M \wedge B$$

---

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

**1P**

Solarfelder und Akku parallel, d.h.  $\varphi(\text{Para}) = 1 - (1 - \varphi(\text{SF}))^3 * (1 - \varphi(A))$

Somit gilt für das Gesamtsystem:

$$\varphi(S) = \varphi(\text{Para}) * \varphi(M) * \varphi(B) = ((1 - (1 - \varphi(\text{SF}))^3 * (1 - \varphi(A))) * \varphi(M) * \varphi(B))$$