

## Beispiel Code-Wandler: BCD-zu-7-Segmente-Wandler

Aufgabe:

Es ist die vollständige, minimierte Schaltung eines BCD –zu-7-Segmente-Wandlers (vgl. Skript Schaltalgebra) zu realisieren. Anzuzeigen sind nur die Ziffern "0" bis "9" und "Blank" (= nichts leuchtet).

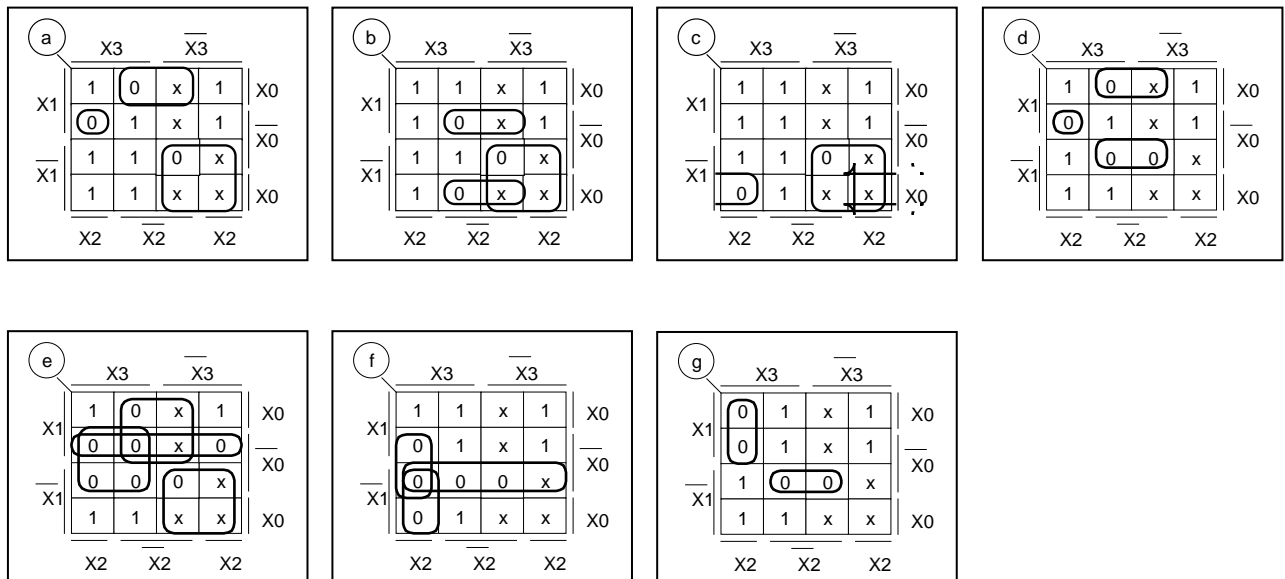
Es stehen NOT-Gatter (6 Gatter pro integriertem Schaltungs-Baustein [= Chip]), AND-, NAND-OR- und NOR-Gatter (vier 2-Input-Gatter pro Chip, bzw. drei 3-Input-Gatter pro Chip) zur Verfügung. Es sind möglichst wenige Chips einzusetzen.

Weitere Vorgaben (Vereinbarungen):

- 4 Eingänge am Wandler (als 4-stelliges Wort aufzufassen): 0000 stellt die Ziffer "0" dar, 0001 die Ziffer "1" etc. bis 1001 für die Ziffer "9".
- "Blank" werde mit allen Wandler-Eingängen im Zustand hoch (logisch 1) realisiert.
- Ein "hoher" Eingang (logisch 1) an der Anzeige bringe das entsprechende Segment zum Leuchten, Eingang "tief" (logisch 0) an der Anzeige bewirke ein nicht-leuchtendes Segment.
- Alle nicht erwähnten weiteren Eingangsmuster sollen für die Minimierung der Schaltung vorteilhaft genutzt werden.

Lösung:

Weniger 0 bei den Ausgängen, daher Wahl KKNF.

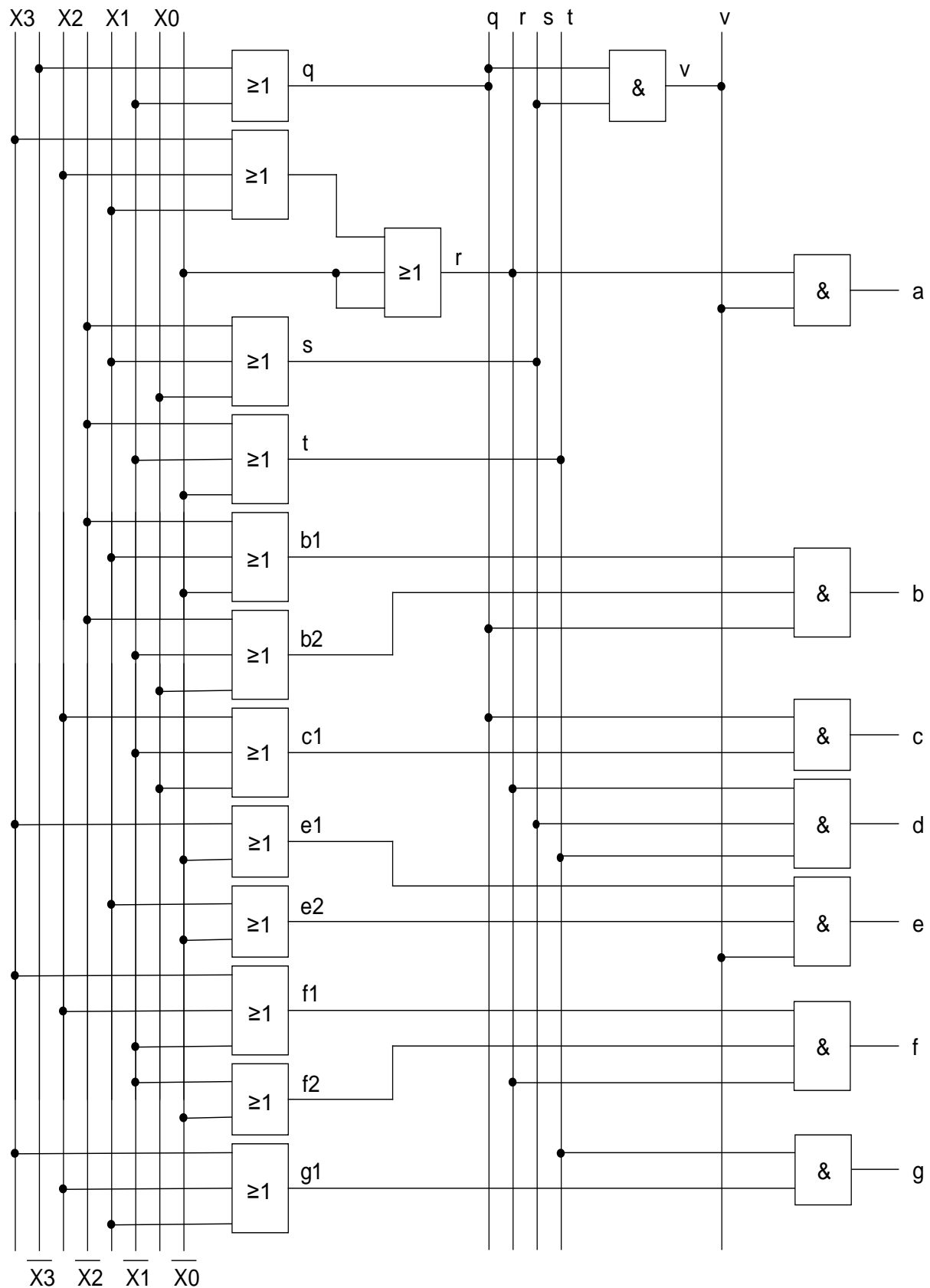


$$\begin{aligned}
 q &= (\overline{X3} + \overline{X1}) & r &= (X3 + X2 + X1 + \overline{X0}) & s &= (\overline{X2} + X1 + X0) & t &= (\overline{X2} + \overline{X1} + \overline{X0}) \\
 a &= qrs & b &= (\overline{X2} + X1 + \overline{X0})(\overline{X2} + \overline{X1} + X0)q & c &= (X2 + \overline{X1} + X0)q & d &= rst \\
 e &= (\overline{X2} + X1)(X3 + \overline{X0})(X1 + \overline{X0})q & f &= (X3 + X2 + \overline{X0})(X3 + X2 + \overline{X1})(\overline{X1} + \overline{X0}) & g &= (X3 + X2 + X1)t
 \end{aligned}$$

Anmerkungen:

- Wo ersichtlich, wurden gemeinsame Blöcke bereits mit q, r, s und t gekennzeichnet.
- Da der Maxterm r sowieso bereit gestellt werden muss, bringt die Vereinfachung  $(X3 + X2 + \overline{X0})$  in f nichts und wir können schreiben:  $f = r(X3 + X2 + \overline{X1})(\overline{X1} + \overline{X0})$ .
- Ähnliches gilt für den Term  $(\overline{X2} + X1)$  in e: da s sowieso bereitgestellt werden muss, kann e wie folgt formuliert werden:

$$\begin{aligned}
 e &= s(X3 + \overline{X0})(X1 + \overline{X0})q \\
 q &= (\overline{X3} + \overline{X1}) & r &= (X3 + X2 + X1 + \overline{X0}) & s &= (\overline{X2} + X1 + X0) & t &= (\overline{X2} + \overline{X1} + \overline{X0}) \\
 v &= qs \\
 a &= rv & b &= (\overline{X2} + X1 + \overline{X0})(\overline{X2} + \overline{X1} + X0)q & c &= (X2 + \overline{X1} + X0)q & d &= rst \\
 e &= (X3 + \overline{X0})(X1 + \overline{X0})v & f &= (X3 + X2 + \overline{X1})(\overline{X1} + \overline{X0})r & g &= (X3 + X2 + X1)t
 \end{aligned}$$

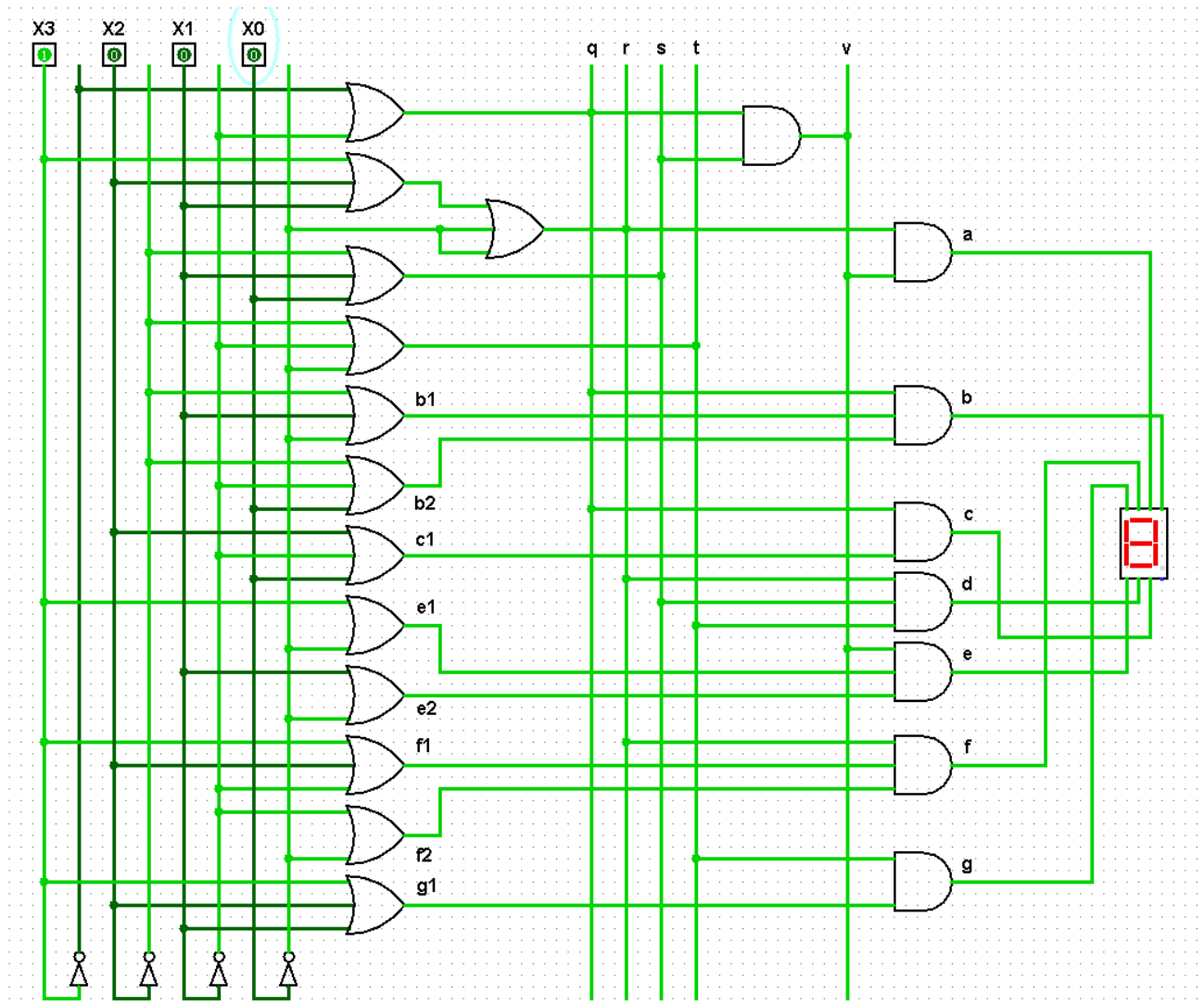


Realisierte Schaltung

Verwendete Bausteine (total 8):

- 1 NOT-IC (4 Not-Tore verwendet von 6), nicht gezeichnet
- 1 OR-IC 2-Input (4 Tore von 4)
- 3 OR-IC 3-Input (9 von 9)
- 1 AND-IC 2-Input (4 von 4)
- 2 AND-IC 3-Input (4 von 6)

Umgesetzt in Logisim:



Vgl. auch Datei *bcdTo7segments.circ*.