
Klausur

Name: Matrikelnummer:

Studiengang: Unterschrift:

Bearbeitungszeit: 90 Minuten

Gesamtpunktzahl: 60 Punkte

1. Verwenden Sie einen nicht netzwerkfähigen Taschenrechner!
 2. Verwenden Sie ein ein beidseitig handgeschriebenes DIN A 4 Formelblatt!
 3. Verwenden Sie keinen Bleistift oder Rotstift!
 4. Unleserliche Angaben werden nicht gewertet!
 5. Verwenden Sie den Notizbogen für Nebenrechnungen!
 6. Geben Sie den Rechenweg klar strukturiert und leserlich an!
 7. Nur Lösungen in den Lösungsfeldern werden gewertet!
Nutzen Sie ggf. die Lösungsboxen der englischen Version!

Examination

Name: Student number:

Branch of studies: Signature:

Working time: 90 Minutes

Total points: 60 Points

1. Use a pocket calculator without network capabilities!
2. Use a double sided hand-written DIN A 4 formulary!
3. Do not use lead pen or red ink!
4. Unreadable sections are not assessed!
5. Use the additional sheets for side calculatipons!
6. Give your solutions clearly structured and readable!
7. Solutions will be assessed if given in the text boxes!
Use the text boxes of the german version if required!

Aufgabe 1: Digitale Codes und Arithmetik (15 Punkte)

1. Geben Sie den Wertebereich des 64-Bit Binärkode an. (1 Punkt)

$$0, 1, 2, \dots, 2^{64} - 1$$

2. Geben Sie den Wertebereich des 8-Bit 127-Exzess Kode an. (1 Punkt)

$$-127, -126, -125, \dots, (2^8 - 1) - 127$$

3. Geben Sie den Wertebereich des 32-Bit Zweierkomplement Kode an. (2 Punkte)
Von den 32-Bit sind 8-Bit Nachkommastellen.

$$-2^{23}, -2^{23} + 1 \cdot 2^{-8}, -2^{23} + 2 \cdot 2^{-8}, \dots, 2^{23} - 2^{-8}$$

4. Transformieren Sie folgende Zahl im IEEE-Kode in den 16-Bit Zweierkomplement Kode. Bestimmen Sie die nötige Anzahl Nachkommastellen. (4 Punkte)

$$1 \quad 1000 \ 0011 \quad 1101 \ 0000 \ 0000 \ 0000 \ 0000 \ 000$$

$$VZ = 1 \text{ negative Zahl}$$

$$Charakteristik = 1000 \ 0011$$

$$Exponent = Charakteristik - 127$$

$$= 131 - 127$$

$$= 4$$

$$Mantisse = (1,)1101 \ 0000 \ 0000 \ 0000 \ 0000 \ 000$$

$$= 1 + 2^{-1} + 2^{-2} + 2^{-4}$$

$$= 1,8125$$

$$Dezimalwert = -1,8125 \cdot 2^4 = -29,4$$

$$Binär = -0000 \ 0000 \ 1100 \ 1,011$$

$$Zweierkomplement = \underline{1111 \ 1111 \ 0011 \ 0,101}$$

Digital Codes and Arithmetic

(15 Punkte)

1. Give the range of values for 64-bit binary encoding. (1 Point)

2. Give the range of values for 8-bit 127-excess encoding. (1 Point)

3. Give the range of values for 32-bit Two's Complement encoding. (2 Points)
Eight of the 32-Bit are fraction bits.

4. Transform the following IEEE number to 16 bit Two's Complement code. (4 Punkte)
Determine the required number of fractional bits.

1 1000 0011 1101 0000 0000 0000 0000

5. Addieren Sie die folgenden zwei IEEE-Zahlen. Geben Sie alle Rechenschritte an. (7 Punkte)

1	1000 0011	1101 0000 0000 0000 0000 000
0	1000 0010	0101 0000 0000 0000 0000 000

Wert 1 entspricht dem Wert aus Aufgabenteil 1!

Wert 1	Wert 2
VZ = 1	VZ = 0
Exponent = 4	Exponent = 3
Mantisse = (1,)1101 0...0	Mantisse = (1,)0101 0...0

Wähle Exponent 3 und gleiche Mantissen an.

Mantisse 1 + Mantisse 2	Mantisse 1 + Mantisse 2
$- 0011, 1010 0...0$ Binär + $0001, 0101 0...0$ Binär	$1100, 0110 0...0$ ZK + $0001, 0101 0...0$ ZK ----- $1101, 1001 0...0$ ZK

Mantisse des Ergebnisses hat führende 1, ist also negativ!

Mantisse negieren und normalisieren!

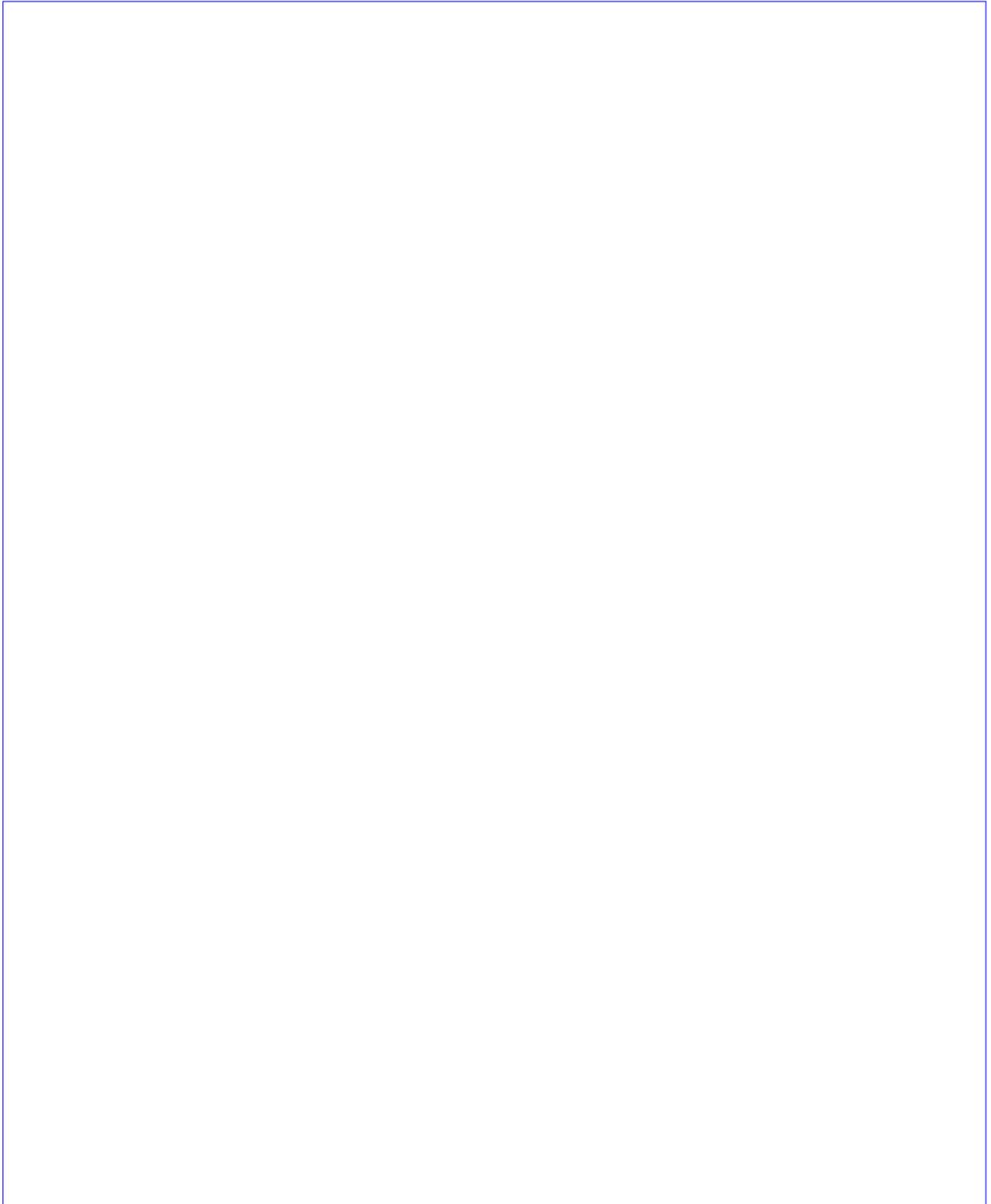
Ergebnis	
VZ = 1	negative Zahl
Mantisse = $1101, 1001 0...0$	ZK
= $-0010, 0111 0...0$	Binär
= $-0001, 0011 10..0 \cdot 2^1$	Binär
Charakteristik = Exponent+127	127-Exzess
= $1 + 127$	127-Exzess
= $1000 0000$	127-Exzess

Ergebnis: 1 1000 0000 0011 1000 0000 0000 0000 000

5. Add the following two IEEE numbers. Give all steps of computation. **(7 Points)**

1 1000 0011 1101 0000 0000 0000 0000 000

0 1000 0010 0101 0000 0000 0000 0000 000



Aufgabe 2: Boole'sche Algebra und Optimierung (15 Punkte)

-1-11
1-101
10001
01-00
1-100
01-10
00100
10000

1. Die oben angegebenen Terme seien Belegungen aller Implikate der Funktion f . Optimieren Sie die Funktion mit dem Verfahren von Quine McCluskey oder mit KV-Diagramm. **(12 Punkte)**

1. Ordnung		2. Ordnung		3. Ordnung	
#	abcde	#	abcde	#	abcde
16*	10000	(16,17)*	1000-	(16,17,20,21)	10-0-
4*	00100	(16,20)*	10-00	(16,20,17,21)	10-0-
8*	01000	(4,12)*	0-100	(4,12,20,28)	--100
17*	10001	(4,20)*	-0100	(4,20,12,28)	--100
12*	01100	(8,12)*	01-00	(8,12,10,14)	01--0
20*	10100	(8,10)*	010-0	(8,10,12,14)	01--0
10*	01010	(17,21)*	10-01	(20,21,28,29)	1-10-
11*	01011	(12,14)*	011-0	(20,28,21,29)	1-10-
21*	10101	(12,28)*	-1100	(10,11,14,15)	01-1-
28*	11100	(20,21)*	1010-	(10,14,11,15)	01-1-
14*	01110	(20,28)*	1-100	(11,27,15,31)	-1-11
27*	11011	(10,11)*	0101-	(11,15,27,31)	-1-11
15*	01111	(10,14)*	01-10		
29*	11101	(11,27)*	-1011		
31*	11111	(11,15)*	01-11		
		(21,29)*	1-101		
		(28,29)*	1110-		
		(14,15)*	0111-		
		(27,31)*	11-11		
		(15,31)*	-1111		
		(29,31)	111-1		

Boolean algebra and optimization

(15 Points)

-1-11

1-101

10001

01-00

1-100

01-10

00100

10000

1. Let the given patterns be allocations for sum terms of the function f (12 Points)
(product of sums). Optimize the function with the algorithm of Quine
McCluskey or with KV-diagram.

zu 1. (Fortsetzung der Optimierung mit Quine McCluskey)

Primterme	4	8	10	11	12	14	15	16	17	20	21	27	28	29	31
(29,31)														x	x
(16,17,20,21)								⊗	⊗	x	x				
(4,12,20,28)	⊗				x					x			x		
(8,10,12,14)		⊗	x		x	x									
(20,21,28,29)										x	x		x	x	
(10,11,14,15)			x	x		x	x								
(11,15,27,31)				x			x					⊗			x

$$f = (\bar{a} \vee b \vee d) \wedge (\bar{c} \vee \bar{d} \vee \bar{e}) \wedge (a \vee \bar{b} \vee e) \wedge (\bar{a} \vee \bar{c} \vee d) \wedge (\bar{b} \vee \bar{d} \vee \bar{e})$$

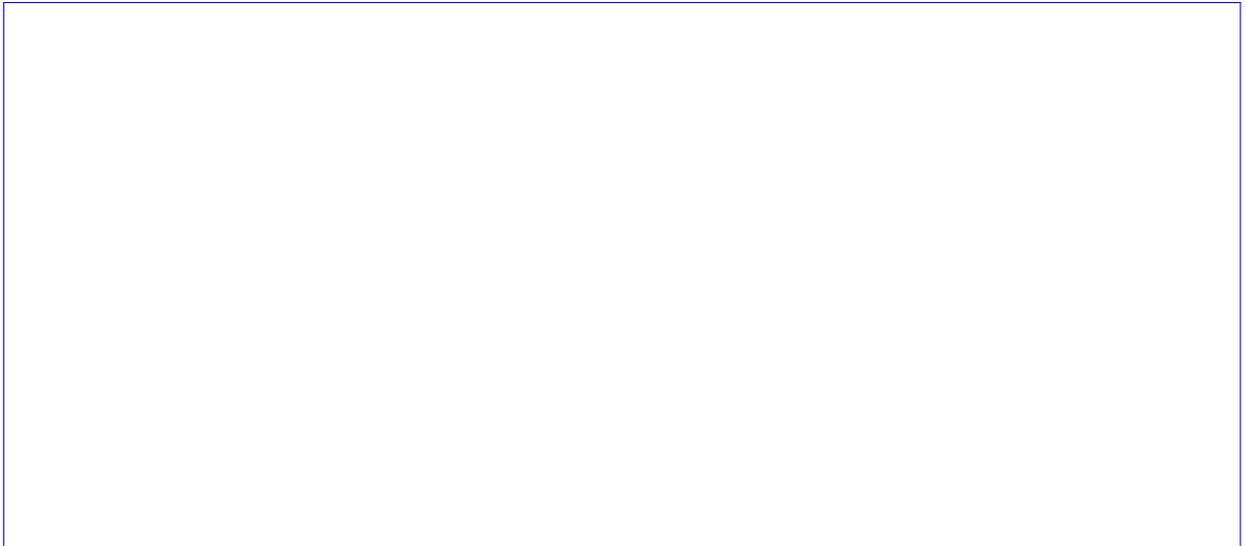
2. Transformieren Sie eine der folgenden boole'schen Gleichung in das NAND-NOT Operatorensystem. (⊕ bezeichnet die XOR-Funktion) **(3 Punkte)**

$$SUM = A \oplus B \oplus CI$$

$$CO = (A \wedge B) \vee CI \wedge (A \vee B)$$

$$\begin{aligned} SUM &= A \oplus B \oplus CI = \overline{(A \oplus B) \wedge CI} \vee ((A \oplus B) \wedge \overline{CI}) \\ &= \overline{\overline{((\bar{A} \wedge B) \vee (\bar{B} \wedge A))} \wedge CI} \vee (((\bar{A} \wedge B) \vee (\bar{B} \wedge A)) \wedge \overline{CI}) \\ &= (((\bar{A} \wedge B) \wedge (\bar{B} \wedge A)) \wedge CI) \vee \overline{\overline{((\bar{A} \wedge B) \vee (\bar{B} \wedge A))} \wedge \overline{CI}} \\ &= \overline{\overline{((\bar{A} \wedge B) \wedge (\bar{B} \wedge A)) \wedge CI} \vee \overline{\overline{((\bar{A} \wedge B) \vee (\bar{B} \wedge A))} \wedge \overline{CI}}} \\ &= (((\bar{A} \wedge B) \wedge (\bar{B} \wedge A)) \wedge \overline{CI}) \wedge \overline{\overline{((\bar{A} \wedge B) \vee (\bar{B} \wedge A)) \wedge \overline{CI}}} \\ &= \overline{\overline{((\bar{A} \wedge B) \wedge (\bar{B} \wedge A)) \wedge \overline{CI}} \wedge \overline{\overline{((\bar{A} \wedge B) \vee (\bar{B} \wedge A)) \wedge \overline{CI}}}} \\ \\ CO &= (A \wedge B) \vee CI \wedge (A \vee B) = (A \wedge B) \vee (CI \wedge A) \vee (CI \wedge B) \\ &= \overline{\overline{(A \wedge B) \vee (CI \wedge A) \vee (CI \wedge B)}} = \overline{\overline{(A \wedge B)} \wedge \overline{\overline{(CI \wedge A)} \wedge \overline{\overline{(CI \wedge B)}}}} \\ &= \overline{\overline{(A \wedge B)} \wedge \overline{\overline{(CI \wedge A)} \wedge \overline{\overline{(CI \wedge B)}}}} \\ &= \overline{\overline{(A \wedge B)} \wedge \overline{\overline{(CI \wedge A)} \wedge \overline{\overline{(CI \wedge B)}}}} \end{aligned}$$

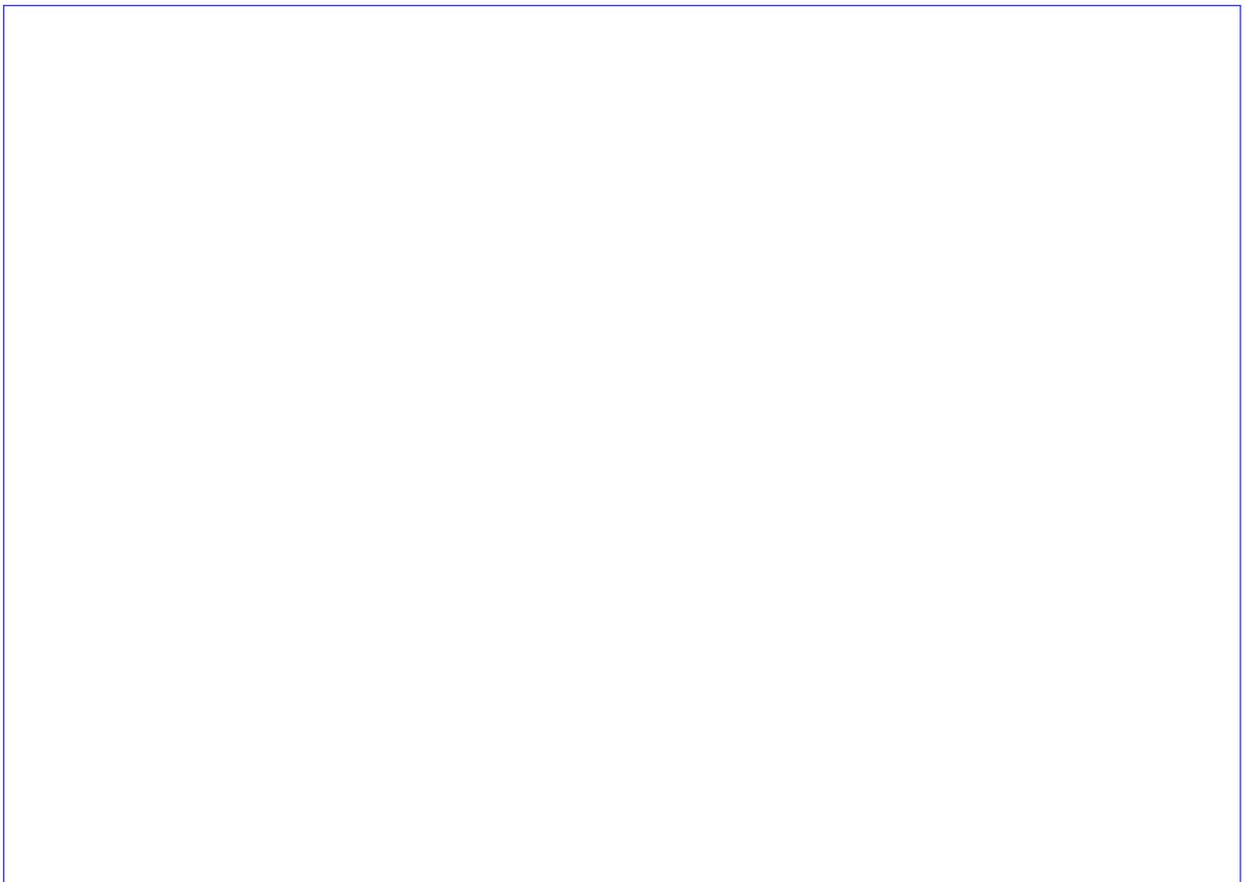
zu 1. (cont'd optimization with Quine McCluskey)



2. Transform one of the following boolean function to the NAND-NOT operator system. (\oplus is the XOR function) **(3 Points)**

$$SUM = A \oplus B \oplus CIN$$

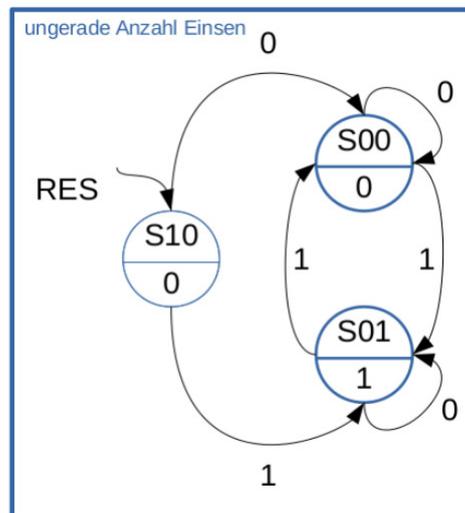
$$CO = (A \wedge B) \vee C \wedge (A \vee B)$$



Aufgabe 3: Endliche Determ. Automaten (15 Punkte)

Als Parity werden ein oder mehrere Kontrollbits bezeichnet, die aus einer zu übermittelnden Nachricht errechnet werden. Mit EVEN-Parity bezeichnet man ein Paritätsbit, welches die Nachricht zu einer geraden Anzahl Einsen ergänzt. Mit ODD-Parity bezeichnet man ein Paritätsbit, welches die Nachricht zu einer ungeraden Anzahl Einsen ergänzt.

1. Entwickeln Sie einen endlichen deterministischen Automaten zur Berechnung eines 1-Bit EVEN-Paritätsbit für einen 1-Bit Eingabestrom. **(5 Punkte)**
 Geben Sie den Automaten als MOORE-Architektur an.



Endliche Determ. Automaten

(15 Punkte)

Parities are one or more control bits, that are derived from a message to be sent. An EVEN parity is a parity bit that complements the number of ones in a message to an even number of ones. An ODD parity is a parity that complements the number of ones in a message to an odd number of ones.

1. Develop a finite state machine to compute a 1-bit EVEN parity from a 1-bit input sequence. The state machine shall be a MOORE architecture. (5 Punkte)

2. Geben Sie die Zustandsübergangsfunktion des Automaten aus Aufgabenteil 1 in Kanonischer Disjunktiver Normalform an. **(5 Punkte)**

S_t	I	RES	S_{t+1}	Zustandsname	$S(1:0)$	$S_t(1:0)$	I	RES	$S_{t+1}(1:0)$
–	–	1	$S10$			–	–	1	1 0
$S10$	0	0	$S00$	$S00$	00	1 0	0	0	0 0
$S10$	1	0	$S01$	$S01$	01	1 0	1	0	0 1
$S00$	0	0	$S00$	$S01$	01	0 0	0	0	0 0
$S00$	1	0	$S01$	$S10$	10	0 0	1	0	0 1
$S01$	0	0	$S01$			0 1	0	0	0 1
$S01$	1	0	$S00$			0 1	1	0	0 0

$$S_{t+1}(1) = RES$$

$$S_{t+1}(0) = (S_t(1) \wedge \overline{S_t(0)} \wedge I \wedge \overline{RES}) \vee (S_t(1) \wedge S_t(0) \wedge I \wedge \overline{RES}) \vee (\overline{S_t(1)} \wedge S_t(0) \wedge \overline{I} \wedge \overline{RES})$$

3. Geben Sie die Ausgabefunktion des Automaten aus Aufgabenteil 1 mit asynchronem Reset als Konjunktive Normalform an. **(5 Punkte)**

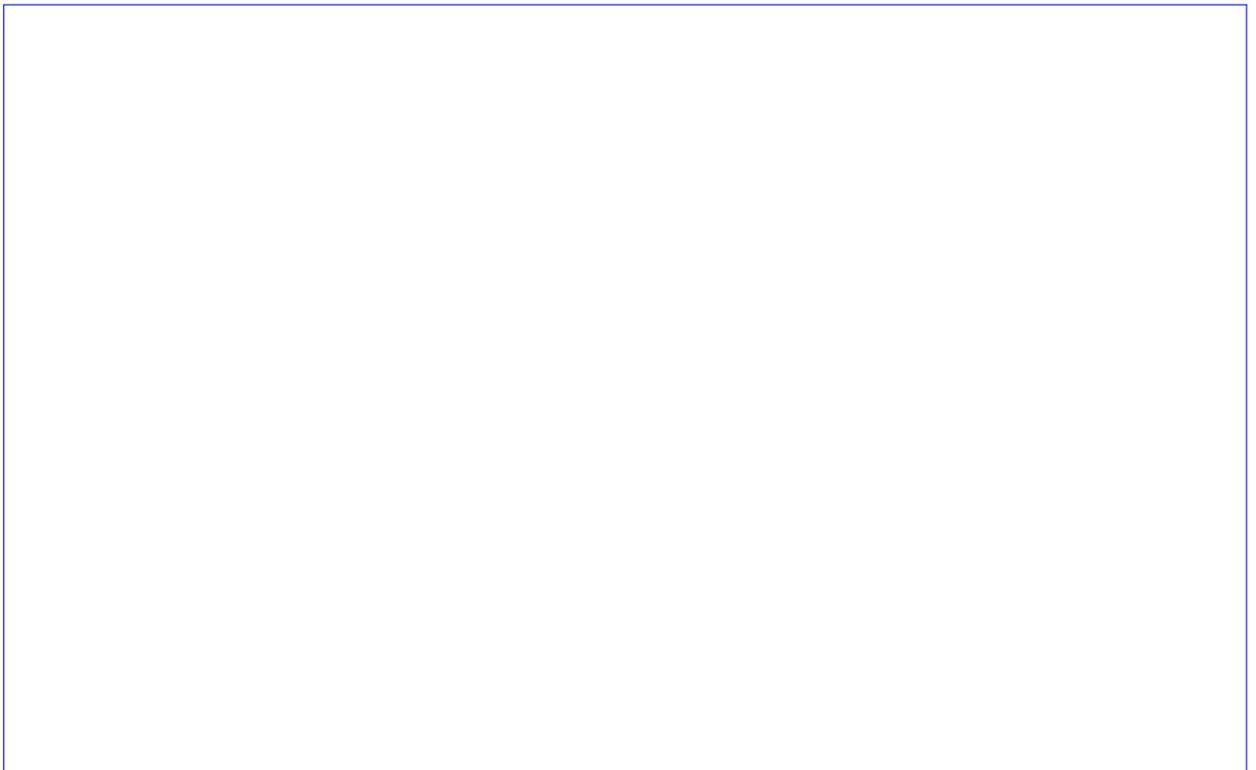
S_t	RES	P	Zustandsname	$S(1:0)$	$S_t(1:0)$	RES	P
–	1	0			–	1	0
$S10$	0	0	$S00$	00	10	0	0
$S00$	0	0	$S01$	01	00	0	0
$S01$	0	1	$S10$	10	01	0	1

$$P = \overline{RES} \wedge (\overline{S_t(1)} \vee S_t(0) \vee RES) \wedge (S_t(1) \vee S_t(0) \vee RES)$$

2. Give the state transfer function of the finite state machine from part 1 as **(5 Punkte)** canonical sum of products.



3. Give the output function of the finite state machine in part 1 with **(5 Punkte)** asynchronous reset as product of sums.

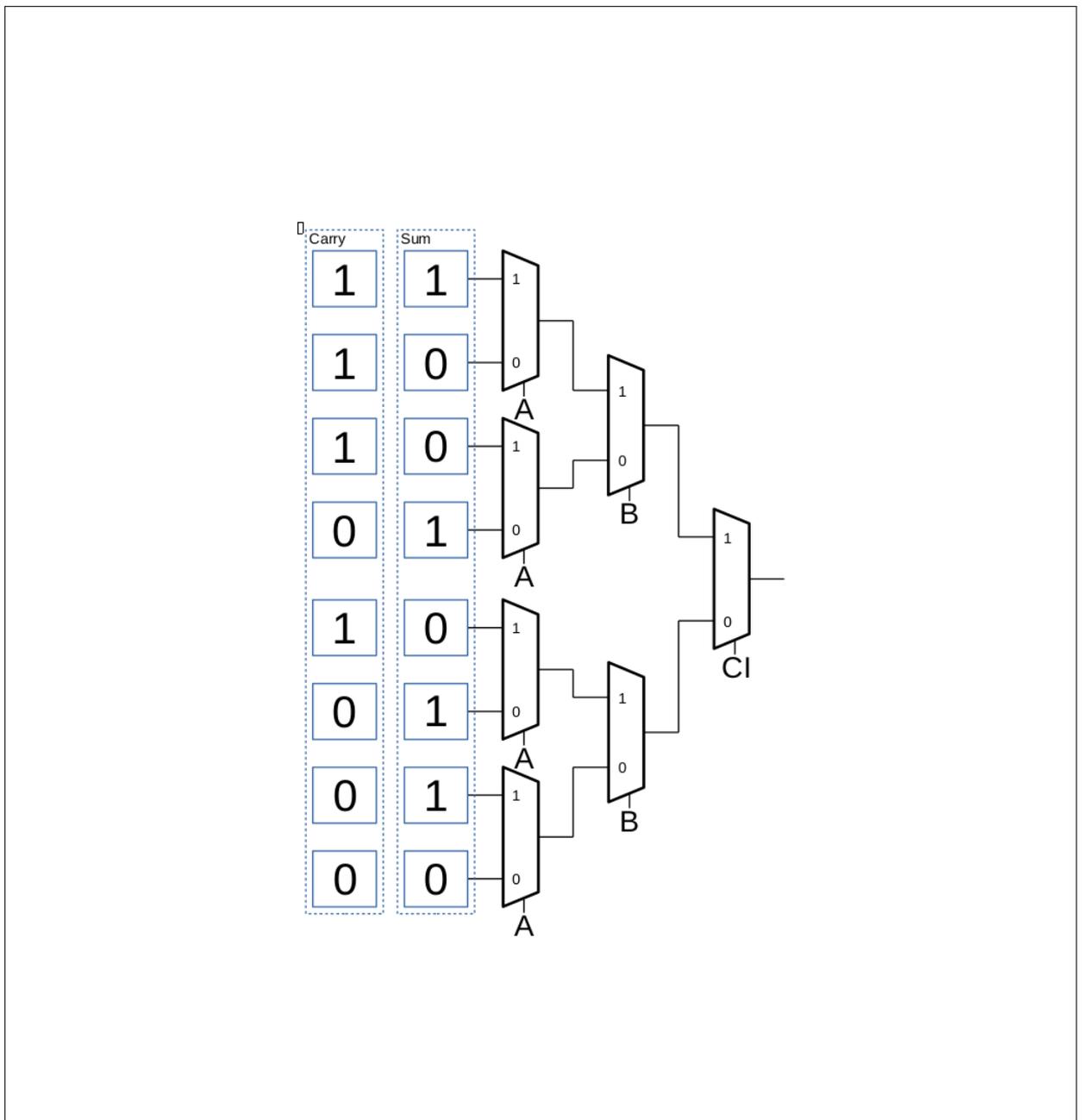


Aufgabe 4: Digitale Schaltungsentwicklung (15 Punkte)

$$CO = (A \wedge B) \vee (CI \wedge A) \vee (CI \wedge B)$$

$$SUM = A \oplus B \oplus CI \quad \text{mit } \oplus \text{ bezeichnet die XOR - Funktion}$$

1. Entwerfen Sie die Schaltung für oben dargestellten Funktionen als Baumstruktur von 2:1 Multiplexern. (10 Punkte)



Boolean Algebra and Optimization

(15 Points)

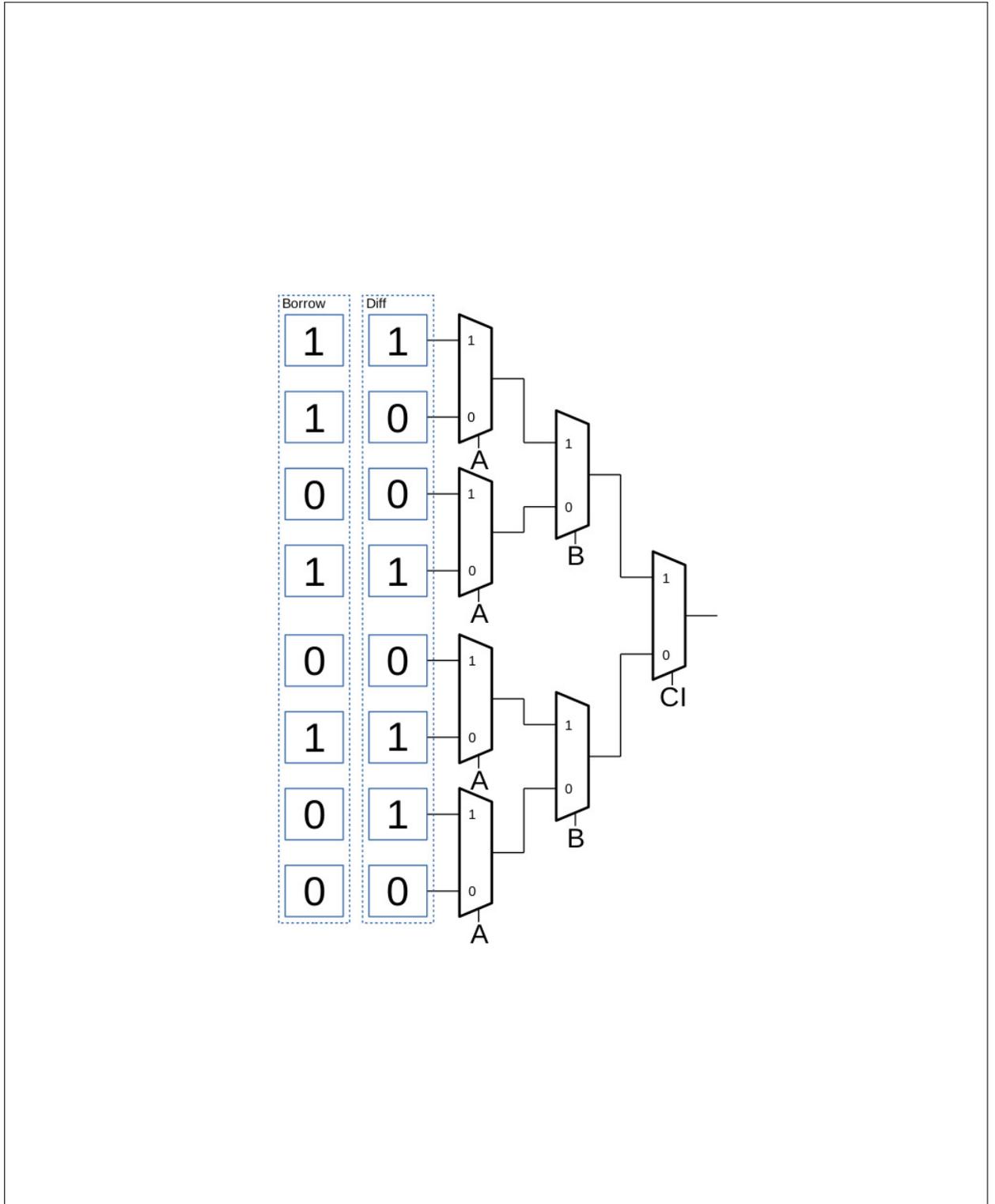
$$CO = (A \wedge B) \vee (CI \wedge A) \vee (CI \wedge B)$$

$$SUM = A \oplus B \oplus CI \quad \text{mit } \oplus \text{ bezeichnet die XOR – Funktion}$$

1. Develop a digital circuit for the given functions as a tree of 2:1 multiplexers. (10 Punkte)



2. Entwerfen Sie für die Schaltung aus Aufgabenteil 1 das Programm für einen (5 Punkte) Subtrahierer.



2. For the circuit in part 1, develop a program for a subtractor.

(5 Punkte)

