

Übungsaufgaben zur Zahlendarstellung (Digitaltechnik Kapitel 4):

Die Aufgaben sind nur von Nutzen, wenn sie ohne Taschenrechner gelöst werden.

1) 8-stellige positive Dualzahlen:

Wandeln Sie die positiven Dezimalzahlen in eine 8-stellige Dualzahl um.

Die Binärzahl ist weiters in eine Hexedezimalzahl umzuwandeln.

dezimal	binär	hex
213	1101 0101	D5
255	1111 1111	FF
128	1000 0000	80
66	0100 0010	42
1	0000 0001	01
89	0101 1001	59
100	0110 0100	64
200	1100 1000	C8
50	0011 0010	32

2) 8-stellige Dualzahlen mit positivem und negativem Vorzeichen.

Negative Zahlen sind im Zweierkomplement darzustellen.

sonst, wie Punkt 1)

dezimal	binär	hex
10	0000 1010	0A
64	0100 0000	40
127	0111 1111	7F
-10	1111 0110	F6
-64	1100 0000	C0
-127	1000 0001	81
-56	1100 1000	C8
-20	1110 1100	EC

3) positive Zahlen: hier sind so viele Stellen, wie notwendig anzuschreiben

Mit Verwendung des Taschenrechners,

allerdings nur unter Verwendung der 4 Grundrechnungsarten

(keine automatischen Umwandlungen).

dezimal	binär	oktal	hex
10	1010	12	A
20	10100	24	14
188	10111100	274	BC
400	110010000	620	190
1809	111 0001 0001	3421	711
32767	111 1111 1111 1111	77777	7FFF
65535	1111 1111 1111 1111	177777	FFFF
1964	111 1010 1100	3654	7AC

4) Rechnen mit Binärzahlen (rechnen Sie ohne Taschenrechner):

(Rechenregeln: $0+0=0$; $0+1=1$; $1+1=10$; $1+1+1=11$; $1+1+1+1=100$; ...)

$$\begin{array}{r}
 1110\ 1100\ (236) \\
 +\ 1100\ 1010\ (202) \\
 \hline
 1\ 1011\ 0110\ (438)
 \end{array}
 \qquad
 \begin{array}{r}
 1110\ 1100\ (236) \\
 -\ 1100\ 1010\ (202) \\
 \hline
 0010\ 0010\ (34)
 \end{array}$$

$$\begin{array}{r}
 1110\ 1100\ (236) \\
 +\ 0001\ 0111\ (23) \\
 +\ 1100\ 1010\ (202) \\
 \hline
 1\ 1100\ 1101\ (461)
 \end{array}
 \qquad
 \begin{array}{r}
 0110\ 0111\ (103) \\
 +\ 0000\ 0011\ (3) \\
 +\ 0100\ 1011\ (75) \\
 \hline
 1011\ 0101\ (181)
 \end{array}$$

5) logische Verknüpfung von Bytes

Ähnliche Funktionen kommen in vielen Anwendungen der Datenverarbeitung vor: Mikroprozessoren, C++, in PCs bei der Verarbeitung von Netzwerkadressen, ...

UND	1010 1100 <u>1100 1010</u> 1000 1000	ODER	1010 1100 <u>1100 1010</u> 1110 1110
EXOR	1010 1100 <u>1100 1010</u> 0110 0110	NAND	1010 1100 <u>1100 1010</u> 0111 0111
UND	1100 1100 <u>1111 0000</u> 1100 0000	ODER	1100 1100 <u>1111 0000</u> 1111 1100
EXOR	1100 1100 <u>1111 0000</u> 0011 1100	NOR	1100 1100 <u>1111 0000</u> 0000 0011

6) Rechnen mit C++

Beachte: 0 gilt als "falsch" ("false", "logisch 0")

alle anderen ganzen Zahlen gelten als "wahr" ("true", "logisch 1")

logische Verknüpfung von Bytes (oder Worten): | (ODER), & (UND), ~ (Negation)

logische Verknüpfung von "wahr" und "falsch": || (ODER), && (UND), ! (Negation)

Negative Zahlen werden im 2erKomplement dargestellt.

Hexadezimale Zahlen werden mit 0x... dargestellt, die Darstellung 0b... für binäre Zahlen gibt es nur bei bestimmten µP-Programmen.

C++ Ausdruck	Wert des Ausdrucks	
(2&4)	0	
(2 4)	6	
(0b0000 0010 & 0b0000 0100)	0	
(0b0000 0010 0b0000 0100)	6	
(15&15)	15	
(1 0)	1	WAHR
(4 0)	1	WAHR
(0x0F & 0x3F)	15	
((17+4) (7==8))	1	WAHR
((17+4)&& (7==8))	0	FALSCH
((17+4) (7==8))	21	
((17+4)& (7==8))	0	
((-1) (7==8))	1	WAHR
(~3)	-4	
(0xC1)	193	
(~0xC1)	-194	
(!1)	0	FALSCH
(!5)	0	FALSCH
(!(3==4) && (4>3))	1	WAHR
(!(((0x0C+3) & (0x03)) > 5) && (0x0A==10))	1	WAHR