Kurs 1708 im WS 2005/2006

Korrigierte Fassung der

Lösungsvorschläge zur

Hauptklausur

am 04. Februar 2006

Hinweis:
Leider hatte sich in die Aufgabenstellung zu 5c ein Druckfehler eingeschlichen, der den Sinn der Aufgabe etwas entstellte. In dieser Musterlösung wird daher zunächst ein Lösungsvorschlag zu der „falschen“ Aufgabenstellung gegeben, nach der auch die Korrektur stattfinden wird. Diese wird natürlich so „großzügig“ wie möglich durchgeführt werden. Im Anhang finden Sie dann die korrekte Fassung der Aufgabe und ihre Musterlösung. Wir bitten für diesen Fehler um Verständnis.

Aufgabe 1: Zahlendarstellung (14 Punkte)

a) Gegeben sei die grobe Näherung der Zahl $\pi \approx 3{,}140625$. Stellen Sie diese Zahl als einfach-genaue Zahl der Länge 32 bit (*single*) nach dem IEEE-754-Standard binär dar.

$\pi \approx (-1)^0 \cdot 2^1 \cdot 1.5703125_{10}$	normalisierte Mantisse und Exponent als Dezimalzahlen
$(1.M)_2 = 1.1001001_2$	normalisierte Mantisse als Binärzahl

π im IEEE-Format:

$C = (1+127)_{10} = 128_{10} = 10000000_2$	Charakteristik als Dezimal- und Binärzahl
$\pi = 0\|10000000\|10010010000000000000000_2$	π als Binär- und Hex.-Zahl
= $40490000	

b) Stellen Sie die in hexadezimaler Schreibweise angegebene Zahl Z = $7F800000 der Länge 32 bit (*single*) nach dem IEEE-754-Standard als Binärzahl dar. Um welchen Wert handelt es sich?

Binärdarstellung:	$Z = 0\|11111111\|00000000000000000000000_2$	
Wert:	$Z = +\infty$	(positiv unendlich)

c) Wandeln Sie die folgende, in hexadezimaler Form gegebene 32-bit-Zahl Z_{32} nach dem IEEE-754-Standard in ihre Hexadezimaldarstellung nach dem 64-bit-Format um.

Z_{32} = $E2C534FD

32 bit, binär:	1|11000101|10001010011010011111101		
Vorzeichen:	1 ⇔ –	Mantisse (1.M):	$C534FD
Charakteristik:	11000101_2	Exponent:	01000110_2
64 bit, binär:	1|10001000101|10001010011010011111.... 1010000000000000000000000000000		
Vorzeichen:	1 ⇔ –	Mantisse(1.M):	$18A69FA0000000
Charakteristik:	10001000101_2	Exponent:	01000110_2

Z_{64} = $C458A69FA0000000

Aufgabe 2: Adressierungsarten (6 Punkte)

a) Was ist ein Vorteil der Programmzähler-relativen Adressierung?

Diese Adressierungsart erlaubt es, Programme so zu schreiben, daß sie an einer beliebigen Stelle im Arbeitsspeicher ablauffähig sind ("dynamisch verschiebbarer Programmcode").

b) Mit welcher Adressierungsart bzw. Adressierungsarten kann man in einem dynamisch verschiebbaren Programm den Zugriff auf Daten realisieren, wenn dieser Zugriff möglichst schnell und nicht über den Programmzähler geschehen soll? Begründen Sie Ihre Aussage.

Speicher-relative bzw. Register-relative Adressierung. Damit auch die Daten verschiebbar sind, muß relative Adressierung eingesetzt werden. Damit die Berechnung der effektiven Adresse schnell geht, muß eine einstufige Adressierung eingesetzt werden, so daß ein weiterer Speicherzugriff vermieden wird.

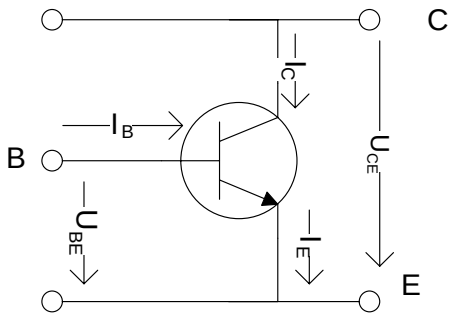
c) Die absolute Adressierung kann man auch durch eine relative Adressierung mit Offset 0 realisieren. Welche Nachteile besitzt diese Realisierung jedoch gegenüber der absoluten Adressierung? Nennen Sie wenigstens zwei Nachteile.

Die Ausführung wird aufgrund komplizierter Adressierungsarten verzögert, da die Berechnung der effektiven Adresse Zeit benötigt. Da kein Offset angegeben werden muß, ist der Befehl i.d.R. kürzer und benötigt daher weniger Speicherplatz. Außerdem wird die Erkennung von Datenabhängigkeiten komplizierter, was einen erhöhten Hardwareaufwand nach sich zieht.

Aufgabe 3: Bipolarer Transistor und Firewire (15 Punkte)

Der in Kurseinheit 5, Abschnitt 5.1.3, beschriebene bipolare Transistor wird wegen der unterschiedlichen Dotierung der Anschlußbereiche (Basis - B: p-Dotierung, Emitter – E: n-Dotierung, Kollektor - K: n-Dotierung) auch als npn-Transistor bezeichnet.

a) Zeichnen Sie in das folgende Bild für diesen npn-Transistor die Namen (B, C, E) der Anschlußbereiche sowie durch Pfeile die Richtung der Spannungen und der Ströme mit ihren Bezeichnungen nach nebenstehender Tabelle ein.



Bez.	Bedeutung
U_{CE}	Kollektor-Emitter-Spannung
U_{BE}	Basis-Emitter-Spannung
I_C	Kollektorstrom
I_B	Basisstrom
I_E	Emitterstrom

b) Beantworten Sie folgende Fragen mit “richtig” oder “falsch” Begründen Sie Ihre Aussagen, wenn es verlangt wird.

Ist es mit FireWire möglich, eine Videokamera C mit einem Videobearbeitungsgerät G, dieses mit einem weiteren Gerät H zu verbinden und dieses wiederum an eine zweite FireWire-Schnittstelle von C anzuschließen ? falsch

Begründung: FireWire erlaubt fast beliebige Netzstrukturen mit Verzweigungen und Verkettungen, jedoch keine geschlossenen Schleifen.

FireWire verwendet 6-polige Kabel. richtig

Ohne Spannungsversorgung kann man FireWire mit einem 4-poligen Kabel realisieren. richtig

Begründung:
Dies ist möglich, wenn alle angeschlossenen Geräte über eine eigene Spannungsversorgung verfügen.

Mit FireWire lassen sich nur feste Netzwerkstrukturen realisieren. falsch

Begründung:
Das Bussystem ist selbstkonfigurierend, d.h. nach dem Einschalten oder dem Dazuschalten von Geräten ermitteln die Knoten, welcher die Funktion des Wurzelknotens wahrnehmen darf.

Nur ein vorher festgelegtes Gerät kann bei FireWire den Takt erzeugen. falsch

Bei der isochronen Übertragung erhöht *Subaction Concatenation* die physikalische Bandbreite von FireWire. falsch

Begründung:
Die physikalische Bandbreite bleibt zwar gleich, die verfügbare Bandbreite für den Benutzer wird jedoch erhöht.

Aufgabe 4: MESI-Protokoll (20 Punkte)

Gegeben sei ein speichergekoppeltes Multiprozessorsystem mit zwei Prozessoren P1, P2, die über einen Bus mit einem globalen Speicher verbunden sind und jeweils über einen lokalen Cache C1, C2 verfügen, der nach dem Write-Back-Verfahren verwaltet wird. Zur Wahrung der Cache-Kohärenz wird das MESI-Protokoll verwendet. Die Caches seien als *3-Way Set Associative Caches* realisiert und haben je eine Größe von einem einzigen Eintrag (*Cache Line*), der jeweils genau ein Speicherwort aufnehmen kann. Falls noch freie (bzw. ungültige) Einträge zur Verfügung stehen, werden die Teilcaches (*Way*) in aufsteigender Reihenfolge aufgefüllt. Im Falle eines voll besetzten Caches werden die Einträge nach der Strategie LRU (*Least Recently Used*) überschrieben.

Ergänzen Sie die folgende Tabelle, wobei die Buchstaben die Zustände

M = *exclusive modified*, E = *exclusive unmodified*, S = *shared unmodified*, I = *invalid/empty*

repräsentieren. (Es reicht, die jeweiligen Änderungen einzutragen.) Die Zahlen hinter den Aktionen und Zuständen bezeichnen Speicheradressen. So bedeuten z.B.:

read 18	*„Lese die Speicherzelle 18.“*
E/8	*„Eintrag enthält Wert der Speicherzelle 8 im Zustand exclusive.”*
I/–	„*Eintrag ist ungültig oder leer.*“

Prozessor	Aktion	Cache C1			Cache C2		
		Way 1	Way 2	Way 3	Way 1	Way 2	Way 3
-	-	E/8	E/12	I/-	E/6	I/-	I/-
P1	read 18			E/18			
P2	write 18			I/-		M/18	
P1	write 8	M/8					
P2	read 10						E/10
P1	read 10			S/10			S/10
P2	read 8	S/8			S/8		
P1	write 8	M/8			I/-		
P2	write 10			I/-			M/10

Aufgabe 5: Schnittstellen (30 Punkte)

a) Welche minimale Übertragungszeit erreicht das Centronics-Protokoll ohne Empfangsbestätigung, wenn das Ausgabegerät mit der Datenübernahme 0,2 µs, mit der Bearbeitung eines Datums 0,8 µs beschäftigt ist, das Gerät einen Puffer von 4 Bytes zur Verfügung stellt und die Verzögerung der Übertragung durch das Leitungsmedium und Gatterschaltzeiten vernachlässigbar sind? Berechnen Sie die minimale Gesamtübertragungszeit für einen Block der Größe 1 kByte (1024 Bytes). (Das Signal ACK braucht hierbei nicht berücksichtigt zu werden.)

Rechnung:

Nach Bild 7.1-10 der KE7 darf das Strobe-Signal frühestens nach der 0,5 µs langen „Einschwingzeit“ (E) der Daten beginnen, es muß mindestens 0,5 µs (A) aktiviert sein, und die Stabilisierungsphase (S) der Daten muß mindestens 0,5 µs dauern.

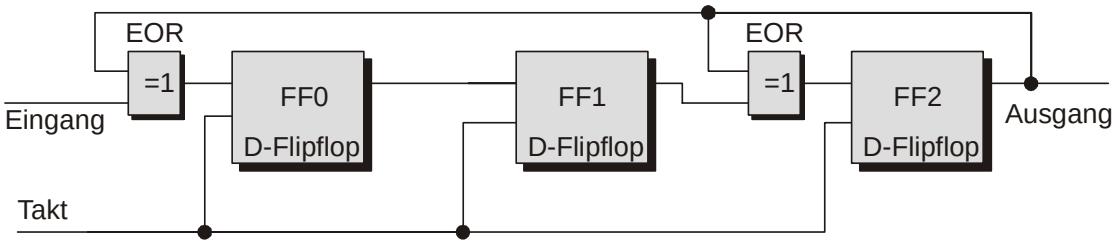
Übernahme (Ü) und Bearbeitung (B) jedes einzelnen Datums, zusammen 1 µs, überlappen sich für 0,5 µs mit der Stabilisierungsphase, für die restlichen 0,5 µs mit der Einschwingzeit der folgenden Übertragung. Daher ist der Puffer ohne Bedeutung.

$$(0{,}5\ \mu s\ E + 0{,}5\ \mu s\ A + \underbrace{0{,}5\ \mu s\ S) + (0{,}5\ \mu s\ E}_{0{,}2\ \mu s\ Ü\ +\ 0{,}8\ \mu s\ B} + 0{,}5\ \mu s\ A + \underbrace{0{,}5\ \mu s\ S) + (0{,}5\ \mu s\ E}_{0{,}2\ \mu s\ Ü\ +\ 0{,}8\ \mu s\ B}$$

$\Rightarrow$ Übertragungszeit für 1 kbyte: $(1024 * 1{,}5 + 1)\ \mu s = 1537\ \mu s$

wobei die 1 µs die Zeit vor der Übernahme des 1. Datums berücksichtigt.

b) Die zur Sicherung der Übertragung eines HDLC-Rahmens (nach Unterabschnitt 7.3.2) gegen Übertragungsfehler verwendete Prüfsumme (*Checksum*) wird nach dem Verfahren der zyklischen Redundanzprüfung (*Cyclic Redundancy Check* – CRC) gebildet und kann durch ein „rückgekoppeltes Schieberegister“ ermittelt werden. Im folgenden Bild ist – als vereinfachtes Beispiel – solch ein Schieberegister für die Berechnung einer 3-bit-Prüfsumme skizziert:



Die Schaltung erwartet als serielle Eingabe alle Datenbits des Übertragungsrahmens in der dort vorgegebenen Reihenfolge, ggfs. durch das *Bit Stuffing* um einige Bits erweitert. Nach der Auswertung aller Datenbits wird der Inhalt der D-FFs (in der Reihenfolge FF2, FF1, FF0) als 3-bit-Prüfsumme an die Datenbits angehängt. (Die dazu benötigte Schaltung ist im oben stehenden Bild nicht dargestellt.)
Berechnen Sie mit Hilfe des skizzierten Schieberegisters die Prüfsumme für die folgenden Datenbits:

1. Bit → 10110011 ← letztes Bit

Vor Beginn der „Berechnung“ seien alle Flipflops im Zustand ‚0’. Tragen Sie in folgende Tabelle die Zustände der Flipflops nach den Taktsignalen ein.

Eingabebit	Takt	FF0	FF1	FF2
		0	0	0
1	1	1	0	0
0	2	0	1	0
1	3	1	0	1
1	4	0	1	1
0	5	1	0	0
0	6	0	1	0
1	7	1	0	1
1	8	0	1	1

Prüfsumme: 110 (nach absteigenden Bit-Wertigkeiten)

c) Es werde der folgende in hexadezimaler Schreibweise angegebene Übertragungsrahmen im HDLC-Protokoll empfangen:

$7E3C2F49737420393E1F0D8A7D5D4C7E.

Bei der Übertragung sei eine 16-bit-CRC-Prüfsumme verwendet worden. Analysieren Sie den empfangenen Übertragungsrahmen und geben Sie die Informationen in hexadezimaler Form an:

Flag:	$7E
Empfänger-Adresse:	$3C
Steuerfeld:	$2F
Datenfeld:	$49737420393E 1F0D8A7D
FCS:	$5D4C
Flag:	$7E

Die Daten werden im 8-bit-ASCII-Code übertragen (s. Tabelle im Anhang). Geben Sie an, welcher Text im Datenfeld eingetragen ist. Berücksichtigen Sie dabei das *Bit-Stuffing*-Verfahren.

Empfangene Daten in Binärdarstellung:

0100 1001 0111 0011 0111 0100 0010 0000 0011 1001 0011 1110 0001 1111 0000 1101 1000 1010 0111 1101

Korrigierte Daten in Binärdarstellung:

0100 1001 0111 0011 0111 0100 0010 0000 0011 1001 0011 1110 0011 1110 0011 0110 0010 1001 1111 1000 Erweiterung zur Tetrade

Korrigierte Daten in Hexadezimaldarstellung: $49737420393E3E36 29 F8

Nachricht im Klartext: Ist 9>>6)°

Aufgabe 6: A/D-Wandlung (15 Punkte)

In dieser Aufgabe sollen Sie die Analog/Digitalwandlung durch einen 8-bit-A/D-Baustein mit sukzessiver Approximation nachvollziehen. Die Spannungswerte des zu wandelnden Signals liegen dabei zwischen U_{max} = 2,56 V und U_{min} = 0 V. Das Signal soll in diskreten Abständen von jeweils t=1 Zeiteinheiten abgetastet werden. Die folgende Tabelle zeigt einige der abgetasteten Spannungen in Volt.

t	0	1	2	3	4	5	6
U	1,0	1,84147	0,24319	1,41211	0,71209	0,86764	0,08221

a) Geben Sie allgemein die Berechungsformel der Ausgangsspannung $U_{DA}(2^i)$, i=0,..,7, des integrierten D/A-Wandlers für den Digitalwert an, in dem nur das Bit i = 1 ist. Tragen Sie die so berechneten „Schwellspannungen“ für die sukzessive Approximation in die unten stehende Tabelle ein.

Formel:

$$U_{DA}(2^i) = [(U_{max} - U_{min}) / 2^8] * 2^i = [(2{,}56 - 0) / 256 \text{ V}] * 2^i = 2^i * 0{,}010 \text{ V}$$

i	0	1	2	3	4	5	6	7
$U_{DA}(2^i)$ in V	0,010	0,020	0,040	0,080	0,160	0,320	0,640	1,280

b) Wandeln Sie die Spannungswerte der abgetasteten Daten nach oben stehender Tabelle für t=0, 3, 4, 6 in ihre Digitalwerte D(t) um und berechnen Sie die dabei auftretenden relativen Fehler nach folgender Formel:

$$r(t) = [U(t) - U_{DA}(D(t))] / U(t) .$$

	t = 0	**t = 3**	**t = 4**	**t = 6**
D(t)	100 = $064	141 = $8D	71 = $47	8 = $08
U_{DA}(D(t)) in V	1,000	1,410	0,710	0,080
r(t) in **%**	0	0,15	0,29	2,68

c) Tragen Sie die Ergebnisse der einzelnen Vergleichsschritte der sukzessiven Approximation bei der Wandlung des Spannungswertes U = 0,86764 V (für t=5) in die folgende Tabelle ein. Dabei bedeuten:

VS: die Spannung in Volt, mit der U verglichen wird,

VE: das Ergebnis dieses Vergleichs (≥ , <),

D_i: der resultierende Wert des Bits i im Digitalwert D(5) zu t=5.

Schritt	Bit i	VS	VE	D_i
1	7	1,28	<	0
2	6	0,64	≥	1
3	5	0,96	<	0
4	4	0,80	≥	1
5	3	0,88	<	0
6	2	0,84	≥	1
7	1	0,86	≥	1
8	0	0,87	<	0

Endergebnis:

D(5) = 01010110 = $56

Anhang: Der ASCII-Code

2. Tetrade

1. Tetrade	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	(LZ)	!	„	#	$	%	&	,	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
9	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	ƒ
A	á	í	ó	ú	ñ	Ñ	ª	º	¿	®	¬	½	¼	¡	«	»
B	_	_	_	¦	¦	Á	Â	À	©	¦	¦	+	+	¢	¦	+
C	+	-	-	+	-	+	ã	Ã	+	+	-	-	¦	-	+	¤
D	ð	Ð	Ê	Ë	È	ı	Í	Î	Ï	+	+	_	_	¦	Ì	_
E	Ó	ß	Ô	Ò	õ	Õ	µ	þ	Þ	Ú	Û	Ù	ý	Ý	¯	´
F	-	±	_	¾	¶	§	÷	¸	°	¨	·	¹	³	²	_	

(LZ): Leerzeichen (*blank*)

Ursprüngliche Fassung der Aufgabe 5c

Übertragungsrahmen: $7E3C2F49737420393E1F0D8**F**A7D5D4C7E.

Musterlösung:

d) Es werde der folgende in hexadezimaler Schreibweise angegebene Übertragungsrahmen im HDLC-Protokoll empfangen:

$7E3C2F49737420393E1F0D8**F**A7D5D4C7E.

Bei der Übertragung sei eine 16-bit-CRC-Prüfsumme verwendet worden. Analysieren Sie den empfangenen Übertragungsrahmen und geben Sie die Informationen in hexadezimaler Form an:

Flag:	$7E
Empfänger-Adresse:	$3C
Steuerfeld:	$2F
Datenfeld:	$49737420393E1F0D8**F**A7D
FCS:	$5D 4C
Flag:	$7E

Die Daten werden im 8-bit-ASCII-Code übertragen (s. Tabelle im Anhang). Geben Sie an, welcher Text im Datenfeld eingetragen ist. Berücksichtigen Sie dabei das *Bit-Stuffing*-Verfahren.

Empfangene Daten in Binärdarstellung:

0100 1001 0111 0011 0111 0100 0010 0000 0011 1001 0011 1110 0001 1111 0000 1101 1000 **1111** 1010 0111 1101

Korrigierte Daten in Binärdarstellung:

0100 1001 0111 0011 0111 0100 0010 0000 0011 1001 0011 1110 0011 1110 0011 0110 0011 1111 0011 1111

Korrigierte Daten in Hexadezimaldarstellung: $49737420393E3E363F3F

Nachricht im Klartext: Ist 9>>6??