

Klausur zur Vorlesung

"Grundlagen der technischen Informatik" und "Grundlagen der Rechnerarchitektur" Sommersemester 1999

1. Teil: GTI

Der erste Teil (GTI) der Klausur umfaßt 6 Aufgaben, die in maximal 1 Stunde zu bearbeiten sind. **Es sind keine Hilfsmittel zugelassen.** Schreiben Sie Ihre Lösungen nur auf die dafür vorgesehenen Blätter! Verwenden Sie **kein eigenes Konzeptpapier**; notfalls erhalten Sie welches bei der Aufsicht.

Lassen Sie die Aufgaben und ihre Lösungen zusammengeheftet!

Schreiben Sie auf jedes Blatt (auch auf das Konzeptpapier) in Blockschrift Ihren **Namen** und Ihre **Matrikelnummer**.

Streichen Sie bei Angabe mehrerer Lösungsansätze die nicht zu bewertenden Lösungen durch! Bei mehreren präsentierten Lösungen wird die Aufgabe nicht gewertet!

Abschreiben und abschreiben lassen führt zum Nichtbestehen der Klausur.

Es sind in diesem ersten Teil insgesamt 60 Punkte erreichbar.

Viel Erfolg!

Vor- und Nachname: _____

Matrikelnummer: _____ Fachbereich: _____

Studienfach (ankreuzen):

Informatik	<input type="checkbox"/>
Ingenieur-Informatik (Elektrotechnik)	<input type="checkbox"/>
Ingenieur-Informatik (Maschinenbau)	<input type="checkbox"/>
Ingenieur-Informatik (Informatik)	<input type="checkbox"/>
Wirtschafts-Informatik	<input type="checkbox"/>
Informatik als Nebenfach	<input type="checkbox"/>
Lehramt Informatik	<input type="checkbox"/>

Aufgabe	1	2	3	4	5	6	Summe
Punkte	6	12	12	12	8	10	60
Erreicht							

Summe	GTI:		GRA:		Gesamt:		Note:	
-------	------	--	------	--	---------	--	-------	--

Name:

Matrikel-Nummer:

Aufgabe 1 (6 min, 6 Punkte): Zahldarstellung/ Komplementrechnung

Führen Sie die folgenden Rechnungen im 2er Komplement mit 8-stelligen Dualzahlen (inklusive Vorzeichen-Bit) durch:

$$75 + 60 \quad -75 + 60 \quad -75 - 8$$

Geben Sie an, ob das Ergebnis richtig (gültig) oder falsch (Bereichsüberschreitung) ist.

Achtung: Kennzeichnen Sie Ihre Lösung eindeutig durch evtl. Durchstreichen der ersten Lösung.

Lösung:

2er	V							
75		0	1	0	0	1	0	1
+60		0	0	1	1	1	1	0
Übertrag								
		1	0	0	0	0	1	1
ankreuzen		richtig:				falsch:		<input checked="" type="checkbox"/>

Ungültiges Vorzeichen im Ergebnis!

2er	V							
-75		1	0	1	1	0	1	0
+60		0	0	1	1	1	1	0
Übertrag								
		1	1	1	1	0	0	0
ankreuzen		richtig:		<input checked="" type="checkbox"/>		falsch:		

2er	V							
-75		1	0	1	1	0	1	0
-8		1	1	1	1	1	0	0
Übertrag								
		1	1	0	1	0	1	1
ankreuzen		richtig:		<input checked="" type="checkbox"/>		falsch:		

Ersatztabellen:

2er	V							
75								
+60								
Übertrag								
ankreuzen		richtig:				falsch:		

2er	V							
-75								
+60								
Übertrag								
ankreuzen		richtig:				falsch:		

2er	V							
-75								
-8								
Übertrag								
ankreuzen		richtig:				falsch:		

Name:

Matrikel-Nummer:

Aufgabe 2 (12 min, 2 + 5 + 5 Punkte): Vereinfachung Bool'scher Funktionen

Gegeben sei eine Bool'sche Funktion

$$F = a \bar{b} \bar{c} d + a b \bar{c} d + a \bar{b} \bar{c} \bar{d} + \bar{a} (\bar{b} c d + b c d) + \bar{a} c \bar{d} + \bar{a} \bar{b} \bar{c} d + \bar{a} \bar{b} \bar{c} \bar{d}$$

(a) Geben Sie F in der Minterm-Normalform an.

$$F_{\text{Minterm}} = a \bar{b} \bar{c} d + a b \bar{c} d + a \bar{b} \bar{c} \bar{d} + \bar{a} \bar{b} c d + \bar{a} b c d + \bar{a} b c \bar{d} + \bar{a} \bar{b} c \bar{d} + \bar{a} \bar{b} \bar{c} d + \bar{a} \bar{b} \bar{c} \bar{d}$$

(b) Bestimmen Sie eine minimale Darstellung von F mit Hilfe des Karnaugh- Diagramms: Füllen Sie dazu das auf der nächsten Seite angegebene Diagramm zunächst entsprechend mit Einsen aus. Markieren Sie dann die gefundenen Primimplikanten.

Geben Sie die minimale Darstellung hier an:

$$F_{\text{Min}} = \bar{a} c + \bar{b} \bar{c} + a \bar{c} d$$

(c) Minimieren Sie die Funktion

$$G = a \bar{b} \bar{c} + \bar{a} \bar{b} c + \bar{a} b c + a b c + a \bar{b} c$$

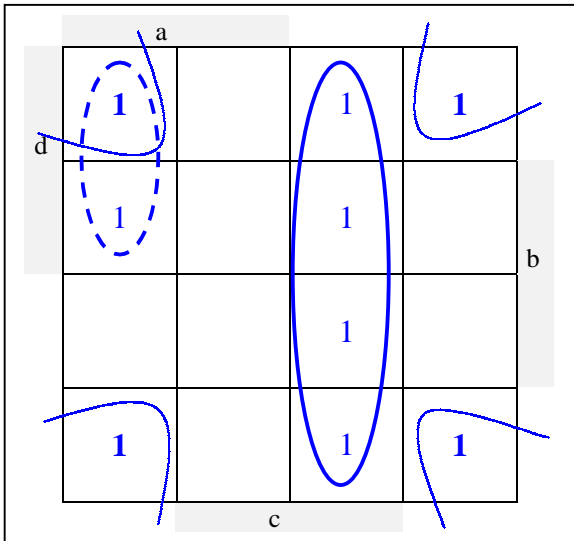
mit Hilfe des Quine/McClusky Verfahrens. Das Verfahren benötigt hier 2 Iterationen. Geben Sie die Zwischenergebnisse nach den angegebenen Aktionen des Verfahrens in beiden Iterationen an. Bei der Summe der Konsensusse sind **alle** (d.h. **auch doppelt auftretende**) anzugeben!

Name:

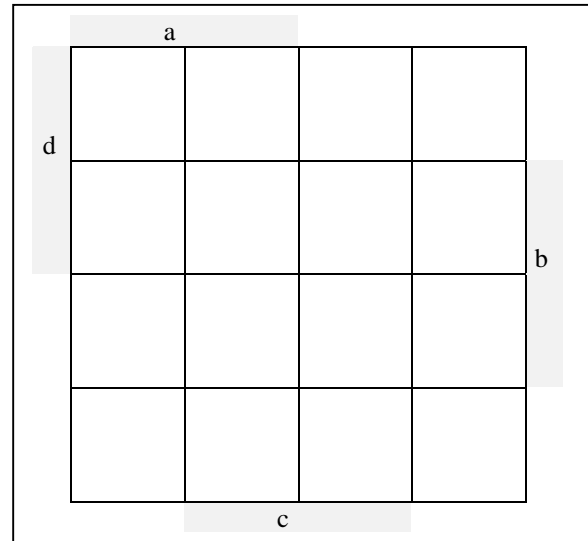
Matrikel-Nummer:

Zu Aufgabe 2 (b):

Auszufüllendes Karnaugh-Diagramm:



Ersatzdiagramm:



Teil (c): Quine / McClusky Verfahren

Nr.	Aktion	Zwischenergebnisse
	Ausgangsfunktion G	$a \bar{b} \bar{c} + \bar{a} \bar{b} c + \bar{a} b c + a b c + a \bar{b} c$
1	Summe aller „Simplen Konsensusse“	$a \bar{b} + \bar{a} c + \bar{b} c + b c + a c$
	Nach „Streichen von Verlängerungen“ und „Vereinfachen“	$a \bar{b} + \bar{a} c + \bar{b} c + b c + a c$
2	Summe aller „Simplen Konsensusse“	$c + c$
	Nach „Streichen von Verlängerungen“ und „Vereinfachen“	$a \bar{b} + c$

Ersatztable:

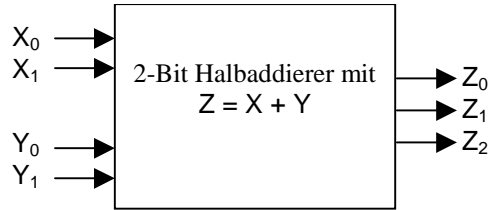
Nr.	Aktion	Zwischenergebnisse
	Ausgangsfunktion G	$a \bar{b} \bar{c} + \bar{a} \bar{b} c + \bar{a} b c + a b c + a \bar{b} c$
1	Summe aller „Simplen Konsensusse“	
	Nach „Streichen von Verlängerungen“ und „Vereinfachen“	
2	Summe aller „Simplen Konsensusse“	
	Nach „Streichen von Verlängerungen“ und „Vereinfachen“	

Name:

Matrikel-Nummer:

Aufgabe 3 (12 min, 12 Punkte): PLA

- (a) Geben Sie die durch einen 2-Bit-Halbaddierer realisierte Bool'sche Funktion $f(x,y) = z$ mit $z = x + y$ in Tabellenform an (X_0, Y_0, Z_0 ist jeweils das niederwertigste Bit; Z_2 entspricht dem Übertrag (Carry)):



Hinweis: Ein 2-Bit Halbaddierer addiert zwei positive 2-Bit Zahlen und liefert als Ergebnis eine positive 3-Bit Zahl (d.h. eine 2-Bit Zahl und einen Übertrag).

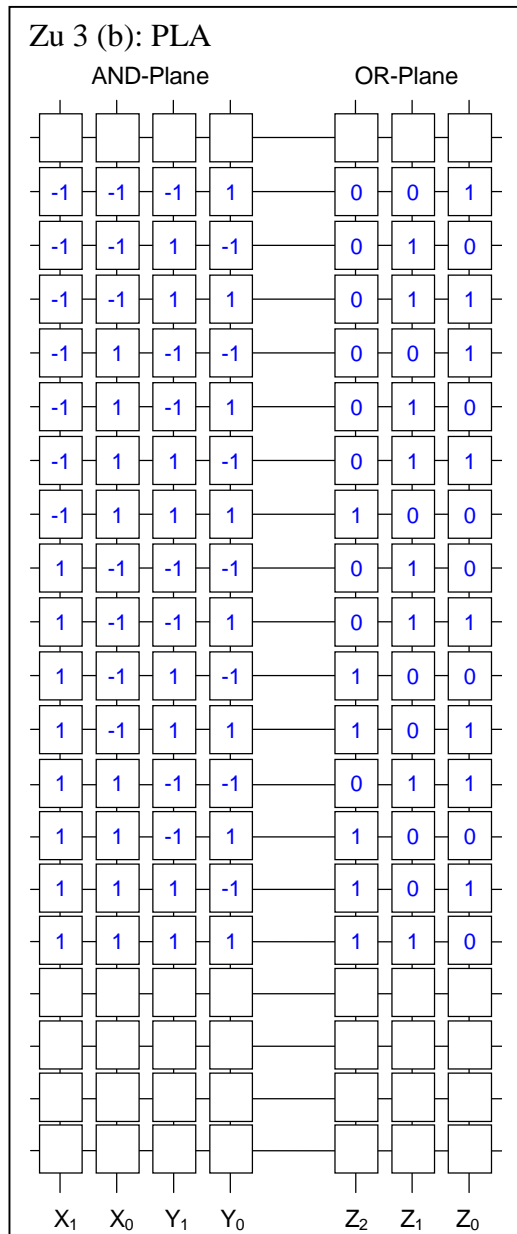
- (b) Entwickeln Sie einen PLA, der die Funktion aus Teil (a) berechnet. Minimieren Sie die Anzahl der benötigten Zeilen des PLA **ohne** die Funktion **zu vereinfachen**. Das Zusammenfassen von Teilfunktionen in einer Zeile ist dagegen erlaubt! Tragen Sie -1, 1 oder 0 entsprechend in die Kästchen ein. Kästchen nicht benötigter Zeilen lassen Sie frei.

Wieviel Zeilen werden minimal benötigt:

15

Zu 3 (a): Wahrheitstabelle

X_1	X_0	Y_1	Y_0		Z_2	Z_1	Z_0
0	0	0	0		0	0	0
0	0	0	1		0	0	1
0	0	1	0		0	1	0
0	0	1	1		0	1	1
0	1	0	0		0	0	1
0	1	0	1		0	1	0
0	1	1	0		0	1	1
0	1	1	1		1	0	0
1	0	0	0		0	1	0
1	0	0	1		0	1	1
1	0	1	0		1	0	0
1	0	1	1		1	0	1
1	1	0	0		0	1	1
1	1	0	1		1	0	0
1	1	1	0		1	0	1
1	1	1	1		1	1	0



Name:

Matrikel-Nummer:

Ersatzseite zu Aufgabe 3.

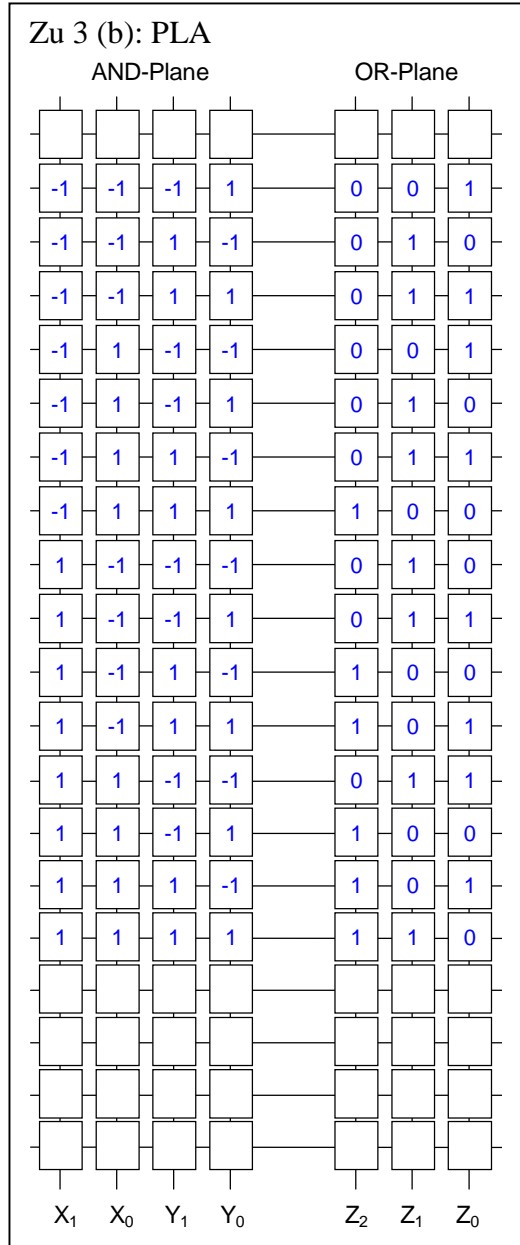
Achtung: Kennzeichnen Sie Ihre Lösung eindeutig durch evtl. Durchstreichen der ersten Lösung.

Wieviel Zeilen werden minimal benötigt:

15

Zu 3 (a): Wahrheitstabelle

X_1	X_0	Y_1	Y_0		Z_2	Z_1	Z_0
0	0	0	0		0	0	0
0	0	0	1		0	0	1
0	0	1	0		0	1	0
0	0	1	1		0	1	1
0	1	0	0		0	0	1
0	1	0	1		0	1	0
0	1	1	0		0	1	1
0	1	1	1		1	0	0
1	0	0	0		0	1	0
1	0	0	1		0	1	1
1	0	1	0		1	0	0
1	0	1	1		1	0	1
1	1	0	0		0	1	1
1	1	0	1		1	0	0
1	1	1	0		1	0	1
1	1	1	1		1	1	0



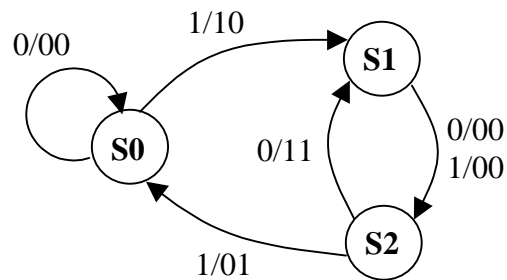
Name:

Matrikel-Nummer:

Aufgabe 4 (12 min, 12 Punkte): VHDL-Aufgabe

Gegeben sei der folgende Automat (als Zustandsübergangsgraph). Vervollständigen Sie die nachstehende VHDL-Beschreibung, indem Sie in die Kästchen die **Zahlen** der korrekten Textblöcke aus dem Baukasten einsetzen.

Hinweis: Bauelemente dürfen mehrfach verwendet werden. Nicht alle Bauelemente werden benötigt.



S1, S2, S3 sind Zustände;
1/01 steht für $X = 1, Y(1) = 0, Y(0) = 1$.

Lösungsrahmen: siehe nächste Seite

Lösungsrahmen zu Aufgabe 4:

ACHTUNG: "Zahlen" und nicht "Wörter" eintragen!

```

  4      19 is
  5 ( CLK : In    std_logic;
      RES : In    std_logic;
      X  : In    std_logic;
      Y  : Out   std_logic_vector (1 downto 0) );
end AUTOMAT;

```

```

  2 BEHAVIORAL of 19 is

```

```

type fsm_state is (S0, S1, S2);
signal CURRENT_STATE, NEXT_STATE: fsm_state;

```

```

  8
SEQ: process (CURRENT_STATE, X)
begin

```

```

  case CURRENT_STATE is
  when S0 =>
    if 12 then
      Y 16 "00";
      NEXT_STATE 23 ;

```

```

    else
      Y 16 "10";
      NEXT_STATE 24 ;

```

```

    end if;
  when S1 =>
    Y 16 "00";
    NEXT_STATE 25 ;

```

```

  when S2 =>
    if 12 then
      Y 16 "11";
      NEXT_STATE 24 ;

```

```

    else
      Y 16 "01";
      NEXT_STATE 23 ;

```

```

    end if;
  end 6 ;
end 9 ;

```

```

MEM: process(CLK, RES)
begin
  if (RES = '0') then
    CURRENT_STATE 16 S0;
  else
    if (CLK'event and CLK = '1') then
      CURRENT_STATE 16 NEXT_STATE;
    end if;
  end if;
end 9 ;

```

```

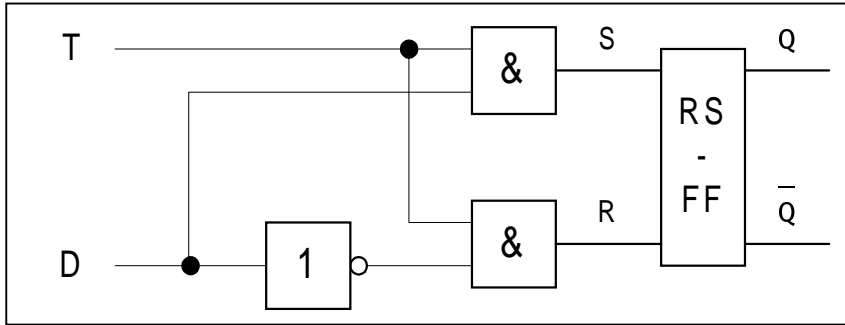
end 20 ;

```

Baukasten:

- 1: interface
- 2: architecture
- 3: structure
- 4: entity
- 5: port
- 6: case
- 7: start
- 8: begin
- 9: process
- 10: description
- 11: sequence
- 12: X = '0'
- 13: X == '0'
- 14: X = LOW
- 15: X = '1'
- 16: <=
- 17: :=
- 18: VHDL
- 19: AUTOMAT
- 20: BEHAVIORAL
- 21: SEQ
- 22: MEM
- 23: <= S0
- 24: <= S1
- 25: <= S2
- 26: = S0
- 27: = S1
- 28: = S2

Aufgabe 5 (8 min, 2 + 6 Punkte): CMOS-Technologie



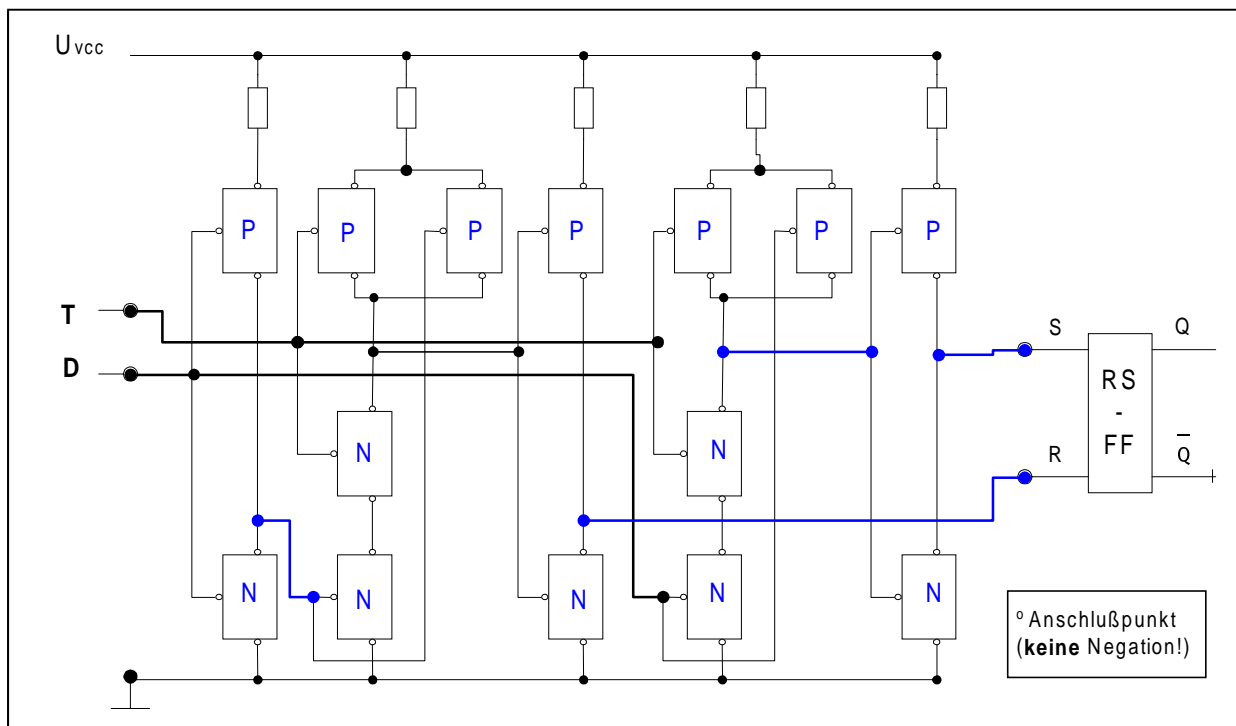
Das obere Bild beinhaltet ein RS-Flip-Flop, das durch die vorstehenden Gatter zu einem D-Flip-Flop erweitert wurde.

- (a) Geben Sie für die gegebene Eingabefolge von T und D die Folge der Ausgabewerte für Q an.

Eingabe T/D	1/0	0/0	1/1	0/0	1/0	1/1	1/0	0/1
Ausgabe Q	0	0	1	1	0	1	0	0

- (b) Vervollständigen Sie die Implementierung im unteren Bild mit Hilfe der in der Tabelle angegebenen Bauelemente zu einer reinen CMOS Schaltung. Tragen Sie dazu in die freien Kästen **P**, **N**, bzw. **L** für das jeweils einzusetzende Schaltzeichen ein und fügen Sie **3 fehlende Verbindungen** ein, so daß sich genau dieselbe Funktion wie im oberen Bild ergibt.

P	PMOS Transistor (Anreicherungstyp, ist durchgeschaltet falls Gate Low)	N	NMOS Transistor (Anreicherungstyp, ist durchgeschaltet falls Gate High)	L	Überbrückung



Name:

Matrikel-Nummer:

Aufgabe 6 (10 min, 5 + 5 Punkte): Multiplikation

Multiplizieren Sie die folgenden beiden 4-Bit Zahlen X und Y.

$$X = 1011_2 = -5_{10} \qquad Y = 1110_2 = -2_{10}$$

Wichtig: X und Y sind im **2er-Komplement** dargestellt !!

- (a) Verwenden sie den aus der Vorlesung bekannten „**Vorzeichen-gerechten shift & add**“ Algorithmus zur Berechnung des Produktes $AC \ \& \ Q := Q * M$ mit $M := X$ und $Q := Y$. Geben Sie in jeder Zeile den aktuellen Wert von AC und Q *nach* Ausführung der in der Zeile *zuvor* anzugebenden Operation an.

V	AC	Q	Nächste Operation
0	0000	1110	1. Shift
0	0000	0111	Add M = 11011
1	1011	0111	2. Shift
1	1101	1011	Add M = 11011
1	1000	1011	3. Shift
1	1100	0101	Add M = 11011
1	0111	0101	4. Shift
1	1011	1010	Korrektur: Sub M = Add 00101
0	0000	1010	Ergebnis: +10

- (b) Führen Sie dieselbe Multiplikation mit dem **Booth-Verfahren** durch. Dabei sei $M := Y = 1110_2$ (in Q wird dem Verfahren entsprechend $X = 1011_2$ abgelegt). Geben Sie wie in (a) die Inhalte der Register A und Q nach jeder Operation in Binärdarstellung an. Markieren Sie am Ende deutlich die Bits in A und Q, die das Ergebnis angeben!

A	Q	Nächste Operation
0000	1011 0	Sub M = Add 0010 ₂
0010	1011 0	Shift
0001	0101 1	Shift
0000	1010 1	Add M = 1110 ₂
1110	1010 1	Shift
1111	0101 0	Sub M = Add 0010 ₂
0001	0101 0	Fertig, Ergebnis = 0001010 ₂ = 10

Name:

Matrikel-Nummer:

Ersatzseite zu Aufgabe 6.

Achtung: Kennzeichnen Sie Ihre Lösung eindeutig durch evtl. Durchstreichen der ersten Lösung.

(a)

V	AC	Q	Nächste Operation
0	0000	1110	

(b)

A	Q	Nächste Operation
0000		