

Rückblickübungen zu Zahlensysteme und Grundoperationen

Ziel: Die in Unterrichtsblock 01 'Zahlensysteme' und Unterrichtsblock 02 'Arithmetische und logische Grundoperationen' erarbeiteten Grundlagen dürfen Sie mit den folgenden Rückblickübungen vertiefen und damit festigen!

Hilfen: Ausgeteilte Unterlagen, Teams, OneNote, WorkBench, GitLab unter <https://infmod.gitlab.io/m114>

Fragen bzw. Übungen

1. Was versteht man unter codieren und was unter decodieren?

Eine **Codierung** ist eine eindeutige Zuordnung bzw. Verschlüsselung der Zeichen eines Zeichenvorrats wie z.B. wird eine Dezimalzahl eben mit einer Hexadezimalzahl dargestellt! Die Qualität dieser Codierung ist sehr schwach, da eine **Decodierung** in diesem Fall sehr einfach ist, d.h. dass die Umwandlung von der Hexadezimalzahl in die Dezimalzahl einfach möglich ist.

2. Nennen Sie die drei wichtigsten Arten von Logikgattern und schreiben Sie jeweils die Wertetabelle mit minimal möglichen Eingängen auf!

Die drei wichtigsten Arten von Logikgattern sind **AND**, **OR** und **NOT**. Ein **NOT**-Logikgatter hat nur einen Eingang wie z.B., bei der Wertetabelle rechts 'a' und einen Ausgang 's'. Ein **AND**- und **OR**-Logikgatter hat mindestens zwei Eingänge wie z.B. bei der Wertetabelle rechts 'a' und 'b' und dem Fall entsprechenden Status am AND-Ausgang 't' bzw. OR-Ausgang 'u'!

Fall:	b	a	s	t	u
1	0	0	1	0	0
2	0	1	0	0	1
3	1	0	1	0	1
4	1	1	0	1	1

3. Welche Zusatzverknüpfungen können aus den bei Aufgabe 2 genannten drei wichtigsten Arten von Logikgattern definiert werden? Nennen und beschreiben Sie mindestens vier Stück!

Mit den Grundverknüpfungen vom AND, OR und NOT der letzten Aufgabe 2 können alle beliebigen Verknüpfungen hergestellt werden wie beispielsweise **NAND**, **NOR**, **EXOR**, **EXNOR**!

Ein **NAND**-Glieder ist dabei ein invertiertes AND-Glied (d.h. AND-Glied mit angehängtem NOT-Glied) und hat bei zwei Eingängen die rechts definierten Ausgangsstatuswerte 'v'.

Fall:	b	a	v	w	x	y
1	0	0	1	1	0	1
2	0	1	1	0	1	0
3	1	0	1	0	1	0
4	1	1	0	0	0	1

Ein **NOR**-Glieder ist dementsprechend ein invertiertes OR-Glied (d.h. OR-Glied mit angehängtem NOT-Glied) und hat die Ausgangsstatuswerte 'w' in der Wertetabelle rechts.

Ein **EXOR** ist ein exklusiv OR-Glied, welches dann am Ausgang den Status 1 hat, wenn nur ein Eingang den Status 1 hat, wie dies das Symbol '=1' und der Ausgang 'x' bei der oben definierten Wertetabelle repräsentiert.

Ein **EXNOR** hat dann am Ausgang den Status 1, wenn alle Eingänge den gleichen Status '1' oder '0' haben, wie dies das Symbol '=' und die Variable 'y' in der oben definierten Wertetabelle klar repräsentiert.

4. Was versteht man bei einem Zahlensystem unter Basis? Nennen Sie zudem die Basis einer Dezimal, Dual-, Oktal- und Hexadezimalzahl!

Die **Basis**-Zahl repräsentiert die Anzahl der im entsprechenden Zahlensystem definierter, verschiedener Ziffern. So sind im **Dualsystem** nur die Zahlen 0 und 1, während im **Dezimalsystem**, wie dies die Menge $ID = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0\}$ klar zeigt, 10 verschiedene Ziffern, im **Oktalsystem** 8 Ziffern und im **Hexadezimalsystem** 16 verschiedene Ziffern wie $IH = \{0; 1; 2; 3; 4; 5; 6; 7; 8; 9; A; B; C; D; E; F\}$ definiert sind. Die Anzahl verschiedener Ziffern in einem Zahlensystem definiert auch gleich dessen Basis, d.h. die Anzahl der Ziffern ist die Basiszahl.

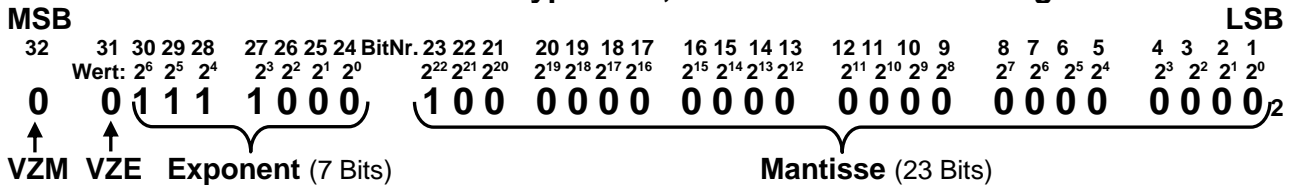
Dez	Hex	Oktal	Dual
0	0	0	0 0 0 0
1	1	1	0 0 0 1
2	2	2	0 0 1 0
3	3	3	0 0 1 1
4	4	4	0 1 0 0
5	5	5	0 1 0 1
6	6	6	0 1 1 0
7	7	7	0 1 1 1
8	8	10	1 0 0 0
9	9	11	1 0 0 1
10	A	12	1 0 1 0
11	B	13	1 0 1 1
12	C	14	1 1 0 0
13	D	15	1 1 0 1
14	E	16	1 1 1 0
15	F	17	1 1 1 1
16	10	20	1 0 0 0 0

5. Warum verwendet man Hexadezimalzahlen bei Computersystemen?

Bei modernen Computern ist eine Speicherplatzadresse z.B. 128 Bits lang. Dank der **Hexadezimalzahlen** wird diese Anzahl Ziffern der Adresse um ein 4-faches kürzer, wie dies die obere Wertetabelle klar zeigt.

6. Wie unterscheiden Sie reelle Zahlen von ganzen Zahlen?

Reelle Zahlen sind z.B. vom Datentyp REAL, der mit meist 32 Bits folgendes darstellt:



Das oben definierte Bitmusterbeispiel zeigt:

- Das MSB ist das Vorzeichenbit der Mantisse. Wenn es den Status 0 hat, ist diese reelle Zahl positiv, bei Status 1 negativ.
- Die Definition vom Exponent wird mit 8 Bit definiert wie z.B. $0111\ 1000_2 = +120$. Das BitNr 31, d.h. das VZE wird das Vorzeichen des Exponenten definiert. Damit können Exponentenwerte von -128 bis 127 definiert werden!
- Die Definition der Mantisse umfasst 23 Bits. Das '0.' vor dieser Mantisse wird dann vom Computer voran angelegt. Da beim oben definierten Beispiel das Bit mit der Wertigkeit 2^{22} mit dem Ziffernwert $2^{-1} = 0.5$ den Status "1" hat, beträgt diese reelle Zahl $0.5E120$.

Ganze Zahlen haben z.B. den Datentyp INTEGER. Eine Variable mit diesem Datentyp umfasst bei bestimmten C-Compilern 32 Bits. Das MSB, d.h. hier das Bit mit dem Stellenwert 2^{31} definiert bei diesem Datentype das Vorzeichen. Mit den restlichen 31 Bits können damit Zahlen von $-2^{31}-1$ bis 2^{31} dargestellt werden.

Aus der Definition dieser beiden Datentypen erkennen Sie, dass die Verarbeitung von reellen Zahlen viel komplexer und aufwendiger ist, als jene von ganzen Zahlen! Deshalb verwenden wir bei Variablen, denen nur ganzzahlige Werte zugewiesen werden, sicher nicht den Datentype REAL!

7. Nennen Sie mindestens vier verschiedene Datentypen und erklären Sie vorhandene Unterschiede! Bei der letzten Aufgabe 6 wurden bereits die Datentypen 'REAL' und 'INTEGER' erklärt. Weitere Datentypen sind:

- **BOOLEAN** definiert den Zustand einer Sache, d.h. ob diese 'wahr' bzw. 'falsch' ist. Wenn dabei z.B. der Booleanwert 'Wochenende' deklariert wurde, dann ist der Status dieser Booleanvariablen beim Wochentag Samstag bzw. Sonntag TRUE und bei den anderen Wochentagen FALSE.
- **CHAR** definiert einen Charakter, d.h. ein Zeichen von der meist erweiterten ASCII-Codetabelle und umfasst damit pro Zeichen 1 Byte.

8. Welche Stellenwerte haben die kursiv und unterstrichenen Ziffern der Dezimalzahl 342563245, der Hexadezimalzahl EBD9C1₁₆ und der Binärzahl 0111011010101₂?

- Dezimalzahl 342563245: 4 hat den Stellenwert 10^7 , 5 den Stellenwert 10^5 und 2 den Stellenwert 10^2 !
- Hexadezimalzahl EBD9C1₁₆ E hat den Stellenwert 16^5 , D den Stellenwert 16^3 und 1 den Stellenwert 16^0 !
- Binärzahl 0111011010101₂ Das unterstrichene 1 ganz links hat den Stellenwert 2^{10} , das nächste 2^6 und jenes recht 2^3 !

Bei den folgenden Aufgaben schreiben Sie klare, saubere und vollständige Lösungswege mit Ihrem Tablet bzw. auf Ihre Reinblätter oder Ihrem M114-Arbeitsheft!

9. Lösen Sie die folgenden Umrechnungen:

- $0100'1110_2 = 4E_{16} = 4 \cdot 16^1 + 14 \cdot 16^0 = 4 \cdot 16 + 14 = \underline{78}$
- $011'0111_2 = 37_{16} = 3 \cdot 16^1 + 7 \cdot 16^0 = 3 \cdot 16 + 7 = \underline{55}$
- $01'1100.11_2 = 0001'1100.1100_2 = 1C.C_{16} = 1 \cdot 16^1 + C \cdot 16^0 + C \cdot 16^{-1} = 16 + 12 + 0.75 = \underline{28.75}$
- $01'0111.011_2 = 0001'0111.0110_2 = 17.6_{16} = 1 \cdot 16^1 + 7 \cdot 16^0 + 6 \cdot 16^{-1} = 16 + 7 + 0.375 = \underline{23.375}$
- $1057 \text{ Mit Taschenrechnermethode DIV 16 und Rest } 421_{16} = \underline{0100'0010'0001_2}$
- $45673 \text{ Mit Taschenrechnermethode DIV 16 und Rest } B269_{16} = \underline{0'1010'0010'0110'1001_2}$
- $45.625 \text{ Mit Taschenrechnermethode DIV 16 und Rest } 2D.A_{16} = \underline{0'0010'1101.1010_2}$
- $13.4375 \text{ Mit Taschenrechnermethode DIV 16 und Rest } D.7_{16} = \underline{0'1101.0111_2}$
- $AF3_{16} = 10 \cdot 16^2 + 15 \cdot 16^1 + 3 \cdot 16^0 = 2560 + 240 + 3 = \underline{2803}$
- $765_8 = 7 \cdot 8^2 + 6 \cdot 8^1 + 5 \cdot 8^0 = 448 + 48 + 5 = \underline{501}$
- $94A.6_{16} = 9 \cdot 16^2 + 4 \cdot 16^1 + 10 \cdot 16^0 + 6 \cdot 16^{-1} = 2304 + 64 + 10 + 0.375 = \underline{2378.375}$
- $245.3_8 = 2 \cdot 16^2 + 4 \cdot 16^1 + 5 \cdot 16^0 + 3 \cdot 16^{-1} = 512 + 64 + 5 + 0.1875 = \underline{581.1875}$
- $432 \text{ Mit Taschenrechnermethode DIV 16 und Rest } 1B0_{16} = \underline{01'1010'0000_2} = \underline{660_8}$

$$m) 335_8 : 15_{16} =$$

$$\begin{array}{r} 011011101_2 : 10101_2 = 01010_2 \\ -10101 \\ \hline 0011010 \\ -10101 \\ \hline 001011_2 \text{ Rest} \end{array}$$

$$n) 7323_8 : 159_{16} =$$

$$\begin{array}{r} 11110101010101_2 : 101011001_2 = 1011_2 \\ -101011001 \\ \hline 10010000110 \\ -101011001 \\ \hline 00111011011 \\ -101011001 \\ \hline 010000010_2 = \text{Rest} = 64 \end{array}$$

11. Welche Dualzahlen ergeben sich, wenn die Dualzahlen von 93 und 56 binärmässig mit AND, OR und NOR verknüpft werden!

Wie erarbeitet, hat das AND-Gatter dann den Status 1, wenn alle Eingänge den Status 1 haben und in den restlichen Fällen den Status 0.

Das OR-Gatter hat dann den Status 0, wenn keiner der Eingänge gesetzt ist und in allen andern Fällen den Status 1.

Das OR-Gatter hat wegen der Invertierung dann den Status 1, wenn keiner der Eingänge den Status 1 hat, in allen andern Fällen damit den Status 0.

Damit ergibt sich:

93 = 5D ₁₆ = 0101'1100 ₂	} Eingänge
56 = 38 ₁₆ = 0011'1000 ₂	
AND: 0001'1000 ₂	} Ausgänge
OR: 0111'1100 ₂	
NOR: 1000'0011 ₂	
XOR: 0110'0100 ₂	
XNOR: 1001'1011 ₂	