

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

Klausur

Grundlagen der Technischen Informatik

14. April 2012

Name	
Matrikelnummer	
Studienrichtung	

Aufgabe	1	2	3	4	5	Σ
max. Punkte	15	15	20	10	20	80
erreichte Punkte						
Note						

Organisatorische Hinweise

Bitte sorgfältig lesen und die Kenntnisnahme durch Unterschrift bestätigen

1. Bitte legen Sie Ihren Studentenausweis bereit.
 2. Als Hilfsmittel sind nur Schreibmaterialien und ein beidseitig handbeschriebenes DIN A4-Blatt zugelassen.
 3. Schmierpapier wird nicht abgegeben und auch nicht korrigiert.
 4. Sie können bei der Aufsicht zusätzliche Bearbeitungsblätter anfordern.
 5. Unleserliches wird nicht bewertet.
-

Erklärung

1. 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.
2. 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.
3. Ich habe die obigen Hinweise zur Kenntnis genommen.

Erlangen, den 14. April 2012

.....
Unterschrift

Einwilligung

Ich bin damit einverstanden, dass mein vorläufiges Ergebnis anonymisiert, jedoch unter Angabe der Matrikelnummer, am Mitteilungsbrett und auf der Webseite des Lehrstuhls für Informatik 12 veröffentlicht wird.

Die Bekanntgabe des vorläufigen Ergebnisses begründet keinen Rechtsanspruch.

Die Bekanntgabe des endgültigen Ergebnisses erfolgt ausschließlich durch das Prüfungsamt.

Erlangen, den 14. April 2012

.....
Unterschrift

Aufgabe 1 (Zahlensysteme)

(15 Punkte)

- a) Wie lautet der Wertebereich einer mit n Bit darstellbaren vorzeichenlosen Binärzahl? (1 Punkt)
- b) Wie lautet der Wertebereich einer n Bit Binärzahl in 2er-Komplement-Darstellung? (1 Punkt)
- c) Konvertieren Sie die Zahl 100010100101_2 in das Oktal- und Hexadezimalsystem. (1 Punkt)
- d) Konvertieren Sie die Dezimalzahl -137_{10} in eine 9 Bit breite Binärzahl in 2er-Komplement-Darstellung. (2 Punkte)
- e) Beschreiben Sie den allgemeinen Aufbau einer Zahl Z mit n Stellen in einem polyadischen Zahlensystem zur Basis B . (1 Punkt)
- f) Ein 5 Bit Binärstring $(b_4b_3b_2b_1b_0)_2$ wird verwendet um eine Festkommazahl x mit 2 Nachkommastellen darzustellen, das heißt $x = \sum_{0 \leq i \leq 4} 2^{i-2} b_i$. Wandeln Sie $3,75_{10}$ in das spezifizierte Festkommaformat um. (2 Punkte)

- g) Verwenden Sie das aus Aufgabe 1f) bekannte Format und multiplizieren Sie die Festkommazahlen 00101_2 und 00111_2 . Rechnen Sie dabei in Binärdarstellung und stellen Sie das Ergebnis im spezifizierten Festkommaformat dar. Der Genauigkeitsverlust soll dabei so gering wie möglich gehalten werden. (3 Punkte)

- h) In dieser Aufgabe soll mit Gleitkommazahlen nach dem IEEE-Standard 754 gearbeitet werden. Das Format der Gleitkommazahl sieht dabei wie folgt aus: *Vorzeichen (1 Bit), Exponent (8 Bit), Mantisse (23 Bit)*

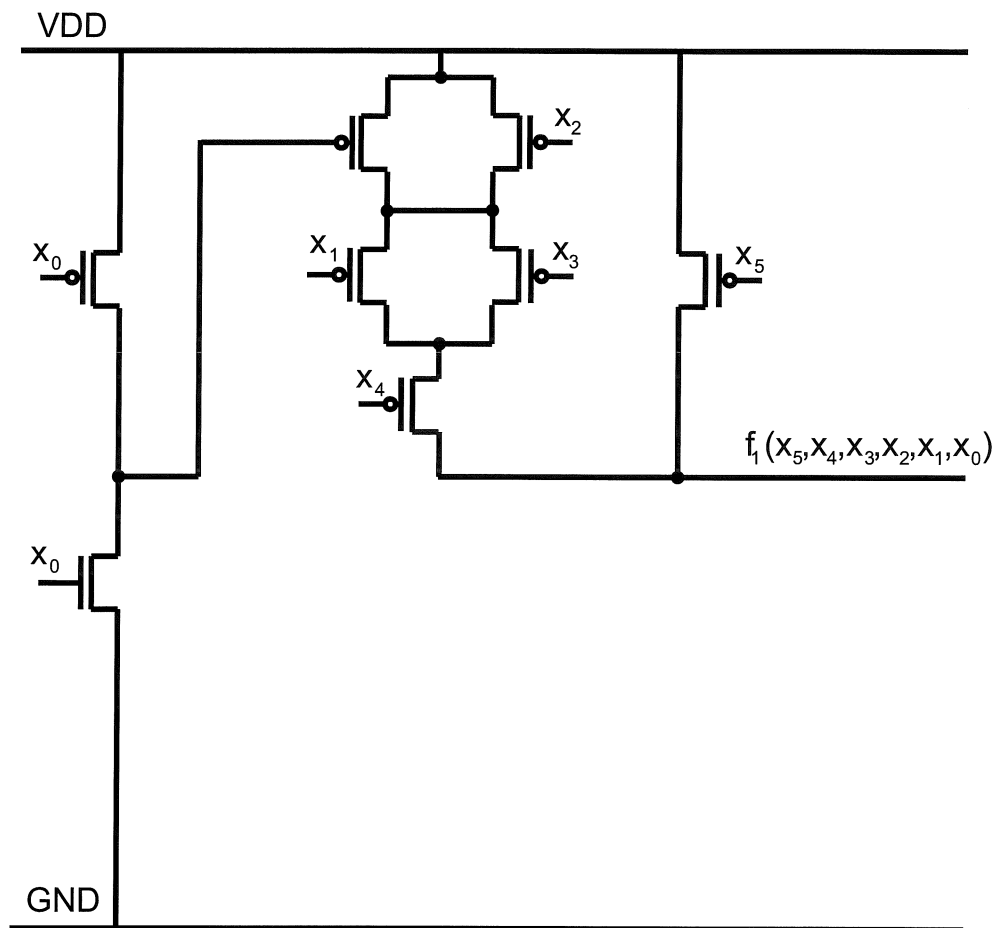
<i>V</i>	<i>E</i>	<i>M</i>
31	30 23	22 0

- Wie lautet die im Format des IEEE-Standard 754 gegebene Gleitkommazahl $11000001.00001010.00000000.00000000_2$ im Dezimalsystem? (4 Punkte)

Aufgabe 2 (Minimierung von Schaltfunktionen)

(15 Punkte)

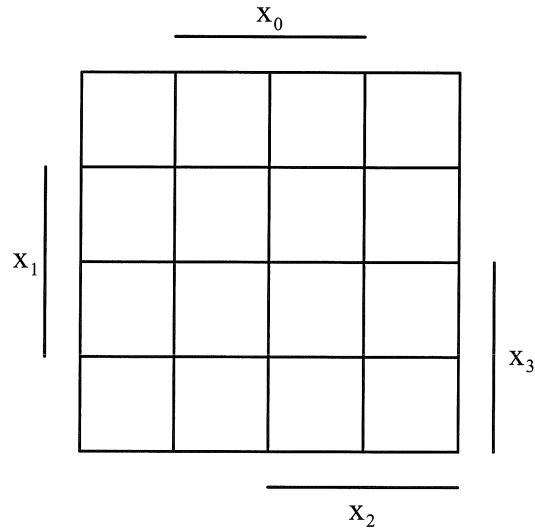
- a) Ergänzen Sie in nachfolgender Abbildung das Pull-Down-Netzwerk der dargestellten CMOS-Schaltung. (3 Punkte)



- b) Kennzeichnen bzw. beschriften Sie in der Abbildung aus Teilaufgabe 2a) die NMOS- und PMOS-Transistoren. (1 Punkt)
- c) Geben Sie den schaltalgebraischen Ausdruck der in Teilaufgabe 2a) realisierten Schaltfunktion $f_1(x_5, x_4, x_3, x_2, x_1, x_0)$ an. (2 Punkte)

- d) Gegeben sei die Funktionstabelle für die Schaltfunktion $f_2(x_3, x_2, x_1, x_0)$. Bestimmen Sie alle Primimplikate mit Hilfe eines Symmetriediagramms. Zeichnen Sie dazu die gefundenen Blöcke in das Symmetriediagramm ein und geben Sie diese zusätzlich algebraisch an. Welche von diesen Termen sind Kernimplikate? (4 Punkte)

x_3	x_2	x_1	x_0	$f_2(x_3, x_2, x_1, x_0)$
0	0	0	0	-
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	-
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

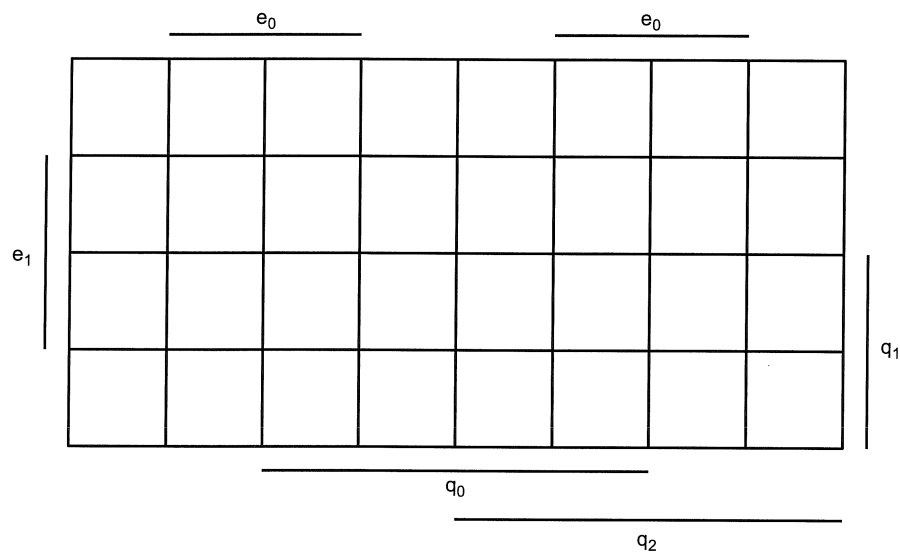


Primimplikate:

Kernimplikate:

- e) Gegeben sei die Nullblocküberdeckung $f_3 = (\bar{x}_3 + x_1) \cdot (x_2 + x_1) \cdot (x_3 + x_2 + x_0)$ sowie die Redundanzmenge $R = \{x_3\bar{x}_2x_1\bar{x}_0, x_3\bar{x}_2x_1x_0, x_3x_2x_1\bar{x}_0, \bar{x}_3x_2x_1\bar{x}_0\}$ einer Schaltfunktion f_3 . Bestimmen Sie eine disjunktive Minimalform (DMF) für die Funktion f_3 mit Hilfe des Nelson/Petricks-Verfahrens. Geben Sie hierbei alle Teilschritte an. (5 Punkte)

- d) Bestimmen Sie die disjunktive Minimalfunktion (DMF) der Ansteuerfunktion T_0 mit Hilfe des vorgegebenen Symmetriediagramms. (3 Punkte)



$T_0 =$

- e) Bestimmen Sie unter Einsatz von schaltalgebraischen Methoden die disjunktive Minimalfunktion (DMF) der Ausgabefunktion y . (2 Punkte)

- f) Wie oft leuchtet die Lampe bei Analyse folgender Proteinsequenzen? (2 Punkte)

- i) TATTAGGATAT =
- ii) TATATAT =
- iii) TATAT =
- iv) TATAGATAT =

Aufgabe 4 (Codierung und Arithmetik)

(10 Punkte)

a) Welcher der folgenden Codes ist kein Huffman-Code? Begründen Sie Ihre Antwort und geben Sie den entsprechenden Codierungsbaum an, falls ein gültiger Code gegeben ist. (2 Punkte)

i) $C_1 = [0, 10, 111, 101]$

ii) $C_2 = [00, 010, 011, 10, 110]$

iii) $C_3 = [1, 000, 001, 010, 011]$

b) Das aus fünf Zeichen bestehende Alphabet einer Quelle wird optimal Huffman-codiert. Nach der Codierung wird eine Nachricht, bestehend aus sechs zufällig ausgewählten Zeichen, an einen Empfänger gesendet. Die am Empfänger beobachtete Bitsequenz ist: 11100000010110. Es ist nur bekannt, dass das Alphabet der Quelle entweder mit der Wahrscheinlichkeitsverteilung $W_1 = [A : 0,15; B : 0,03; C : 0,33; D : 0,38; E : 0,11]$ oder $W_2 = [A : 0,14; B : 0,21; C : 0,06; D : 0,32; E : 0,27]$ verteilt ist und bei der Konstruktion des Codes die mit der kleineren Auftrittshäufigkeit verbundene Kante mit einer 0 und die mit der höheren verbundene Kante mit einer 1 codiert wurde. Ermitteln Sie die vom Sender verwendete Verteilung und begründen Sie Ihre Entscheidung durch Angabe der Codierung der einzelnen Symbole und der decodierten Nachricht. (3 Punkte)

- c) Um die vier Bit Codesymbole $X = (x_3, x_2, x_1, x_0)$ einer Quelle gegen Übertragungsfehler abzusichern, wird ein (7,4)-Hamming-Code eingesetzt. (5 Punkte)
- i) Sichern Sie die Nachricht AD81F zeichenweise gegen Übertragungsfehler, indem Sie den Prüfvektor $Y = (y_2, y_1, y_0)$ berechnen und die neue Nachricht angeben. Verwenden Sie die folgende Formatierung: $X \rightarrow (X, Y)$
 - ii) Am Empfänger wird bei einer späteren Übertragung die folgende Bitsequenz beobachtet:
1101111 1011001 1011100 0101001 1010101
Korrigieren Sie eventuell vorhandene Fehler und geben Sie die decodierte Nachricht an.

Aufgabe 5 (VHDL)

(20 Punkte)

Es soll die Steuerung einer Kaffeemaschine (siehe Abb. 1) entwickelt werden. Das Bedienfeld der Kaffeemaschine besitzt die Tasten 'Enter', '+' und '-'. Zuerst soll mit den Tasten '+' und '-' die Tassengröße (Anzahl der Wassereinheiten) gewählt und mit der Taste 'Enter' bestätigt werden. Analog soll für die Wahl der Kaffeestärke und, falls das Modell einen Zuckertank besitzt, für die Zuckerstärke verfahren werden. Nachdem alle Eingaben mit 'Enter' bestätigt wurden, wird der Kaffee zubereitet. Die Steuerung besitzt für jedes Verbrauchsmittel einen eigenen Einheitenzähler, der sicherstellen soll, dass genügend Einheiten des jeweiligen Verbrauchsmittels zur Verfügung stehen, bevor die gewünschte Menge akzeptiert wird.

Wichtig: Alle Eingangssignale der Schaltung (bis auf *Zuckertank installiert*) sind bei Aktivierung nur für einen Takt auf '1' gesetzt.

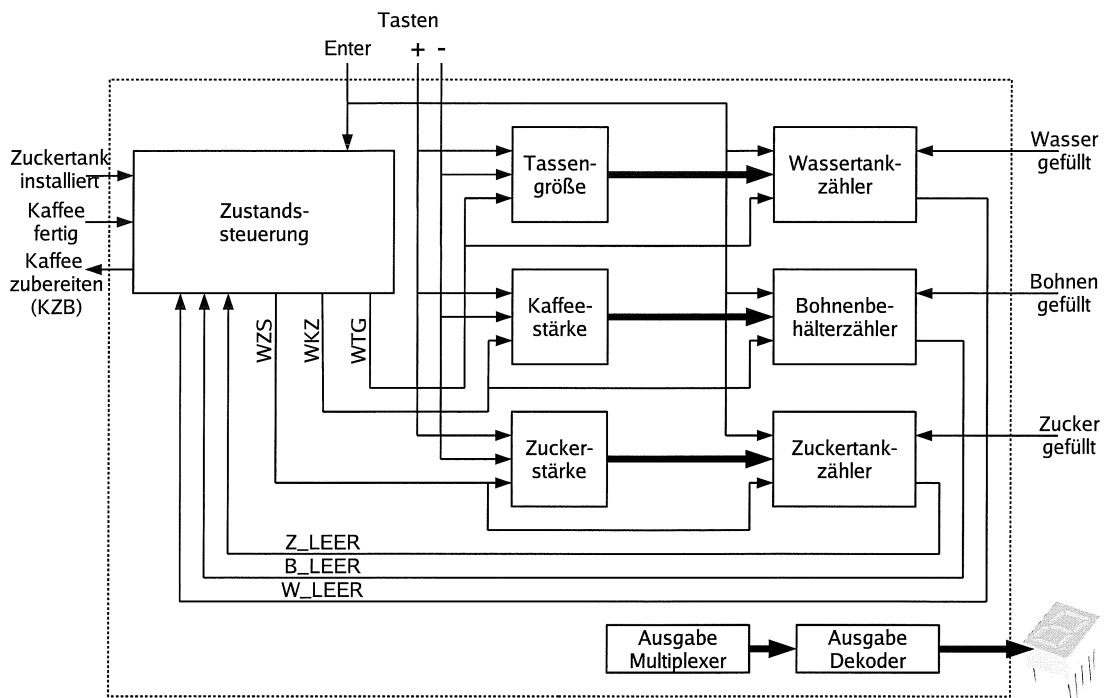


Abbildung 1: Architektur der Steuerung. Signale sind dünn, Signalvektoren fett gezeichnet.

Die zu implementierende synchron getaktete Schaltung besteht aus folgenden Schaltungsblöcken:

- **Zustandssteuerung:** Die Zustandssteuerung bildet ein synchron getaktetes Schaltwerk, welches das in Abbildung 2 gezeigte Zustandsdiagramm implementiert. Es wird immer nur das dem aktuellen Zustand entsprechende Ausgangssignal auf '1' gesetzt, während die restlichen Signale auf '0' zu setzen sind.
- **Tassengröße, Kaffeestärke, Zuckerstärke:** Diese Module dienen der Wahl der Tassengröße, Kaffeestärke und Zuckerstärke. Der Wertebereich der einstellbaren Einheiten soll zwischen 0 und 9 liegen.
- **Wassertankzähler, Bohnenbehälterzähler, Zuckertankzähler:** Diese Module speichern den aktuellen Tank- bzw. Behälterinhalt. Außerdem signalisieren sie der Zustandssteuerung, ob die angeforderte Menge noch vorhanden ist. Wird der entsprechende Tank bzw. Behälter gefüllt, werden die Zähler zurückgesetzt.

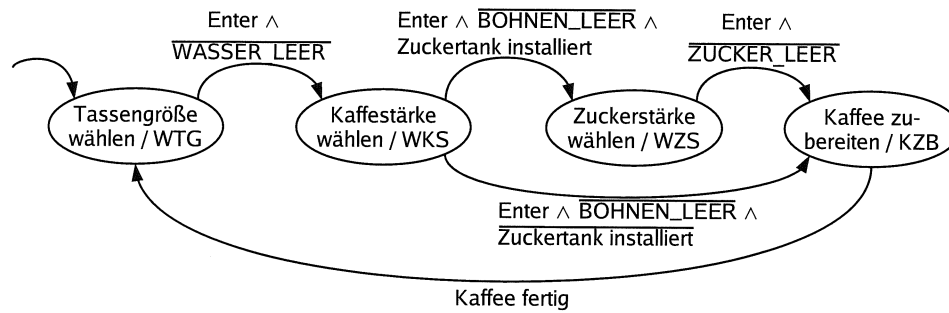


Abbildung 2: Zustandsdiagramm des Moore-Automaten, der im Block *Zustandssteuerung* implementiert werden soll. Es sind nur Kombinationen, die einen Zustandswechsel bewirken annotiert.

- **Ausgabe Multiplexer:** Dieser selektiert in Abhängigkeit des Zustands der Zustandssteuerung die zur Anzeige benötigten Signale.
 - **Ausgabe Dekoder:** Dieser nimmt den binären Wert entgegen und steuert die LED-Anzeige an.
- a) Welche zwei wichtigen Signale, die zur Zustandsspeicherung und Initialisierung unabdingbar sind, fehlen in Abb.1? (1 Punkt)
- b) Geben Sie die Schnittstellenbeschreibung des Schaltungsblockes *Zustandssteuerung* in Form einer Entity in VHDL an. (3 Punkte)

```
entity zustandssteuerung is
```

```
end entity;
```

- c) Geben Sie eine Implementierung des Schaltungsblockes *Zustandssteuerung* in Form einer VHDL Architecture-Beschreibung an. Verwenden Sie hierzu das vorgegebene Code-Skelett (Auf die Angabe von *IEEE-Libraries* kann verzichtet werden.) (7 Punkte)

```
architecture zsl of zustandssteuerung is  
  TYPE te_state IS (t.WTG, _WKS, _WZS, _KZB);  
  signal se_state : te_state;           -- State  
begin -- zsl
```

```
end architecture;
```

- d) Geben Sie eine Implementierung des Schaltungsblockes *Bohnenbehälterzähler* in Form einer VHDL Architecture-Beschreibung an. Der Bohnenbehälter hat eine Kapazität von 220 Einheiten. Um einen Unterlauf zu vermeiden, speichern Sie anstatt der im Behälter vorhandenen, die seit der letzten Füllung verbrauchte Menge. Schreiben Sie Eingangssignale gross und verwenden Sie hierzu das vorgegebene Code-Skelett (*Auf die Angabe von IEEE-Libraries kann verzichtet werden.*) (5 Punkte)

```
architecture btcl of BeanTrunkCounter is
```

```
begin -- btcl  
  pBeanCounter: process (CLK)  
    begin
```

```
    end process; -- pBeanCounter  
end architecture;
```

- e) Zeichnen Sie in Abbildung 1 die Eingänge sowie die Signale zu den Eingängen des *Ausgabe Multiplexers* ein, damit die LED-Anzeige zur Unterstützung des Benutzers die aktuell gewählten Einheiten bei der Auswahl des entsprechenden Verbrauchsmittels anzeigt. (1 Punkt)
- f) Geben Sie eine Implementierung des Schaltungsblockes *Ausgabe Multiplexer* in Form einer VHDL Architecture-Beschreibung an. Verwenden Sie hierzu das vorgegebene Code-Skelett (Auf die Angabe von *IEEE-Libraries* kann verzichtet werden.) (3 Punkte)

```
architecture mux1 of AusgabeMultiplexer is
```

```
begin -- mux1
```

```
end architecture;
```