

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

Klausur Grundlagen der Technischen Informatik

29. März 2023

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) 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) Wie lautet der Wertebereich einer n -stelligen vorzeichenlosen Zahl im Zahlensystem zur Basis neun? (2 Punkte)
- c) Wie heißen die Zahlensysteme zur Basis 8 und 16? (1 Punkt)
- d) Subtrahieren Sie die beiden BCD-Zahlen 0111 0001 0101 - 0011 0111 1001. Führen Sie die Subtraktion durch, ohne die Binärdarstellung zu verlassen. (3 Punkte)
- e) Wandeln Sie die Ternärzahl 21200_3 in die entsprechende Zahl zur Basis neun um. (2 Punkte)

Kopiervorlage: nur für Fachschaften

- f) In dieser Aufgabe sollen Zahlen im *IEEE half-precision* Gleitkommaformat betrachtet werden. 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 10010 0110110000 und 1 01011 1001010000 aus, und geben Sie das Ergebnis im gleichen Format an. (6 Punkte)

Kopiervorlage: nur für Fachschaften

Aufgabe 2 (Minimierung von Schaltfunktionen)

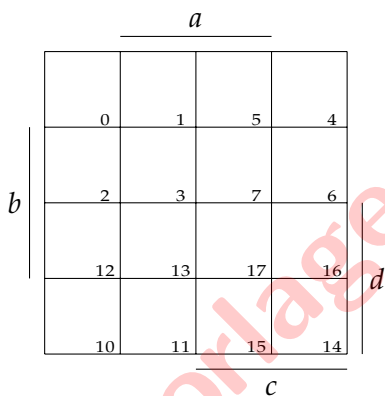
(16 Punkte)

Im Folgenden ist die Funktionstabelle der Funktion $y = \log_2(x)$ gegeben, die den Logarithmus Dualis einer vorzeichenlosen 4-Bit-Binärzahl $x = dcba$ im IEEE half-precision Gleitkommaformat ausgibt, wobei jedes der 16 Bits von y durch eine Schaltfunktion $y_i(d, c, b, a)$ beschrieben wird.

d	c	b	a	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	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	1	1	1	1	0	0	1	0	1	0	1	1	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	1	0	1	0	0	1	0	0
0	1	1	0	0	1	0	0	0	0	0	1	0	0	1	0	1	0	1	1
0	1	1	1	0	1	0	0	0	0	0	1	1	0	0	1	1	1	0	1
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0
1	0	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	1	1	1
1	0	1	0	0	1	0	0	0	0	1	0	1	0	1	0	0	1	1	0
1	0	1	1	0	1	0	0	0	0	1	0	1	1	1	0	1	0	1	1
1	1	0	0	0	1	0	0	0	0	1	1	0	0	1	0	1	0	1	1
1	1	0	1	0	1	0	0	0	0	1	1	0	1	1	0	0	1	1	0
1	1	1	0	0	1	0	0	0	0	1	1	1	0	0	1	1	1	1	1
1	1	1	1	0	1	0	0	0	0	1	1	1	1	0	0	1	0	0	0

a) Bestimmen Sie alle Primimplikate und eine KMF der Funktion y_0 .

(3 Punkte)



b) Bestimmen Sie alle disjunktiven Minimalformen (DMF) der Funktion y_1 .

1. Bestimmen Sie zunächst mittels Quine-McCluskey-Verfahren alle Primimplikanten von y_1 .
(5 Punkte)

2. Vervollständigen Sie die gegebene Überdeckungstabelle mit den bestimmten Primimplikanten, und stellen Sie den Petrick-Ausdruck auf. Geben Sie anschließend alle disjunktiven Minimalformen für y_1 an.
(4 Punkte)

k	PI	j									p_i	c_i	
		3	6	10	11	12	13	14	15	16			
0				×	×	×	×					A	
1				×		×		×		×		B	
2				×	×			×	×			C	
3		×					×					D	
4			×							×		E	

- c) Die Funktion $f(d, c, b, a) = (d + c) \cdot (d + c + a) \cdot (\bar{d} + b) \cdot (d + \bar{c} + a)$ ist eine Schaltfunktion für y_7 . Vereinfachen Sie die Funktion und entwerfen Sie eine CMOS-Schaltung für f . Für die Schaltung sind a, b, c und d als Eingänge verfügbar, jedoch nicht deren Negationen. Geben Sie das Pull-Up-Netzwerk sowie das Pull-Down-Netzwerk an, und zeichnen Sie die resultierende CMOS-Schaltung in das vorgegebene Diagramm ein. (4 Punkte)

VDD —————

GND —————

Kopiervorlage: nur für Fachschaften

Kopiervorlage: nur für Fachschaften

Aufgabe 3 (Automaten und Flipflops)

(16 Punkte)

- a) Professor Eich leitet den Lehrstuhl für Koleopterologie an der Uni Alabastia und ist hobbymäßig begnadeter Sänger. Um seine Gesangslinien zu verbessern, beauftragt er Sie mit dem Erstellen eines Automaten, mit dem man den Rhythmus einer Gesangslinie analysieren kann. Dazu wird der Gesang zuerst in den $\frac{4}{4}$ Takt untergliedert, wobei in Eichs Gesangslinien in jedem Takt nur folgende Notenlängen auftreten: Viertelnoten V (Länge(\downarrow) = $\frac{1}{4}$), punktierte Viertelnoten P , die um die Hälfte länger als eine Viertelnoten sind (Länge(\downarrow) = $\frac{3}{8}$), und halbe Noten H (Länge(\downarrow) = $\frac{1}{2}$). Der Automat erhält als Eingabe periodisch eine Notenlänge und soll '1' ausgeben, wenn nach dem Singen der Note der Takt abgeschlossen wird. Ansonsten soll '0' ausgegeben werden. Beim erfolgreichen Abschluss eines Taktes soll wieder in den Anfangszustand gesprungen werden. Falls mit dem Singen einer Note ein Takt überschritten wird, so soll in einen Fehlerzustand übergegangen werden, in dem man unabhängig von den Eingaben bleibt. Beispiel: Folgt nach einer halben Note und einer Viertelnote wiederum eine halbe Note, so wird die Taktlänge überschritten ($\frac{1}{2} + \frac{1}{4} + \frac{1}{2} > \frac{4}{4}$) und der Fehlerzustand betreten (Ausgabe '0'). Folgt jedoch nach einer halben Note und einer Viertelnote wiederum eine Viertelnote, so wird der Takt abgeschlossen ($\frac{1}{2} + \frac{1}{4} + \frac{1}{4} = \frac{4}{4}$), was mit der Ausgabe '1' signalisiert wird.

1. Geben Sie den obig beschriebenen Zustandsautomaten als vollständig spezifizierten Mealy-Automaten mit maximal acht 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)

3. Im Folgenden ist ein Extrakt aus drei Takten des Songs *So wie es war* von Harald Stile gegeben, welches von links nach rechts vom Automaten verarbeitet wird. Geben Sie sowohl die Eingabe- als auch die Ausgabesequenz an. (2 Punkte)

$\text{♩} = 165$

Es ist nicht so | wie es mal | war ohh

- b) Im Folgenden ist die Automatentafel für eine abgeänderte Variante des Rhythmusautomaten gegeben. Der aktuelle Zustand (q_1q_0) wird hierbei durch ein JK-Flipflop (für q_1) und T-Flipflop (für q_0) gespeichert. Vervollständigen Sie in der Automatentafel die Ansteuerfunktionen für die taktflankengesteuerten JK- bzw. T-Flipflops. (3 Punkte)

Name	Zustand		Eingabe		Nachfolgezustand		Ansteuerfunktion			Ausgabe
	q_1	q_0	i_1	i_0	q'_1	q'_0	J_1	K_1	T_0	O
Viertel	0	0	0	0	0	1				0
Viertel	0	1	0	0	1	0				1
Viertel	1	0	0	0	0	0				0
Viertel	1	1	0	0	1	0				1
Punktiert	0	0	0	1	1	1				1
Punktiert	0	1	0	1	1	1				0
Punktiert	1	0	0	1	0	1				1
Punktiert	1	1	0	1	0	0				1
Halbe	0	0	1	0	0	1				0
Halbe	0	1	1	0	1	0				0
Halbe	1	0	1	0	1	0				1
Halbe	1	1	1	0	0	0				1
Unbelegt	-	-	-	-	-	-				-

- c) Wie viele Flipflops braucht man für einen arbiträren Automaten mit 129 Zuständen minimal? (1 Punkt)
- d) Worin besteht der Unterschied zwischen einem Schaltnetz und einem Schaltwerk? (1 Punkt)

- e) Es sei die untenstehende Beschreibung eines Zustandsautomaten in VHDL gegeben. Zeichnen Sie ein Schaltwerk, das die beschriebene Funktionalität der *architecture* umsetzt. Sie dürfen auf alle aus der Vorlesung und Übung bekannten Schaltbausteine zurückgreifen. Benutzen Sie zur Speicherung des Zustands *state* zwei D-Flipflops. Der Zustand werde initial zurückgesetzt.
Hinweis: Sie benötigen für die Lösung keine Addierer, dürfen diese aber verwenden. (4 Punkte)

```
entity automaton is
  port(
    clk, rst : in std_logic;
    output : out std_logic
  );
end entity;

architecture behaviour of automaton is
  signal state : unsigned(1 downto 0);
begin
  updater: process(clk)
  begin
    if rising_edge(clk) then
      output <= '0';
      if rst = '1' then
        state <= "00";
      elsif state = "00" then
        state <= state + 2;
      else
        state <= state - 1;
        output <= '1';
      end if;
    end if;
  end process;
end architecture;
```

Kopiervorlage: nur für Fachschaften

Aufgabe 4 (Codierung und Rechnerarithmetik)

(16 Punkte)

- a) Bei der Elektromyographie (EMG) wird die Muskelaktivität mit Hilfe von Elektroden oder Sensoren gemessen, die auf der Hautoberfläche angebracht werden. Hierbei wurden mit einem Sensor mit einer Frequenz von 2 KHz Potentiale im Bereich von $[0, 0,3]$ mV erfasst, die äquidistant in sechs Potentialbereiche unterteilt wurden. Für eine Greifbewegung über einen Zeitraum von einer Sekunde wurde folgende Verteilung der Potentiale aufgezeichnet:

Potential-Bereich	P_1	P_2	P_3	P_4	P_5	P_6	Σ
Anzahl Messungen	600	100	25	75	400	800	2000

1. Angenommen, die EMG-Messungen sind die einzigen Daten, die gespeichert werden. Welche Speicherkapazität (in Bit) wird benötigt, um jede Messung für einen Zeitraum von 20 Sekunden 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 EMG-Messwert liegt im Bereich $[P_3, P_5]$.“
B := „Der EMG-Messwert liegt im Bereich P_2 oder P_3 .“
3. Erstellen Sie einen Huffman-Codierungsbaum für die obige Quelle. (3 Punkte)

- b) Gegeben sei eine Quelle, deren Alphabet aus 3-Bit-Binärwörtern $x_2x_1x_0$ besteht. Geben Sie für das Codewort 000 die restlichen möglichen Codewörter für eine minimale Hamming-Distanz von $HD_{min} = 2$ beziehungsweise $HD_{min} = 3$ an.

(2 Punkte)

$$HD_{min} = 2: 000,$$

$$HD_{min} = 3: 000,$$

- c) Welche minimale Hamming-Distanz ist nötig, um Einfachfehler zu korrigieren?

(1 Punkt)

- d) Der Zähler für führende Nullen ist eine wichtige kombinatorische Schaltung bei der Entwicklung von Fließkomma-Architekturen zur Durchführung der Normalisierungsoperation. In einem Binärwort sind führende Nullen die Nullziffern an den höchstwertigen Stellen des Binärwortes bis zu der Stelle, an der die erste 1 steht. Für das 3-Bit-Binärwort $x = 001_2$ ist die Anzahl der führenden Nullen zwei. Die folgende Tabelle zeigt die resultierenden Rückgabebits q_1q_0 eines Zählers, der die führenden Nullen von 3-Bit-Binärwörtern $a_2a_1a_0$ zählt, wobei "–" einem *don't care* entspricht.

a_2	a_1	a_0	q_1	q_0
1	–	–	0	0
0	1	–	0	1
0	0	1	1	0
0	0	0	1	1

1. Realisieren Sie das Schaltnetz für q_1 unter ausschließlicher Verwendung von NOR-Gattern mit zwei Eingängen.

(1 Punkt)

2. Realisieren Sie das Schaltnetz für q_0 unter ausschließlicher Verwendung von NOR-Gattern mit zwei Eingängen.

(3 Punkte)

- e) Gegeben sei die Schaltfunktion $f_1(x_2, x_1, x_0) = x_0 \cdot (x_1 \oplus x_2)$. Entwickeln Sie f_1 für die Variablenordnung x_0, x_1, x_2 nach dem Entwicklungssatz der Schaltalgebra. Geben Sie alle Zwischenschritte an. Zeichnen Sie anschließend das resultierende *Ordered Binary Decision Diagram* (OBDD). (3 Punkte)

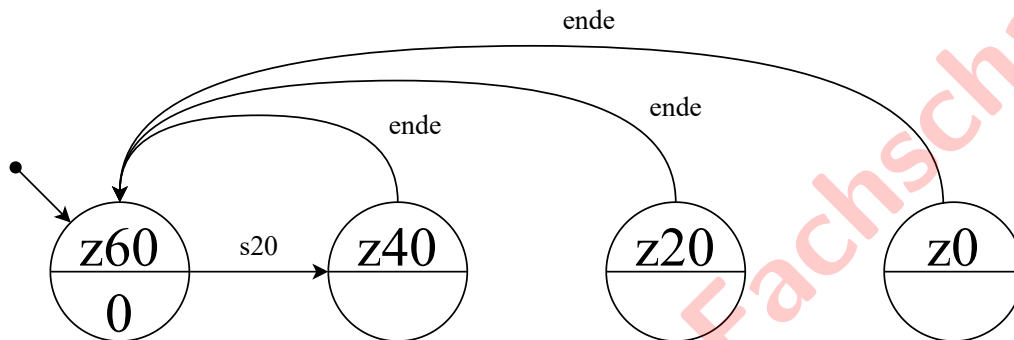
Kopiervorlage: nur für Fachschaften

Kopiervorlage: nur für Fachschaften

Aufgabe 5 (VHDL)

(16 Punkte)

Da Professor Torbens Lieblingslied *The Boxer* von *Simon & Garfunkel* ist, entwickelt er gerade ein Videospiel, das Boxkämpfe virtuell nachbildet. Hierbei ist es nötig, den Status der Boxer zu überwachen, die gegeneinander antreten. Jeder Boxer hat zu Beginn jeder Runde 60 Punkte. Für einen leichten Treffer werden dem Boxer 20 Punkte abgezogen, für einen schweren Treffer 40 Punkte, während dem Boxer bei keinem Treffer auch keine Punkte abgezogen werden. Die Punkte eines Boxers können nicht tiefer als auf 0 fallen. Am Ende jeder Runde werden die Punkte des Boxers auf 60 zurückgesetzt, unabhängig davon, wie viele Punkte er zu diesem Zeitpunkt noch hat. Diese Funktionalität wird durch den unten unvollständig spezifizierten Moore-Automaten teilweise beschrieben:



Vervollständigen Sie im Folgenden den spezifizierten Automaten, und implementieren Sie ihn als *synchrone* Schaltung, die *asynchron* zurückgesetzt werden kann, in VHDL. Im Zustand *z0* soll das einzige binäre Ausgangssignal *is_ko* auf 1 gesetzt werden, ansonsten soll es immer auf 0 bleiben. Die Eingangsinformation, ob und welcher Schlag erfolgt ist, wird mit *input* bezeichnet und wird durch folgende Tabelle codiert:

Leichter Treffer (s20):	00
Schwerer Treffer (s40):	01
Ende der Runde (ende):	10
Kein Treffer (s0):	11

Zusätzlich steht folgender Datentyp zur Verfügung:

```
type boxer_state is (z60, z40, z20, z0);
```

- Vervollständigen Sie den unvollständig spezifizierten Automatengraphen, sodass er die oben beschriebene Funktionalität vollständig darstellt. (3 Punkte)
- Vervollständigen Sie folgende *entity*-Deklaration. Alle Ein- und Ausgänge sollen vom Typ *std_logic* oder *std_logic_vector* sein. Die Eingabe werde als 2-Bit *std_logic_vector* codiert: (2 Punkte)

```
entity boxer_tracker is
  port(

);
end boxer_tracker;
```

- 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: boxer_state ,  
    input: std_logic_vector(1 downto 0));  
    return boxer_state is  
begin
```

Kopiervorlage: nur für Fachschaften

```
end function;
```

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

```
architecture behavioral of boxer_tracker is  
  type boxer_state is (z60, z40, z20, z0);  
  signal state : boxer_state;  
begin
```

Kopiervorlage: nur für Fachschaften

```
end architecture;
```

e) Erläutern Sie die Unterschiede zwischen Signalen und Variablen.

(2 Punkte)

f) Was macht eine Testbench, und welchen Zweck erfüllt sie?

(2 Punkte)

Kopiervorlage: nur für Fachschaften

Kopiervorlage: nur für Fachschaften

Kopiervorlage: nur für Fachschaften