

# Digitaltechnik

## Übungsaufgaben 1

24. 10. 11

### Problemlösung mit Digitalschaltungen anhand elementarer Beispiele

Die Problemlösung ergibt sich, indem man die gewünschte Funktionsweise möglichst exakt unter Verwendung von Ausdrücken der Art UND – ODER – NICHT beschreibt oder indem man die Wahrheitstabelle aufstellt, also alle Möglichkeiten durchnimmt und sich überlegt, was dabei jeweils herauskommen muß. Keine Vereinfachung, keine Beschränkung auf bestimmte Gattertypen.

- Entwerfen Sie die Blinksteuerung für ein Kraftfahrzeug (Abb. 1). Gegeben sind die Steuersignale L, R und W vom Blinkhebel und von der Warnblinktaste. Anzusteuern sind:
  - der Blinkimpulsgeber (BIG),
  - die rechten Blinkleuchten (BLR),
  - die linken Blinkleuchten (BLL).

Alle Signale sind aktiv High. Funktionen:

- L aktiv: links blinken,
- R aktiv: rechts blinken,
- W aktiv: links und rechts blinken.

Die Blinkimpulse (BI) kommen vom Blinkimpulsgeber, der über das Steuersignal BIG immer dann einzuschalten ist, wenn geblinkt werden soll. Die Schaltung ist mit elementaren Gattern (UND, ODER, NAND, NOR) und Negatoren aufzubauen.

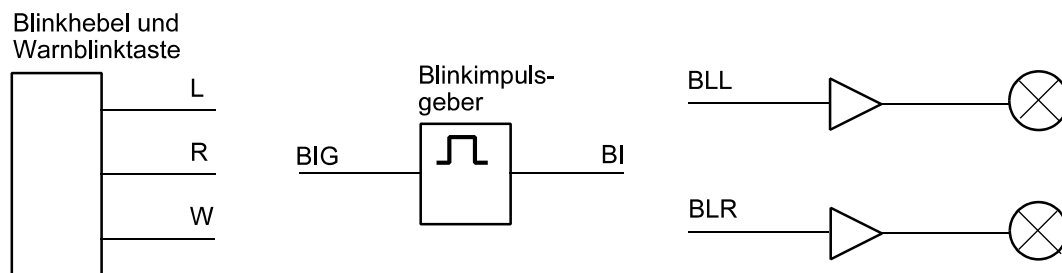


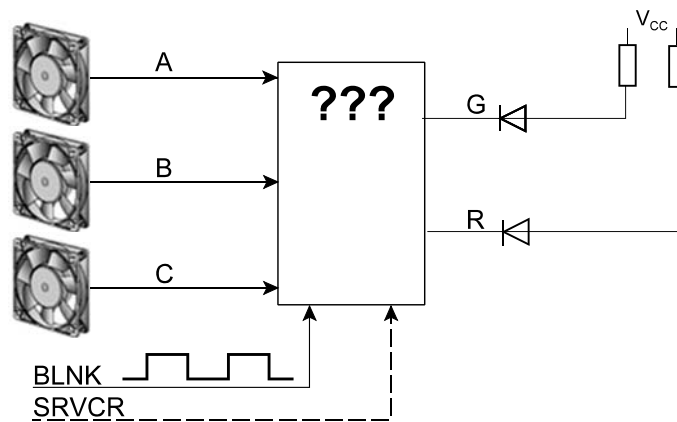
Abb. 1

- In einem Gehäuse sind drei Lüfter A, B, C angeordnet. Jeder Lüfter hat einen Sensor, der ein Low-Signal liefert, wenn der betreffende Lüfter arbeitet. Ist der Lüfter ausgefallen, liefert der Sensor ein High-Signal. Entwerfen Sie eine Überwachungsschaltung (Abb. 2), die zwei Kontroll-LEDs G, R folgendermaßen ansteuert:
  - die grüne LED (G) soll leuchten, wenn alle drei Lüfter arbeiten,
  - die rote LED (R) soll leuchten, wenn einer der Lüfter (gleich welcher) ausgefallen ist,
  - die rote LED (R) soll blinken, wenn zwei oder alle drei Lüfter ausgefallen sind. Entsprechende Blinkimpulse sind verfügbar (Signal BLNK).

Ansteuerung der LEDs: aktiv Low. Bauelementebasis: Gatter nach eigener Wahl. Minimierung ist nicht erforderlich.

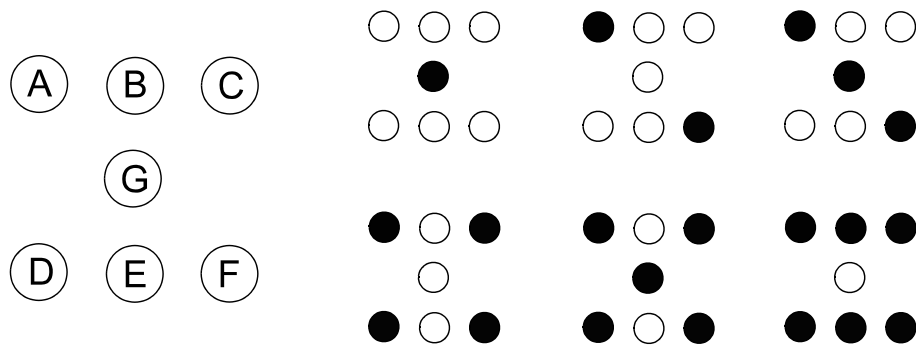
*Zusatzaufgabe (zu lösen nach Behandlung der Latches und Flipflops):*

Die Fehleranzeige soll solange gehalten werden, bis der Servicetechniker einen entsprechenden Schalter betätigt (Kontakt SRVCR). Der Kontakt wirkt aktiv Low.



**Abb. 2**

3. Es ist ein elektronischer Würfel zu bauen (Abb. 3). Als Grundlage steht ein Zähler zur Verfügung, der zyklisch Einzelsignale EINS, ZWEI usw. bis SECHS abgibt. Mit anderen Worten: Es ist eine kombinatorische Schaltung zu entwerfen (mit beliebigen Gattern), die mit den Signalen EINS bis SECHS die LEDs A bis G je nach Zählerstand erregt (Ansteuerung aktiv high).



**Abb. 3**

## Problemlösung mittels Boolescher Algebra; Dokumentation von Digitalschaltungen

Es geht um das Lesen von Schaltplänen (auch in Kabelbaumdarstellung) und um die wechselseitige Wandlung zwischen Schaltbild (Gatterschaltung) und Schaltgleichung.

- Abb. 1 zeigt den Schaltplan eines kombinatorischen Netzwerks. Geben Sie die Schaltgleichungen für beide Ausgangssignale A, B an.

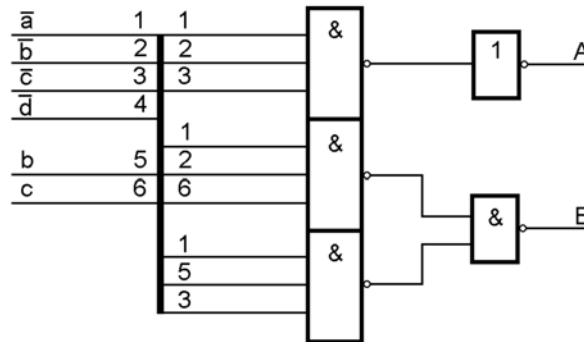


Abb. 1

- An kombinatorischen Schaltungen messen Sie Signalbelegungen (Abb. 2). Ist das in Ordnung? Tip: Gehen Sie Gatter für Gatter durch und ermitteln Sie so Schritt für Schritt die Werte, die sich bei ordnungsgemäßer Funktion ergeben müssten. L = 0, H = 1.

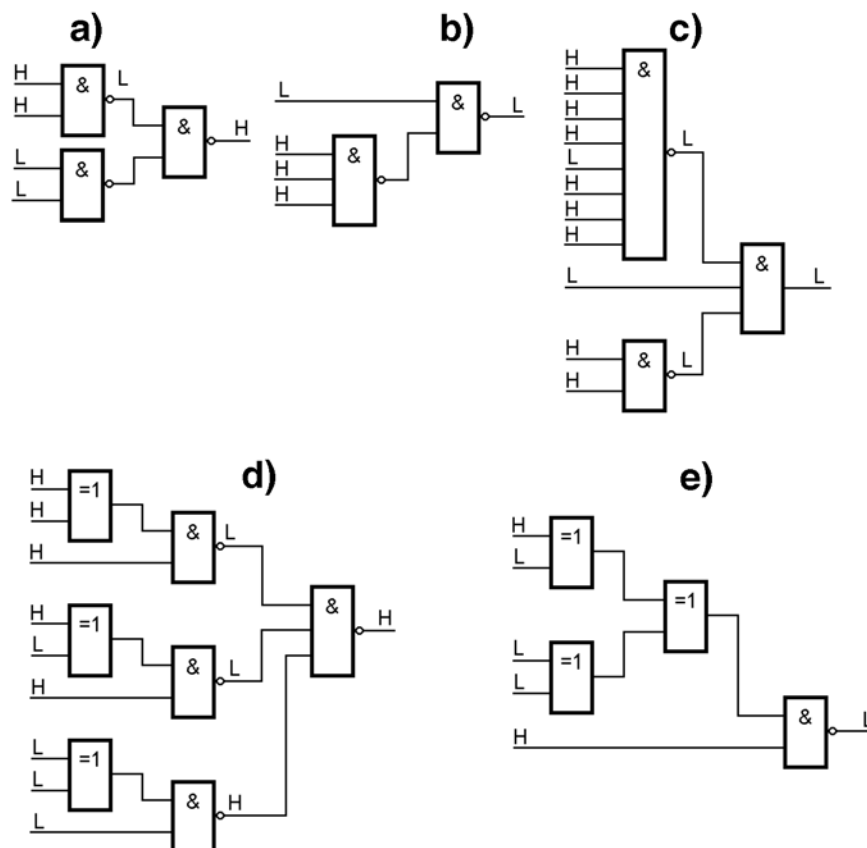


Abb. 2

3. Am Eingang einer Schaltung messen Sie die dargestellten Signalverläufe (Abb. 3). Was erwarten Sie am Ausgang? (Signalverlauf in idealisierter Form in das Diagramm einzeichnen.)

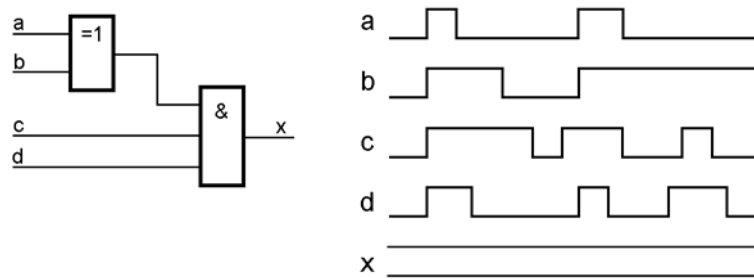


Abb. 3

4. In einer Schaltung wird ein Schaltkreis über ein kombinatorisches Netzwerk zurückgesetzt. Sie messen die dargestellten Signalverläufe (Abb. 4). Ist das in Ordnung? Kennzeichnen und erläutern Sie ggf. gefundene Fehler.

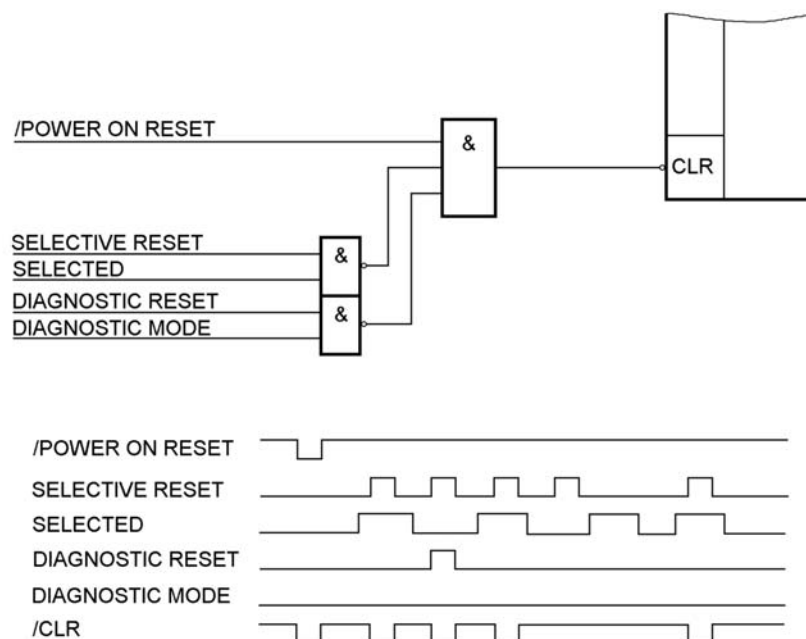


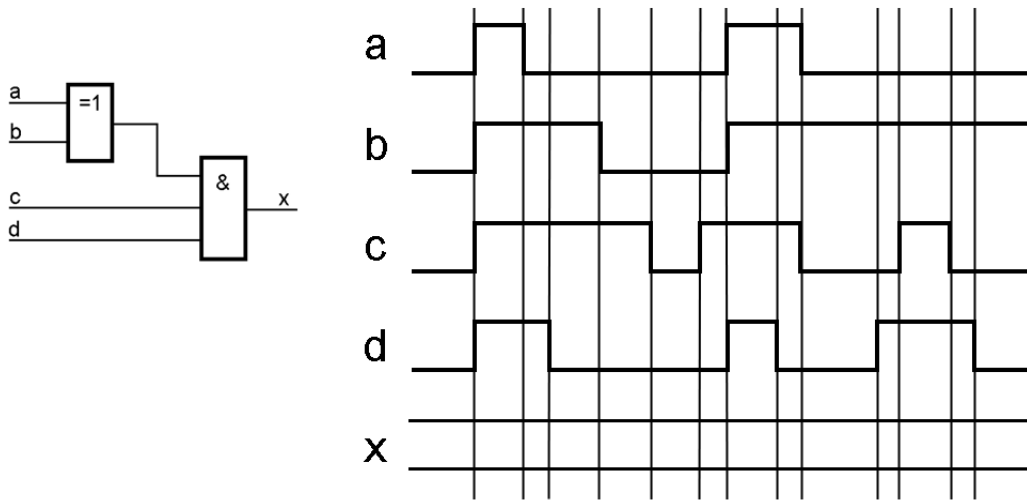
Abb. 4

5. Wir beziehen uns auf Abb. 5.
- a) Gelten folgende Aussagen? (Antwort mit "Ja" oder "Nein".)
    1. Funktionselement 3 ist mit Funktionselement 4 verbunden.
    2. Funktionselement 2 ist mit Funktionselement 7 verbunden.
    3. Funktionselement 4 ist mit Funktionselement 10 verbunden.
    4. Funktionselement 3 ist mit Funktionselement 8 verbunden.
  - b) Das Signal Nr. 2, das an Funktionselement 3 seinen Ursprung hat, ist zu verfolgen. An welchen anderen Funktionselementen messen Sie? (Nummern angeben.)

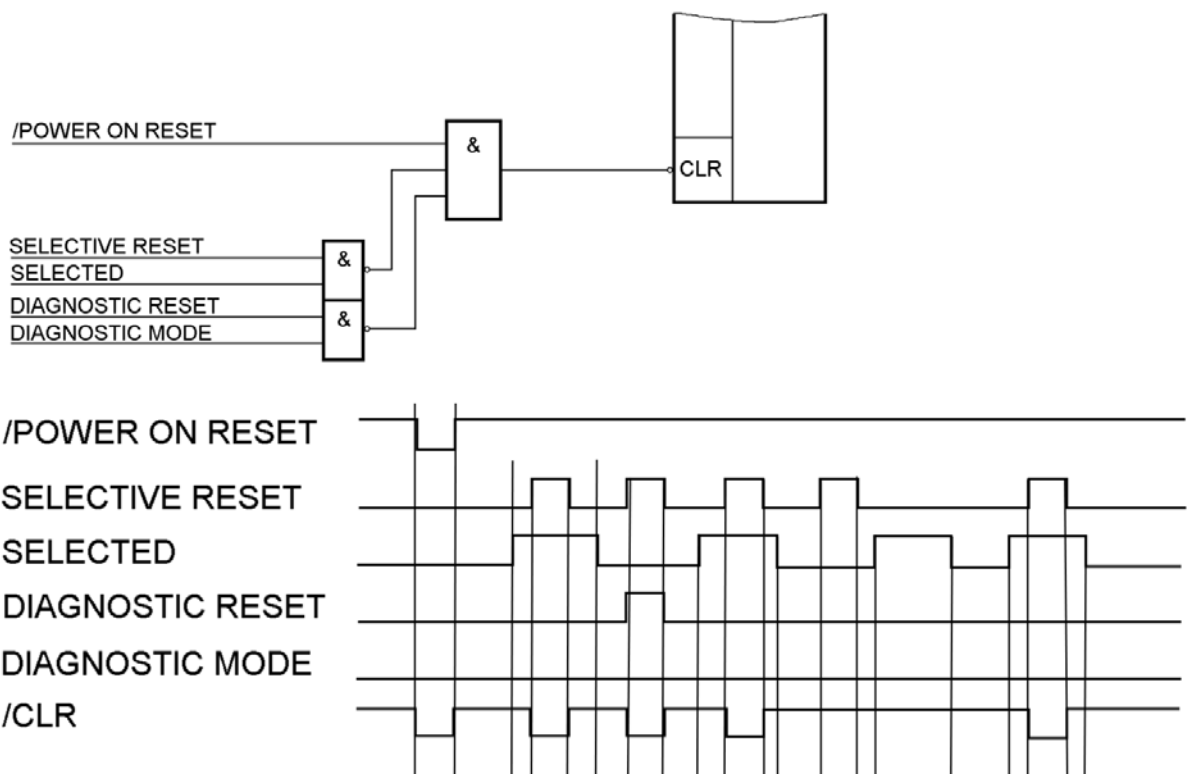


Hilfe zu Aufgabe 3:

Prinzip der Aufgaben 3 und 4: an allen Signalflanken Hilfslinien ziehen und die einzelnen Abschnitte daraufhin untersuchen, ob die Ausgangssignale o.k. sind oder nicht.



Hilfe zu Aufgabe 4:



## Binär- und Hexadezimalzahlen

Prinzip des Binärsystems. Umrechnungen dezimal – binär – hexadezimal (in alle Richtungen). Dezimal höchstens dreistellig; Zweierpotenzen bis  $2^9 = 512$ . (Das genügt, um die Dotted Decimal Notation der IP-Adressen zu verstehen.)

Hinweis: Der Rechner im Zubehör von Windows

Er genügt in den weitaus meisten Fällen. Wichtig: auf die “wissenschaftliche” Ansicht umstellen.

Rechenbeispiele:

1. Stellen Sie die folgenden Bitfolgen hexadezimal dar:
  - a) 1101 0110B
  - b) 01 1011B
  - c) 10110011 10001110B
2. Geben Sie folgende hexadezimal dargestellte Werte binär an:
  - a) 3F2H
  - b) 22CCH
  - c) 127H
3. Eine Einrichtung soll an einem Systembus unter der Adresse C028H erreichbar sein. Die Belegung der drei höchstwertigen Adreßbits ist an einem Drehschalter einzustellen, dessen 8 Stellungen mit 0...7 beschriftet sind. Welchen Wert stellen Sie ein?
4. Geben Sie die IP-Adresse 164.244.50.111 in hexadezimaler und in binärer Darstellung an.
5. Aus der meßtechnischen Analyse eines Datenstroms erkennen Sie folgendes Bitmuster, das offensichtlich als IP-Adresse dient: 1101 0110 1110 0011 0100 0011 1110 1000B. Geben Sie die IP-Adresse in Dotted Decimal Notation an.

*Lösung zu 1.*

Von rechts außen an die Binärangabe in Vierergruppen unterteilen. Freie höherwertige Bitpositionen mit Nullen auffüllen. Entsprechend ergibt sich:

- a) 1101 0110B = D6H
- b) 01 1011B = 1BH
- c) 10110011 10001110B = B38EH

*Lösung zu 2.*

Die Hexadezimalstellen einzeln wandeln. Wenn nicht von vornherein bekannt ist, daß höchstwertige Bits weggelassen werden können (z. B. bei der vierstelligen Hex-Angabe eines 14-Bit-Wortes), so sollten Sie die führenden Nullen mit angeben. Es ergibt sich:

- a) 3F2H = 0011 1111 0010B
- b) 22CCH = 0010 0010 1100 1100B
- c) 127H = 0001 0010 0111B

*Lösung zu 3.*

Es interessiert nur die höchstwertige Hexadezimalstelle CH = 1100B. Hiervon brauchen wir die drei höchstwertigen Bits 110B = 6. Also ist Schalterstellung 6 zu wählen.

*Lösung zu 4.*

164.244.50.111 = A4 F4 32 6FH = 1010 0100 1111 0100 0011 0010 0110 1111B.

*Lösung zu 5.*

1101 0110 1110 0011 0100 0011 1110 1000B = 161.227.67.232

*Rechenhilfe zu 4. und 5.:* die Werte einzeln in den Windows-Taschenrechner eintippen und entsprechend wandeln. *Achtung:* binäre Ergebnisse enden mit der höchstwertigen Eins und sind deshalb ggf.auf die jeweilige Stellenzahl (z. B. 8 Stellen) zu ergänzen (links mit Nullen auffüllen)

## Kombinatorische Schaltungen entwerfen (1)

Wahrheitstabellen, Boolesche Gleichungen, Grundgatter und Gatterschaltungen sind bekannt. Jetzt wird dieses Wissen im Verbund angewendet. Vereinfachung ist zunächst nicht erforderlich.

- Entwerfen Sie eine Schaltung, die die folgende Schaltgleichung verwirklicht. Es sind NAND-Gatter mit zwei Eingängen zu verwenden. Die negierten Signale können als gegeben vorausgesetzt werden.

$$D = \bar{A} \bar{B} C \vee A \bar{B} C \vee \bar{A} B \bar{C}$$

- Entwerfen Sie Schaltungen, die folgende Funktionen realisieren:

a)  $A \oplus (B \vee C)$

b)  $(A \oplus B) \cdot C$

- mit NAND-Gattern,
- mit NOR-Gattern.

(Jeweils einschließlich der ggf. erforderlichen Negationen.)

- Viele moderne Interfaces sind Punkt-zu-Punkt-Verbindungen, die an Schaltverteiler angeschlossen sind. Das Grundelement eines typischen Schaltverteilers (Switch Fabric) ist das Schalterelement von Abb. 1. Ist das Steuersignal  $C = 0$ , so sind die Verbindungen von A nach X und von B nach Y durchgeschaltet. Ist  $C = 1$ , sind A mit Y und B mit X verbunden. Entwerfen Sie ein solches Schalterelement mit NAND-Gattern, die zwei Eingänge haben. (Jeweils einschließlich der ggf. erforderlichen Negationen.)

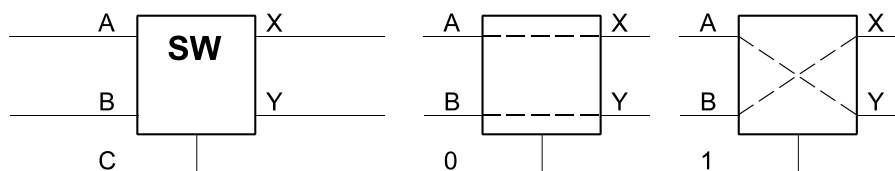


Abb. 1

- Ist folgende Gleichung richtig? – Beweisen Sie bitte Ihre Aussage (das Beweisverfahren ist freigestellt).

$$A \oplus (B \vee C) = (A \oplus B) \vee (A \oplus C)$$

- Entwerfen Sie eine Schaltung, die die folgende Schaltgleichung verwirklicht. Es sind NAND- und XOR-Gatter mit zwei Eingängen zu verwenden.

$$A \cdot (B \vee \overline{C \oplus D})$$

## Kombinatorische Schaltungen entwerfen (2), Schaltungsvereinfachung, KV-Diagramm

Was sitzen muß: die DeMorganschen Regeln und das KV-Diagramm. Das muß aber nicht weiter getrieben werden als unbedingt nötig. Tricks können entfallen – wir leben nicht mehr in den 60er und 70er Jahren... Im Internet stehen ergänzende Texte zur Schaltalgebra im allgemeinen und zum Minimieren mit dem KV-Diagramm im besonderen. Die Minimierung wird dort am Beispiel des Siebensegmentdecoders in allen Einzelheiten gezeigt.

- Ein Mikrocontroller hat zwei Zugriffssteuersignale, die aktiv Low wirken: PSEN# für den Befehlslesezugriff und RD# für den Datenlesezugriff. Geben Sie eine Schaltung an, die ein vereinheitlichtes Zugriffssignal RD\_U# liefert, das in beiden Zugriffsfällen aktiv wird. Und zwar möglichst einfach...
- Mit welchem der dargestellten Gatter ist die Verknüpfung von Abb. 1 auf einfachste Weise zu realisieren?

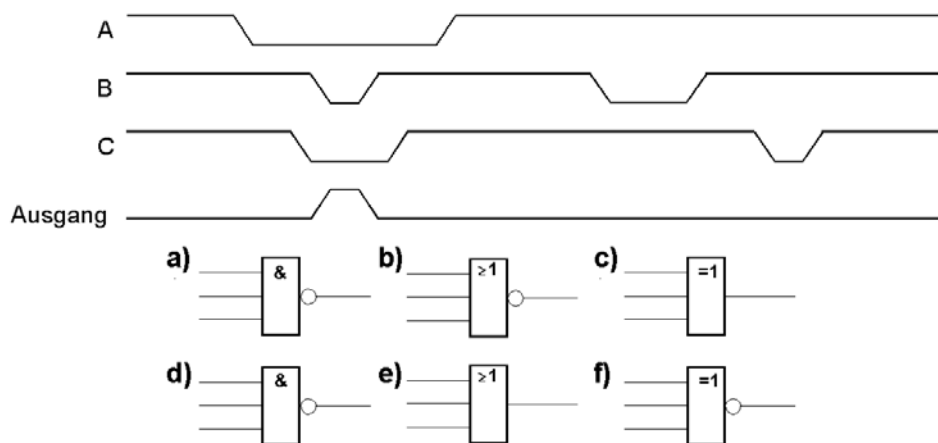


Abb. 1

- Minimieren Sie folgende Schaltfunktion (Abb. 2) mittels KV-Diagramm (Karnaugh-Plan).

Weitere Übungsbeispiele:

a)  $\bar{a} \cdot \bar{b} \cdot \bar{d} \vee \bar{a} \cdot \bar{b} \cdot c \cdot d \vee a \cdot b \cdot \bar{c} \cdot \bar{d} \vee a \cdot \bar{b} \cdot c \cdot \bar{d}$

Die Belegung  $a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d}$  kommt nie vor.

b)  $\bar{a} \cdot \bar{b} \cdot \bar{c} \vee \bar{a} \cdot \bar{b} \cdot c \cdot \bar{d} \vee \bar{a} \cdot b \cdot \bar{c} \cdot \bar{d} \vee a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} \vee a \cdot \bar{b} \cdot c \cdot \bar{d} \vee a \cdot b \cdot c \cdot d$

Die Belegung  $\bar{a} \cdot b \cdot \bar{c} \cdot \bar{d}$  kommt nie vor.

Schaltgleichung:  
 $\bar{a} \bar{b} \bar{c} \bar{d} \vee \bar{a} \bar{b} \bar{c} d \vee a \bar{b} \bar{c} \bar{d} \vee$   
 $\bar{a} b c d \vee \bar{a} \bar{b} c d \vee \bar{a} b c \bar{d}$

Die Belegung  $\bar{a} \bar{b} c \bar{d}$  kommt nie vor.

		CD:					
		0 0	0 1	1 1	1 0		
AB:						AB:	
00		0 = $\bar{a} \bar{b} \bar{c} \bar{d}$	1 = $\bar{a} \bar{b} \bar{c} d$	3 = $\bar{a} \bar{b} c d$	2 = $\bar{a} \bar{b} c \bar{d}$	00	
01		4 = $\bar{a} b \bar{c} \bar{d}$	5 = $\bar{a} b \bar{c} d$	7 = $\bar{a} b c d$	6 = $\bar{a} b c \bar{d}$	01	
11		12 = $a b \bar{c} \bar{d}$	13 = $a b \bar{c} d$	15 = $a b c d$	14 = $a b c \bar{d}$	11	
10		8 = $a \bar{b} \bar{c} \bar{d}$	9 = $a \bar{b} \bar{c} d$	11 = $a \bar{b} c d$	10 = $a \bar{b} c \bar{d}$	10	
		CD:					
		0 0	0 1	1 1	1 0		

Abb. 2

## Kombinatorische Schaltungen entwerfen (3), elementare Operationen und Verarbeitungsschaltungen

Der elementare Schaltungsentwurf schließt ab mit dem Einsatz von Multiplexern als universelle Logikschaltungen. Wir beschränken uns auf den einfachsten Fall: Variable an den Adreßeingängen, Einsen und Nullen an den Dateneingängen (1:1-Umsetzung der Wahrheitstabelle). Dann geht es darum, wie man mit bitweisen Booleschen Verknüpfungen Bits und Bitfelder setzen, löschen, vergleichen und transportieren kann (in der Programmiersprache C geht es ebenso zu...). Schließlich sollte man die typischen Besonderheiten der vorzeichenlosen und ganzen Binärzahlen kennen (letztere in Zweierkomplementdarstellung). Spitzfindigkeiten des Schaltungsentwurfs (Übertragungsvorausschau usw.) werden nicht behandelt.

1. Entwerfen Sie eine Auswahl-schaltung, die eines von fünf Eingangssignalen E1...E5 auswählen kann.
  - a) mit beliebigen Gattern,
  - b) mit 2-zu-1-Multiplexern.
2. Realisieren Sie eine XOR-Verknüpfung von drei Variablen  $A \oplus B \oplus C$  mit einem Multiplexer.
3. Realisieren Sie die Verknüpfung  $(A \oplus B) \cdot C$  mit einem Multiplexer.
4. Entwerfen Sie eine Mehrheitserkennungsschaltung (Voter), die drei Signale A, B, C bewertet und jene Belegung zum Ausgang weitergibt, die der Mehrheit der Eingangssignale gemeinsam ist. Beispiele: wenn A, B, C alle = 1, dann wird eine Eins ausgegeben, wenn A = 0, B = 1, C = 0, wird eine Null ausgegeben. Hierzu soll ein Multiplexer 1-aus-8 eingesetzt werden (Abb. 1).

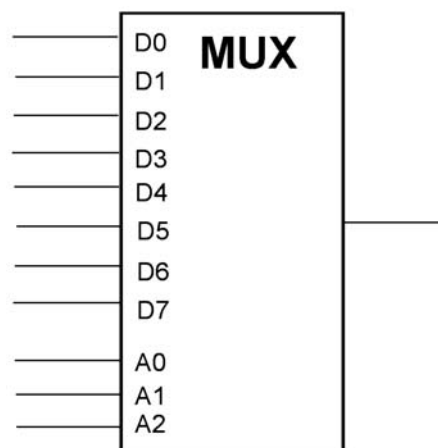


Abb. 1

5. Einige Beispiele zum Setzen, Löschen und Ändern von Bits (Abb. 2).

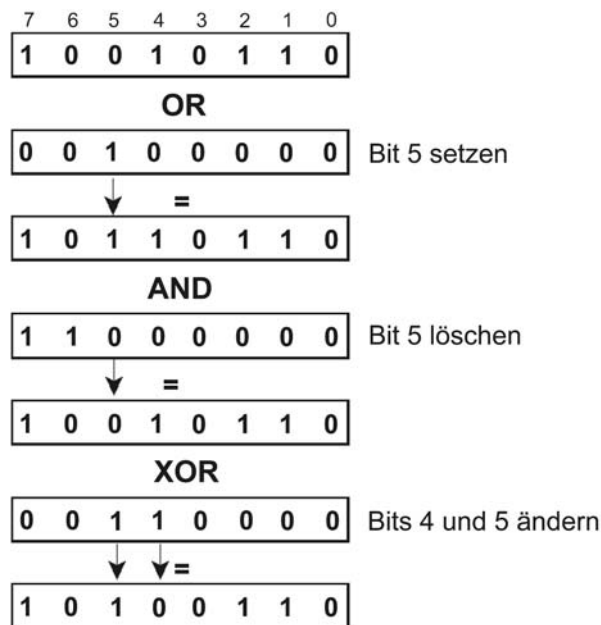


Abb. 2

6. Geben Sie eine Vergleichsschaltung an, die zwei Bytes A7...A0 und B7...B0 miteinander vergleicht und die Aussage Gleich/Ungleich liefert (Ausgangssignal EQUAL).
7. Welche Wertebereiche kann man mit 12 Bits darstellen?
  - a) als natürliche (vorzeichenlose) Binärzahl,
  - b) als ganze Binärzahl.
8. Nennen Sie die größte negative Zahl, die sich mit 23 Bits darstellen lässt.
9. Wie verhält sich der Ausgangsübertrag eines Zweierkomplement-Rechenwerks, wenn A–B gerechnet wird und gilt:
  - a)  $A > B$ ,
  - b)  $A = B$ ,
  - c)  $A < B$ .

(Zu betrachten für vorzeichenlose und ganze Zahlen.)
10. Erklären Sie kurz die Begriffe Wrap-Around-Arithmetik und Sättigungsarithmetik. Berechnen Sie die Ergebnisse der folgenden Rechenoperationen gemäß der Sättigungsarithmetik. Die Werte sind vorzeichenlose 16-Bit-Binärzahlen. Ergebnisse hexadezimal angeben.
  - a)  $1234H + 18H$ .
  - b)  $1234H - A000H$ .
11. Erläutern Sie kurz den Fachbegriff Overflow-Bedingung. Was zeigt diese Bedingung an? Wie wird sie gebildet?
12. Wir haben ein 8-Bit-Addierwerk. Wie kann man damit zwei Binärzahlen addieren, die 24 Bits lang sind?