

# KLAUSUR

## Digitaltechnik 1

Wintersemester 2022/2023

## MUSTERLÖSUNG

Prüfungsfach: Digitaltechnik 1

Studiengang: Technische Informatik

Semestergruppe: TIB2, IEP2

Fachnummer: 1052032

Erlaubte Hilfsmittel: keine

Zeit: 90 min.

**Tragen Sie hier bitte Ihren Daten ein:**

Name:	Vorname:	Matrikelnummer:

**Wichtiger Hinweis für die Bearbeitung der Aufgaben:**

Schreiben Sie bitte Ihre Lösungen möglichst auf die Aufgabenblätter.

Sollte der vorgesehene Platz nicht reichen, verwenden Sie bitte jeweils die Rückseite.

Viel Erfolg wünscht Ihnen

Jonas Fuhrmann und Michael Koidis

# 1 Boolesche Algebra

## 1.1 Boolesche Gleichung

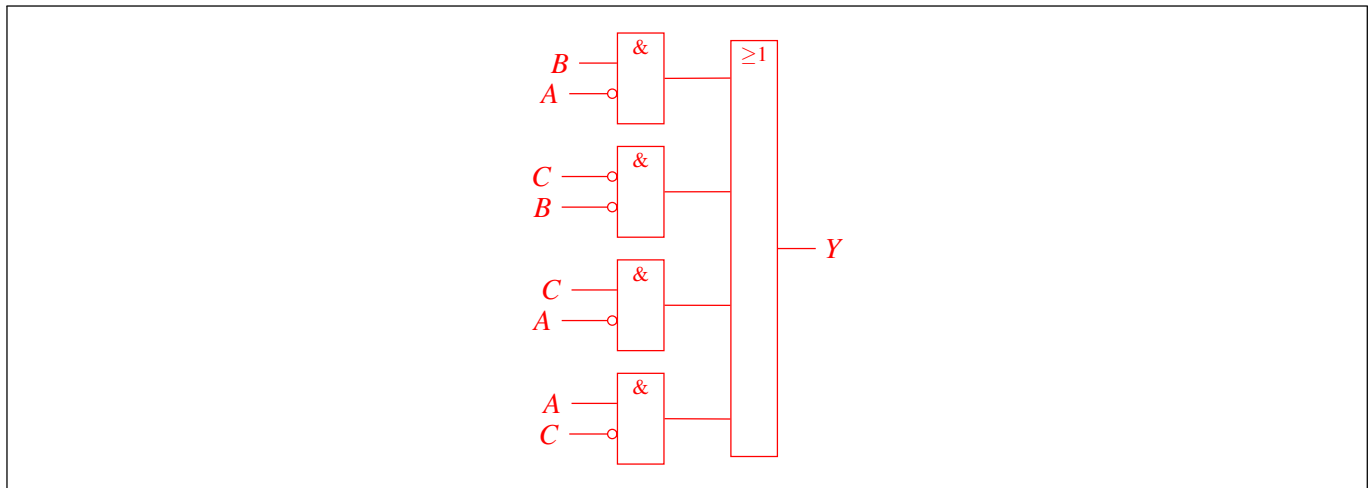
(7 Punkte)

### Schaltungsanalyse

Gegeben ist die boolesche Gleichung:  $Y = B \wedge \bar{A} \vee \bar{C} \wedge \bar{B} \vee C \wedge \bar{A} \vee A \wedge \bar{C}$

Zeichnen Sie dazugehörige Schaltung

(2 Punkte)



Wie ist die Funktionslänge  $l$  und die Schachteltiefe  $k$  der Schaltung?

$l = 12$

(1 Punkt)

$k = 2$

(1 Punkt)

### Funktionstabelle

Bestimmen Sie die Funktionstabelle der booleschen Gleichung  $Y = B \wedge \bar{A} \vee \bar{C} \wedge \bar{B} \vee C \wedge \bar{A} \vee A \wedge \bar{C}$  (3 Punkte)

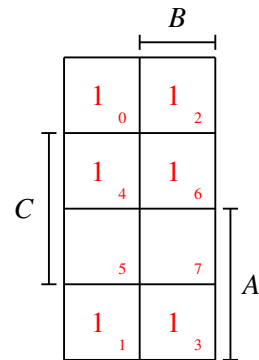
	C	B	A	$B \wedge \bar{A}$	$\bar{C} \wedge \bar{B}$	$C \wedge \bar{A}$	$A \wedge \bar{C}$	$Y$
0	0	0	0		1			1
1	0	0	1		1		1	1
2	0	1	0	1				1
3	0	1	1				1	1
4	1	0	0			1		1
5	1	0	1					
6	1	1	0	1		1		1
7	1	1	1					

## 1.2 Minimierung

(6 Punkte)

Bestimmen Sie die disjunktive Minimalform  $Y_{DMF}$  der booleschen Gleichung  $Y = \bar{A} \wedge \bar{B} \vee A \wedge B \vee B \wedge C \vee A \wedge \bar{B}$ . Übertragen Sie zuerst Ihre Lösung aus Aufgabe 1.1 in die Tabelle 1, füllen dann das KV-Diagramm aus und bestimmen danach die disjunktive Minimalform  $Y_{DMF}$ .

	C	B	A	Y
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	
6	1	1	0	1
7	1	1	1	



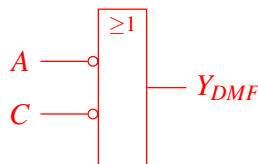
$$Y_{DMF} = \bar{A} \vee \bar{C}$$

(3 Punkte)

Tabelle 1: Ergebnis aus Aufgabe 1.1

Zeichnen Sie die Schaltung der oben bestimmten disjunktiven Minimalform  $Y_{DMF}$ ?

(1 Punkt)



Wie ist die Funktionslänge  $l$  und die Schachteltiefe  $k$  der zur disjunktiven Minimalform  $Y_{DMF}$  gehörenden Schaltung?

$$l = 2$$

(1 Punkt)

$$k = 1$$

(1 Punkt)



## 2.2 Zahlendarstellung nach IEEE 754

(5 Punkte)

Wandeln Sie die Dezimalzahl  $(-3,75)_{10}$  in eine Gleitkommazahl in einfacher Genauigkeit nach IEEE 754 in hexadezimaler Schreibweise um.

Hinweis: Eine Gleitkommazahl in einfacher Genauigkeit (32 Bit) ist nach IEEE 754 wie folgt codiert:

Bits	1	8	23
	VZ von $M$	$E + 127$	$ M $ ohne $m_0$

- Das Bit 31 (MSB) kennzeichnet das Vorzeichen.
- Die nächsten 8 Bit 30...23 geben den Exponenten an (Offsetdarstellung um 127).
- Die nächsten 23 Bit 22...0 geben die normalisierte Mantisse ohne die Vorkomma-Eins an.

Abbildung 1: Darstellung von Gleitkommazahl in einfacher Genauigkeit (32 Bit) nach IEEE 754

<b>normalisierte Zahl</b>	$\pm$	$0 < \text{Exponent} < \text{max}$	Mantisse beliebig
<b>denormalisierte Zahl</b>	$\pm$	0000 0000	Mantisse nicht alle Bits 0
<b>Null</b>	$\pm$	0000 0000	0...0
<b>Unendlich</b>	$\pm$	1111 1111	0...0
<b>NaN</b>	$\pm$	1111 1111	Mantisse nicht alle Bits 0

Tabelle 3: Sonderfälle Gleitkommazahl in einfacher Genauigkeit (32 Bit) nach IEEE 754

$$(-3,75)_{10} = (-11,11)_2 = (-1,111)_2 \cdot 2^1 \quad (1 \text{ Punkt})$$

VZ: „-“  $\rightarrow$  1 (0.5 Punkt)


$$\text{Exp: } 1 + 127 = 128 = 1000\ 0000 \quad (1 \text{ Punkt})$$

Mant. 111 0000 0000 0000 0000 0000 (0.5 Punkt)

$\underbrace{1}_{\text{VZ}} \underbrace{100\ 0000\ 0}_{\text{Exp.}} \underbrace{111\ 0000\ 0000\ 0000\ 0000\ 0000}_{\text{Mant.}}$  (1 Punkt)

$\underbrace{1100}_{\text{C}} \underbrace{0000}_{\text{0}} \underbrace{0111}_{\text{7}} \underbrace{0000}_{\text{0}} \underbrace{0000}_{\text{0}} \underbrace{0000}_{\text{0}} \underbrace{0000}_{\text{0}} \underbrace{0000}_{\text{0}}$

Ergebnis: (C070 0000) Hex (1 Punkt)

Prüfungsfach: <b>Digitaltechnik 1</b>	Wintersemester 2022/2023	
Name, Vorname:	Mat.-Nr.:	

## 2.3 Blockcodes

(11 Punkte)

Gegeben ist die Generatormatrix

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Wie viele Nachrichtenstellen  $m$  haben Codewörter die mit der obigen Generatormatrix  $\mathbf{G}$  erzeugt werden können?

Die Generatormatrix hat allgemein die Dimension:  $m \cdot (m + k)$ .  
In diesem Fall ist die Dimension von  $\mathbf{G}$ :  $2 \cdot 6$ .  
Somit ist  $m = 2$  (1 Punkt)

Wie lautet die mit Hilfe der Generatormatrix  $\mathbf{G}$  gezeugten Codeworte  $\mathbf{Y}_0$  bis  $\mathbf{Y}_3$ ?


		1	0	0	0	1	1		
		0	1	1	0	1	0		
0	0	0	0	0	0	0	0	$\mathbf{Y}_0 = 000000$	(1.5 Punkte)
0	1	0	1	1	0	1	0	$\mathbf{Y}_1 = 011010$	(1.5 Punkte)
1	0	1	0	0	0	1	1	$\mathbf{Y}_2 = 100011$	(1.5 Punkte)
1	1	1	1	1	0	0	1	$\mathbf{Y}_3 = 111001$	(1.5 Punkte)

Wie groß ist die Hammingdistanz des mit der Generatormatrix  $\mathbf{G}$  erzeugten Codes?

$d(\mathbf{Y}_1, \mathbf{Y}_2) = 3$ ,  $d(\mathbf{Y}_2, \mathbf{Y}_3) = 4$ ,  $d(\mathbf{Y}_3, \mathbf{Y}_4) = 3$   
 $d(\mathbf{Y}_1, \mathbf{Y}_3) = 3$ ,  $d(\mathbf{Y}_2, \mathbf{Y}_4) = 3$   
 $d(\mathbf{Y}_1, \mathbf{Y}_4) = 4$   
 Hammingdistanz  $h = 3$  (3 Punkt)

Wie viele Bitfehler können sicher erkannt werden?

$e^* = h - 1 = 2$  (1 Punkt)

Prüfungsfach: <b>Digitaltechnik 1</b>	Wintersemester 2022/2023	
Name, Vorname:	Mat.-Nr.:	

## 2.4 VHDL

(9 Punkte)

Gegeben ist die in VHDL beschriebene Komponente INCREMENTER. Vervollständigen Sie die Komponente so, dass die Funktion  $Y = (X) + 1$  ausgeführt und die Condition Flags, Carry (CF), Overflow (OF), Zero (ZF) und Negative (NF) korrekt berechnet werden!

*Tipp: Das Overflow Flag ist dann gesetzt, wenn das MSB des Operanden 0 und das MSB des Ergebnisses 1 entspricht.*

```

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.numeric_std.all;

ENTITY ADDER IS
    PORT(IN1, IN2 : IN  STD_LOGIC_VECTOR(2 downto 0);
          C_IN    : IN  STD_LOGIC;
          SUM     : OUT STD_LOGIC_VECTOR(2 downto 0);
          C_OUT   : OUT STD_LOGIC;
END ADDER;
ARCHITECTURE BEHAV OF ADDER IS
    SIGNAL result: STD_LOGIC_VECTOR(3 downto 0);
BEGIN
    result <= STD_LOGIC_VECTOR(UNSIGNED('0' & IN1) + UNSIGNED(IN2) + C_IN);
    SUM    <= result(2 downto 0);
    C_OUT  <= result(3);
END BEHAV;
-----
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY INCREMENTER IS
    PORT(X : IN  STD_LOGIC_VECTOR(2 downto 0);
          Y : OUT STD_LOGIC_VECTOR(2 downto 0);
          CF, OF, ZF, NF : OUT STD_LOGIC
    );
END INCREMENTER;
ARCHITECTURE FUNC OF INCREMENTER IS
    SIGNAL SUM : STD_LOGIC_VECTOR(2 downto 0);
    SIGNAL C : STD_LOGIC;
    -----
BEGIN
    ADD: ENTITY work.ADDER
        PORT MAP(
            IN1    => X,
            IN2    => "000" (alt. "001"),
            C_IN   => '1' (alt. '0'),
            SUM    => SUM,
            C_OUT  => C (alt. CF)
        );
    Y <= SUM
    CF <= C
    -----
    OF <= (NOT X(2)) AND SUM(2)
    -----
    ZF <= (NOT SUM(2)) AND (NOT SUM(1)) AND (NOT SUM(0))
    -----
    NF <= SUM(2)
    -----
END FUNC;

```

(0.5 Punkt)

(1 Punkt)

(1 Punkt)

(0.5 Punkt)

(1 Punkt)

(2 Punkt)

(2 Punkt)

(1 Punkt)

### 3 Hardware

Die in Abbildung 2 dargestellte 8 Bit-ALU enthält neben einem 8 Bit Addierer eine 8 Bit-Logik-Einheit, ein 8-faches AND-Gatter sowie einen Block „Status“ zur Bildung des Carry-Flags (CF), Overflow-Flags (OF), Zero-Flags (Z) und Negativ-Flags (N).

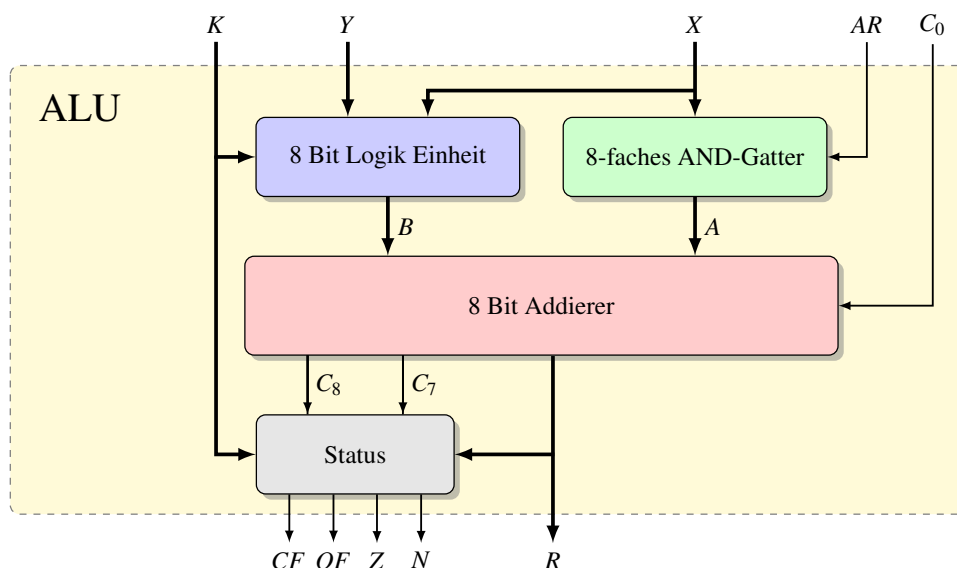


Abbildung 2: Aufbau 8-Bit ALU

Die Signale haben folgende Bitbreite:

Signalname	A	B	X	Y	R	K	AR	C <sub>0</sub>	C <sub>7</sub>	C <sub>8</sub>	CF	OF	Z	N
Breite in Bit	8	8	8	8	8	4	1	1	1	1	1	1	1	1

Tabelle 4: Bitbreite der Signale

$AR=0$  sperrt das 8-fach AND-Gatter, d.h.  $A = 0$ .  $AR=1$  schaltet  $X$  nach  $A$  durch, d.h.  $A = X$ .


Die gültigen Steuerworte  $K$  sind der Tabelle 5 zu entnehmen.

Steuerwort (K)	Ergebnis für Stelle $B_i$	Logik-Funktion
$(0000) = 0_H$	$B_i = 0$	Kontradiktion
$(0001) = 1_H$	$B_i = 1$	Tautologie
$(0010) = 2_H$	$B_i = X_i$	Identität X
$(0011) = 3_H$	$B_i = Y_i$	Identität Y
$(0100) = 4_H$	$B_i = \overline{X}_i$	Bitweise Invertierung X
$(0101) = 5_H$	$B_i = \overline{Y}_i$	Bitweise Invertierung Y
$(1000) = 8_H$	$B_i = X_i \vee Y_i$	OR
$(1001) = 9_H$	$B_i = X_i \wedge Y_i$	AND

Tabelle 5: Wirkung des Steuersignals (K) auf  $B_i$  in Abhängigkeit von  $X_i$  und  $Y_i$  ( $i = 0, \dots, 7$ ).





Prüfungsfach: <b>Digitaltechnik 1</b>	Wintersemester 2022/2023	
Name, Vorname:	Mat.-Nr.:	

## 4 Verständnisfragen

(7 Punkte)

### Rechenleistung

(2 Punkte)

Welche Ansätze zur Steigerung der Rechenleistung gibt es? Nennen Sie 4!

- |                           |             |
|---------------------------|-------------|
| (1) Höhere Taktfrequenz   | (0.5 Punkt) |
| (2) Pipelining-Konzept    | (0.5 Punkt) |
| (3) Superskalares Konzept | (0.5 Punkt) |
| (4) Multi-Core-Konzept    | (0.5 Punkt) |

### Speicherelemente

(5 Punkte)

Erklären Sie den Unterschied zwischen **ungesteuerten**, **zustandsgesteuerten** und **flankengesteuerten** Flipflops. Warum werden in den meisten Fällen taktzustandsgesteuerte Mehrspeicher-Flipflops oder flankengesteuerte Flipflops eingesetzt?

- Ungesteuertes Flipflop: Eingangssignale wirken sich zu jedem Zeitpunkt auf den Zustand des FF aus. (1 Punkt)
- Zustandsgesteuertes Flipflop: Eingangssignale wirken sich nur im aktiven Zustand des zusätzlichen Taktsignals auf den Zustand des FF aus. Das Ausgangssignal kann sich während der gesamten aktiven Taktphase ändern. (1 Punkt)
- Flankengesteuerten Flipflops: Das Ausgangssignal ändert sich nur während der aktiven Taktflanke ( $0 \rightarrow 1$  = positive Flanke,  $1 \rightarrow 0$  = negative Flanke). Spätere oder frühere Änderungen der Eingangssignale beeinflussen den Ausgang nicht. (1 Punkt)

Hintereinandergeschaltete zustandsgesteuerte Flipflops schalten bei  $T=1$  durch alle FF durch. Eine Änderung am Eingang darf sich innerhalb einer aktiven Taktphase nicht auf die nachfolgende Schaltung auswirken  $\rightarrow$  Flankengesteuertes FF o. Mehrspeicher FF (2 Punkt)