

Prof. Dr.-Ing. Jürgen Teich
Lehrstuhl für Informatik 12
(Hardware-Software-Co-Design)
Friedrich-Alexander-Universität Erlangen-Nürnberg

Klausur Grundlagen der Technischen Informatik

28. September 2018

Vorname	
Nachname	
Matrikelnummer	

Aufgabe	1	2	3	4	5	Σ
Max. Punkte	16	16	16	16	16	80
Erreichte Punkte						
Note						

Organisatorische Hinweise

Bitte sorgfältig lesen und die Kenntnisnahme durch Unterschrift bestätigen

- a) Bitte legen Sie Ihren Studentenausweis bereit.
 - b) Als Hilfsmittel sind nur Schreibmaterialien und ein beidseitig handbeschriebenes DIN A4-Blatt zugelassen.
 - c) Verwenden Sie weder Rot- noch Bleistifte.
 - d) Konzeptpapier wird nicht bewertet.
 - e) Sie können bei der Aufsicht zusätzliche Bearbeitungsblätter anfordern.
 - f) Unleserliches wird nicht bewertet.
 - g) Bei mehreren präsentierten Lösungen wird die Aufgabe nicht gewertet. Streichen Sie daher bei Angabe mehrerer Lösungsansätze die nicht zu bewertenden Lösungen durch.
 - h) Die Bearbeitungszeit beträgt 120 Minuten.
-

Erklärung

- a) Im Falle einer während der Prüfung auftretenden Prüfungsunfähigkeit zeige ich dies sofort der Aufsicht an und befolge deren Anweisungen. Ich weiß, dass ich die volle Beweislast trage. Ich lasse mir das Formular des Prüfungsamts, das für diese Fälle vorgesehen ist, aushändigen und verfare nach den dort niedergelegten Richtlinien.
- b) Ich weiß, dass im Falle des Täuschungsversuchs oder der Benutzung unerlaubter Hilfsmittel („Unterschleif“) der Prüfungsausschuss die Entscheidung treffen kann, die betroffene Prüfungsleistung als mit „nicht ausreichend“ bewertet gelten zu lassen.
- c) Ich habe die obigen Hinweise zur Kenntnis genommen.

Erlangen, den

Unterschrift

Kopiervorlage: nur für Fachschaften

Aufgabe 1 (Zahlensysteme)

(16 Punkte)

a) Wie heißen die Zahlensysteme zur Basis 8 und 16?

(1 Punkt)

b) Nachrichten in "Latin-1"- bzw. "Base64"-Codierung können als Zahlen codiert in zwei unterschiedlichen polyadischen Zahlensystemen mit 256 bzw. 64 Ziffern aufgefasst werden. Die Werte der verwendeten Ziffern für die beiden unterschiedlichen polyadischen Zahlensysteme sind in den Tabellen 1 und 2 aufgeführt. Wandeln Sie die Drei-Zeichen-Nachricht "FAU" in "Latin-1"-Codierung, welche in Tabelle 1 genauer spezifiziert ist, in die "Base64"-Codierung um.

(4 Punkte)

Tabelle 1: ISO/IEC 8859-1 AKA "Latin-1"-Codierung

Wert	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	Unbelegt															
1_	Unbelegt															
2_	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
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_	Unbelegt															
9_	Unbelegt															
A_	NBSP	ı	ç	£	¤	¥	ı	§	¨	©	ª	«	¬	SHY	®	ˆ
B_	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C_	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D_	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E_	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F_	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Tabelle 2: Ziffernwerte der "Base64"-Codierung

Wert	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1_	Q	R	S	T	U	V	W	X	Y	Z	a	b	c	d	e	f
2_	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
3_	w	x	y	z	0	1	2	3	4	5	6	7	8	9	+	/

- c) Was sind die minimal und maximal möglichen Ergebnisse bei der Multiplikation $Z_n \cdot Z_m$ einer vorzeichenlosen n -stelligen Zahl zur Basis 8 mit einer vorzeichenlosen m -stelligen Zahl zur Basis 16? (2 Punkte)
- d) Wie viele 2-zu-1-Multiplexer benötigen Sie, um jede beliebige Funktion f mit n Booleschen Eingangsvariablen als ein Schaltnetz bestehend nur aus 2-zu-1-Multiplexern sowie den Konstanten **wahr** und **falsch** aufzubauen? Begründen Sie Ihre Antwort. (2 Punkte)
- e) In dieser Aufgabe soll mit 16-Bit Gleitkommazahlen gearbeitet werden. Diese werden analog zum IEEE-Format gebildet. Das Format der Gleitkommazahl sieht dabei wie folgend aus:
Vorzeichen (1 Bit), Exponent (5 Bit), Mantisse (10 Bit)

V	E	M
15	14	10 9
		0

Führen Sie nun die Multiplikation der beiden in diesem Format dargestellten Gleitkommazahlen 1 10001 1110000000 und 0 01100 0010010010 aus, und geben Sie das Ergebnis im gleichen Format an. (7 Punkte)

Aufgabe 2 (Minimierung von Schaltfunktionen)

(16 Punkte)

a) Sei eine Schaltfunktion $f_1(e, d, c, b, a)$ mit Primimplikanten PI und Einsstellenmenge E gegeben:

$$PI = \{\bar{e}\bar{c}\bar{a}, \bar{e}d\bar{a}, dc\bar{a}, b\bar{a}, ec\}$$

$$E = \{\bar{e}d\bar{c}\bar{b}\bar{a}, \bar{e}d\bar{c}b\bar{a}, \bar{e}dcb\bar{a}, edcb\bar{a}\}$$

1. Tragen Sie in folgende Tabelle ein, welcher Primimplikant welche Einsstelle überdeckt.

(1 Punkt)

k	PI	j				p_i	c_i
		8	10	12	30		
0	$\bar{e}\bar{c}\bar{a}$					A	3
1	$\bar{e}d\bar{a}$					B	3
2	$dc\bar{a}$					C	3
3	$b\bar{a}$					D	2
4	ec					E	2

2. Stellen Sie den Petrick-Ausdruck für f_1 auf.

(1 Punkt)

3. Bestimmen Sie mit dem Petrick-Verfahren alle kostenminimalen DMFs von f_1 . (3 Punkte)

Kopiervorlage: nur für Fachschaften

b) Bestimmen Sie alle Primimplikanten der Schaltfunktion

$$f_2(e, d, c, b, a) = \bar{e}d\bar{c}\bar{b}\bar{a} + \bar{e}d\bar{c}b\bar{a} + edc\bar{b}\bar{a} + edc\bar{b}a + \bar{e}d\bar{c}b\bar{a} + \bar{e}d\bar{c}ba + \bar{e}d\bar{c}\bar{b}\bar{a} + \bar{e}d\bar{c}\bar{b}a + \bar{e}d\bar{c}b\bar{a} + \bar{e}d\bar{c}ba + \bar{e}d\bar{c}\bar{b}\bar{a} + \bar{e}d\bar{c}\bar{b}a$$

durch Verwendung des *Quine/McCluskey*-Verfahrens.

1. Stellen Sie die Klassen $Q_{5,0}$ bis $Q_{5,5}$ auf.

(1 Punkt)

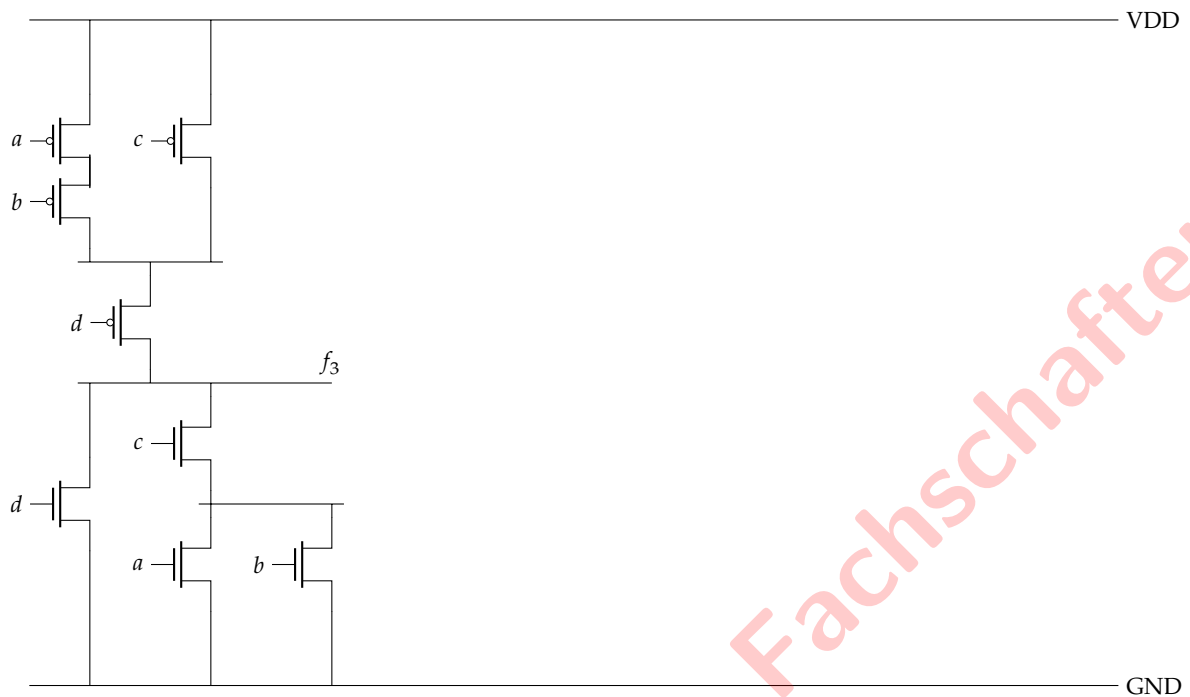
2. Bestimmen Sie nun basierend auf den Klassen $Q_{5,0}$ bis $Q_{5,5}$ alle Primimplikanten mittels des *Quine/McCluskey*-Verfahrens. Markieren Sie die im Verlauf des Verfahrens genutzten Terme durch Unterstreichen.

(4 Punkte)

3. Bestimmen Sie eine DMF von f_2 .

(1 Punkt)

c) Sei die CMOS-Schaltung einer Schaltfunktion $f_3(d, c, b, a)$ gegeben:



1. Geben Sie einen schaltalgebraischen Ausdruck für f_3 an. (1 Punkt)

2. Sei eine Schaltfunktion

$$f_4(d, c, b, a) = \overline{f_3(d, c, b, a)} \cdot (\bar{b} + c) + f_3(d, c, b, a) \cdot \bar{a}$$

gegeben. Geben Sie das Pull-Down-Netz für f_4 an. (1 Punkt)
 Hinweis: Verwenden Sie f_3 als Synonym für $f_3(d, c, b, a)$.

3. Ergänzen Sie die obige Schaltung durch eine CMOS-Schaltung von f_4 . (3 Punkte)

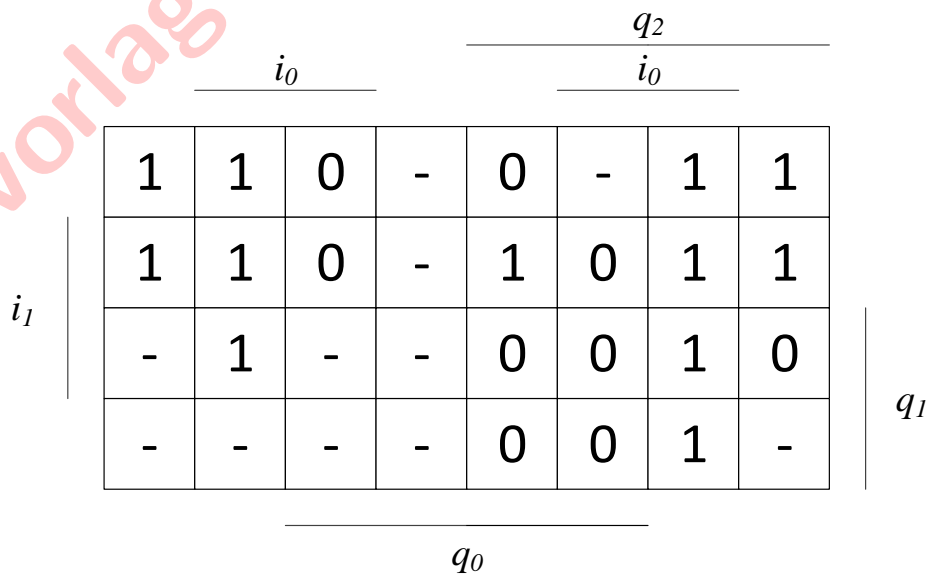
Kopiervorlage: nur für Fachschaften

1. Vervollständigen Sie die nachfolgend gegebene Automatentafel unter Verwendung von taktflankengesteuerten D-, JK- bzw. T-Flipflops. (6 Punkte)

Zustandsname	Aktueller Zustand			Eingabe		Nachfolgezustand			Ansteuerfunktion				Ausgabe		
	q_2	q_1	q_0	i_1	i_0	q'_2	q'_1	q'_0	D_2	J_1	K_1	T_0	o_2	o_1	o_0
idle	0	0	0	0	0										
idle	0	0	0	0	1										
idle	0	0	0	1	0										
idle	0	0	0	1	1										
getCrate	0	0	1	0	0										
getCrate	0	0	1	0	1										
getCrate	0	0	1	1	0										
getCrate	0	0	1	1	1										
getBottle	0	1	0	0	0										
getBottle	0	1	0	0	1										
getBottle	0	1	0	1	0										
getBottle	0	1	0	1	1										
print	0	1	1	0	0										
print	0	1	1	0	1										
print	0	1	1	1	0										
print	0	1	1	1	1										
maint	1	0	0	0	0										
maint	1	0	0	0	1										
maint	1	0	0	1	0										
maint	1	0	0	1	1										

2. Nehmen Sie im Folgenden an, dass die Ansteuerfunktion J_1 des JK-Flipflops im unten gegebenen Symmetriediagramm spezifiziert sei. Entwickeln Sie eine KMF der Ansteuerfunktion J_1 , und geben Sie den resultierenden schaltalgebraischen Ausdruck an.

Hinweis: Die hier angegebene Ansteuerfunktion J_1 entspricht nicht der Lösung von Teilaufgabe 1. (3 Punkte)



3. Zeichnen Sie ein Schaltwerk für den in Teilaufgabe 1 realisierten Automaten. Verwenden Sie dazu die in Teilaufgabe 2 bestimmte KMF der Ansteuerfunktion J_1 , und gehen Sie davon aus, dass Sie die benötigten Ansteuerfunktionen D_2 , K_1 und T_0 als Eingangssignale zur Verfügung haben. Auf die Realisierung der Ausgabe des Automaten kann verzichtet werden. (3 Punkte)

- b) Geben Sie den allgemeinen Aufbau eines Mealy-Automaten als Blockschaltbild an. (2 Punkte)

- c) Zeichnen Sie die vollständige Schaltung eines Active-HIGH RS-Latches. (2 Punkte)

Aufgabe 4 (Codierung und Rechnerarithmetik)

(16 Punkte)

- a) Gegeben ist ein quantisiertes 8-Bit Grauwertbild. Jedes Pixel des ursprünglichen Bildes hat einen Grauwert aus dem Intervall $[0, 255]$. Durch die Quantisierung wurden diese Grauwerte in 8 gleich große Bereiche aufgeteilt. Dadurch sind nur noch 3 Bit für die Speicherung der Grauwerte notwendig. Die Anzahl der Pixel für entsprechende Grauwerte des 10000 Pixel großen Grauwertbildes ist im Folgenden für jeden Bereich dargestellt.

Bereiche	G1	G2	G3	G4	G5	G6	G7	G8	Σ
Anzahl Pixel	200	300	400	1000	1100	1300	2700	3000	10000

1. Was ist der Informationsgehalt der Aussage "Der Grauwert ist aus dem Grauwertbereich G3 oder G4 oder G5"? (2 Punkte)
2. Nennen Sie einen Vorteil von Huffman-Codierung gegenüber Shannon-Fano-Codierung. (1 Punkt)
3. Konstruieren Sie den Huffman-Baum für obige Bildquelle. Hierbei sollen die mit der kleineren Auftrittshäufigkeit verbundenen Kanten mit einer 0 und die mit der höheren Auftrittshäufigkeit verbundenen Kanten mit einer 1 codiert werden. (3 Punkte)

Kopiervorlage: nur für Fachcharakteren

b) Die oben genannten Graubereiche (G1-G8) stellen nun das Alphabet einer Quelle dar. Dieses Alphabet kann entsprechend durch 3-Bit-Binärwörtern $x_3x_2x_1$ codiert werden.

1. Welche minimalen Hamming-Distanzen sind für die Erkennung sowie die Korrektur von Einfachfehlern notwendig? (1 Punkt)

2. Wie viele Prüfbits sind für die Hamming-Codierung des oben definierten Alphabetes notwendig? (1 Punkt)

3. Geben Sie nun die Konstruktionsvorschrift für die Prüfbits entsprechend des Hamming-Code-Verfahrens an. (1 Punkt)

4. Statt der 8 Bereiche soll nun jeder Grauwert im Intervall $[0, 255]$ einzeln codiert werden. Wie viele Prüfbits sind hierbei für die Hamming-Codierung notwendig? (1 Punkt)

Kopiervorlage: nur für Fachschaften

- c) Zeichnen Sie unter ausschließlicher Verwendung von NOR-Gattern ein Schaltnetz, welches das Prüfbit y_1 für ein 3-Bit-Binärwort gemäß Aufgabe b) realisiert. (3 Punkte)

- d) Konstruieren Sie ein Schaltnetz eines Halbsubtrahierers unter ausschließlicher Verwendung von NOR-Gattern, welches zwei 1-Bit-Binärzahlen x und y subtrahiert. Negative Ergebnisse sollen im 2er-Komplement dargestellt werden. (3 Punkte)

Kopiervorlage: nur für Fachschaften

Aufgabe 5 (VHDL)

(16 Punkte)

Im Folgenden sind sowohl die Zustandsüberföhrungsfunktion λ (get_next_state) als auch die Ausgabefunktion δ (get_output) eines endlichen Automaten in VHDL gegeben.

```

type state_t is
  (A, B, C, D, H);
type output_t is
  (L0, L1, R0, R1, X);

function get_next_state(
  state: state_t,
  input: std_logic)
  return state_t is
begin
  case state is
    when A =>
      return B;
    when B =>
      if input = '0' then
        return A;
      else
        return C;
      end if;
    when C =>
      if input = '0' then
        return H;
      else
        return D;
      end if;
    when D =>
      if input = '1' then
        return A;
      end if;
    when H =>
      end case;
  return state;
end function;

```

```

function get_output(
  state: state_t,
  input: std_logic)
  return output_t is
begin
  case state is
    when A =>
      if input = '0' then
        return R1;
      else
        return L1;
      end if;
    when B =>
      if input = '0' then
        return L1;
      else
        return L0;
      end if;
    when C =>
      if input = '0' then
        return R1;
      else
        return L1;
      end if;
    when D =>
      if input = '0' then
        return R1;
      else
        return R0;
      end if;
    when H =>
      end case;
  return X;
end function;

```

Ausgehend von diesen beiden Funktionen sollen Sie in dieser Aufgabe den zugehörigen endlichen Automaten als synchrone Schaltung mit asynchronem Reset in VHDL realisieren.

- a) Vervollständigen Sie folgende entity-Deklaration. Alle Ein- und Ausgänge sollen vom Typ std_logic oder std_logic_vector sein. (2 Punkte)

```

entity automaton is
  port(

  );
end automaton;

```


- b) Wie viele Flipflops werden zur Speicherung des Zustands minimal benötigt? (1 Punkt)
- c) Zeichnen Sie, ausgehend von der VHDL-Spezifikation von λ und δ , den zugehörigen Automatengraphen. Der Startzustand sei A . (4 Punkte)

- d) Vervollständigen Sie den Rumpf der folgenden architecture-Beschreibung unter Verwendung der beiden obigen Funktionen. (3 Punkte)

```
architecture behavioral of automaton is
  signal state : state_t;
  signal output_temp : output_t;
begin
```

```
output <= to_std_logic_vector(output_temp);
end architecture;
```

- e) Implementieren Sie eine Funktion in VHDL, die das Paritätsbit eines in der Länge parametrisierten Bitvektors zurückgibt, sodass gerade Parität entsteht. (3 Punkte)

- f) Geben Sie die Werte der Signale d1, d2 und d3 nach dem nächsten Simulationsschritt des folgenden process an, wenn zu dessen Beginn gilt: d1="110", d2="111" und d3="001". Alle Signale sind vom Typ std_logic_vector(2 downto 0). (3 Punkte)

— *anfangs: d1="110", d2="111", d3="001"*

```
process(d1, d2, d3)
  variable temp : std_logic_vector(2 downto 0) := "000";
begin
  temp := d2 and d3;
  d1 <= temp xor temp;
  temp := not temp;
  d3 <= d3 and temp;
  d2 <= d1;
end process;
```