

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

Klausur

Grundlagen der Technischen Informatik

27. Juli 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 27. Juli 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 27. Juli 2012

.....
Unterschrift

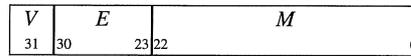
Aufgabe 1 (Zahlensysteme)

(15 Punkte)

- a) Wie lautet der Wertebereich einer mit n Bit darstellbaren vorzeichenlosen Binärzahl? (1 Punkt)
- b) Konvertieren Sie die Dezimalzahl 157_{10} in eine vorzeichenlosen Binärzahl. (1 Punkt)
- c) Wie lautet der Wertebereich einer n Bit Binärzahl in 2er-Komplement-Darstellung? (1 Punkt)
- d) Konvertieren Sie die Dezimalzahl -157_{10} in eine 9 Bit breite Binärzahl in 2er-Komplement-Darstellung. (1 Punkt)
- e) Stellen Sie die Zahl $x = 2^{24} - 4$ im Oktal- und Hexadezimalsystem dar. (2 Punkte)
- f) Konvertieren Sie die Zahl 937 in einen Binärstring in BCD-Darstellung. (1 Punkt)
- g) Stellen Sie die Zahlen 0-15 in Gray-Code dar, und erklären Sie das Einsatzgebiet für die Gray-Codierung. (2 Punkte)

Dezimal	0	1	2	3	4	5	6	7
Gray-Code	0000_2							
Dezimal	8	9	10	11	12	13	14	15
Gray-Code								1000_2

- 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)*



a : 110000010101010101010101010101

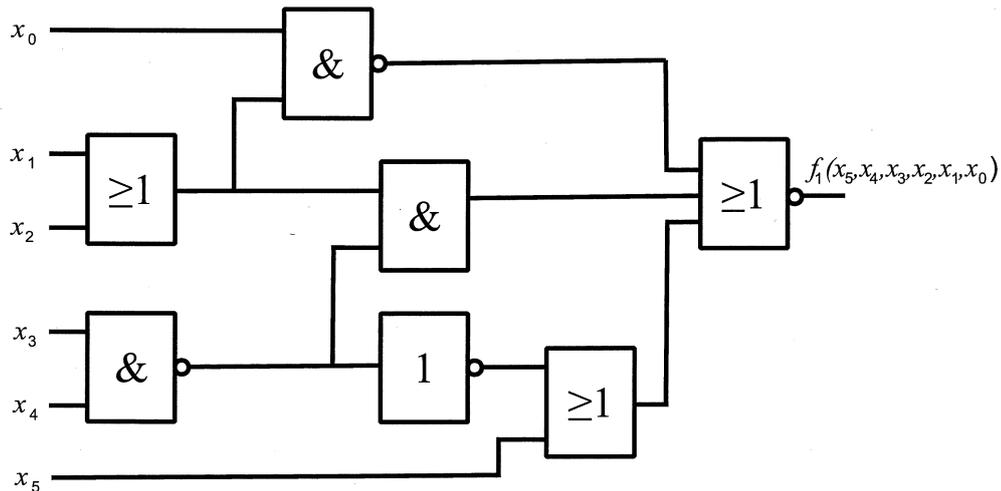
b : 101111100100000000000000000000

Gegeben seien die beiden Gleitkommazahlen a und b . Berechnen Sie den Wert des Produkts $a \cdot b$ als IEEE-754-Gleitkommazahl. (6 Punkte)

Aufgabe 2 (Minimierung von Schaltfunktionen)

(15 Punkte)

- a) Bestimmen Sie einen schaltalgebraischen Ausdruck für die Schaltfunktion $f_1(x_5, x_4, x_3, x_2, x_1, x_0)$, welche durch folgendes Schaltnetz gegeben ist: (2 Punkte)

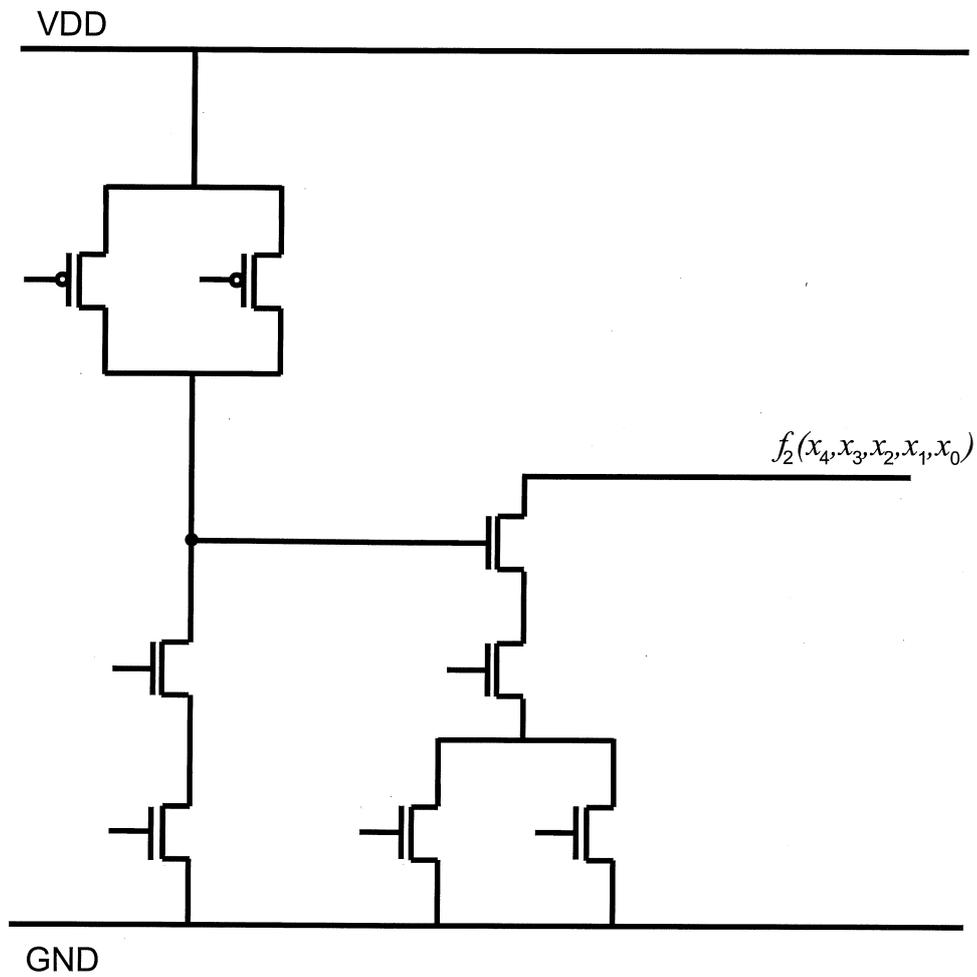


- b) Die Schaltfunktion

$$f_2(x_4, x_3, x_2, x_1, x_0) = \overline{(\overline{x_3} + \overline{x_4}) \cdot (x_2 + x_1) \cdot x_0} + \overline{x_0} \overline{x_2}$$

soll in CMOS-Technologie realisiert werden. Geben Sie hierzu unter Verwendung der Rechenregeln der Booleschen Algebra die Schaltfunktion des Pull-Up-Netzwerks f_{PUN} in Disjunktiver Minimalform (DMF) und die Schaltfunktion des Pull-Down-Netzwerks f_{PDN} in Konjunktiver Minimalform (KMF) an. (3 Punkte)

- c) In der nachfolgenden Abbildung ist die unvollständige CMOS-Schaltung dargestellt, die die Schaltfunktion aus Aufgabe 2b) realisiert. Ergänzen Sie in der Abbildung das Pull-Up-Netzwerk und beschriften Sie alle NMOS- und PMOS-Eingänge. Nehmen Sie hierbei die Ergebnisse aus Aufgabe 2b) zu Hilfe. (2 Punkte)



d) Gegeben sei nachfolgendes Symmetriediagramm. Bestimmen Sie daraus eine gültige KMF der dargestellten Schaltfunktion. (3 Punkte)

		x_0				
		0	1	1	0	
x_1		0	0	1	1	x_3
		-	-	0	1	
		0	1	1	1	
		0	1	1	1	
		x_2				

KMF:

e) Gegeben sei die folgende KNF

$$f_3 = (x_3 + x_2 + x_1 + x_0) \cdot (x_3 + \bar{x}_2 + x_1 + x_0) \cdot (\bar{x}_3 + x_2 + x_1 + x_0) \cdot (\bar{x}_3 + x_2 + x_1 + \bar{x}_0) \cdot (\bar{x}_3 + \bar{x}_2 + \bar{x}_1 + x_0)$$

sowie die Redundanzmenge $R = \{\bar{x}_3 x_2 \bar{x}_1 x_0, \bar{x}_3 \bar{x}_2 x_1 \bar{x}_0, x_3 x_2 \bar{x}_1 \bar{x}_0\}$ der Schaltfunktion f_3 . Bestimmen Sie eine disjunktive Minimalform (DMF) für die Funktion f_3 mit Hilfe des Nelson/Petrick-Verfahrens. Geben Sie hierbei alle Teilschritte an. (5 Punkte)

Aufgabe 3 (Automaten und Flipflops)

(20 Punkte)

- a) Worin unterscheiden sich die drei Automatentypen Mealy, Moore und Medwedev? (1 Punkt)
- b) Kann **jeder** Mealy- in einen äquivalenten Moore-Automat transformiert werden? (1 Punkt)
- c) Gegeben sei der Mealy-Automat in Abbildung 1. Welcher der vier Moore-Automaten in Abbildung 2 spezifiziert das gleiche Ein-/Ausgabeverhalten wie der Mealy-Automat? (2 Punkte)

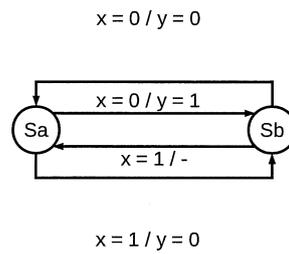
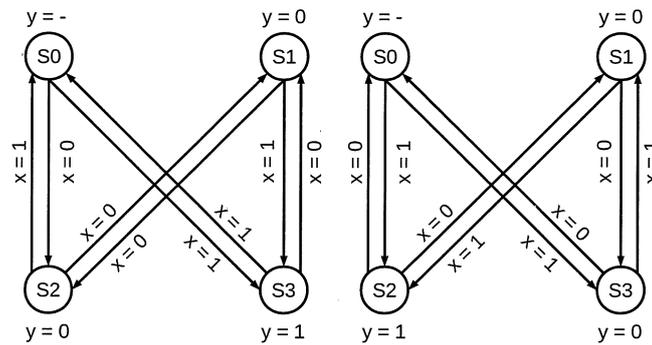
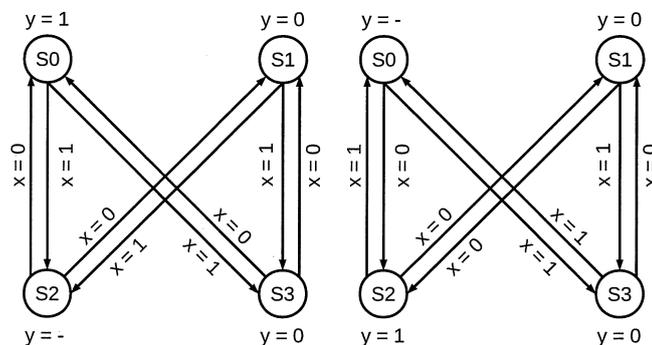


Abbildung 1: Mealy-Automat



(a)

(b)



(c)

(d)

Abbildung 2: Moore-Automaten

- d) Gegeben sei die Automatentafel in Tabelle 1.
Vervollständigen Sie die Tabelle für die Belegungen von Toggle-FlipFlops T_0 und T_1 zur Umsetzung der Zustandsspeicherung (q_0, q_1). (2 Punkte)

x	q_0	q_1	q'_0	q'_1	T_0	T_1	y
0	0	0	1	1			0
0	0	1	0	0			0
0	1	0	1	0			1
0	1	1	0	1			0
1	0	0	1	0			0
1	0	1	0	1			1
1	1	0	1	1			1
1	1	1	0	0			0

Tabelle 1: Automatentafel

- e) Bestimmen Sie eine DMF der Ansteuerfunktionen für T_0 und T_1 sowie der Ausgabefunktion y mit Hilfe der vorgegebenen Symmetriediagramme. (4 Punkte)

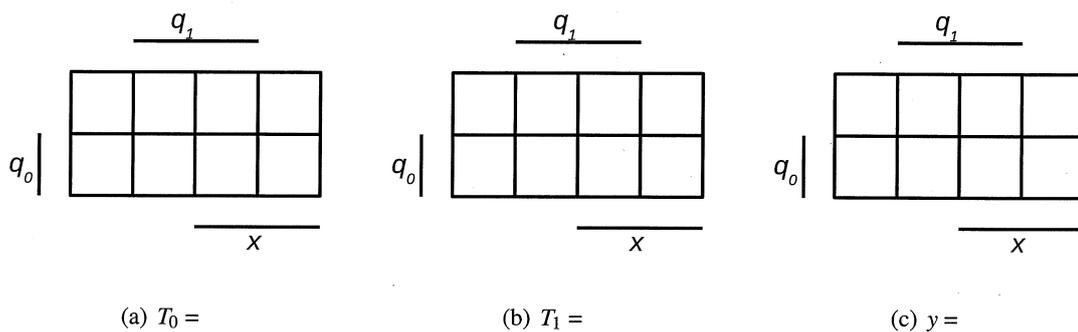


Abbildung 3: Symmetriediagramme

- f) Zeichnen Sie das vollständige Schaltwerk des in d) definierten Automaten unter Zuhilfenahme der Ergebnisse aus Aufgabe e). (4 Punkte)

- g) Wieviele Bits braucht man zur Kodierung von 10 Zuständen eines Mealy-Automaten, wenn die Zustände mithilfe eines Gray-Codes kodiert werden sollen? (1 Punkt)
- h) Wieviele Bits braucht man zur Kodierung von 12 Zuständen eines Moore-Automaten, wenn die Zustände mithilfe von One-Hot-Kodierung kodiert werden sollen? (1 Punkt)
- i) Bei welchen der in Abbildung 4 dargestellten Automaten handelt es sich um *deterministische Automaten* (DA). Geben Sie zudem eine kurze Begründung für ihre Antwort an, wenn es sich nicht um einen DA handelt. (4 Punkte)

- (a) DA / Kein DA , weil
- (b) DA / Kein DA , weil
- (c) DA / Kein DA , weil
- (d) DA / Kein DA , weil

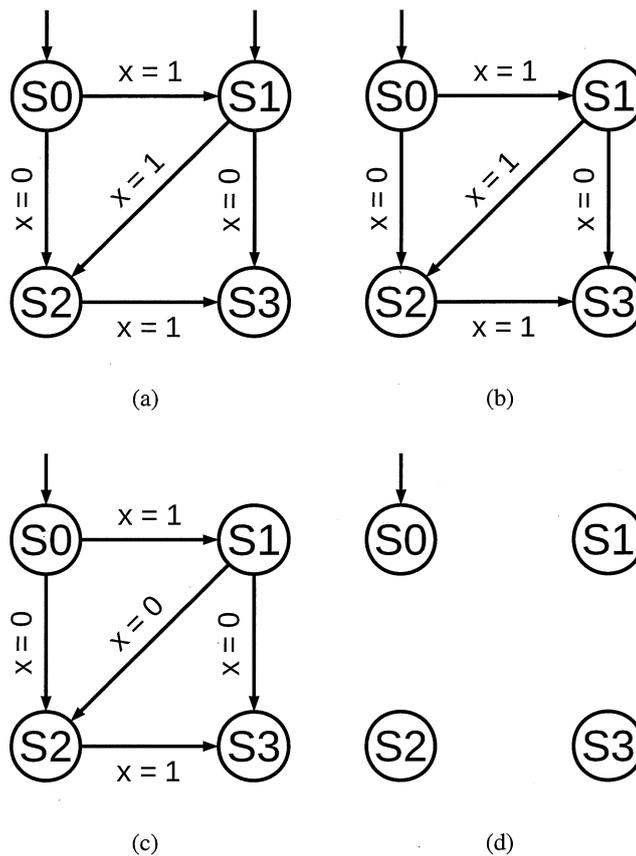


Abbildung 4: Automaten

Aufgabe 4 (Arithmetik)

(10 Punkte)

- a) Realisieren Sie einen Halbaddierer im NOR-Basissystem, der die Summe s und den Übertrag c aus den beiden Operanden a und b berechnet. Für die Implementierung der Schaltung stehen Ihnen nur NOR-Gatter mit zwei Eingängen zur Verfügung (**keine** Inverter!). Geben Sie zunächst die jeweiligen schaltalgebraischen Ausdrücke für die Schaltfunktionen $s(a, b)$ und $c(a, b)$ an und entwickeln Sie anhand dieser im Anschluss die zugehörige Halbaddiererschaltung. Geben Sie weiterhin die jeweilige Laufzeit für die Berechnung von c und s an, wobei die Verzögerung eines NOR-Gatters τ Nanosekunden beträgt. (6 Punkte)

- b) Geben Sie die Schaltung für einen Volladdierer im NOR-Basissystem an, der aus den Operanden a , b , und c_0 die Summe s und den Übertrag c berechnet. Verwenden Sie hierfür den in Teilaufgabe a) entwickelten Halbaddierer in Form des in Abb. 4 dargestellten Symbols. Geben Sie anhand Ihrer Implementierung aus Teilaufgabe a) die Laufzeitverzögerung für die Berechnung von s und c an. Geben Sie weiterhin die maximale Taktfrequenz für die entwickelte Schaltung an. (Hinweis: Sollten Sie in Teilaufgabe a) keine Lösung erhalten haben, verwenden Sie in diesem Fall für Ausgang $s = 103 \tau_{ns}$ und $c = 24 \tau_{ns}$.) (4 Punkte)

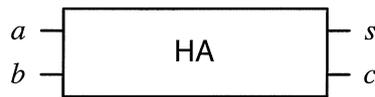


Abbildung 5: Symbol für einen Halbaddierer, mit a und b als Operanden, sowie Summe s und Übertrag c .

Aufgabe 5 (VHDL)

(20 Punkte)

Es soll ein digitaler Mittelwellenempfänger (siehe Abb. 6) implementiert werden. Durch Drücken der Tasten 'Frequenz +' bzw. 'Frequenz -' lässt sich die Trägerfrequenz des gewünschten Senders einstellen. Das Radio besitzt vier Speicherplätze für Sender. Durch Drücken einer Speicherplatztaste (1-4) lässt sich der entsprechende Sender aufrufen. Durch gleichzeitiges Drücken von 'Store' und einer Speicherplatztaste kann der aktuell eingestellte Sender in dem entsprechenden Speicherplatz abgespeichert werden. Zusätzlich besitzt das Radio einen Umschalter, mit dem man zwischen Radio und einer externen Audioquelle, die am 'Line In'-Eingang angeschlossen ist, umschalten kann. Des Weiteren existiert eine digitale Lautstärkesteuerung mit den Tasten 'Lautstärke +' und 'Lautstärke -'. **Wichtig:** Die Eingangssignale der Taster 'Frequenz +', 'Frequenz -', 'Lautstärke +' und 'Lautstärke -' sind bei Aktivierung nur für einen Takt auf '1' gesetzt. Der 'Line In'-Schalter ist ein Schalter. Die Eingangssignale der restlichen Taster sind bei Betätigung aktiv.

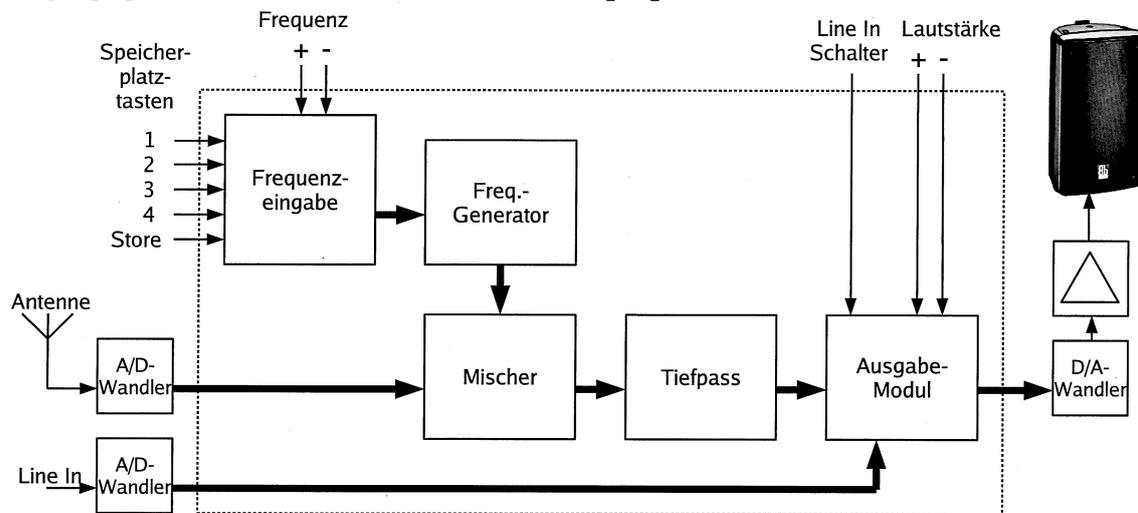


Abbildung 6: Architektur des Mittelwellenempfängers. Signale sind dünn, Signalvektoren fett gezeichnet.

Die zu implementierende synchron getaktete Schaltung besteht aus folgenden Schaltungsblöcken:

- **Frequenzeingabe:** Mit Hilfe des Moduls 'Frequenzeingabe' kann die Frequenz des aktuellen Radiosenders gesetzt werden. Dazu werden die oben beschriebenen Tasten (Frequenz- und Speicherplatztasten) benötigt. Die Ausgabe ist die eingestellten Frequenz, welche durch ein 8-Bit breites Datenwort repräsentiert wird. ("00000000" = 440 kHz und "11111111" = 2990 kHz).
- **Frequenzgenerator:** Der Frequenzgenerator erzeugt ein sinusförmiges 8-Bit Trägersignal mit der durch die Frequenzeingabe eingestellte Frequenz.
- **Mischer:** Der AM-Mischer mischt das hochfrequente Antennensignal in das Basisband, indem er das Antennensignal mit dem Trägersignal multipliziert. Das Ausgangssignal der Multiplikation ist 16 Bit breit, wobei nur die 8 höchstwertigen Bits an das Nachfolgemodul übergeben werden sollen.
- **Tiefpass:** Der Tiefpass filtert die durch die Mischung entstandenen hochfrequenten Störanteile aus dem Signal.
- **Ausgabemodul:** Im Ausgabemodul kann das Ausgabesignal durch Betätigung des 'Line In'-Schalters selektiert werden. Ist der 'Line In'-Schalters eingeschaltet (das entsprechende Si-

gnal ist auf '1'), soll das Signal der externen Audioquelle am 'Line In'-Eingang wiedergegeben werden, ansonsten das Radiosignal. Zusätzlich kann die Lautstärke durch Drücken der Lautstärketasten '+' und '-' um eine Lautstärkestufe erhöht bzw. verringert werden. Es sollen 256 verschiedene Lautstärkestufen implementiert werden, wobei beim Wert 128 die Ausgangsamplitude gleich der Eingangsamplitude sein soll. Das Ausgangssignal ist 8 Bit breit.

- a) Um wieviel kHz erhöht sich die Trägerfrequenz durch ein einmaliges Drücken der Taste 'Frequenz +'?
(1 Punkt)
- b) Geben Sie die Schnittstellenbeschreibung des Schaltungsblockes *Frequenzeingabe* in Form einer Entity in VHDL an.
(3 Punkte)

```
entity frequenzeingabe is
```

```
end entity;
```

- c) Geben Sie eine Implementierung des Schaltungsblockes *Frequenzeingabe* 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 fel of frequenzeingabe is
```

```
begin -- fel
```

```
end architecture;
```

- d) Geben Sie eine Implementierung des Schaltungsblockes *Mischer* in Form einer VHDL Architecture-Beschreibung an. VHDL und die Synthese unterstützen den Operator '*'. Schreiben Sie Eingangssignale groß und verwenden Sie hierzu das vorgegebene Code-Skelett (*Auf die Angabe von IEEE-Libraries kann verzichtet werden.*) (3 Punkte)

```
architecture m1 of mischer is
```

```
begin -- m1
```

```
end architecture;
```

- e) Geben Sie eine Implementierung des Schaltungsblockes *Ausgabemodul* in Form einer VHDL Architecture-Beschreibung an. VHDL und die Synthese unterstützen den Operator '*'. Schreiben Sie Eingangssignale groß und verwenden Sie hierzu das vorgegebene Code-Skelett (*Auf die Angabe von IEEE-Libraries kann verzichtet werden.*) (6 Punkte)

```
architecture aml of ausgabemodul is
```

```
begin -- aml
```

```
end architecture;
```