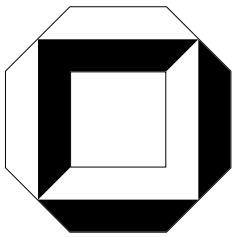


Universität Karlsruhe (TH)
Institut für Technische Informatik
Prof. Dr. Wolfgang Karl

Klausur Rechnerstrukturen
Wintersemester 2008/09
Musterlösung

Aushang der Ergebnisse: ab Ende April 2009

Musterlösung 1: Quantifizierung *10P*

a) **1P**

($\frac{1}{2}$P) C_{eff} muss schneller geladen werden, daher erfordert die Übertaktung typischerweise auch eine Erhöhung der Spannungsversorgung.

($\frac{1}{2}$P) Die Frequenz ist heute um ein vielfaches höher (1MHz→1GHz, also etwa Faktor 1000, die Spannungsabsenkung beträgt jedoch nur einen maximalen Faktor 17,64).

b) **3P**

Dynamisch ($\frac{1}{2}$P): $P_{switching}$, $P_{shortcircuit}$ ($\frac{1}{2}$P)
Statisch ($\frac{1}{2}$P): $P_{leakage}$, P_{static} ($\frac{1}{2}$P)

Eine immer größere Rolle spielt $P_{leakage}$ ($\frac{1}{2}$P), deswegen Absenkung der Spannungsversorgung ($\frac{1}{2}$P) (genauer: Herabsetzen der Schwellwertspannung, damit auch Herabsetzen der Schaltspannung möglich).

c) Es handelt sich um die Energiedichte ($\frac{1}{2}$P) bestehend aus Anzahl von Schaltvorgängen pro Zeiteinheit (Schaltfrequenz) und Anzahl von Schaltelementen pro Fläche ($\frac{1}{2}$P). **1P**

d) Effizienz (CPI) **0,5P**

e) Sie stellen die architekturunabhängige ($\frac{1}{2}$P) Vergleichbarkeit der Leistung unterschiedlicher Prozessoren/Systeme sicher ($\frac{1}{2}$P). **1P**

f) Gleiches OS und gleicher Compiler/Compilereinstellungen ($\frac{1}{2}$P) zur Sicherstellung, dass keine hieraus resultierenden Störeinflüsse das Messergebnis verfälschen ($\frac{1}{2}$P). **1P**

g) SPECint zur Messung der Integer-Rechenleistung ($\frac{1}{2}$P), SPECfp zur Messung der Fließkomma-Rechenleistung ($\frac{1}{2}$P). **1P**

h) ($\frac{1}{2}$P) Es handelt sich um die $SPEC_{ratio}$. **1P**
($\frac{1}{2}$P) $SPEC_{ratio} = \frac{Referenzzeit}{Laufzeit\ auf\ Testsystem}$

i) Aggressive Optimierung. **0,5P**

Musterlösung 2: Hardwareentwurf *10P*

a) Der Gestaltungsgrundsatz der Orthogonalität ($\frac{1}{2}$P) fordert die unabhängige Spezifizierung und Implementierung unabhängiger Einheiten ($\frac{1}{2}$P). Dies ist im vorliegenden Beispiel verletzt. **1P**

b) Hier wurde der Gestaltungsgrundsatz der Symmetrie verletzt ($\frac{1}{2}$P), d.h. die Forderung nach gleichartiger Behandlung gleichartiger Befehlstypen ($\frac{1}{2}$P). **1P**

c) Einem solchen Entwurf liegt das Prinzip der Sparsamkeit ($\frac{1}{2}$P) zugrunde. Erzielt wird dies durch Wiederverwendbarkeit ($\frac{1}{2}$P) der einzelnen Komponenten. In VHDL geschieht dies durch Verwendung sogenannter `components` ($\frac{1}{2}$P). **3P**

Die Kapselung findet in der sogenannten `entity` ($\frac{1}{2}$P) statt, wohingegen die Funktionsbeschreibung Teil der `architecture` ($\frac{1}{2}$P) ist. Zur Validierung ist zusätzlich eine sogenannte Testbench ($\frac{1}{2}$P) erforderlich.

d) **1,5P**

- Es handelt sich um die XOR-Funktion oder Antivalenz. *0,5P*
- `x<='0' when a=b else '1';` *1P*

oder `x<='1' when a/=b else '0';`

oder
```
x<='1' when a='0' and b='1'
   else '1' when a='1' and b='0'
   else '0';
```

e) **3,5P**

(e.1) *1,5P*
```
process(a,b,clk)
begin
  if clk'event and clk='0' then
    x<=a and b;
  end if;
end process;
```

Bewertung: Korrekte Sensitivity-List ($\frac{1}{2}$P), korrekte Taktung ($\frac{1}{2}$P), korrekte Zuweisung ($\frac{1}{2}$P).

Lösungsalternativen:

- `falling'edge(clk)` statt `clk'event and clk='0'`
- Statt `x<=a and b` folgendes Konstrukt

```
if a='1' and b='1' then
  x<='1';
else
  x<='0';
end if;
```

(e.2) Die Signale `a` und `b` müssen [bidirektional|Eingang] ($\frac{1}{2}$P) sein, d.h. [`inout`|`in`] ($\frac{1}{2}$P). *1P*

(e.3) Es gilt aufgrund der beschriebenen Umstände $y = x_{t-1}$. *1P*

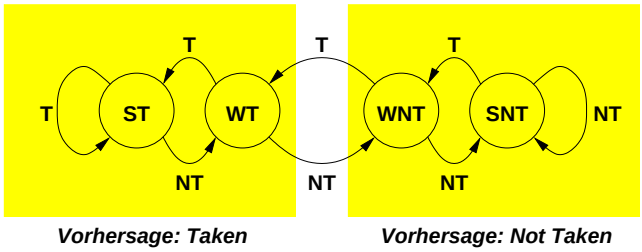
Musterlösung 3: Prozessorarchitektur *10P*

a) Anlaufen/Leerlaufen der Pipeline sowie Konflikte. **1P**

b) Schleifen ($\frac{1}{2}$P), diese werden i.a. mehrfach durchlaufen, d.h. in den Schleifenkörper zurückgesprungen ($\frac{1}{2}$P). **1P**

c) **2P**

(c.1) *1P*



(c.2) Einführung von Hysterese ($\frac{1}{2}$P), verhindert Denaturation zu einem 1-Bit-Prädiktor, d.h. Flattern zwischen Weakly-Zuständen ($\frac{1}{2}$P). *1P*

d) **2P**

(d.1) Diese Prädiktoren erzielen (verglichen mit anderen einfacheren dynamischen Prädiktoren) eine bessere Durchmischung der Indizierung der Verlaufstabelle ($\frac{1}{2}$P). Dies erzielt eine Verringerung von Interferenz ($\frac{1}{2}$P). *1P*

(d.2) *1P*

Adressteil	BHR	gselect 4/4	gshare 8/8
00000001	10000000	00000000	10000001
10100101	00000011	10100011	10100110

e) *(je $\frac{1}{2}$P pro korrekter Teilantwort)* **3P**

- Issue: Holen der Befehle (aus Instruction Queue) und Operanden
- Ausführung/Execute: Anstoßen (Dispatch) der ausführungsbereiten Befehle mit Konfliktüberprüfung.
- Rückschreibphase: Schreiben der Ergebnisse auf Ergebnisbus und Kennzeichnen des Umordnungspuffers als verfügbar.

f) **1P**

- Control Speculation (Spekulative Befehlsausführung, ld.s, chk.s) *0,5P*
- Data Speculation (Vorgezogene Ladeoperationen, ld.a, chk.a) *0,5P*

Musterlösung 4: Parallelverarbeitung *10P*

Parallelisierung und Parallelverarbeitung: *3P*

a) Vier Klassen von Rechnerarchitekturen: ($\frac{1}{2}$P) **1P**

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

Schwachpunkt: Heutige Prozessoren (mit modernen Parallelarbeitstechniken wie Superskalarität, VLIW,...) lassen sich nicht oder nur schwer einordnen ($\frac{1}{2}$P).

b) Schritte bei der Parallelisierung: **2P**

- Aufteilung oder Dekomposition
- Zuweisung
- Festlegung (Orchestration)
- Abbildung

(Jeweils $\frac{1}{2}$P pro korrekter Antwort)

Quantitative Maßzahlen: *1P*

c) Ausführungszeit eines parallelen Programms **1P**

$$T = T_{comp} + T_{comm} + T_{idle}$$

T_{comp}	Berechnungszeit
T_{comm}	Kommunikationszeit
T_{idle}	Untätigkeitszeit

($\frac{1}{2}$P für Formel, $\frac{1}{2}$P für Erklärung)

Verbindungsnetze: *4P*

d) (1P) In Richtung Wurzelknoten kommt es zu einem Blockierungsproblem. Die Kommunikationskanäle am Wurzelknoten werden sehr stark ausgelastet, es kommt zu einem Flaschenhals. **2P**

(1P) Lösung in der Praxis: Fat Tree, d.h. die Kommunikationskanäle werden größer, je näher man sich der Wurzel nähert.

e) **2P**

- Statische und dynamische Verbindungsnetze *1P*
- 2×2 Kreuzschienenverteiler als Grundelement *($\frac{1}{2}$P)* *1P*

Multiprozessoren: *2P*

f) Gültigkeitsproblem **1P**

- Mehrere Kopien des gleichen Speicherwortes müssen miteinander in Einklang gebracht werden
- Die Kohärenz muß sichergestellt werden.

g) Verkettung *($\frac{1}{2}P$)* **1P**

- Verkettung mehrerer (spezialisierter) Pipelines
- Die Ergebnisse einer Pipeline werdern sofort der nächsten Pipeline zur Verfügung gestellt

($\frac{1}{2}P$ für Erklärung)

Musterlösung 5: Speicherhierarchie *10P*

a) $t_a = r_{H(L1)} * t_{H(L1)} + (1 - r_{H(L1)}) * (r_{H(L2)} * t_{H(L2)} + (1 - r_{H(L2)}) * t_{Mem})$ **1P**

b) (1P) Alternative A: $t_a = 70\,\% * 10\,ns + 30\,\% * (50\,\% * 30ns + 50\,\% * 100\,ns) = 26.5\,ns$ **2.5P**

(1P) Alternative B: $t_a = 80\,\% * 12\,ns * 20\,\% * (30\,\% * 25ns + 70\,\% * 100\,ns) = 25.1\,ns$

($\frac{1}{2}$P) Entwurfsalternative B ist zu wählen, da hier das System eine geringere mittlere Zugriffszeit aufweist.

c) Bei Write-Invalidate-Protokollen werden bei Veränderung einer Kopie alle andere Kopien des Datums invalidiert (1P); bei Write-Update-Protokollen werden sie auf den neuesten Wert aktualisiert (1P). **2P**

d) *($\frac{1}{2}$P Abzug pro Fehler)* **3.5P**

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

e) MOESI vermeidet durch den Zustand *modified* bedingte Hauptspeicherzugriffe ($\frac{1}{2}$P). Stattdessen würden die Daten direkt vom (Owner-)Cache zum lesenden Cache kopiert ($\frac{1}{2}$P). **1P**

Musterlösung 6: Fehlertoleranz *10P*

a) ($\frac{1}{2}$P) Ausfallrate λ **1P**
($\frac{1}{2}$P) $MTBF = \frac{1}{\lambda}$

b) Für die Mean Time Between Failure ($\frac{1}{2}$P) gilt $MTBF = MTTF + MTTR$ ($\frac{1}{2}$P) bei konstantem λ ($\frac{1}{2}$P), d.h. die Benutzung von MTBF beschränkt sich auf die Betriebsphase ($\frac{1}{2}$P). **2P**

c) Der Knoten sendet entweder korrekte Nachrichten oder keine Nachrichten, jedoch nie falsche ($\frac{1}{2}$P). Bei einem Fail-Stop-System werden aufgrund eines aufgetretenen Fehlers gar keine Nachrichten mehr versandt ($\frac{1}{2}$P). **1P**

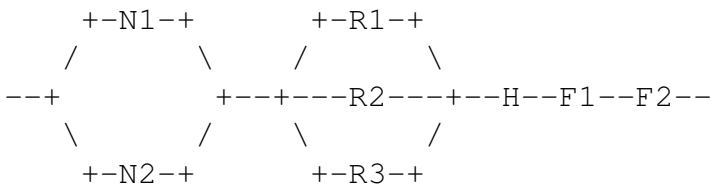
d) **3P**

- Cold Standby: keine Alterung durch Nutzung, aber lange Übergabezeit im Fehlerfall *0,5P*
- Hot Standby: schnelle Übergabezeit, aber betriebsbedingte Alterung (sowie Energieaspekte) *0,5P*
- Graceful Degradation: Umverteilung der Last auf verbleibende Knoten ($\frac{1}{2}$P), dadurch bleibt das System bleibt betriebsfähig, jedoch bei verringerter Leistungsfähigkeit ($\frac{1}{2}$P). *1P*
- $\Phi_n^m = \sum_{k=n}^{m} \Phi(K)^k * (1 - \Phi(K))^{m-k}$ *1P*

Hinweis: Bei Cold/Hot Standby sind neben der Failoverzeit die Alterungseffekte wesentlich, nicht die Energieaspekte – ein Hot-Standby-System verschleißt durch Benutzung, das Cold-Standby-System weist jedoch nur eine Alterung wie durch Lagerung auf.

e) ($\frac{1}{2}$P) Ausfallwahrscheinlichkeit des Gesamtsystems $\Psi(R_{1,2,3}) = (1 - \Phi(R))^3$ **1P**
($\frac{1}{2}$P) Funktionswahrscheinlichkeit des Gesamtsystems: $\Phi(R_{1,2,3}) = 1 - \Psi(R_{1,2,3})$

f) (1P) Zuverlässigkeitsblockdiagramm **2P**



Hinweis für alle zukünftigen Lerner: Das obige Diagramm ist in ASCII-Darstellung. Sie dürfen das in Klausuren gerne ganz normal zeichnen und müssen für die Verbindungslinien keine einzelnen −, +, / und \ verwenden...

(1P) Systemformel $S = (R_1 \vee R_2 \vee R_3) \wedge (N_1 \vee N_2) \wedge H \wedge F_1 \wedge F_2$