

## 13. Übung Funktionsblöcke

**Inhalt:** Multiplexer, Demultiplexer, Dekodierer, Kodewandler, Busse

### 1. Aufgabe

Eine Boole'sche Funktion sei durch folgendes Schaltnetz vorgegeben:

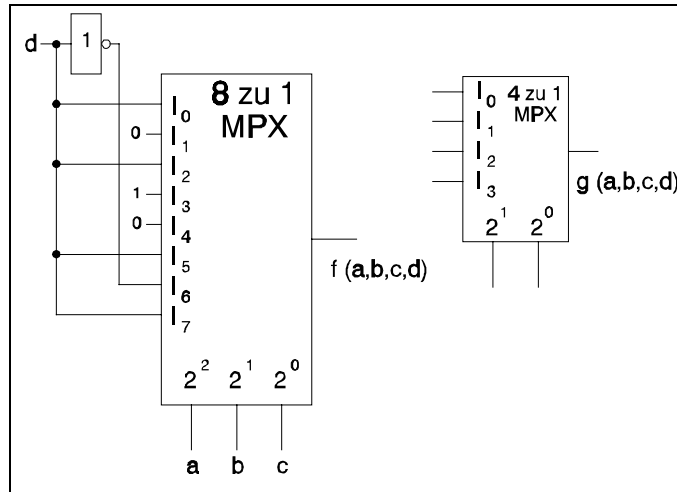


Bild 13.1

Die Boole'sche Funktion  $g(a,b,c,d) = f(a,b,c,d) \vee ab\bar{c}d$  soll unter alleiniger Verwendung eines 4 zu 1 Multiplexers realisiert werden. Geben Sie die MPX-Beschaltung für  $g$  an.

### 2. Aufgabe

Entwerfen Sie folgende Multiplexerschaltungen

- a) 4 Eingänge auf 1 Ausgang
- b) 4\*4 Eingänge auf 4 Ausgänge
- c) 2\*8 Eingänge auf 8 Ausgänge

Entwerfen Sie folgende Demultiplexerschaltungen

- d) 1 Eingang auf 4 Ausgänge
- e) 4 Eingänge auf 4\*4 Ausgänge
- f) 8 Eingänge auf 2\*8 Ausgänge

### 3. Aufgabe

- a) Entwerfen Sie eine Schaltung, die von zwei Eingangsbit die gerade und ungerade Parität bildet.
- b) Erweitern Sie den Entwurf, so daß eine beliebige Kaskadierbarkeit entsteht und bilden Sie die Parität eines Bytewertes.

### 4. Aufgabe

Entwerfen Sie einen Modul, der in Abhängigkeit von einer Steuerspannung das Paritätsbit eines 4-Bit-Wertes erzeugt (Paritätssgenerator) oder die Parität eines 5-Bit-Wertes prüft (Paritätsprüfer) und erkannte Fehler signalisiert.

## 5. Aufgabe

Entwerfen Sie mit Hilfe der Wahrheitstabelle einen Gray-Kode-nach-BCD-Konverter.

Eingänge				Ausgänge			
A	B	C	D	F1	F2	F3	F4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1

- Verwenden Sie zwei 4-zu-1-Multiplexer und Ergänzungslogik.
- Verwenden Sie vier 8-zu-1-Multiplexer.
- Verwenden Sie vier 2-zu-1-Multiplexer und Ergänzungslogik.
- Welche Realisierung erfordert die wenigsten Schaltkreise?

## 6. Aufgabe

Realisieren Sie die folgenden Boole'schen Funktionen unter Verwendung eines BCD-nach-Dezimal-Dekoders mit low-aktiven Ausgängen und einer Ergänzungslogik (AND- und OR-Gatter).

- $f(a,b,c) = \overline{abc} \vee \overline{ab}c \vee \overline{bc}a$
- $g(a,b,c) = \overline{ab} \vee \overline{ab}$
- $h(a,b,c,d) = \overline{abc} \vee \overline{abcd} \vee \overline{abcd} \vee \overline{abcd}$

## 7. Aufgabe

- Entwerfen Sie einen Prioritätsdekoder, der an den Ausgängen den Binärcode des höchstpriorisierten aktiven Einganges ausgibt. Die 8 Eingänge und der Enable-Eingang sind low-aktiv, die Ausgänge sind high-aktiv. Es existiert ein spezieller (low-aktiver) Ausgang, der anzeigt, daß alle Eingänge inaktiv sind.
- Zeigen Sie, wie man zwei solche Prioritätsdekoder für 16 Eingänge zusammenschalten kann.
- Zeigen Sie, wie man vier solche Prioritätsdekoder für 32 Eingänge zusammenschalten kann.

