Name: Vorname: Matr.-Nr.:

Aufgabe 1 (10 Punkte)

Geben Sie für die folgenden Aussagen an, ob sie richtig oder falsch sind.

a) Unter Hardware-Interlocking versteht man eine Maßnahme, den Systembus gegen unerlaubte Zugriffe von externen Rechnerkomponenten zu sperren.

☐ richtig ☐ falsch

b) Bei RISC-Prozessoren werden neben dem Befehlsumfang auch die möglichen Adressierungsarten eingeschränkt.

☐ richtig ☐ falsch

c) Unter dem Begriff DMA (Direct Memory Access) versteht man den direkten Zugriff des Prozessors auf den Arbeitsspeicher ohne Umweg über den Cache.

☐ richtig ☐ falsch

d) Wenn die Technik der ungeordneten Befehlsausführung angewandt wird, können superskalare Prozessoren stets pro Taktzyklus zwei oder mehrere nützliche Befehle eines Programms beenden.

☐ richtig ☐ falsch

e) Das Ausgangssignal eines Digital/Analog-Wandlers kann unendlich viele verschiedene (analoge) Spannungswerte annehmen.

☐ richtig ☐ falsch

Name: Vorname: Matr.-Nr.:

Aufgabe 2 (16 Punkte)

Nennen Sie für die im folgenden genannten Paare von Verfahren bzw. Begriffe jeweils wenigstens einen Vorteil und einen Nachteil:

a) Unter einem „Trap" versteht man

(2)

b) Die wesentlichen Nachteile des Rückschreibverfahrens gegenüber dem Durchschreibverfahren bestehen darin, dass

(2)

c) Benennen Sie die Stufen einer (Befehls)Pipeline eines (skalaren) RISC-Prozessors. Wie groß ist die maximal mögliche Steigerung des Durchsatzes?

(3)

d) Nennen Sie mindestens zwei Pipelinehemmnisse und beschreiben Sie Möglichkeiten zu deren Lösung.

(3)

e) Ein direct-mapped Cache hat gegenüber einem vollassoziativen Cache den

Vorteil:

Nachteil:

(3)

f) In welchen Zuständen kann sich ein Prozeß befinden und welche Zustandsübergänge gibt es? Skizzieren Sie ein Diagramm mit gerichteten Übergängen!

Skizze:

(3)

Name: Vorname: Matr.-Nr.:

Aufgabe 3 Pipelining (12 Punkte)

a) Bei einer fünfstufigen Pipeline eines (skalaren) RISC-Prozessors bewirken Sprungbefehle und durchgeführte, bedingte Verzweigungsbefehle, dass die Pipeline für drei Taktzyklen angehalten werden muss. Um wieviel Prozent erhöht sich die durchschnittliche Dauer pro Befehl, wenn ein Testprogramm zu 20% aus den o.g. Befehlen besteht und wenn man annimmt, dass es daneben zu keinen weiteren Pipelinehemmnissen kommt?

(6)

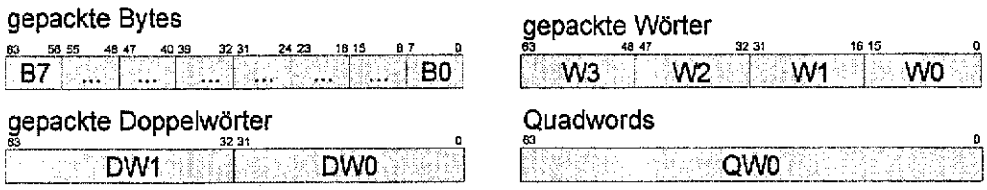
b) Der RISC-Prozessor verfügt über eine Prefetch Queue für bis zu 20 Befehle. Im Durchschnitt seien vier der vorab geholten Befehle Verzweigungsbefehle. Die Sprungzielvorhersage habe eine Trefferquote von 90%. Mit welcher Wahrscheinlichkeit befinden sich die benötigten Sprungbefehle in der Prefetch Queue?

(6)

Name: Vorname: Matr.-Nr.:

Aufgabe 4 MMX-Rechenwerk (20 Punkte)

Ein MMX-Rechenwerk (*Multimedia Extension*) unterstützt gepackte Datenformate nach untenstehendem Bild, die in einem 64 -bit-Register wahlweise 8 Bytes, 4 (16 -bit-) Wörter, 2 (32 -bit-)Doppelwörter oder ein 64 -bit-Wort (Quadword) unterbringen. Alle Werte können vorze ichenlos oder vorzeichenbehaftet sein. Negative Werte werden dabei im 2er-Komplement dargestellt. Spezielle MMX-Befehle wirken parallel auf diese Datenformate, d.h. es können z.B. durch einen einzigen Addier -Befehl zweimal 8 Bytes a ddiert wer den. Ein Übertrag zw ischen den einzelnen Werten der gepackten Daten findet dabei nicht statt. Die Datenbreite wird im Assemblerbefehl spezifiziert. Der erwähnte Addier -Befehl hat z.B. die Form: PADDX, wobei X=B,W,D für 8 Bytes, 4 Wörter, 2 Doppelwörter ste ht. Alle Befehle werden im Zwe iadreß-Format benutzt, d.h. das Ergebnis wir in einem der Eingaberegister abgelegt.



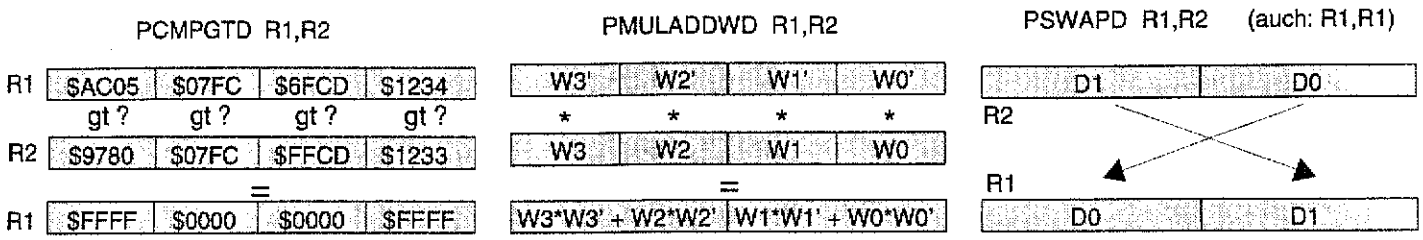
Der MMX Befehlssatz enthält u.a. die in folgender Tabelle angegebenen Befehle:

Mnemo		Bemerkung
	Logische Operationen (wirken auf alle 64 Bits)	
PAND	*Packed And*	bitweise Und-Verknüpfung
PANDN	*Packed And-Not*	bitweise Und-Verknüpfung mit negiertem 1. Operanden
POR	*Packed Or*	bitweise Oder-Verknüpfung
PXOR	*Packed Exclusive Or*	bitweise Antivalenz-Verknüpfung
	Vergleichsbefehle (für X = B, W, D, s. Skizze)	
PCMPEQX	*P. Compare Equal*	elementeweiser Vergleich auf gleich
PCMPGTX	*P. Compare Greater*	elementeweiser Vergleich auf größer
	Transferbefehle	
MOVQ	*Move Quadword*	Transfer zw. MMX-Reg. und MMX-Reg. bzw. Speicher
MOVD	*Move Doubleword*	Transfer zw. MMX-Reg. und MMX-Reg./Speicher
PSWAPD	*P. Swap Doubleword*	Vertauschen der Doppelwörter in Register, s. Skizze
	Multiplizier-Addierbefehl (W x W → D, s. nachfolgende Skizze)	
PMULADDWD	Multiply-Add	4fache Multiplikation mit Addition zu Doppelwörtern
PADDX	*Packed Add*	Parallele Addition mit X=B,W,D

Als Ergebnis liefern die Vergleichsbefehle für jedes Element (B,W,D) den Wert $F..F (also eine Folge von ‚1'-Bits), wenn der Vergleich wahr ist, andernfalls den Wert $0..0 (also eine Folge von ‚0'-Bits).

Name: Vorname: Matr.-Nr.:

Skizze:



1. In zwei 64-bit-Registern seien die folgenden gepackten Wörter vorgegeben:

R1 = (W3, W2, W1, W0) und R2 = (W3', W2', W1', W0').

Geben Sie eine Befehlsfolge aus den beschriebenen Befehlen an, die im Ergebnisregister das paarweise (wortweise) Maximum der beiden Operandenregister R1, R2 enthält. Die Eingaberegister dürfen dabei überschrieben werden.

Führen Sie die Befehlsfolge an folgendem Beispiel durch:

R1 = ($C7A0, $50FE, $0800, $7F0B) und
R2 = ($9F40, $50FE, $A060, $3400).

(8)

2. Geben Sie an, wie man mit dem Befehl PMULADDWD „doppelt -genaue" Multiplikationen 16 x 16 → 32 bit ausführen kann ?

(4)

Name: Vorname: Matr.-Nr.:

3. Im Speicher seien ab der Startadresse Adr_1 acht 16-bit-Wörter Wi, i=0,..,7, und ab der Startadresse Adr_2 acht 16 -bit-Koeffizienten Ci, i=0,..,7 abgelegt. Für diese Zahlen soll der Ausdruck

$$S = \sum_{i=0,..,7} Ci * Wi$$

berechnet werden.

Geben Sie eine MMX-Befehlsfolge für die geforderte Berechnung an.

Führen Sie die Befehlsfolge symbolisch mit den Bezeichnern Wi, Ci durch.

(8)

Hinweise:

- Die Summe soll im niederwertigen Teil des Ergebnisregisters stehen; der Wert des höherwertigen Teils im Register sei irrelevant.
- Die Startadressen der Operanden Wi, Ci für i=4,..,7 seien Adr_1+4 und Adr_2+4.

Name: Vorname: Matr.-Nr.:

Aufgabe 5 Segmentierter Speicher (12 Punkte)

a) Erläutern Sie kurz die wesentlichen Merkmale der segmentorientierten Speicherverwaltung.

(3)

b) Im Kurs haben Sie die *First-fit*-Zuweisungsstrategie zur Einlagerung von Segmenten in den Arbeitsspeicher kennengelernt, bei der die freien Lücken im Speicherbereich nach aufsteigenden Anfangsadressen geordnet werden und das Segment in die erste Lücke eingelagert wird, in die es hineinpaßt. In der Praxis wird häufig auch die alternative *Rotating-First-fit*-Strategie eingesetzt:

- Die Suche nach der ersten 'passenden' Lücke beginnt stets unmittelbar nach dem zuletzt eingelagerten Segment.
- Ist die Suche bis zum Ende des Speicherbereichs erfolglos, so wird sie am Beginn des Speicherbereichs fortgesetzt (*Round Robin*).
- Wird dabei der Ausgangspunkt der Suche wieder erreicht, so kann das Segment nicht eingelagert werden.

Das Betriebssystem kann jederzeit Speicherplatz freigeben, indem es die darin abgelegten Segmente auslagert.

Betrachtet werde nun ein 20 kbyte großer Speicherbereich, in dem die folgenden Segmente, deren Größe in kbyte angegeben ist, eingelagert werden sollen:

Segment	A	B	C	D	E	F	G	H	I	J
Größe	2	3	7	4	1	2	1	4	1	1

Es finde die nachstehende Folge von Ein- und Auslagerungen statt:

A B C <u>B</u> D E F <u>A</u> <u>C</u> G H I J <u>G</u> <u>F</u> B

Darin kennzeichne die Unterstreichung eines Segmentnamens die o.g. Auslagerung dieses Segments durch das Betriebssystem.

Tragen Sie in die Tabelle auf der folgenden Seite die Belegung des Speicherbereichs nach jeder Aktion ein, wobei Sie jedes Segment durch seinen Namen repräsentieren sollen. Dabei werde von einem zunächst völlig freien Speicherbereich ausgegangen. Jedes Kästchen der Tabelle stehe für einen 1 kbyte großen Speicherausschnitt. Durch '-' werden freie Speicherplätze markiert.

Name: Vorname: Matr.-Nr.:

Speicherbelegungsplan:

Aktion	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A	A	A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B																				
C																				
B																				
D																				
E																				
F																				
A																				
C																				
G																				
H																				
I																				
J																				
G																				
F																				
B																				

(6)

c) Welchen Vorteil bietet das *Rotating-first-fit*-Verfahren gegenüber der *First-fit*-Strategie ?

(3)

Aufgabe 6 Timer (20 Punkte)

Geben sei ein 16-bit-Zeitgeber/Zähler-Baustein (Timer) mit dem im folgenden Bild dargestellten Steuerregister CR. Die darin eingetragenen Bitfelder sollen im wesentlichen die im Kurs beschriebenen Funktionen haben.

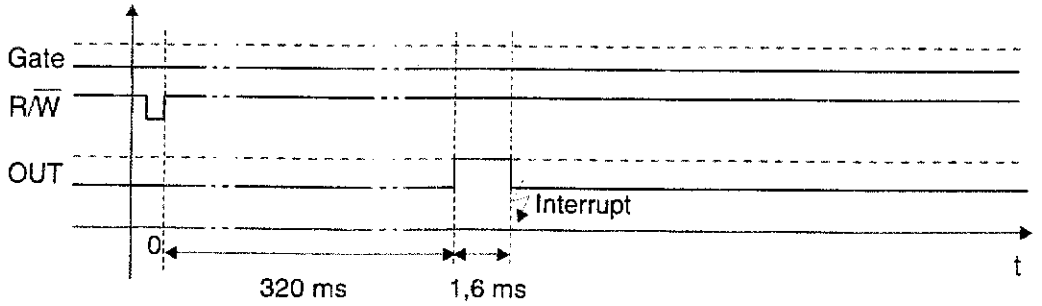
IE	OE	M2	M1	M0	CM	CL	PR

Es gelten die folgenden Abweichungen und Festlegungen:

- der Timer arbeitet n ur mit einem einzigen Anfangswert, d.h. einem Auffangregister (*Latch*) pro Zähler; er ist re-triggerbar;
- das Rücksetz-Bit fehlt;
- durch IE=1 werden Interrupts zur CPU durchgelassen;
- durch OE=1 wird der Timer-Ausgang freigeschaltet;
- der Zähler kann nur dual i m 16 -bit-Modus (CM=1) oder 2x8 -bit-Modus (CM=0) betrieben werden, nicht jedoch als Dezimalzähler;
- CL wählt zwischen einem externen 1 MHz -Takt (CL=1) oder dem internen Systemtakt (f >> 1MHz);
- durch PR=1 wird ein 1:8-Frequenzteiler aktiviert;
- durch die Bits M2, M1, M0 werden die folgenden Arbeitsweisen selektiert:

M2	M1	M0	Mode
0	0	0	periodischerZählmodus (Taktgenerator), nur Hardware-Triggerung
0	0	1	periodischerZählmodus, Hardware- und Software-Triggerung
0	1	0	Monoflop-Betrieb, nur Hardware-Triggerung
0	1	1	Monoflop-Betrieb, Hardware- und Software-Triggerung
1	0	0	*Strobe*-Betrieb, nur Hardware-Triggerung
1	0	1	*Strobe*-Betrieb, Hardware- und Software-Triggerung
1	1	0	Impulsmessung
1	1	1	Frequenzmessung

a) Am Ausgang OUT werde das folgende Zeitdiagramm (nicht maßstäblich) aufgezeichnet:



Geben Sie einen geeigneten Programmierwert für das o.g. Steuerregister CR sowie den Anfangswert für den Zähler an, die zu diesem Ausgangssignal führen können. (Der Wert für CR soll binär und hexadezimal, für das Latch dezimal und hexadezimal angegeben werden.) Begründen Sie Ihre Angaben.

(3)

b) Geben Sie die Werte für CR und Latch für den Fall an, daß ein symmetrisches Rechtecksignal (*Square Wave*) mit minimaler Frequenz am Ausgang OUT erzeugt wird. Wie groß ist diese Frequenz ? Wie groß seine Schwingungsdauer ? Setzen Sie dabei voraus, daß der Ausgang OUT (erst) nach jedem vollständigen Zählzyklus des Timers seinen Zustand wechselt.

(5)

c) Der Prozessor hat keine Hardware-Möglichkeit festzustellen, ob der von ihm im periodischen Zählmodus gestartete Timer über seinen Gate-Eingang angehalten wurde oder nicht. Geben Sie die Skizze für ein Programm(stück) an, durch das der Prozessor den Zustand des *Gates* (zu einem bestimmten Zeitpunkt) feststellen kann.

(4)

d) Geben Sie die Skizze für ein Programm(stück) an, das den Timer als sog. Wobbel-Generator betreibt, d.h. die Frequenz der Rechteckschwingung am Ausgang OUT steigt von einer (programmierbaren) Startfrequenz bis zu einer Endfrequenz an und fällt danach wieder zur Anfangsfrequenz ab. Dieser Vorgang wird zyklisch fortgesetzt, wobei die Zeitspanne für einen Zyklus variable ist.

(5)

e) Programmieren Sie den Timer als Watch-Dog, der nach 1 ms einen Alarm (Interrupt) auslöst, wenn er nicht rechtzeitig wieder getriggert wird.

(3)