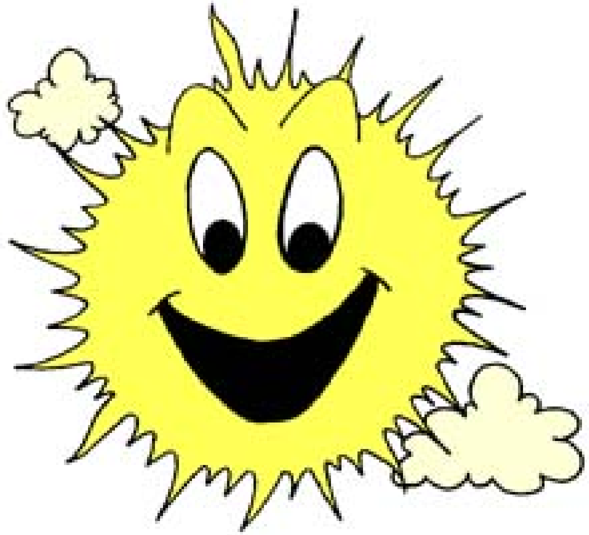
Wiederholung u.a. der 13. Vorlesung



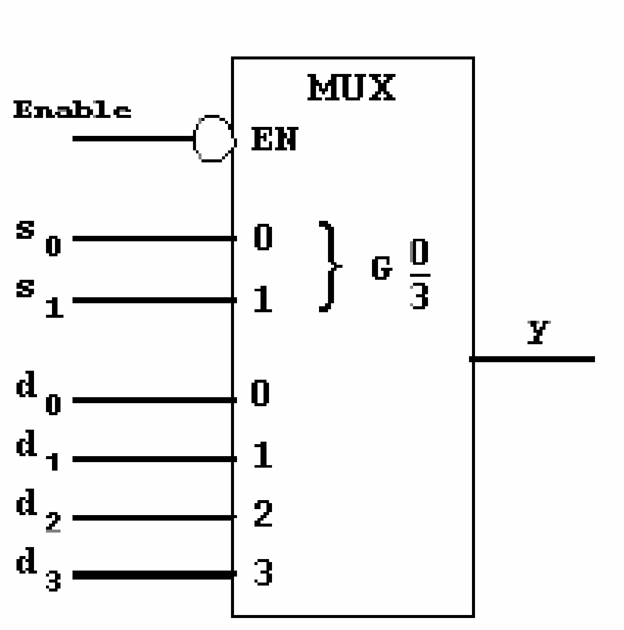
Regeln für Klausur:

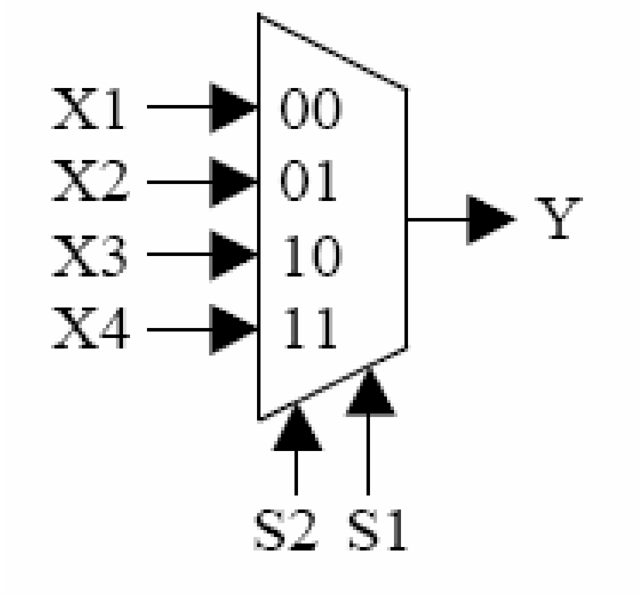
Personalausweis (amtl. Ausweis mit Lichtbild)

Zugelassen: 1 handgeschriebenes Blatt

Nicht zugelassen: Handy, Taschenrechner, sonstige Aufzeichnungen und Hilfsmittel

8.1 Multiplexer DIN / ISO -Symbol





DIN 40900-Symbol für einen 4:1 Multiplexer

s0 und s1: Steuereingänge

d0-d3: Dateneingänge

G: Steuereingänge steuern Daten-Eingänge durch UND-Verknüpfung

ISO-Symbol für einen 4:1 Multiplexer

S1 und S2: Steuereingänge

X1-X4: Dateneingänge

Zahlen geben Indexbereich an

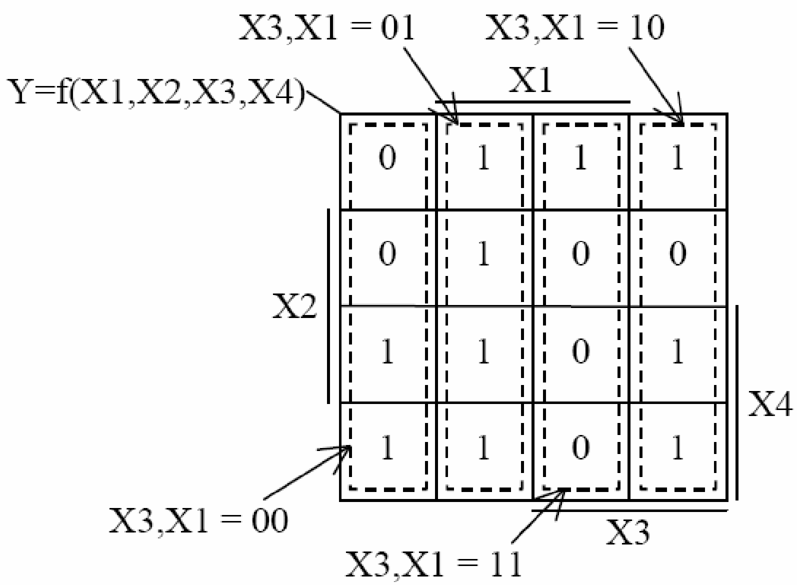
8.2 Realisierung von Funktionen mit MUX

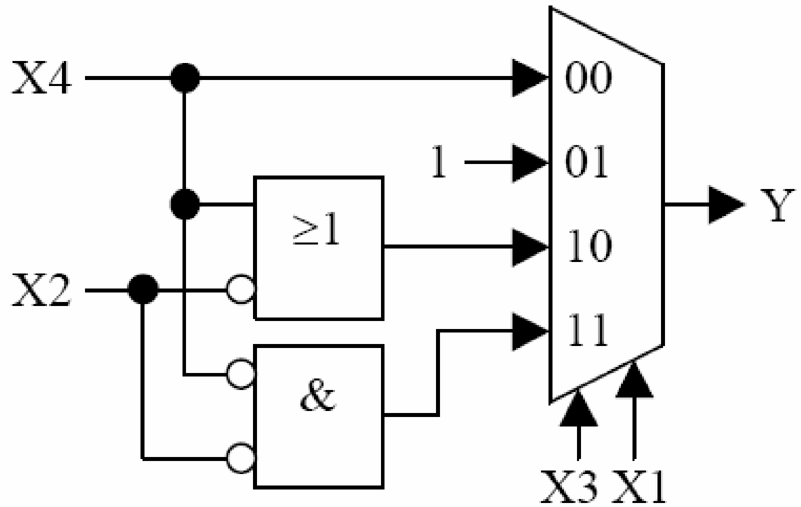
Multiplexer sind gut für die Realisierung boolescher Funktionen geeignet:

Bei N Eingangsvariablen am einfachsten mit 2^N:1 Multiplexern:

- Die Eingangsvariablen schaltet man an die Selektionseingänge
- An die Dateneingänge schaltet man, je nach Erfordernis der zu realisierenden Funktion eine 0 oder 1.

8.2 Beispiel 3: Funktion N=4 mit 4:1 MUX (2)



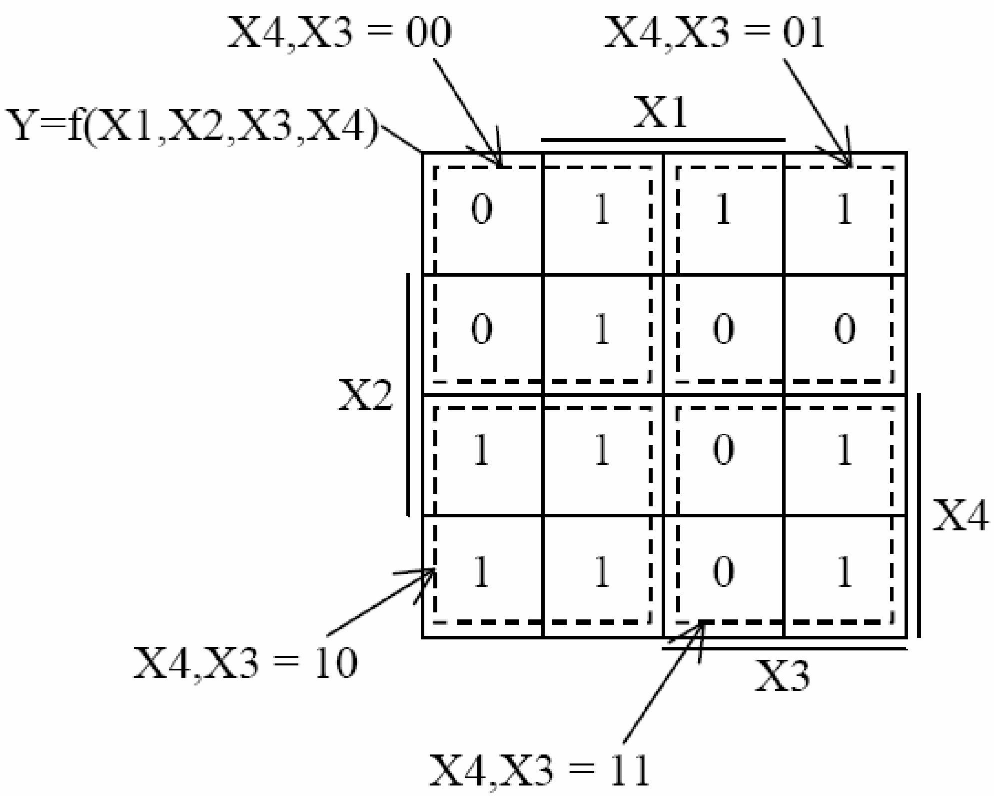


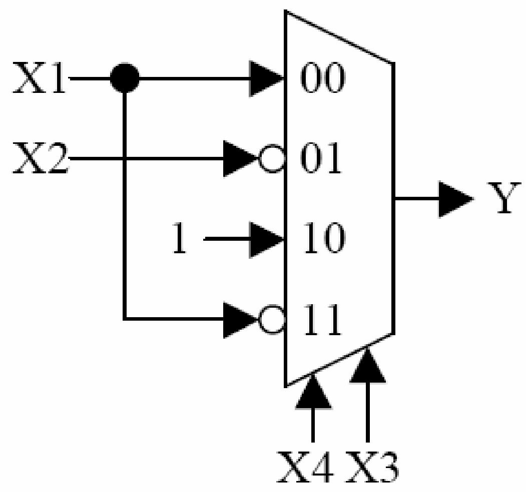
Steuerkombination X3,X1	Ausgangsfunktion
00	$Y = X4$
01	$Y = 1$
10	$Y = X4 \vee (\neg X2)$
11	$Y = (\neg X4) \wedge (\neg X2)$

Die gefundene Lösung ist noch nicht optimal.

→ Suche andere Kombination der Ansteuersignale ...

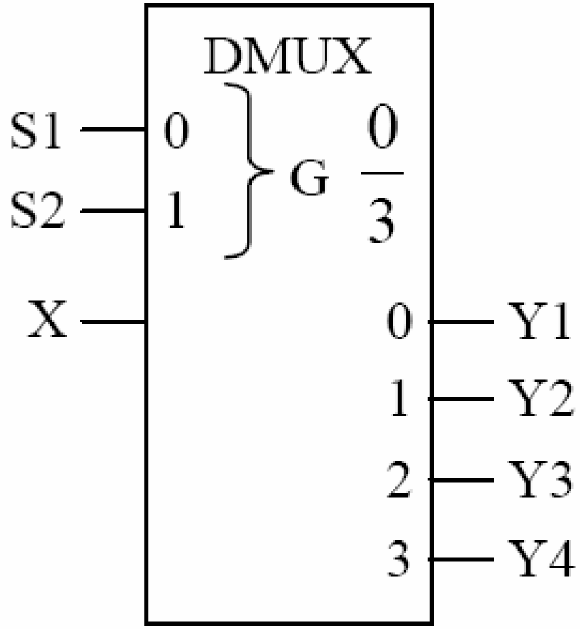
8.2 Beispiel 3: Funktion N=4 mit 4:1 MUX (3)

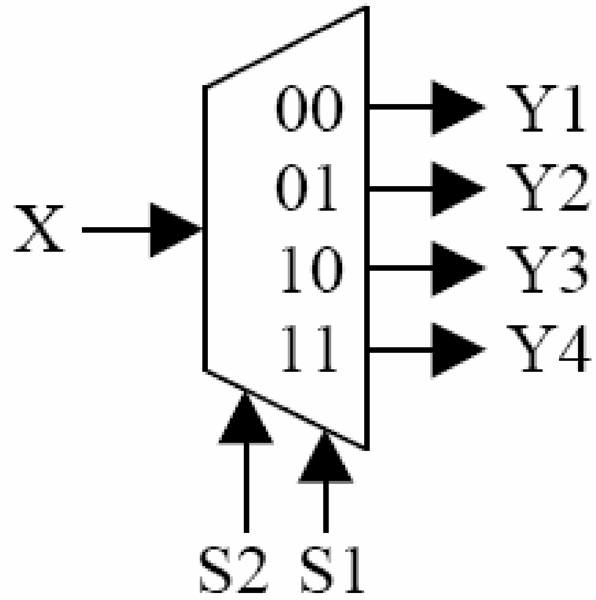




Aus dem KV-Diagramm und andere Wahl der Unterdiagramme ergibt sich einfache Lösung

8.3 Demultiplexer DIN / ISO -Symbol





DIN 40900-Symbol für einen 1:4 Demultiplexer

s0 und s1: Steuereingänge

X: Dateneingang

Y1-Y4: Datenausgang

G: Steuereingänge steuern Daten-Ausgänge durch UND-Verknüpfung

ISO-Symbol für einen 1:4 Demultiplexer

S1 und S2: Steuereingänge

Y1-Y4: Datenausgänge

Zahlen geben Indexbereich an

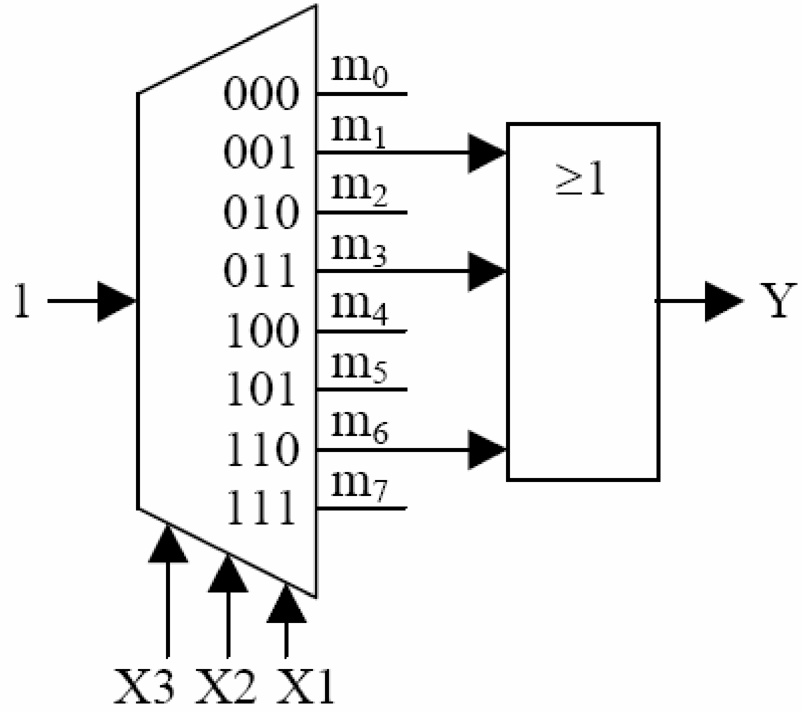
8.4 Realisierung von Funktionen mit DMUX

Legt man bei einem Demultiplexer den Eingang X auf 1, liefert jeder Ausgang eine Minterm-Funktion m_i der Steuereingänge. Die gewünschte Funktion erhält man durch die disjunktive Verknüpfung der gewünschten Minterme:

Mit einem **1:2^N Demultiplexer** und einem **N Bit** breiten **ODER-Gatter** lässt sich **jede Funktion** aus N Eingangsvariablen realisieren.

8.4 Beispiel: Funktionen N=3 und 1:8 DMUX

X3	X2	X1	Y	
0	0	0	0	
0	0	1	1	m_1
0	1	0	0	
0	1	1	1	m_3
1	0	0	0	
1	0	1	0	
1	1	0	1	m_6
1	1	1	0	



Demultiplexer liefert direkt alle Minterme m_i aus den N Eingangsvariablen.

Man verbindet die benötigten Minterme aus dem $1{:}2^N$ DMUX mit den Eingängen eines nachfolgenden ODER-Gatters (evtl. unbenutzte Eingänge auf 0 legen).

Damit ist die vorgegebene Funktion realisiert.

X. Wiederholung

- Wertetabelle
  - Don‘t care
  - Nummerieren ($m_0 ... m_n$ bzw. M_0 ... M_n)
  - DNF: $Y = m_i \vee ... \vee m_k$
  - KNF: $Y = M_j \wedge ... \wedge M_l$
  - KV-Diagramm: Reihenfolge der Variablen
- Digitalschaltungen
  - Symbol für invertierte Ein- und Ausgänge
  - Morgan‘sche Gesetz: Gatter
- Zahlendarstellung:
  - 0, +0, -0 ...
  - Ganzzahl, Festkomma, Gleit- oder Fließkomma

X. Wiederholung (2)

- Multiplexer
- Demultiplexer
- Kombinatorische Grundschaltungen
  - Rechenwerke (s. a. Rechnergrundlagen)

X. Wiederholung (4)

x 4	x 3	x 2	x 1	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	*
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	*
1	0	0	0	0
1	0	0	1	*
1	0	1	0	1
1	0	1	1	*
1	1	0	0	*
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

		x1	x1		
	1	1	1	1	
x2	1	*	*	0	
x2	1	*	0	0	x4
	0	*	0	*	x4
			x3	x3	

Don‘t Care Terme können belegt werden, wie man sie braucht, d.h. mit 1 oder 0

X. Wiederholung (5)

Definitionen:

- **Implikant**: Verbund aus Min- oder Maxterm führt zu Term geringerer Komplexität, d.h. reduzierte Anzahl Eingangsvariablen. Auch einzelne Min- oder Maxterme.
- **Primimplikant**: Implikant, der in keinem anderen Implikanten vollständig enthalten ist.
- **Kernprimimplikant**: Primimplikant, der mindestens einen Min- oder Maxterm enthält, der in keinem anderen Primimplikanten enthalten ist.

		X1	X1	
	0	1	1	0
X2	1	1	1	0
			X3	X3

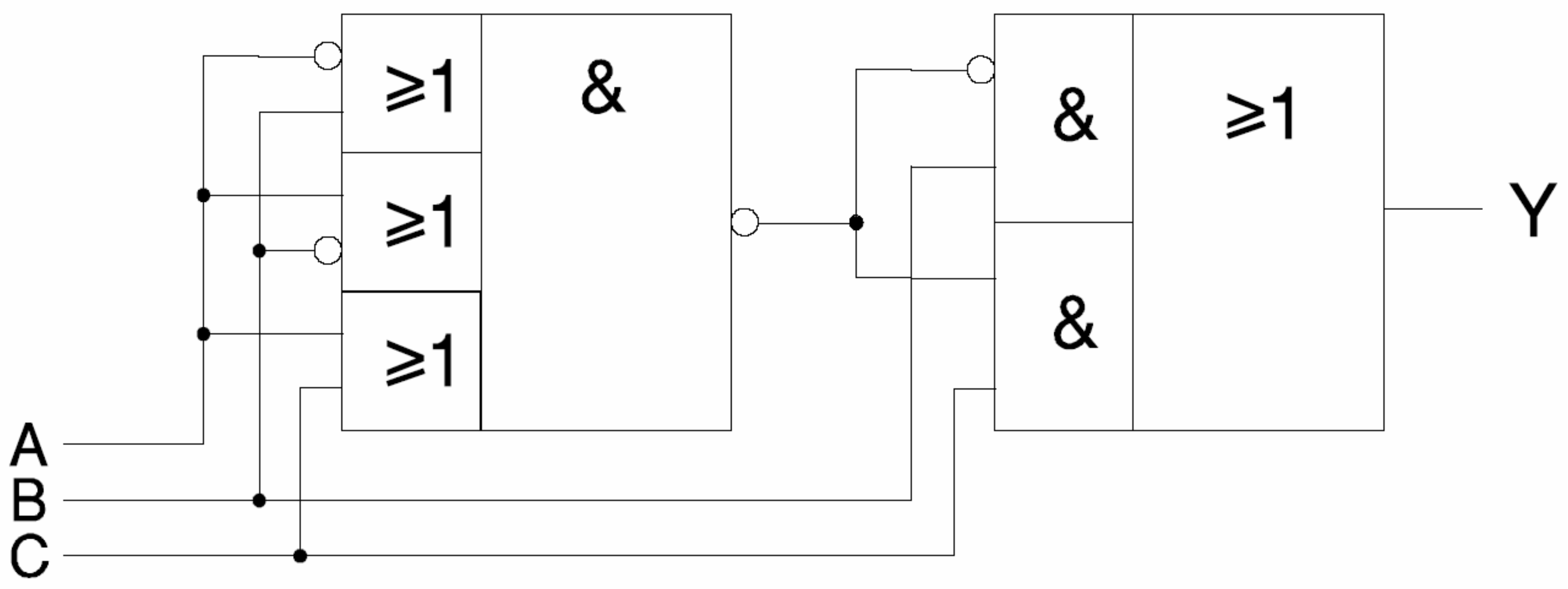
$I1 = (\neg X3) \wedge X2$

$I2 = (\neg X3) \wedge X1$

$I3 = X1$

X. Wiederholung (6) Analyse/Vereinfachung

Informatik



Hörsaalübung:

- Wertetabelle
- KV-Diagramm
- DMF und/oder KMF
- (Vereinfachte Schaltung zeichnen)

X. Wiederholung (7)

C	B	A	
x3	x2	x1	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

		x1	x1	
	0	0	1	0
x2	0	1	1	1
			x3	x3

DMF=(A∧B) ∨ (A ∧C) ∨ (B ∧C)

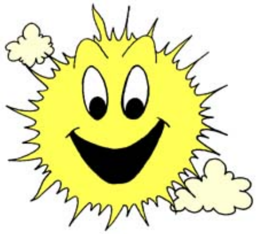
		x1	x1	
	0	0	1	0
x2	0	1	1	1
			x3	x3

KMF = (B∨C) ∧ (A∨C) ∧ (A∨B)

Weitere Hörsaalübung:

- Realisieren Sie die Schaltung mit einem Multiplexer
- Realisieren Sie die Schaltung mit einem Demultiplexer und einem Oder-Gatter

Ende der Wiederholung



→ → → → → → → → → → → → → → → → →

8.5 Code-Umsetzer

- Prioritätsencoder
- BCD zu 7-Segment-Dekoder

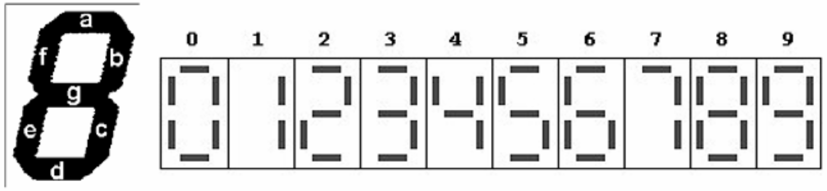
Prioritätsencoder werden in Rechnern benötigt, um Ereignisse, die auch gleichzeit auftreten können, auf einzelnen Leitungen zu signalisieren.

Der Rechner kann diese Ereignisse jedoch nur sequenziell abarbeiten.

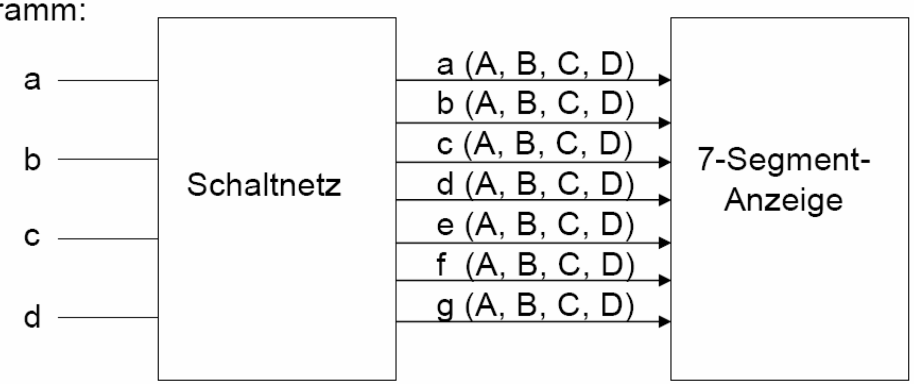
Bei mehreren aktivierten Leitungen muss das Ereignis mit der höchsten Priorität ermittelt werden.

→ Hardware: Prioritätsencoder (s. Script / Literatur)

8.5.2 BCD zu 7-Segment-Decoder



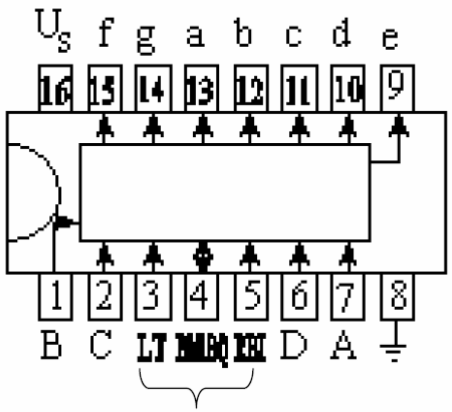
Blockdiagramm:



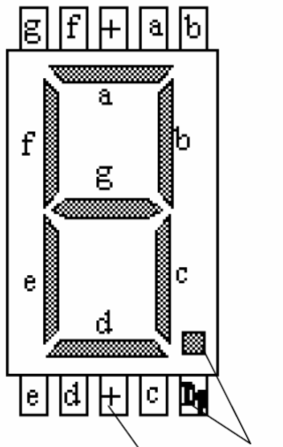
Für diese recht komplexe Aufgaben gibt es fertige Bausteine, z.B. TTL 7447

Bauelemente:

Dekoder: TTL-IC 7447



Z.B. „Lampentest“



Dezimalpunkt

Anschlusspunkt(e); nur einmal zu beschalten