Allgemeiner Hinweis: Die Übungsgruppenanmeldung in TISS läuft von Donnerstag, 02.11., 20:00 Uhr bis Sonntag, 19.11., 13:00 Uhr. Die Platzvergabe in den einzelnen Übungsgruppen erfolgt nach dem *first come, first served*-Prinzip. Wir können Ihnen keinen Platz in einer bestimmten Übungsgruppe garantieren. Wir haben aber sichergestellt, dass für alle Studierende ein Platz in einer Übungsgruppe zur Verfügung steht, die zumindest 8 anerkannte Beispiele bei den Übungsblättern 1 und 2 haben.

Aufgabe 1: Zweierkomplement, Multiplikation

Es sind die folgenden Zahlen gegeben:

$$A = (-9)_{10}$$
$$B = (-4)_{10}$$
$$C = (23)_{10}$$

Führen Sie die nachfolgenden Berechnungen mit einer Maschinenwortlänge von 8 Bit in Zweierkomplementdarstellung durch und geben Sie die Ergebnisse auch als decodierte Dezimalzahl an:

a) $A * B$

b) $B * C$

c) $A * C$

Aufgabe 2: Exzessdarstellung

Gegeben sind zwei Zahlen. Die Zahlen sind in Exzessdarstellung mit dem asymmetrischen Exzess $e = (105)_{10}$ codiert.

a) Bestimmen Sie den Summanden B der Rechnung $A + B = C$ und geben Sie den Wert von B binär (in der oben genannten Exzessdarstellung) und dezimal an.

$$\begin{aligned} A &= 10110100 \\ C &= 11110010 \end{aligned}$$

b) Berechnen Sie die Summe $A + B = C$ in Exzessdarstellung und geben Sie den Wert von C binär (in der oben genannten Exzessdarstellung) und dezimal an.

$$\begin{aligned} A &= 10000000 \\ B &= 1010000 \end{aligned}$$

Aufgabe 3: IEEE verkürzt mit Sonderfällen

Gegeben sind die folgenden im 16-Bit-Gleitpunktformat $\mathbb{F}(2, 11, -14, 15, \text{true})$, welches Sie bereits aus Übungsblatt 2 (Aufgabe 1) kennen, codierten Zahlen:

$$
\begin{aligned}
A &= 1\,00000\,0010100000 \\
B &= 1\,01010\,1101100000 \\
C &= 0\,11111\,0000000000 \\
D &= 0\,11100\,0000000000
\end{aligned}
$$

Führen Sie mit den Zahlen folgende Berechnungen durch und codieren Sie das Ergebnis jeweils im angegebenen Gleitpunktformat. Runden Sie mittels *round to even*.

a) $A + B$

b) $A * C$

c) D/A

Aufgabe 4: Paritätsbits

Ein Code, dessen Codewörter eine Länge von 5 Bit haben, soll folgenermaßen störsicherer gemacht werden: Jeweils fünf Codewörter werden in Form einer Matrix angeordnet. Dann wird zeilen- und spaltenweise ein Parity-Bit (even parity) berechnet, wobei auch über die Parity-Bit-Spalte ein Parity-Bit bestimmt wird.

a) Kodieren Sie 00010, 11011, 00000, 10001, 01011.

b) Geben Sie die korrigierten Codewörter an.

$$\begin{matrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \end{matrix}$$

c) Wie viele Bits kann diese Methode in jeweils fünf Codewörtern korrigieren?

d) (**Bonus-Aufgabe**, muss nicht gelöst werden, gibt aber bei richtiger Präsentation in der UE-Gruppe ein Mitarbeitsplus.)
Warum ist das Parity-Bit, das spaltenweise über die Parity-Bit-Spalte berechnet wird, auch ein korrektes Parity-Bit für die Parity-Bit-Zeile?

Aufgabe 5: Mittlerer Informationsgehalt, mittlere Wortlänge und Redundanz eines Codes

Gegeben sei folgender Code für das Alphabet $\{ r \mid s \mid t \mid u \mid v \}$. Die Tabelle enthält neben den Codewörtern auch die Auftrittswahrscheinlichkeit und den Informationsgehalt der einzelnen Zeichen des Quellalphabets.

	p_i	h_i	Codewort
r	0.40	1.32	01
s	0.25	2.00	1
t	0.20	2.32	001
u	0.10	3.32	0001
v	0.05	4.32	0000

a) Berechnen Sie

$H =$

$L =$

$R =$

b) Ist der Code optimal? Begründen Sie Ihre Antwort.

Aufgabe 6: Huffman-Code

a) Gegeben sei ein Alphabet bestehend aus den Zeichen $\{\, a \mid b \mid c \mid d \mid e \mid f \,\}$. Betrachten Sie die nachfolgende Tabelle mit den jeweiligen Auftrittswahrscheinlichkeiten der einzelnen Zeichen und füllen Sie die Tabelle vollständig aus. Erstellen Sie dazu einen passenden Huffman-Codebaum und geben Sie zusätzlich die mittlere Wortlänge L an.

	p	Code	l	p · l
a	0.3			
b	0.05			
c	0.13			
d	0.08			
e	0.25			
f	0.19			

b) Gegeben ist folgender Code.

	p	h	h · p	Code	l	p · l
a	0.8	0.32	0.256	1	1	0.8
b	0.18	2.47	0.4446	10	2	0.36
c	0.02	5.64	0.1128	00	2	0.04

(i) Berechnen Sie die Redundanz.

(ii) Um die Redundanz zu verringern, werden zwei aufeinanderfolgende Zeichen miteinander codiert. Füllen Sie dazu folgende Tabelle aus.

	p	Code	l	p · l	h · p
aa		1			0.412
ab		00			0.4026
ac		01010			0.095
ba		110			0.4
bb		0010			0.16
bc		0011010			0.029
ca		111010			0.095
cb		11011010			0.029
cc		01011010			0.0045

(iii) Geben Sie R und die mittlere Wortlänge L an.

Aufgabe 7: Boole'sche Algebra – Gleichheit von zwei Ausdrücken mit Wahrheitstabelle überprüfen

Überprüfen Sie die nachfolgenden booleschen Ausdrücke F_1 und F_2 mittels Wahrheitstabelle auf logische Äquivalenz. Begründen Sie Ihre Antwort.

Hinweis: $\uparrow$ entspricht NAND, $\downarrow$ entspricht NOR, $\oplus$ entspricht XOR.

Hinweis: Sie können die resultierenden Wahrheitswerte eines binären Operators direkt unter diesen Operator in der Wahrheitstabelle schreiben.

a) $F_1 = (a \oplus b) \Rightarrow (a \uparrow d)$
$F_2 = (d \Rightarrow (a \wedge b)) \vee (b \equiv d) \vee (a \downarrow b)$

Anmerkung: F_2 ist nicht vollständig geklammert. Es gibt zwei Möglichkeiten Klammern einzufügen um den Ausdruck vollständig zu klammern. Überlegen Sie sich ob es einen Unterschied macht, wenn eine dieser beiden Varianten als F_2 für die Überprüfung der Äquivalenz verwendet wird.

a	b	c	d	$(a \oplus b)$	$\Rightarrow$	$(a \uparrow d)$	$\equiv$	$(d \Rightarrow (a \wedge b))$	$\vee$	$(b \equiv d)$	$\vee$	$(a \downarrow b)$
0	0	0	0									
1	0	0	0									
0	1	0	0									
1	1	0	0									
0	0	1	0									
1	0	1	0									
0	1	1	0									
1	1	1	0									
0	0	0	1									
1	0	0	1									
0	1	0	1									
1	1	0	1									
0	0	1	1									
1	0	1	1									
0	1	1	1									
1	1	1	1									

Begründung:

b) $F_1 = (a \oplus b) \Rightarrow (a \uparrow d)$
$F_2 = (d \Rightarrow (a \wedge b)) \vee (b \equiv d) \vee \neg(c \Rightarrow d)$

a	b	c	d	$(a \oplus b)$	$\Rightarrow$	$(a \uparrow d)$	$\equiv$	$(d \Rightarrow (a \wedge b))$	$\vee$	$(b \equiv d)$	$\vee$	$\neg(c \Rightarrow d)$
0	0	0	0									
1	0	0	0									
0	1	0	0									
1	1	0	0									
0	0	1	0									
1	0	1	0									
0	1	1	0									
1	1	1	0									
0	0	0	1									
1	0	0	1									
0	1	0	1									
1	1	0	1									
0	0	1	1									
1	0	1	1									
0	1	1	1									
1	1	1	1									

Begründung:

Aufgabe 8: Boole'sche Algebra – Ausdruck mittels Gesetzen umformen

Überprüfen Sie folgende boolesche Ausdrücke durch algebraisches Umformen auf Äquivalenz.

$F_1 = ((c \equiv d) \downarrow (\neg c \oplus b)) \vee a$
$F_2 = (((c \supset d) \wedge (d \supset c)) \vee ((c \supset b) \wedge (b \supset c))) \supset a$

Hinweise: $\downarrow$ entspricht NOR, $\oplus$ entspricht XOR, $\supset$ entspricht $\Rightarrow$. Foliensatz 5, Folie 12 kann hilfreich sein.