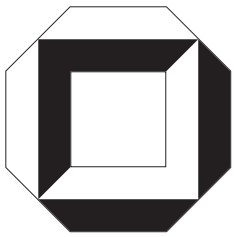


Universität Karlsruhe (TH)
Institut für Technische Informatik
Prof. Dr. Wolfgang Karl

Klausur Rechnerstrukturen
Wintersemester 2006/07

19. März 2007

Musterlösung

Aushang der Ergebnisse: ab wird auf der RS-Website bekanntgegeben

Musterlösung 1: Fertigung und Sprungvorhersage *12P*

a) ($\frac{1}{2}$P) $dpw = \frac{\pi(d_{wafer} * \frac{1}{2})^2}{a_{die}} - \frac{\pi * d_{wafer}}{\sqrt{2 * a_{die}}}$ *1,5P*

($\frac{1}{2}$P) $dpw_{k=2cm} = \frac{\pi(20 * \frac{1}{2})^2}{(2cm)^2} - \frac{\pi * 20}{\sqrt{2 * (2cm)^2}} = 25\pi - \frac{10\pi}{\sqrt{2}} (\simeq 18)$

($\frac{1}{2}$P) $dpw_{k=1cm} = \frac{\pi(20 * \frac{1}{2})^2}{(1cm)^2} - \frac{\pi * 20}{\sqrt{2 * (1cm)^2}} = 100\pi - \frac{20\pi}{\sqrt{2}} (\simeq 86)$

b) Durch die Verkleinerung der Die-Größe wird anteilig weniger Verschnitt erzielt ($\frac{1}{2}$P). Wie man anhand der Berechnungsformel erkennen kann, liegt dies darin begründet, dass (bei konstanter Die-Größe) der Verschnitt lediglich mit der Quadratwurzel der Flächenvergrößerung zunimmt ($\frac{1}{2}$P). *1P*

c) ($\frac{1}{2}$P) $yield_{die} = yield_{wafer} * (1 + \frac{dpua * a_{die}}{\alpha})^{-\alpha}$ *1,5P*

($\frac{1}{2}$P) $yield_{k=2cm} = 1 * (1 + \frac{0.5 * (2cm)^2}{1})^{-1} = 1 * (1 + 2)^{-1} = 1 * \frac{1}{3} (= 0,333)$

($\frac{1}{2}$P) $yield_{k=1cm} = 0,75 * (1 + \frac{0.75 * (1cm)^2}{1})^{-1} = \frac{3}{4} * (1 + \frac{3}{4})^{-1} = \frac{3}{4} * \frac{4}{7} = \frac{3}{7} (= 0,430)$

d) ($\frac{1}{2}$P) $cost_{die} = \frac{cost_{wafer}}{dpw * yield_{die}}$ *1P*

($\frac{1}{2}$P) Die Kosten verringern sich um einen Faktor $5 * \frac{5}{2} = 12,5$.

e) *1,5P*

Die statische Annahme *Sprünge zurück werden genommen, Sprünge nach vorne nicht* (alternativ aufgrund Aufgabenstellung: *always not taken*) erreicht die geforderte Qualität, da lediglihch die Abbruchkriterien falsch vorhergesagt werden. (1P). Anders als beim 1-Bit-Prädiktor tritt jedoch kein Kippen des Vorhersageergebnis und damit verbundene Fehlvorhersage bei der nächsten Vorhersage ein ($\frac{1}{2}$P).

f) *(je korrekte Zeile 1P, je Fehler $\frac{1}{2}$P Abzug)* *4,5P*

	S1				S2			
Durchlauf	Init	Vhs.	Sprung	Präd. neu	Init	Vhs.	Sprung	Präd. neu
1	(**NT**, T)	NT	T	(**T**, T)	(NT,**NT**)	NT	T	(NT,**T**)
2	(T, **T**)	T	NT	(T,**NT**)	(**NT**, T)	NT	T	(**T**, T)
3	(T, **NT**)	NT	T	(T,**T**)	(T, **T**)	T	NT	(T, **NT**)
4	(**T**, T)	T	NT	(**NT**, T)	(**T**, NT)	T	NT	(**NT**, NT)

($\frac{1}{2}$P) Die sinnvollste Initialisierung für S1 wäre aufgrund des aufgetretenen Sprungmusters von S1 und S2 (T,NT).

g) Selektionsprädiktor (zur Auswahl des Vorhersageprädiktors) und Vorhersageprädiktor (zur eigentlichen Sprungvorhersage). *1P*

Musterlösung 2: Hardwareentwurf *14 P*

a) Rücksetzbarer ($\frac{1}{2}$P) Abwärtszähler ($\frac{1}{2}$P) mit dem Wertebereich [15:0] ($\frac{1}{2}$P) *1,5P*

b) Der Zähler lässt sich bei Zusammenfallen der Taktflanke und Rücksetzsignal ($\frac{1}{2}$P) nicht auf den Initialwert setzen ($\frac{1}{2}$P). Dies liegt darin begründet, dass die Abfrage auf Rücksetzung dem eigentlichen Zählvorgang nachgeschaltet ist ($\frac{1}{2}$P). Für eine korrekte Abfolge sind die if- und elsif-Zweige zu vertauschen ($\frac{1}{2}$P), ausgeschrieben: *2P*

```
process(clk,rst)
begin
  if rst='0' then
    count<=(others>='1');
  elsif clk'event and clk='1' then
    count<=count-1;
  end if;
end process;
```

(Keine gesonderte Wertung, falls letzte Teilantwort enthalten.)

c) *2P*

```
entity counter is
  port(
    clk, rst: in bit;
output: out unsigned(3 downto 0)
  );
end entity;
```

(Je 1P pro korrekter Signalgruppe, $\frac{1}{2}$PAbzug pro semantischem Fehler.)

d) A wird der Wert 2 mit einer Verzögerung von 5ns zugewiesen ($\frac{1}{2}$P). Es handelt sich um eine Simulationsanweisung ($\frac{1}{2}$P). Es kann lediglich der reine Zuweisungsteil synthetisiert werden, nicht aber die Verzögerung. *($\frac{1}{2}$P für unvollständige Zuweisung, $\frac{1}{2}$P für Angabe des Teils.)* *2P*

e) Es werden 9 Aus- und 7 Eingänge benötigt ($\frac{1}{2}$P). Es sind keine Speicherglieder erforderlich ($\frac{1}{2}$P), weswegen sowohl 22L10 als auch 22V10 verwendet werden können ($\frac{1}{2}$P). Es werden nur Logikressourcen benötigt (keine Speicherglieder), d.h. ausreichend im Sinne der Aufgabenstellung ist 22L10 ($\frac{1}{2}$P*nur bei kompletter Antwort*). *2P*

f) Es werden zwei Eingänge ($\frac{1}{2}$P) und $2^2 = 4$ Ausgänge ($\frac{1}{2}$P) benötigt, um alle möglichen Zustände abzudecken. Hierdurch lassen sich alle zweiwertigen Funktionen bei konstantem Flächenbedarf ($\frac{1}{2}$P) und konstanter Durchlaufzeit ($\frac{1}{2}$P) erzeugen. *2P*

g) ($\frac{1}{2}$P) Verhaltensebene (behavioral) zur Systemspezifikation oder Standardmodelle *2,5P*
($\frac{1}{2}$P) Register-Transfer-Level (RTL) für synthetisierbare Modelle in zielunabhängiger Hardwarebeschreibung
($\frac{1}{2}$P) Gatterebene (Gate/Logic Level) beim Entwurf mit einfachen PLDs
($\frac{1}{2}$P) Layout-Ebene für Full-custom Design

($\frac{1}{2}$P) Die Layout-Ebene wird beim Entwurf mit FPGAs nicht berührt und spielt daher für den Entwickler keine Rolle.

Musterlösung 3: Leistungsbewertung *11P*

a) Zugriffs- und Übertragungszeit (je $\frac{1}{2}$P) *1P*

b) (1P) Herleitung: $U = \frac{D}{D_{max}} = D * X$ mit $D = A$, d.h. $X = \frac{U}{D}$ *2P*
(1P) Berechnung: $X = \frac{0{,}8}{40/s} = (\frac{4}{5} * \frac{1}{40})s = \frac{1}{50}s = 20ms$

Die erforderliche Mindestzugriffszeit bei einer Maximalauslastung von 80% und einer durchschnittlichen Ankunftsrate A=40/s von Aufträgen beträgt 20ms.

c) ($\frac{1}{2}$P) Es liegt das Gesetz von Little zugrunde. *2P*
($\frac{1}{2}$P) $W = \frac{Q}{D}$ mit D=A $\leftrightarrow D = \frac{Q}{A}$
(1P) $D = \frac{4}{0.1s} = \frac{40}{1s} = 40/s$

Dies ist exakt für eine Ankunftsrate A=40/s der Fall.

d) ($\frac{1}{2}$P) $s = \frac{t_{seq}}{t_{par}}$
($\frac{1}{2}$P) $t_{seq} = 2n^2$
($\frac{1}{2}$P) $t_{par} = \frac{n^2}{P} + n^2$
($\frac{1}{2}$P) $s = \frac{2n^2}{\frac{n^2}{P}+n^2} = \frac{2P}{P+1}$ *2P*

Hinweis: Aus der Aufgabenstellung geht nicht hervor, dass bereits lokale Aufsummationen erfolgen. Es wird lediglich gesagt, dass die lokal berechneten Werte in eine globale Summe akkumuliert werden. Daher gilt für t_{par} der angegebene Fall.

e) $lim_{N,P\to\infty}\frac{2P}{P+1} = 2$, es lassen sich keine Beschleunigungswerte größer 2 erzielen ($\frac{1}{2}$P); *1,5P*
somit liegt der sequentielle Anteil der Berechnung bei 50% ($\frac{1}{2}$P) und begrenzt gemäß Amdahl's Law den Speedup auf 2 ($\frac{1}{2}$P).

f) ($\frac{1}{2}$P) Gesetz von Amdahl: $T(n) = T(1) * (\frac{1-a}{n} + a)$ *1,5P*

Aus der Aufgabenstellung heraus ergibt sich $n \to \infty$ und $S(n) = \frac{T(1)}{T(n)} = 4$ und es gilt somit:

($\frac{1}{2}$P) $\frac{1-a}{n} = 0$, d.h. $\frac{1}{S(n)} = a$.
($\frac{1}{2}$P) Somit ergibt sich $a = \frac{1}{4} = 25\%$.

Der sequentielle Codeanteil beträgt also 25%.

g) ($\frac{1}{2}$P) $E(n) = \frac{S(n)}{n} = \frac{4}{128} = \frac{1}{32}$ *1P*
Die Effizienz beträgt für n=128 Knoten $\frac{1}{32}$, also geringfügig mehr als 3%.
Für die Beschleunigung gilt allgemein $1 \leq S(n) \leq n$ ($\frac{1}{2}$P) bzw. (alternativ, wg. Aufgabenstellung) $S(n) = \frac{T(1)}{T(n)}$ bzw. $S(n) = n * E(n)$.

Musterlösung 4: Kommunikation und Kohärenz *14P*

a) Der Datenaustausch über Knotengrenzen hinweg geschieht explizit über Kommunikation ($\frac{1}{2}$P) anstatt über gemeinsame Speicherbereiche ($\frac{1}{2}$P). *1P*

b) Verzeichnis- (directory-based) bzw. tabellenbasierte Protokolle ($\frac{1}{2}$P); hierbei wird in einer Verzeichnistabelle für jeden Blockrahmen im lokalen Speicher vermerkt ($\frac{1}{2}$P), ob dieser in den lokalen oder einen entfernten Cache-Speicher als Cache-Block übertragen worden ist ($\frac{1}{2}$P). Die Zustände werden als Kopien festgehalten ($\frac{1}{2}$P). *2P*

c) *(je Fehler $\frac{1}{2}$PAbzug)* *4P*

		Prozessor/Cache 1		Prozessor/Cache 2	
Prozessor	Aktion	Line 1	Line 2	Line 1	Line 2
-	(init)	I/-	S/2	S/2	E/4
1	wr 2		M/2	I/-	
1	rd 4	S/4			S/4
2	rd 8			E/8	
2	rd 10				E/10
2	rd 2		S/2	S/2	
1	wr 6	M/6			
2	wr 2		I/-	M/2	
1	rd 6				

d) Einfache Bäume weisen ein Blockierungsproblem in Richtung Wurzel auf ($\frac{1}{2}$P); dies wird im sogenannten Fat Tree ($\frac{1}{2}$P) dadurch adressiert, dass die Kommunikationskanäle größer/leistungsfähiger werden, je mehr man sich der Wurzel nähert ($\frac{1}{2}$P). Es handelt sich bei diesen Strukturen um statische Verbindungsnetze ($\frac{1}{2}$P). *2P*

e) Das Omega-Netzwerk besteht aus $p = 2{*}n$ Ein-/Ausgängen ($\frac{1}{2}$P) und umfasst $n = ldp$ Stufen ($\frac{1}{2}$P) von Zweierschaltern. Diese sind untereinander nach dem Grundmuster der Mischpermutation ($\frac{1}{2}$P) verknüpft. Es werden $\frac{p}{2} * ldp$ Zweierschalter benötigt ($\frac{1}{2}$P). *2P*

f) ($\frac{1}{2}$P) Anzahl Knoten: $N = K^n$ *1,5P*
($\frac{1}{2}$P) Knotengrad: $2 * n$
($\frac{1}{2}$P) Diameter: $n * \lfloor \frac{k}{2} \rfloor$

g) ($\frac{1}{2}$P) $W = A \oplus B = (001) \oplus (111) = (110)$, es müssen also 2 Dimensionen traversiert werden. *1,5P*
($\frac{1}{2}$P) Pfad 1: $(001) \rightarrow (011) \rightarrow (111)$
($\frac{1}{2}$P) Pfad 2: $(001) \rightarrow (101) \rightarrow (111)$

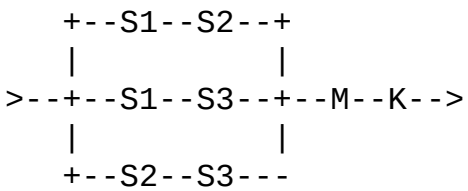
Musterlösung 5: Fehlertoleranz *9P*

a) ($\frac{1}{2}$P) $MTTF$: Mean time to failure (mittlere Dauer bis zum Ausfall) *1P*
($\frac{1}{2}$P) $MTTR$: Mean time to repair (mittlere Reparaturdauer)

b) ($\frac{1}{2}$P) $MTBF = MTTF + MTTR$ *1,5P*
($\frac{1}{2}$P) $V = \frac{MTTF}{MTTF+MTTR} = \frac{MTTF}{MTBF}$
($\frac{1}{2}$P) In Realsystemen gilt typischerweise $MTTR \ll MTTF$ und damit $MTBF \simeq MTTF$.

c) ($\frac{1}{2}$P) $MTTF = \frac{1}{\lambda} \leftrightarrow \lambda = \frac{1}{MTTF}$ *1,5P*
λ ist bei dieser Betrachtungsweise konstant ($\frac{1}{2}$P) und daher nur auf die Betriebsphase anwendbar ($\frac{1}{2}$P).

d) Die Zuverlässigkeit wird dominiert durch die Kommunikationseinheit K, daher gilt: *1P*



e) ($\frac{1}{2}$P) $S = ((S_1 \wedge S_2) \vee (S_1 \wedge S3) \vee (S_2 \wedge S_3))$ *1P*
($\frac{1}{2}$P) $\wedge M \wedge K$

f) Es handelt sich um ein 2-aus-3-(Redundanz)System ($\frac{1}{2}$P), daher gilt: *1P*

($\frac{1}{2}$P)
$$\Phi(E) = \sum_{k=2}^{3} \binom{3}{n} * \Phi(S)^k * (1 - \Phi(S))^{3-k}$$

g) ($\frac{1}{2}$P) Für die Ausfallwahrscheinlichkeit gilt $\Psi(R) = 1 - \Phi(R)$. *1P*
($\frac{1}{2}$P) Für die Funktionswahrscheinlichkeit gilt: $\Phi(R) = \Phi(E) * \Phi(M) * \Phi(K)$

oder direkt: $\Psi(R) = 1 - (\Phi(E) * \Phi(M) * \Phi(K))$.

h) Gemäß der Vorgaben nicht: Auch im veränderten Szenario werden mindestens zwei intakte Sonnensegel zum Betrieb der Sonde benötigt. *1P*