

Prof. Dr. Oliver Keszöcze
Lehrstuhl für Informatik 12
(Hardware-Software-Co-Design)
Friedrich-Alexander-Universität Erlangen-Nürnberg

Klausur Grundlagen der Technischen Informatik

06. April 2022

Vorname	
Nachname	
Matrikelnummer	
Raum	<input type="checkbox"/> BASPH <input type="checkbox"/> H7 <input type="checkbox"/> K1
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

Kopiervorlage: nur für Fachschaften

Aufgabe 1 (Zahlensysteme)

(16 Punkte)

a) Wie heißen die Zahlensysteme zur Basis 8 und 16? (1 Punkt)

b) Wie lautet der Wertebereich einer n -stelligen vorzeichenlosen Zahl im Zahlensystem zur Basis sieben? (2 Punkte)

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

d) Beantworten Sie folgende Auswahlfragen. Jede richtige Antwort gibt einen Punkt, jede falsche Antwort führt zum Abzug eines Punktes, nicht beantwortete Fragen werden nicht gewertet, weniger als null Punkte sind nicht möglich. (3 Punkte)

1. Mit Und- und Oder-Gattern sowie den Konstanten **wahr** und **falsch** lässt sich jede beliebige Schaltfunktion darstellen. wahr falsch

2. Bei der Umwandlung von Ganzzahlen im Quintärsystem in Zahlen im Binärsystem kann es zu Rundungsfehlern kommen. wahr falsch

3. Die Addition zweier Zahlen im Zweierkomplement mit 32 Bit ist aufwändiger als die Addition zweier vorzeichenloser Binärzahlen mit 32 Bit. wahr falsch

e) Konvertieren Sie die Binärzahl 11110000_2 in Vorzeichen-Betrags-Darstellung in die entsprechende Zahl zur Basis 10. (2 Punkte)

f) Geben Sie die fehlenden Codewörter für die folgenden 3-stelligen Binärzahlen in Gray-Codierung an. (2 Punkte)

Dezimalzahl	0	1	2	3	4	5	6	7
Gray-Codewort	000	—	—	010	—	—	—	100

g) Wandeln Sie die Ternärzahl 21200_3 in die entsprechende Zahl zur Basis 6 um. (4 Punkte)

Kopiervorlage: nur für Fachschaften

Aufgabe 2 (Minimierung von Schaltfunktionen)

(16 Punkte)

Im Folgenden ist die Funktionstabelle der Funktion $y = \text{sqrt}(x)$ gegeben, die die Wurzel eines ganzzahligen Wertes $x = 0 \dots 15$ im IEEE half-precision Gleitkommaformat ausgibt, wobei jedes der 16 Bits von y durch eine Schaltfunktion $y_i(x_3, x_2, x_1, x_0)$ beschrieben wird.

x_3	x_2	x_1	x_0	y_{15}	y_{14}	y_{13}	y_{12}	y_{11}	y_{10}	y_9	y_8	y_7	y_6	y_5	y_4	y_3	y_2	y_1	y_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	1	1	1	0	1	1	0	1	0	1	0	0	0
0	0	1	1	0	0	1	1	1	1	1	0	1	1	1	0	1	1	0	1
0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	0	0	0	0	0	0	1	1	1	1	0	0	0
0	1	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	1	1	0
0	1	1	1	0	1	0	0	0	0	0	1	0	1	0	0	1	0	1	0
1	0	0	0	0	1	0	0	0	0	0	1	1	0	1	0	1	0	0	0
1	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	1
1	0	1	1	0	1	0	0	0	0	1	0	1	0	1	0	0	0	1	0
1	1	0	0	0	1	0	0	0	0	1	0	1	1	1	0	1	1	0	1
1	1	0	1	0	1	0	0	0	0	1	1	0	0	1	1	0	1	1	0
1	1	1	0	0	1	0	0	0	0	1	1	0	1	1	1	1	0	1	1
1	1	1	1	0	1	0	0	0	0	1	1	1	0	1	1	1	1	0	1
1	1	1	1	0	1	0	0	0	0	1	1	1	0	1	1	1	1	1	0

a) Bestimmen Sie alle Primimplikate und eine KMF der Funktion y_5 .

(3 Punkte)

				x_0				
				0	1	5	4	
x_1					2	3	7	6
					12	13	17	16
				10	11	15	14	
				x_2				
				x_3				

b) Bestimmen Sie alle DMF der Funktion y_3 mittels des Nelson-Petrick-Verfahrens.

- Bestimmen Sie zunächst alle Primimplikanten der Funktion y_3 mit dem Nelson-Verfahren. (5 Punkte)

		x_0					
		0	0	1	0		
		<small>0</small>	<small>1</small>	<small>5</small>	<small>4</small>		
x_1	1	1	1	0			
	<small>2</small>	<small>3</small>	<small>7</small>	<small>6</small>			
	0	0	1	1			
	<small>12</small>	<small>13</small>	<small>17</small>	<small>16</small>			x_3
1	0	0	1				
	<small>10</small>	<small>11</small>	<small>15</small>	<small>14</small>			
		x_2					

- Vervollständigen Sie die gegebene Überdeckungstabelle mit den Primimplikanten und stellen Sie den Petrick-Ausdruck auf. Geben Sie anschließend alle disjunktiven Minimalformen für y_3 an. (4 Punkte)

k	PI	j								p_i	c_i
		2	3	5	7	10	14	16	17		
0					×				×	A	
1								×	×	B	
2				×	×					C	
3			×		×					D	
4							×	×		E	
5		×	×							F	
6						×	×			G	

- c) Sei $f(x_3, x_2, x_1, x_0) = x_3x_2x_1 + x_3x_0 + x_3x_2\bar{x}_1 + \bar{x}_2x_1x_0 + x_3x_1x_0 + x_3x_1\bar{x}_0$ eine Schaltfunktion für y_9 . Entwerfen Sie eine CMOS-Schaltung für f , die mit möglichst wenig Transistoren auskommt, indem Sie die Funktion f geeignet umformen. Geben Sie das Pull-Up-Netzwerk sowie das Pull-Down-Netzwerk an, und zeichnen Sie die resultierende CMOS-Schaltung in das vorgegebene Diagramm ein. x_0, x_1, x_2 und x_3 seien als Eingänge gegeben, jedoch nicht deren Negationen.

(4 Punkte)

VDD —————

GND —————

Kopiervorlage: nur für Fachschaften

Kopiervorlage: nur für Fachschaften

Aufgabe 3 (Automaten und Flipflops)

(16 Punkte)

a) In dieser Aufgabe soll ein Automat entworfen werden, der das Schere/Stein/Papier-Spiel gegen eine KI modelliert. Das Spiel wird rundenweise gespielt. In jeder Runde wählt der Spielende und die KI zeitgleich eines der drei Symbole 'Schere', 'Stein' und 'Papier' aus. Dabei gilt: 'Schere' schlägt 'Papier', 'Stein' schlägt 'Schere' und 'Papier' schlägt 'Stein'. Wählt der Spielende und die KI die gleiche Eingabe so gilt die Runde als unentschieden und die nächste Runde beginnt. Gewinnt man die Runde gegen die KI, erhält man einen Punkt und spielt weiter. Verliert man jedoch die Runde, ist das Spiel für einen vorbei und es soll in einen 'Verloren'-Zustand übergegangen werden, in dem man unabhängig von der Eingabe bleibt. Entwerfen Sie dieses Spiel als Mealy-Automaten mit den Eingaben 'Schere', 'Stein' und 'Papier' und den Ausgaben '+1' (bei einem Sieg), '0' (bei einem Unentschieden) sowie 'X' bei einer Niederlage. Die KI wählt ihren Zustand automatisch aus, sodass nur die eigene Eingabe zu modellieren sei. Die Strategie der KI besteht darin, zyklisch zuerst 'Schere', dann 'Stein' und schließlich 'Papier' auszuwählen. So nimmt die KI beispielsweise in der ersten und vierten Runde 'Schere', während sie in der zweiten oder fünften Runde auf 'Stein' sowie in der dritten und sechsten Runde auf 'Papier' setzt.

1. Geben Sie den obig beschriebenen Zustandsautomaten als vollständig spezifizierten Mealy-Automaten mit maximal vier Zuständen an. Die Zustände können beliebig benannt werden. (4 Punkte)

2. Begründen Sie, wie viele Zustände eine minimale Umsetzung des obigen Mealy-Automaten als Moore-Automat benötigt. (1 Punkt)

b) Im Folgenden ist die Automatentafel für eine abgeänderte Variante von Schere/Stein/Papier gegeben. Der aktuelle Zustand (q_1q_0) wird hierbei durch ein JK-Flipflop (für q_1) und T-Flipflop (für q_0) gespeichert.

1. Vervollständigen Sie in der Automatentafel die Ansteuerfunktionen für die taktflankengesteuerten JK- bzw. T-Flipflops. (2 Punkte)

Name	Zustand		Eingabe		Nachfolgezustand		Ansteuerfunktion			Ausgabe O
	q_1	q_0	i_1	i_0	q'_1	q'_0	J_1	K_1	T_0	
Sieg	0	0	0	0	1	0				0
Sieg	0	1	0	0	1	0				0
Sieg	1	0	0	0	1	0				0
Sieg	1	1	0	0	1	0				0
Unentschieden	0	0	0	1	0	1				1
Unentschieden	0	1	0	1	0	1				1
Unentschieden	1	0	0	1	0	1				1
Unentschieden	1	1	0	1	0	1				1
Niederlage	0	0	1	0	0	0				0
Niederlage	0	1	1	0	0	0				0
Niederlage	1	0	1	0	0	0				0
Niederlage	1	1	1	0	0	0				0
Unbelegt	-	-	-	-	-	-				-

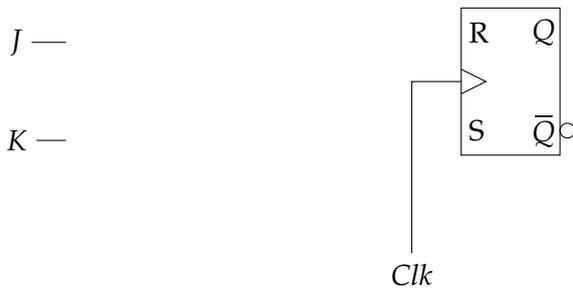
2. Zeichnen Sie ein Schaltwerk ohne Ausgabefunktion des realisierten Automaten. Es gelte:

- $DNF_{J_1} = \bar{q}_1 \bar{q}_0 \bar{i}_1 \bar{i}_0 + \bar{q}_1 q_0 \bar{i}_1 \bar{i}_0$
- $KMF_{K_1} = (i_0 + i_1) \cdot (\bar{i}_0 + \bar{i}_1)$

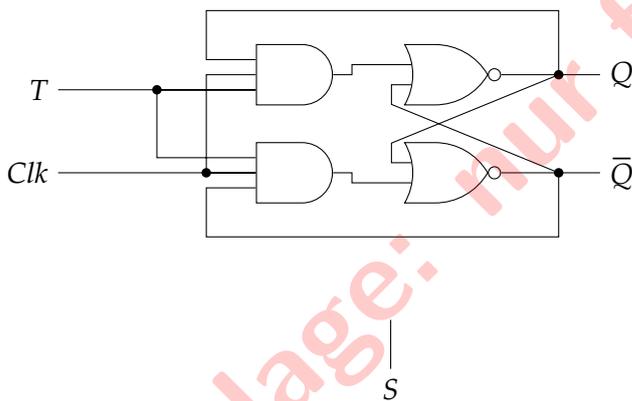
Gehen Sie davon aus, dass T_0 als Eingangssignal zur Verfügung steht. Sie dürfen DNF_{J_1} minimieren. Verwenden Sie nur Gatter mit zwei Eingängen. (3 Punkte)

3. Wie viele FlipFlops werden minimal benötigt, um die Ausgabefunktion O dieses Automaten umzusetzen? (1 Punkt)

c) Erweitern Sie folgendes RS-Flipflop zu einem JK-Flipflop. Sie dürfen dazu die aus der Vorlesung bekannten Logikgatter verwenden. (2 Punkte)



d) Ergänzen Sie folgendes Active-HIGH T-Flipflop um einen asynchronen Set-Eingang S , der unabhängig vom Clock-Signal das Flipflop setzt. Die Logikgatter dürfen hierbei um weitere Eingänge erweitert werden. Weiterhin dürfen die bekannten Gatter aus der Vorlesung verwendet werden. (2 Punkte)



e) Wann ist ein Automat vollständig spezifiziert? (1 Punkt)

(1 Punkt)

Kopiervorlage: nur für Fachschaften

Aufgabe 4 (Codierung und Rechnerarithmetik)

(16 Punkte)

- a) Ein Logger welcher im Segelflugsport eingesetzt wird, misst alle 10 Sekunden den Luftdruck (in hPa), womit sich auf die Flughöhe zurückschließen lässt, und speichert diesen. Dabei wird angenommen, dass sich der Luftdruck im Intervall [201,1013] hPa befindet, welcher äquidistant in sieben Bereiche unterteilt wurde. Im Folgenden wurde für einen Flug über einen Zeitraum von 60 Minuten die folgende Verteilung der Luftdruckmessungen aufgezeichnet:

Luftdruck-Bereich	L1	L2	L3	L4	L5	L6	L7	Σ
Anzahl Messungen	12	30	54	72	108	60	24	360

1. Nehmen Sie an, dass die Luftdruckmessungen die einzigen Daten sind, welche der Logger lokal speichert. Welche Speicherkapazität (in Bit) ist für eine Flugzeit von 20 Minuten nötig um jede Messung zu speichern? Alle Bereiche seien mit der gleichen Anzahl an Bits codiert. (1 Punkt)

2. Geben Sie die Formel zur Berechnung des Informationsgehalts an und berechnen Sie für die obige Quelle den Informationsgehalt der folgenden Aussagen. (2 Punkte)

A := „Der Luftdruck liegt im Bereich [L4, L5]“

B := „Der Luftdruck liegt im Bereich L2 oder L6“

3. Erstellen Sie einen Huffman-Codierungsbaum für die obige Quelle.

(3 Punkte)

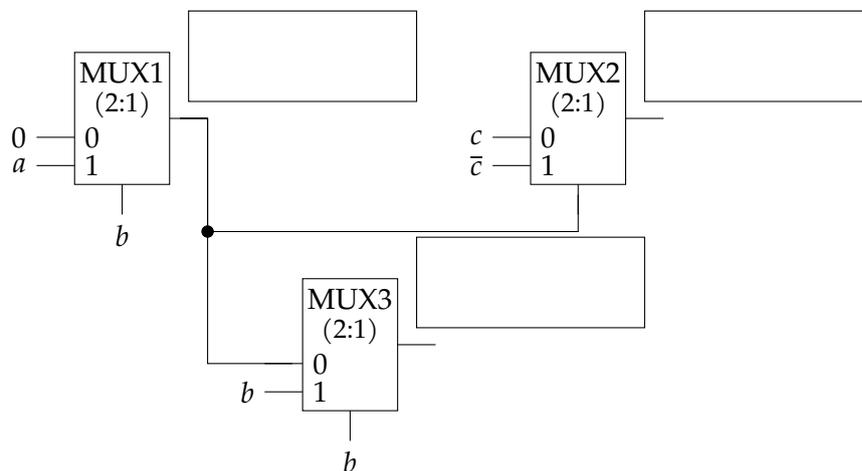
b) (7,4) Hamming-Codes bestehen aus 4 Informationsstellen $x_3x_2x_1x_0$ und 3 Prüfstellen $y_2y_1y_0$ und werden beispielsweise in Quantencomputern eingesetzt, um die Quanteninformation vor Rauschen zu schützen.

1. Welche Arten von Fehlern können mit (7,4) Hamming-Codes erkannt beziehungsweise korrigiert werden? (1 Punkt)
2. Geben Sie die Konstruktionsvorschrift der 3 Prüfstellen für (7,4) Hamming-Codes und ergänzen Sie anschließend die Prüfstellen zur Codierung der gegebenen Informationsstellen. (2 Punkte)

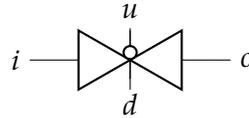
Codewort	Informationsstellen				Prüfstellen		
	x_3	x_2	x_1	x_0	y_2	y_1	y_0
1. CW	0	1	0	1			
2. CW	1	0	1	1			

3. Die Übertragung von Messwerten, die mit jeweils 30 Bits codiert wurden, soll vor Übertragungsfehlern geschützt werden. Geben Sie die minimale Anzahl an Prüfbits zur Codierung der 30-Bit-Binärwörter für eine Hamming-Distanz von $HD_{min} = 3$ an. (1 Punkt)

c) Sei folgendes Schaltnetz aus drei Multiplexern (MUX1, MUX2 und MUX3) gegeben, welche mit den Eingangssignalen a, b, c, \bar{c} und 0 belegt sind. Geben Sie für die Multiplexer MUX1, MUX2 und MUX3 die jeweils resultierende Schaltfunktion an. (3 Punkte)

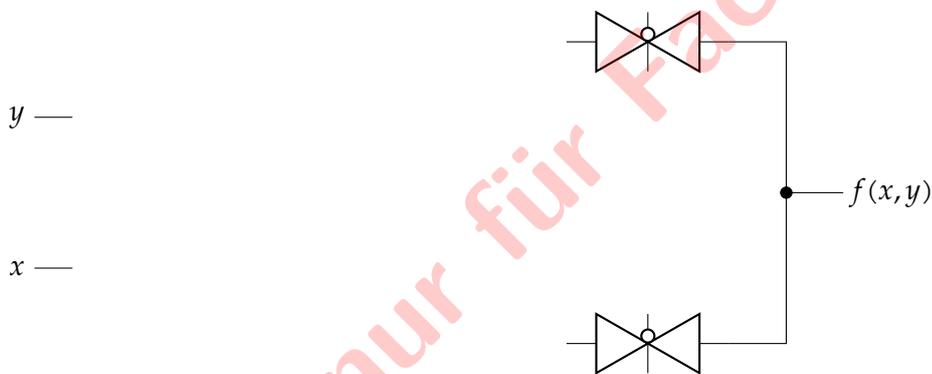


- d) *Transmission-Gates* oder *Übertragungsgater* leiten oder sperren die am Eingang anliegenden Signale abhängig vom Steuersignal. Transmissions-Gates bestehen aus zwei Steuereingängen u und d , sowie einem Eingang i und einem Ausgang o (siehe Abbildung). Das Eingangssignal i wird durchgeleitet (liegt am Ausgang an), falls der untere Steuereingang mit $d = 1$ und der obere mit $u = 0$ belegt wird. Für andere Konfigurationen am Steuereingang sperren sie.



Verschalten Sie die folgenden zwei Transmissions-Gates zu einem Schaltnetz, welches die Funktion $f(x,y) = x \oplus y$ realisiert. Hierfür stehen Ihnen ausschließlich Invertierer und die Signale x und y zur Verfügung. (3 Punkte)

Hinweis: Verwenden Sie das Signal y als Steuersignal beider Transmissions-Gates und das Signal x als Eingangssignal, welches entweder weitergeleitet oder gesperrt wird.



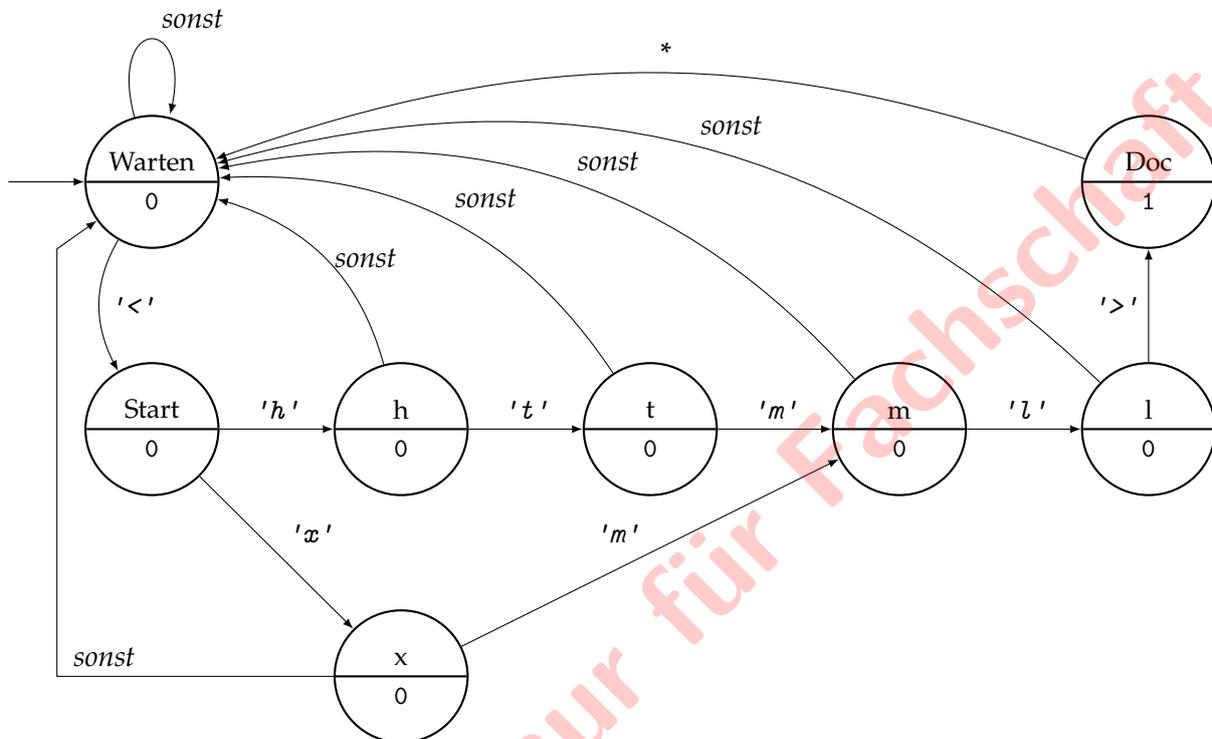
Kopiervorlage: nur für Fachschichten

Kopiervorlage: nur für Fachschaften

Aufgabe 5 (VHDL)

(16 Punkte)

Implementieren Sie einen sogenannten Sniffer (eine Komponente, die Kommunikation überwacht), der URLs in einem Zeichenstrom erkennt. Es sollen XML- und HTML-Dokumente erkannt werden, die mit `<xml>` bzw. `<html>` beginnen. Diese Funktionalität wird durch den im folgenden Automaten-Graphen beschriebenen Automaten implementiert:



Implementieren Sie im Folgenden den spezifizierten Automaten als *synchrone* Schaltung, die sich *asynchron* zurücksetzen lässt, in VHDL. Im Zustand Doc soll das einzige Ausgangssignal `is_doc` auf 1 gesetzt werden, sonst stets auf 0 bleiben. Das eingehende Zeichen werde mit `input` bezeichnet. Weiterhin stehe folgender Datentyp zur Verfügung:

```
type sniffer_state is (warten, start, h, t, x, m, l, doc);
```

- a) Vervollständigen Sie folgende `entity`-Deklaration. Alle Ein- und Ausgänge sollen vom Typ `std_logic` oder `std_logic_vector` sein. Das Eingabezeichen werde als 8-Bit ASCII-Zeichen codiert. (2 Punkte)

```
entity sniffer is  
port(
```

```
);  
end sniffer;
```

- b) Wie viele Flipflops werden zur Speicherung des Zustands minimal benötigt? (1 Punkt)

- c) Vervollständigen Sie folgende Funktion, die für eine gegebene Kombination von Zustand und Eingabe den nächsten Zustand zurückliefert. (4 Punkte)

```
function get_next_state(state: sniffer_state , input: character)
    return sniffer_state is
begin
```

Kopiervorlage: nur für Fachschaften

```
end function ;
```

- d) Vervollständigen Sie folgende Funktion, die zurückliefert, ob eine URL erkannt wurde. (1 Punkt)

```
function is_done(state: sniffer_state) return std_logic is  
begin
```

```
end function;
```

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

```
architecture behavioral of sniffer is  
  signal state : sniffer_state;  
  signal input_char : character;  
  signal is_doc_reg : std_logic;  
begin  
  — vorgegeben: Umwandlung von std_logic_vector nach character  
  input_char <= convert_to_ascii(input);  
  is_doc <= is_doc_reg;
```

```
end architecture;
```

f) Erläutern Sie die Unterschiede zwischen Signalen und Variablen. (2 Punkte)

g) Was macht eine Testbench und welchen Zweck erfüllt sie? (2 Punkte)

h) Erläutern Sie kurz, wieso die funktionale Verifikation eines Hardware-Entwurfs noch wichtiger ist als bei Software. (1 Punkt)

Kopiervorlage: nur für Fachschaften

Kopiervorlage: nur für Fachschaften