



Karlsruher Institut für Technologie

Institut für Technische Informatik

Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung

Prof. Dr. rer. nat. Wolfgang Karl

Klausur Rechnerstrukturen

Sommersemester 2015

Aufgabenteil

5. August 2015

Aufgabe 1: Verbindungsstrukturen und Fehlertoleranz 10 P

Verbindungsstrukturen 6 P

- a) Aus welchen Bestandteilen setzt sich die Übertragungszeit T_{msg} einer Nachricht zusammen? 1 P
- b) Zeichnen Sie das Grundmuster der Mischpermutation mit 8 Eingängen und 8 Ausgängen auf. Verwenden Sie hierfür als Hilfe die Vorgaben auf den Lösungsblättern. 1 P
- c) Was ist die Bisektionsbandbreite eines Netzwerks? 1 P
- d) Nennen Sie jeweils zwei statische und dynamische Verbindungsstrukturen. 2 P
- e) Nennen Sie zwei Vermittlungstypen des Datentransfers für Verbindungsstrukturen. 1 P

Fehlertoleranz 4 P

Gegeben sei ein elektronisches Gerät bestehend aus einem Mikrocontroller M , zwei Kommunikationsbausteinen $K1$ und $K2$ und zwei Speichern $S1$ und $S2$. Zum korrekten Betrieb des Geräts werden der Mikrocontroller und jeweils mindestens ein Kommunikationsbaustein und ein Speicher benötigt.

- f) Zeichnen Sie das Zuverlässigkeitsblockdiagramm des elektronischen Geräts. 1 P
- g) Ermitteln Sie die Systemfunktion S des elektronischen Geräts und zeichnen Sie außerdem den Strukturbaum für das elektronische Gerät. 2 P
- h) Geben Sie die allgemeine Formel für die Berechnung der Zuverlässigkeit φ_m^n eines n-aus-m-Systems an. 1 P

Aufgabe 2: Low-Power-Entwurf und Leistungsbewertung

10 P

Low-Power-Entwurf

5 P

Ein im Jahre 1998 entworfener Prozessor wurde mit einer Kernspannung von 2,0 V bei einer Frequenz von 800 MHz betrieben. 14 Jahre später liegt die Kernspannung eines Prozessors des gleichen Herstellers bei 1,0 V und die zugehörige Frequenz bei 4,0 GHz. Die Kapazität C_{eff} blieb über diesen Zeitraum konstant.

- Zeigen Sie auf, wie sich die aufgenommene elektrische Schaltleistung $P_{switching}$ innerhalb dieses Entwicklungssprungs verändert hat. 2 P
- Welchen Zusammenhang gibt es zwischen Versorgungsspannung und maximal möglicher Frequenz? Wie lässt sich dieser Zusammenhang erklären? 2 P
- Geben Sie die allgemeine Formel zur Berechnung der elektrischen Leistungsaufnahme P_{total} bei CMOS-Schaltungen an. 1 P

Leistungsbewertung

5 P

Ein Rechner mit der oben angegebenen CPU aus dem Jahr 2012 soll nun einer Leistungsbewertung unterzogen werden.

- Zur Laufzeitmessung wird ein Benchmark X auf der gegebenen Architektur ausgeführt. Während die Abarbeitung dieses Benchmarks auf einem Referenzsystem 10,5 Sekunden benötigte, wird auf der aktuellen Architektur eine Ausführungszeit von 4,2 Sekunden erreicht. Berechnen Sie die $SPEC_{ratio_X}$. 1 P
- Sie haben für eine Programmausführung auf der gegebenen Rechnerarchitektur die in der Tabelle aufgeführten Werte experimentell bestimmt. Berechnen Sie mit diesen Werten den MFLOPS-Wert für dieses Programm und die gegebene Architektur. 2 P

Floatingpointoperation	Anzahl Operationen	Zyklenzahl Operation
Operation 1	$0,3 \cdot 10^6$	7
Operation 2	$0,5 \cdot 10^6$	5
Operation 3	$0,2 \cdot 10^6$	12
Operation 4	$1,0 \cdot 10^6$	3

- Nennen und erläutern Sie zwei weitere Maße, die zur Leistungsbewertung einer gegebenen Rechnerarchitektur herangezogen werden können. 2 P

Aufgabe 3: Quantitative Maßzahlen

10 P

Quantitative Maßzahlen

10 P

Zur Ausführung eines gegebenen Algorithmus stehen zwei Rechensysteme zur Auswahl und Sie sollen entscheiden, welche der Architekturen und zugehörigen Implementierungen des Algorithmus am besten geeignet ist.

Auf System A sind $\frac{4}{5}$ des Algorithmus parallel ausführbar mit 16 Kernen. Bei der parallelen Ausführung werden 480 Einheitsoperationen ausgeführt. Die Rechenzeit des gesamten Algorithmus beträgt bei sequentieller Ausführung 2 Minuten.

Auf System B sind $\frac{2}{3}$ des Algorithmus parallel ausführbar mit 10 Kernen. Bei der parallelen Ausführung werden 280 Einheitsoperationen ausgeführt. Die Rechenzeit des gesamten Algorithmus beträgt bei sequentieller Ausführung 1 Minute, 40 Sekunden.

- a) Berechnen Sie für beide Systeme die Rechenzeit bei paralleler Ausführung mittels der Formel aus Amdahls Gesetz, sowie die Beschleunigung S , Effizienz E , Parallelindex I und Auslastung U . 5 P
Welches System würde den gegebenen Algorithmus am schnellsten ausführen?
- b) Welche Beschleunigung wäre auf System A maximal möglich, wenn die Anzahl der Kerne beliebig erhöht werden könnte? 1 P
- c) Welches System würden Sie wählen, wenn der Ausfall eines CPU-Kerns ohne Leistungsverlust tolerierbar sein soll? Begründen Sie. 1 P
- d) Als weitere Alternative wird überlegt, System B mit einem anwendungsspezifischen Spezialprozessor zu erweitern. Die Hälfte des nicht-parallelisierbaren Anteils des Algorithmus kann auf diesem Prozessor 10 mal schneller ausgeführt werden als auf einem normalen CPU-Kern. Berechnen Sie die Rechenzeit auf diesem System. 2 P
- e) Geben Sie das allgemeine theoretische Maximum für die Effizienz eines Systems an. Kann dieser Wert in der Praxis überschritten werden? Falls ja, unter welchen Umständen? 1 P

Aufgabe 4: Hardware-Entwurf und Parallele Architekturen 10 P

Parallele Architekturen 3 P

- a) Nennen Sie zwei verschiedene Speicherarchitekturen für Multiprozessoren und geben Sie je ein passendes Programmiermodell mit an. 2 P
- b) Geben Sie die Klassifikation nach Flynn für Architekturen an. 1 P

Hardware-Entwurf 7 P

- c) Nennen Sie drei Entwurfsschritte, die üblicherweise bei einem VHDL-Entwurf durchgeführt werden. 1.5 P
- d) Der Entwurf in VHDL sieht verschiedene Hauptbestandteile vor. Nennen Sie diese. 1.5 P
- e) Geben Sie eine Beschreibung in VHDL an, die folgende Funktionalität realisiert: Ein Zähler wird alle fünf Takte bis zu einem maximalen Wert um eins erhöht. Dabei wird der maximale Wert über eine externe Schnittstelle gesetzt. In jeder vierten Erhöhung wird für zwei Erhöhungen ein Ausgangssignal gesetzt. Der Ausgang wird auf 0 beim Erreichen des maximalen Werts gesetzt. 4 P

Vervollständigen Sie die gegebene Codeschablone und achten Sie dabei auf eine sinnvolle Benennung der Bezeichner!

Hinweis: Die Anweisung (`others => '0'`) bedeutet, dass alle Werte eines Vektors auf 0 gesetzt werden.

Aufgabe 5: Speicherhierarchie

10 P

Cache-Kohärenzprotokoll

6 P

Ein Vierprozessorsystem sei speichergekoppelt. Die Caches haben je eine Größe von zwei Cache-Zeilen, welche je genau ein Speicherwort aufnehmen können. Die Füllung des Caches erfolgt von der niedrigsten Cache-Zeile aufwärts, sofern noch freie Zeilen zur Verfügung stehen, andernfalls wird gemäß LRU-Strategie verdrängt. Als Cache-Kohärenzprotokoll komme das aus der Übung bekannte MESI-Protokoll zum Einsatz. Der Cache sei initial leer. Aktionen, die durch das Cache-Kohärenzprotokoll ausgelöst werden und die eine Zustandsänderung einer Cache-Zeile bewirken, werden von der LRU-Strategie nicht als Zugriff gewertet.

- a) Vervollständigen Sie die auf dem Lösungsblatt angegebene Tabelle: Geben Sie jeweils Inhalt der Cache-Zeile und MESI-Zustand an. 4 P
- b) Welchen allgemeinen Vorteil bringt der Zustand O des MOESI-Protokolls mit sich? 2 P

Cache-Leistung

4 P

Sie sollen prüfen, ob es vorteilhaft wäre eine bestehende Speicherhierarchie mit einer durchschnittlichen Antwortzeit von 12 ns mit einer von zwei neuen Speicherhierarchien (Variante A und Variante B) zu ersetzen. In Variante A findet der Zugriff auf die nächste Ebene parallel und in Variante B sequentiell statt.

	Variante A	Variante B
Zugriffszeit L1	8 ns	3 ns
Hitrate L1	90 %	80 %
Zugriffszeit L2	30 ns	17 ns
Hitrate L2	60 %	75 %
Zugriffszeit Hauptspeicher	100 ns	100 ns

- c) Um eine fundierte Aussage treffen zu können, berechnen Sie die durchschnittliche Antwortzeit für beide Varianten. Ist eine Ersetzung sinnvoll und wenn ja, welche Variante wählen Sie? 2 P
- d) Was besagt die Inklusionseigenschaft bei hierarchischen Cache-Speichern? 1 P
- e) Was ist der Unterschied zwischen einem direkt abgebildeten und einem assoziativen Cache? 1 P

Aufgabe 6: Tomasulo und Vektorverarbeitung

10 P

Algorithmus von Tomasulo

5 P

- a) Untenstehend finden Sie den Zustand der Reservierungstabelle und der Registerdatei eines Superskalarprozessors nach Abarbeitung des ersten Taktes der in Listing 1 dargestellten Befehlsfolge. Geben Sie den Zustand der Reservierungstabelle, sowie der Registerdatei nach Ablauf von Takt 4, d.h. nach drei weiteren Takten, unter Berücksichtigung der in Listing 1 dargestellten Befehlsfolge wieder. Pro Takt kann ein Befehl in die Reservierungstabelle eingetragen werden. Eine Addition benötigt 2 Takte, eine Multiplikation 6 Takte und eine Division 9 Takte.

5 P

Takt	Befehlsfolge
1	add R4, R1, R3
2	div R3, R2, R4
3	add R2, R1, R2
4	mul R1, R2, R3

Listing 1

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

Feld	R1	R2	R3	R4
Value	(R1)	(R2)	(R3)	–
Valid	1	1	1	0
RS	–	–	–	Add/Sub 1

Registerdatei

Unit	Empty	InFU	Op	Dest	Src1	Vld1	RS1	Src2	Vld2	RS2
Add/Sub 1	0	0	add	R4	(R1)	1	–	(R3)	1	–
Add/Sub 2	1									
Mul 1	1									
Div 1	1									

Reservierungstabelle

Vektorverarbeitung**5 P**

- b) Vervollständigen Sie das Code-Fragment auf dem Lösungsblatt mittels Vektorbefehlen in Assembler um das folgende Programm zu realisieren: 4 P

```
unsigned char i;
unsigned char a[64], b[64], c[64];
for (i = 0; i < 64; i++) {
    c[i] = a[i];
    if (a[i] == 0xff) {
        c[i] = b[i];
    }
}
```

- c) Vektorbefehle werden oft durch ein spezielles Speichersystem mit Verschränkung (memory interleaving) und mehreren Speicherbänken unterstützt. Wie lange dauert ein Ladebefehl eines 16-elementigen Vektors bei 4 Speicherbänken und einer Latenz von 2 Zyklen 1 P
- mit einem Stride von 4,
 - mit einem Stride von 1?



Karlsruher Institut für Technologie

Institut für Technische Informatik

Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung

Prof. Dr. rer. nat. Wolfgang Karl

Klausur Rechnerstrukturen

Sommersemester 2015

Lösungsteil

Name: _____

Vorname: _____

Matrikelnummer: _____

Tragen Sie bitte auf jedem Blatt Ihren Namen und Ihre Matrikelnummer ein. Bitte tragen Sie alle Lösungen und Rechenwege an den vorgesehen Stellen ein und geben Sie keine zusätzlichen Blätter ab, ohne dies dem Aufsichtspersonal mitzuteilen.

Hinweis: Bei Rechenaufgaben ist die Angabe des Rechenwegs zwingend erforderlich. Ergebnisse ohne Rechenweg werden **nicht** gewertet.

Zum Bestehen der Klausur sind mindestens 20 Punkte erforderlich.

() Ich wünsche **keine** Notenveröffentlichung über einen anonymisierten Code auf der Website des Lehrstuhls.

(Bei Ankreuzen kann die Note erst in der Klausureinsicht erfragt werden.)

Erreichte Punkte (wird vom Institut ausgefüllt):

Aufgabe	1	2	3	4	5	6
Punkte	/10	/10	/10	/10	/10	/10
Summe:						/60

Aufgabe 1: Verbindungsstrukturen und Fehlertoleranz

10 P

Verbindungsstrukturen

6 P

a) Übertragungszeit:

1 P

b) Mischpermutation:

1 P

0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	1	0	0
0	1	1	0	1	1	1
1	0	0	1	0	0	0
1	0	1	1	0	1	1
1	1	0	1	1	1	0
1	1	1	1	1	1	1

c) Bisektionsbandbreite:

1 P

d) Statisch:

2 P

-
-

Dynamisch:

-
-

e) Vermittlungstypen:

1 P

-
-

Fehlertoleranz**4 P**

f) Zuverlässigkeitsblockdiagramm:

1 P

g) Systemfunktion:

2 P

Strukturbaum:

h) Formel Zuverlässigkeit:

1 P

Aufgabe 2: Low-Power-Entwurf und Leistungsbewertung

10 P

Low-Power-Entwurf

5 P

a) Veränderung elektrische Leistung:

2 P

b) Zusammenhang Spannung – Frequenz:

2 P

c) Formel P_{total} :

1 P

Leistungsbewertung

5 P

d) $SPEC_{ratio_X}$:

1 P

e) Berechnung MFLOPS:

2 P

f) Zwei Maße zur Leistungsbewertung:

2 P

-
-

Aufgabe 3: Quantitative Maßzahlen**10 P****Quantitative Maßzahlen****10 P**a) $T_A(n) :$ *5 P* $T_B(n) :$ $S_A(n) :$ $S_B(n) :$ $E_A(n) :$ $E_B(n) :$ $I_A(n) :$ $I_B(n) :$ $U_A(n) :$ $U_B(n) :$

Schnellstes System:

b) Maximale Beschleunigung System A: *1 P*

c) System das Ausfall eines Kerns tolerieren kann: *1 P*

d) Rechenzeit mit Spezialprozessor: *2 P*

e) Theoretisches Maximum der Effizienz: *1 P*

Überschreiten möglich?

Aufgabe 4: Hardware-Entwurf und Parallele Architekturen

10 P

Parallele Architekturen

3 P

a)	<u>Speicherarchitektur</u>	<u>Programmiermodell</u>

2 P

b) Klassifikation nach Flynn:

1 P

-
-
-
-

Hardware-Entwurf

7 P

c) Entwurfsschritte:

1.5 P

-
-
-

d) Hauptbestandteile:

1.5 P

-
-
-

e) _____ counter is

4P

```

Port(
    clk : in std_logic;
    reset : in std_logic;
    _____ : in std_logic(n-1 downto 0);
    _____ : out in std_logic
);
END _____;

_____ of counter is
begin
    count : _____ (clk, reset)

        variable _____ : std_logic_vector(n-1 downto 0);
        variable _____ : std_logic_vector(2 downto 0);
    begin
        if (reset = '1') then
            _____ <= '0';
            _____ := (others => '0');
            _____ := (others => '0');
        elsif (rising_edge(clk)) then

            if (_____ = "00"
                and not(counter_int = (others => '0'))) then
                _____ <= '1';
            elsif (counter_int(1 downto 0) = "10") then
                _____ <= '0';
            end if;

            if (_____ = _____) then
                _____ := (others => '0');
            elsif (_____ )
                _____ := _____ + 1;
            end if;

            if (_____ = "100") then
                _____ := (others => '0');
            else
                _____ := _____ + 1;
            end if;
        end if;
    end _____;
END _____;

```

Aufgabe 5: Speicherhierarchie

10 P

Cache-Kohärenzprotokoll

6 P

a)

4 P

Proz.	Aktion	Proz. 1		Proz. 2		Proz. 3		Proz. 4	
		Zeile 1	Zeile 2	Zeile 1	Zeile 2	Zeile 1	Zeile 2	Zeile 1	Zeile 2
	init	-	-	-	-	-	-	-	-
1	rd 2								
3	rd 5								
4	rd 1								
2	rd 2								
3	wr 1								
4	rd 6								
1	rd 4								
2	rd 5								
4	wr 1								
3	rd 4								

b) Vorteil Zustand O:

2 P

Cache-Leistung**4 P**

c) Berechnung:

2 P

- Variante A:

- Variante B:

- Empfehlung und Begründung:

d) Inklusionseigenschaft:

1 P

e) Unterschied:

1 P

10 P

5 P

 $5P$ [illegible]

Vektorverarbeitung**5 P**

Vektorbefehle, die zur Lösung der Aufgabe zur Verfügung stehen:

```

ADDV V1,V2,V3          # V1 = V2 + V3
ADDVS V1,V2,R1         # V1 = V2 + R1
SUBV V1,V2,V3          # V1 = V2 - V3
SUBVS V1,V2,R1         # V1 = V2 - R1
SEQVS V1,R1            # Vergleiche Elemente in V1 mit R1 und setze VMR
CVM                    # Clear Vector Mask Register (VMR)
MOV R1,#imm            # Schreibe Wert #imm in R1

```

Hinweis: Kommentare zu den Befehlen sind nicht notwendig.

```

b) MOV  R1, 64          # R1 mit 64 initialisieren
    MTC1 VLR, R1        # vector-length register := 64
    LV   V1, Ra          # a[64] in V1 laden
    LV   V2, Rb          # b[64] in V2 laden

```

4 P

```

SV  Rc, V3              # V3 in c[64] speichern

```

c) Stride von 4:

1 P

Stride von 1: