Rechnerorganisation

Mathematische Grundlagen (1)

Boolesche Algebren: BMA, BAA (2,3)

Kombinatorische Schaltungen (4,5)

Automaten (6,7)

Sequentielle Schaltungen (8)

Programmierbare Strukturen (9)

Rechneraufbau und ~funktion (10,11)

Informationskodierung (12,**13**,14)

Prüfung Rechnerorganisation

Freitag, 06. März 2020

11:00 – 12:30, 90 Minuten

Audimax

Erlaubt sind:

- Arbeitsblätter,
- Unbeschriebene Notizblätter

Verboten sind:

- Mitschriften
- Taschenrechner
- Handys
- Nachbars Hilfe

Hinweis zum Praktikum RO

Praktikum Rechnerorganisation

Name	Vorname	Matrikel-Nr.	Studiengang

Rechnerorganisation	Datum	Ergebnis	Unterschrift des
Versuch 1: Digitale Grundschaltungen			
Versuch 2: Emulator 8086			
Informationen im Internet: http://www.tu-ilmenau.de/iks/lehre/bachelor-studiengae ecture_id=40			

Im Sommersemester (April, Mai) möglich

Zum Abrechnen melden Sie das Praktikum bitte zunächst im Thoska-System an!
Geben Sie **anschließend** die Testatkarte im **Raum Z 1031** ab (Sekretariat IKS).

Diese Testatkarte ist für den Bachelorstudiengang IN ab der PO-Version **2013** gültig.

Für die Studiengänge BT, und II ist diese Testatkarte **nicht** geeignet.

Hinweis zum Praktikum TI

TECHNISCHE UNIVERSITÄT ILMENAU

Testatkarte

Fakultät für Informatik und Automatisierung

Praktikum Technische Informatik

Name ______ Vorname ______ Matrikel-Nr. ______ Studiengang ______

Technische Informatik 1	Datum	Ergebnis	Unterschrift des Verantwortlichen
Versuch 1			
Versuch 2			
Informationen im Internet: http://tu-ilmenau.de/iks/lehre/bachelor-studiengaenge/			

Technische Informatik 2	Datum	Ergebnis	
Versuch 1			
Versuch 2			
Informationen im Internet: http://tu-ilmenau.de/?r-p-ti2			

Im Sommersemester (April, Mai) möglich

ERGEBNIS	Unterschrift des Verantwortlichen	Eintragung im Online-System (Datum)

Zum Abrechnen melden Sie das Praktikum bitte zunächst im Thoska-System an!

Geben Sie dann die Testatkarte im Raum Z 2055 ab (Sekretariat RAuES).

Diese Testatkarte ist für die Bachelorstudiengänge BT und II ab der PO-Version 2013 sowie für den Diplomstudiengang MB gedacht.

Für andere Studiengänge ist diese Testatkarte nicht geeignet.

Datenkodierung

Daten

alphanumerische Zeichen Zahlen

- **ASCII-Kode:**

- Vb: α-numerische Zeichen (in der Tabelle: Spalte α-n)
- Nb: $N_8 \ldots$ Menge von Bytes (dargestellt als Hexadezimalzahlen $(ASC)_{16}$ bzw. Dezimalzahlen $(ASC)_{10}$)
- Kodiervorschrift: siehe nachfolgende Tabelle

Zeichenkodierung - ASCII

Buchstaben

$(ASC)_{16}$	$(ASC)_{10}$	α-n	$(ASC)_{16}$	$(ASC)_{10}$	α-n
40	64	@	60	96	‘
41	65	A	61	97	a
42	66	B	62	98	b
43	67	C	63	99	c
44	68	D	64	100	d
45	69	E	65	101	e
46	70	F	66	102	f
47	71	G	67	103	g

Zeichenkodierung - ASCII

Ziffern

30	48	0
31	49	1
32	50	2
33	51	3
34	52	4
35	53	5
36	54	6
37	55	7
38	56	8
39	57	9

Zeichenkodierung - ASCII

Sonderzeichen

$(ASC)_{16}$	$(ASC)_{10}$	α-n
20	32	SP
21	33	!
22	34	”
23	35	#
24	36	$
25	37	%
26	38	&
27	39	’
28	40	(

Zeichenkodierung - ASCII

Steuerzeichen

$(ASC)_{16}$	$(ASC)_{10}$	α-n
00	00	NUL
01	01	SOH
02	02	STX
03	03	ETX
04	04	EOT
05	05	ENQ
06	06	ACK
07	07	BEL
08	08	BS
09	09	HT
0A	10	LF

Zeichenkodierung - Unicode

- internationaler Standard
- für jedes sinntragende Schriftzeichen oder Textelement aller bekannten Schriftkulturen und Zeichensysteme ein digitaler Code festgelegt
- wird ständig um Zeichen weiterer Schriftsysteme ergänzt
- ursprünglich mit 16 Bit definiert (2^{16} = 65.536 Elemente)
- ab Unicode 2.0 (Juli 1996) auf 17 Unicode-Blöcke zu je 65.534 Elementen definiert (insgesamt 1.114.112 Codepunkte)
- kodiert in UTF-8 U+0000 bis U+10FFFF
  - *8-Bit UCS Transformation Format*,
  - wobei UCS wiederum *Universal Character Set*

www.wikipedia.de

Zeichenkodierung - Unicode

Ebene und Bezeichnung	von	bis	belegt	nicht belegt	definiert	undefiniert
Ebene 0: **BMP** (Basic Multilingual Plane)	00000	0FFFD	65 310	224	61 456	3 854
Ebene 1: **SMP** (Supplementary Multilingual Plane)	10000	1FFFD	11 984	51 550	9 991	1 993
Ebene 2: **SIP** (Supplementary Ideographic Plane)	20000	2FFFD	47 648	17 886	47 624	24
Ebene 3: **TIP** (Tertiary Ideographic Plane)	30000	3FFFD	0	65 534	0	0
Ebene 4: *noch nicht belegt*	40000	4FFFD	0	65 534	0	0
Ebene						0
Ebene						0
Ebene						0
Ebene						0
Ebene						0
Ebene						0
Ebene 11: *noch nicht belegt*	B0000	BFFFD	0	65 534	0	0
Ebene 12: *noch nicht belegt*	C0000	CFFFD	0	65 534	0	0
Ebene 13: *noch nicht belegt*	D0000	DFFFD	0	65 534	0	0
Ebene 14: **SSP** (Supplementary Special-purpose Plane)	E0000	EFFFD	368	65 166	337	31
Ebene 15: **PUA** (reserved for the Private Use Area)	F0000	FFFFD	65 534	0	65 534	0
Ebene 16: **PUA** (reserved for the Private Use Area)	100000	10FFFD	65 534	0	65 534	0
Summen der Ebenen 0 bis 16	000000	10FFFD	253 466	860 612	240 517	12 949

- 17 Ebenen (0 .. 10H)
- je 2^{16} = 65.536 mögliche Codierungen (0000 .. FFFFH)
- FFFEH und FFFFH nicht für die Kodierung benutzt

- ergibt 17 * 65.534 = **1.114.078 mögliche Zeichen (Codepoints)**

Datenkodierung

Daten

alphanumerische Zeichen

Zahlen

- BCD
- vorzeichenbehaftete Zahlen
- 2K-Zahlen
- Gleitkomma-Zahlen

Zahlenkodierung - BCD

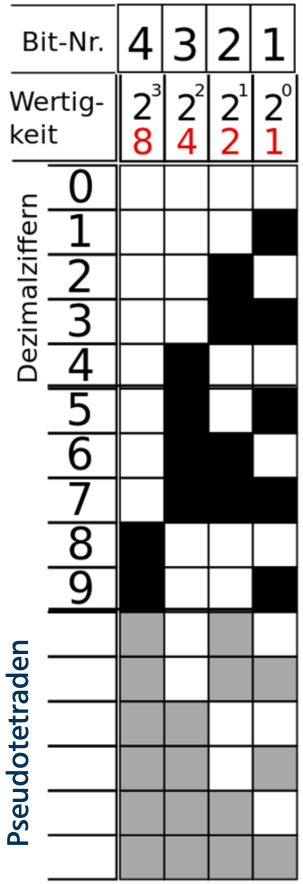
BCD – Binary Coded Decimals (siehe Arbeitsblätter S. 29)

Tetraden | Pseudotetraden

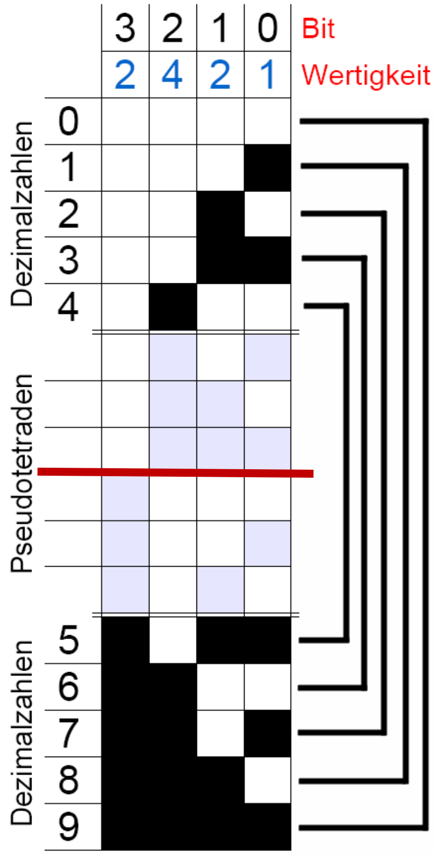
z_i (i ... Dezimalziffernwert)				t_j
BCD (direkt)	Aiken	3xS	Gray	$2^3 2^2 2^1 2^0$
0	0	-	0	0 0 0 0
1	1	-	1	0 0 0 1
2	2	-	3	0 0 1 0
3	3	0	2	0 0 1 1
4	4	1	7	0 1 0 0
5	-	2	6	0 1 0 1

Zahlenkodierung - BCD

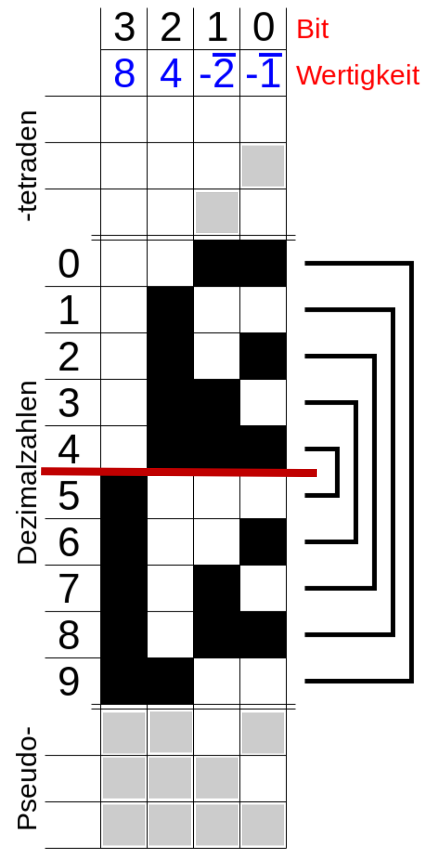
direkter BCD Code



Aiken Code



3xS Code

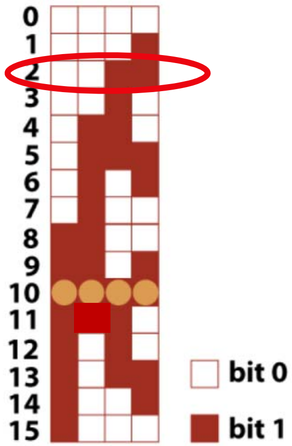


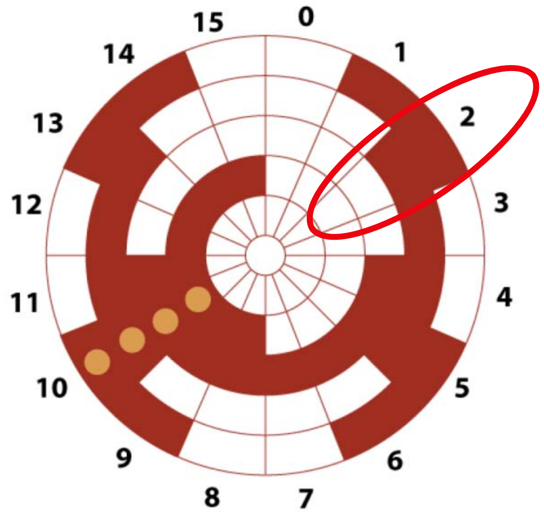
Zahlenkodierung – Gray-Code

Dezimalzähler: Kodierung wie in Arbeitsblättern

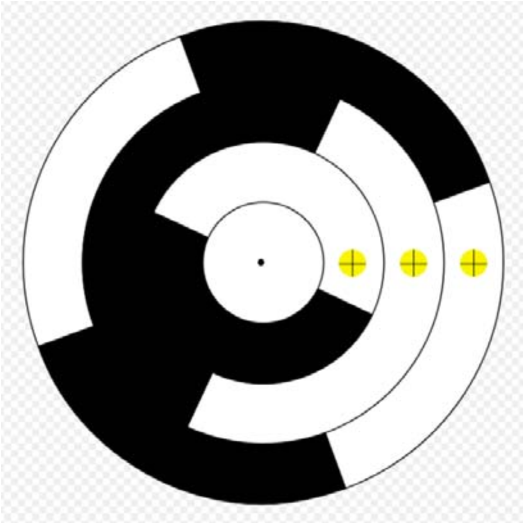
Bilder: Hexadezimal-Zähler, (keine Pseudotetraden)

z_i (i . . . Dezimalziffernwert)				t_j
BCD (direkt)	Aiken	3xS	Gray	$2^3 2^2 2^1 2^0$
			0	0 0 0 0
			1	0 0 0 1
			3	0 0 1 0
			2	0 0 1 1
			7	0 1 0 0
			6	0 1 0 1

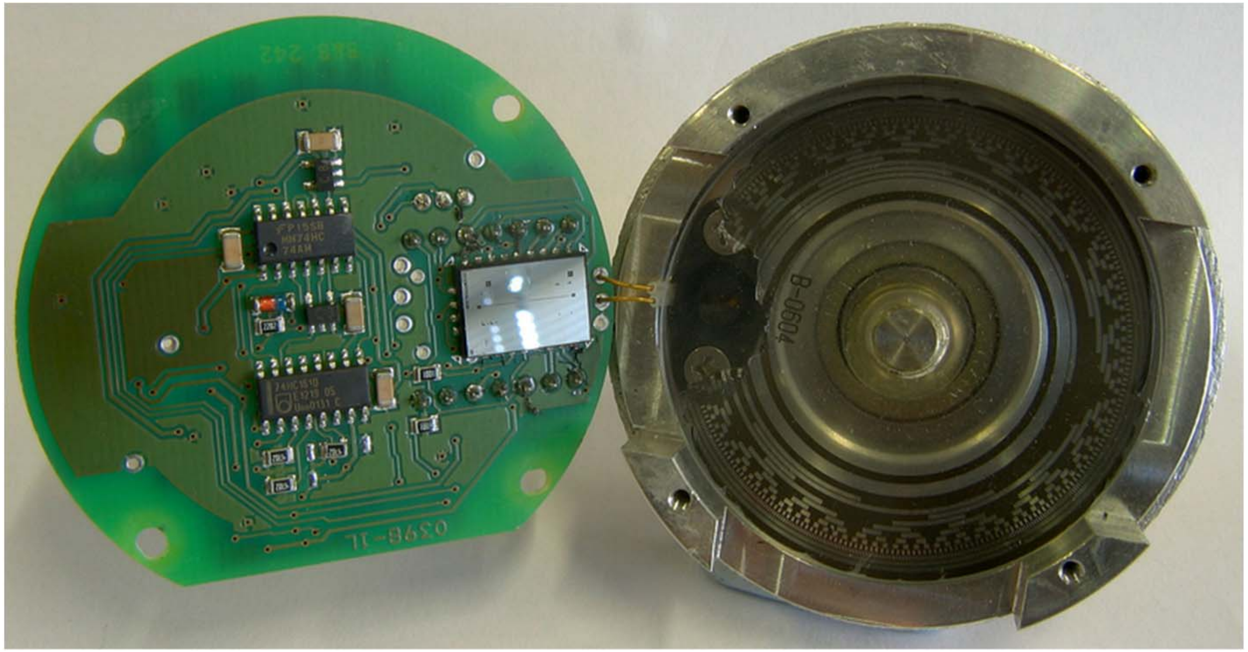




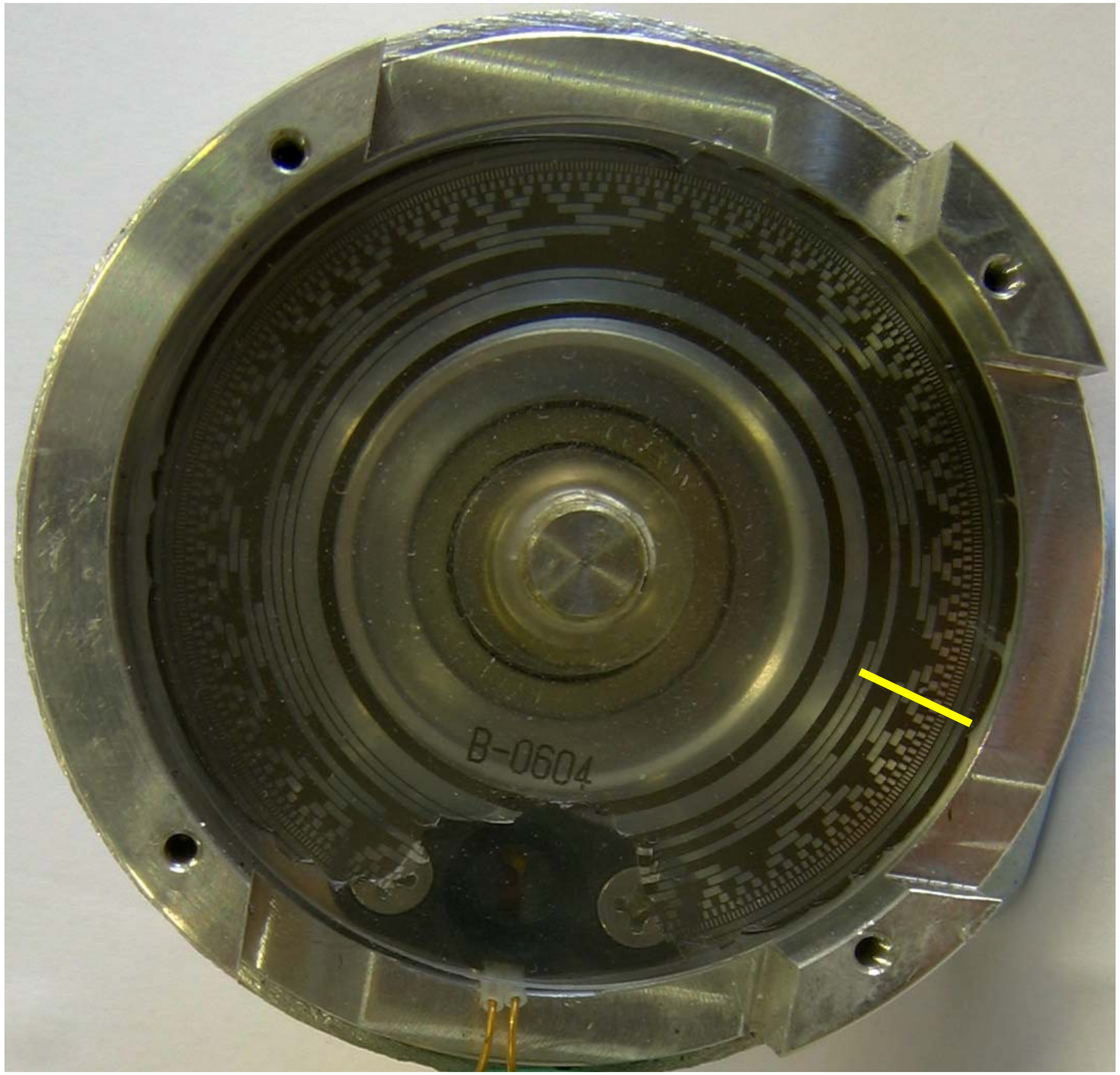
Zahlenkodierung - Gray-Code



Ein Gray-Code Absolutwertgeber mit 13 bits



Zahlenkodierung - Gray-Code



13 bit
=
8192 Inkremente

Zahlenkodierung - BCD

Operationen

Addition + Korrektur

- bei Addition $\mathcal{K}(z_i + z_i') = \mathcal{K}(z_i) + \mathcal{K}(z_i') + \mathbf{C}$
- bei Subtraktion $\mathcal{K}(z_i - z_i') = \mathcal{K}(z_i) - \mathcal{K}(z_i') - \mathbf{C}$

	BCD	Aiken	3xS
Korrektur erforderlich	bei **ü** oder **p**	nur bei **p**, dann abhängig von **ü**	immer, aber abhängig von **ü**
Konstante C	0110	−0110 bei **ü** +0110 sonst	+0011 bei **ü** −0011 sonst

Datenkodierung

Daten

alphanumerische Zeichen

Zahlen

- BCD
- vorzeichenbehaftete Zahlen
- 2K-Zahlen
- Gleitkomma-Zahlen

Datenkodierung

Daten

alphanumerische Zeichen

Zahlen

- BCD
- vorzeichenbehaftete Zahlen
- 2K-Zahlen
- Gleitkomma-Zahlen

Zahlenkodierung – 2K-Zahlen

Konegative Zahlen (siehe Arbeitsblätter S. 30)

Ergänzung zu 2^n bzw. 2^n-1

- Vb: $-2^{n-1} < z < 2^{n-1}$
- Nb: N_n (üblich: N_8, N_{16}, N_{32})
- KV:

$$\mathcal{K} = (z) = \begin{cases} z_n \leftrightarrow z \geq 0 \\ \overline{z_n} \text{ sonst} \end{cases}$$

z_n . . . n-stellige Dualzahl von z

$\overline{z_n}$ . . . Komplement der Dualzahl

Komplementbildung

Man unterscheidet Einer- und Zweierkomplemente und dementsprechend als konegative Zahlen 1K- und 2K-Zahlen

- 1K-Zahlen: $\overline{z_n} = 2^n - 1 - z_n$
- 2K-Zahlen: $\overline{z_n} = 2^n - z_n$

$z_{n1} + \overline{z_{n1}} = 2^n$

Zahlenkodierung – 2K-Zahlen

- ## Einer-Komplement (1K):

Der Name leitet sich aus der Tatsache ab, daß die Zahl von lauter "Einsen" subtrahiert wird.

1111	2^4 -1
- 0101	- 5
1010	Einer-Komplement (bitweise Negation) der Zahl -5

- ## **Zweier-Komplement (2K):**

Der Name entstammt der Überlegung, dass eine positive Zahl mit n Bits in ihr negatives Pendant umgewandelt werden kann, indem die Zahl von der "Zweier"-Potenz 2^n abgezogen wird.

10000	2^4
- 0101	- 5
1011	Zweier-Komplement der Zahl -5 (entspricht der 1K-Zahl +1)

$$z_{n1} + \overline{z_{n1}} = 2^n$$

Zahlenkodierung – 2K-Zahlen

Bildung der 2K-Zahlen

(a) Subtraktion von 2^n: $\overline{z_n} = 2^n - z_n$

(b) 1K-Zahl (Negation) + 1: $\overline{z_n} = 2^n - 1 - z_n + 1$

(c) beginnend von rechts die erste 1 suchen,

diese bleibt stehen,

alle Ziffern links davon invertieren

Zahlenkodierung – 2K-Zahlen

Operationen (siehe Arbeitsblätter S. 30)

$$z_{n1} > z_{n2}\ ,\ z_{n1} + z_{n2} = s_n\ ,\ z_{n1} - z_{n2} = d_n$$

Operation	1k-Zahlen Ergebnis	1k-Zahlen Korrektur	2K-Zahlen Ergebnis	2K-Zahlen Korrektur
$z_{n1} + z_{n2}$	s_n	-	$z_{n1} + z_{n2} = s_n$	-
$z_{n1} + \overline{z_{n2}}$	$d_n + 2^n - 1$	$-2^n + 1$	$z_{n1} - z_{n2} + 2^n = d_n + 2^n$	-2^n
$\overline{z_{n1}} + z_{n2}$	$\overline{d_n}$	-	$-(z_{n1} - z_{n2}) + 2^n = \overline{d_n}$	-
$\overline{z_{n1}} + \overline{z_{n2}}$	$\overline{s_n} + 2^n - 1$	$-2^n + 1$	$-(z_{n1} + z_{n2}) + 2^n + 2^n = \overline{s_n} + 2^n$	-2^n

$$\text{2K:}\quad z_{n1} + \overline{z_{n1}} = 2^n$$

Das war‘s für dieses Semester

Viel Erfolg bei der Prüfung!