



Bitte hier unbedingt  
Matrikelnummer und  
Adresse eintragen,  
sonst keine Bearbeitung  
möglich.

--	--	--	--	--	--	--	--

Postanschrift: FernUniversität, D-58084 Hagen

Name, Vorname

Straße, Nr.

PLZ, Wohnort

## PRÜFUNGSKLAUSUR FÜR DEN BACHELOR-STUDIENGANG INFORMATIK WS 2009/2010

**KLAUSUR:** 25211 Computersysteme (Bachelor)  
**DATUM:** 20.03.2010  
**UHRZEIT:** 10:00 – 13:00 Uhr  
**KLAUSURORT:**

### Bearbeitungshinweise (Bitte vor Arbeitsbeginn durchlesen!)

- Schreiben Sie Ihre Klausur bitte nicht mit Bleistift.
- Geben Sie bitte im oberen Feld Ihren Namen, Matrikel-Nr. und Adresse vollständig an, und schreiben Sie Ihren Namen und ihre Matrikelnummer auf jedes Lösungsblatt, das Sie abgeben.
- Die Reihenfolge, in der Sie die Aufgaben/Teilaufgaben lösen, ist Ihnen freigestellt. Kreuzen Sie in der Tabelle (s.u.) in der grau hinterlegten Zeile an, welche Aufgaben Sie bearbeitet haben.
- Bei jeder Aufgabe ist die erreichbare Höchstpunktezahl vermerkt. Sie haben die Klausur bestanden, wenn Sie **50** Punkte erreichen.
- Irgendwelche Hilfsmittel wie Studienbriefe, Glossare, Bücher, Aufzeichnungen, Taschenrechner, etc., dürfen während der Klausur nicht benutzt werden. Ihre Benutzung sowie andere Täuschungsversuche führen dazu, dass Ihre Klausur mit der Note 5 bewertet wird. Neben Schreibgerät und Papier ist die Benutzung eines doppelseitig beschriebenen DIN A4 Blattes mit eigenen Notizen (handschriftlich oder maschinell) gestattet.

<b>Aufsicht:</b>	<b>Bemerkungen:</b>
<b>Datum, Unterschrift:</b>	

<b>Aufgabe</b>	1	2	3	4	5	6	7	8	9	10	11	12	13	
<b>Bearbeitet</b>														

<b>Maximale Punktzahl</b>	10	4	4	14	18	8	3	5	3	4	8	8	11	<b>100</b>
<b>Erreichte Punktzahl</b>														
<b>Korrektur</b>														

<b>Prüfergebnis/Note</b>	
--------------------------	--

1. Prüfer: Prof. Dr. Wolfram Schiffmann	2. Prüfer: Prof. Dr. Jörg Keller
Datum, Unterschrift	Datum, Unterschrift



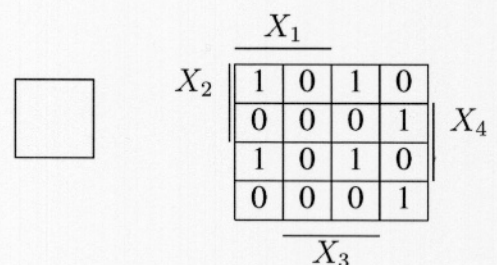
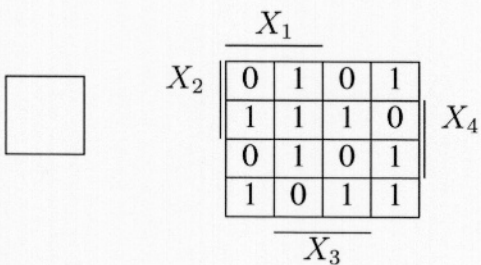
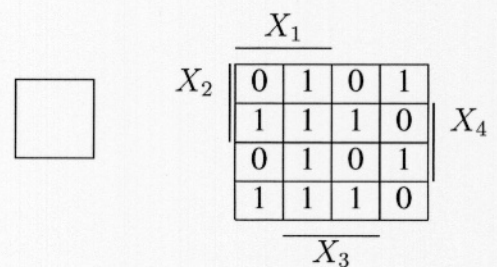
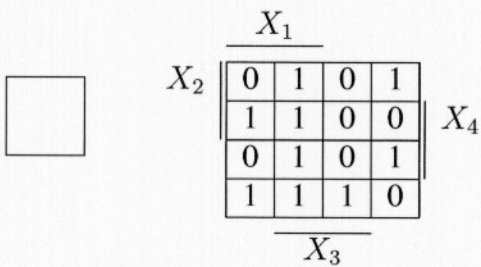
**Aufgabe 1 (12 Punkte):**

a) Gegeben ist die folgende Schaltfunktion durch ihre Wertetabelle.

$X_1X_2X_3X_4$	$g(X_1X_2X_3X_4)$
0000	0
0001	1
0010	1
0011	0
0100	1
0101	0
0110	0
0111	1
1000	1
1001	0
1010	1
1011	1
1100	0
1101	1
1110	1
1111	1

Platz für Nebenrechnung:

Kreuzen Sie bitte an, welches der folgenden Karnaugh-Diagramme zu der Wertetabelle passt. (2 P.)



Name:

Vorname:

Matr.-Nr.:

3

b) Gegeben ist das folgende Karnaugh-Diagramm.

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

Platz für Nebenrechnung:

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

Kreuzen Sie bitte an, welche der folgenden Terme Primterme dieses Karnaugh-Diagramms darstellen. (4 P.)

$X_1$

$X_1X_4$

$X_1X_4\bar{X}_3$

$\bar{X}_1\bar{X}_2\bar{X}_3X_4$

$\bar{X}_2\bar{X}_3X_4$

$\bar{X}_1X_2\bar{X}_3\bar{X}_4$

$\bar{X}_1X_2\bar{X}_3X_4$

$X_1X_2X_3$



Name:

Vorname:

Matr.-Nr.:

4

c) Gegeben ist die folgende reduzierte Primtermtabelle, d.h. die Kernprimterme und die von ihnen überdeckten Minterme sind bereits entfernt.

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

Bitte kreuzen Sie an, welche der folgenden Aussagen zutreffen. (2 P.)

Primterm P1 dominiert Primterm P2.

Primterm P2 dominiert Primterm P1.

Primterm P2 wird von keinem anderen Primterm dominiert.

Die Restüberdeckung muss aus mindestens 3 Primtermen bestehen.

Primterm P3 wird von Primterm P2 dominiert.



Name:

Vorname:

Matr.-Nr.:

5

d) Gegeben ist folgende Primtermtabelle. Bitte kreuzen Sie links von den Primtermen an, welche der Primterme zum Minimalpolynom gehören. Wir machen die Annahme, dass die Kosten aller Primterme gleichgroß sind. Wenn es mehrere Minimalpolynome geben sollte, dann wählen Sie bitte eines davon aus. (4 P.)

	Primterme	Minterme								
		M1	M2	M3	M4	M5	M6	M7	M8	M9
<input type="checkbox"/>	P1	X		X	X	X				
<input type="checkbox"/>	P2						X	X		
<input type="checkbox"/>	P3						X		X	
<input type="checkbox"/>	P4	X				X				X
<input type="checkbox"/>	P5			X		X				
<input type="checkbox"/>	P6							X	X	



Name:

Vorname:

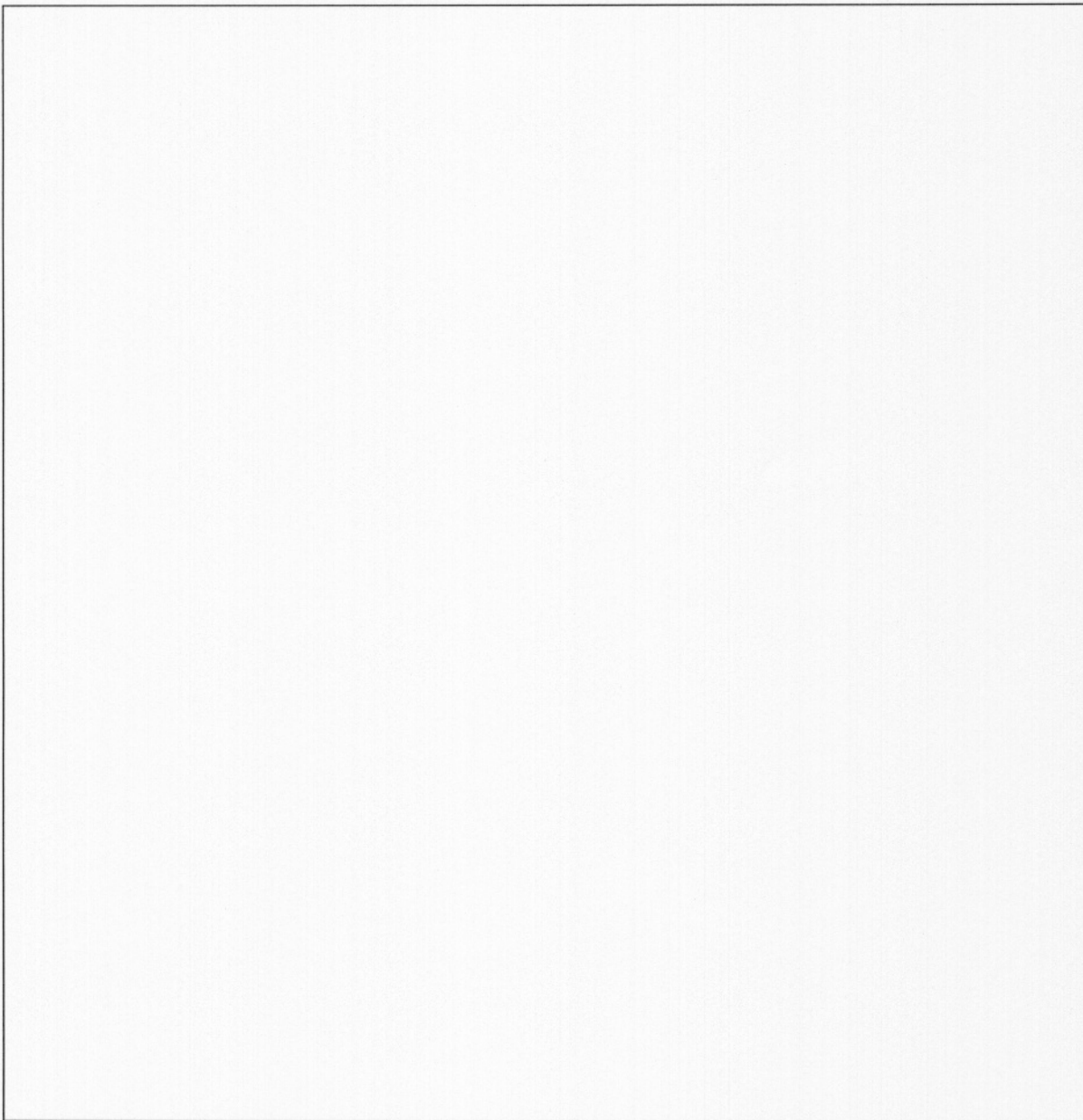
Matr.-Nr.:

6

**Aufgabe 2 (4 Punkte):**

Beweisen Sie folgende Aussage:

Die Binärdarstellung  $p$ , die das Produkt zweier  $n$ -stelliger Binärdarstellungen  $a$  und  $b$  darstellt, hat höchstens  $2n$  Stellen, wobei  $n \geq 2$  gelten soll.





Name:

Vorname:

Matr.-Nr.:

7

### Aufgabe 3 (5 Punkte):

Gegeben sei ein  $2^t$ -Wege  $n$ -Bit Multiplexer MUX1, wobei  $t \geq 2$ .

a) Kreuzen Sie an, welche der folgenden Aussagen über das Schaltelement MUX1 zutreffen. (3 P.)

Die Anzahl der Eingangssignale ist kleiner als die Anzahl der Ausgangssignale.

Die Anzahl der Eingangssignale beträgt  $2^t \cdot n + t$ , die Anzahl der Ausgangssignale  $n$ .

Die Anzahl der Eingangssignale beträgt  $2^t + n$ , die Anzahl der Ausgangssignale  $n$ .

Die Anzahl der Eingangssignale beträgt  $2^t \cdot n + t$ , die Anzahl der Ausgangssignale  $n + t$ .

Die Anzahl der Eingangssignale wird mit wachsendem  $t$  größer.

Die Anzahl der Ausgangssignale wird mit wachsendem  $t$  größer.

b) Gegeben seien weiterhin ein  $n$ -bit Multiplexer MUX2 und mehrere  $2^{t-1}$ -Wege  $n$ -bit Multiplexer MUX3.1, MUX3.2, ... .

Welche der folgenden Aussagen treffen zu? (2 P.)

Aus dem MUX2 und dem MUX3.1 kann ein MUX1 konstruiert werden.

Aus dem MUX2 und MUX3.1 und MUX3.2 kann ein MUX1 konstruiert werden.

Man braucht immer mehr als zwei der MUX3.\*, um einen MUX1 zu konstruieren.

Die Tiefe des MUX1 ist größer als die Tiefe des MUX2, wenn nur UND-Gatter mit 2 Eingängen zugelassen werden.



Name:

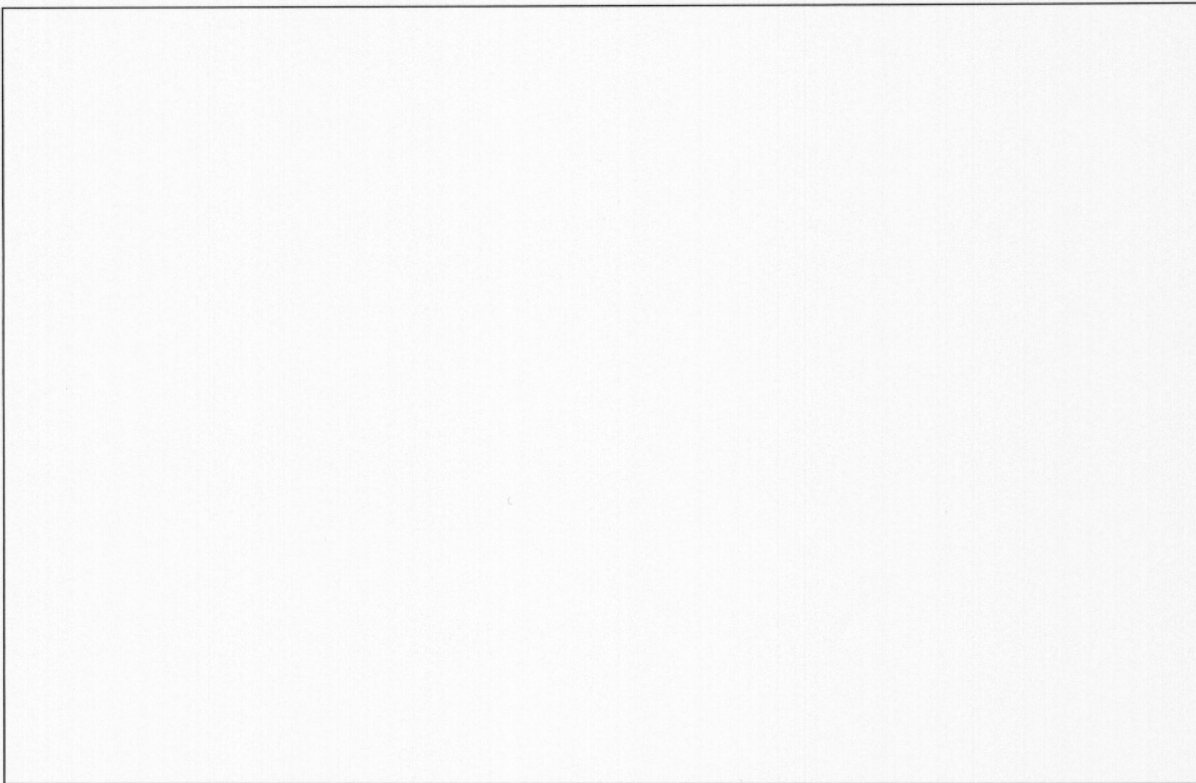
Vorname:

Matr.-Nr.:

8

### Aufgabe 4 (4 Punkte):

Konstruieren Sie ein SR-Latch aus zwei Invertern und zwei NAND-Gattern. (2 P.)



Stellen Sie die Funktionstabelle auf. (2 P.)

S	R	$Q(t)$	$Q(t+1)$	$\bar{Q}(t+1)$
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		







Name:

Vorname:

Matr.-Nr.:

10

b) Gegeben sei ein Automat mit den 8 Zuständen 0 bis 7, bei dem mehrere Zustandspaare die Gleichung (3.25) als erste Bedingung möglicher Äquivalenz erfüllen. Diese Ausgangs-Zustandspaare befinden sich in der linken Spalte der Tabelle, die mit **Stufe 0** gekennzeichnet ist, zusammen mit den zugehörigen Folge-Zustandspaaren in der rechten Spalte. Führen Sie eine Stufe der Zustandsminimierung durch, indem Sie in der Tabelle, die mit **Stufe 1** gekennzeichnet ist, Streichungen durchführen. Dabei soll die Tabelle von oben nach unten durchgegangen werden und bereits getätigte Streichungen berücksichtigt werden. (5 P.)

Stufe 0		Stufe 1	
(0,4)	(4,6) (5,8)	(0,4)	(4,6) (5,8)
(0,6)	(2,8)	(0,6)	(2,8)
(0,7)	(4,7) (3,8)	(0,7)	(4,7) (3,8)
(2,3)	(5,7) (0,4)	(2,3)	(5,7) (0,4)
(2,5)	(4,6)	(2,5)	(4,6)
(2,8)	(3,5) (0,4)	(2,8)	(3,5) (0,4)
(3,5)	(2,7) (0,6)	(3,5)	(2,7) (0,6)
(3,8)	(3,7)	(3,8)	(3,7)
(4,6)	(2,5)	(4,6)	(2,5)
(4,7)	(6,7) (3,5)	(4,7)	(6,7) (3,5)
(5,8)	(2,3) (0,6)	(5,8)	(2,3) (0,6)
(6,7)	(2,3) (4,7)	(6,7)	(2,3) (4,7)



**Aufgabe 6 (4 Punkte):**

Gegeben sei ein einfacher Computer wie in Kurseinheit 4 des Kurstextes beschrieben. Die Ausführung eines Call-Befehls `Call Zieladresse` läuft nach dem Holen dieses Befehls in mehreren Schritten ab. Der Stack soll dabei vom Ende des Speichers her wachsen, und bei einer Speichergröße von  $k$  soll der Stackpointer am Anfang den Wert  $k$  haben.

Welche der angegebenen Reihenfolgen ist richtig?

PC++  
AR  $\leftarrow$  Zieladresse  
M[SP]  $\leftarrow$  PC  
SP  $\leftarrow$  SP-1  
PC  $\leftarrow$  AR

PC++  
AR  $\leftarrow$  Zieladresse  
SP  $\leftarrow$  SP-1  
M[SP]  $\leftarrow$  PC  
PC  $\leftarrow$  AR

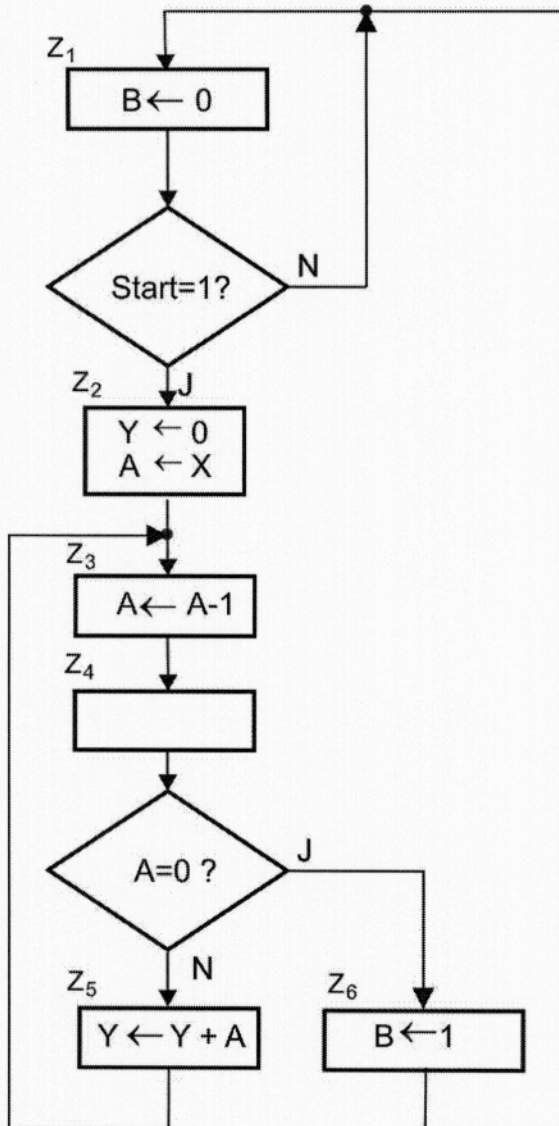
AR  $\leftarrow$  Zieladresse  
SP  $\leftarrow$  SP-1  
M[SP]  $\leftarrow$  PC  
PC  $\leftarrow$  AR  
PC++

PC++  
SP  $\leftarrow$  SP-1  
M[SP]  $\leftarrow$  PC  
PC  $\leftarrow$  AR  
AR  $\leftarrow$  Zieladresse



**Aufgabe 7 (11 Punkte):**

a) Gegeben sei das folgende ASM-Diagramm, wobei die Eingabe aus einer Binärzahl  $X$  und einem Steuersignal  $\text{Start}$  besteht, es Register für Binärzahlen  $A$ ,  $B$ , und  $Y$  gibt, und die Ausgabe aus den Werten der Register  $Y$  und  $B$  besteht.



Im weiteren soll stets gelten, dass das ASM-Diagramm im Zustand  $Z_1$  startet und  $\text{Start}=1$  gilt. Welche der folgenden Aussagen treffen zu? (6 P.)

Bei Eingabe  $X = 5$  hat die Variable  $A$  beim Erreichen des Zustands  $Z_6$  den Wert 0.

Bei Eingabe  $X = 5$  hat die Variable  $A$  beim Erreichen des Zustands  $Z_6$  den Wert 1.

Bei Eingabe  $X = 5$  hat die Variable  $Y$  beim Erreichen des Zustands  $Z_6$  den Wert 15.

Bei Eingabe  $X = 5$  hat die Variable  $Y$  beim Erreichen des Zustands  $Z_6$  den Wert 10.

Der Zustand  $Z_5$  wird bei Eingabe  $X = 1$  nie erreicht.

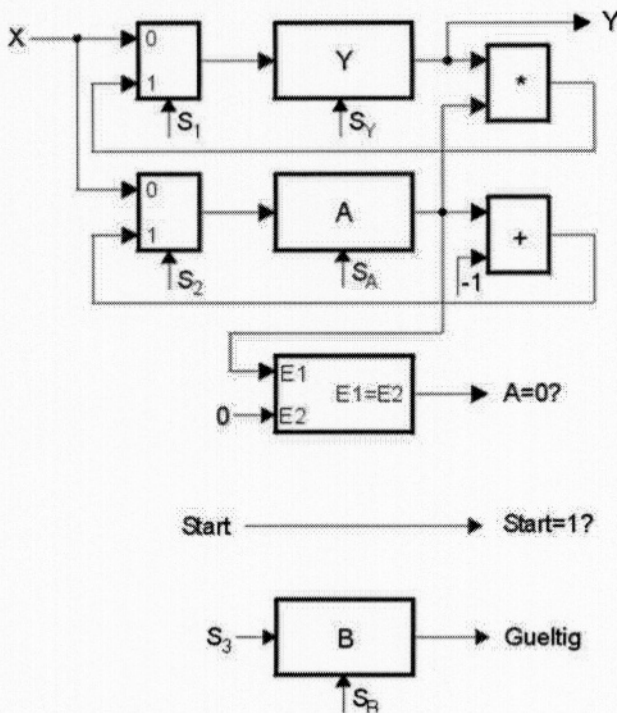
Der Zustand  $Z_5$  wird bei Eingabe  $X = 2$  nie erreicht.

Der Zustand  $Z_4$  kann nur eingespart werden, wenn die Entscheidungsbox darunter zu  $A = 1?$  geändert wird.

Der Zustand  $Z_4$  kann auch eingespart werden, wenn die Entscheidungsbox darunter nicht geändert wird.



b) Gegeben sei das folgende Operationswerk.



Welche der folgenden Aussagen treffen zu? (3 P.)

Das Operationswerk stellt das Operationswerk zu dem ASM-Diagramm aus a) dar.

Um Zustand Z2 des ASM-Diagramms zu implementieren, muss der obere Eingang des oberen Multiplexers mit 0 statt mit X verbunden werden.

Um Zustand Z3 des ASM-Diagramms zu implementieren, muss der Addierer durch einen Subtrahierer ersetzt werden.

Um Zustand Z4 des ASM-Diagramms zu implementieren, muss der Eingang E1 des Vergleichers mit dem Ausgang des Addierers verbunden werden.

Um Zustand Z5 des ASM-Diagramms zu implementieren, muss der Multiplizierer durch einen Addierer ersetzt werden.

Um die Zustände Z1 und Z6 des ASM-Diagramms zu implementieren, kann der Eingang des Registers B mit dem Ausgang eines Multiplexers verbunden werden, dessen Eingänge mit den konstanten Werten 0 und 1 belegt sind und dessen Steuereingang mit dem Signal  $S_3$  verbunden ist.



Name:

Vorname:

Matr.-Nr.:

14

c) Geben Sie die Anzahl der Steuersignale zwischen dem Steuerwerk und obigem Operationswerk an! (2 P.)



**Aufgabe 8 (8 Punkte)**

Entscheiden Sie, welche der folgenden Aussagen zum Thema CISC/RISC-Prinzipien korrekt sind.

- a)  RISC-Architekturen müssen zur Decodierung von Maschinenbefehlen stets ein mikroprogrammierbares Steuerwerk verwenden.
- b)  Wichtige Voraussetzung für eine effiziente Implementierung einer RISC-Architektur ist die Verfügbarkeit preiswerter Speichertechnologie.
- c)  CISC-Architekturen zeichnen sich durch umfangreiche Befehlssätze und mächtige Maschinenbefehle mit vielen Befehlsformaten, aber wenigen Adressierungsarten aus.
- d)  Bei CISC-Architekturen erhöhen Datenabhängigkeiten die Zahl der für die abhängigen Befehle benötigten Taktzyklen.
- e)  Bei RISC-Architekturen kann gegenüber CISC-Architekturen die Chip-Fläche auf dem Prozessor effizienter genutzt werden.
- f)  CISC-Architekturen werden wegen der Vielzahl an Adressierungsarten und der daraus resultierenden Möglichkeiten Operanden und Daten zu laden und zu speichern auch LOAD/STORE-Architekturen genannt.
- g)  Wegen des kleineren Befehlssatzes bei RISC-Architekturen wird dort grundsätzlich mehr Speicherplatz für die Unterbringung von Programmen benötigt.
- h)  Die Registerfenster-Technologie (window register organization) dient bei RISC-Architekturen einer effizienten Parameterübergabe zu/von Unterprogrammen.



## Aufgabe 9 (14 Punkte)

Ausgangspunkt des Befehls-Pipelining ist die Aufspaltung eines Befehls in Teilschritte. Das Befehls-Pipelining setzt voraus, dass für die zugrundeliegende Befehlsarchitektur eine Folge von Teilschritten gefunden wird, die für *alle* Befehle gleich ist. Für die folgende Aufgabe nehmen wir an, dass alle Befehle (inklusive der zugehörigen Adressierungsarten) in den folgenden fünf Teilschritten, welche Ihnen aus der DLX-Architektur bekannt sind, bearbeitet werden.

- 1. Instruction Fetch (IF):** Befehl holen (Opcode und Registeradressen können in einem Maschinenwort untergebracht werden).
- 2. Instruction Decode (ID):** Befehl dekodieren und gleichzeitig die Quelloperanden aus dem Registerblock lesen.
- 3. Execute (EX):** Führe eine arithmetische bzw. logische Operation mit den Operanden aus.
- 4. Memory Access (MEM):** Daten holen (*LOAD*) oder Speichern (*STORE*).
- 5. Write Back (WB):** Ergebnis in einem prozessorinternen Register speichern.

Der zeitliche Ablauf der Teilschritte bei konfliktfreiem Befehls-Pipelining sieht dabei wie folgt aus:

IF	ID	EX	MEM	WB		
	IF	ID	EX	MEM	WB	
		IF	ID	EX	MEM	WB

Bei dieser Definition der Pipeline gehen wir, ohne weitere Annahmen zu machen, davon aus, dass ein aus dem ALU-Ergebnisregister in das Universalregister oder Architekturregister zurückgeschriebener Wert erst am Ende der WB-Stufe stabil und damit weiterverwendbar anliegt.

Pipelinekonflikte können durch zusätzliche Hardware oder Software (Einfügen von NOP, *no-operation-Befehle*) behoben werden. In den folgenden Teilaufgaben soll nur die Software-Lösung betrachtet werden.

Wir untersuchen einen Pipelinekonflikt im folgenden Programmfragment:

ADD	R3,R1,R2	$R3 = R1 + R2$
SUB	R4,R1,R3	$R4 = R1 - R3$



Name:

Vorname:

Matr.-Nr.:

17

a) Welche Art von Pipelinekonflikt liegt hier genau vor? (2 P.)

Vervollständigen Sie die beiden nachfolgenden Sätze:

Bei dem vorliegenden Pipelinekonflikt handelt es sich um einen

Dieser wird durch eine

verursacht, die in der Verwendung von

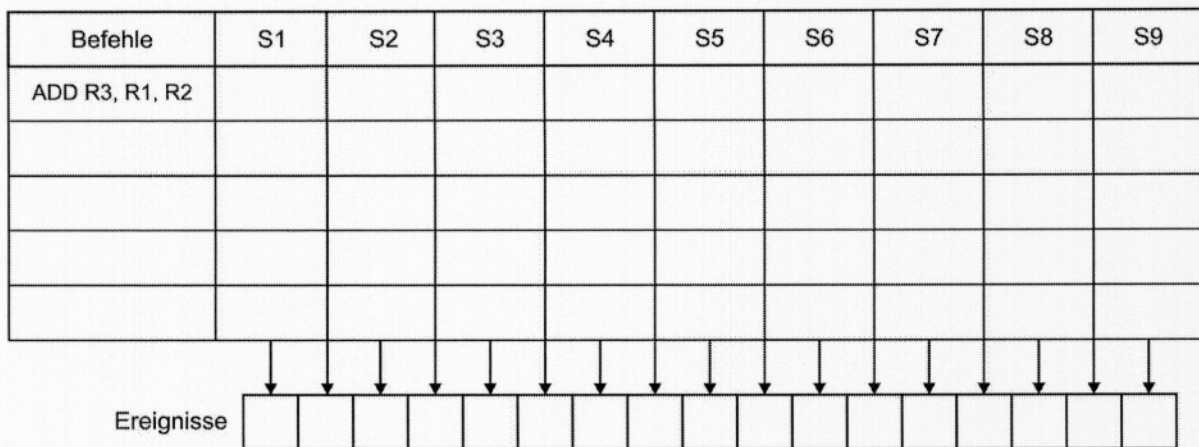
liegt.



b) Ausführung des Programmfragments auf der oben definierten Pipeline  
(6 P.)

b1) Skizzieren Sie den optimalen Ablauf des Programms in der Pipeline, indem Sie das folgende Pipeline-Ablaufzeitdiagramm unter Beachtung der Kausalität ausfüllen. Dazu soll in die einzelnen Felder der darin ausgeführte jeweilige Pipeline-Teilschritt **Si** eingetragen werden.

Verwenden Sie dazu statt der ausgeschriebenen Worte {**Instruction Fetch, Instruction Decode, Execute, Memory Access, Write Back**} die Kürzel {**IF, ID, EX, MEM, WB**}. Bei NOP-Befehlen sollen ungenutzte Pipeline-Phasen durch das Kürzel '-' markiert werden.



b2) Markieren Sie im Pipeline-Ablaufdiagramm die Ereignisse **E1**, **E2**, **E3** und **E4**, indem Sie das entsprechende Kürzel in die an den entsprechenden Stellen vorgesehenen Felder unterhalb des Diagrammes eintragen.

**E1** Wert von R3 durch den ADD-Befehl errechnet,

**E2** Wert von R3 liegt *vor* dem Registerblock an,

**E3** Wert von R3 wurde im Registerblock gespeichert und

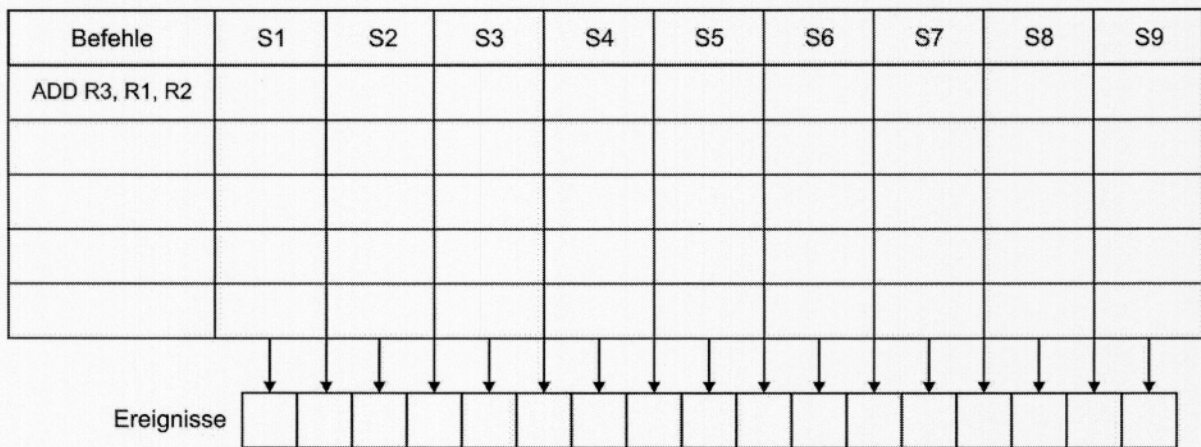
**E4** R3 liegt als Eingangsoperand für den SUB-Befehl an der ALU vor.



c) Ausführung des Programmfragments auf einer beschleunigten DLX-Pipeline (6 P.)

In diesem Aufgabenteil kann davon ausgegangen werden, dass ein aus dem ALU-Ergebnisregister in ein Architekturregister zurückgeschriebener Wert bereits nach der ersten Hälfte der WB-Stufe stabil und damit weiterverwendbar anliegt und dass die Operandenwerte erst in der zweiten Hälfte der ID-Stufe von den Architekturregistern in die ALU-Eingaberegister übertragen werden. In der WB-Stufe geschriebene Registerwerte können also bereits in demselben Takt wieder in der ID-Stufe gelesen werden.

c1) Skizzieren Sie den optimalen Ablauf des Programms in der Pipeline, indem Sie das folgende Pipeline-Ablaufzeitdiagramm unter Beachtung der Kausalität ausfüllen.

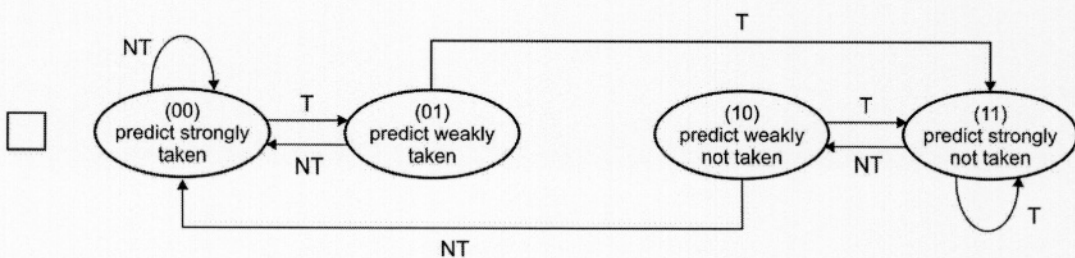
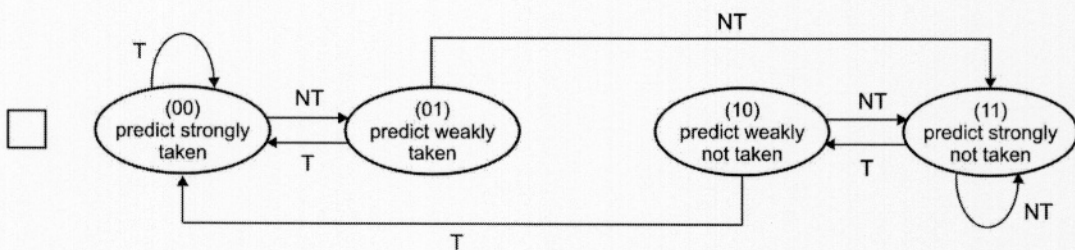
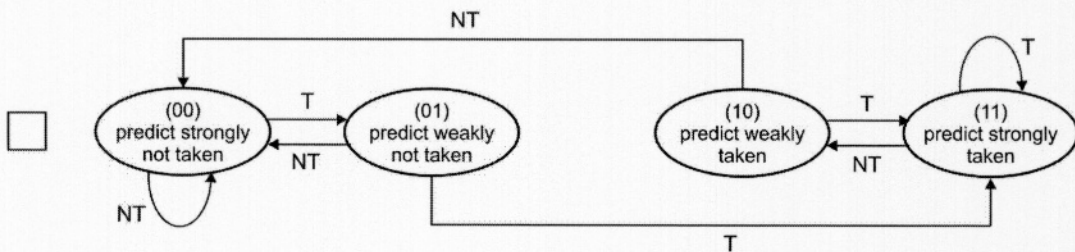
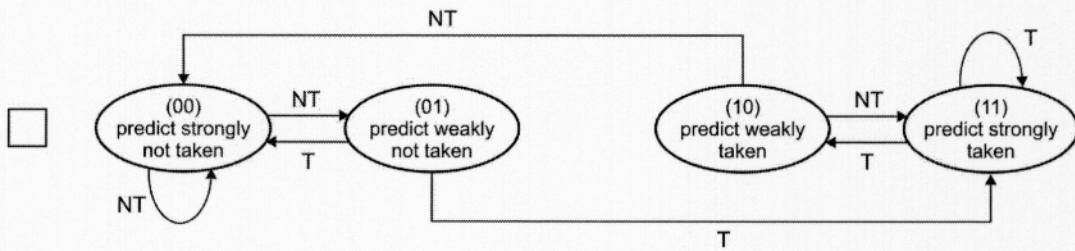
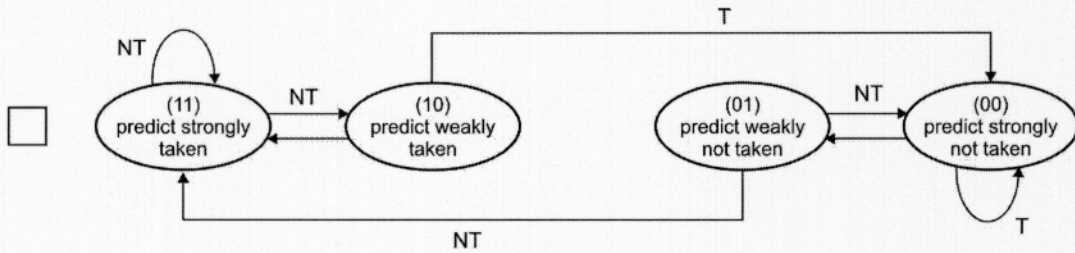


c2) Markieren Sie im Pipeline-Ablaufdiagramm die Ereignisse **E1**, **E2**, **E3** und **E4**.



### Aufgabe 10 (5 Punkte):

Markieren Sie die Zustandsgraphen, die einen Zwei-Bit-Prädiktor mit Hysteresisähler korrekt beschreiben!





### Aufgabe 11 (15 Punkte)

Es sollen drei kleine Cache-Speicher untersucht werden, die jeweils 8 Cache-Blöcke zu je 4 Worten (1 Wort = 4 Byte) enthalten und nach verschiedenen Techniken organisiert sind. Der erste Cache-Speicher verwendet eine direkt abgebildete Organisation, die mit DM bezeichnet wird. Der zweite Cache-Speicher arbeitet 2-fach satzassoziativ und wird mit dem Kürzel A2 referenziert. Der dritte Cache-Speicher mit der Bezeichnung AV ordnet die Cache-Blöcke vollassoziativ zu. Wenn Cache-Blöcke ersetzt werden müssen, verwendet man wenn nötig die LRU-Strategie.

- a) Welche der drei Organisationsform erfordert KEINE Ersetzungsstrategie? (1 P.)

--

- b) Vervollständigen Sie die nachfolgende Tabelle für die Aufteilung einer 32-Bit langen Adresse in Tag-Teil, Index-Teil und die Wortadresse! Geben Sie für jeden Teil die Anzahl der Bits an! (2 P.)

	Tag-Teil	Index-Teil	Wortadresse
DM			
A2			
AV			

- c) Der Einfachheit halber berücksichtigen wir im Folgenden nur die niederwertigen 8-Bit der Adresse ( $A_7 \dots A_0$ ) und nehmen an, dass die restlichen 24-Bit der Adresse ( $A_{31} \dots A_8$ ) einen konstanten Wert haben. Zu Beginn seien die Cache-Speicher leer und dann sollen Zugriffe auf folgende Adressen (in hexadezimaler Darstellung) erfolgen:

C6, 89, A7, D3, C2, C4, A3, D0, DD, C3

Ermitteln Sie das Verhalten der einzelnen Cache-Speicher, indem Sie die vorgegebenen Tabellen vervollständigen!











**Aufgabe 12 (8 Punkte):**

a) Leiten Sie das Gesetz von Amdahl für den Speedup  $S(n)$  eines  $n$ -Prozessorsystems her! Der sequentielle Anteil soll mit  $f$  bezeichnet werden. (3 P.)

b) Sie haben bei einem 8-Prozessorsystem einen Speedup von  $S(8) = 4$  gemessen. Welchen sequentiellen Anteil hat das Programm nach dem Amdahl'schen Gesetz? (2 P.)

c) Welcher Speedup ist bei gleichbleibendem sequentiellen Anteil mit einem 32-Prozessorsystem zu erwarten? Lohnt sich der Aufwand (mit Begründung)? Gegen Sie eine Obergrenze für den Speedup an! (3 P.)