

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

Klausur Grundlagen der Technischen Informatik

20. Juni 2020

Vorname	
Nachname	
Matrikelnummer	
Raum	<input type="checkbox"/> Mensa <input type="checkbox"/> Tentoria
Sitzplatz	

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) Sie können bei der Aufsicht zusätzliche Bearbeitungsblätter anfordern.
 - e) Unleserliches wird nicht bewertet.
 - f) Bei mehreren präsentierten Lösungen wird die Aufgabe nicht gewertet. Streichen Sie daher bei Angabe mehrerer Lösungsansätze die nicht zu bewertenden Lösungen durch.
 - g) 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

Aufgabe 1 (Zahlensysteme)

(16 Punkte)

a) Wie lautet der Wertebereich einer n -stelligen Binärzahl im Zweierkomplement? (2 Punkte)

b) Nachrichten in "Latin-1"- bzw. "Base32"-Codierung können als Zahlen codiert in zwei unterschiedlichen polyadischen Zahlensystemen mit 256 bzw. 32 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 Acht-Zeichen-Nachricht "JFXGMMJS" in "Base32"-Codierung, welche in Tabelle 2 genauer spezifiziert ist, in die "Latin-1"-Codierung um.

(5 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	£	¤	¥	¦	§	¨	©	^a	«	¬	SHY	®	-
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 "Base32"-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	2	3	4	5	6	7

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

(1 Punkt)

d) Wie lautet der Wertebereich einer n -stelligen vorzeichenlosen Ternärzahl (Zahl im Zahlensystem zur Basis drei)? (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

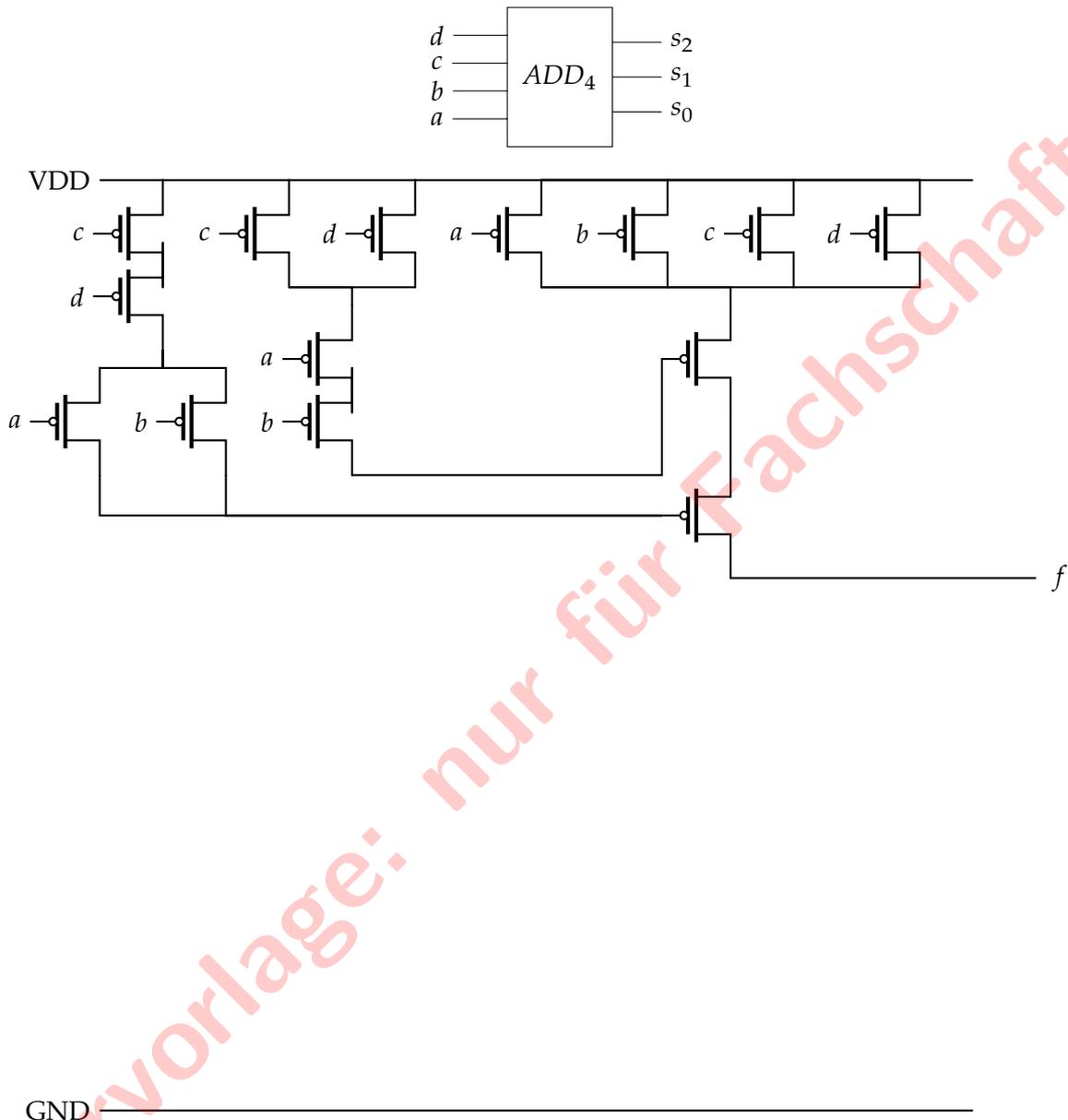
Führen Sie nun die Multiplikation der beiden in diesem Format dargestellten Gleitkommazahlen $1\ 10101\ 1110000000$ und $1\ 01010\ 0001010000$ aus, und geben Sie das Ergebnis im gleichen Format an. (6 Punkte)

Kopiervorlage: nur für Fachschaften

Aufgabe 2 (Minimierung von Schaltfunktionen)

(16 Punkte)

a) Im Folgenden sei das Pull-Up-Netz einer CMOS-Schaltung gegeben, die eines der Summenbits $s_i = f(a, b, c, d)$ eines 4-Bit-Addierers $ADD_4(d, c, b, a) = s_2 s_1 s_0$ realisiert.



1. Bestimmen Sie die Schaltfunktion von f .

(2 Punkte)

2. Vervollständigen Sie die CMOS-Schaltung um das Pull-Down-Netzwerk.

(4 Punkte)

3. Bestimmen Sie eine konjunktive Minimalform (KMF) für das Summenbit s_1 , dessen Schaltfunktion durch folgendes Symmetriediagramm gegeben ist. (2 Punkte)

a			
0	0	1	0
<small>0</small>	<small>1</small>	<small>5</small>	<small>4</small>
0	1	1	1
<small>2</small>	<small>3</small>	<small>7</small>	<small>6</small>
1	1	0	1
<small>12</small>	<small>13</small>	<small>17</small>	<small>16</small>
0	1	1	1
<small>10</small>	<small>11</small>	<small>15</small>	<small>14</small>
c			

b d

4. Welches der drei Summenbits s_i des Addierers – s_0 , s_1 oder s_2 – berechnet f ? Begründen Sie Ihre Antwort. (2 Punkte)

Kopiervorlage: nur für Fachschaften

- b) Bestimmen Sie alle Prim- und Kernimplikate der durch folgendes Symmetriediagramm definierten Funktion $g(a, b, c, d, e)$ durch Anwendung des Quine/McCluskey-Verfahrens. (6 Punkte)

		a				e			
		a		a		a		a	
	b	-	1	1	0	1	1	1	0
		0	1	1	-	1	1	1	0
		1	1	1	1	1	1	1	-
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	1	0
		1							

Kopiervorlage: nur für Fachschaften

Aufgabe 3 (Automaten)

(16 Punkte)

a) Im Folgenden soll ein Automat entworfen werden, der die Versfüße eines elegischen Distichons erkennt. Dabei erhält der Automat pro Takt entweder eine Kürze (elementum breve, \cup) oder eine Länge (elementum longum, $-$). Wir betrachten nur den Anfang des elegischen Distichons, der lediglich aus Daktylen ($- \cup \cup$) und Spondeen ($- -$) besteht, die in beliebiger Reihenfolge nacheinander auftreten. Der Automat sei als Moore-Automat umzusetzen, der eine potenziell unendlich lange Liste an Längen und Kürzen sequenziell verarbeitet und pro erkanntem Daktylus oder Spondeus den erkannten Versfuß ausgibt. Sollte die Sequenz eine ungültige Struktur aufweisen, wird in einen Fehlerzustand *Error* übergegangen, und jede weitere Eingabe führt zu keiner weiteren Zustandsänderung. Der Automat befindet sich zu Beginn im Zustand *Start*.

Beispiel für eine gültige Struktur: $- \cup \cup - - - - \cup \cup$

Beispiel für ungültige Strukturen: $\cup - \cup \cup, - - \cup -, - - \cup \cup$

Eingaben:

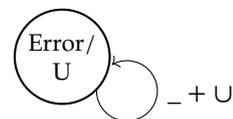
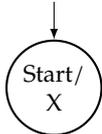
- Länge (Eingabe $-$).
- Kürze (Eingabe \cup).

Ausgaben:

- Es wurde noch kein Versfuß erkannt (Ausgabe X).
- Ein Daktylus wurde erkannt (Ausgabe D).
- Ein Spondeus wurde erkannt (Ausgabe S).
- Ungültige Eingabe (Ausgabe U).

1. Vervollständigen Sie den obig beschriebenen Zustandsautomaten. Die Zustände können beliebig benannt werden.

Hinweis: Eine minimale Umsetzung des Automaten besteht aus sechs Zuständen. (3 Punkte)



2. Geben Sie die Ausgabesequenz für die Verarbeitung des folgenden skandierten Ausschnitts aus dem Werk Ovids an. Die Längen und Kürzen werden von links nach rechts empfangen.

$- - - \cup \cup - - - - - \cup \cup - -$ (1 Punkt)
 ad caelum que manus et splendida brachia tollens

3. Begründen Sie, ob der obige Automat auch als Mealy-Automat repräsentiert werden kann. (1 Punkt)

b) Im Folgenden ist die Automatentafel für einen weiteren Automaten gegeben, der in einem skandierten Gedicht Jamben (∪ —) und Trochäen (— ∪) erkennt.

1. Vervollständigen Sie die Automatentafel unter Verwendung von taktflankengesteuerten JK- bzw. T-Flipflops. (2 Punkte)

Name	Zustand		Eingabe <i>i</i>	Nachfolgezustand		Ansteuerfunktion			Ausgabe	
	<i>q</i> ₁	<i>q</i> ₀		<i>q'</i> ₁	<i>q'</i> ₀	<i>J</i> ₁	<i>K</i> ₁	<i>T</i> ₀	<i>o</i> ₁	<i>o</i> ₀
Start	0	0	0	0	1				0	0
Start	0	0	1	1	0				0	0
Jambus	0	1	0	1	1				1	1
Jambus	0	1	1	0	0				0	1
Trochäus	1	0	0	0	0				1	0
Trochäus	1	0	1	1	1				1	1
Fehler	1	1	0	1	1				1	1
Fehler	1	1	1	1	1				1	1

2. Geben Sie die disjunktive Minimalform (DMF) für *o*₁ an. (2 Punkte)

3. Weisen Sie den möglichen Eingaben und Ausgaben die korrespondierende Codierung aus obiger Automatentafel zu. (2 Punkte)

Eingabe	<i>i</i>	Ausgabe	<i>o</i> ₁	<i>o</i> ₀
—		Fehler		
∪		Trochäus		
		Jambus		
		In Bearbeitung		

4. Zeichnen Sie ein Schaltwerk ohne Ausgabefunktion des realisierten Automaten. Verwenden Sie dazu die folgenden minimierten KMF der Ansteuerfunktionen *J*₁ und *K*₁:

$$KMF_{J_1} = (q_0 + i) \cdot (\bar{q}_0 + \bar{i}), KMF_{K_1} = \bar{q}_0 \bar{i}$$

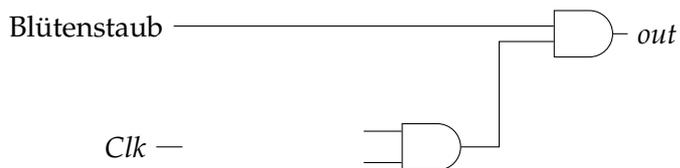
Gehen Sie davon aus, dass *T*₀ als Eingangssignal zur Verfügung steht. (3 Punkte)

5. Geben Sie begründet den Typ des Automaten an.

(1 Punkt)

c) *Irgendwo blüht die Blume des Abschieds und streut immerfort Blütenstaub, den wir atmen, herüber; auch noch im kommenden Wind atmen wir Abschied.* – Rainer Maria Rilke (1875 - 1926)

Nun soll das Signal *Blütenstaub* statt *immerfort* nur bei steigender Flanke des Clock-Signals *Clk* am Ausgang *out* anliegen. Vervollständigen Sie dazu die untenstehende Schaltung. (1 Punkt)



Kopiervorlage: nur für Fachschaften

Kopiervorlage: nur für Fachschaften

Aufgabe 4 (Codierung und Rechnerarithmetik)

(16 Punkte)

- a) In einer Schachspiel-Datenbank werden Partien nach den getätigten Anfangszügen gespeichert. In der Datenbank wurden bisher 33 Partien erfasst. In diesen Partien wurde mit sechs unterschiedlichen Zügen begonnen, welche mit der jeweiligen Auftrittshäufigkeit in folgender Tabelle dargestellt sind. In der Tabelle repräsentiert beispielsweise das Kürzel $Be4$ den Zug „Bauer auf e4“ und $Sf3$ den Zug „Springer auf f3“.

Anfangszug	$Be4$	$Bd4$	$Sf3$	$Bg3$	$Bb3$	$Bc4$	Σ
Auftrittshäufigkeit	12	8	7	3	2	1	33

1. Geben Sie die Gleichung zur Berechnung des Informationsgehalts an. Ordnen Sie anschließend den Informationsgehalt der folgenden Aussagen in aufsteigender Reihenfolge an.

(2 Punkte)

A := „Als Anfangszug wurde auf Reihe 4 gesetzt ($Be4$, $Bd4$ oder $Bc4$)“

B := „Als Anfangszug wurde auf Reihe 3 gesetzt ($Sf3$, $Bg3$ oder $Bb3$)“

C := „Als Anfangszug wurde der Springer auf f3 gesetzt ($Sf3$)“

$$I(\boxed{}) < I(\boxed{}) < I(\boxed{})$$

2. Konstruieren Sie den Huffman-Baum für die obige Tabelle.

(3 Punkte)

Kopiervorlage: nur für Fachschaften

- b) Gegeben sei eine Quelle, deren Alphabet aus 4-Bit-Binärwörtern $x_3x_2x_1x_0$ besteht. Geben Sie die maximale Anzahl an gültigen Codewörtern für eine Hammingdistanz von $HD_{min} = 1$ beziehungsweise $HD_{min} = 3$ an. (2 Punkte)

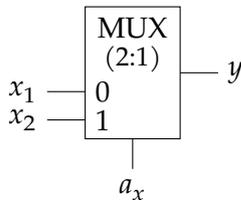
$$HD_{min} = 1 :$$

$$HD_{min} = 3 :$$

- c) Markieren Sie das fehlerhafte Bit in folgender mit gerader Blockparität codierten Nachricht. Die Paritätsbits seien fehlerfrei. (1 Punkt)

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

- d) Realisieren Sie den dargestellten 2:1-Multiplexer, welcher abhängig vom Steuersignal a_x entweder das Eingangssignal x_1 oder x_2 auf den Ausgang y legt, unter ausschließlicher Verwendung von NAND-Gattern. (3 Punkte)



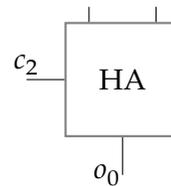
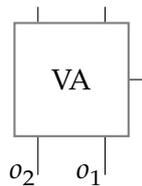
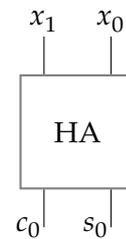
- e) Geben Sie die minimale Anzahl an 2:1-Multiplexern an, um einen 4:1-Multiplexer zu realisieren. (1 Punkt)

f) Im Folgenden soll eine sogenannte popcnt-Operation (*population count*) als Schaltnetz, unter ausschließlicher Verwendung von Halbaddierern und Volladdierern, entworfen werden. Eine popcnt-Operation zählt die Anzahl der vorkommenden Einsen in einem Binärwort. In der folgenden Tabelle sind die resultierenden Rückgabebits $o_2o_1o_0$ der popcnt-Operation für ausgewählte 4-Bit-Binärwörter $x_3x_2x_1x_0$ dargestellt.

Eingabe $x_3x_2x_1x_0$	Ausgabe $o_2o_1o_0$
0 0 0 0	0 0 0
0 0 0 1	0 0 1
...	...
1 1 1 0	0 1 1
1 1 1 1	1 0 0

1. Realisieren Sie zuerst ein Schaltnetz einer popcnt-Einheit für ein 2-Bit-Binärwort x_1x_0 mit den entsprechenden Ausgabebits o_1o_0 . (1 Punkt)

2. Realisieren Sie nun die popcnt-Einheit für ein 4-Bit-Binärwort durch geeignetes Verknüpfen der Summenbits (s_1, s_0) und Übertragbits (c_2, c_1, c_0) mit den entsprechenden Eingängen der gegebenen Halb- beziehungsweise Volladdierer. (3 Punkte)

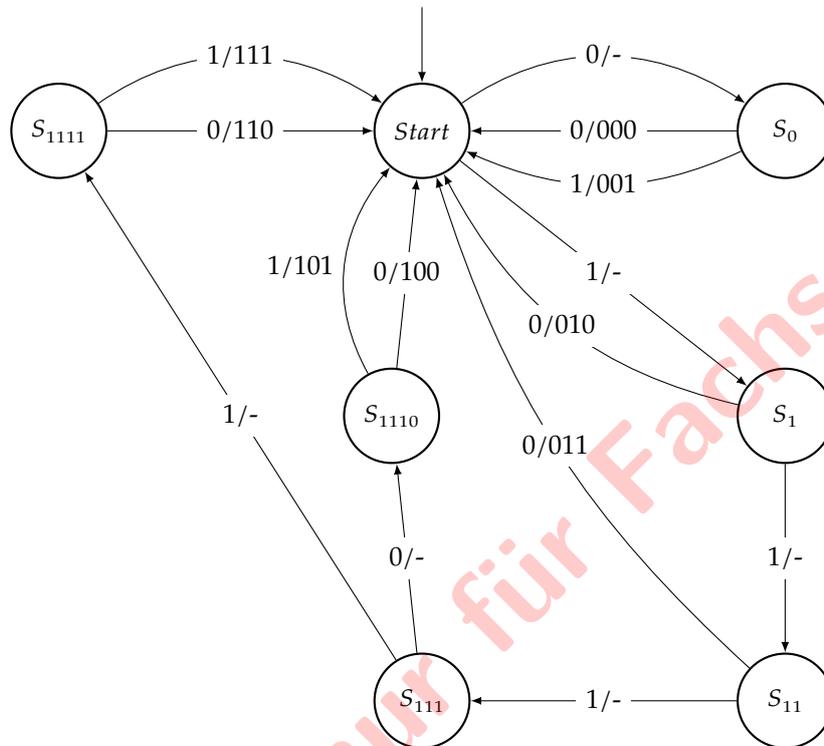


Kopiervorlage: nur für Fachschaften

Aufgabe 5 (VHDL)

(16 Punkte)

Folgender Automat decodiert Codewörter eines vorgegebenen Huffman-Codes in ihre ursprünglichen 3-Bit-Wörter. Eingabe ist jeweils das nächste Bit des zu decodierenden Codewortes; Ausgabe ist das decodierte Wort, falls das Codewort fertig decodiert wurde, oder „-“, falls das Codewort noch nicht fertig decodiert wurde.



Implementieren Sie im Folgenden den spezifizierten Automaten als *synchrone* Schaltung, die sich ebenfalls *synchron* zurücksetzen lässt, in VHDL. Das nächste Eingabebit werde im Signal *in* übertragen. Ist die Ausgabe als ein Bitvektor spezifiziert, soll das Ausgabesignal *out* auf diesen Bitvektor gesetzt werden und ein weiteres Ausgabesignal *out_valid* auf 1 gesetzt werden. Ist die Ausgabe als „-“ spezifiziert, soll das Ausgabesignal *out* unverändert bleiben und *out_valid* auf 0 gesetzt werden.

- a) Vervollständigen Sie folgende entity-Deklaration. Alle Ein- und Ausgänge sollen vom Typ `std_logic` oder `std_logic_vector` sein. (3 Punkte)

```

entity huffman is
    port(

```

```

    );
end huffman;
```

- b) Decodieren Sie folgende Nachricht mittels des spezifizierten Automaten. (1 Punkt)

110000111100

- c) Codieren Sie folgende Nachricht, sodass sie vom spezifizierten Automaten korrekt decodiert werden kann. (1 Punkt)

100101110111

- d) Vervollständigen Sie den Rumpf der folgenden architecture. (10 Punkte)

```
architecture behavioral of huffman is  
  type huffman_state is (start , s0 , s1 , s11 , s111 , s1110 , s1111);  
  signal state : huffman_state;  
begin
```

Kopiervorlage: nur für Fachschaften

end architecture ;

- e) Erläutern Sie die Unterschiede zwischen Datenfluss-, Verhaltens- und Strukturbeschreibung einer Hardware-Komponente in VHDL. (1 Punkt)

Kopiervorlage: nur für Fachschaften