Klausur (WS 2018/2019)
Digitaltechnik

Institut für Technik der Informationsverarbeitung
Prof. Dr.-Ing. Dr. h. c. Jürgen Becker

Klausur: Digitaltechnik
Datum: 19. März 2019

Teilnehmer:
Matr.-Nr.:
ID:

Hörsaal:
Platz:

Es gelten die folgenden Regelungen:

- Die Bearbeitungszeit beträgt 120 Minuten.
- Es sind keine Hilfsmittel erlaubt, außer
  - einem doppelseitig und handschriftlich beschriebenen DIN-A4-Blatt.
- Nutzen Sie nur **dokumentenechte Schreibgeräte** – keine Bleistifte oder rote Farbe!
- Die Verwendung von eigenem Papier ist nicht zugelassen.
- Vermeiden Sie das Beschreiben der Rückseiten.
- Bei Bedarf erhalten Sie Zusatzblätter von der Aufsicht.
  - Versehen Sie solche Blätter unbedingt mit Ihrer Matrikelnummer.
  - Ordnen Sie jedes zusätzliche Lösungsblatt einer Aufgabe eindeutig zu.

Die vorliegende Klausur besteht aus **35 Blättern** und einer dreiseitigen Formelsammlung.

		Seite	≈ Pkt. in %	Punkte
Aufgabe 1:	Allgemeine Fragen	2	11	**28**
Aufgabe 2:	Automaten	5	12	**32**
Aufgabe 3:	Boolsche Algebra	9	12	**32**
Aufgabe 4:	CMOS und Gatter	14	14	**36**
Aufgabe 5:	Optimale Codes	21	12	**30**
Aufgabe 6:	Minimierung	25	12	**32**
Aufgabe 7:	Optimale Codes	28	12	**30**
Aufgabe 8:	Zahlensysteme	32	12	**30**
				$\sum$ **250**

Aufgabe 1: Allgemeine Fragen 28

Aufgabe 1.1: Codierung und Zahlensysteme

A) Wie viele Binärstellen sind mindestens nötig um 60 Zeichen zu codieren (ohne Fehlererkennung)? 1

ld(60) = 5.907 –> 6 Bit

B) Welche ist die größte darstellbare Zahl bei Verwendung der 2er-Komplementdarstellung mit 5 bit? 1

$2^{5-1} - 1 = 15$

C) Wie viele Symbole werden im Tetrahexagesimalsystem (Basis 64) benötigt, um eine 40-stellige Binärzahl darzustellen? Begründen Sie Ihre Antwort. 3

Basis 64 –> ld(64) = 6 Bit –> mit einem Symbol werden 6 Bit dargestellt
40 / 6 = 6,6 –> 7 Symbole

D) Welche minimale Hammingdistanz weißt der Stibitz-Code auf? Begründen Sie Ihre Antwort durch ein Beispiel. 2

HDmin = 1; z.B. 1 = 0100 und 2 = 0101 –> 1 Bit Änderung –> HDmin = 1

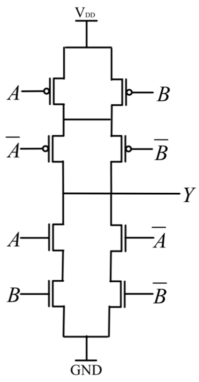
E) Wie viele gültige Codewörter gibt es bei einem 7 aus 8 Code? Geben Sie zwei gültige Codewörter explizit an. 2

es gibt genau 8 gültige Codewörter; z.B. 0111 1111 und 1011 1111

F) Geben Sie jeweils die DMF der Funktionen $a \equiv b$ sowie $a \oplus b$ an. 2

$a \equiv b = ab + \bar{a}\bar{b}$
$a \oplus b = \bar{a}b + a\bar{b}$

G) In Abbildung 1.1 ist ein CMOS-Schaltkreis und eine dazugehörige unvollständige Funktionstabelle gegeben. Vervollständigen Sie die Funktionstabelle und benennen Sie die realisierte Schaltfunktion. **3**



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Abbildung 1.1: CMOS Schaltkreis und Funktionstabelle

Exklusivoder / Antivalenz

Aufgabe 1.2: Blocksicherung

Bei einer Datenübertragung wird der Bitstrom 0111 0100 0110 000 empfangen. Über die Codierung wissen Sie lediglich, dass Blocksicherung mit doppelter Quersummenergänzung ohne Scrambling verwendet wurde und die Übertragung fehlerfrei ist.

A) Geben Sie die Nutzdatenwörter aus dem empfangenen Bitstrom in der richtigen Reihenfolge an. Begründen Sie Ihre Antwort! **4**

Es werden 15 Bit Übertragen. Demnach gibt es genau zwei Möglichkeiten, wie der Block in Zeilen und Spalten zusammengestzt sein könnte 3x5 oder 5x3, d.h.:

1. Möglichkeit	oder	2. Möglichkeit
01110		011
10001		101
10000		000
		110
		000

In Möglichkeit eins gibt es einen Paritätsfehler, während Möglichkeit zwei fehlerfrei ist

–> verwendete Nutzdaten sind: 01, 10, 00, 11

B) Angenommen, Sie wenden auf die oben genannte Übertragung ein Scrambling an, wie lautet dann der resultierende modifizierte Bitstrom? **1**

01010 10010 11000
alternativ: 011 100 100 100 010

C) Welchen Vorteil hat Scrambling bei der Blocksicherung? **1**

Es können Bündelfehler erkannt und ggf. korrigiert werden

Aufgabe 1.3: Digitalisierung

A) Geben Sie vier unterschiedliche Signaltypen an. **4**

kontinuierliches Signal: analoges signal, physikalisches Signal
zeitdiskretes, wertkontinuierliches Signal: diskrete Abtastzeitpunkte
zeitkontinuierliches , wertdiskretes Signal: diskrete Werte
zeitdiskretes, wertdiskretes Signal: digitales Signal:

B) Erklären Sie die “weiche” Diskrimination. **2**

Einfügenen einnes undefinierten Bereiches zwischen Intervallübergänge.

C) Wie viele Bits werden in einem Speicher mindestens benötigt um 800 Abtastpunkte eines Signals mit vier Quantisierungsstufen zu speichern? Begründen Sie Ihre Antwort. **2**

4 Wertebereiche –> 2 Bit pro Abtastzeitpunkt –> 1.600 Bit

Aufgabe 2: Automaten

32

Aufgabe 2.1: Automatendarstellung

Ein allgemeiner Moore-Automat ist anhand des Graphen in Abbildung 2.1 dargestellt.

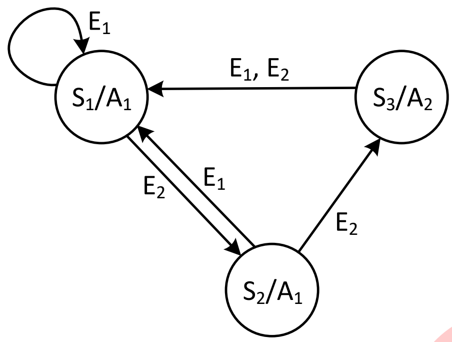


Abbildung 2.1: Allgemeiner Moore-Automat

A) Eine Methode zur Darstellung des Verhaltens eines Automaten ist die Automatentafel. Wie wird sie gebildet? Erklären Sie dabei den Unterschied in der Darstellung zwischen Mealy- und Moore-Automaten.

4

Die Automatentafel entsteht durch Bilden des kartesischen Produktes aus Eingabe- und Zustandsmenge. **Alternativ:** Links vertikal (1.Spalte): Zustände / Oben horizontal (1.Zeile): Folgezustände und darunter Eingaben. Kreuzungsstellen:

- Mealy: Folgezustand und Ausgabe
- Moore: nur Folgezustand

B) Füllen Sie die Automatentafel in Tabelle 2.1 entsprechend dem Automaten von Abbildung 2.1 aus. Achten Sie dabei auf eine richtige Beschriftung der Spalten.

4

S^ν	$S^{\nu+1}$		A^ν
	E_1	E_2	
S_1	S_1	S_2	A_1
S_2	S_1	S_3	A_1
S_3	S_1	S_1	A_2

Tabelle 2.1: Automatentafel vom Moore-Automaten in Abb. 2.1

Aufgabe 2.2: Automatenanalyse

Gegeben sei folgendes Ablaufdiagramm eines Automaten (Abbildung 2.2).

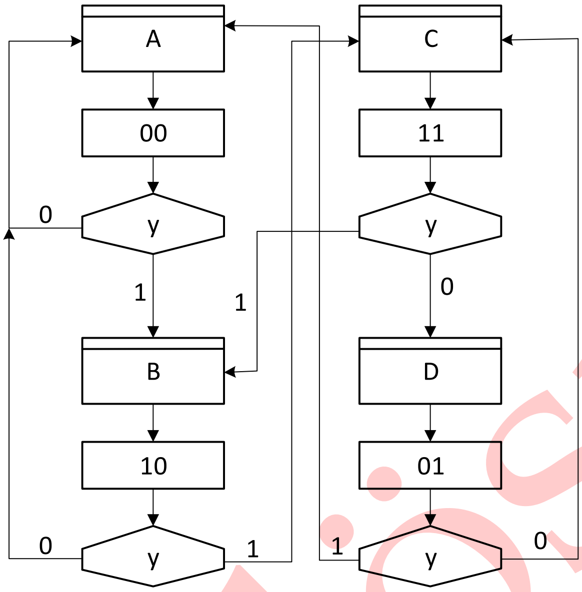


Abbildung 2.2: Ablaufdiagramm eines Automaten

A) Füllen Sie nun die Ablauftabelle (Tabelle 2.2) aus. Tragen Sie hierzu für jeden Zustand und jede mögliche Eingabe den codierten Folgezustand sowie die Ausgabe ein. Verwenden Sie für die Codierung des Folgezustandes (alphabetisch aufsteigend: A, B, etc.) den Gray-Code mit dem initialen Codewort 00. **6**

Zustand S^v	**Eingabe** $E^v = y$	**Folgezustand (codiert)** S^{v+1}	**Ausgabe** A
A	0	00 (A)	00
	1	01 (B)	00
B	0	00 (A)	10
	1	11 (C)	10
C	0	10 (D)	11
	1	01 (B)	11
D	0	11 (C)	01
	1	00 (A)	01

Tabelle 2.2: Ablauftabelle des Automaten in Abb. 2.2

B) Welchen Typ hat der Automat von Abbildung 2.2? Begründen Sie Ihre Antwort. **2**

Es handelt sich um einen Moore-Automaten, weil die Ausgabe nur vom Zustand abhängt.

Aufgabe 2.3: Realisierung von Automaten mit FlipFlops

A) Der Zustandsautomat aus Tabelle 2.3 soll mit einem *RS-FlipFlop* (mit den Eingängen r_0 und s_0) für das erste Bit S_0 und einem *JK-FlipFlop* (mit den Eingängen j_1 und k_1) für das zweite Bit S_1 realisiert werden. Ergänzen Sie in der Ablauftabelle die fehlenden Ansteuerbits für die Eingänge s_0 und j_1 der FlipFlops. Verwenden Sie nach Möglichkeit "don't care" Stellen. **8**

Zustand	**Eingabe**	**Folgezustand**	**FlipFlop Ansteuerung**			
$S^v = (S_0^v, S_1^v)$	$E^v = (E_0^v, E_1^v)$	S^{v+1}	s_0	r_0	j_1	k_1
0,0	0,0	0,0	0	-	0	-
	0,1	1,1	1	0	1	-
	1,0	1,1	1	0	1	-
	1,1	0,1	0	-	1	-
0,1	0,0	0,0	0	-	-	1
	0,1	0,1	0	-	-	0
	1,0	0,1	0	-	-	0
	1,1	1,0	1	0	-	1
1,0	0,0	0,0	0	1	0	-
	0,1	0,1	0	1	1	-
	1,0	0,1	0	1	1	-
	1,1	0,0	0	1	0	-
1,1	0,0	1,0	-	0	-	1
	0,1	0,0	0	1	-	1
	1,0	0,1	0	1	-	0
	1,1	1,1	-	0	-	0

Tabelle 2.3: Ablauftabelle eines unbekannten Zustandsautomaten

B) Die Ansteuerfunktionen für die FlipFlops sollen nun minimiert werden. Geben Sie mit Hilfe der in Abbildung 2.3 vorgegebenen Symmetriediagramme jeweils eine disjunktive minimale Ansteuerfunktion für die beiden Eingänge r_0 und k_1 an. 8

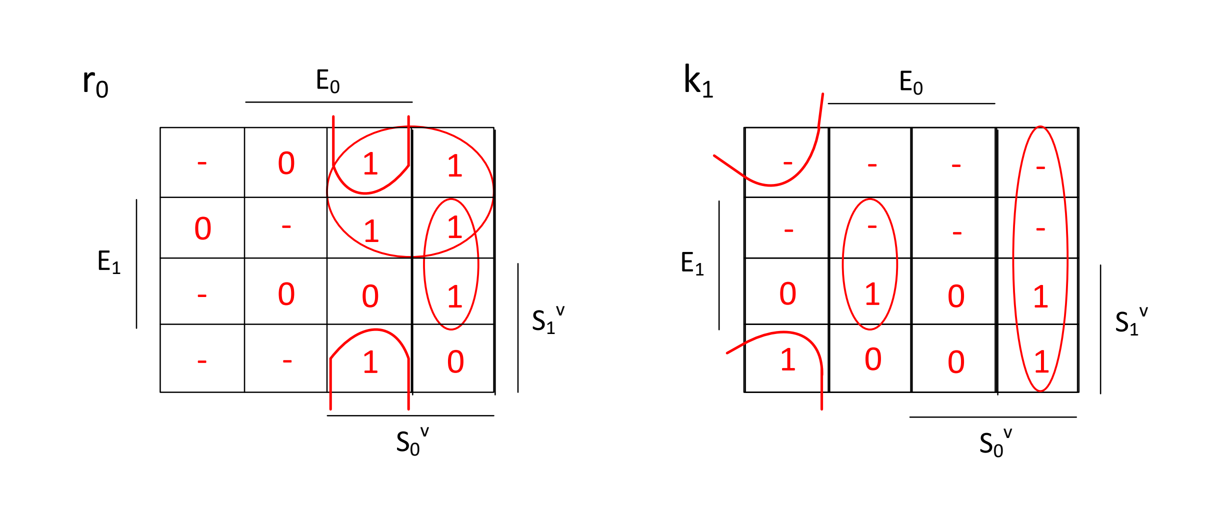


Abbildung 2.3: Symmetriediagramme für Ansteuerfunktionen

$r_0 = (S_0^v \wedge \overline{S_1^v}) \vee (S_0^v \wedge \overline{E_0} \wedge E_1) \vee (S_0^v \wedge E_0 \wedge \overline{E_1})$

$k_1 = (S_0^v \wedge \overline{E_0}) \vee (\overline{S_0^v} \wedge E_0 \wedge E_1) \vee (\overline{E_0} \wedge \overline{E_1})$

Aufgabe 3: Boolsche Algebra

32

Aufgabe 3.1: Axiome und Regeln der Boolschen Algebra

A) Nennen Sie *zwei* grundlegende Forderungen, die ein Axiomensystem (z. B. die 5 Huntingtonschen Axiome) erfüllen muss. 2

Widerspruchsfrei (logisch), vollständig, Axiome unabhängig

B) Gegeben sei die Boolsche Algebra $BA = [K, \top, \perp, \bar{\ }, O, I]$ nach Huntington. Ferner sei M eine Menge mit beliebig vielen Elementen und MA eine Mengenalgebra über M. Wie lässt sich die Mengenalgebra MA auf die gegebene BA abbilden? Vervollständigen Sie dazu die untenstehende Tabelle. 2

BA:	K	$\top$	$\perp$	$\bar{\ }$	O	I
MA:	$P(M)$	$\cap$	$\cup$	C_M	$\emptyset$	M

C) Geben Sie den *dualen* Satz zu der Regel $a \vee 0 = a$ an. 2

$a \wedge 1 = a$

Aufgabe 3.2: Beweis eines Satzes mittels Boolscher Algebra

Im folgenden soll die Gültigkeit des Satzes

$$x \cdot (y \oplus z) = (x \cdot y) \oplus (x \cdot z) \tag{3.1}$$

mittels der Huntingtonschen Axiome (H1 - H5) und der Regeln (R1 - R12) aus dem Formelblatt bewiesen werden. Zusätzlich zu den genannten Regeln darf die Definition der XOR Beziehung $x \oplus y = \overline{x}y \vee x\overline{y}$ verwendet werden. Die übrigen XOR Regeln aus der Formelsammlung dürfen *nicht* verwendet werden. Für den Nachweis sollen beide Seiten von Gleichung (3.1) auf den selben Ausdruck zurückgeführt werden.

A) Führen Sie den Beweis für Satz (3.1) indem Sie die Einzelschritte für jede Seite der Gleichung zeilenweise in die zugehörige Tabelle (siehe unten) eintragen. Geben Sie jeweils die Nummer des verwendeten Satzes gemäß Formelblatt an (oder „XOR“ für die XOR Definition). In jedem Einzelschritt darf *genau eine* Regel/Axiom auf beliebig viele Teilausdrücke angewendet werden. (*Hinweis:* Die Verwendung des Kommutativgesetzes H2 sowie das Weglassen unnötiger Klammern muss nicht explizit angegeben werden.) **10**

Linke Seite

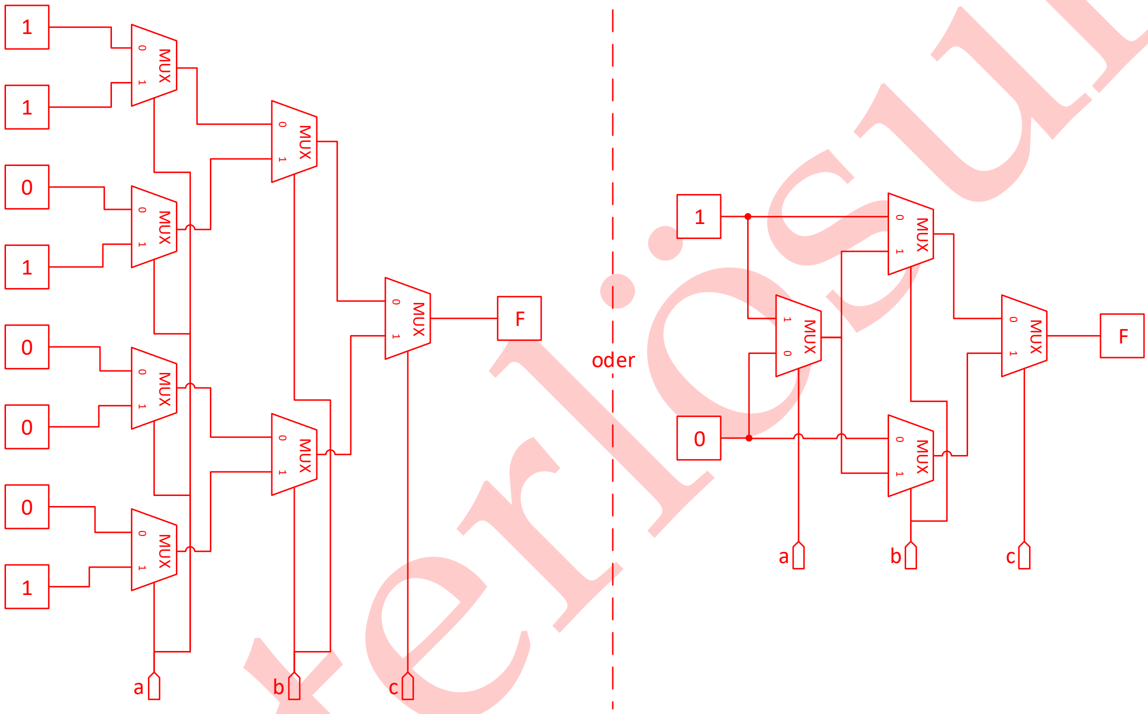
Regel	Term
-	$x \cdot (y \oplus z)$
XOR	$x \cdot (\overline{y}z \vee y\overline{z})$
H3	$x\overline{y}z \vee xy\overline{z}$

Rechte Seite

Regel	Term
-	$(x \cdot y) \oplus (x \cdot z)$
XOR	$\overline{(xy)} \cdot x \cdot z \lor x \cdot y \cdot \overline{(xz)}$
R12b	$(\overline{x} \lor \overline{y}) \cdot x \cdot z \lor x \cdot y \cdot (\overline{x} \lor \overline{z})$
H3	$\overline{x}xz \lor x\overline{y}z \lor xy\overline{x} \lor xy\overline{z}$
R8b	$0z \lor x\overline{y}z \lor 0y \lor xy\overline{z}$
R5b	$0 \lor x\overline{y}z \lor 0 \lor xy\overline{z}$
R5a	$x\overline{y}z \lor xy\overline{z}$ entspricht linker Seite ■

Aufgabe 3.3: Schaltfunktionen und Entwicklungssatz

A) Gegeben sei die Schaltfunktion $F(c, b, a) = (a \vee \overline{b} \vee c) \wedge (a \vee b \vee \overline{c}) \wedge (a \vee \overline{b} \vee \overline{c}) \wedge (\overline{a} \vee b \vee \overline{c})$ in konjunktiver Normalform (KNF). Vervollständigen Sie die unten abgebildete 2-zu-1 Multiplexer-Schaltung, um die Funktion zu realisieren. (*Hinweis:* Es ist nicht erforderlich die Schaltung bzw. die Schaltfunktion zu minimieren oder zu entwickeln.) **4**



B) Was besagt der Hauptsatz der Schaltalgebra? **2**

Jede Schaltfunktion lässt sich als Disjunktion von Mintermen oder als Konjunktion von Maxtermen *eindeutig* darstellen.

C) Wozu wird der Entwicklungssatz der Schaltalgebra typischerweise verwendet? **2**

Umwandlung eines beliebigen schaltalgebraischen Ausdrucks in eine der beiden Normalformen (KNF bzw. DNF)

D) Gegeben sei die Logikfunktion $G(c,b,a) = \bar{a}(b \oplus (c \lor a))$. Für diese Funktion soll nun die *Disjunktive Normalform (DNF)* mit Hilfe des Entwicklungssatzes der Schaltalgebra bestimmt werden. Entwickeln Sie die Funktion dazu sukzessive nach den einzelnen Variablen und vervollständigen Sie die nachfolgende Tabelle mit den Restfunktionen. Ermitteln Sie anschließend die DNF anhand der Ergebnisse.

8

$G_{DNF}(c,b,a) = c\bar{b}\bar{a} \lor \bar{c}b\bar{a}$

$G_{DNF}(c,b,a) = [\,\bar{c} \land G(0,b,a)\,] \lor [\,c \land G(1,b,a)\,]$ mit	
$G(0,b,a) =$ $\bar{a}(b \oplus a)$	$G(1,b,a) =$ $\bar{a}(b \oplus 1) = \bar{a}\bar{b}$
$G(0,b,a) = [\bar{b} \land G(0,0,a)] \lor [b \land G(0,1,a)]$	$G(1,b,a) = [\bar{b} \land G(1,0,a)] \lor [b \land G(0,1,a)]$
$G(0,0,a) =$ 0	$G(1,0,a) =$ $\bar{a}$
$G(0,1,a) =$ $\bar{a}$	$G(1,1,a) =$ 0
$G(0,0,a) = [\bar{a} \land G(0,0,0)] \lor [a \land G(0,0,1)]$	$G(1,0,a) = [\bar{a} \land G(1,0,0)] \lor [a \land G(1,0,1)]$
$G(0,0,0) =$ 0	$G(1,0,0) =$ 1
$G(0,0,1) =$ 0	$G(1,0,1) =$ 0
$G(0,1,a) = [\bar{a} \land G(0,1,0)] \lor [a \land G(0,1,1)]$	$G(1,1,a) = [\bar{a} \land G(1,1,0)] \lor [a \land G(1,1,1)]$
$G(0,1,0) =$ 1	$G(1,1,0) =$ 0
$G(0,1,1) =$ 0	$G(1,1,1) =$ 0

Aufgabe 4: CMOS und Gatter

36

Aufgabe 4.1: Allgemeine Fragen

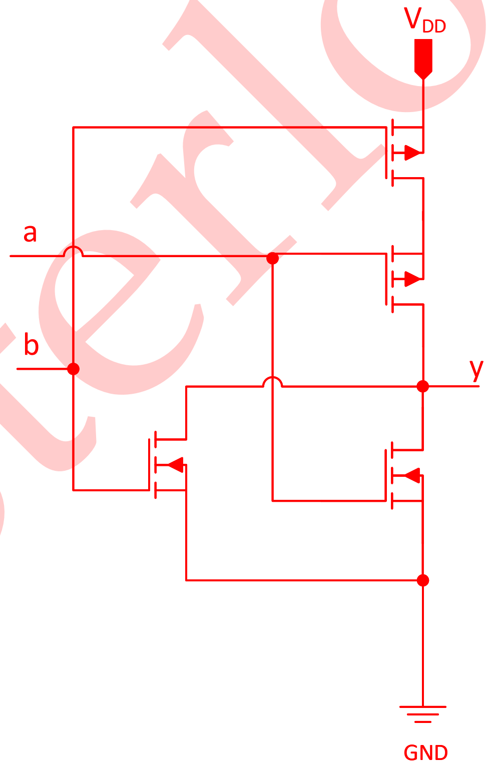
A) Nennen Sie den ausgeschriebenen Begriff zum Buchstaben „C“ im Akronym CMOS und erklären Sie die Bedeutung dieses Begriffs im Kontext der CMOS-Schaltungen.

1

C → Complementary
Die Pullup/Pulldown Netze, beziehungsweise die Transistoren (NMOS/PMOS), die diese Netze realisieren sind komplementär.

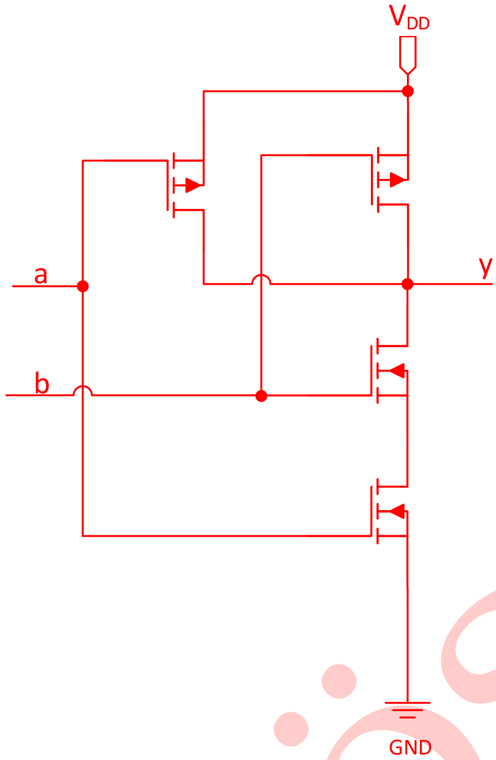
B) Zeichnen Sie eine NOR-Schaltung mit den zwei Eingängen a und b und dem Ausgang y in CMOS Technologie. Verwenden Sie **positive Logik**, d.h. der CMOS-Pegel V_{DD} entspricht einer logischen ‘1’.

2



C) Zeichnen Sie erneut eine NOR-Schaltung mit den zwei Eingängen a und b und dem Ausgang y in CMOS Technologie. Verwenden Sie nun **negative Logik**, d.h. der CMOS-Pegel V_{DD} entspricht einer logischen ‘0’.

2



V_{DD}
a
b
y
GND

Aufgabe 4.2: Schaltungsanalyse

Hinweis: Verwenden Sie für die folgenden Aufgaben eine **positive Logik**, d.h. der CMOS-Pegel V_{DD} entspricht einer logischen '1'.

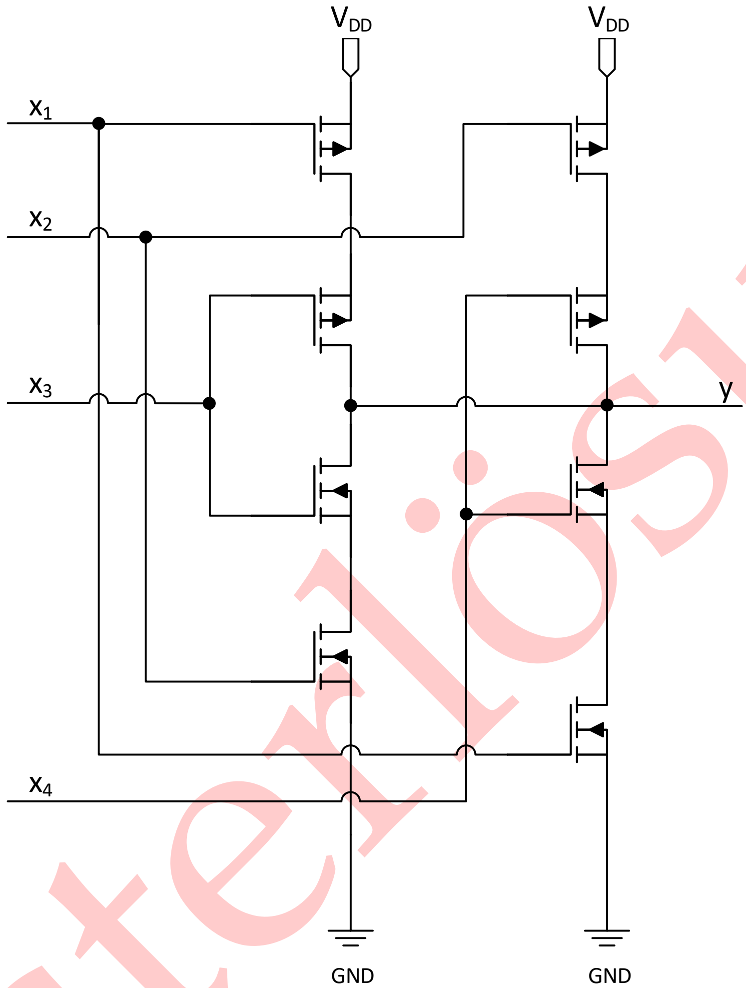


Abbildung 4.1: Schaltung 1

Betrachten Sie im Folgenden Schaltung 1 in Abbildung 4.1. Die Funktion $F = \overline{x_1}\,\overline{x_3} \vee \overline{x_2}\,\overline{x_4}$ für das Pullup- und die Funktion $G = x_2x_3 \vee x_1x_4$ für das Pulldown- Netz sind gegeben.

A) Prüfen Sie algebraisch, ob Schaltung 1 wohldefiniert ist. **3**

$F \stackrel{?}{=} \overline{G}$

$\overline{G} = \overline{x_2 x_3} \wedge \overline{x_1 x_4} = (\overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_4})$

$= \overline{x_2}\,\overline{x_1} \vee \overline{x_2}\,\overline{x_4} \vee \overline{x_3}\,\overline{x_1} \vee \overline{x_3}\,\overline{x_4} \neq F$

$\overline{F} \stackrel{?}{=} G$

$\overline{F} = \overline{\overline{x_1}\,\overline{x_3}} \wedge \overline{\overline{x_2}\,\overline{x_4}} = (x_1 \vee x_3) \wedge (x_2 \vee x_4)$

$= x_1\,x_2 \vee x_1\,x_4 \vee x_3\,x_2 \vee x_3\,x_4 \neq G$

Die Schaltung ist nicht wohldefiniert.

B) Prüfen Sie algebraisch, ob Schaltung 1 kurzschlussfrei ist. **2**

$F \wedge G \stackrel{?}{=} 0$

$F \wedge G = (\overline{x_1}\,\overline{x_3} \vee \overline{x_2}\,\overline{x_4}) \wedge (x_2 x_3 \vee x_1 x_4)$

$= 0$

Die Schaltung ist kurzschlussfrei.

C) Vervollständigen Sie die Funktionstabelle für die Pullup-Funktion F und die Pulldown-Funktion G und markieren Sie alle Kurschlüsse. Kennzeichnen sie weiterhin Eingangskombinationen, die dazu führen, dass die Schaltung Hochohmig wird. Markieren Sie dazu in den Zeilen der entsprechenden Eingangskombinationen die entsprechende Spalte mit einem x. **4**

x_1	x_2	x_3	x_4	F	G	Kurzschluss	Hochohmig
0	0	0	0	1	0		
0	0	0	1	1	0		
0	0	1	0	1	0		
0	0	1	1	0	0		x
0	1	0	0	1	0		
0	1	0	1	1	0		
0	1	1	0	0	1		
0	1	1	1	0	1		
1	0	0	0	1	0		
1	0	0	1	0	1		
1	0	1	0	1	0		
1	0	1	1	0	1		
1	1	0	0	0	0		x
1	1	0	1	0	1		
1	1	1	0	0	1		
1	1	1	1	0	1		

In Schaltung 2 (Abbildung 4.2) wird Schaltung 1 (Abbildung 4.1) mit zwei vorgeschalteten Invertern erweitert.

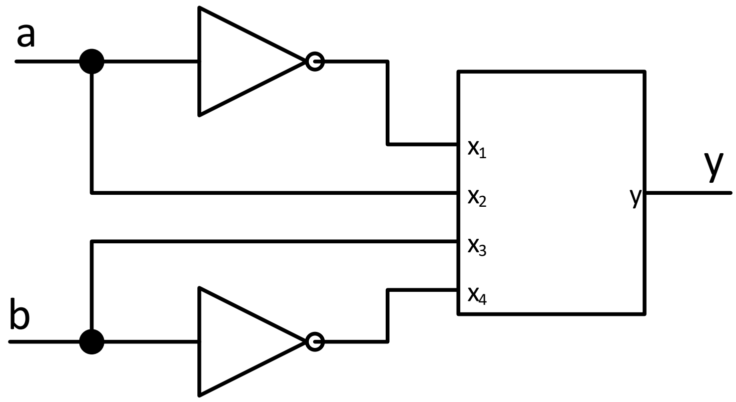


Abbildung 4.2: Schaltung 2

D) Setzen Sie nun a und b in die Funktionen F und G ein, sodass die entstehenden Gleichungen F' und G' Schaltung 2 (Abbildung 4.2) beschreiben. Geben Sie dann die Funktionstabelle zu dieser Schaltung an.

3

$F' = a\,\overline{b} \lor \overline{a}\,b$
$G' = a\,b \lor \overline{a}\,\overline{b}$

a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

E) Um welche aus der Vorlesung bekannte Funktion handelt es sich?

1

exklusives oder (XOR)

F) Gibt es für Schaltung 2 (Abbildung 4.2) Eingangskombinationen, die einen Kurzschluss oder einen Hochohmig Zustand am Ausgang y der Schaltung verursachen? Beachten Sie, dass Schaltung 1 (Abbildung 4.1) als Teilschaltung verwendet wird und begründen Sie Ihre Antwort. **3**

Die Gesamtschaltung ist für alle Eingangskombinationen definiert (siehe Tabelle in C). Die Kurzschlussfreiheit folgt aus der Kurzschlussfreiheit aller Teilschaltungen. Die Eingangkombinationen, für die die Teilschaltung nicht definiert ist (siehe C) werden durch die Verschaltung der vorgeschalteten Inverter vermieden, sie treten nie auf. (Wir ignorieren hier wie immer Gatterlaufzeiten & dadurch verursachte Glitches).

Aufgabe 4.3: Schaltungssynthese

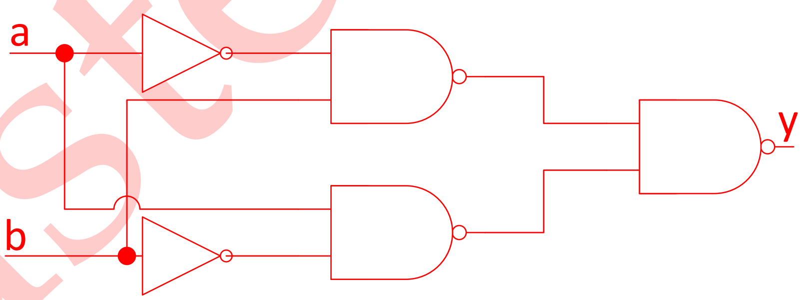
Hinweis: Verwenden Sie für die folgenden Aufgaben eine **positive Logik**, d.h. der CMOS-Pegel V_{DD} entspricht einer logischen '1'.

Im Folgenden soll die Funktion $y = \bar{a}b \vee a\bar{b}$ in CMOS Technologie realisiert werden.

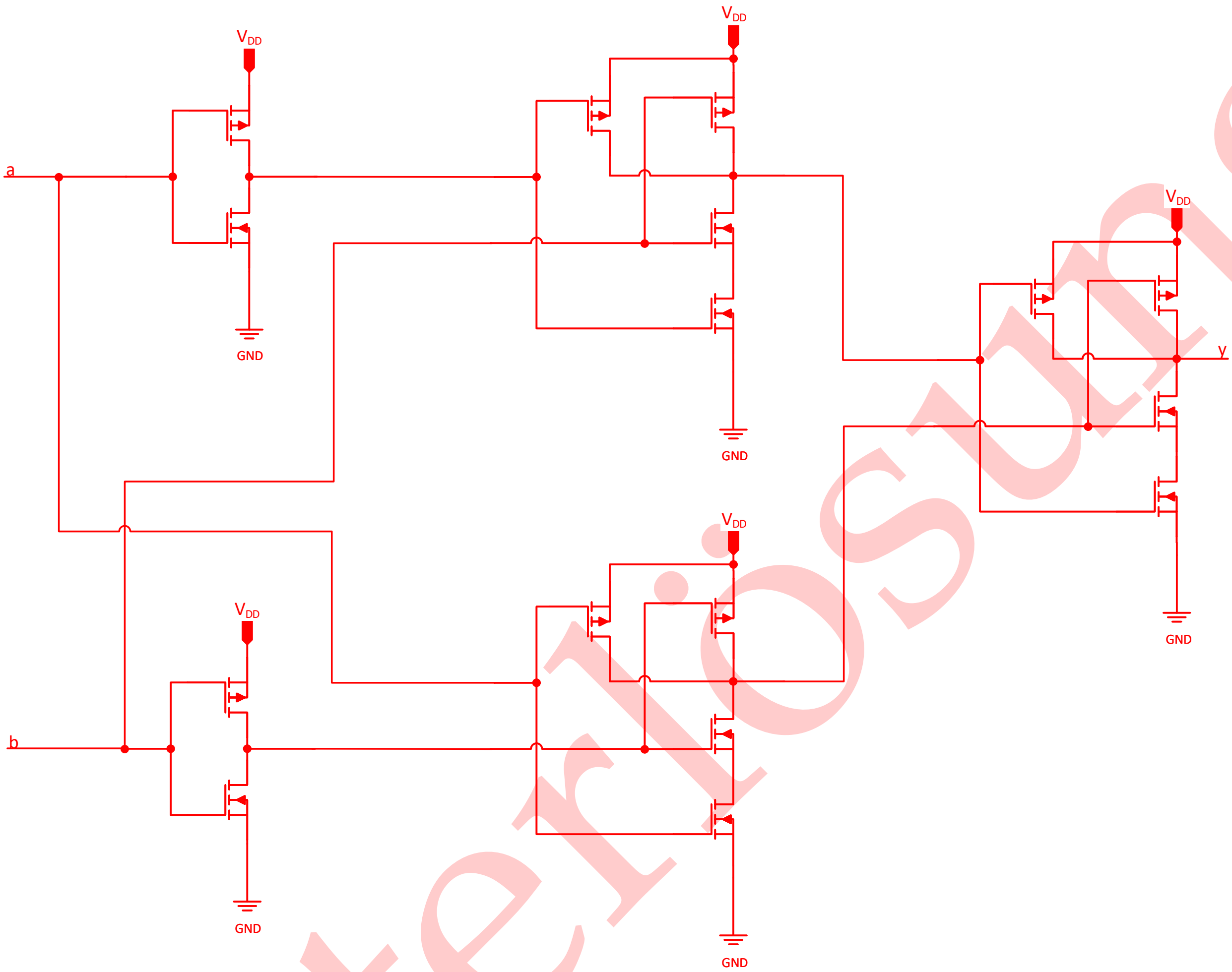
A) Stellen Sie die Funktion ausschließlich mit NAND und NOT Operationen dar. **2**

$y = \overline{\overline{\bar{a}\, b} \wedge \overline{a\, \bar{b}}}$

B) Zeichnen Sie den Schaltplan zu Ihrer Lösung aus A). Verwenden Sie ausschließlich NAND Gatter und Inverter. **4**



C) Zeichnen Sie nun die Schaltung aus B) als CMOS-Schaltung aus PMOS und NMOS Transistoren. **8**



D) Nennen Sie einen Vorteil von Schaltung 2 aus Abbildung 4.2 im Vergleich zu Ihrer Schaltung aus C). **1**

Weniger Transistoren nötig
1 Stufe weniger, d.h. in der Regel kürzere Laufzeit

Aufgabe 5: Optimale Codes

30

Aufgabe 5.1: Shannon-Fano-Verfahren

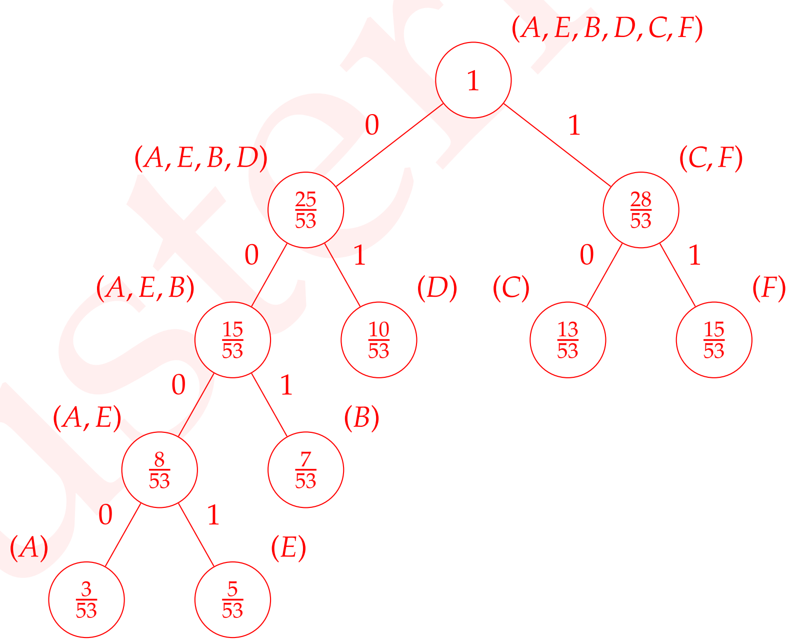
Gegeben sei eine gedächtnislose Quelle S. Gemäß der in Tabelle 7.1 dargestellten Auftrittswahrscheinlichkeiten erzeugt sie Symbole aus dem Zeichenvorrat $Y = \{A, B, C, D, E, F\}$.

Symbol	A	B	C	D	E	F
Wahrscheinlichkeit	$\frac{3}{53}$	$\frac{7}{53}$	$\frac{13}{53}$	$\frac{10}{53}$	$\frac{5}{53}$	$\frac{15}{53}$
Codewort	0000	001	10	01	0001	11

Tabelle 5.1: Auftrittswahrscheinlichkeiten und Codewörter der von S erzeugten Symbole

A) Wenden Sie das Shannon-Fano-Verfahren an, um die von S erzeugten Symbole zu codieren. Sortieren Sie die Symbole dafür mit aufsteigenden Auftrittswahrscheinlichkeiten von links nach rechts und erzeugen Sie den vollständigen Codierbaum. Weisen Sie den linken Ästen des entstehenden Baums eine „0“ und den rechten Ästen des Baums eine „1“ zu. Tragen Sie die Codewörter, die sich ergeben, in die dafür vorgesehenen Felder in Tabelle 7.1 ein.

7



B) Codieren Sie das Wort „*CBCAE*“ unter Verwendung der in A) erzeugten Codierung. **2**

Mit der ausgefüllten Tabelle folgt „10 001 10 0000 0001“ als Codierung.

C) Wie lautet die allgemeine Formel zur Berechnung der mittleren Codewortlänge eines Codes über dem N-elementigen Zeichenvorrat $X = \{x_1, \ldots, x_N\}$? Nehmen Sie an, dass $p(x_i)$ die Auftrittswahrscheinlichkeit und $m(x_i)$ die Codewortlänge eines $x_i \in X$ bezeichnet. **1**

$\bar{m} = \sum_{i=1}^{N} m(x_i)p(x_i)$

D) Bestimmen Sie die mittlere Codewortlänge der in A) erzeugten Codierung und geben Sie diese als vollständig gekürzter Bruch an. **3**

$\bar{m} = \left(\frac{3}{53} \cdot 4 + \frac{7}{53} \cdot 3 + \frac{13}{53} \cdot 2 + \frac{10}{53} \cdot 2 + \frac{5}{53} \cdot 4 + \frac{15}{53} \cdot 2\right) \text{ Bit} = \frac{129}{53} \text{ Bit}$

E) Welche mittlere Codewortlänge ergibt sich, wenn alle Symbole, die von S erzeugt werden, binär und mit Codewörtern gleicher Länge codiert werden sollen? **2**

$\bar{m} = \lceil \log_2 6 \rceil \text{ Bit} = 3 \text{ Bit}$

Aufgabe 5.2: Huffman-Verfahren

A) Welches informationstheoretische Maß dient bei der Codierung einer Quelle als größte untere Schranke der mittleren Codewortlänge $\bar{m}$? Geben Sie dessen Bezeichnung und eine allgemeingültige Formel zur Berechnung an. Benennen Sie dabei alle als Teil der Berechnungsvorschrift verwendeten Größen. **2**

Das theoretische Minimum für die minimale Codewortlänge ist gegeben durch die Entropie, d. h. dem mittleren Informationsgehalt, der Quelle. Für eine Quelle, die ein Symbol $x \in X$ aus dem N-elementigen Zeichenvorrat $X = \{x_1, \ldots, x_N\}$ mit Wahrscheinlichkeit $p(x)$ erzeugt, ist sie gegeben durch $H = -\sum_{i=1}^{N} p(x_i) \log_2 (x_i)$.

Betrachtet wird nun die Mensch-Maschine-Schnittstelle eines eingebetteten Systems. Sie erlaubt es dem Nutzer, die Konfiguration des Systems zu modifizieren. Um die Benutzereingabe entgegenzunehmen, kommen fünf Tasten zur Anwendung. Gemäß ihrer Funktionalität werden die Tasten im Folgenden als `RESET`, `INC`, `DEC`, `PREV` und `NEXT` bezeichnet.

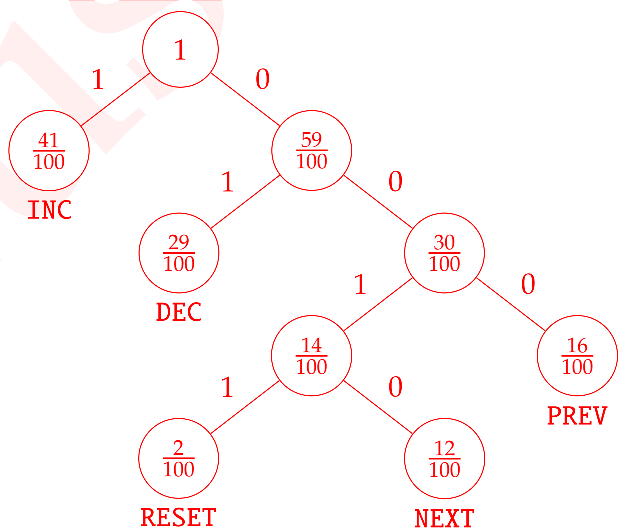
Tastendrücke, die über die Lebensdauer des Systems hinweg getätigt werden, sollen zu Diagnosezwecken in einem nichtflüchtigen Speicher protokolliert werden. Um dies effizient zu realisieren, soll eine Huffman-Codierung der Tastendruckereignisse erfolgen.

Taste	`RESET`	`INC`	`DEC`	`PREV`	`NEXT`
Häufigkeit	2	41	29	16	12
Wahrscheinlichkeit	0,02	0,41	0,29	0,16	0,12
Codewort	0011	1	01	000	0010

Tabelle 5.2: Häufigkeiten, Wahrscheinlichkeiten und Codewörter der Tastendrücke

B) Bei einem Testlauf haben sich die in Tabelle 7.2 dargestellten Tastendruckhäufigkeiten ergeben. Nehmen Sie an, dass diese Häufigkeiten repräsentativ für die Auftrittswahrscheinlichkeiten der einzelnen Tastendrücke sind. Bestimmen Sie die Wahrscheinlichkeiten und tragen Sie diese in dezimaler Form oder als vollständig gekürzter Bruch in Tabelle 7.2 ein. **1**

C) Erzeugen Sie auf Basis der in B) bestimmten Wahrscheinlichkeiten eine Huffman-Codierung der Tastendruckereignisse. Ordnen Sie den Knoten mit der niedrigeren Auftrittswahrscheinlichkeit dabei immer als linkes Kind an und konstruieren Sie den vollständigen Codierbaum. Weisen Sie den linken Ästen des Baums eine „1" und den rechten Ästen des Baums eine „0" zu. Beschriften Sie die Blätter des Baums mit den dazugehörigen Tastendruckereignissen. Tragen Sie die Codewörter, die sich ergeben, in Tabelle 7.2 ein. **7**



D) Bestimmen Sie die mittlere Codewortlänge der im Zuge von C) erzeugten Codierung und geben Sie diese in dezimaler Form oder als vollständig gekürzter Bruch an. **3**

$\bar{m} = \left(\frac{2}{100} \cdot 4 + \frac{41}{100} \cdot 1 + \frac{29}{100} \cdot 2 + \frac{16}{100} \cdot 3 + \frac{12}{100} \cdot 4\right) \text{ Bit} = \frac{203}{100} \text{ Bit}$

Abschließend zeigt Tabelle 7.3 eine alternative präfixfreie Codierung für die oben beschriebenen Tastendruckereignisse. Bei ihrer Erzeugung ist als Zielalphabet nicht das binäre Alphabet $T_2 = \{0, 1\}$, sondern das ternäre Alphabet $T_3 = \{0, 1, 2\}$ zur Anwendung gekommen.

Taste	RESET	INC	DEC	PREV	NEXT
Codewort	10	2	0	12	11

Tabelle 5.3: Codierung des Tastendrücke für ein ternäres Zielalphabet

E) Gegeben sei nun eine Folge von Tastendruckereignissen. Codiert man sie nach dem Code in Tabelle 7.3, erhält man das Wort „210111122“. Wie lautet die zugrundeliegende Folge? **2**

`INC, RESET, NEXT, NEXT, INC, INC`

Aufgabe 6: Minimierung

32

Aufgabe 6.1: Symmetriediagramme

A) Gegeben sei eine Schaltfunktion y in Minimalform wie folgt:
$y = (a \wedge \overline{b}) \vee (a \wedge c) \vee (\overline{b} \wedge d) \vee (\overline{a} \wedge b \wedge \overline{c} \wedge \overline{d})$
Füllen Sie das leere Symmetriediagramm im Folgenden anhand der gegebenen Schaltfunktion y aus. Es sollen dabei alle Kästchen ausgefüllt werden. Das Symmetriediagramm muss am Ende vollständig definiert sein.

6

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

B) Was wird unter dem Begriff Primterm verstanden?

2

Terme mit minimaler Anzahl von Literalen, die trotzdem nur Einsstellen (Nullstellen) der zu beschreibenden Funktion erfassen

C) Gegeben sei das Symmetriediagramm aus Abbildung 6.1. Bestimmen Sie aus dem dort gegebenen Symmetriediagramm zwei unterschiedliche disjunktive Minimalformen (DMF) der Schaltfunktion, die sich mindestens in einem Primimplikanten unterscheiden. Zur grafischen Hilfestellung wurde das Symmetriediagramm zweimal abgebildet. **6**

$y1$ (a über Spalten 2–3; b neben Zeilen 2–3; d neben Zeilen 3–4; c unter Spalten 3–4)

1	-	0	1
1	-	1	1
0	1	-	0
1	0	0	1

$y2$ (a über Spalten 2–3; b neben Zeilen 2–3; d neben Zeilen 3–4; c unter Spalten 3–4)

1	-	0	1
1	-	1	1
0	1	-	0
1	0	0	1

Abbildung 6.1: Symmetriediegramm einer Schaltfunktion

$y1 = (\bar{a} \wedge \bar{b}) \vee (a \wedge b) \vee (b \wedge \bar{d})$

$y2 = (\bar{a} \wedge \bar{b}) \vee (a \wedge b) \vee (\bar{a} \wedge \bar{d})$

D) Bei welchen Termen aus Aufgabe C) handelt es sich um Kerne? **2**

$(\bar{a} \wedge \bar{b})$ und $(a \wedge b)$ sind Kerne.

Aufgabe 6.2: Nelson/Petrick Verfahren

16

A) Im folgenden sei eine Überdeckungstabelle gegeben:

	A	B	C	D	E	F	G	H	I	J
p_1	X		X		X	X				
p_2		X		X					X	X
p_3			X	X			X	X		X
p_4					X	X	X			X
p_5	X	X			X		X	X		

Tabelle 6.1: Überdeckungstabelle des Petrick-Verfahrens

Wenden Sie die Regeln zur Streichung von Kernen sowie die Spalten- und Zeilendominanzregeln des Petrick-Verfahrens auf Tabelle 6.1 an. Gehen Sie dabei entsprechend der folgenden Reihenfolge: **Kerne -> Spaltendominanz -> Zeilendominanz** vor. Die Kosten der Terme sind identisch. Nutzen sie die unten stehende Tabelle für ihre Antwort und tragen sie alle Schritte vollständig ein.

Hinweis: Geben Sie unter "Regel" an, welche Dominazregel zum Tragen kommt (Kern, Spalten-, Zeilendominaz), unter "Streichbar" welche Splaten/Zeilen gestrichen werden und unter "Begründung" wieso die genannten Zeilen oder Spalten gestrichen werden können, also welche andere Zeile/Spalte sie dominiert bzw von welcher sie dominiert werden.

Regel	Streichbar	Begründung
Kern	B,D,I,J	Kern p_2 wegen Spalte I
Spaltendominanz	E	E dominiert A und F
Spaltendominanz	G	G dominiert H
Zeilendominanz	p_4	p_4 wird von p_1 dominiert
Kern	A, C, F	Kern p_1 wegen Spalte F
Zeilendominanz	p_3 oder p_5	p_3 und p_5 dominieren sich gegenseitig
Kern	p_3 oder p_5	Kern wegen Spalte H

Welches Ergebnis erhalten Sie nachdem Sie die Minimierung vollständig durchgeführt haben?

p_1,p_2 und entweder p_3 oder p_5 werden ausgewählt

Aufgabe 7: Optimale Codes

30

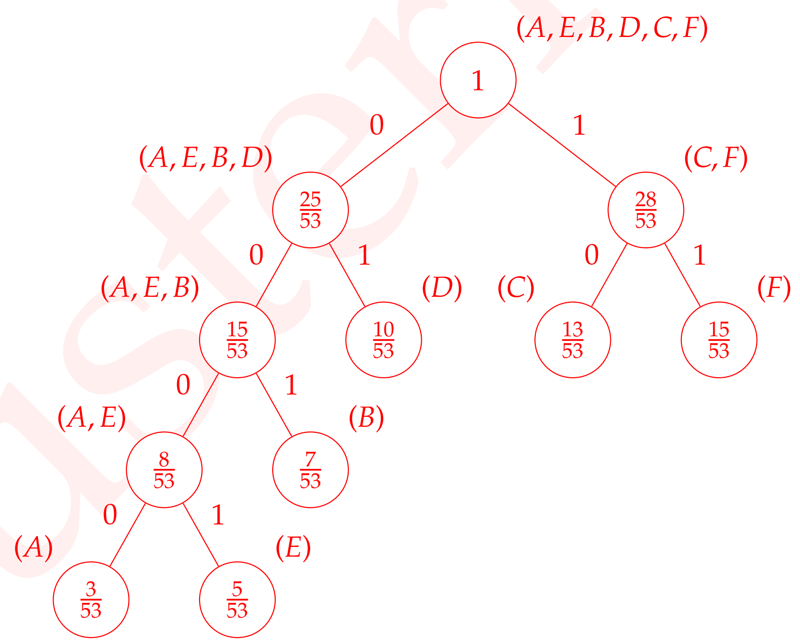
Aufgabe 7.1: Shannon-Fano-Verfahren

Gegeben sei eine gedächtnislose Quelle S. Gemäß der in Tabelle 7.1 dargestellten Auftrittswahrscheinlichkeiten erzeugt sie Symbole aus dem Zeichenvorrat $Y = \{A, B, C, D, E, F\}$.

Symbol	A	B	C	D	E	F
Wahrscheinlichkeit	$\frac{3}{53}$	$\frac{7}{53}$	$\frac{13}{53}$	$\frac{10}{53}$	$\frac{5}{53}$	$\frac{15}{53}$
Codewort	0000	001	10	01	0001	11

Tabelle 7.1: Auftrittswahrscheinlichkeiten und Codewörter der von S erzeugten Symbole

A) Wenden Sie das Shannon-Fano-Verfahren an, um die von S erzeugten Symbole zu codieren. Sortieren Sie die Symbole dafür mit aufsteigenden Auftrittswahrscheinlichkeiten von links nach rechts und erzeugen Sie den vollständigen Codierbaum. Weisen Sie den linken Ästen des entstehenden Baums eine „0“ und den rechten Ästen des Baums eine „1“ zu. Tragen Sie die Codewörter, die sich ergeben, in die dafür vorgesehenen Felder in Tabelle 7.1 ein. 7



B) Codieren Sie das Wort „*CBCAE*“ unter Verwendung der in A) erzeugten Codierung. **2**

Mit der ausgefüllten Tabelle folgt „10 001 10 0000 0001“ als Codierung.

C) Wie lautet die allgemeine Formel zur Berechnung der mittleren Codewortlänge eines Codes über dem N-elementigen Zeichenvorrat $X = \{x_1, \ldots, x_N\}$? Nehmen Sie an, dass $p(x_i)$ die Auftrittswahrscheinlichkeit und $m(x_i)$ die Codewortlänge eines $x_i \in X$ bezeichnet. **1**

$\bar{m} = \sum_{i=1}^{N} m(x_i)p(x_i)$

D) Bestimmen Sie die mittlere Codewortlänge der in A) erzeugten Codierung und geben Sie diese als vollständig gekürzter Bruch an. **3**

$\bar{m} = \left(\frac{3}{53} \cdot 4 + \frac{7}{53} \cdot 3 + \frac{13}{53} \cdot 2 + \frac{10}{53} \cdot 2 + \frac{5}{53} \cdot 4 + \frac{15}{53} \cdot 2\right) \text{Bit} = \frac{129}{53} \text{Bit}$

E) Welche mittlere Codewortlänge ergibt sich, wenn alle Symbole, die von S erzeugt werden, binär und mit Codewörtern gleicher Länge codiert werden sollen? **2**

$\bar{m} = \lceil \log_2 6 \rceil \text{ Bit} = 3 \text{ Bit}$

Aufgabe 7.2: Huffman-Verfahren

A) Welches informationstheoretische Maß dient bei der Codierung einer Quelle als größte untere Schranke der mittleren Codewortlänge $\bar{m}$? Geben Sie dessen Bezeichnung und eine allgemeingültige Formel zur Berechnung an. Benennen Sie dabei alle als Teil der Berechnungsvorschrift verwendeten Größen. **2**

Das theoretische Minimum für die minimale Codewortlänge ist gegeben durch die Entropie (d. h. den mittleren Informationsgehalt) der Quelle. Für eine Quelle, die ein Symbol $x \in X$ aus dem N-elementigen Zeichenvorrat $X = \{x_1, \ldots, x_N\}$ mit Wahrscheinlichkeit $p(x)$ erzeugt, ist sie gegeben durch $H = -\sum_{i=1}^{N} p(x_i) \log_2 (p(x_i))$.

Betrachtet wird nun die Mensch-Maschine-Schnittstelle eines eingebetteten Systems. Sie erlaubt es dem Nutzer, die Konfiguration des Systems zu modifizieren. Um die Benutzereingabe entgegenzunehmen, kommen fünf Tasten zur Anwendung. Gemäß ihrer Funktionalität werden die Tasten im Folgenden als `RESET`, `INC`, `DEC`, `PREV` und `NEXT` bezeichnet.

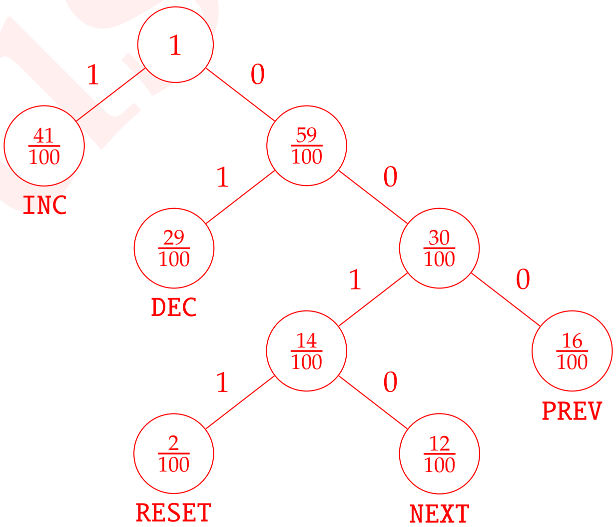
Tastendrücke, die über die Lebensdauer des Systems hinweg getätigt werden, sollen zu Diagnosezwecken in einem nichtflüchtigen Speicher protokolliert werden. Um dies effizient zu realisieren, soll eine Huffman-Codierung der Tastendruckereignisse erfolgen.

Taste	`RESET`	`INC`	`DEC`	`PREV`	`NEXT`
Häufigkeit	2	41	29	16	12
Wahrscheinlichkeit	0,02	0,41	0,29	0,16	0,12
Codewort	0011	1	01	000	0010

Tabelle 7.2: Häufigkeiten, Wahrscheinlichkeiten und Codewörter der Tastendrücke

B) Bei einem Testlauf haben sich die in Tabelle 7.2 dargestellten Tastendruckhäufigkeiten ergeben. Nehmen Sie an, dass diese Häufigkeiten repräsentativ für die Auftrittswahrscheinlichkeiten der einzelnen Tastendrücke sind. Bestimmen Sie die Wahrscheinlichkeiten und tragen Sie diese in dezimaler Form oder als vollständig gekürzter Bruch in Tabelle 7.2 ein. **1**

C) Erzeugen Sie auf Basis der in B) bestimmten Wahrscheinlichkeiten eine Huffman-Codierung der Tastendruckereignisse. Ordnen Sie den Knoten mit der niedrigeren Auftrittswahrscheinlichkeit dabei immer als linkes Kind an und konstruieren Sie den vollständigen Codierbaum. Weisen Sie den linken Ästen des Baums eine „1“ und den rechten Ästen des Baums eine „0“ zu. Beschriften Sie die Blätter des Baums mit den dazugehörigen Tastendruckereignissen. Tragen Sie die Codewörter, die sich ergeben, in Tabelle 7.2 ein. **7**



D) Bestimmen Sie die mittlere Codewortlänge der im Zuge von C) erzeugten Codierung und geben Sie diese in dezimaler Form oder als vollständig gekürzter Bruch an. **3**

$\bar{m} = \left(\frac{2}{100} \cdot 4 + \frac{41}{100} \cdot 1 + \frac{29}{100} \cdot 2 + \frac{16}{100} \cdot 3 + \frac{12}{100} \cdot 4\right) \text{ Bit} = \frac{203}{100} \text{ Bit}$

Abschließend zeigt Tabelle 7.3 eine alternative präfixfreie Codierung für die oben beschriebenen Tastendruckereignisse. Bei ihrer Erzeugung ist als Zielalphabet nicht das binäre Alphabet $T_2 = \{0, 1\}$, sondern das ternäre Alphabet $T_3 = \{0, 1, 2\}$ zur Anwendung gekommen.

Taste	`RESET`	`INC`	`DEC`	`PREV`	`NEXT`
Codewort	10	2	0	12	11

Tabelle 7.3: Codierung des Tastendrücke für ein ternäres Zielalphabet

E) Gegeben sei nun eine Folge von Tastendruckereignissen. Codiert man sie nach dem Code in Tabelle 7.3, erhält man das Wort „210111122“. Wie lautet die zugrundeliegende Folge? **2**

`INC, RESET, NEXT, NEXT, INC, INC`

Aufgabe 8: Zahlensysteme

30

Aufgabe 8.1: Umrechnung von Zahlensystemen

A) Vervollständigen Sie die Tabelle 8.1, indem Sie die offenen Felder durch die entsprechende Konvertierung ergänzen.

12

Hexadezimal	Dezimal	Oktal	Binär
800_H	2048_D	4000_O	$1000\ 0000\ 0000_B$
FDB_H	4059_D	7733_O	$1111\ 1101\ 1011_B$
$E4_H$	228_D	344_O	$1110\ 0100_B$
123_H	291_D	443_O	$0001\ 0010\ 0011_B$

Tabelle 8.1: Umrechnung von Zahlensystemen

B) Konvertieren Sie die Zahl 421_5 (Basis 5) in das Zahlensystem mit der Basis 9 um. Geben Sie dabei die für die Umrechnung nötigen Zwischenschritte an.

3

$$421_5 \widehat{=} 111_D \tag{8.1}$$
$$111 : 9 = 12\ R3 \tag{8.2}$$
$$12 : 9 = 1\ R3 \tag{8.3}$$
$$1 : 9 = 0\ R1 \tag{8.4}$$
$$421_5 \widehat{=} 133_9 \tag{8.5}$$

Aufgabe 8.2: Fließkommazahl

Abbildung 8.1 zeigt eine Darstellung von Fließkommazahlen nach dem IEEE-754-2008-Standard mit halber Genauigkeit (16 Bit). Das höchstwertige Bit stellt das Vorzeichen V dar, die nachfolgenden fünf Bits den Exponenten E und die niederwertigsten zehn Bits die Mantisse M.

V	E4	E3	E2	E1	E0	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0

Abbildung 8.1: 16-Bit-Fließkommazahlenformat

A) Konvertieren Sie $17,25_D$ in das oben definerte Fließkommaformat. **4**
Sollte eine solche Konvertierung nicht möglich sein, begründen Sie dies. Geben Sie das Ergebnis andernfalls in binärer und in hexadezimaler Form an.

$17{,}25 = 10001.01 = 1.000101 * 2^4 = 1.000101 * 2^{19-15}$
M = 00 0101 0000
E = 19 = 1 0011
binär: 0100 1100 0101 0000 => hex: 0x4C50

B) Konvertieren Sie 0_D in das oben definerte Fließkommaformat. **1**
Sollte eine solche Konvertierung nicht möglich sein, begründen Sie dies. Geben Sie das Ergebnis andernfalls in binärer und in hexadezimaler Form an.

binär: 0000 0000 0000 0000 => hex: 0x0000

C) Konvertieren Sie die Zahl $12,0_D * 2^{16}$ in das oben definerte Fließkommaformat. **2**
Sollte eine solche Konvertierung nicht möglich sein, begründen Sie dies. Geben Sie das Ergebnis andernfalls in binärer und in hexadezimaler Form an.

Größtmöglichster Exponent E=11110 => 30 -15 = 15
2^{16} nicht Darstellbar => keine gültige Konvertierung möglich

D) Berechnen Sie unter Verwendung des Stibitz-Codes folgende Aufgabe:

8

-17 - 12

Hinweis: Rechnen Sie mit einem geeigneten Komplement

12 => 0100 0101
17 => 0100 1010
2er - Komplement
-12 => 1011 1011
-17 => 1011 0110

1011 1011
1011 0110
1 111 11 +

0111 0001
1 0011 0011 (Korrektur)

1 1010 0100

$a_{n-1} = b_{n-1} = c_{n-1} = 1$ Carry Überlauf => Negatives Ergebnis
Komplement
1010 0100 (2er Komplement) => 0101 1100 => 29
Ergebnis = -29

10er: Komplement
-17 => 83 => 1011 0110
-12 => 88 => 1011 1011

71 => 29

1010 0100 => 71

Zusatzblatt zu Aufgabe ☐: