

Universität Erlangen-Nürnberg  
Technische Fakultät  
Lehrstuhl für Hardware-Software-Co-Design  
Prof. Dr.-Ing. Jürgen Teich

Klausur  
Technische Informatik 1

10. März 2005

Name	
Matrikelnummer	
Studienrichtung	

Aufgabe	1	2	3	4	5	$\Sigma$
max. Punkte	20	20	20	20	20	100
erreichte Punkte						
<b>Note</b>						

## Aufgabe 1 (Codierung und Fehlerkorrektur)

(20 Punkte)

In einer Wetterstation wird zu jeder vollen Stunde die Lufttemperatur gemessen. Dabei wird der gemessene Temperaturwert in Grad Celsius als Binärzahl in 2-er Komplementdarstellung auf ein halbes Grad genau abgespeichert.

- a) Wie viele Bits sind zur Abspeicherung eines Temperaturwertes erforderlich, wenn sich der Messbereich von  $-20 \dots 40$  Grad erstrecken soll? (Begründung)  
Wie groß ist dann die täglich anfallende Datenmenge? (3 Punkte)
- b) Ein Unterschreiten des Messbereichs soll durch die kleinste negative Zahl und ein Überschreiten entsprechend durch die größte positive Zahl codiert werden.  
Geben Sie diese beiden Codierungen an. (2 Punkte)

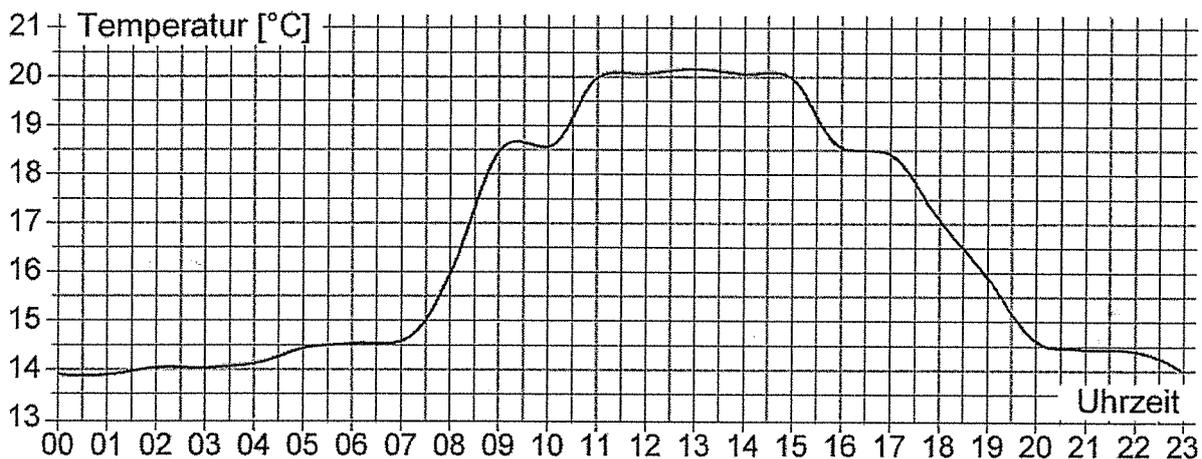


Abbildung 1: Temperaturverlauf eines Tages (0:00 bis 23:00 Uhr).

Der in der Abbildung 1 gegebene Temperaturverlauf soll nun durch Verwendung einer Huffman-Codierung effizienter gespeichert werden.

- c) Benennen Sie kurz die Schritte zum Erstellen eines Huffman-Codes. (2 Punkte)
- d) Erstellen Sie den Huffman-Baum und geben Sie dabei Zwischenschritte an. (6 Punkte)
- e) Wie groß ist nun die an diesem Tag angefallene Datenmenge?  
(Vernachlässigen Sie die Tatsache, dass das Codebuch, bzw. der Codierungsbaum auch gespeichert werden muss) (2 Punkte)

Zur sicheren Datenübertragung an die Wetterzentrale wird ein Hamming-Code mit drei Prüfbits verwendet.

- f) Wie viele Informationsstellen darf dann ein gesendetes Codewort maximal enthalten, damit Einzelbitfehler korrigiert werden können? (Formel und Zahlenwert) (2 Punkte)
- g) Geben Sie den Prüfbitgenerator zur Berechnung der drei Prüfbits an. (3 Punkte)

Hinweis:

Die Diskretisierung der Temperaturwerte soll so erfolgen, dass der entstehende Fehler minimal wird.

## Aufgabe 2 (Zahlendarstellungen)

(20 Punkte)

Seit Douglas Adams' Science-Fiction Serie "The Hitchhikers Guide to the Galaxy" ist allgemein bekannt, dass die Antwort auf das Leben, das Universum und Alles 42 ist. Die passende Frage dazu erfährt man am Ende des Buches "The Restaurant at the End of the Universe", als der Hauptdarsteller Arthur Dent durch zufälliges Aneinanderlegen von Scrabble-Steinen folgendes erhält:

- "Six by nine. — Forty-two."
- "That's it. That's all there is."

Um diese Rechnung zu überprüfen, sollen Sie nun folgendes tun:

- Berechnen Sie  $6 \cdot 9$  in folgender Gleitkommadarstellung:  
Vorzeichen (V) : 1 Bit breit (1: negativ)  
Exponent (E) : 7 Bit breit (somit ist der Bias 63)  
Mantisse (M) : 8 Bit breit (1,M wie beim IEEE-Format üblich)  
Anordnung: VEM  
Geben Sie das Ergebnis als Gleitkommazahl an. (6 Punkte)
- Berechnen Sie  $6 \cdot 9$  im Zahlensystem zur Basis 2.  
Wie heißt dieses Zahlensystem? (5 Punkte)
- Berechnen Sie  $6 \cdot 9$  im Zahlensystem zur Basis 8.  
Wie heißt dieses Zahlensystem? (3 Punkte)
- Wie heißen die Zahlensysteme zur Basis 10 und 16? (2 Punkte)
- Berechnen Sie  $6 \cdot 9$  im Zahlensystem zur Basis 13. (4 Punkte)

Auch wenn das Ergebnis des letzten Aufgabenteils anders vermuten lässt, Douglas Adams sagte einst: "I may be a pretty sad person, but I don't make jokes in base 13."

## Aufgabe 3 (Logikminimierung)

(20 Punkte)

Im Hochhaus zu den Erlen fällt die Anzeige für das aktuelle Stockwerk des Aufzuges aus. Da auf die Schnelle kein Elektroniker aufzutreiben ist, bittet Sie der Hausmeister um Hilfe. Wie Sie schnell feststellen, ist das PLA der Anzeige, welches die Ansteuerungssignale für eine Sieben-Segment-Anzeige (siehe Abb. 2) aus einer BCD-Zahl generiert, defekt. Durch eine kurze Messung bestimmen Sie, dass die Sieben-Segment-Anzeige Active-HIGH ist. In Ihrem Hobbyelektronikkasten finden Sie ein PLA des gleichen Typs. Dieses sollen Sie benutzen, um die Anzeige zu reparieren.

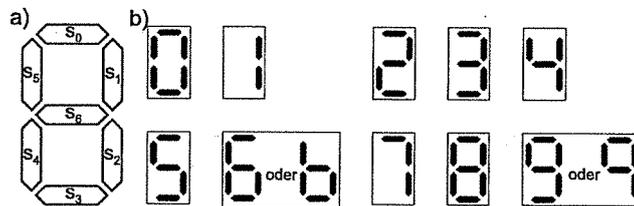


Abbildung 2: a) Sieben-Segment-Anzeige und b) Darstellung der Dezimalzahlen 0 bis 9.

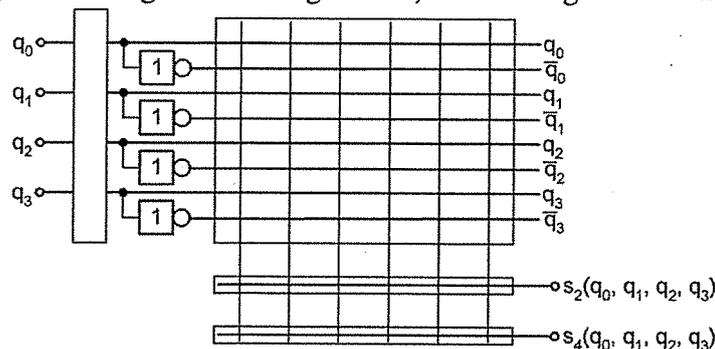


Abbildung 3: Vereinfachte Darstellung für die Kontaktierung Ihres PLAs.

- Stellen Sie die Funktionstabellen für die 7 Ansteuersignale  $s_0$  bis  $s_6$ , welche aus den Eingängen  $q_0$  bis  $q_3$  ermittelt werden, auf. (3 Punkte)
- Bestimmen Sie die konjunktive Minimalform für das Ansteuersignal  $s_0$  mit Hilfe eines Symmetriediagramms. (2 Punkte)
- Bestimmen Sie die konjunktive Minimalform für das Ansteuersignal  $s_2$  mit Hilfe des Quine/McCluskey-Verfahrens. Bestimmen Sie, welche Primimplikate reine Freistellenüberdeckungen sind. (4 Punkte)
- Bestimmen Sie die konjunktive Minimalform für das Ansteuersignal  $s_4$  mit Hilfe des Nelson/Petrick-Verfahrens. (4 Punkte)
- Setzen Sie die zwei Ansteuersignale  $s_2$  und  $s_4$  in eine Programmierung des PLA-Bausteins (Abb. 3) um. (Hinweis: Wandeln Sie dafür die konjunktive Minimalform, welche Sie in Aufgabe 3c) und 3d) ermittelt haben, in eine disjunktive Minimalform um.) (3 Punkte)
- Gegeben sind die folgenden sechs Mengen von Gattertypen: (1) {and, or}, (2) {nand, not}, (3) {nor, not}, (4) {nand}, (5) {nor} und (6) {xor}.

Welche der sechs Mengen von Gattertypen sind ausreichend, um alle Booleschen Funktionen darzustellen?

Stellen Sie für jede Menge von Gattertypen, welche Sie für ausreichend erachten, die drei Grundgatter and, or und not unter ausschließlicher Verwendung von Gattertypen aus dieser Menge dar. (4 Punkte)

## Aufgabe 4 (Automaten und Flipflops)

(20 Punkte)

Gegeben sei folgende Spezifikation einer zu entwerfenden Benutzer-Schnittstelle eines CD-Spielers der Marke Co-Design mit den binären Eingangssignalen 'Start', 'Pause', und 'Stop' (Abb. 4)

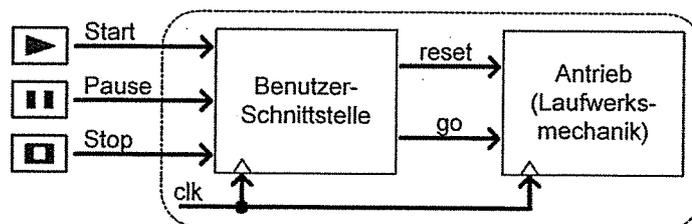
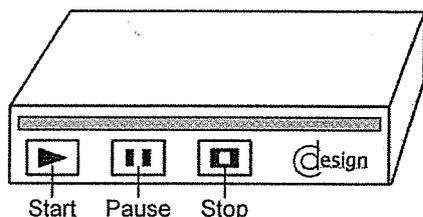


Abbildung 4: Co-Design CD-Spieler.

Abbildung 5: Innere Struktur des CD-Spielers.

Der Spindelmotor und der Antrieb des Laserkopfes soll allein mit dem internen Signal 'go'=1 gesteuert werden (Abspielen)(Abb. 5). Der Ausgang 'reset'=1 positioniert den Lesekopf an den Anfang der CD. Mit Drücken der Pause-Taste soll der Antrieb angehalten werden (go=0), bis erneut die Pause-Taste gedrückt wird (Umschalten zwischen Abspielen und Pause). Nach dem Einschalten soll sich das Schaltwerk im Startzustand befinden, in dem kein Titel abgespielt wird. Der Startzustand kann nur durch Betätigung der Start-Taste verlassen werden. Wird die Taste 'Start' während der Wiedergabe oder während des Pause-Betriebs gedrückt, so soll die Wiedergabe mit dem ersten Titel der CD neu beginnen.

Der CD-Spieler soll unabhängig von allen anderen Eingangssignalen in den Startzustand wechseln, wenn die Stop-Taste gedrückt wird.

Aufgaben für den Entwurf der Benutzer-Schnittstelle:

- Beschreiben Sie das Verhalten der Benutzer-Schnittstelle durch einen Mealy-Automaten, indem Sie für das Schaltwerk den Automatengraphen zeichnen. (4 Punkte)
- Beschreiben Sie das Verhalten durch die entsprechende Automatentafel. (3 Punkte)
- Codieren Sie nun die Zustände geeignet. Wieviele Zustandsspeicher (Flipflops) benötigen Sie? (3 Punkte)
- Wählen Sie nun D-FlipFlops zur Speicherung der Zustände und bestimmen Sie in einer Tabelle die Ansteuerfunktionen dieser FlipFlops. (5 Punkte)
- Minimieren Sie nun die Ausgabefunktionen für 'reset' und 'go' sowie für die Zustandsüberföhrungsfunktion und geben Sie schließlich das Schaltwerk mit entsprechenden FlipFlops und Gattern graphisch an. (Wenn Sie anhand der Tabelle in Punkt d) feststellen, dass ein Zustands- oder Eingabe-Bit keine Auswirkungen auf die Ausgabe oder die Zustandsüberföhrung hat, so brauchen Sie dieses Bit nicht weiter berücksichtigen.) (5 Punkte)

Hinweise:

- Der Einfachheit halber nehmen wir an, dass der CD-Spieler am Ende der CD automatisch wieder mit Titel 0 beginnt, falls nicht wieder die Stopp-Taste gedrückt wird.
- Start, Pause und Stop werden der Benutzerschnittstelle nur für die Dauer eines Taktes lang mit einer '1' signalisiert.
- Sorgen Sie dafür, dass der 'reset'-Eingang des Antriebs nur für eine Taktperiode mit einer '1' aktiviert wird, wenn der Antrieb auf den Anfang springen soll, sonst soll das Signal stets '0' sein.

## Aufgabe 5 (VHDL)

(20 Punkte)

In dieser Aufgabe soll eine einfaches synchrones 4 Bit Divisionsschaltwerk in VHDL entwickelt werden, welche als Dividend und Divisor je eine positive ganze Zahl als Eingabe erhält. Der Dividend besitzt doppelte Genauigkeit, so dass zwei Register zu seiner Speicherung benötigt werden. Zur Division soll das 'Restoring Division'-Verfahren aus der Vorlesung eingesetzt werden.

- a) Um die Arbeitsweise der Divisionseinheit besser verstehen zu können, berechnen Sie:  
 $A : B = Q + R$  (Quotient + Rest) mit  $A=18$  und  $B=5$  unter Verwendung der 'Restoring Division'.  
(Rechnung im Dualsystem) (4 Punkte)
- b) Für  $A$ ,  $B$ ,  $Q$  und  $R$  soll eine minimale Anzahl von Registern verwendet werden. Geben Sie an, wie Sie die von Ihnen gewählten Register ( $R_1 \dots R_n$ ) für  $A$ ,  $B$ ,  $Q$  und  $R$  benutzen. (3 Punkte)
- c) Welche Eigenschaften müssen Dividend und Divisor zur fehlerfreien Berechnung erfüllen?  
(2 Punkte)
- d) Geben Sie in VHDL eine Entity-Beschreibung des 4 Bit Dividierers an. (3 Punkte)
- e) Geben Sie in VHDL eine Verhaltensbeschreibung (Architecture) der Divisionseinheit an. (Die Division darf bis zu fünf Taktzyklen für die Bestimmung eines Ergebnisses benötigen.)  
(8 Punkte)

## Hinweise:

- Ihre Hardware braucht die eigentlich notwendigen Überprüfungen aus Punkt c) nicht berücksichtigen.
- Es steht Ihnen frei, ein Signal 'Start' zu verwenden, welches während der jeweils ersten Taktperiode den Beginn einer Division signalisiert. Dieses (für Sie externe) Signal liefert eine '1' von einer Periode Dauer.
- Während der Division sollen die internen Register, welche Ergebnisse speichern, dem Ausgangssignalen  $Q$  und  $R$  zugewiesen werden. Mehr ist der Einfachheit halber nicht erforderlich.
- Bei dem in der Vorlesung vorgestellten Verfahren wird für den Dividenden ein zusätzliches Bit benötigt, welches unter anderem angibt, wie das in jedem Schritt erhaltene Quotientenbit lautet. In einem Prozessor wird dazu üblicherweise das Carry-Bit verwendet. Hier reicht es alternativ, ein oder mehrere Register zu verwenden, welche 5 statt 4 Bit breit sind.
- Es gibt in VHDL den Konkatenations-Operator & zum Verschmelzen von Signalen und Bitvektoren, dieser ist unter anderem für Schiebeoperationen hilfreich. Bsp. :

```
X <="111001";  
Y <='1' & "0110" & X(2 downto 1); -- ergibt für Y "1011000"
```