This document contains supplemental material for the screen casts about

- Number Representation and
- Logic.

Mainly this supplemental material consists of the additional explanations and exercises that were previously included in the lecture.

These explanations and exercises should support you in understanding the content in the screen casts. You should make use of the material. Try to comprehend it in the same way as the screen casts. Certainly it is up to you whether and how to use it.
Note that this document does not necessarily contain all steps towards the correct solutions of the exercises. Anyhow it contains the exercises and partially their solutions. In the exam you might need to write down complete solutions including the correct steps towards the solution.

A good approach is to watch the screen cast until you understood everything. Whenever you do not understand an aspect, repeat the video. Often it is necessary to write down examples on your own, not only looking at the slides. Often you see what you do not understand, when you write it down on your own.
You can work with the additional explanations and exercises in different ways. For example, you can watch the video completely and practice afterwards, or interrupt the video and perform the exercises directly after the respective topic was discussed.

Number Representation

- Generic Representation

  With $M = N = 2$ we get $n = n_{-2} \cdot B^{-2} + n_{-1} \cdot B^{-1} + n_0 \cdot B^0 + n_1 \cdot B^1$.

- Dual System – Examples

  a) $10\ 0000_2 = 32_{10}$
  b) $1000\ 0000_2 = 128_{10}$
  c) $1111\ 1111_2 = 255_{10}$
  d) $11\ 1011\ 1101_2 = 957_{10}$

- Octal System – Examples

  a) $174_8 = 124_{10} = 1\ 111\ 100_2$
  b) $1111_8 = 585_{10} = 1\ 001\ 001\ 001_2$
  c) $1234_8 = 668_{10} = 1\ 010\ 011\ 100_2$

- Hexadecimal System – Examples

  a) $4321_{16} = 17185_{10} = 100\ 0011\ 0010\ 0001_2$
  b) $AFFE_{16} = 45054_{10} = 1010\ 1111\ 1111\ 1110_2$
  c) $EDE_{16} = 3806_{10} = 1110\ 1101\ 1110_2$

- Connections Between the Systems

  $2|5|3|6|3|2|0|0|4|4|3_8 = 10|10\ 1|011\ |110|0\ 11|01\ 0|000\ |000|1\ 00|10\ 0|011 = AB\ CD\ 01\ 23_{16}$

- Conversion from the Decimal System

  a) $21_{10} = 10101_2$
  b) $43_{10} = 101011_2$
  c) $763_{10} = 1011111011_2$
  d) $0{,}375_{10} = 0{,}011_2$
  e) $0{,}28125_{10} = 0{,}01001_2$
  f) $0{,}828125_{10} = 0{,}110101_2$

- Observations? – Bit Shift Operations

  a) $21_{10} \cdot 2_{10} = 10101_2 \cdot 10_2 = 101010_2 = 42_{10}$
  b) $43_{10} : 2_{10} = 101011_2 : 10_2 = 10101{,}1_2 = 21{,}5_{10}$
  c) $7{,}875_{10} \cdot 2_{10} = 111{,}111_2 \cdot 10_2 = 1111{,}11_2 = 15{,}75_{10}$
  d) $7{,}875_{10} : 2_{10} = 111{,}111_2 : 10_2 = 11{,}1111_2 = 3{,}9375_{10}$

- Negative Integers – Sign Bit

  a) With $N = 8$ all integers within $[-127\,; 127]$ can be represented.
  b) With $N = 16$ all integers within $[-32.767\,; 32.767]$ can be represented.

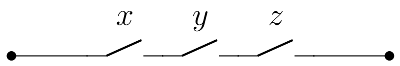
- Negative Integers – Excess-K Code
  a) With $N = 8$ and $K = 1$ all integers within $[-1\,; 254]$ can be represented.
  b) With $N = 8$ and $K = 128 = 2^{N-1}$ all integers within $[-128\,; 127]$ can be represented.
  c) With $N = 8$ and $K = 254$ all integers within $[-254\,; 1]$ can be represented.
  d) With $N = 16$ and $K = 32.768 = 2^{N-1}$ all integers within $[-32.768\,; 32.767]$ can be represented.
- Negative Integers – One's Complement
  a) With $N = 8$ all integers within $[-127\,; 127]$ can be represented.
  b) With $N = 16$ all integers within $[-32.767\,; 32.767]$ can be represented.
- Negative Integers – Two's Complement
  a) With $N = 8$ all integers within $[-128\,; 127]$ can be represented.
  b) With $N = 16$ all integers within $[-32.768\,; 32.767]$ can be represented.
- Calculation with Integers
  a) $-5_{10} + 3_{10} = 1011_2 + 0011_2 = 1110_2 = -2_{10}$
  b) $-2_{10} - 4_{10} = -2_{10} + (-4_{10}) = 1110_2 + 1100_2 = 1|1010_2 = -6_{10}$
  c) $-2_{10} - (-5_{10}) = -2_{10} + 5_{10} = 1110_2 + 0101_2 = 1|0011_2 = 3_{10}$
  d) $108_{10} : 6_{10} = 1101100_2 : 110_2 = 10010_2 = 18_{10}$
  e) $11_{10} \cdot 4_{10} = 1011_2 \cdot 100_2 = 101100_2 = 44_{10}$
- Rational and Real Numbers

  The decimal numbers 0,1, 0,2, 0,3, 0,4, 0,6, 0,7, 0,8 and 0,9 are periodic in the binary representation, 0,0 and 0,5 are not.
- Ranges of the Number Representations – Unsigned Integers with N Bits
  a) Smallest value: $0_{10} = \underbrace{0\ldots0}_{N}{}_2$ (for $N = 8$ we get $0_{10} = 0000\,0000_2$)
  b) $(2^{N-1})_{10} = 1\underbrace{0\ldots0}_{N-1}{}_2$ (for $N = 8$ we get $128_{10} = 1000\,0000_2$)
  c) Largest value: $(2^N - 1)_{10} = \underbrace{1\ldots1}_{N}{}_2$ (for $N = 8$ we get $255_{10} = 1111\,1111_2$)
- Ranges of the Number Representations – Integers in Two's Complement with N Bits
  a) Smallest value: $-(2^{N-1})_{10} = 1\underbrace{0\ldots0}_{N-1}{}_2$ (for $N = 8$ we get $-128_{10} = 1000\,0000_2$)
  b) $-1_{10} = \underbrace{1\ldots1}_{N}{}_2$ (for $N = 8$ we get $-1_{10} = 1111\,1111_2$)
  c) $0_{10} = \underbrace{0\ldots0}_{N}{}_2$ (for $N = 8$ we get $0_{10} = 0000\,0000_2$)
  d) $1_{10} = \underbrace{0\ldots0}_{N-1}1_2$ (for $N = 8$ we get $1_{10} = 0000\,0001_2$)
  e) Largest value: $(2^{N-1} - 1)_{10} = 0\underbrace{1\ldots1}_{N-1}{}_2$ (for $N = 8$ we get $127_{10} = 0111\,1111_2$)

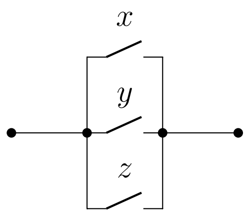
Logic

Several truth tables in this section are not filled completely. Please fill those tables, this is a good exercise to work with truth tables.

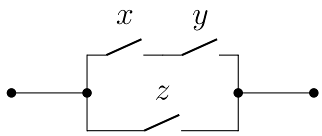
- Logical expressions for some given circuits



$x \land y \land z$

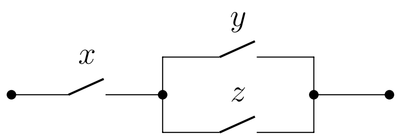


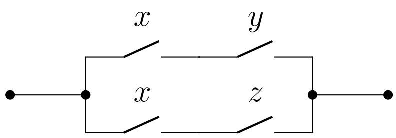
$x \lor y \lor z$



$x \land y \lor z$

- (Non-)Equivalence of expressions





a) $(x \land (y \lor z)) = x \land (y \lor z) \neq x \land y \lor z$

b) $((x \land y) \lor (x \land z)) = (x \land y) \lor (x \land z) = x \land y \lor x \land z$

c) Application of distributive law: $x \land (y \lor z) = (x \land y) \lor (x \land z)$
(Compare these expressions to the circuits above.)

d) Usage of a truth table:

x	y	z	$y \lor z$	$x \land (y \lor z)$	$(x \land y)$	$(x \land z)$	$(x \land y) \lor (x \land z)$	$\stackrel{?}{=}$	?
0	0	0	0	0	0	0	0	$0 = 0$	✓
0	0	1	…	…	…	…	…	…	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

- Truth tables for De Morgan's laws

a) $a \land b = \overline{\overline{a} \lor \overline{b}}$

a	b	$\overline{a}$	$\overline{b}$	$\overline{a} \lor \overline{b}$	$\overline{\overline{a} \lor \overline{b}}$	$a \land b$
0	0	1	1	1	0	0
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	0	1	1

Obviously $a \land b = \overline{\overline{a} \lor \overline{b}}$ holds, since the two rightmost columns are identical.

b) $a \vee b = \overline{\overline{a} \wedge \overline{b}}$

a	b	$\overline{a}$	$\overline{b}$	$\overline{a} \wedge \overline{b}$	$\overline{\overline{a} \wedge \overline{b}}$	$a \vee b$
0	0	1	1	1	0	0
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	0	1	1

Obviously $a \vee b = \overline{\overline{a} \wedge \overline{b}}$ holds, since the two rightmost columns are identical.

- Proofs for some non-trivial calculation rules for Boolean Algebra

a) Commutative laws $a \wedge b = b \wedge a$ and $a \vee b = b \vee a$

a \ b	0	1
0	0	0
1	0	1

a \ b	0	1
0	0	1
1	1	1

The truth tables for *logical and* (left table) and *logical or* (right table) are symmetric on the diagonal, thus a and b can be exchanged. This is a simple proof for the commutative laws.

b) Associative laws $a \wedge (b \wedge c) = (a \wedge b) \wedge c$ and $a \vee (b \vee c) = (a \vee b) \vee c$

a	b	c	$b \wedge c$	$a \wedge (b \wedge c)$	$a \wedge b$	$(a \wedge b) \wedge c$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	1	0
1	1	1	1	1	1	1

a	b	c	$b \vee c$	$a \vee (b \vee c)$	$a \vee b$	$(a \vee b) \vee c$
0	0	0	0	0	0	0
0	0	1	...	...	...	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮

The corresponding columns are identical, thus the identities are proven.

c) Distributive laws $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$ and $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$

a	b	c	$b \vee c$	$a \wedge (b \vee c)$	$a \wedge b$	$a \wedge c$	$(a \wedge b) \vee (a \wedge c)$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

a	b	c	$b \wedge c$	$a \vee (b \wedge c)$	$a \vee b$	$a \vee c$	$(a \vee b) \wedge (a \vee c)$
0	0	0	0	0	0	0	0
0	0	1	...	...	...	...	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

The corresponding columns of both sides of the equations above are identical, thus the identities of distributive laws are proven.

d) Simplification laws $a \wedge (a \vee b) = a$ and $a \vee (a \wedge b) = a$

These identities shall be intuitively clear. Certainly can be proven by creating the truth tables.

Here we show a nice calculation from one to the other term of the simplification laws under usage of the distributive law. Certainly, all three terms simplify to a.

$$a \wedge (a \vee b) = (a \wedge a) \vee (a \wedge b) = a \vee (a \wedge b)$$

e) Further simplification laws $a \wedge (\overline{a} \vee b) = a \wedge b$ and $a \vee (\overline{a} \wedge b) = a \vee b$

These identities shall also be intuitively clear. Certainly can be proven by creating the truth tables. Here we show another proof using the distributive laws.

$$a \wedge (\overline{a} \vee b) = (a \wedge \overline{a}) \vee (a \wedge b) = 0 \vee (a \wedge b) = a \wedge b$$
$$a \vee (\overline{a} \wedge b) = (a \vee \overline{a}) \wedge (a \vee b) = 1 \wedge (a \vee b) = a \vee b$$

- Number of n-ary Boolean functions

The number of n-ary Boolean functions are given by the function 2^{2^n}. This function possesses an extreme growth behavior.

n	2^{2^n}	2^{2^n}	Counting this number is possible with
1	2^2	4	2 bits
2	2^4	16	4 bits
3	2^8	256	1 Byte = 8 bits
4	2^{16}	65.536	2 Bytes = 16 bits
5	2^{32}	4.294.967.296	4 Bytes = 32 bits
6	2^{64}	18.446.744.073.709.551.616	8 Bytes = 64 bits

- CNF, DNF, Simplification of logical expressions

  Given a truth table, the CNF can be determined in another way as shown in the slides: The terms with value 0 are selected and combined as in the DNF. Then the whole expression needs to be negated. Application of De Morgans's laws twice results in the CNF:

$$\begin{aligned} f(a,b,c) &= \overline{(\overline{a} \wedge b \wedge \overline{c}) \vee (\overline{a} \wedge b \wedge c)} \\ &= \overline{(\overline{a} \wedge b \wedge \overline{c})} \wedge \overline{(\overline{a} \wedge b \wedge c)} \\ &= (a \vee \overline{b} \vee c) \wedge (a \vee \overline{b} \vee \overline{c}) \ . \end{aligned}$$

  Now we can simplify this expression by the application of the distributive law, then eliminating duplicates and finally applying simplification laws.

$$\begin{aligned} f(a,b,c) &= (a \vee \overline{b} \vee c) \wedge (a \vee \overline{b} \vee \overline{c}) \\ &= (a \wedge a) \vee (a \wedge \overline{b}) \vee (a \wedge \overline{c}) \vee (\overline{b} \wedge a) \vee (\overline{b} \wedge \overline{b}) \vee (\overline{b} \wedge \overline{c}) \vee \\ &\quad (c \wedge a) \vee (c \wedge \overline{b}) \vee (c \wedge \overline{c}) \\ &= a \vee (a \wedge \overline{b}) \vee (a \wedge \overline{c}) \vee \overline{b} \vee (\overline{b} \wedge \overline{c}) \vee \\ &\quad (c \wedge a) \vee (c \wedge \overline{b}) \vee 0 \\ &= [a \vee (a \wedge \overline{b}) \vee (a \wedge \overline{c}) \vee (a \wedge c)] \vee [\overline{b} \vee (\overline{b} \wedge \overline{c}) \vee (\overline{b} \wedge c)] \\ &= a \vee \overline{b} \ . \end{aligned}$$

- Correctly marked areas in Karnaugh Maps

  a) Correctly marked areas in Karnaugh Maps possess 2^n 1's for $n \geq 0$, i. e. $1, 2, 4, 8, \ldots$ 1's in a rectangular area. The following table shows the size of the area with 1's (which must be correctly marked) and the number of literals necessary to provide the value of the expression.

Part of area	whole	1/2	1/4	1/8	1/16	...
Required number of literals	0	1	2	3	4	...

  b) Concrete example: If a Karnaugh Map for a function with four variables is created (with 16 cells), then the following table shows the number of correctly marked cells and the number of required literals to provide the value of the expression.

Number of correctly marked cells	16	8	4	2	1
Required number of literals	0	1	2	3	4

- Truth table, CNF, DNF, Karnaugh Map

a	b	$f(a,b)$
0	0	1
0	1	1
1	0	0
1	1	1

  a) CNF: $f(a,b) = \overline{a} \vee b$

  b) DNF: $f(a,b) = (\overline{a} \wedge \overline{b}) \vee (\overline{a} \wedge b) \vee (a \wedge b)$

c) Simplification of the DNF:
$(\overline{a} \wedge \overline{b}) \vee (\overline{a} \wedge b) \vee (a \wedge b) = \overline{a} \wedge (\overline{b} \vee b) \vee (a \wedge b) = \overline{a} \wedge 1 \vee (a \wedge b) = \overline{a} \vee (a \wedge b) = \overline{a} \vee b$

d) Karnaugh Map

$f(a,b)$		b
	1 (0)	1 (1)
a	0 (2)	1 (3)

Here, we directly see that $f(a,b) = \overline{a} \vee b$.

- And one more: Truth table, CNF, DNF, Karnaugh Map

a	b	c	$f(a,b,c)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

a) CNF: $f(a,b,c) = (a \vee b \vee c) \wedge (a \vee \overline{b} \vee \overline{c}) \wedge (\overline{a} \vee b \vee \overline{c}) \wedge (\overline{a} \vee \overline{b} \vee c)$

b) DNF: $f(a,b,c) = (\overline{a} \wedge \overline{b} \wedge c) \vee (\overline{a} \wedge b \wedge \overline{c}) \vee (a \wedge \overline{b} \wedge \overline{c}) \vee (a \wedge b \wedge c)$
The terms in CNF and DNF look similar here (invert the order in one of the normal forms) in this example. This does not hold in general, it is caused by the (very) special values of $f(a,b,c)$. So, constructing one normal form based on the other by just changing $\vee$ to $\wedge$ and vice versa does not work in general, just here and in other rare cases!

c) Here is a calculation, starting with the CNF, applying associative and distributive laws and elimination of duplicates, and ending in the DNF. Several intermediate steps are left out here and are a nice exercise.

$$\begin{aligned} f(a,b,c) &= [(a \vee b \vee c) \wedge (a \vee \overline{b} \vee \overline{c})] \wedge [(\overline{a} \vee b \vee \overline{c}) \wedge (\overline{a} \vee \overline{b} \vee c)] \\ &= [a \vee (b \wedge \overline{c}) \vee (\overline{b} \wedge c)] \wedge [\overline{a} \vee (b \wedge c) \vee (\overline{b} \wedge \overline{c})] \\ &= (a \wedge b \wedge c) \vee (a \wedge \overline{b} \wedge \overline{c}) \vee (\overline{a} \wedge b \wedge \overline{c}) \vee (\overline{a} \wedge \overline{b} \wedge c) \ . \end{aligned}$$

d) Karnaugh Map

$f(a,b,c)$			a	a
		c	c	
	0 (0)	1 (1)	0 (5)	1 (4)
b	1 (2)	0 (3)	1 (7)	0 (6)

Here, we directly see that it is not possible to group two or more cells correctly. So, no simplification can be done on this way.

- Structure of Karnaugh Maps

$f(a)$

a

0	1

$f(a,b)$

b a

0	1
2	3

$f(a,b,c)$

a c b

0	1	5	4
2	3	7	6

$f(a,b,c,d)$

b d c a

0	1	5	4
2	3	7	6
10	11	15	14
8	9	13	12

$f(a,b,c,d,e)$

a c e e d b

0	1	5	4	20	21	17	16
2	3	7	6	22	23	19	18
10	11	15	14	30	31	27	26
8	9	13	12	28	29	25	24

$f(a,b,c,d,e,f)$

b d f f e c e a

0	1	5	4	20	21	17	16
2	3	7	6	22	23	19	18
10	11	15	14	30	31	27	26
8	9	13	12	28	29	25	24
40	41	45	44	60	61	57	56
42	43	47	46	62	63	59	58
34	35	39	38	54	55	51	50
32	33	37	36	52	53	49	48

$f(a,b,c,d,e,f,g)$

a c e e g g g g f d b f

0	1	5	4	20	21	17	16	80	81	85	84	68	69	65	64
2	3	7	6	22	23	19	18	82	83	87	86	70	71	67	66
10	11	15	14	30	31	27	26	90	91	95	94	78	79	75	74
8	9	13	12	28	29	25	24	88	89	93	92	76	77	73	72
40	41	45	44	60	61	57	56	120	121	125	124	108	109	105	104
42	43	47	46	62	63	59	58	122	123	127	126	110	111	107	106
34	35	39	38	54	55	51	50	114	115	119	118	102	103	99	98
32	33	37	36	52	53	49	48	112	113	117	116	100	101	97	96

In this representation the small numbers give the line numbers in the truth table when the variables are arranged there according to the Karnaugh Map. They start with 0 and end with $2^{|X|}-1$, where X is the set of variables.

The marked areas need to contain 1's only, need be contiguous, and each of both dimensions needs to have a size of a power of 2.

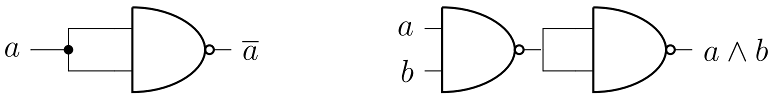
Obviously, Karnaugh Maps can be created with one contiguous area for *true* and one contiguous area for *false* for each variable with up to four variables. For more variables, the areas for the fifth and following variables need to be split up as visible in the diagrams.

- NAND logic

a	b	$a \wedge b$	$\overline{a \wedge b}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

a) The proofs for expressing locical expressions with NAND gates only can be done using truth tables.

b) Here are two examples for the visualization of NAND logic: $\overline{a}$ and $a \wedge b$.



c) Express a XOR b with NAND gates only:

$$
\begin{aligned}
\mathrm{XOR}(a,b) &= (a \wedge \overline{b}) \vee (\overline{a} \wedge b) \\
&= (a \wedge \overline{(b \wedge b)}) \vee (\overline{(a \wedge a)} \wedge b) \\
&= \overline{[\overline{(a \wedge \overline{(b \wedge b)}))} \wedge \overline{(a \wedge \overline{(b \wedge b)})}]} \vee \overline{[\overline{(\overline{(a \wedge a)} \wedge b)} \wedge \overline{(\overline{(a \wedge a)} \wedge b)}]} \\
&= \overline{\overline{(x \wedge x)} \wedge \overline{(y \wedge y)}}\ ,
\end{aligned}
$$

where

$$
\begin{aligned}
x &= \overline{[\overline{(a \wedge \overline{(b \wedge b)})} \wedge \overline{(a \wedge \overline{(b \wedge b)})}]}\ , \\
y &= \overline{[\overline{(\overline{(a \wedge a)} \wedge b)} \wedge \overline{(\overline{(a \wedge a)} \wedge b)}]}\ .
\end{aligned}
$$

Based on the simple representation $(a \wedge \overline{b}) \vee (\overline{a} \wedge b)$ the expression with NAND gates only has been derived. In this case 23 NAND gates are required.

d) Another representation of XOR with only four NAND gates exists as shown on the lecture slides. We have a short look at the outputs of the four gates to prove that those four NAND gates calculate the XOR function.

   - The output of the leftmost gate is $\overline{x \wedge y}$.
   - Thus the outputs of the upper and lower gates are $\overline{x \wedge \overline{x \wedge y}}$ and $\overline{\overline{x \wedge y} \wedge y}$.
   - Then the output of the rightmost gate is $\overline{\overline{x \wedge \overline{x \wedge y}} \wedge \overline{\overline{x \wedge y} \wedge y}}$.
   - Applying De Morgan's laws twice leads to $(x \wedge (\overline{x} \vee \overline{y})) \vee ((\overline{x} \vee \overline{y}) \wedge y)$, which simplifies to $(x \wedge \overline{y}) \vee (\overline{x} \wedge y) = \mathrm{XOR}(x,y)$.