

**Modul Computersysteme
Prüfungsklausur SS 2011**

Lösungsvorschläge

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

Aufgabe 1 (12 Punkte):

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

	X_1				
X_2	1	1	0	0	
	1	1	1	0	X_4
	0	0	1	1	
	1	0	0	1	
	X_3				

Kreuzen Sie bitte an, welche der folgenden Terme Primterme der Schaltfunktion f darstellen. (3 P.)

X_1

X_1X_2

$X_1\bar{X}_3\bar{X}_4$

X_3X_4

$\bar{X}_1\bar{X}_2\bar{X}_3$

$\overline{X_3X_4}$

b) Gegeben ist die folgende Primtermtabelle, wobei alle Primterme P_1 bis P_5 gleiche Kosten haben sollen.

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

Bitte kreuzen Sie an, welche der folgenden Aussagen zutreffen. (3 P.)

Primterm P3 ist kein Kernimplikant.

Primterm P2 dominiert Primterm P1.

Das Minimalpolynom besteht aus zwei Primtermen.

Das Minimalpolynom besteht aus drei Primtermen.

Primterm P2 ist ein Kernimplikant.

Keiner der Primimplikanten ist ein Kernimplikant.

c) Bitte kreuzen Sie an, welche der folgenden Aussagen zutreffen. (4 P.)

Es gibt 2^{2^n} Schaltfunktionen $f : \{0, 1\}^n \rightarrow \{0, 1\}$.

Es gibt für jede Schaltfunktion genau ein Minimalpolynom.

Es gibt eine Schaltfunktion, für die kein Minimalpolynom existiert.

Ein Primterm ist ein Monom, in dem alle Variablen als Literale vorkommen.

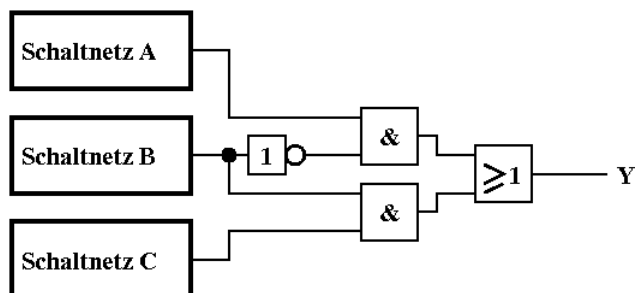
Ein Kernimplikant muss nicht unbedingt ein Primimplikant sein.

Ein Minimalpolynom muss stets alle Kernimplikanten einer Schaltfunktion als Monome enthalten.

Die Kosten $L(e)$ eines boole'schen Ausdrucks e sind definiert als die Anzahl von Vorkommen der Zeichen \wedge , \vee und \sim in e .

Die Formelgröße einer Schaltfunktion f ist definiert als die Länge der kürzesten disjunktiven Normalform für f .

d) Wieviele Gatter müsste man zu dem folgenden Schaltnetz hinzufügen, wenn man den beim Finden des quasi-korrespondierenden boole'schen Ausdrucks notwendigen Umbau durchführt? (2 P.)
Hinweise: Die Schaltnetze A, B und C haben jeweils Kosten 3, 4, 5, und enthalten auch die Eingänge. Es werden keine Eingänge invertiert. Wenn das Schaltnetz einen korrespondierenden boole'schen Ausdruck hat, d.h. wenn keine Gatter hinzugefügt werden müssen, dann geben Sie als Antwort „Null“ an.



Anzahl der hinzuzufügenden Gatter:

Lösung: a) Der zweite, der dritte und der fünfte Term sind Primterme der Schaltfunktion f im Karnaugh-Diagramm.
 b) Die erste, vierte und sechste Aussage treffen zu.
 c) Die erste, vierte, sechste und siebte Aussage treffen zu.
 d) Das Schaltnetz B muss verdoppelt werden, man muss also 4 Gatter hinzufügen, da B Kosten 4 hat, und keine Eingänge invertiert werden.

Aufgabe 2 (7 Punkte):

a) Rechnen Sie jeweils aus (5 P.):

$$[0110]_2 = \boxed{}$$

$$[1110]_2 = \boxed{}$$

$$\langle 1101 \rangle_2 = \boxed{}$$

$$\text{bin}_6(17) = \boxed{}$$

$$\text{twoc}(-2) = \text{mit 8 Stellen } \boxed{}$$

b) Bitte ergänzen Sie die folgenden Gleichungen. (2 P.)

$$\langle a_4 a_3 a_2 a_1 a_0 \rangle = \langle a_4 a_3 \rangle \cdot \boxed{} + a_2 \cdot 2^2 \boxed{} \langle a_1 a_0 \rangle$$

$$[a_4 a_3 a_2 a_1 a_0] = \boxed{} a_4 \cdot \boxed{} + \langle a_3 a_2 \rangle \cdot \boxed{} + \langle a_1 a_0 \rangle$$

Lösung:

a)

$$[0110]_2 = 4 + 2 = 6$$

$$[1110]_2 = -8 + 4 + 2 = -2$$

$$\langle 1101 \rangle_2 = 8 + 4 + 1 = 13$$

$$\text{bin}_6(17) = 010001$$

$$\text{twoc}_8(-2) = 1 \dots 10$$

b)

$$\langle a_4 a_3 a_2 a_1 a_0 \rangle = \langle a_4 a_3 \rangle \cdot \boxed{2^3} + a_2 \cdot 2^2 \boxed{+} \langle a_1 a_0 \rangle$$

$$[a_4 a_3 a_2 a_1 a_0] = \boxed{-} a_4 \cdot \boxed{2^4} + \langle a_3 a_2 \rangle \cdot \boxed{2^2} + \langle a_1 a_0 \rangle$$

Aufgabe 3 (6 Punkte):

a) Erstellen Sie die Wertetabelle für einen signalisierenden 1-Bit Coder. (4 P.)

a_0	a_1	b_2	b_1	b_0
0	0			
0	1			
1	0			
1	1			

b) Erstellen Sie die Wertetabelle für einen Halbaddierer. (2 P.)

a	b	c	s
0	0		
0	1		
1	0		
1	1		

Lösung:

a) Die Wertetabelle lautet

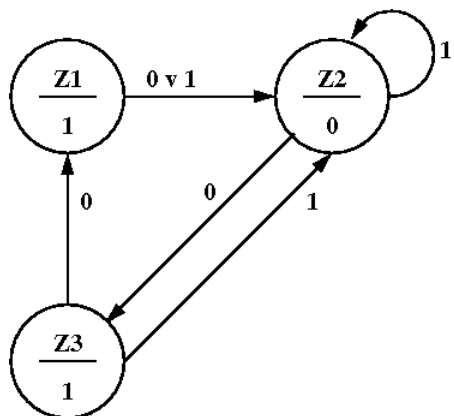
a_0	a_1	b_2	b_1	b_0
0	0	0	1	X
0	1	0	0	1
1	0	0	0	0
1	1	1	X	X

b) Die Wertetabelle lautet

a	b	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

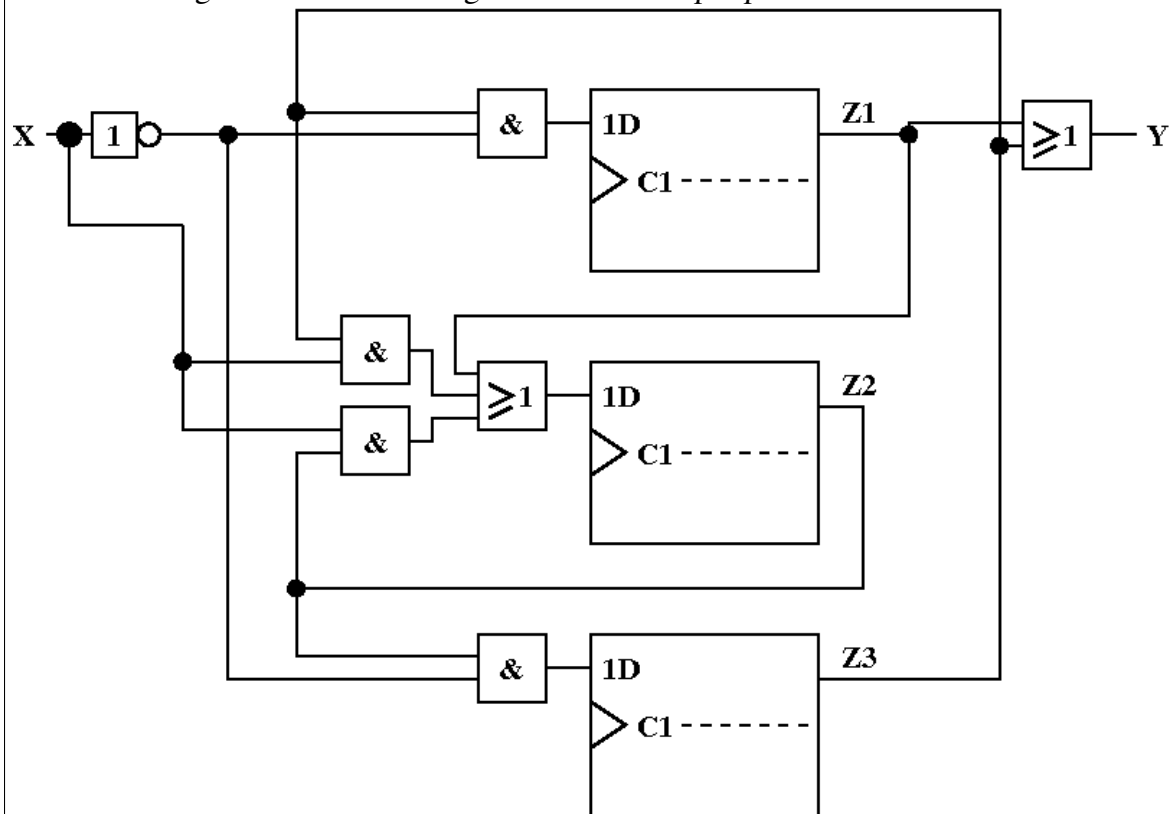
Aufgabe 4 (6 Punkte):

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.



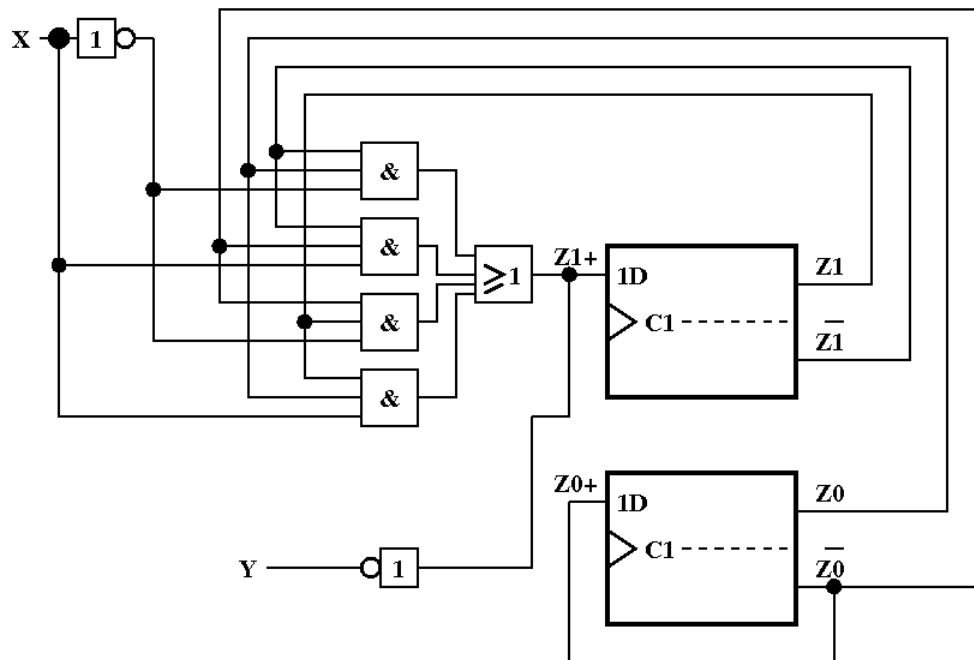
Lösung:

Das vollständige Schaltwerk benötigt neben den 3 Flipflops 7 Gatter.



Aufgabe 5 (9 Punkte):

Gegeben ist ein Schaltwerk mit D-Flipflops (x : Eingabe, y : Ausgabe, Zustand in $z_1 z_0$ gespeichert) gemäß folgender Abbildung.



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. Genauer handelt es sich um einen sog. vorauslaufenden Moore-Automaten, da y nur vom zukünftigen Zustand abhängt.

b) Stellen Sie Zustandstabelle des Schaltwerks auf. (8 P.)

Hinweis: Bestimmen Sie zunächst als Gleichungen, wie z_1^+ und z_0^+ von x , y , z_1 , z_0 und ihren Inversen abhängen.

Lösung:

Die Gleichungen lauten

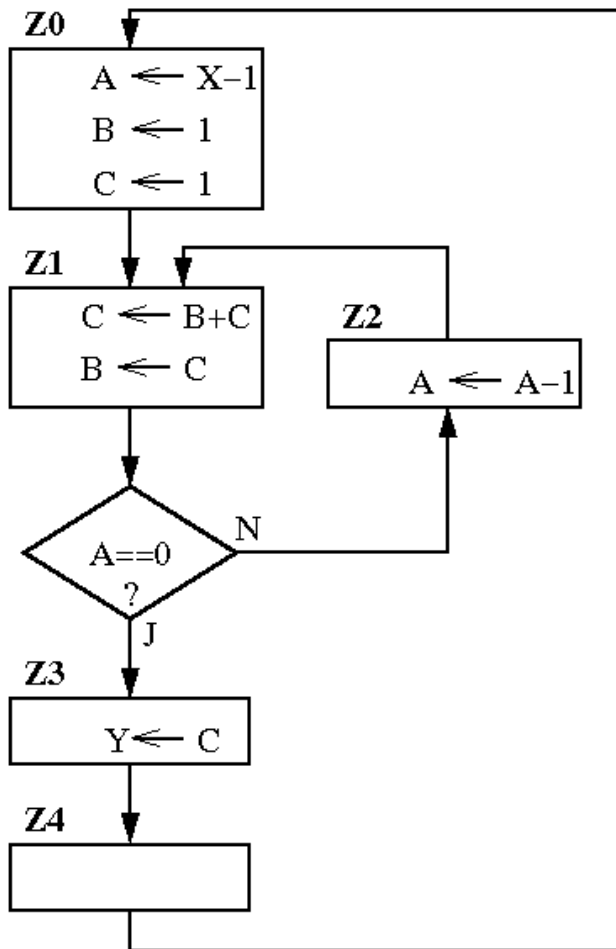
$$\begin{aligned} z_0^+ &= \bar{z}_0 \\ z_1^+ &= \bar{z}_1 z_0 \bar{x} \vee \bar{z}_1 \bar{z}_0 x \vee z_1 \bar{z}_0 \bar{x} \vee z_1 z_0 x \\ y &= z_1^+ \end{aligned}$$

Damit ergibt sich die Zustandstabelle wie folgt:

z_1	z_0	x	z_1^+	z_0^+	y
0	0	0	0	1	1
0	0	1	1	1	0
0	1	0	1	0	0
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	0	1	1
1	1	0	0	0	1
1	1	1	1	0	0

Aufgabe 6 (10 Punkte):

Gegeben sei das folgende ASM-Diagramm eines komplexen Schaltwerks, wobei die Eingabe aus dem Signal X besteht, es Register für Zahlen A , B , C und Y gibt, und die Ausgabe aus dem Wert des Registers Y besteht, wenn Zustand Z_4 erreicht wird.



a) Welchen Wert hat Y im Zustand Z_4 , wenn das Schaltwerk im Zustand Z_0 mit $X = 4$ gestartet wurde? (3 P.)

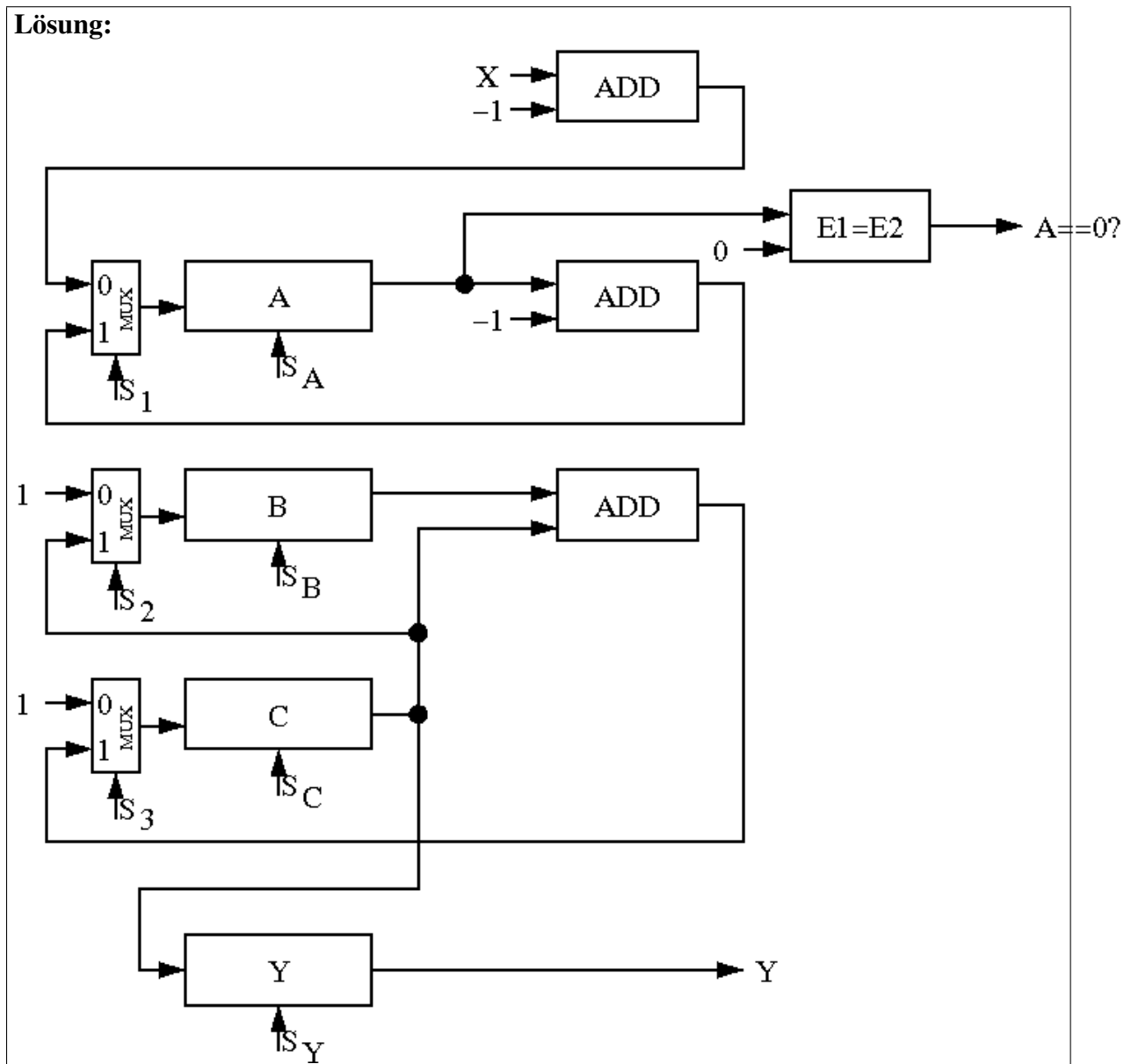
Lösung: Das Schaltwerk berechnet die Fibonacci-Zahlen. Am Ende von Zustand Z_1 haben B, C , und A die folgenden Werte (wobei A den Wert jeweils schon am Anfang des Zustands hat):

B	C	A
1	2	3
2	3	2
3	5	1
5	8	0

Nun wird zu Zustand Z_3 und Zustand Z_4 verzweigt, so dass Y den Wert 8 hat.

b) Vervollständigen Sie das zugehörige Operationswerk. (5 P.)

Lösung:



c) Vervollständigen Sie die Steuertabelle. (2 P.)

Lösung:

	Z_0	Z_1	Z_2	Z_3	Z_4
$S1$	0	X	1	X	X
$S2$	0	1	X	X	X
$S3$	0	1	X	X	X

Aufgabe 7 Fragen zur Rechnerarchitektur (8 Punkte):

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

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

- a) Befehlspipelining ist eine Architekturtechnik.
- b) Superskalare Prozessoren stellen spezielle Mikroarchitekturen dar.
- c) Architekturstechniken 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 Mikroarchitekturstechnik.
- f) Dynamisches Befehlsscheduling in Superskalarprozessoren ist eine Architekturstechnik.
- g) Statisches Befehlsscheduling wird durch den Compiler vorgenommen.
- h) Ordnen Sie den folgenden Prozessortypen zu, ob sie hauptsächlich auf einer Architekturstechnik (A) oder einer Mikroarchitekturstechnik (M) basieren:
 - Mikroprogrammierter CISC-Prozessor
 - EPIC-Prozessor
 - Skalärer RISC-Prozessor
 - Superskalärer RISC-Prozessor
 - VLIW-Prozessor

Lösung: 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 8 Gleitkommadarstellung (8 Punkte):

Gegeben sei die Dezimalzahl $Z_{10} = -101,75$.

- a) Stellen Sie die Zahl Z_{10} als Dualzahl mit Vorzeichenbit und Betrag dar. (1 P.)

$$Z_2 = \dots\dots\dots, \dots\dots\dots_2$$

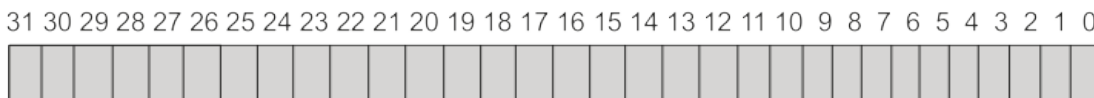
- b) Stellen Sie nun die Zahl Z_{10} als gebrochene, normalisierte Zahl Z_{32} zur Basis 2 im 32-bit-Format des IEEE-754-Standards dar und tragen Sie dazu die entsprechenden Werte für Vorzeichen, verschobenen Exponenten (auch Charakteristik oder biased exponent genannt) und Mantisse in das folgende Schema ein. (2 P.)

$$Z_{32} = (-1)^{\dots\dots\dots} \cdot 2^{\dots\dots\dots 10} \cdot (\dots\dots, \dots\dots\dots)_2$$

- c) Geben Sie den verschobenen Exponenten als 8-Bit-Dualzahl an! (1 P.)

$$e = \dots\dots\dots_2$$

- d) Tragen Sie die Zahl Z_{32} in den folgenden Bitrahmen ein und bezeichnen Sie die unterscheidbaren Bitfelder. (3 P.)



- e) Geben Sie die Zahl Z_{32} als Hexadezimalzahl Z_{16} an. (1 P.)

$$Z_{16} = \dots\dots\dots$$

Hinweis:

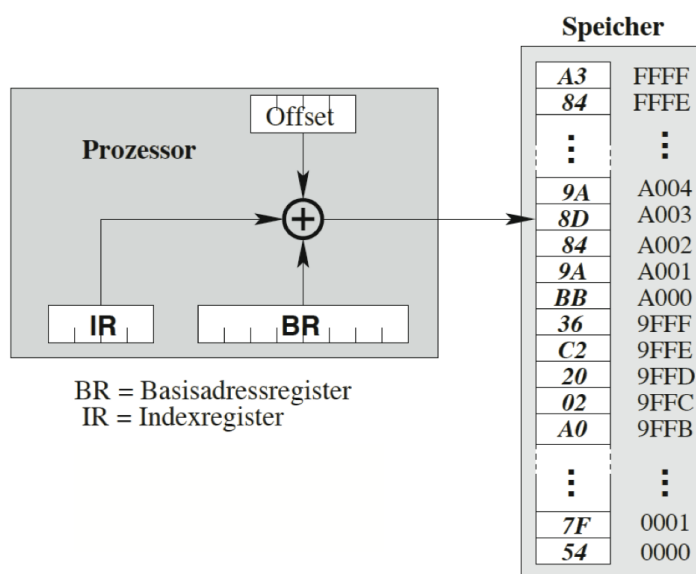
Die Indizes $\dots_2, \dots_{10}, \dots_{16}$ und \dots_{32} kennzeichnen jeweils Zahlen im Binär-, Dezimal- sowie Hexadezimal-System bzw. im 32-bit-IEEE-Format.

Lösung: Zu a): $Z_2 = -1100101,11_2$
 Zu b): $Z_{32} = (-1)^1 \cdot 2^{133_{10}} \cdot (1,100101110)_2$
 Zu c): $e=10000101_2$

Zu d):
 Zu e): $Z_{16} = C2CB8000_{16}$

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:

$$\text{BR}=\text{9F41} \quad \text{IR}=\text{C3} \quad \text{Offset}=\text{FC}$$

- 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 P.)

	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! (3 P.)

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

Lösung:

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

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

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

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

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 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 P.)

S1

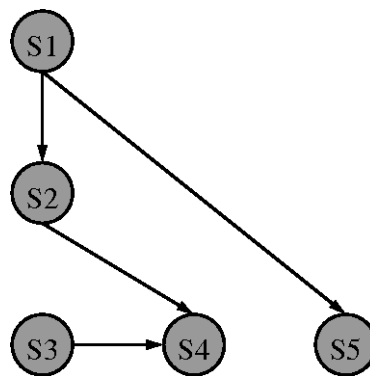
S2

S3

S4

S5

Lösung:



- 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 führt? (1 P.)

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

Lösung: Zwischen S1 und S2 sowie zwischen S3 und S4 müssen jeweils zwei Leerbefehle eingefügt werden.

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

Lösung: 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 11 Sprungvorhersage mit 2-Bit Sättigungs-Prädiktoren (12 Punkte):

Gegeben sei das folgende DLX-Programm (R0 enthält stets den Wert 0). Betrachten Sie jeweils vier Durchläufe:

```

1  INIT:   LOAD R1,#0 ; R1=0
2         LOAD R2,#2 ; R2=2
3  START:  CMP R1,R0  ; R1==0?
4         BRNZ L1   ; if ( R1 !=R0 ) goto L1
5         LOAD R1,#1 ; R1=1
6  L1:    CMP R1,R2  ; R1==R2?
7         BRNZ L2   ; if ( R1 !=R2 ) goto L2
8         LOAD R1,#0 ; R1=0
9         BRA START ; goto START
10 L2:    LOAD R1,#2 ; R1=2
11         BRA START ; goto START
    
```

a) Erstellen Sie eine Tabelle der Sprungverläufe, die beim Ablauf dieses Programms auftreten. (2 P.)

Durchlauf	BRNZ L1	BRNZ L2	BRA START	BRA START
1				
2				
3				
4				

b) Für die beiden bedingten Sprünge soll jeweils ein Zwei-Bit-Prädiktor mit Sättigungszähler, der im Zustand Weakly Not Taken (WNT) initialisiert ist, mit Hilfe der beigefügten Tabelle analysiert werden. Markieren Sie dabei die falschen Vorhersagen! Wie viele falsche Vorhersagen gibt es? (4 P.)

Durchlauf	BRNZ L1			BRNZ L2		
	Vorher- sage	- Sprung -	neuer Zustand	Vorher- sage	- Sprung -	neuer Zustand
1						
2						
3						
4						

Anzahl der falschen Vorhersagen:

- c) Betrachten Sie nun einen (2,2)-Korrelationsprädiktor, dessen Schieberegister mit (NT,NT) vorbelegt ist und dessen vier Prädiktoren ebenfalls mit WNT initialisiert seien. Füllen Sie die beigefügte Tabelle aus und markieren Sie dabei die falschen Vorhersagen! Wie viele falsche Vorhersagen gibt es? (6 P.)

Durchlauf	BRNZ L1			BRNZ L2		
	Vorher- sage	- Sprung -	neuer Zustand	Vorher- sage	- Sprung -	neuer Zustand
1						
2						
3						
4						

Anzahl der falschen Vorhersagen:

Lösung: Zu a):

Durchlauf	BRNZ L1	BRNZ L2	BRA START	BRA START
1	NT	T	—	T
2	T	NT	T	—
3	NT	T	—	T
4	T	NT	T	—

Zu b):

Durchlauf	BRNZ L1			BRNZ L2		
	Vorher- sage	- Sprung -	neuer Zustand	Vorher- sage	- Sprung -	neuer Zustand
1	WNT	NT	SNT	WNT	T	WT
2	SNT	T	WNT	WT	NT	WNT
3	WNT	NT	SNT	WNT	T	WT
4	SNT	T	WNT	WT	NT	WNT

Sechs Vorhersagen sind fehlerhaft (fett gedruckt).

Zu c):

Durchlauf	BRNZ L1			BRNZ L2		
	Vorher- sage	- Sp. -	neuer Zustand	Vorher- sage	- Sp.-	neuer Zustand
1	NT-NT:WNT	NT	NT-NT:SNT	NT-NT:SNT	T	NT-NT:WNT
2	NT-T:WNT	T	NT-T:WT	T-T:WNT	NT	T-T:SNT
3	T-NT:WNT	NT	T-NT:SNT	NT-NT:WNT	T	NT-NT:WT
4	NT-T:WT	T	NT-T:ST	T-T:SNT	NT	T-T:SNT

Aufgrund der Korrelationen der Sprünge sind nur noch drei Vorhersagen fehlerhaft (fett gedruckt).

Vorgehensweise: Wir beginnen im 1. Durchlauf BRNZ L1 mit der Vorhersage NT-NT:WNT. Da die Vorhersage zutrifft, wechselt der Zustand des NT-NT-Prädiktors auf SNT. Die Vorhersage für BRNZ L2 ergibt sich aus NT-NT:SNT. Dies trifft aber nicht zu, weshalb der Zustand des NT-NT-Prädiktors auf WNT gesetzt werden muss. Im 2. Durchlauf müssen wir den NT-T-Prädiktor benutzen. Dieser ist mit WNT initialisiert. Da der Sprung genommen wird, ist die Vorhersage falsch und als neuer Zustand wird NT-T:WT gesetzt. Für den zweiten Sprungbefehl BRNZ L2 müssen wir nun die Vorhersage des T-T-Prädiktors verwenden. Dieser ist ebenfalls mit WNT initialisiert und wegen der korrekten Prognose ändert sich sein Zustand auf T-T:SNT. Beim 3. und 4. Durchlauf verfahren wir genauso und erhalten schließlich die oben angegebene Tabelle. Bitte dabei beachten, dass die aktuelle Prognose sich immer aus dem (letzten neuen) Zustand des betreffenden Prädiktors ergibt.

Aufgabe 12 Zugriffszeit mit L_1 -Cache (6 Punkte):

Ein Computersystem verfüge über eine L_1 -Cache, der zum Puffern von Hauptspeicherinhalten (HS) genutzt wird. Die Zugriffszeiten betragen $t_{Cache} = 2ns$ und $t_{HS} = 100ns$. Der Hauptspeicher habe genügend Kapazität für alle Daten und kann direkt (d.h. ohne zusätzliche Latenzzeit bei einem Cache-Fehlzugriff) zugegriffen werden.

Bei der Ausführung von drei typischen Benchmark-Programmen P_i wurden folgende Trefferraten h_i gemessen:

h_1	80%
h_2	85%
h_3	90%

Bestimmen Sie im Folgenden jeweils t_S als Funktion von h_1, h_2, h_3, t_{Cache} und t_{HS} !

Die exakten numerischen Ergebnisse sind nicht zum Erreichen der max. Punktzahl erforderlich.

- Wie groß ist die mittlere Speicherzugriffszeit t_S , wenn alle drei Benchmark-Programme gleich gewichtet werden? (2 P.)
- Wie groß ist die mittlere Speicherzugriffszeit t_S , wenn die Benchmark-Programme im Häufigkeitsverhältnis $P_1 : P_2 : P_3 = 2 : 2 : 4$ ausgeführt werden? (4 P.)

Lösung: Zu a): $t_S = \frac{1}{3}(h_1 \cdot t_{Cache} + (1 - h_1) \cdot t_{HS} + h_2 \cdot t_{Cache} + (1 - h_2) \cdot t_{HS} + h_3 \cdot t_{Cache} + (1 - h_3) \cdot t_{HS}) = \frac{1}{3}((h_1 + h_2 + h_3) \cdot t_{Cache} + (3 - h_1 - h_2 - h_3) \cdot t_{HS}) = \frac{1}{3}(5,1 + 45)ns = 16,7 ns$

Zu b): Mit den angegebenen Häufigkeiten ergibt sich: $T_S = (\frac{h_1}{4} + \frac{h_2}{4} + \frac{h_3}{2}) \cdot t_{Cache} + (1 - \frac{h_1}{4} - \frac{h_2}{4} - \frac{h_3}{2}) \cdot t_{HS} = (0,8625 \cdot 2 + 0,1375 \cdot 100)ns = 15,475 ns$

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 $S = 90$ erreicht wird? (2 P.)

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

$$a = \text{—————}$$

- b) Wie hoch ist die Effizienz E , wenn nur ein Speedup von $S = 60$ gemessen wird? (1 P.)

$$E = \text{.....}$$

- c) Welcher Speedup und welche Effizienz wird bei einem sequentiellen Anteil von $a = 5\%$ erreicht? (2 P.)

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

$$S = \text{—————}$$

Lösung: Zu a): $S = \frac{1}{\frac{1-a}{100} + a} = 90 \quad \Rightarrow \quad a = \frac{1}{891} \approx 0,1122\%$

Zu b): $E = \frac{S}{p} = 0,6 = 60\%$

Zu c): Es wird $S = \frac{1}{0,05 + \frac{1-0,05}{100}} = \frac{2000}{119} \approx 16,8$ bzw. $E \approx 16,8\%$ erreicht.