Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*

Stoff → *B06: Speicherplatz als rares Gut – Dateien und ihr Platzbedarf*

* Lenziele und Materialien erläutern!
* Erkennen, dass jede Datei ein Binärwert ist!
* Mit dem HEX-Editor Dateien analysieren!
* Dezimales SI-System und binäre Systeme erkennen!
* Wichtige Dateiarten unterscheiden!
* Codierungsarten für Texte definieren! → *ASCII-Code, Unicode*
* Hinweis auf zusätzliches Lernmaterial!

Übungen bzw. Aufgaben

* Block 5 'Fehler in der Datenübertragung finden' ist erarbeitet und die 4 enthaltenen, gelösten und besprochenen Aufgaben korrigieren!
* Block 6 'Speicherplatz als rares Gut – Dateien und ihr Platzbedarf' erarbeiten und Fragen notieren bzw. melden und minestens die enthaltenen 6 Aufgaben lösen!

Ausblick

Fr. 10. Nov.: - Speicherplatz als rares Gut → *B07: Kompression*
Fr. 17. Nov.: - Speicherplatz als rares Gut → *B08: Reduktion*
- Rückblickübungen → *B06..B08*
Fr. 24. Nov.: - Speicherplatzarbeiten erledigen → *B06..B08*
- Vektorgrafiken → *B09*

Stoffplan

Freitag:	KW	SW	Themen (Theorie und Übungen)	
25.08.2023	34	01	00 Begrüssung und Einleitung 01 Die Zahlensysteme BIN, HEX und DEZ kennenlernen	
01.09.2023	35	02	02 Arithmetische und logische Grundoperationen binär	
08.09.2023	36	03	Rückblickübungen zu Block 01 und 02 lösen	
15.09.2023	37	04	03 Die Logik und den Prozessor verstehen	
22.09.2023	38	05	Prüfung Block 01 und 02 04 Grosse Zahlen in kleinen Variablen ablegen, wie geht das?	P1
29.09.2023	39	06	Rückblickübungen zu Block 03 und 04 lösen	
			Herbstferien	
20.10.2023	42	07	05 Fehler in der Datenübertragung finden und korrigieren	
27.10.2023	43	08	Arbeit zu Block 02 bis und mit 04 schreiben	A1
03.11.2023	44	09	06 Speicherplatz als rares Gut – Dateien und ihr Platzbedarf	
10.11.2023	45	10	07 Speicherplatz als rares Gut – Dateien und ihr Platzbedarf, Kompression	
17.11.2023	46	11	08 Speicherplatz als rares Gut – Reduktion	
24.11.2023	47	12	Arbeit zu Block 06 bis und mit Block 08 schreiben 09 Vektorgrafiken – Eine Alternative zu den Pixeln	A2
01.12.2023	48	13	10 Verschlüsselung – Geschichte und Grundsätzliches	
08.12.2023	49	14	**Maria Empfängnis**	
15.12.2023	50	15	11 Verschlüsselung – Moderne Verfahren	
22.12.2023	51	16	Arbeit zu Block 09 bis und mit Block 11 schreiben	A3
			Weihnachtsferien	
12.01.2024	02	17	12 Kryptographie und Steganographie definieren und anwenden	
19.01.2024	03	18	Rückblickübungen über erarbeitete M114-Themen lösen	
26.01.2024	04	19	Rückblickübungen über erarbeitete M114-Themen abschliessen Modul abschliessen	

Rückblick

Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen

- 1 Logiksche Grundoperationsarbeit mit Ihrem Simulator wie z.B. WorkBench → *B2*
- 1 Logik - Prozessor - Aufgabe→ *B3*
- 1 Zahlen- und Variablen - Aufgabe→ *B4*

Anzahl Punkte	Note
20..22	6.0
19	5.8
18	5.5
17	5.3
16	5.0
15	4.8
14	4.5
13	4.3
12	4.0
11	3.8
10	3.5
9	3.3
8	3.0
7	2.8
6	2.5
5	2.3
4	2.0
3	1.8
2	1.5
1	1.3
0	1.0

M114 INF22xL Arbeit zu Unterrichtsblock 2, 3 und 4!

Sie erledigten zusammen mit einem Klassenkollegen bzw. einer Klassenkollegin die drei auf dem Prüfungsbogen klar definierten Aufgaben vollständig und klar. Wenn Sie einzelne Beschreibungen lieber mit dem PC erledigen wie z.B. das Schema mit Ihrem Simulator, dann geben Sie schlussendlich ein pdf-File mit dem Namen 'INF22x A1 name1_name2.pdf' auf die TEAMS-Aufgabe ab.

Merke:
- Alle in den folgenden 3 Aufgaben klar geforderten Entwicklungen bzw. Lösungen dokumentieren Sie entweder auf diesem Prüfungsbogen oder im File mit dem auf der Wandtafel definierten Namen!
- Am Ende geben Sie diesen Prüfungsbogen wieder ab und speichern Ihr File in der entsprechenden geöffnete TEAMS-Aufgabe!
- Bei jeder Aufgabe wird wie gelernt nicht nur das Endresultat, sondern auch die Vollständigkeit des Lösungsweges und die **Klarheit**, **Sauberkeit** und **Vollständigkeit** bewertet.

Maximal: 22 Punkte

Fach: ***M114***	Thema: ***Logik, Prozessor, Variablen*** *(B2..B4!)*	Punkte: **20**	S-INF22xL	Note: **6.0**

Rückblick

* **Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen**
  - **1 Logiksche Grundoperationsarbeit mit Ihrem Simulator wie z.B. WorkBench → *B2***
  - **1 Logik - Prozessor - Aufgabe→ *B3***
  - **1 Zahlen- und Variablen - Aufgabe→ *B4***

M114 INF22xL Arbeit zu Unterrichtsblock 2, 3 und 4!

Sie erledigten zusammen mit einem Klassenkollegen bzw. einer Klassenkollegin die drei auf dem Prüfungsbogen klar definierten Aufgaben vollständig und klar. Wenn Sie einzelne Beschreibungen lieber mit dem PC erledigen wie z.B. das Schema mit Ihrem Simulator, dann geben Sie schlussendlich ein pdf-File mit dem Namen 'INF22x A1 name1_name2.pdf' auf die TEAMS-Aufgabe ab.

Weitere Test:

24.11.23 Arbeit Speicherplatz

22.12.23 Arbeit Vektorgrafiken und Verschlüsselung

xxxx Blitzprüfung

Anzahl Punkte	Note
20..22	6.0
19	5.8
18	5.5
17	5.3
16	5.0
15	4.8
14	4.5
13	4.3
12	4.0
11	3.8
10	3.5
9	3.3
8	3.0
7	2.8
6	2.5
5	2.3
4	2.0
3	1.8
2	1.5
1	1.3
0	1.0

Merke:
- Alle in den folgenden 3 Aufgaben klar geforderten Entwicklungen bzw. Lösungen dokumentieren Sie entweder auf diesem Prüfungsbogen oder im File mit dem auf der Wandtafel definierten Namen!
- Am Ende geben Sie diesen Prüfungsbogen wieder ab und speichern Ihr File in der entsprechenden geöffnete TEAMS-Aufgabe!
- Bei jeder Aufgabe wird wie gelernt nicht nur das Endresultat, sondern auch die Vollständigkeit des Lösungsweges und die **Klarheit**, **Sauberkeit** und **Vollständigkeit** bewertet.

Maximal: 22 Punkte

Fach: ***M114***	Thema: ***Logik, Prozessor, Variablen*** *(B2..B4!)*	Punkte: **20**	S-INF22xL	Note: **6.0**

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*
  - Redundanz ist definiert und wurde an 1-aus10-Code und 2-aus-10-Code anwendet!
  - Hamming-Abstand ist definiert und Redundanzen wurden berechnet!
  - Prüfziffern sind definiert und wurden an CRC-Prüfung-Lernaufgabe angewendet!
  - Fehlererkennung und automatische Korrektur ist definiert! → *Paritätsbits, Hamming-Code*
  - Zusätzliches Lernmaterial sind erläutert und wurden durchgearbeitet! → *Fragen, Probleme*
  - Die 4 enthaltenen, erledigten Aufgaben und eventuell die Zusatzaufgaben besprechen!

Materialien

- Präsentation "Übertragungsfehler"
- Aufgaben "Übertragungsfehler"
- Musterlösungen

Dezimalziffer	2-aus-5-Code-Walking-Code	1-aus-10-Code	BCD-Code
0	11000	0000000001	0000
1	00011	0000000010	0001
2	00101	0000000100	0010
3	00110	0000001000	0011
4	01001	0000010000	0100
5	01010	0000100000	0101
6	01100	0001000000	0000
7	10001	0010000000	0111
8	10010	0100000000	1000
9	10100	1000000000	1001

1-aus-10-Code	
Stellenzahl	10
bewertbar	ja
stetig	ja
Gewicht	1
Maximaldistanz	2
Minimaldistanz	2
Hamming-Abstand	2
Redundanz	0.990

2-aus-5-Code	
Stellenzahl	5
bewertbar	ja/nein
stetig	nein
Gewicht	2
Maximaldistanz	4
Minimaldistanz	2
Hamming-Abstand	2
Redundanz	0.678

BCD-Code	
Stellenzahl	4
bewertbar	ja
stetig	nein
Gewicht	3
Maximaldistanz	1
Minimaldistanz	4
Hamming-Abstand	1
Redundanz	0.375

Rückblick

* **Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen**
* **Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären***

Ich kann...

- Erklären, was die beiden Begriffe «Redundanz» und «Hamming-Abstand» mit der Erkennung von Übertragungsfehlern zu tun haben.
- Das CRC-Prüfzifferverfahren beschreiben.
- Die Fehlererkennung mittels Paritätsbits erklären.
- Exemplarisch eine automatische Fehlerkorrektur mittels Hamming-Code durchführen.

Materialien

- Präsentation "Übertragungsfehler"
- Aufgaben "Übertragungsfehler"
- Musterlösungen

Dezimalziffer	2-aus-5-Code-Walking-Code	1-aus-10-Code	BCD-Code
0	11000	0000000001	0000
1	00011	0000000010	0001
2	00101	0000000100	0010
3	00110	0000001000	0011
4	01001	0000010000	0100
5	01010	0000100000	0101
6	01100	0001000000	0000
7	10001	0010000000	0111
8	10010	0100000000	1000
9	10100	1000000000	1001

1-aus-10-Code	
Stellenzahl	10
bewertbar	ja
stetig	ja
Gewicht	1
Maximaldistanz	2
Minimaldistanz	2
Hamming-Abstand	2
Redundanz	0.990

2-aus-5-Code	
Stellenzahl	5
bewertbar	ja/nein
stetig	nein
Gewicht	2
Maximaldistanz	4
Minimaldistanz	2
Hamming-Abstand	2
Redundanz	0.678

BCD-Code	
Stellenzahl	4
bewertbar	ja
stetig	nein
Gewicht	3
Maximaldistanz	1
Minimaldistanz	4
Hamming-Abstand	1
Redundanz	0.375

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*

→ Redundanz

Dezimalziffer	2-aus-5-Code-Walking-Code	1-aus-10-Code
0	11000	0000000001
1	00011	0000000010
2	00101	0000000100
3	00110	0000001000
4	01001	0000010000
5	01010	0000100000
6	01100	0001000000
7	10001	0010000000
8	10010	0100000000
9	10100	1000000000

$$\text{Redundanz} = \frac{\text{nicht genutzte Kombinationen}}{\text{alle möglichen Kombinationen}}$$

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*
  - Redundanz
  - Hamming-Abstand ist definiert und Redundanzen wurden berechnet!

Die Zahlen 1 bis 9 haben beim 1 aus 10 Code in Bezug auf die Zahl 0 einen Hammingabstand von **2**, der 2 aus 5 Code **2 bis 4**!

Dezimalziffer	2-aus-5-Code	1-aus-10-Code
0	11000	0000000001
1	00011	0000000010
2	00101	0000000100
3	00110	0000001000
4	01001	0000010000
5	01010	0000100000
6	01100	0001000000
7	10001	0010000000
8	10010	0100000000
9	10100	1000000000

Beim BCD-Code haben hingegeben die Zahlen 1 bis 7 in Bezug auf die Zahl 0 eine Hamming-distanz von **1 bis 3**, wie dies die Tablle rechts klar zeigt!

=> Hamming-Distanz definiert damit die Anzahl der unterschiedlichen Bit‘s zwischen 2 betrachteten Zahlen!

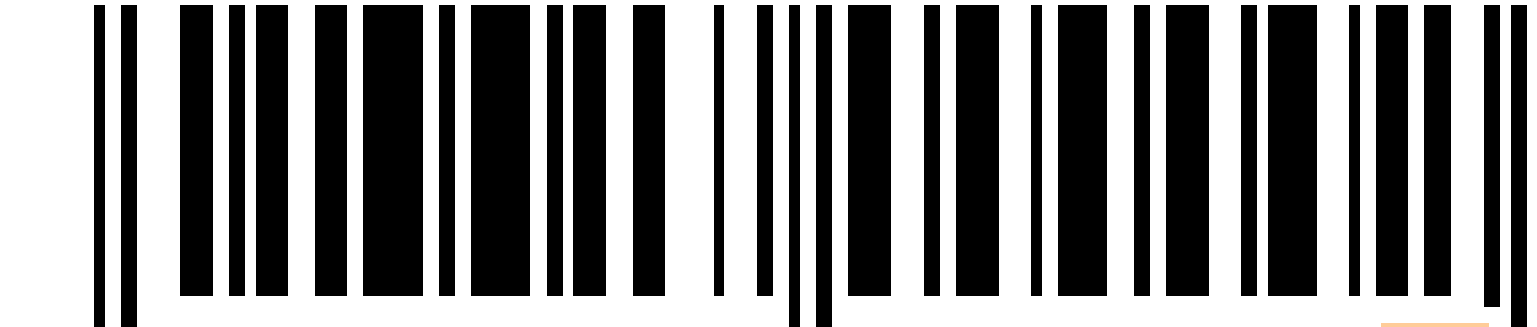
Dezimaler Wert Dezimal	Binärer Code Binär	Hamming- Distanz
0	000	(0)
1	001	1
2	010	1
3	011	2
4	100	1
5	101	2
6	110	2
7	111	3

Tabelle, welche die Hamming-Distanz von Dezimalzahlen verglichen zur Zahl ‚0‘ zeigt!

Stellenzahl
bewertbar
stetig
Gewicht
Maximaldistanz
Minimaldistanz
Hamming-Abstand
Redundanz

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*
  - Redundanz
  - Hamming-Abstand ist definiert und Redundanzen wurden berechnet!
  - Prüfziffern sind definiert



EAN-Ziffern	4	0	1	3	3	1	8	4	6	7	2	3
	×	×	×	×	×	×	×	×	×	×	×	×
Gewichtung	1	3	1	3	1	3	1	3	1	3	1	3
	=	=	=	=	=	=	=	=	=	=	=	=
Produkt	4	0	1	9	3	3	8	12	6	21	2	9

Quersumme = 78 nächste Zehnerzahl = 80

Prüfziffer: 80 - 78 = 2

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*
  - Redundanz
  - Hamming-Abstand ist definiert und Redundanzen wurden berechnet!
  - Prüfziffern sind definiert und wurden an CRC-Prüfung-Lernaufgabe angewendet!

In der Informatik ist das **CRC-Verfahren** (Cyclic Redundancy Check) zur Bildung von Prüfziffern binärer Werte sehr weit verbreitet. Damit werden zum Beispiel Ethernet-Frames im Netzwerk oder komprimierte Dateien auf ihre Korrektheit überprüft.

Lernaufgabe "Funktionsweise CRC-Prüfung"

Prüfverfahren zur Fehlererkennung Was ist ein Cyclic Redundancy Check (CRC)? **Der Cyclic Redundancy Check ist ein Prüfverfahren, mit dem sich Fehler in Datenblöcken erkennen lassen**. Das Verfahren kommt beispielsweise bei der Speicherung von Daten oder bei der Datenübertragung in Netzwerken zum Einsatz.

Definieren Sie die mit CRC geprüfte Nachricht m = 100‘1010$_2$ mit Generatorpolynom g = 1101$_2$!

=> Mit Anhang aus n-1 Nullen ergibt sich m‘ = 10‘0101‘0000, welche nun durch das Generatorpolynom „dividiert“ wird!

=> Welche Nachricht wird übertragen?

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*
  - Redundanz
  - Hamming-Abstand ist definiert und Redundanzen wurden berechnet!
  - Prüfziffern sind definiert und wurden an CRC-Prüfung-Lernaufgabe angewendet!
  - Fehlererkennung und automatische Korrektur ist definiert! → *Paritätsbits*

Die Idee des Paritätsbits ist einfach: Das Paritätsbit wird an einen zu übermittelnden Binärwert (zum Beispiel an ein Byte) angehängt.

Beim "Even-Parity-Verfahren" zeigt es an, ob die Anzahl Einsen in diesem Byte gerade (0) oder ungerade (1) ist. So kann nach der Übertragung einfach festgestellt werden, ob ein Fehler aufgetreten ist.

Ein Beispiel: Das Byte 10101010 soll übertragen werden. Die Anzahl Einsen im Byte ist gerade. Das Paritätsbit ist demnach 0 und wird ans Byte angehängt. Somit wird der Binär-Wert 101010100 übermittelt.

Der Empfänger trennt wiederum das Byte vom Paritätsbit und überprüft, ob die Anzahl der Einsen im Byte gerade ist.

									Paritätsbit Zeile
Byte 1	1	1	0	0	0	1	1	1	1
Byte 2	0	0	0	1	1	1	0	1	0
Byte 3	1	1	1	1	1	1	1	0	1
Byte 4	0	0	0	0	0	1	1	1	1
Byte 5	0	1	1	0	0	1	1	0	0
Byte 6	1	0	0	1	0	0	1	0	1
Byte 7	1	0	1	0	1	0	0	0	0
Byte 8	0	1	0	1	0	1	1	0	0
Paritätsbit Spalte	0	0	1	0	1	0	1	1	

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*
  - Redundanz
  - Hamming-Abstand ist definiert und Redundanzen wurden berechnet!
  - Prüfziffern sind definiert und wurden an CRC-Prüfung-Lernaufgabe angewendet!
* Fehlererkennung und automatische Korrektur ist definiert! → *Paritätsbits, Hamming-Code*

Ein zweiter Ansatz zur automatischen Fehlerkorrektur ist der Hamming Code.

Ein Beispiel: Es soll das Byte **1001 0000** gesichert übertragen werden.
Der Hamming-Code stellt uns für die Übertragung dieses Bytes einen Rahmen zur Verfügung. Man kann sich diesen Rahmen als Zug mit 12 Wagen vorstellen

Platz	12	11	10	9	8	7	6	5	4	3	2	1
Data												

Als Erstes wird das zu übertragende Byte in die violetten Wagen "abgefüllt":

Platz	12	11	10	9	8	7	6	5	4	3	2	1
Data	1	0	0	1		0	0	0		0		

Position 12: 1100
Position 9: 1001
XOR: 0101

Das Resultat der XOR-Verknüpfung wird nun in die grünen Wagen "abgefüllt":

Platz	12	11	10	9	8	7	6	5	4	3	2	1
Data	1	0	0	1	0	0	0	0	1	0	0	1

Position 12: 1100
Position 9: 1001
Position 4: 0100
Position 1: 0001
XOR: 0000

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*
  - Redundanz
  - Hamming-Abstand ist definiert und Redundanzen wurden berechnet!
  - Prüfziffern sind definiert und wurden an CRC-Prüfung-Lernaufgabe angewendet!
* Fehlererkennung und automatische Korrektur ist definiert! → *Paritätsbits, Hamming-Code*

Ein zweiter Ansatz zur automatischen Fehlerkorrektur ist der Hamming Code.

Ein Beispiel: Es soll das Byte **1001 0000** gesichert übertragen werden.
Der Hamming-Code stellt uns für die Übertragung dieses Bytes einen Rahmen zur Verfügung. Man kann sich diesen Rahmen als Zug mit 12 Wagen vorstellen

Platz	12	11	10	9	8	7	6	5	4	3	2	1
Data												

Als Erstes wird das zu übertragende Byte in die violetten Wagen "abgefüllt":

Platz	12	11	10	9	8	7	6	5	4	3	2	1
Data	1	0	0	1		0	0	0		0		

Position 12: 1100
Position 9: 1001
XOR: 0101

Das Resultat der XOR-Verknüpfung wird nun in die grünen Wagen "abgefüllt":

Platz	12	11	10	9	8	7	6	5	4	3	2	1
Data	1	0	0	1	0	1	0	0	1	0	0	1

Position 12: 1100
Position 9: 1001
Position 4: 0100
Position 1: 0001
XOR: 0000

Fehler und dessen Platz-nummerkann erkannt werden!

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*
  - Redundanz
  - Hamming-Abstand ist definiert und Redundanzen wurden berechnet!
  - Prüfziffern sind definiert und wurden an CRC-Prüfung-Lernaufgabe angewendet!
* Fehlererkennung und automatische Korrektur ist definiert! → *Paritätsbits, Hamming-Code*

Sehr gute Erklärung finden Sie z.B. beim angehängten Lern-Video!

Ein zweiter Ansatz zur automatischen Fehlerkorrektur ist der Hamming Code.

Ein Beispiel: Es soll das Byte **1001 0000** gesichert übertragen werden.

Der Hamming-Code stellt uns für die Übertragung dieses Bytes einen Rahmen zur Verfügung. Man kann sich diesen Rahmen als Zug mit 12 Wagen vorstellen

Platz	12	11	10	9	8	7	6	5	4	3	2	1
Data												

Nun werden die kompletten 12 Bit an den Empfänger übermittelt.

Auf der Empfänger-Seite werden nun die Nummern **aller** Wagen, welche eine Eins enthalten mit XOR verknüpft:

Position 12:	1100
Position 9:	1001
Position 4:	0100
Position 1:	0001
XOR-Verknüpfung:	0000

Ist das Resultat 0000, so wurden die 12 Byte korrekt übertragen.

Zusätzliches Lernmaterial

Weitere Erklärungen und Videos zu den Themen

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*
  - Redundanz
  - Hamming-Abstand ist definiert und Redundanzen wurden berechnet!
  - Prüfziffern sind definiert und wurden an CRC-Prüfung-Lernaufgabe angewendet!
  - Fehlererkennung und automatische Korrektur ist definiert! → *Paritätsbits, Hamming-Code*
  - Zusätzliches Lernmaterial sind erläutert und wurden durchgearbeitet! → *Fragen, Probleme*

Die 4 enthaltenen, erledigten Aufgaben und eventuell die Zusatzaufgaben besprechen!

Ihre persönliche Lösung können Sie mit der Musterlösungsvorgabe auf TEAMS-Dateien-M114 (Kef)-Unterrichtshilfe vergleichen!

Aufgabe 5.1: Codes mit erhöhter Redundanz

Neben dem bereits bekannten 1 aus 10 Code gibt es auch den 2 aus 5 Code um die Übertragungsqualität zu erhöhen. Dabei werden pro Dezimalziffer jeweils 5 Bits verwendet, wovon zwei den Wert 1 haben und die restlichen drei auf 0 gesetzt sind. Geben Sie für beide Codes die folgenden Werte an:

a) Hamming-Abstand **b)** Anzahl möglicher Kombinationen **c)** Anzahl gültiger und ungültiger Kombinationen **d)** Redundanz Aufgabe

Aufgabe 5.2: Paritätsbit

Folgende Datenblöcke mit der jeweiligen Datenlänge von 1 Byte wurden mit angehängtem Paritätsbit, d.h. dem Even Parity-Bit übertragen. Welche folgende, binäre Datenblöcke sind fehlerfrei angekommen?

a) 110011001 **b)** 111001111 **c)** 000000111 **d)** 000000000 **e)** 110000101 **f)** 001010110

Aufgabe 5.3: Paritätsbit pro Zeile und pro Spalte

Wir erweitern die Idee des Paritätsbits ein wenig, indem wir nach 8 Bytes (Sie folgende 8 Zeilen) jeweils ein Paritätsbit pro Spalte berechnen und dies dem Empfänger auch zusenden.

a) Es wurde ein Bit falsch übertragen. Finden und korrigieren Sie es.

b) Berechnen Sie die Redundanz, welche die Paritätsbits verursachen.

									Paritätsbit Zeile
Byte 1	1	1	0	0	0	1	1	1	**1**
Byte 2	0	0	0	1	1	1	0	1	**0**
Byte 3	1	1	1	1	1	1	1	0	**1**
Byte 4	0	0	0	0	0	1	1	1	**1**
Byte 5	0	1	1	0	0	1	1	0	**0**
Byte 6	1	0	0	1	0	0	1	0	**1**
Byte 7	1	0	1	0	1	0	0	0	**0**
Byte 8	0	1	0	1	0	1	1	0	**0**
Paritätsbit Spalte	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**1**	

Aufgabe 5.4: Hamming Code

Ein mit Hamming-Code gesichertes binäre Datenbyte wurde wie fehlerhaft von einem Empfänger entgegengenommen. Der Empfänger erhielt dabei folgende, binären Werte: 1001 0110 0101

a) An welcher Stelle ist der Fehler aufgetreten?

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*

Stoff → *B06: Speicherplatz als rares Gut – Dateien und ihr Platzbedarf*

Lenziele und Materialien erläutern!

Lernziele zu dieser Lerneinheit

Ich kann...

- Aufzeigen, dass jegliche Dateien lediglich einen Binärwert darstellen.
- Erklären, warum es viele verschiedene Dateiarten braucht.
- Die wichtigsten Dateiarten nennen und den Anwendungen zuordnen.
- Den Speicherbedarf von Dateiarten abschätzen.

Materialien

- Präsentation "Speicherplatz als rares Gut 1"
- Aufgaben "Speicherplatz als rares Gut 1"
- Musterlösungen

06 Speicherplatz als rares Gut – Dateien und ihr P...

Rückblick

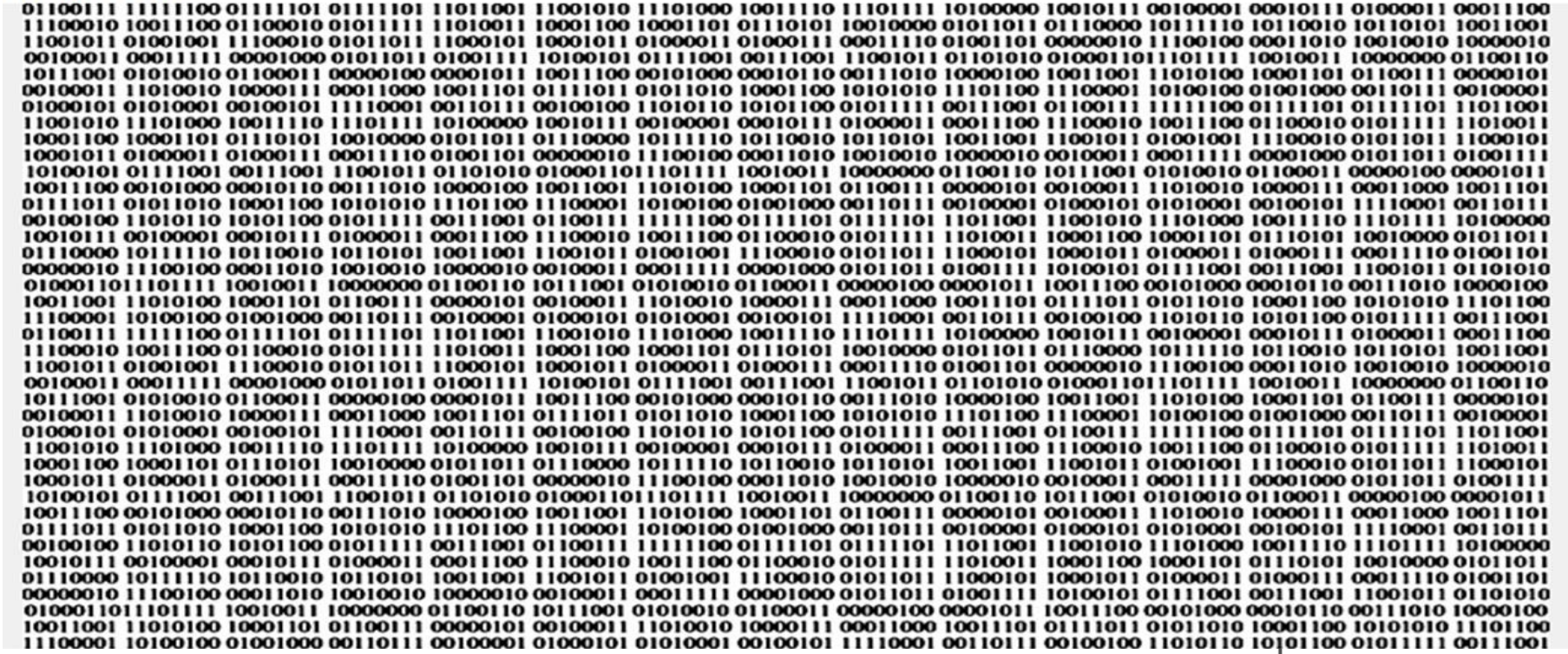
* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*

Stoff → *B06: Speicherplatz als rares Gut – Dateien und ihr Platzbedarf*

* Lenziele und Materialien erläutern!
* Erkennen, dass jede Datei ein Binärwert ist!

Prinzipiell sieht jede Datei wie eine lange Binärzahl aus und besteht aus einer grossen Anzahl von Nullen und Einsen.

Wir können uns also fragen: Welche Informationen sind in folgender Abbildung abgelegt?



Rückblick

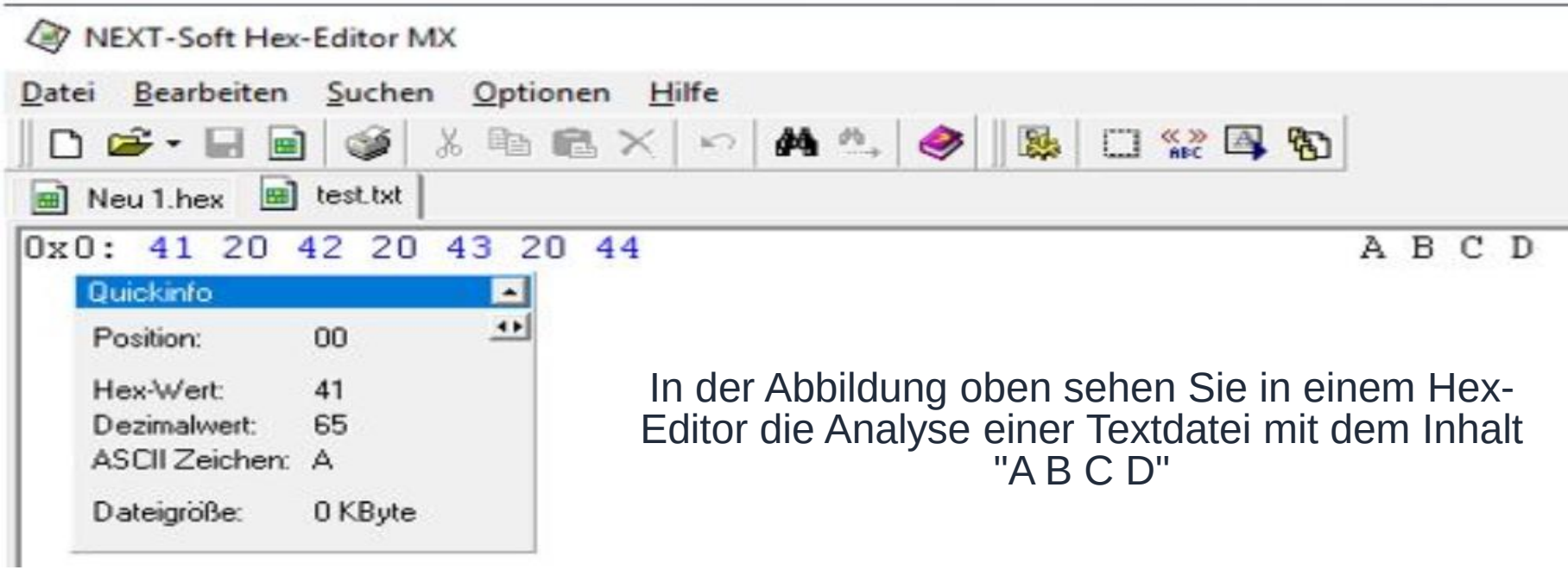
* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*

Stoff → *B06: Speicherplatz als rares Gut – Dateien und ihr Platzbedarf*

* Lenziele und Materialien erläutern!
* Erkennen, dass jede Datei ein Binärwert ist!
* Mit dem HEX-Editor Dateien analysieren!

Bei einem HEX-Editor handelt es sich um eine Software, welche die einzelnen Dateien oder Speicherinhalte binär darstellt und gleichzeitig Hexadezimal zusammenfasst.

Auf jeden Fall ist der HEX-Editor ein wichtiges Analyse-Tool zur genaueren Untersuchung unbekannter oder korrupter Dateien.



In der Abbildung oben sehen Sie in einem Hex-Editor die Analyse einer Textdatei mit dem Inhalt "A B C D"

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*

Stoff → *B06: Speicherplatz als rares Gut – Dateien und ihr Platzbedarf*

* Lenziele und Materialien erläutern!
* Erkennen, dass jede Datei ein Binärwert ist!
* Mit dem HEX-Editor Dateien analysieren!
* Dezimales SI-System

Einheit	Zusammenhang	Anzahl Bit
1 Bit (b)	Grundeinheit	$1 \cdot 10^0$ Bit
1 Byte (B)	8 Bit	$8 \cdot 10^0$ Bit
1 Kilobyte (KB)	1000 Byte	$8 \cdot 10^3$ Bit
1 Megabyte (MB)	1000 Kilobyte	$8 \cdot 10^6$ Bit
1 Gigabyte (GB)	1000 Megabyte	$8 \cdot 10^9$ Bit
1 Terabyte (TB)	1000 Gigabyte	$8 \cdot 10^{12}$ Bit
1 Petabyte (PB)	1000 Terabyte	$8 \cdot 10^{15}$ Bit

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*

Stoff → *B06: Speicherplatz als rares Gut – Dateien und ihr Platzbedarf*

* Lenziele und Materialien erläutern!
* Erkennen, dass jede Datei ein Binärwert ist!
* Mit dem HEX-Editor Dateien analysieren!
* Dezimales SI-System und binäre Systeme erkennen!

Einheit	Zusammenhang	Anzahl Bit	IEC-Bezeichnung
1 Bit (b)	Grundeinheit	$1 \cdot 2^0$ Bit	Bit
1 Byte (B)	8 Bit	$8 \cdot 2^0$ Bit	Byte
1 Kilobyte (KB)	1024 Byte	$8 \cdot 2^{10}$ Bit	Kibibyte (KiB)
1 Megabyte (MB)	1024 Kilobyte	$8 \cdot 2^{20}$ Bit	Mebibyte(MiB)
1 Gigabyte (GB)	1024 Megabyte	$8 \cdot 2^{30}$ Bit	Gibibyte (GiB)
1 Terabyte (TB)	1024 Gigabyte	$8 \cdot 2^{40}$ Bit	Tebibyte (TiB)
1 Petabyte (PB)	1024 Terabyte	$8 \cdot 2^{50}$ Bit	Pebibyte (PiB)

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*

Stoff → *B06: Speicherplatz als rares Gut – Dateien und ihr Platzbedarf*

* Lenziele und Materialien erläutern!
* Erkennen, dass jede Datei ein Binärwert ist!
* Mit dem HEX-Editor Dateien analysieren!
* Dezimales SI-System und binäre Systeme erkennen!
* **Wichtige Dateiarten unterscheiden!**

Grundsätzlich werden vier grundlegende Dateiarten unterschieden:

- **Ausführbare Dateien:** Anwendungen oder Dateien mit Befehlen / Scripts (.exe, .bat, .php, etc.)
- **Systemdateien:** Dienen zur Konfiguration von Hard- und Software (Conf-Files) (.bin, .drv, .ini, .sys, etc.)
- **Bibliotheken:** Enthalten Programmierwerkzeuge für Anwendungen (.cls, .dat, .dll, etc.)
- **Nutzerdaten:** Vom Nutzer (mit verschiedenen Anwendungen) erstellte Dateien

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*

Stoff → *B06: Speicherplatz als rares Gut – Dateien und ihr Platzbedarf*

* Lenziele und Materialien erläutern!
* Erkennen, dass jede Datei ein Binärwert ist!
* Mit dem HEX-Editor Dateien analysieren!
* Dezimales SI-System
* Wichtige Dateiarten u
* Codierungsarten für Texte definieren! → *ASCII-Code*

Tipp

Alt + 0178 → ² (m²)
Alt + 0179 → ³ (m³)

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	⌂

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	└	224	E0	α
129	81	ü	161	A1	í	193	C1	┴	225	E1	ß
130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
131	83	â	163	A3	ú	195	C3	├	227	E3	π
132	84	ä	164	A4	ñ	196	C4	─	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
134	86	å	166	A6	ª	198	C6	╞	230	E6	µ
135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
141	8D	ì	173	AD	¡	205	CD	═	237	ED	ø
142	8E	Ä	174	AE	«	206	CE	╬	238	EE	ε
143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
144	90	É	176	B0	░	208	D0	╨	240	F0	≡
145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
147	93	ô	179	B3	|	211	D3	╙	243	F3	≤
148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
150	96	û	182	B6	╢	214	D6	╓	246	F6	÷
151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	°
153	99	Ö	185	B9	╣	217	D9	┘	249	F9	∙
154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
155	9B	¢	187	BB	╗	219	DB	█	251	FB	√
156	9C	£	188	BC	╝	220	DC	▄	252	FC	ⁿ
157	9D	¥	189	BD	╜	221	DD	▌	253	FD	²
158	9E	₧	190	BE	╛	222	DE	▐	254	FE	■
159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	□

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*

Stoff → *B06: Speicherplatz als rares Gut – Dateien und ihr Platzbedarf*

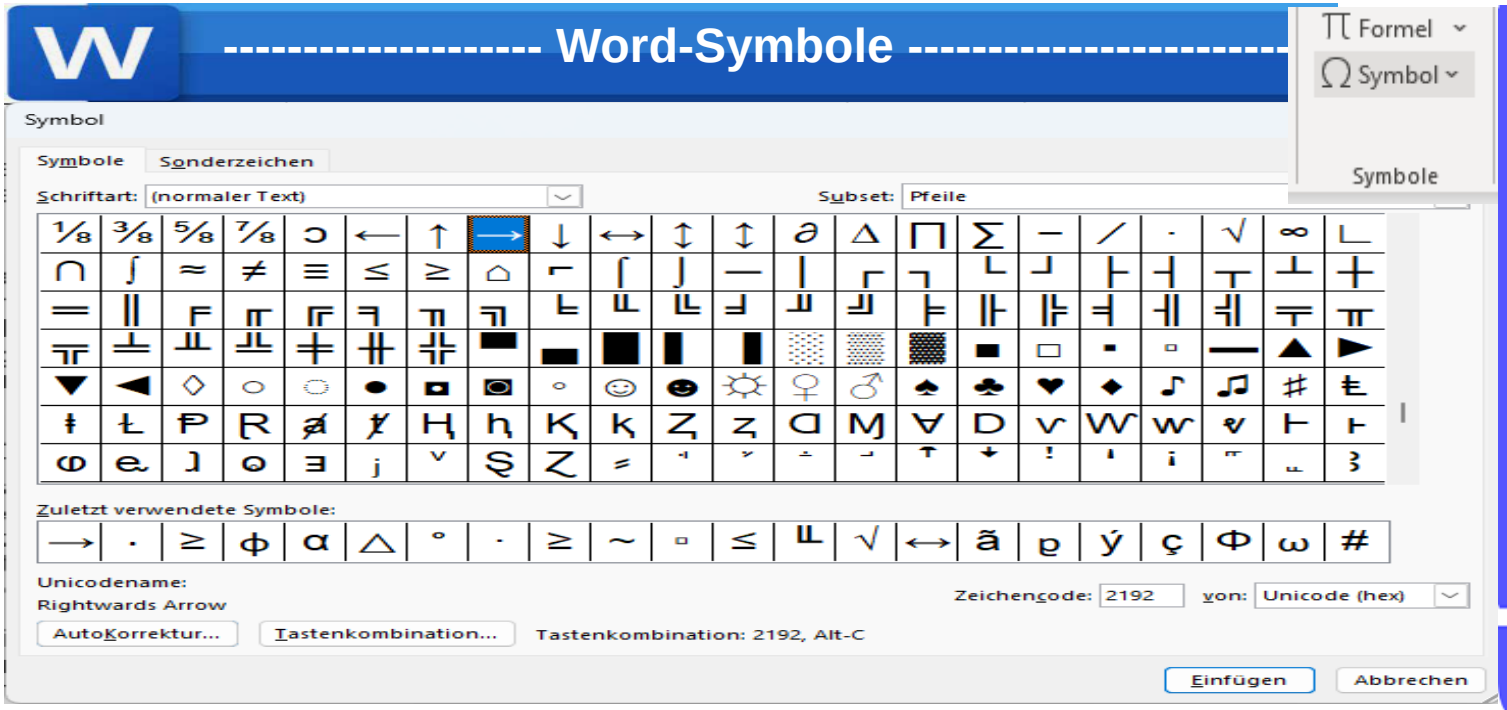
* Lenziele und Materialien erläutern!
* Erkennen, dass jede Datei ein Binärwert ist!
* Mit dem HEX-Editor Dateien analysieren!
* Dezimales SI-System
* Wichtige Dateiarten u
* Codierungsarten für Texte definieren! → *ASCII-Code, Unicode*

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*

Stoff → *B06: Speicherplatz als rares Gut – Dateien und ihr Platzbedarf*

* Lenziele und Materialien erläutern!
* Erkennen, dass jede Datei ein Binärwert ist!
* Mit dem HEX-Editor Dateien analysieren!
* Dezimales SI-System
* Wichtige Dateiarten u
* Codierungsarten für Texte definieren! → *ASCII-Code, Unicode*



Rückblick

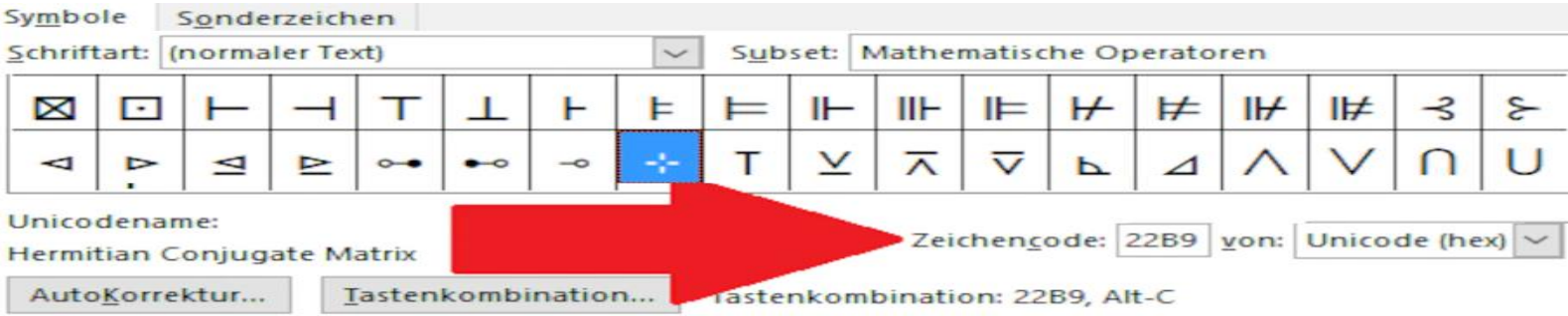
* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*

Stoff → *B06: Speicherplatz als rares Gut – Dateien und ihr Platzbedarf*

* Lenziele und Materialien erläutern!
* Erkennen, dass jede Datei ein Binärwert ist!
* Mit dem HEX-Editor Dateien analysieren!
* Dezimales SI-System
* Wichtige Dateiarten u
* Codierungsarten für Texte definieren! → *ASCII-Code, Unicode*
* Hinweis auf zusätzliches Lernmaterial!

Weitere Erklärungen zum Unicode

Der Unicode ist ein internationaler Standard für Zeichensätze, der in Computern für vielsprachige Textverarbeitung aber auch im Internet für die Codierung der Hypertext Markup Language (HTML) und auch für neue Internet-Protokolle verwendet wird. Er umfasst Schriftzeichen und Symbole aus den unterschiedlichsten Kulturen. Entsprechend umfangreich ist die Datenbank für Unicode- Zeichen, die etwa 230.000 Zeichen umfasst und eine Reserve von nahezu 1 Million Zeichen bietet.



Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*

Stoff → *B06: Speicherplatz als rares Gut – Dateien und ihr Platzbedarf*

* Lenziele und Materialien erläutern!
* Erkennen, dass jede Datei ein Binärwert ist!
* Mit dem HEX-Editor Dateien analysieren!
* Dezimales SI-System
* Wichtige Dateiarten u
* Codierungsarten für Texte definieren! → *ASCII-Code, Unicode*
* Hinweis auf zusätzliches Lernmaterial!

Übungen bzw. Aufgaben

* Block 5 'Fehler in der Datenübertragung finden' ist erarbeitet und die 4 enthaltenen, gelösten und besprochenen Aufgaben korrigieren!
* Block 6 'Speicherplatz als rares Gut – Dateien und ihr Platzbedarf' erarbeiten und Fragen notieren bzw. melden und minestens die enthaltenen 6 Aufgaben lösen!

Aufgabe 6.1: Dateityp .pdf

Zu welchem Zweck wurde das PDF-Format entwickelt?

Aufgabe 6.2: Dateigrössen

Öffnen Sie die Datei DummyText.txt (Modul-Share) in Word.
Speichern Sie die Datei in den Formaten .rtf, .pdf und .docx ab.
Prüfen Sie nun die Dateigrössen der einzelnen Files.
Wie lassen sich die Unterschiede erklären?

Sie lösen die folgenden 6 Aufgaben 6.1 bis 6.6 und mindestens die Zusatzaufgaben und melden alle Ihre Probleme bzw. Unklarheiten spätestens bei der Besprechung!

Aufgabe 6.1: Dateityp .pdf
Zu welchem Zweck wurde das PDF-Format entwickelt?

Aufgabe 6.2: Dateigrössen
Öffnen Sie die Datei DummyText.txt (Modul-Share) in Word.
Speichern Sie die Datei in den Formaten .rtf, .pdf und .docx ab.
Prüfen Sie nun die Dateigrössen der einzelnen Files.
Wie lassen sich die Unterschiede erklären?

Aufgabe 6.3: Dateigrössen im Alltag
Untersuchen Sie jeweils eine Text-, eine Musik- und eine Filmdatei aus Ihrem privaten Bestand auf Ihre Dateigrösse.
a) Merken Sie sich einen Richtwert für die jeweilige Grösse.
b) Wie viele dieser Dateien (pro Art) haben auf einer 500GB-Festplatte Platz?

Aufgabe 6.4: Datenübertragung
Wie lange dauert theoretisch ein Backup von 5GB Daten, wenn das Speichermedium eine Schreibgeschwindigkeit von 100Mb/s aufweist?

Aufgabe 6.5: Zeichencodierung
Schreiben Sie den Text Weiterbildung ist clever! im Editor (Windows-Zubehör) und speichern Sie ihn sowohl unter der Option ANSI (entspricht dem erweiterten ASCII) als auch unter der Option Unicode ab.
a) Welche Dateigrösse erwarten Sie für die beiden Dateien? Prüfen Sie nach.
b) Installieren Sie HexEditor MX (Modul-Share) und analysieren Sie beide Dateien.

Aufgabe 6.6: Zeichencodierung
Schreiben Sie nun folgenden Text im Editor und speichern Sie ihn wiederum unter beiden Optionen ab: rot blau grün
Was fällt Ihnen punkto Dateigrösse auf? Was sagt die Analyse?
Was bedeuten 0D und 0A?

Zusatzaufgabe für Interessierte: Unicode
Nehmen Sie sich etwas Zeit und betrachten Sie sich die Auswahl an Zeichensätzen auf http://unicode.org/charts/. Finden Sie...
a) den originalen ASCII-Code und seine Erweiterung
b) die germanischen Runen
c) die Symbole aus der Musik

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*

Stoff → *B06: Speicherplatz als rares Gut – Dateien und ihr Platzbedarf*

* Lenziele und Materialien erläutern!
* Erkennen, dass jede Datei ein Binärwert ist!
* Mit dem HEX-Editor Dateien analysieren!
* Dezimales SI-System und binäre Systeme erkennen!
* Wichtige Dateiarten unterscheiden!
* Codierungsarten für Texte definieren! → *ASCII-Code, Unicode*
* Hinweis auf zusätzliches Lernmaterial!

Übungen bzw. Aufgaben

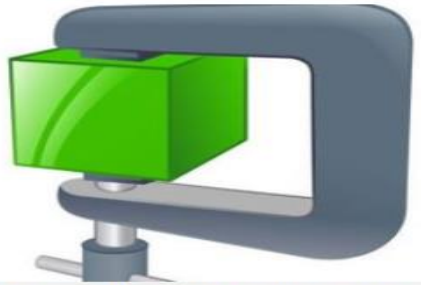
* Block 5 'Fehler in der Datenübertragung finden' ist erarbeitet und die 4 enthaltenen, gelösten und besprochenen Aufgaben korrigieren!
* Block 6 'Speicherplatz als rares Gut – Dateien und ihr Platzbedarf' erarbeiten und Fragen notieren bzw. melden und minestens die enthaltenen 6 Aufgaben lösen!

Ausblick

Fr. 10. Nov.: - Speicherplatz als rares Gut → *B07: Kompression*

Ich kann...

- Den Unterschied zwischen verlustfreier und verlustbehafteter Kompression erklären.
- Die Huffman-Codierung auf einfache Texte anwenden.
- Den Kompressionsfaktor und die Kompressionsrate berechnen.



07 Speicherplatz als rares Gut – Kompression

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*

Stoff → *B06: Speicherplatz als rares Gut – Dateien und ihr Platzbedarf*

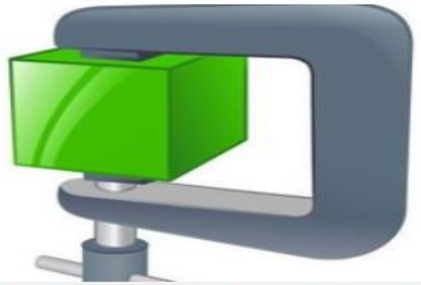
* Lenziele und Materialien erläutern!
* Erkennen, dass jede Datei ein Binärwert ist!
* Mit dem HEX-Editor Dateien analysieren!
* Dezimales SI-System und binäre Systeme erkennen!
* Wichtige Dateiarten unterscheiden!
* Codierungsarten für Texte definieren! → *ASCII-Code, Unicode*
* Hinweis auf zusätzliches Lernmaterial!

Übungen bzw. Aufgaben

* Block 5 'Fehler in der Datenübertragung finden' ist erarbeitet und die 4 enthaltenen, gelösten und besprochenen Aufgaben korrigieren!
* Block 6 'Speicherplatz als rares Gut – Dateien und ihr Platzbedarf' erarbeiten und Fragen notieren bzw. melden und minestens die enthaltenen 6 Aufgaben lösen!

Ausblick

Fr. 10. Nov.: - Speicherplatz als rares Gut → *B07: Kompression*
Fr. 17. Nov.: - Speicherplatz als rares Gut → *B08: Reduktion*
- Rückblickübungen → *B06..B08*
Fr. 24. Nov.: - Speicherplatzarbeiten erledigen → *B06..B08*
- Vektorgrafiken → *B09*



07 Speicherplatz als rares Gut – Kompression

Rückblick

* Bewertung Ihrer geleisteten Arbeit zu Block 2..4 wird nächste Woche besprochen
* Unterrichtsblock 5 'Fehler in der Datenübertragung finden ' ist erarbeitet und die enthaltenen 4 Aufgaben wurden gelöst! → *B05: Gemeldete Probleme klären*

Stoff → *B06: Speicherplatz als rares Gut – Dateien und ihr Platzbedarf*

* Lenziele und Materialien erläutern!
* Erkennen, dass jede Datei ein Binärwert ist!
* Mit dem HEX-Editor Dateien analysieren!
* Dezimales SI-System und binäre Systeme erkennen!
* Wichtige Dateiarten unterscheiden!
* Codierungsarten für Texte definieren! → *ASCII-Code, Unicode*
* Hinweis auf zusätzliches Lernmaterial!

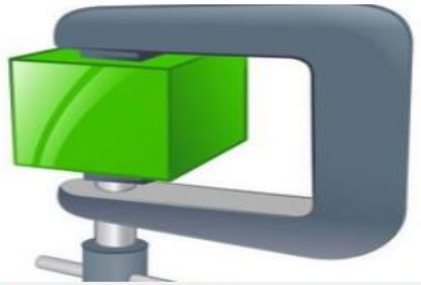
Übungen bzw. Aufgaben

→ Block 5 'Fehler in der Datenübertragung finden' ist erarbeitet und die 4 enthaltenen, gelösten und besprochenen Aufgaben korrigieren!

→ Block 6 'Speicherplatz als rares Gut – Dateien und ihr Platzbedarf' erarbeiten und Fragen notieren bzw. melden und minestens die enthaltenen 6 Aufgaben lösen!

Ausblick

→ Fr. 10. Nov.: - Speicherplatz als rares Gut → *B07: Kompression*
Fr. 17. Nov.: - Speicherplatz als rares Gut → *B08: Reduktion*
- Rückblickübungen → *B06..B08*
Fr. 24. Nov.: - Speicherplatzarbeiten erledigen → *B06..B08*
- Vektorgrafiken → *B09*



07 Speicherplatz als rares Gut – Kompression

Stoffplan

Freitag:	KW	SW	Themen (Theorie und Übungen)	
25.08.2023	34	01	00 Begrüssung und Einleitung 01 Die Zahlensysteme BIN, HEX und DEZ kennenlernen	
01.09.2023	35	02	02 Arithmetische und logische Grundoperationen binär	
08.09.2023	36	03	Rückblickübungen zu Block 01 und 02 lösen	
15.09.2023	37	04	03 Die Logik und den Prozessor verstehen	
22.09.2023	38	05	Prüfung Block 01 und 02 04 Grosse Zahlen in kleinen Variablen ablegen, wie geht das?	P1
29.09.2023	39	06	Rückblickübungen zu Block 03 und 04 lösen	
			Herbstferien	
20.10.2023	42	07	05 Fehler in der Datenübertragung finden und korrigieren	
27.10.2023	43	08	Arbeit zu Block 02 bis und mit 04 schreiben	A1
03.11.2023	44	09	06 Speicherplatz als rares Gut – Dateien und ihr Platzbedarf	
10.11.2023	45	10	07 Speicherplatz als rares Gut – Dateien und ihr Platzbedarf, Kompression	
17.11.2023	46	11	08 Speicherplatz als rares Gut – Reduktion	
24.11.2023	47	12	Arbeit zu Block 06 bis und mit Block 08 schreiben 09 Vektorgrafiken – Eine Alternative zu den Pixeln	A2
01.12.2023	48	13	10 Verschlüsselung – Geschichte und Grundsätzliches	
08.12.2023	49	14	**Maria Empfängnis**	
15.12.2023	50	15	11 Verschlüsselung – Moderne Verfahren	
22.12.2023	51	16	Arbeit zu Block 09 bis und mit Block 11 schreiben	A3
			Weihnachtsferien	
12.01.2024	02	17	12 Kryptographie und Steganographie definieren und anwenden	
19.01.2024	03	18	Rückblickübungen über erarbeitete M114-Themen lösen	
26.01.2024	04	19	Rückblickübungen über erarbeitete M114-Themen abschliessen Modul abschliessen	

A1. In einem C-Programm wird der Float-Variablen 'Temperatur' der Wert 556 zugewiesen. Beschreiben Sie klar und deutlich, wie diese float-Zahl im Speicher des verwendeten Notebook binär und hexadezimal aussehen wird! <6P>

In einem C-Programm wird eine Variablem vom Datentyp 'float' mit 32 Bit definiert. Dabei gilt nach IEEE 754:

→ **0** → **556**

→ **Das MSB mit der Wertigkeit 2^{31} definiert das Vorzeichen dieser Zahl '556', welches mit dem Status '0' positiv definiert.**

A1. In einem C-Programm wird der Float-Variablen 'Temperatur' der Wert 556 zugewiesen. Beschreiben Sie klar und deutlich, wie diese float-Zahl im Speicher des verwendeten Notebook binär und hexadezimal aussehen wird! <6P>

In einem C-Programm wird eine Variablem vom Datentyp 'float' mit 32 Bit definiert. Dabei gilt nach IEEE 754:

0100'0100'0 $= +1.0859375 \cdot 2^{9} = 556$

- **Das MSB mit der Wertigkeit 2^{31} definiert das Vorzeichen dieser Zahl '556', welches mit dem Status '0' positiv definiert.**
- **Mit den 8 Bits '10001000' wird mit dem Biased-Code der Wert '9' vom Exponenten definiert.**

A1. In einem C-Programm wird der Float-Variablen 'Temperatur' der Wert 556 zugewiesen. Beschreiben Sie klar und deutlich, wie diese float-Zahl im Speicher des verwendeten Notebook binär und hexadezimal aussehen wird! <6P>

In einem C-Programm wird eine Variablem vom Datentyp 'float' mit 32 Bit definiert. Dabei gilt nach IEEE 754:

$0100'0100'0000'1011'0000'0000'0000'0000_2 = +1.0859375 \cdot 2^9 = 556$

- **Das MSB mit der Wertigkeit 2^{31} definiert das Vorzeichen dieser Zahl '556', welches mit dem Status '0' positiv definiert.**
- **Mit den 8 Bits '10001000' wird mit dem Biased-Code der Wert '9' vom Exponenten definiert.**

Mit den Bits '000'1011'0000'0000'0000'0000' wird die Mantisse nach '1.' definiert, welche in diesem Fall:

$0 \cdot 0.5 + 0 \cdot 0.25 + 0 \cdot 0.125 + 1 \cdot 0.0625 + 0 \cdot 0.03125 + 1 \cdot 0.015625 + 1 \cdot 0.0078125$ entspricht!

Der Wert '1.' wird immer bei der Verwendung dieses Wertes vom System dazugerechnet.

A1. In einem C-Programm wird der Float-Variablen 'Temperatur' der Wert 556 zugewiesen. Beschreiben Sie klar und deutlich, wie diese float-Zahl im Speicher des verwendeten Notebook binär und hexadezimal aussehen wird! <6P>

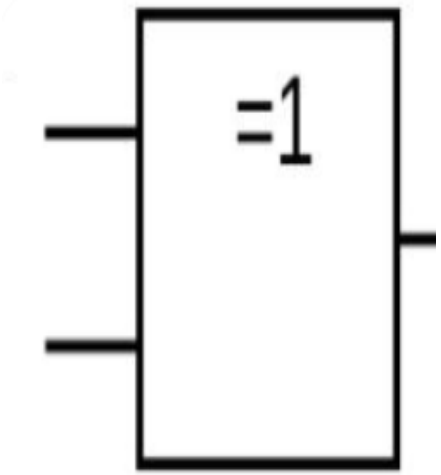
In einem C-Programm wird eine Variablem vom Datentyp 'float' mit 32 Bit definiert. Dabei gilt nach IEEE 754:

$$0100'0100'0000'1011'0000'0000'0000'0000_2 = +1.0859375 \cdot 2^9 = 556$$

- **Das MSB mit der Wertigkeit 2^{31} definiert das Vorzeichen dieser Zahl '556', welches mit dem Status '0' positiv definiert.**
- **Mit den 8 Bits '10001000' wird mit dem Biased-Code der Wert '9' vom Exponenten definiert.**
- **Mit den Bits '000'1011'0000'0000'0000'0000' wird die Mantisse nach '1.' definiert, welche in diesem Fall:**
  $0 \cdot 0.5 + 0 \cdot 0.25 + 0 \cdot 0.125 + 1 \cdot 0.0625 + 0 \cdot 0.03125 + 1 \cdot 0.015625 + 1 \cdot 0.0078125$ entspricht!
- **Der Wert '1.' wird immer bei der Verwendung dieses Wertes vom System dazugerechnet.**

=> (Zur Kontrolle dient z.B. https://www.ultimatesolver.com/de/ieee-754)

A2. Eine Alarm-Horn '**h**' soll dann tönen, wenn bei einer Prozessregelanlage das SW-Tool '**a**' aktiv und das SW-Tool '**c**' nicht aktiv ist oder wenn das SW-Tool '**c**' aktiv ist und dabei das SW-Tools '**a**' nicht aktiv oder dabei das SW-Tool '**b**' aktiv ist. Schreiben Sie dazu wie gelernt die Schaltfunktion und die Wertetabelle. Bauen Sie dann die entsprechende Logikschaltung mit WorkBench auf und testen alle möglichen Fälle, welche Sie dann natürlich auch dokumentieren! Ihr WorkBench-Schema kopieren Sie in Ihr pdf-File mit dem persönlichen Namen 'INF21bL name_vorname', welches Sie dann in der TEAMs-Aufgabe anhängen! <10P>



03 Die Logik und den Prozessor verstehen

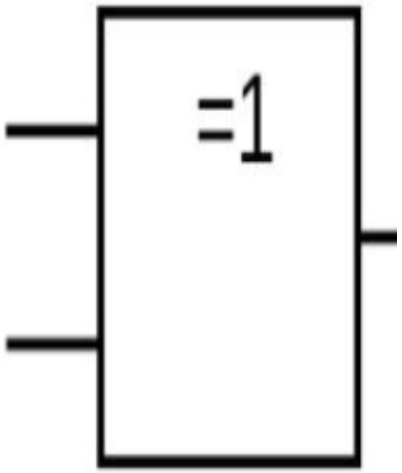
A2. Eine Alarm-Horn '**h**' soll dann tönen, wenn bei einer Prozessregelanlage das SW-Tool '**a**' aktiv und das SW-Tool '**c**' nicht aktiv ist oder wenn das SW-Tool '**c**' aktiv ist und dabei das SW-Tools '**a**' nicht aktiv oder dabei das SW-Tool '**b**' aktiv ist. Schreiben Sie dazu wie gelernt die Schaltfunktion und die Wertetabelle. Bauen Sie dann die entsprechende Logikschaltung mit WorkBench auf und testen alle möglichen Fälle, welche Sie dann natürlich auch dokumentieren! Ihr WorkBench-Schema kopieren Sie in Ihr pdf-File mit dem persönlichen Namen 'INF21bL name_vorname', welches Sie dann in der TEAMs-Aufgabe anhängen! <10P>

Schaltfunktion:

Wertetabelle:

Work Bench - Schaltung:

Test:



A2. Eine Alarm-Horn '**h**' soll dann tönen, wenn bei einer Prozessregelanlage das SW-Tool '**a**' aktiv und das SW-Tool '**c**' nicht aktiv ist oder wenn das SW-Tool '**c**' aktiv ist und dabei das SW-Tools '**a**' nicht aktiv oder dabei das SW-Tool '**b**' aktiv ist. Schreiben Sie dazu wie gelernt die Schaltfunktion und die Wertetabelle. Bauen Sie dann die entsprechende Logikschaltung mit WorkBench auf und testen alle möglichen Fälle, welche Sie dann natürlich auch dokumentieren! Ihr WorkBench-Schema kopieren Sie in Ihr pdf-File mit dem persönlichen Namen 'INF21bL name_vorname', welches Sie dann in der TEAMs-Aufgabe anhängen! <10P>

Schaltfunktion: $h = a\bar{c} + c(\bar{a} + b)$ 2P → Wertetabelle:

Work Bench - Schaltung:

Test:

Andere schrieben auch bei der Schaltfunktion in Bezug auf die Programmiersprache C:

h = a&&!c || c&&(!a || b) ***oder***

h = a AND !c OR c AND (!a OR b)

A2. Eine Alarm-Horn '**h**' soll dann tönen, wenn bei einer Prozessregelanlage das SW-Tool '**a**' aktiv und das SW-Tool '**c**' nicht aktiv ist oder wenn das SW-Tool '**c**' aktiv ist und dabei das SW-Tools '**a**' nicht aktiv oder dabei das SW-Tool '**b**' aktiv ist. Schreiben Sie dazu wie gelernt die Schaltfunktion und die Wertetabelle. Bauen Sie dann die entsprechende Logikschaltung mit WorkBench auf und testen alle möglichen Fälle, welche Sie dann natürlich auch dokumentieren! Ihr WorkBench-Schema kopieren Sie in Ihr pdf-File mit dem persönlichen Namen 'INF21bL name_vorname', welches Sie dann in der TEAMs-Aufgabe anhängen! <10P>

Schaltfunktion: $h = a\bar{c} + c(\bar{a} + b)$ 2P → **Wertetabelle:**

c	b	a	h

INPUT's | OUTPUT's

Work Bench - Schaltung:

Test:

Wie gelernt:
Alphabetische Ordnung

A2. Eine Alarm-Horn '**h**' soll dann tönen, wenn bei einer Prozessregelanlage das SW-Tool '**a**' aktiv und das SW-Tool '**c**' nicht aktiv ist oder wenn das SW-Tool '**c**' aktiv ist und dabei das SW-Tools '**a**' nicht aktiv oder dabei das SW-Tool '**b**' aktiv ist. Schreiben Sie dazu wie gelernt die Schaltfunktion und die Wertetabelle. Bauen Sie dann die entsprechende Logikschaltung mit WorkBench auf und testen alle möglichen Fälle, welche Sie dann natürlich auch dokumentieren! Ihr WorkBench-Schema kopieren Sie in Ihr pdf-File mit dem persönlichen Namen 'INF21bL name_vorname', welches Sie dann in der TEAMs-Aufgabe anhängen! <10P>

Schaltfunktion: $h = a\bar{c} + c(\bar{a} + b)$ 2P → **Wertetabelle:** 3P

Fall:	c	b	a	h
1			0	
2			1	
3			0	
4			1	
5			0	
6			1	
7			0	
8			1	

Work Bench - Schaltung:

Test:

A2. Eine Alarm-Horn '**h**' soll dann tönen, wenn bei einer Prozessregelanlage das SW-Tool '**a**' aktiv und das SW-Tool '**c**' nicht aktiv ist oder wenn das SW-Tool '**c**' aktiv ist und dabei das SW-Tools '**a**' nicht aktiv oder dabei das SW-Tool '**b**' aktiv ist. Schreiben Sie dazu wie gelernt die Schaltfunktion und die Wertetabelle. Bauen Sie dann die entsprechende Logikschaltung mit WorkBench auf und testen alle möglichen Fälle, welche Sie dann natürlich auch dokumentieren! Ihr WorkBench-Schema kopieren Sie in Ihr pdf-File mit dem persönlichen Namen 'INF21bL name_vorname', welches Sie dann in der TEAMs-Aufgabe anhängen! <10P>

Schaltfunktion: $h = a\bar{c} + c(\bar{a} + b)$ 2P → **Wertetabelle:** 3P

Fall:	c	b	a	h
1	0	0	0	
2	0	0	1	
3	0	1	0	
4	0	1	1	
5	1	0	0	
6	1	0	1	
7	1	1	0	
8	1	1	1	

Work Bench - Schaltung:

Test:

A2. Eine Alarm-Horn '**h**' soll dann tönen, wenn bei einer Prozessregelanlage das SW-Tool '**a**' aktiv und das SW-Tool '**c**' nicht aktiv ist oder wenn das SW-Tool '**c**' aktiv ist und dabei das SW-Tools '**a**' nicht aktiv oder dabei das SW-Tool '**b**' aktiv ist. Schreiben Sie dazu wie gelernt die Schaltfunktion und die Wertetabelle. Bauen Sie dann die entsprechende Logikschaltung mit WorkBench auf und testen alle möglichen Fälle, welche Sie dann natürlich auch dokumentieren! Ihr WorkBench-Schema kopieren Sie in Ihr pdf-File mit dem persönlichen Namen 'INF21bL name_vorname', welches Sie dann in der TEAMs-Aufgabe anhängen! <10P>

Schaltfunktion: $h = a\bar{c} + c(\bar{a} + b)$ 2P → **Wertetabelle:** 3P

Fall:	c	b	a	h
1	0	0	0	
2	0	0	1	
3	0	1	0	
4	0	1	1	
5	1	0	0	
6	1	0	1	
7	1	1	0	
8	1	1	1	

Work Bench - Schaltung:

Test:

A2. Eine Alarm-Horn '**h**' soll dann tönen, wenn bei einer Prozessregelanlage das SW-Tool '**a**' aktiv und das SW-Tool '**c**' nicht aktiv ist oder wenn das SW-Tool '**c**' aktiv ist und dabei das SW-Tools '**a**' nicht aktiv oder dabei das SW-Tool '**b**' aktiv ist. Schreiben Sie dazu wie gelernt die Schaltfunktion und die Wertetabelle. Bauen Sie dann die entsprechende Logikschaltung mit WorkBench auf und testen alle möglichen Fälle, welche Sie dann natürlich auch dokumentieren! Ihr WorkBench-Schema kopieren Sie in Ihr pdf-File mit dem persönlichen Namen 'INF21bL name_vorname', welches Sie dann in der TEAMs-Aufgabe anhängen! <10P>

Schaltfunktion: $h = a\bar{c} + c(\bar{a} + b)$ 2P → **Wertetabelle:** 3P

Fall:	c	b	a	h
1	0	0	0	
2	0	0	1	1
3	0	1	0	
4	0	1	1	1
5	1	0	0	1
6	1	0	1	
7	1	1	0	1
8	1	1	1	1

Alle Statuswerte h = 1 aus der Schaltfunktion in die Werte-tabelle eintragen!

Work Bench - Schaltung:

Test:

A2. Eine Alarm-Horn '**h**' soll dann tönen, wenn bei einer Prozessregelanlage das SW-Tool '**a**' aktiv und das SW-Tool '**c**' nicht aktiv ist oder wenn das SW-Tool '**c**' aktiv ist und dabei das SW-Tools '**a**' nicht aktiv oder dabei das SW-Tool '**b**' aktiv ist. Schreiben Sie dazu wie gelernt die Schaltfunktion und die Wertetabelle. Bauen Sie dann die entsprechende Logikschaltung mit WorkBench auf und testen alle möglichen Fälle, welche Sie dann natürlich auch dokumentieren! Ihr WorkBench-Schema kopieren Sie in Ihr pdf-File mit dem persönlichen Namen 'INF21bL name_vorname', welches Sie dann in der TEAMs-Aufgabe anhängen! <10P>

Schaltfunktion: $h = a\bar{c} + c(\bar{a} + b)$ 2P

Wertetabelle: 3P

Fall:	c	b	a	h
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	1
5	1	0	0	1
6	1	0	1	0
7	1	1	0	1
8	1	1	1	1

Work Bench - Schaltung:

Test:

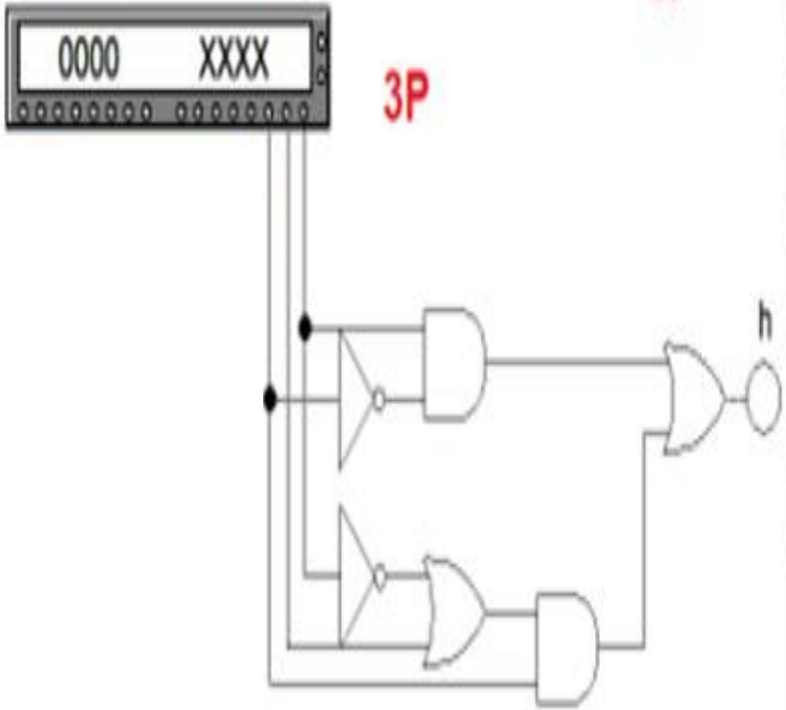
A2. Eine Alarm-Horn '**h**' soll dann tönen, wenn bei einer Prozessregelanlage das SW-Tool '**a**' aktiv und das SW-Tool '**c**' nicht aktiv ist oder wenn das SW-Tool '**c**' aktiv ist und dabei das SW-Tools '**a**' nicht aktiv oder dabei das SW-Tool '**b**' aktiv ist. Schreiben Sie dazu wie gelernt die Schaltfunktion und die Wertetabelle. Bauen Sie dann die entsprechende Logikschaltung mit WorkBench auf und testen alle möglichen Fälle, welche Sie dann natürlich auch dokumentieren! Ihr WorkBench-Schema kopieren Sie in Ihr pdf-File mit dem persönlichen Namen 'INF21bL name_vorname', welches Sie dann in der TEAMs-Aufgabe anhängen! <10P>

Schaltfunktion: $h = a\bar{c} + c(\bar{a} + b)$ 2P

Wertetabelle: 3P

Fall:	c	b	a	h
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	1
5	1	0	0	1
6	1	0	1	0
7	1	1	0	1
8	1	1	1	1

Work Bench - Schaltung: 3P



Test:

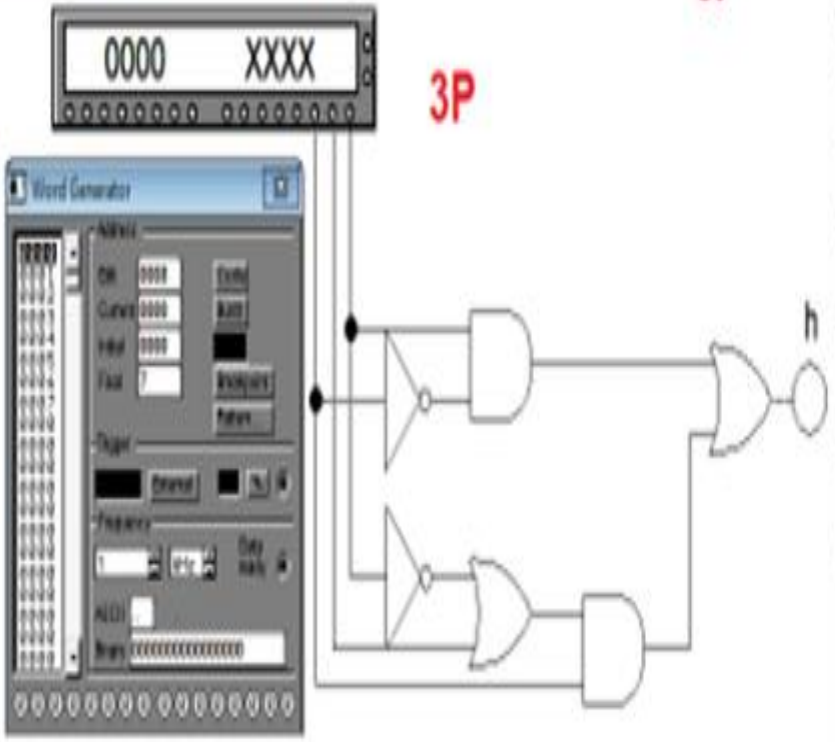
A2. Eine Alarm-Horn '**h**' soll dann tönen, wenn bei einer Prozessregelanlage das SW-Tool '**a**' aktiv und das SW-Tool '**c**' nicht aktiv ist oder wenn das SW-Tool '**c**' aktiv ist und dabei das SW-Tools '**a**' nicht aktiv oder dabei das SW-Tool '**b**' aktiv ist. Schreiben Sie dazu wie gelernt die Schaltfunktion und die Wertetabelle. Bauen Sie dann die entsprechende Logikschaltung mit WorkBench auf und testen alle möglichen Fälle, welche Sie dann natürlich auch dokumentieren! Ihr WorkBench-Schema kopieren Sie in Ihr pdf-File mit dem persönlichen Namen 'INF21bL name_vorname', welches Sie dann in der TEAMs-Aufgabe anhängen! <10P>

Schaltfunktion: $h = a\bar{c} + c(\bar{a} + b)$ 2P

Wertetabelle: 3P

Fall:	c	b	a	h
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	1
5	1	0	0	1
6	1	0	1	0
7	1	1	0	1
8	1	1	1	1

Work Bench - Schaltung: 3P



Test:

Der Word-Generator darf nicht fehlen, denn dieser zeigt ja dann auch die 8 Testfälle!

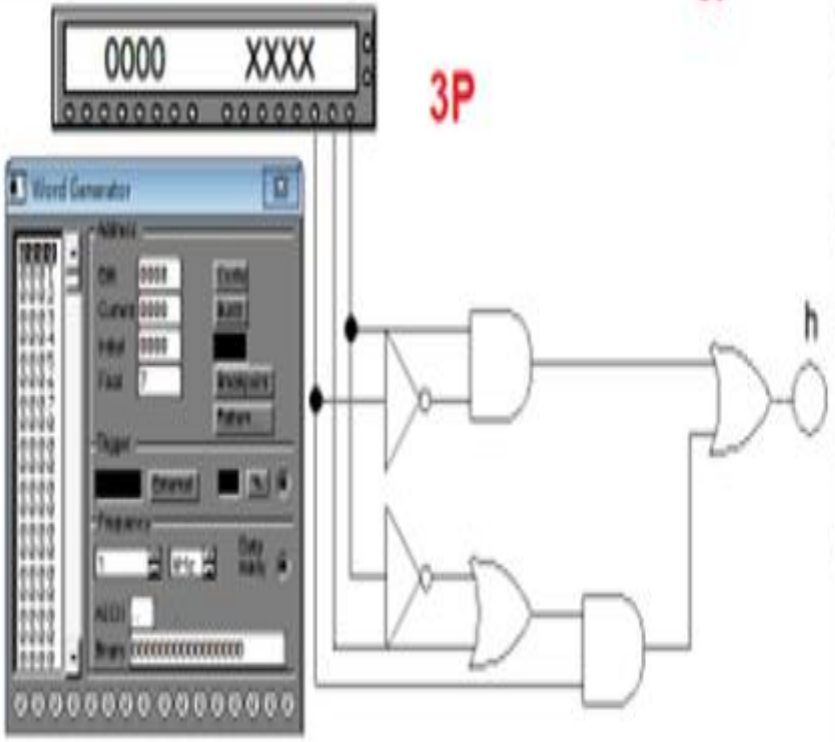
A2. Eine Alarm-Horn '**h**' soll dann tönen, wenn bei einer Prozessregelanlage das SW-Tool '**a**' aktiv und das SW-Tool '**c**' nicht aktiv ist oder wenn das SW-Tool '**c**' aktiv ist und dabei das SW-Tools '**a**' nicht aktiv oder dabei das SW-Tool '**b**' aktiv ist. Schreiben Sie dazu wie gelernt die Schaltfunktion und die Wertetabelle. Bauen Sie dann die entsprechende Logikschaltung mit WorkBench auf und testen alle möglichen Fälle, welche Sie dann natürlich auch dokumentieren! Ihr WorkBench-Schema kopieren Sie in Ihr pdf-File mit dem persönlichen Namen 'INF21bL name_vorname', welches Sie dann in der TEAMs-Aufgabe anhängen! <10P>

Schaltfunktion: $h = a\bar{c} + c(\bar{a} + b)$ 2P

Wertetabelle: 3P

Fall:	c	b	a	h
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	1
5	1	0	0	1
6	1	0	1	0
7	1	1	0	1
8	1	1	1	1

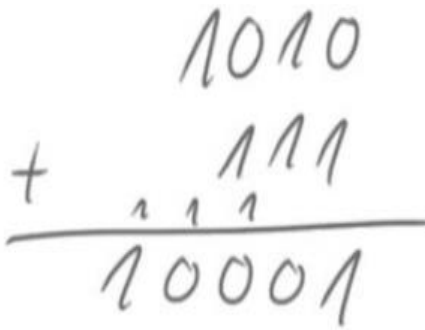
Work Bench - Schaltung: 3P



Test: 2P *Alle 8 rechts definierten Fälle ergeben mit der rechts definierten WorkBench-Schaltung den richtigen Status am Ausgang! Damit ist die Schaltung i.O.!*

A3. **a)** Beschreiben Sie klar und deutlich den Unterschied zwischen einem XOR- und einem NOR-Baustein. Definieren Sie dazu die Wertetabelle und das Symbol jedes dieser Bausteine mit zwei Eingängen! <2P>

b) Beschreiben Sie von zwei Faktoren 'a' und 'b' den Ablauf einer Multiplikation bei einem Co-Prozessor in einem Tablet genau und verständlich! <2P>



02 Arithmetische und logische Grundoperationen bin...

c) Wie erfolgt bei einem Mikroprozessor die Addition von den beiden Summanden 59 und 49 genau? Zeigen Sie dies am klar dargestellten Additionsvorgang mit der resultierenden, dualen Summe! <2P>

A3. **a)** Beschreiben Sie klar und deutlich den Unterschied zwischen einem XOR- und einem NOR-Baustein. Definieren Sie dazu die Wertetabelle und das Symbol jedes dieser Bausteine mit zwei Eingängen! <2P>

e_2	e_1	a_{NOR}	a_{XOR}

b) Beschreiben Sie von zwei Faktoren 'a' und 'b' den Ablauf einer Multiplikation bei einem Co-Prozessor in einem Tablet genau und verständlich! <2P>

```
  1010
+  111
  ...
 10001
```

02 Arithmetische und logische Grundoperationen bin...

c) Wie erfolgt bei einem Mikroprozessor die Addition von den beiden Summanden 59 und 49 genau? Zeigen Sie dies am klar dargestellten Additionsvorgang mit der resultierenden, dualen Summe! <2P>

A3. **a)** Beschreiben Sie klar und deutlich den Unterschied zwischen einem XOR- und einem NOR-Baustein. Definieren Sie dazu die Wertetabelle und das Symbol jedes dieser Bausteine mit zwei Eingängen! <2P>

e_2	e_1	a_{NOR}	a_{XOR}
0	0		
0	1		
1	0		
1	1		

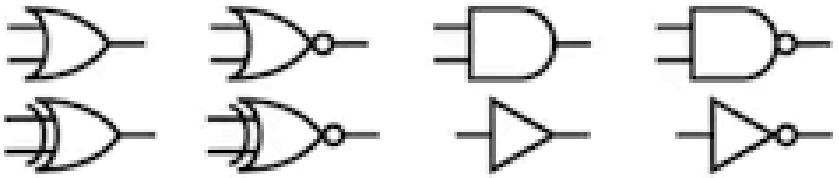
b) Beschreiben Sie von zwei Faktoren 'a' und 'b' den Ablauf einer Multiplikation bei einem Co-Prozessor in einem Tablet genau und verständlich! <2P>

c) Wie erfolgt bei einem Mikroprozessor die Addition von den beiden Summanden 59 und 49 genau? Zeigen Sie dies am klar dargestellten Additionsvorgang mit der resultierenden, dualen Summe! <2P>

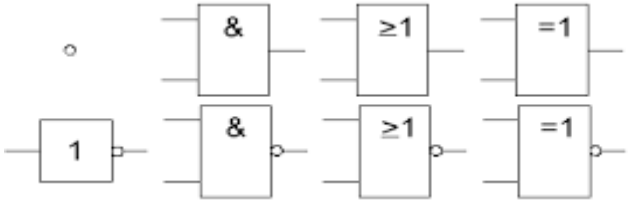
A3. **a)** Beschreiben Sie klar und deutlich den Unterschied zwischen einem XOR- und einem NOR-Baustein. Definieren Sie dazu die Wertetabelle und das Symbol jedes dieser Bausteine mit zwei Eingängen! <2P>

e_2	e_1	a_{NOR}	a_{XOR}
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	0

b) Beschreiben Sie von zwei Faktoren 'a' und 'b' den Ablauf einer Multiplikation bei einem Co-Prozessor in einem Tablet genau und verständlich! <2P>



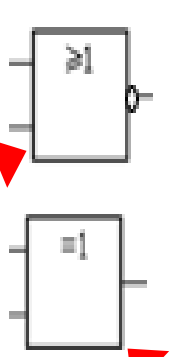
Amerkianische Norm!



Unsere geltende DI-Norm!

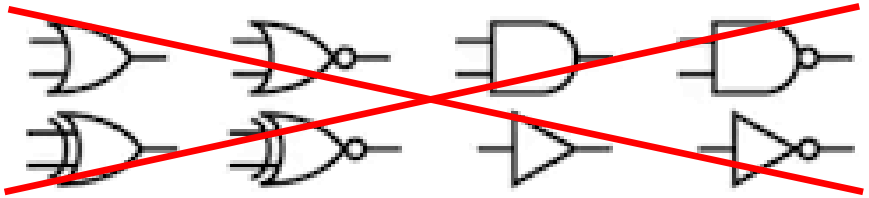
c) Wie erfolgt bei einem Mikroprozessor die Addition von den beiden Summanden 59 und 49 genau? Zeigen Sie dies am klar dargestellten Additionsvorgang mit der resultierenden, dualen Summe! <2P>

A3. **a)** Beschreiben Sie klar und deutlich den Unterschied zwischen einem XOR- und einem NOR-Baustein. Definieren Sie dazu die Wertetabelle und das Symbol jedes dieser Bausteine mit zwei Eingängen! <2P>

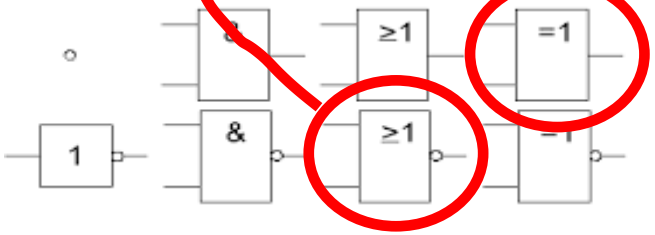


e_2	e_1	a_{NOR}	a_{XOR}
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	0

b) Beschreiben Sie von zwei Faktoren 'a' und 'b' den Ablauf einer Multiplikation bei einem Co-Prozessor in einem Tablet genau und verständlich! <2P>



Amerkianische Norm!

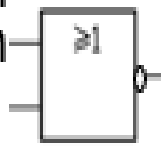


Unsere geltende DI-Norm!

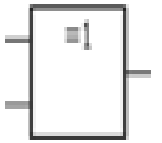
c) Wie erfolgt bei einem Mikroprozessor die Addition von den beiden Summanden 59 und 49 genau? Zeigen Sie dies am klar dargestellten Additionsvorgang mit der resultierenden, dualen Summe! <2P>

A3. a) Beschreiben Sie klar und deutlich den Unterschied zwischen einem XOR- und einem NOR-Baustein. Definieren Sie dazu die Wertetabelle und das Symbol jedes dieser Bausteine mit zwei Eingängen! <2P>

Der Ausgang eines NOR-Bausteins hat dann den Status '1' wenn alle Eingänge den Status '0' haben. Bei allen andern Fällen hat der Ausgaben den Status 0. Damit ergibt sich das Symbol '≥1' mit invertierten Ausgang.



Der Ausgang eins XOR-Baustein hat nur dann den Status 1, wenn nur ein Eingang den Status '1' hat. Damit ergibt sich das Symbol '=1' beim XOR-Baustein.



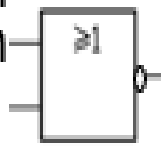
e_2	e_1	a_{NOR}	a_{XOR}
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	0

b) Beschreiben Sie von zwei Faktoren 'a' und 'b' den Ablauf einer Multiplikation bei einem Co-Prozessor in einem Tablet genau und verständlich! <2P>

c) Wie erfolgt bei einem Mikroprozessor die Addition von den beiden Summanden 59 und 49 genau? Zeigen Sie dies am klar dargestellten Additionsvorgang mit der resultierenden, dualen Summe! <2P>

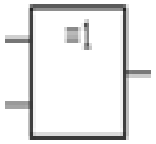
A3. a) Beschreiben Sie klar und deutlich den Unterschied zwischen einem XOR- und einem NOR-Baustein. Definieren Sie dazu die Wertetabelle und das Symbol jedes dieser Bausteine mit zwei Eingängen! <2P>

Der Ausgang eines NOR-Bausteins hat dann den Status '1' wenn alle Eingänge den Status '0' haben. Bei allen andern Fällen hat der Ausgaben den Status 0. Damit ergibt sich das Symbol '≥1' mit invertierten Ausgang.



e_2	e_1	a_{NOR}	a_{XOR}
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	0

Der Ausgang eins XOR-Baustein hat nur dann den Status 1, wenn nur ein Eingang den Status '1' hat. Damit ergibt sich das Symbol '=1' beim XOR-Baustein.



b) Beschreiben Sie von zwei Faktoren 'a' und 'b' den Ablauf einer Multiplikation bei einem Co-Prozessor in einem Tablet genau und verständlich! <2P>

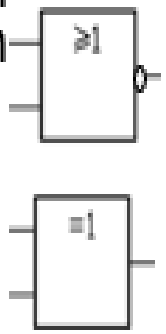
Der Co-Prozessor eines Tablets wird dabei bei jedem Status '1' vom Faktor 'a' den Faktor 'b' als Summanden verwenden, beim Status '0' vom Faktor 'a' einfach den Wert '0' als Summanden verwenden. Begonnen wird dabei mit dem LSB vom Faktor 'a', bei jedem weitern, höheren Summandenbit wird dann der jeweilige Teilsummand um eine Stelle nach links geschoben. Der Faktor ergibt sich dann aus der Summe aller Teilsummanden.

c) Wie erfolgt bei einem Mikroprozessor die Addition von den beiden Summanden 59 und 49 genau? Zeigen Sie dies am klar dargestellten Additionsvorgang mit der resultierenden, dualen Summe! <2P>

A3. a) Beschreiben Sie klar und deutlich den Unterschied zwischen einem XOR- und einem NOR-Baustein. Definieren Sie dazu die Wertetabelle und das Symbol jedes dieser Bausteine mit zwei Eingängen! <2P>

Der Ausgang eines NOR-Bausteins hat dann den Status '1' wenn alle Eingänge den Status '0' haben. Bei allen andern Fällen hat der Ausgaben den Status 0. Damit ergibt sich das Symbol '≥1' mit invertierten Ausgang.

Der Ausgang eins XOR-Baustein hat nur dann den Status 1, wenn nur ein Eingang den Status '1' hat. Damit ergibt sich das Symbol '=1' beim XOR-Baustein.



e_2	e_1	a_{NOR}	a_{XOR}
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	0

b) Beschreiben Sie von zwei Faktoren 'a' und 'b' den Ablauf einer Multiplikation bei einem Co-Prozessor in einem Tablet genau und verständlich! <2P>

Der Co-Prozessor eines Tablets wird dabei bei jedem Status '1' vom Faktor 'a' den Faktor 'b' als Summanden verwenden, beim Status '0' vom Faktor 'a' einfach den Wert '0' als Summanden verwenden. Begonnen wird dabei mit dem LSB vom Faktor 'a', bei jedem weitern, höheren Summandenbit wird dann der jeweilige Teilsummand um eine Stelle nach links geschoben. Der Faktor ergibt sich dann aus der Summe aller Teilsummanden.

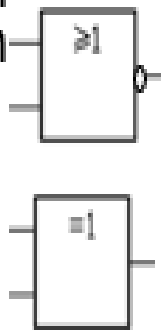
c) Wie erfolgt bei einem Mikroprozessor die Addition von den beiden Summanden 59 und 49 genau? Zeigen Sie dies am klar dargestellten Additionsvorgang mit der resultierenden, dualen Summe! <2P>

59 = $011'1011_2$
49 = $+011'0001_2$

A3. a) Beschreiben Sie klar und deutlich den Unterschied zwischen einem XOR- und einem NOR-Baustein. Definieren Sie dazu die Wertetabelle und das Symbol jedes dieser Bausteine mit zwei Eingängen! <2P>

Der Ausgang eines NOR-Bausteins hat dann den Status '1' wenn alle Eingänge den Status '0' haben. Bei allen andern Fällen hat der Ausgaben den Status 0. Damit ergibt sich das Symbol '≥1' mit invertierten Ausgang.

Der Ausgang eins XOR-Baustein hat nur dann den Status 1, wenn nur ein Eingang den Status '1' hat. Damit ergibt sich das Symbol '=1' beim XOR-Baustein.



e_2	e_1	a_{NOR}	a_{XOR}
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	0

b) Beschreiben Sie von zwei Faktoren 'a' und 'b' den Ablauf einer Multiplikation bei einem Co-Prozessor in einem Tablet genau und verständlich! <2P>

Der Co-Prozessor eines Tablets wird dabei bei jedem Status '1' vom Faktor 'a' den Faktor 'b' als Summanden verwenden, beim Status '0' vom Faktor 'a' einfach den Wert '0' als Summanden verwenden. Begonnen wird dabei mit dem LSB vom Faktor 'a', bei jedem weitern, höheren Summandenbit wird dann der jeweilige Teilsummand um eine Stelle nach links geschoben. Der Faktor ergibt sich dann aus der Summe aller Teilsummanden.

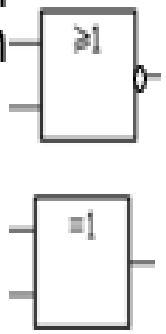
c) Wie erfolgt bei einem Mikroprozessor die Addition von den beiden Summanden 59 und 49 genau? Zeigen Sie dies am klar dargestellten Additionsvorgang mit der resultierenden, dualen Summe! <2P>

$$\begin{array}{rl} 59 = & 011'1011_2 \\ 49 = & +011'0001_2 \\ \hline .08 = & +0110'1100_2 \end{array}$$

A3. **a)** Beschreiben Sie klar und deutlich den Unterschied zwischen einem XOR- und einem NOR-Baustein. Definieren Sie dazu die Wertetabelle und das Symbol jedes dieser Bausteine mit zwei Eingängen! <2P>

Der Ausgang eines NOR-Bausteins hat dann den Status '1' wenn alle Eingänge den Status '0' haben. Bei allen andern Fällen hat der Ausgaben den Status 0. Damit ergibt sich das Symbol '≥1' mit invertierten Ausgang.

Der Ausgang eins XOR-Baustein hat nur dann den Status 1, wenn nur ein Eingang den Status '1' hat. Damit ergibt sich das Symbol '=1' beim XOR-Baustein.



e_2	e_1	a_{NOR}	a_{XOR}
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	0

b) Beschreiben Sie von zwei Faktoren 'a' und 'b' den Ablauf einer Multiplikation bei einem Co-Prozessor in einem Tablet genau und verständlich! <2P>

Der Co-Prozessor eines Tablets wird dabei bei jedem Status '1' vom Faktor 'a' den Faktor 'b' als Summanden verwenden, beim Status '0' vom Faktor 'a' einfach den Wert '0' als Summanden verwenden. Begonnen wird dabei mit dem LSB vom Faktor 'a', bei jedem weitern, höheren Summandenbit wird dann der jeweilige Teilsummand um eine Stelle nach links geschoben. Der Faktor ergibt sich dann aus der Summe aller Teilsummanden.

c) Wie erfolgt bei einem Mikroprozessor die Addition von den beiden Summanden 59 und 49 genau? Zeigen Sie dies am klar dargestellten Additionsvorgang mit der resultierenden, dualen Summe! <2P>

$$\begin{array}{rl} 59 = & 011'1011_2 \\ 49 = & +011'0001_2 \\ \hline .08 = & +0110'1100_2 \end{array}$$

Die Duale Addition links zeigt klar und deutlich den Additionsvorgang bei einem Mikroprozessor!