Modul Computersysteme
Prüfungsklausur WS 2011/2012

Lösungsvorschläge

Prof. Dr. J. Keller
LG Parallelität und VLSI
Prof. Dr.-Ing. W. Schiffmann
LG Rechnerarchitektur

Aufgabe 1 (10 Punkte):

a) Gegeben ist das folgende Karnaugh-Diagramm einer Schaltfunktion f in den vier Variablen X_1 bis X_4.

X_1

X_2

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

X_4

X_3

Vervollständigen Sie die Wertetabelle! (3 P.)

Hinweis: Wie im Kurstext verwenden wir die verkürzende Schreibweise für Karnaugh-Diagramme, bei der gerade die Spalten bzw. Zeilen mit einer Variablen markiert werden, bei denen die entsprechende Variable den Wert 1 hat. Zum Beispiel gilt in den obersten beiden Zeilen $X_2 = 1$, und in den beiden äußeren Spalten $X_3 = 0$.

$X_1\ X_2\ X_3\ X_4$	$f(X_1, \ldots, X_4)$
0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

b) Geben Sie die Anzahl der Primimplikanten der Funktion f aus Teil a) an. (3 P.)

c) Nennen Sie einen der Primimplikanten der Funktion f aus Teil a). (1 P.)

d) Kreuzen Sie in der folgenden Primtermtabelle an, welche der Primimplikanten auch Kernimplikanten sind. (2 P.)

		Minterme/Trägerelemente						
	Primterme	M1	M2	M3	M4	M5	M6	M7
	P1	X	X			X	X	
	P2	X	X				X	X
	P3	X	X		X	X		
	P4					X		X
	P5	X	X	X		X		

e) Kreuzen Sie in der folgenden Primtermtabelle an, welcher der sogenannten Primimplikanten gar kein Primimplikant ist. (1 P.)

		Minterme/Trägerelemente						
	Primterme	M1	M2	M3	M4	M5	M6	M7
	P1	X	X				X	X
	P2	X	X			X	X	
	P3	X	X		X	X		
	P4		X				X	
	P5	X	X	X	X			

Lösung: *a) Die Wertetabelle lautet:*

$X_1\ X_2\ X_3\ X_4$	$f(X_1, \ldots, X_4)$
0000	*1*
0001	*0*
0010	*1*
0011	*1*
0100	*1*
0101	*0*
0110	*0*
0111	*1*
1000	*1*
1001	*1*
1010	*0*
1011	*0*
1100	*0*
1101	*1*
1110	*0*
1111	*0*

b) Die Anzahl der Primimplikanten ist 7, s. Teilaufgabe c).
c) Die Primimplikanten lauten $X_1\bar{X}_3X_4$, $X_1\bar{X}_2\bar{X}_3$, $\bar{X}_2\bar{X}_3\bar{X}_4$, $\bar{X}_1\bar{X}_3\bar{X}_4$, $\bar{X}_1\bar{X}_2\bar{X}_4$, $\bar{X}_1\bar{X}_2X_3$, $\bar{X}_1\bar{X}_2X_4$
d) P3 und P5 sind Kernimplikanten, da sie M4 bzw. M3 als einzige überdecken.
e) P4 kann kein Primimplikant sein, da er von P1 und P2 dominiert wird.

Aufgabe 2 (8 Punkte):

Rechnen Sie jeweils aus (je 1 P.):

$[01011011]_2 =$ ☐

$[11011010]_2 =$ ☐

$< 11011011 >_2 =$ ☐

$< 01011010 >_2 =$ ☐

$bin_8(250) =$ ☐

$bin_8(130) =$ ☐

$twoc(-1) =$ mit 8 Stellen ☐

$twoc(-3) =$ mit 8 Stellen ☐

Lösung:
$[01011011]_2 = 91$
$[11011010]_2 = -128 + 90 = -38$
$< 11011011 >_2 = 128 + 91 = 219$
$< 01011010 >_2 = 90$
$bin_8(250) = 11111010$
$bin_8(130) = 10000010$
$twoc(-1) = 1 \ldots 1$
$twoc(-3) = 1 \ldots 101$

Aufgabe 3 (11 Punkte):

a) Erstellen Sie die Wertetabelle für einen Volladdierer. (3 P.)

a	b	c_{in}	c_{out}	s
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Lösung:
Die Wertetabelle lautet

a	b	c_{in}	c_{out}	s
0	*0*	*0*	*0*	*0*
0	*0*	*1*	*0*	*1*
0	*1*	*0*	*0*	*1*
0	*1*	*1*	*1*	*0*
1	*0*	*0*	*0*	*1*
1	*0*	*1*	*1*	*0*
1	*1*	*0*	*1*	*0*
1	*1*	*1*	*1*	*1*

b) Wir betrachten n-stellige Binärzahlen mit Betrag und Vorzeichen, sog. signed binary digits (sbd). Formal ist die sbd-Darstellung gegeben durch die Funktion

$$sbd : \{0,1\}^{n+1} \to \{-2^n, \ldots, +2^n\} \text{ mit } sbd(a_n, \ldots, a_0) = \begin{cases} \langle a_{n-1}, \ldots, a_0 \rangle & \text{wenn } a_n = 0 \\ -\langle a_{n-1}, \ldots, a_0 \rangle & \text{wenn } a_n = 1 \end{cases}$$

d.h. die oberste Stelle gibt das Vorzeichen an, die restlichen n Stellen den Betrag. Für den Fall $n = 1$ gilt also die folgende Tabelle

$a_1 a_0$	$sbd(a_1, a_0)$
00	0
01	1
10	-0
11	-1

Für den Fall $n = 2$ gilt beispielsweise $sbd(101) = -1$ und $sbd(010) = +2$.

Zu entwerfen ist ein Addierer für zwei 1-stellige sbd-Zahlen ohne Eingangsübertrag, d.h. die Eingaben stellen Zahlen aus dem Bereich $\{-1, 0, +1\}$ dar, das Ergebnis stellt eine Zahl dar, die sich im Bereich $\{-2, \ldots, +2\}$ befindet. Das Ergebnis soll als sbd mit $n = 2$ Stellen ausgegeben werden (ohne Ausgangsübertrag).

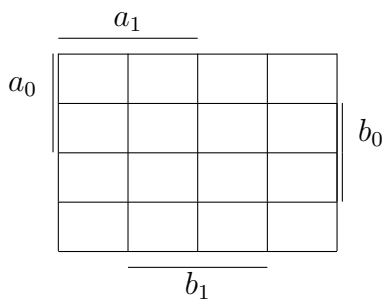
b1) Erstellen Sie die Wertetabelle, wobei beim Ergebnis Null dieses mit $0\ 0\ 0$ kodiert sein soll (d.h. das Vorzeichen der Null spielt keine Rolle). (4 P.)

$a_1\ a_0$	$b_1\ b_0$	s_2	s_1	s_0
00	00			
00	01			
00	10			
00	11			
01	00			
01	01			
01	10			
01	11			
10	00			
10	01			
10	10			
10	11			
11	00			
11	01			
11	10			
11	11			

Lösung: *Die Wertetabelle lautet*

$a_1\ a_0$	$b_1\ b_0$	$s_2 s_1 s_0$
00	*00*	*000*
00	*01*	*001*
00	*10*	*000*
00	*11*	*101*
01	*00*	*001*
01	*01*	*010*
01	*10*	*001*
01	*11*	*000*
10	*00*	*000*
10	*01*	*001*
10	*10*	*000*
10	*11*	*101*
11	*00*	*101*
11	*01*	*000*
11	*10*	*101*
11	*11*	*110*

b2) Bestimmen Sie ein Minimalpolynom für s_2 unter Nutzung eines KV-Diagramms. (4 P.)



Lösung: *Das KV-Diagramm lautet*

$a_0 \backslash a_1$			
1	*1*	*0*	*0*
0	*1*	*0*	*0*
0	*1*	*1*	*0*
0	*0*	*0*	*0*

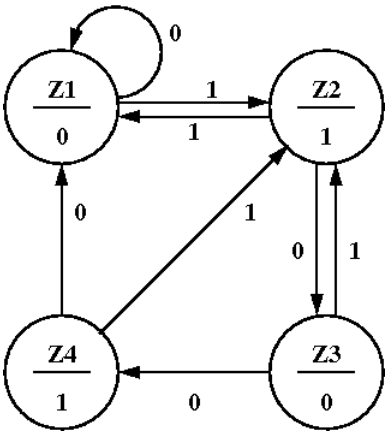
b_0, b_1

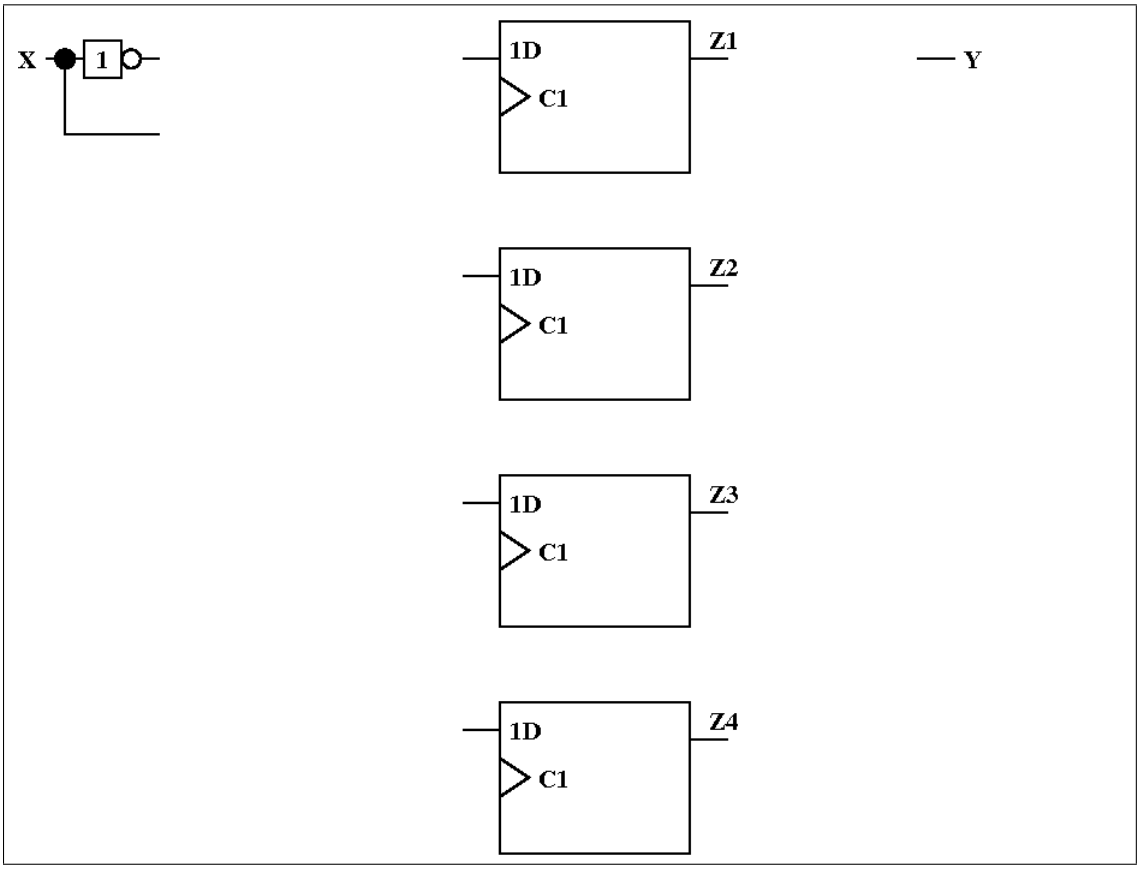
Damit ergeben sich die Primimplikanten $a_0 a_1 \bar{b_0}$, $b_0 b_1 \bar{a_0}$, $a_0 a_1 b_1$, $b_0 b_1 a_1$. *Hiervon sind die ersten beiden Kernimplikanten, von den anderen beiden Primimplikanten kann man sich einen auswählen. Die beiden möglichen Minimalpolynome lauten also:*

$$a_0 a_1 \bar{b_0} \vee b_0 b_1 \bar{a_0} \vee a_0 a_1 b_1, \qquad a_0 a_1 \bar{b_0} \vee b_0 b_1 \bar{a_0} \vee b_0 b_1 a_1 \ .$$

Aufgabe 4 (5 Punkte):

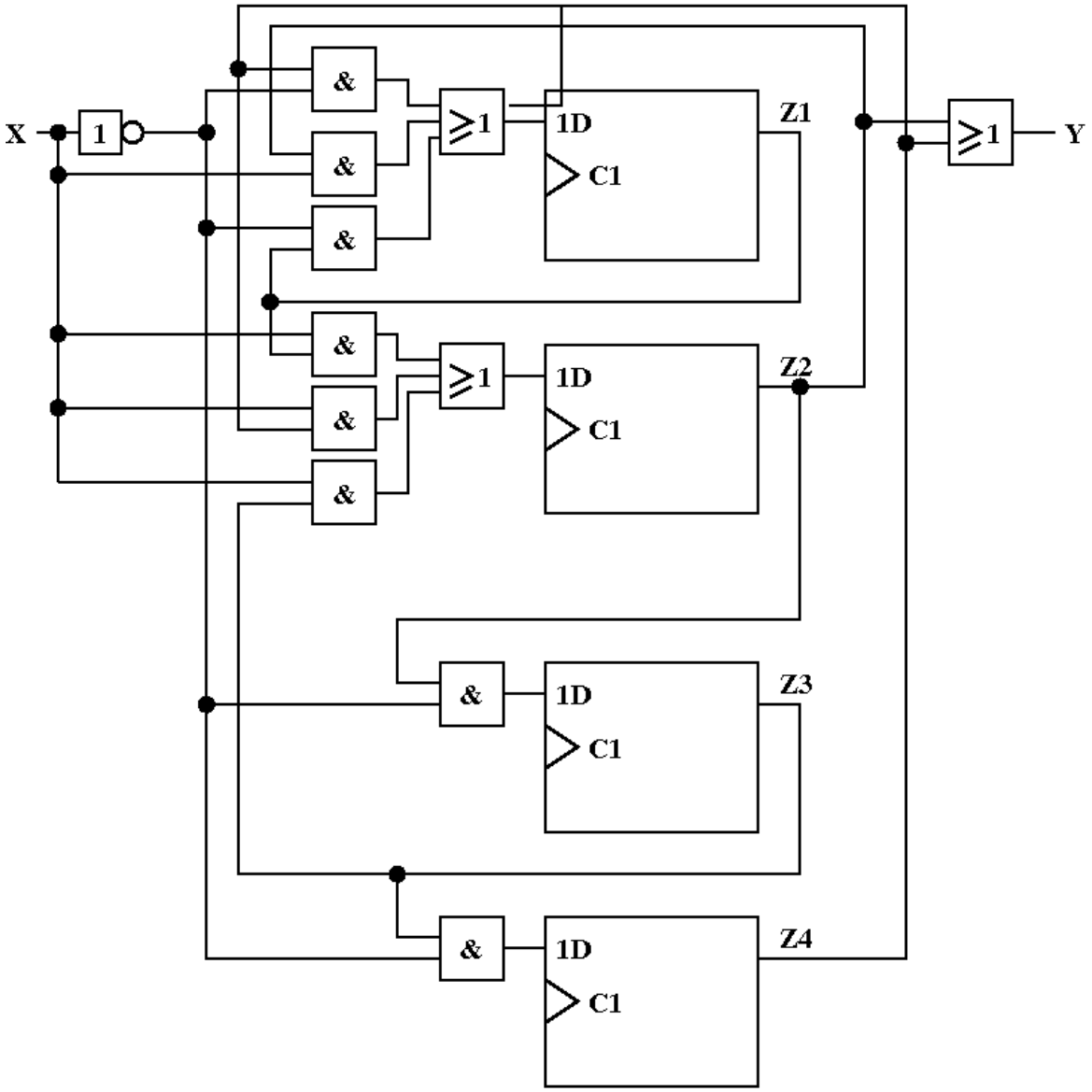
Gegeben sei ein Automat durch das folgende Zustandsdiagramm. Erstellen Sie ein Schaltwerk für diesen Automaten mit One-Hot-Codierung der Zustände unter Nutzung von D-Flipflops.





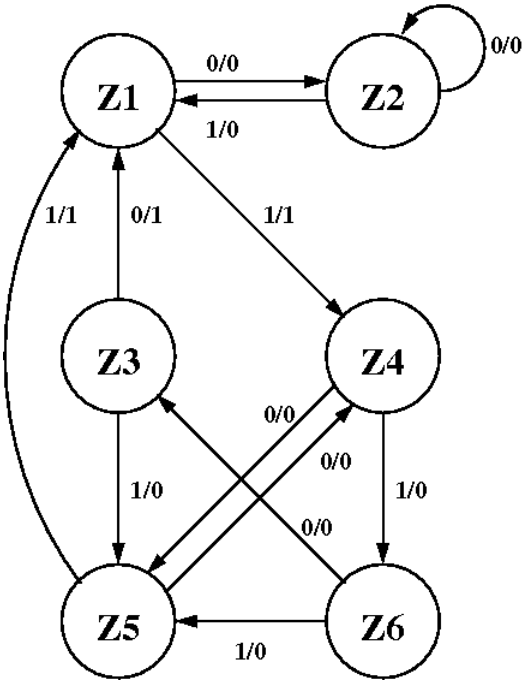
Lösung:

Das vollständige Schaltwerk benötigt neben den 4 Flipflops 12 Gatter, wobei keine Optimierung vorgenommen wurde. Alternativ könnte nämlich auch $Z_2^+ = (Z_1 \vee Z_3 \vee Z_4) \wedge X$ *und* $Z_1^+ = ((Z_1 \vee Z_4) \wedge \bar{X}) \vee Z_2 \wedge \bar{X}$ *gewählt werden.*



Aufgabe 5 (6 Punkte):

Gegeben ist ein Schaltwerk durch seinen Zustandsgraph.



a) Handelt es sich um einen Moore– oder einen Mealy-Automaten? (1 P.)

Lösung: *Es handelt sich um einen Mealy-Automaten, da y von der Eingabe abhängt.*

b) Erstellen Sie eine Liste aller Zustandspaare, die die Gleichung

(3.25) $\forall x \in I : f(x, z_i) = f(x, z_j)$

erfüllen, wobei f die Ausgangsfunktion des Automaten und I die Menge der möglichen Eingaben ist. (2 P.)

Lösung: *Die Zustandspaare (1,5), (2,4), (2,6) und (4,6) erfüllen Gleichung (3.25).*

c) Erstellen Sie für jedes Zustandspaar aus Teil b) die Liste der Folgezustandspaare, d.h. erstellen Sie die Tabelle Stufe 0 für eine Zustandsminimierung. (2 P.)

Lösung: *Die Tabelle lautet:*

(1,5)	*(1,4) (2,4)*
(2,4)	*(1,6) (2,5)*
(2,6)	*(1,5) (2,3)*
(4,6)	*(3,5) (5,6)*

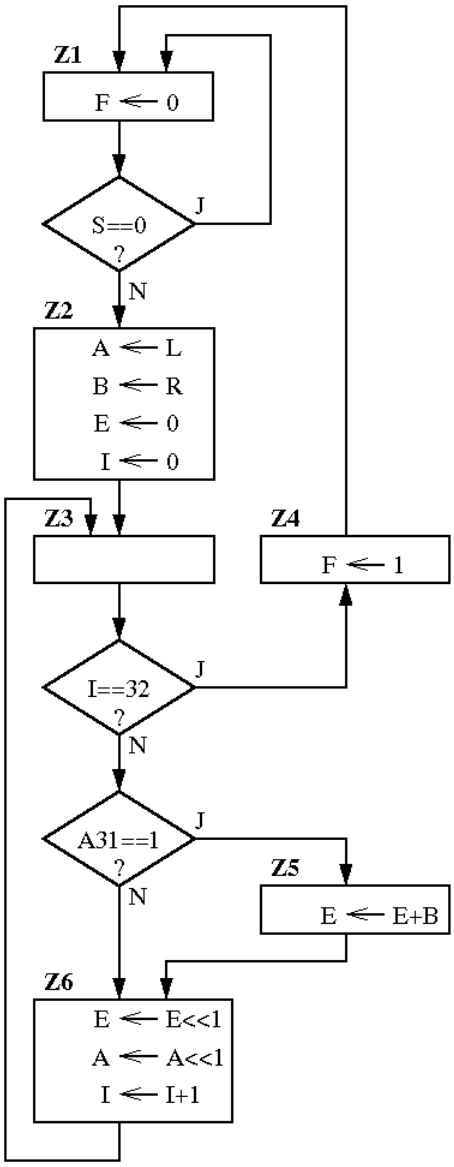
Man sieht aus dieser Tabelle übrigens direkt, dass es bei diesem Automaten keine äquivalenten Zustände gibt.

d) Kann es einen Zustandsgraphen mit 6 Zuständen, 1-Bit-Eingabe und 1-Bit-Ausgabe geben, bei dem kein Paar von Zuständen die obige Gleichung (3.25) erfüllt? Begründen Sie Ihre Antwort. (1 P.)

Lösung: *Nein, einen solchen Graphen kann es nicht geben, da es bei jeder der zwei möglichen Eingaben nur zwei mögliche Ausgaben gibt, d.h. insgesamt gibt es nur vier mögliche Ein-/Ausgabe-Kombinationen eines Zustands. Da es mehr als vier Zustände gibt, muss es zwei Zustände mit gleicher Ein-/Ausgabe-Kombination geben, die dann Gleichung (3.25) erfüllen.*

Aufgabe 6 (7 Punkte):

Gegeben sei das folgende ASM-Diagramm eines komplexen Schaltwerks, wobei die Eingabe aus zwei vorzeichenlosen 32-stelligen Binärzahlen L und R sowie einem Startsignal S besteht, es Register für Zahlen A, B, E, I und F gibt, und die Ausgabe aus dem Wert des Registers F besteht sowie aus dem Wert des Registers E, wenn das Ende von Zustand Z_4 erreicht wird, d.h. wenn der Inhalt des Registers F auf 1 gesetzt wird. Das Schaltwerk berechnet in E das Produkt aus L und R, wobei alle Register und die Arithmetik breit genug sein sollen, dass kein <berlauf stattfindet. Mit A_{31} ist das Bit 31 des Registers A mit Wertigkeit 2^{31} gemeint.



a) Benötigt man zur Realisierung ein Mealy-Steuerwerk oder reicht ein Moore-Steuerwerk? (1 P.)

☐ Mealy ☐ Moore

Lösung: *Moore, da keine bedingte Ausgangsbox vorhanden.*

b) Vervollständigen Sie die Tabelle des Steuerwerks, wobei lediglich die Steuersignale für die Register betrachtet werden. (3 P.)

Hinweis: Jedes Register verfügt über einen Steuereingang $S_{Registername}$, mit dem die Datenübernahme am Ende eines Zustands gesteuert wird. Eine 1 an diesem Steuereingang bewirkt die <bernahme des Wertes am Eingang des betreffenden Registers.

	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6
S_A						
S_B						
S_E						
S_I						
S_F						

Lösung:

	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6
S_A	*X*	*1*	*0*	*X*	*0*	*1*
S_B	*X*	*1*	*0*	*X*	*0*	*0*
S_E	*X*	*1*	*0*	*0*	*1*	*1*
S_I	*X*	*1*	*0*	*X*	*0*	*1*
S_F	*1*	*0*	*0*	*1*	*0*	*0*

c) Geben Sie an, für welche Register Multiplexer im Operationswerk benötigt werden. Unterscheiden Sie dabei Multiplexer mit 2, 3 oder 4 Eingängen. Geben Sie auch an, in welchen Zuständen jedes Register gesetzt wird. (2 P.)

Register	Multiplexer	Zustände
A		
B		
E		
I		
F		

Lösung:

Register	*Multiplexer*	*Zustände*
A	*2-Multiplexer*	*da in Z2, Z6 gesetzt*
B	*—*	*nur in Z2 gesetzt*
E	*3-Multiplexer*	*da in Z2, Z5, Z6 gesetzt*
I	*2-Multiplexer*	*da in Z2, Z6 gesetzt*
F	*2-Multiplexer*	*da in Z1, Z4 gesetzt*

d) Geben Sie an, wieviele Einheiten wie Addierer, Subtrahierer usw. jeweils im Operationswerk benötigt werden, wenn dieses gemäß der Vorgehensweise im Kurstext aufgebaut wird. (1 P.)

Vergleicher für Binärzahlen	
Addierer	
Subtrahierer	

Lösung: *Es werden 3 Vergleicher zur Realisierung der drei Entscheidungsboxen benötigt, wobei man auf den dritten Vergleicher verzichten kann, da hier eine 1-Bit-Zahl mit dem Wert 1 verglichen wird. Man braucht 2 Addierer für die Additionen in den Zuständen Z5 und Z6. Subtrahierer werden nicht benötigt.*

Aufgabe 7: Maschinenbefehle (3 Punkte)

In Abschnitt 4.8.2 (Leitwerk) des Kurstextes wird die Abarbeitung verschiedener Arten von Maschinenbefehlen in mehreren Schritten beschrieben. Dabei wurde in manchen Fällen ein neuer Wert des Programmzählers PC dadurch bestimmt, dass er um die Länge des aktuellen Befehls erhöht wurde, in anderen Fällen wurde der PC mit einem neuen Wert überschrieben. Bitte kreuzen Sie in der folgenden Tabelle an, welche Art der Bestimmung bei welcher Art von Maschinenbefehl vorkommen kann. Dabei könnten bei einem Maschinenbefehl auch beide Arten der Bestimmung vorkommen.

Art des Maschinenbefehls	Bestimmung des PC durch	
	Erhöhen um akt. Befehl	Überschreiben
Verknüpfung von Operanden		
Datentransfer Speicher → Prozessor (Lesen)		
Datentransfer Prozessor → Speicher (Schreiben)		
Sprung (unbedingte Verzweigung)		
bedingte Verzweigung		

Lösung:
Die ausgefüllte Tabelle lautet:

Art des Maschinenbefehls	*Bestimmung des PC durch*	
	Erhöhen um akt. Befehl	*Überschreiben*
Verknüpfung von Operanden	*X*	
Datentransfer Speicher → Prozessor (Lesen)	*X*	
Datentransfer Prozessor → Speicher (Schreiben)	*X*	
Sprung (unbedingte Verzweigung)		*X*
bedingte Verzweigung	*X*	*X*

Bei bedingten Verzweigungen kommt der Fall des Erhöhens vor, wenn der Sprung nicht ausgeführt wird, der Fall des Überschreibens hingegen, wenn der Sprung ausgeführt wird.

Aufgabe 8: Fragen zur Rechnerarchitektur (8 Punkte)

Anhand der nachfolgenden Fragen sollen die Unterschiede zwischen Architektur- und Mikroarchitekturtechniken erarbeitet werden.

Kreuzen Sie bitte die zutreffenden Aussagen an bzw. treffen Sie die korrekten Zuordnungen! (je 1 Punkt)

a) Befehlspipelining ist eine Architekturtechnik. ◯

b) Superskalare Prozessoren stellen spezielle Mikroarchitekturen dar. ◯

c) Architekturtechniken müssen durch entsprechende Systemsoftware (z.B. Compiler) unterstützt werden. ◯

d) Mikroarchitekturen können ausschließlich durch die verfügbaren Maschinenbefehle, Adressierungsarten und für den Programmierer sichtbaren Register beschrieben werden. ◯

e) Renaming ist eine Mikroarchitekturtechnik. ◯

f) Dynamisches Befehlsscheduling in Superskalarprozessoren ist eine Architekturtechnik. ◯

g) Statisches Befehlsscheduling wird durch den Compiler vorgenommen. ◯

h) Ordnen Sie den folgenden Prozessortypen zu, ob sie hauptsächlich auf einer Architekturtechnik (A) oder einer Mikroarchitekturtechnik (M) basieren:

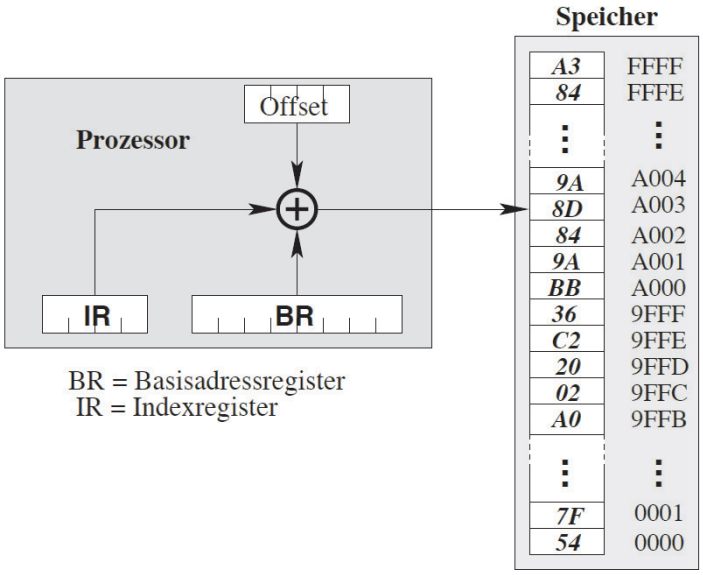
- Mikroprogrammierter CISC-Prozessor .....
- EPIC-Prozessor .....
- Skalarer RISC-Prozessor .....
- Superskalarer RISC-Prozessor .....
- VLIW-Prozessor .....

Lösungsvorschläge

Die Antworten b,c,e und g treffen zu. Bei h) sind nur EPIC und VLIW als A zuzuordnen. Die anderen Prozessoren sind mit M zu kennzeichnen.

Aufgabe 9: Berechnung der effektiven Adresse (6 Punkte)

Ein 8-Bit-Mikroprozessor besitze ein 16-Bit-**Basisadressregister BR** und ein 8-Bit-**Indexregister IR**. Bei einer indizierten Adressierung werde die *effektive Adresse EA* eines Operanden durch die Addition des Inhalts von IR und eines im Befehl angegebenen Offsets zum Inhalt von BR gewonnen. Dabei werden die Inhalte von BR und IR als vorzeichenlose ganze Zahlen, der Offset als vorzeichenbehaftete ganze Zahl im Zweierkomplement aufgefasst. Das folgende Bild verdeutlicht die Bildung der EA und zeigt einen Ausschnitt der Speicherbelegung zum aktuellen Zeitpunkt.



Zahlen ohne besondere Kennzeichnung sind als Hexadezimalzahlen zu interpretieren!

Das Indexregister IR verfüge über die Möglichkeit der automatischen Modifikation (autoinkrement-/autodekrement) um die Werte $n = 1, 2, 3, 4$.

Die Inhalte des Basisadressregisters BR, des Indexregisters IR und des Offsets seien wie folgt belegt:

BR=9F41 IR=C3 Offset=FC

a) Bestimmen Sie die effektive Adresse (EA), die durch einen Lesebefehl mit indizierter Adressierung mit Offset und einer automatischen Postinkrementierung des Indexregisters mit $n = 4$ angesprochen wird. Geben Sie an, welches Datum aus dem Speicher gelesen wird und welchen Wert das Indexregister IR nach der Befehlsausführung hat. Tragen Sie Ihre Lösung in folgende Tabelle ein! (3 Punkte)

	Wert
Effektive Adresse	
Speicherinhalt unter effektiver Adresse	
Registerinhalt von IR nach Befehlsausführung	

b) Bestimmen Sie die effektive Adresse EA, die durch einen Lesebefehl mit indizierter Adressierung mit Offset und einer automatischen Predekrementierung des Indexregisters mit $n=4$ angesprochen wird. Geben Sie an, welches Datum aus dem Speicher gelesen wird und welchen Wert das Indexregister IR nach der Befehlsausführung hat. Tragen Sie Ihre Lösung in folgende Tabelle ein! BR, IR, Offset sollen wie zu Beginn bei Aufgabeneil a) belegt sein. (3 Punkte)

	Wert
Effektive Adresse	
Speicherinhalt unter effektiver Adresse	
Registerinhalt von IR nach Befehlsausführung	

Lösungsvorschläge

Zu a)
Zuerst wird der Offset zum Basisregister BR addiert. Dies ergibt $9F41 - 4 = 9F3D$. Die effektive Adresse EA ergibt sich durch Addition des Inhalts von IR zu diesem Wert EA=$9F3D + C3 = A000$. Der Inhalt der Speicherzelle $A000$ ist BB. Wegen der Postinkrementierung mit $n = 4$ ist das IR nach der Ausführung um 4 zu erhöhen. Der Inhalt des IR ist nach der Ausführung $C3 + 04 = C7$.

	Wert
Effektive Adresse	$A000$
Speicherinhalt unter effektiver Adresse	BB
Registerinhalt von IR nach Befehlsausführung	$C7$

Zu b)
Zuerst wird wieder der Offset zum Basisregister BR addiert, dies ergibt $9F41 - 4 = 9F3D$. Wegen der Predekrementierung mit $n = 4$ ist das IR vor der Ausführung um $n = 4$ zu verringern

$$\begin{array}{rl} C3|_{16} & 1100\,0011|_2 \\ -4|_{16} & 0000\,0100|_2 \\ \hline BF|_{16} & 1011\,1111|_2 \end{array}$$

Dies ergibt $C3 - 04 = BF$. Die EA ergibt sich aus der Addition von $9F3D$ und dem Inhalt des IR

$$\begin{array}{rl} 9F3D|_{16} & 1001\,1111\,0011\,1101|_2 \\ +BF|_{16} & 0000\,0000\,1011\,1111|_2 \\ \hline 9FFC|_{16} & 1001\,1111\,1111\,1100|_2 \end{array}$$

Die effektive Adresse EA ist also $9F3D + BF = 9FFC$. Der Inhalt der Speicherzelle $9FFC$ ist 02. Der Inhalt des IR ist nach der Ausführung BF.

	Wert
Effektive Adresse	$9FFC$
Speicherinhalt unter effektiver Adresse	02
Registerinhalt von IR nach Befehlsausführung	BF

Aufgabe 10: RISC-Pipeline (10 Punkte)

Gegeben sei ein RISC-Prozessor mit einer typischen fünfstufigen Befehls-Pipeline.

a) Beschreiben Sie allgemein den Aufbau (Skizze) und die Funktion der Pipeline! Begründen Sie, warum durch diese Pipeline eine Beschleunigung der Befehlsbearbeitung stattfindet. (2 Punkte)

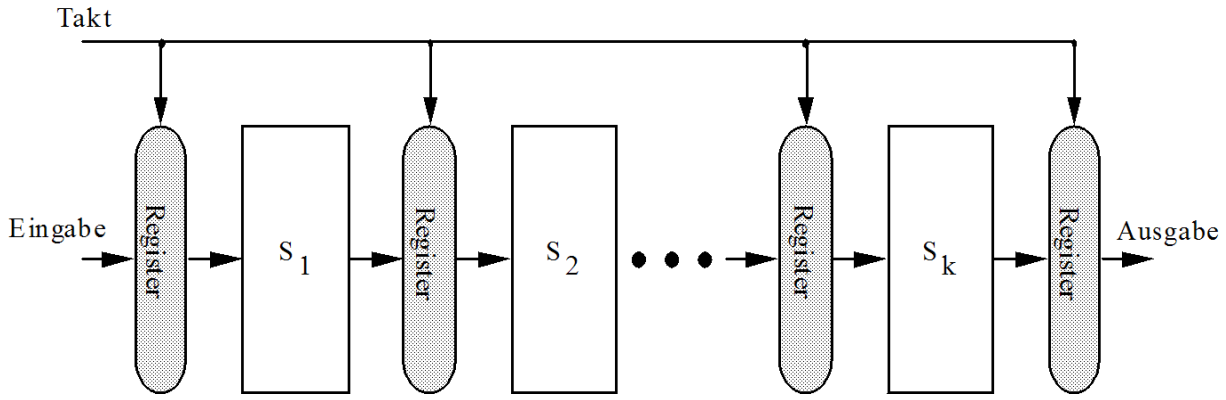
b) Geben Sie die Bezeichnungen der einzelnen Phasen der fünfstufigen Pipeline in der richtigen Reihenfolge der Befehlsabarbeitung an und beschreiben Sie die Funktionalität der einzelnen Phasen. (5 Punkte)

c) RAW-Datenabhängigkeiten von zwei aufeinanderfolgenden Befehlen können bei dem gegebenen RISC-Prozessor nur durch das Einfügen von NOPs behoben werden. Wieviele NOPs müssen eingefügt werden, wenn die Operanden in der zweiten Stufe bereitgestellt werden sollen und erst am Ende der fünften Stufe, nach dem Taktzyklus, im Registersatz zur Verfügung stehen. Begründen Sie Ihre Antwort. (1 Punkt)

d) Erklären Sie, warum die Realisierung von Befehls-Pipelines bei CISC-Prozessoren schwieriger als bei RISC-Prozessoren ist? (2 Punkte)

Lösungsvorschläge

a) Die Pipeline besteht aus fünf Schaltnetzen mit zwischengeschalteten Auffangregistern. Diese trennen die Pipeline-Stufen taktmäßig voneinander ab. Die Befehle werden in Teilschritten abgearbeitet, die überlappen können. Jeder Befehl dauert zwar fünf Takte, durch die Parallelverarbeitung kann aber im Idealfall je Takt ein Befehl beendet werden.



b) **Befehlsbereitstellungs-** oder **IF-Phase** (*Instruction Fetch*): Der Befehl, der durch den Befehlszähler adressiert ist, wird aus dem I-Cache-Speicher (*Instruction Cache*) in einen Befehlspuffer, das sog. Befehlsregister, geladen. Der Befehlszähler wird weitergeschaltet.

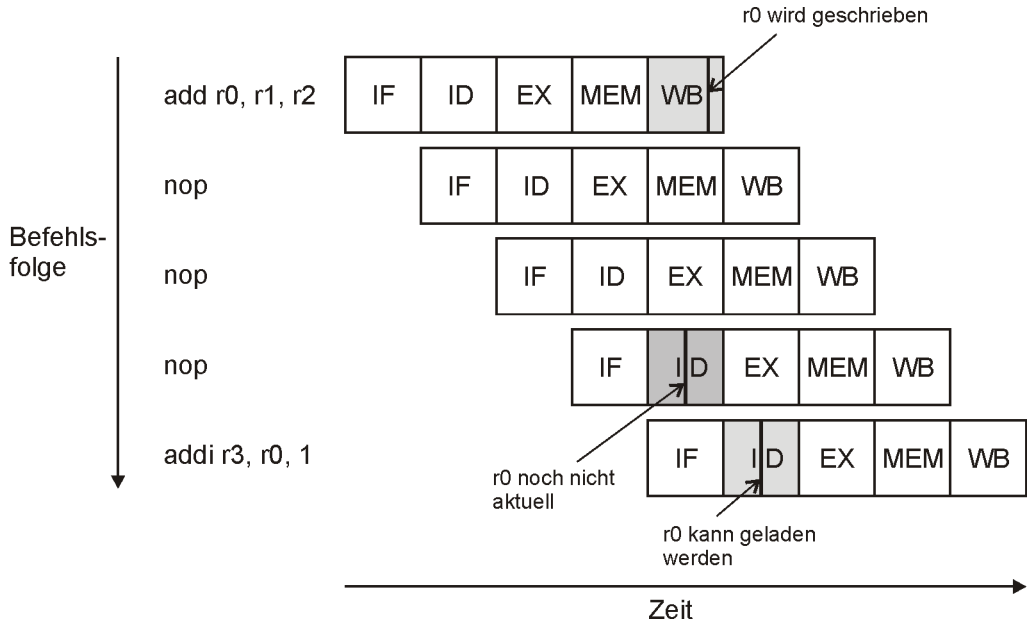
Decodier- und Operandenbereitstellungsphase oder **ID-Phase** (*Instruction Decode/Register Fetch*): Aus dem Operationscode des Maschinenbefehls werden durch ein Decodierschaltnetz Pipeline-interne Steuersignale erzeugt. Die Operanden werden aus Registern (*Register File*) bereit gestellt.

Ausführungs- oder **EX-Phase** (*Execute/Address Calculation*): Die Operation wird von der ALU auf den Operanden ausgeführt. Bei Lade-/Speicherbefehlen berechnet die ALU die effektive Adresse.

Speicherzugriffs- oder **MEM-Phase** (*Memory Access*): Der Speicherzugriff auf den D-Cache (*Data Cache*) wird durchgeführt.

Resultatspeicher- oder **WB-Phase** (*Write Back*): Das Ergebnis wird in ein Architekturregister geschrieben.

c) Es müssen drei NOP-Befehle eingefügt werden.



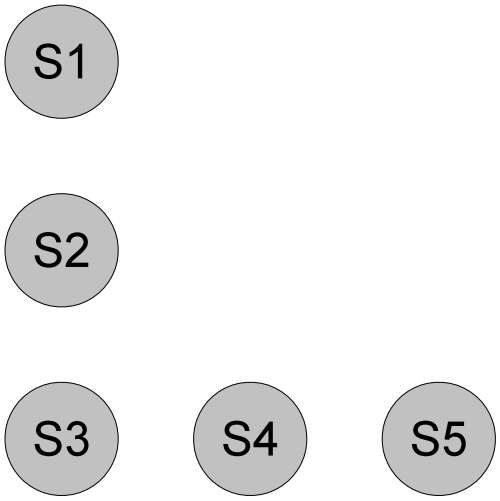
d) Befehle eines CISC-Prozessors können unterschiedlich lang sein. Dabei kann bei dem Laden des ersten Befehlsteils nicht immer direkt festgestellt werden, wie lang der Befehl tatsächlich ist. Man benötigt eine sehr komplexe Dekodiereinheit, die kaum festverdrahtet realisiert werden kann. Üblicherweise werden CISC-Befehle über ein Mikroprogrammsteuerwerk dekodiert und verarbeitet. Heutige, auf x86-Technologie basierende CISC-Prozessoren setzen die einzelnen CISC-Instruktionen in einen oder eine Folge von RISC-ähnlichen Instruktionen um, die dann im Pipeline-Verfahren bearbeitet werden können.

Aufgabe 11: Befehlssequenz analysieren (5 Punkte)

Gegeben sei die folgende Befehlssequenz:

```
S1: ADD R1,R0,R4   ; R1 = R0 + R4
S2: MUL R3,R3,R1   ; R3 = R3 * R1
S3: MUL R6,R7,R5   ; R6 = R7 * R5
S4: SUB R5,R3,R6   ; R5 = R3 - R6
S5: ADD R4,R2,R1   ; R4 = R2 + R1
```

a) Vervollständigen Sie den Abhängigkeitsgraphen der Befehle bzgl. der echten Datenabhängigkeiten! (1 Punkt)



b) Zur Behebung einer Datenabhängigkeit seien **2 Leerbefehle (NOPs)** ausreichend, da der Registerblock in einem einzigen Takt gelesen und geschrieben werden kann.

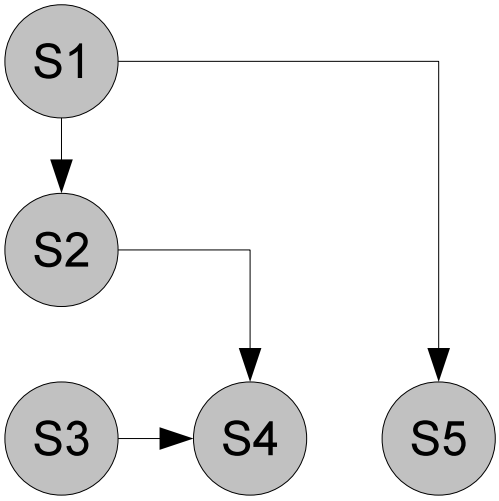
An welchen Stellen der Befehlsfolge müssen NOPs eingefügt werden, damit evtl. vorhandene Datenabhängigkeiten auf Grund der Struktur der DLX–Pipeline nicht zu Fehlberechnungen führen? (1 Punkt)

(Auf die Möglichkeit der Befehlsumordnung werde hier zunächst verzichtet!)

c) Wie kann die Anzahl der notwendigen Leerbefehle durch Befehlsumordnung reduziert werden? Wie viele Leerbefehle werden danach noch benötigt? (3 Punkte)

Lösungsvorschläge

a)



b) Zwischen S1 und S2 sowie zwischen S3 und S4 müssen jeweils zwei Leerbefehle eingefügt werden.

c) Die Befehlssequenz mit Behebung der echten Datenabhängigkeiten durch NOPs lautet also:

S1-NOP-NOP-S2-S3-NOP-NOP-S4-S5

Verschieben von S3 liefert:

S1-S3-NOP-S2-NOP-NOP-S4-S5

Analog liefert das Verschieben von S5:

S1-S3-NOP-S2-S5-NOP-S4

Durch Umordnung der Befehle erreicht man, dass insgesamt nur noch zwei Leerbefehle benötigt werden.

Aufgabe 12: Cache Zugriffe (16 Punkte)

Wir betrachten einen 4-GB-Adressraum, in dem ein Wort aus einem Byte besteht, das einzeln adressiert wird. Das System verfüge über einen Direct Mapped Cache mit 256 Einträgen mit jeweils 16 Wörtern (Bytes). Mit Beginn der Programmausführung ist kein Cache-Eintrag gültig, d.h. der erste Zugriff auf einen Cache-Eintrag führt zu einem *Miss*.

a) (2 Punkte) Aus wie vielen Bits besteht eine Speicheradresse? Stellen Sie die Aufteilung der Speicheradressen in die einzelnen Felder dar und bezeichnen Sie diese! Dem Adressbit der n-ten Position entspricht Wertigkeit 2^n.

Anzahl der Bits einer Speicheradresse: _____

Adressbits	Anzahl der Bits	Bezeichnung
...		
...		
...		

b) (14 Punkte) Gegeben seien folgende Zugriffssequenz und Speicherinhalte. Der Einfachheit halber ist die Kennzeichnung weggelassen, dass es sich um Hexadezimalwerte handelt. R steht für lesenden Zugriff, W für schreibenden Zugriff. Bei einem schreibenden Zugriff wird noch das Datum angegeben, welches in den Speicher geschrieben werden soll. Die Zugriffe sind mit Zahlen von 1 bis 10 durchnummeriert.

Für einige Speicherstellen ist der initiale Inhalt angegeben. Die Speicherstellen sind für Eintragungen in der folgenden Tabelle mit den Buchstaben A bis J versehen.

Zugriffssequenz				**Speicherinhalt**		
Nr.	**R/W**	**Adresse**	**Datum**	**Nr.**	**Adresse**	**Datum**
1	R	C358 01B7		A	0358 0877	73
2	R	C358 01B8		B	0358 0878	15
3	W	0358 0877	FE	C	0358 0879	FF
4	R	0358 0877		D	0358 087A	34
5	R	1234 0877		E	0358 087E	12
6	R	0358 0877		F	0358 087F	66
7	W	1234 0877	18	G	1234 0877	10
8	R	1234 0878		H	1234 0878	CD
9	R	0358 0879		I	C358 01B7	AF
10	R	0358 087F		J	C358 01B8	D8

- Tragen Sie in die folgenden Tabellen die Inhalte des Caches — und zwar jeweils nur das adressierte Byte — und des Speichers in Abhängigkeit von der Zugriffsstrategie für die Zugriffe 3 bis 10 ein (Die Eintragungen für die Zugriffe 1 und 2 sind bereits bis auf die Cache-Tags vorgenommen worden).
- Wenn ein Wort unbekannt ist (z. B. unbekannter oder ungültiger Initalzustand), markieren Sie die entsprechende Zeile mit einem Strich (—).
- Im Falle der Rückschreibestrategie (Write-Back, WB), bei der nur <u>veränderte</u> Cache-Einträge zurückgeschrieben werden, geben Sie zusätzlich an, an welcher Speicherstelle (Nummer der Speicherstelle A bis J) eine Änderung stattfindet. Zusätzlich geben Sie das entsprechend zu speichernde Datum an. Wenn hier kein Speicherzugriff erfolgt, markieren Sie die Stellen ebenfalls mit einem Strich (—).
- Bei allen drei Strategien gehen Sie jeweils davon aus, dass vor dem ersten Zugriff der gesamte Cache ungültig ist und der Speicher den Inhalt der oben abgedruckten Tabelle hat.

Die im Folgenden benutzte Bezeichnung *with Allocation* bedeutet, dass bei einem *Write-Miss* sowohl in den Speicher geschrieben, als auch der Cache-Block aktualisiert wird. Bei *without Allocation* wird bei einem *Write-Miss* nur in den Speicher geschrieben, der Cache-Block allerdings nicht aktualisiert.

In die Zeilen der Tabellen sollen für jeden Zugriff 3 bis 10 die folgenden Angaben eingetragen werden:

- **hit/miss**: jeweils h oder m, je nachdem, ob ein Hit oder ein Miss auftritt
- **Cache-Tag**: der nach dem Zugriff auf den Cache aktuelle Tag des Cache-Blocks
- **Cache-Datum**: Inhalt des ausgewählten Cache-Eintrages nach dem Zugriff
- **Speicher**: Inhalt des Speichers nach dem Zugriff auf die selektierte Speicherstelle

Zusätzlich sollen bei der Strategie Write-Back folgende Angaben gemacht werden:

- **WB-Num**: Buchstabe A bis J der Adresse, die ggf. durch die Write-Back-Strategie durch den aktuellen Zugriff verändert wird. Findet keine Veränderung statt, tragen Sie hier einen Strich (-–) ein.
- **WB-Datum**: Wert, der durch den Schreibzugriff in die selektierte Speicherstelle geschrieben wird.

(i) Write-Through with Allocation

Zugriff	1	2	3	4	5	6	7	8	9	10
hit/miss	m	h								
Cache-Tag										
Cache-Datum	AF	D8								
Speicher	AF	D8								

(ii) Write-Through without Allocation

Zugriff	1	2	3	4	5	6	7	8	9	10
hit/miss	m	h								
Cache-Tag										
Cache-Datum	AF	D8								
Speicher	AF	D8								

(iii) Write-Back with Allocation

Zugriff	1	2	3	4	5	6	7	8	9	10
hit/miss	m	h								
Cache-Tag										
Cache-Datum	AF	D8								
Speicher	AF	D8								
WB-Num.	–	–								
WB-Datum	–	–								

Zugriffssequenz

Nr.	R/W	Adresse	Datum
1	R	C358 01B7	
2	R	C358 01B8	
3	W	0358 0877	FE
4	R	0358 0877	
5	R	1234 0877	
6	R	0358 0877	
7	W	1234 0877	18
8	R	1234 0878	
9	R	0358 0879	
10	R	0358 087F	

Speicherinhalt

Nr.	Adresse	Datum
A	0358 0877	73
B	0358 0878	15
C	0358 0879	FF
D	0358 087A	34
E	0358 087E	12
F	0358 087F	66
G	1234 0877	10
H	1234 0878	CD
I	C358 01B7	AF
J	C358 01B8	D8

Lösungsvorschläge

a) Anzahl der Bits einer Speicheradresse: **32**

Adressbits	Anzahl der Bits	Bezeichnung
31 . . . 12	**20**	**Tag**
11 . . . 4	**8**	**Index**
3 . . . 0	**4**	**Byteauswahl**

b) (i)

Zugriff	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
hit/miss	m	h	m	h	m	m	m	h	m	h
Cache-Tag	C3580	C3580	03580	03580	12340	03580	12340	12340	03580	03580
Cache-Datum	AF	D8	FE	FE	10	FE	18	CD	FF	66
Speicher	AF	D8	FE	FE	10	FE	18	CD	FF	66

(ii)

Zugriff	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
hit/miss	m	h	m	m	m	m	m	m	m	h
Cache-Tag	C3580	C3580	–	C3580	12340	03580	[03580]	12340	03580	03580
Cache-Datum	AF	D8	–	FE	10	FE	[FE]	CD	FF	66
Speicher	AF	D8	FE	FE	10	FE	18	CD	FF	66

In Spalte 7 ist am Index ein anderer Tag gespeichert, als zugegriffen wird. Hier ist sowohl keine als auch die Angabe des alten Tags bzw. Datums zulässig.

(iii)

Zugriff	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
hit/miss	m	h	m	h	m	m	m	h	m	h
Cache-Tag	C3580	C3580	03580	03580	12340	03580	12340	12340	03580	03580
Cache-Datum	AF	D8	FE	FE	10	FE	18	CD	FF	66
Speicher	AF	D8	73	73	10	FE	10	CD	FF	66
WB-Num.	–	–	–	–	A	–	–	–	G	–
WB-Datum	–	–	–	–	FE	–	–	–	18	–

Aufgabe 13: Amdahl's Gesetz (5 Punkte)

Gegeben sei ein Parallelrechner mit 100 gleichwertigen Prozessoren.

a) Wie hoch darf der sequentielle Anteil a nach Amdahl's Gesetz maximal sein, damit noch ein Speedup von $S = 80$ erreicht wird? (2 Punkte)

   Es ist ausreichend, wenn Sie a als ganzzahligen Bruch angeben.

   a=————————

b) Wie hoch ist die Effizienz E, wenn nur ein Speedup von $S = 40$ gemessen wird? (1 Punkt)

   $E =$ .....

c) Welcher Speedup und welche Effizienz wird bei einem sequentiellen Anteil von $a = 2\%$ erreicht? (2 Punkte)

   Es ist ausreichend, wenn Sie S und E als ganzzahligen Bruch angeben.

   S=————————

Anmerkung: Leiten Sie Ihre Ergebnisse her.

Lösungsvorschläge

Zu a): $S = \frac{1}{\frac{1-a}{100}+a} = 80 \qquad \Rightarrow a = \frac{1}{396} \approx 0,2525\%$

Zu b): $E = \frac{S}{p} = 0,4 = 40\%$

Zu c): Es wird $S = \frac{1}{0{,}02+\frac{1-0{,}02}{100}} = \frac{5000}{149} \approx 33,6$ bzw. E $\approx 33,6$ % erreicht.