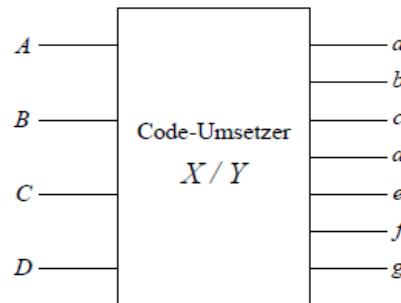


- Das zugrundeliegende Schaltnetz: 8421-BCD-Code in 7-Segment Code Umsetzer

Als Beispiel für einen Code-Umsetzer wird im Kurstext 1608 der 8421-BCD-Code in 7-Segment Code Umsetzer wie folgt eingeführt:

*In digitalen Datenverarbeitungssystemen werden Dezimalziffern sehr oft in einem BCD-Code dargestellt und in 7-Segmenteinheiten zur Anzeige gebracht. Dafür ist ein Code-Umsetzer erforderlich. In dem hier betrachteten Beispiel mögen die Dezimalziffern im 8421-BCD-Code vorliegen. Die Code-Umsetzung kann durch das Blockschaltbild aus Abb. 2.16 dargestellt werden:*



Die Zuordnung (Codierung) der Dezimalziffern vom 8421-BCD-Code zu den Segmenten der 7-Segment-Anzeige wird nach Abb. 2.17 in einer Wertetabelle festgelegt:

Dezimal Ziffer	8421-BCD-Code				7-Segment-Code						
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1

Nach der Vereinfachung ergeben sich für die Ausgangsvariablen die Funktionsgleichungen in der DNF:

$$\begin{aligned}
a &= D \vee (\overline{A} \wedge \overline{C}) \vee (A \wedge C) \vee (A \wedge B) \\
b &= \overline{C} \vee (A \wedge B) \vee (\overline{A} \wedge \overline{B}) \\
c &= A \vee \overline{B} \vee C \\
d &= (\overline{A} \wedge B) \vee (\overline{A} \wedge \overline{C}) \vee (B \wedge \overline{C}) \vee (A \wedge \overline{B} \wedge C) \\
e &= (\overline{A} \wedge B) \vee (\overline{A} \wedge \overline{C}) \\
f &= D \vee (\overline{A} \wedge \overline{B}) \vee (\overline{A} \wedge C) \vee (\overline{B} \wedge C) \\
g &= (\overline{A} \wedge B) \vee (\overline{B} \wedge C) \vee (B \wedge \overline{C}) \vee D
\end{aligned}$$

- **Die Hades Simulation:**

Der als letzte Seite angehängte Screenshot der Simulation zeigt links die 4 Eingangssignale  $A, B, C$  und  $D$ , wobei  $D$  das höchstwertigste und  $A$  das niederwertigste Bit darstellt, also  $8_{10} = 1000_2 = DCBA$ .

Die Eingangssignale wurden initial mit 0 belegt, entsprechend zeigt auch die 7-Segment-Anzeige 0 an. Da diese Anzeige nur eine Ziffer darstellen kann, wird offensichtlich die Darstellung der Dezimalzahlen 10-15 nicht gelingen und unsinnige Ausgaben produzieren.

Die Funktionsgleichungen wurden 1:1 umgesetzt, also entsprechen die Ausgänge der OR-Schaltglieder genau den Gleichungen.

Beispiel:

Das obere OR4, dessen Ausgangssignal mit  $a$  beschriftet ist, bekommt die Eingangssignale:

Eingang A:  $A \wedge B$  (1. AND von links),  
Eingang B:  $A \wedge C$  (2. AND von links),  
Eingang C:  $/A \wedge /C$  (3. AND von links),  
Eingang D:  $D$ ,

dieses OR-Schaltglied realisiert also die Funktionsgleichung zu  $a$ .

Wie auch bei den anderen Simulationen gilt: Durch Klick auf die Eingangssignale belegt man die Eingangsvariablen mit 0 oder 1 und kann dann das Verhalten der Schaltglieder im Schaltnetz verfolgen. Rot steht für 1, grau für 0.

- **Die Simulation besteht aus folgenden Komponenten:**

- 4 Ipins(switch)
- 1 OR2
- 4 OR4
- 4 INV(small)
- 9 AND2
- 2 OR3
- 1 7-Segment-Anzeige
- 1 GND-connection

