

## Aufgabenblatt 11

letzte Aktualisierung: 18. February, 12:30

Ausgabe: 01.02.2002

Abgabe: 11./12.02.2002 Prozent: 100

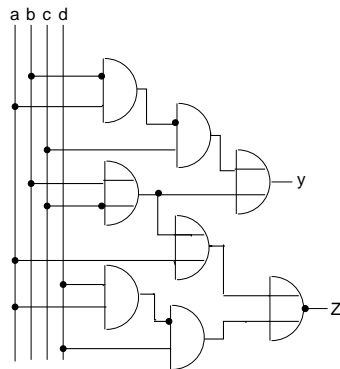
**Thema:** KV-Minimierung, Schaltnetze

Der letzte Test findet in den Tutorien am 11./12.02.2002 statt.

### 1. Aufgabe (20 Prozent): Schaltnetzanalyse

In dieser Aufgabe sollt ihr ein grundsätzliches Verständnis für Schaltnetze und deren zugehörigen Funktionen erhalten.

**1.1. Gleichung aus Schaltung herleiten (Tut)** Gegeben ist das untenstehende Schaltnetz. Bestimmt die Funktion, die das Ausgangssignal für y bestimmt!



**Lösung:**

$$y = \overline{b} \cdot a \cdot c + b + \overline{c}$$

**1.2. Gleichung aus Schaltung herleiten (10 Prozent)** Bestimmt die Funktion für den Ausgang z!

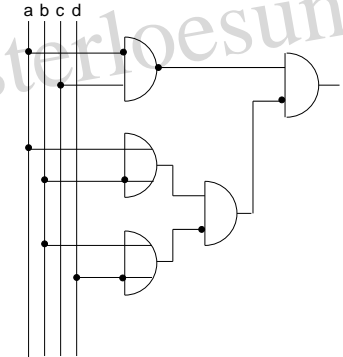
**Lösung:**

$$z = b + \overline{c} + a + \overline{a} \cdot \overline{d} \cdot d$$

**1.3. Schaltung aus Gleichung entwickeln (10 Prozent)** Zeichnet für folgende Gleichung ein Schaltnetz!

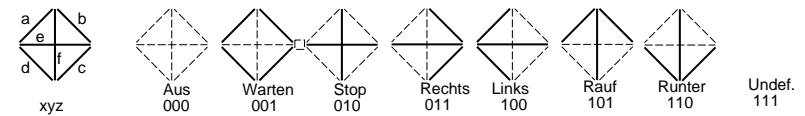
$$y = \overline{(a \cdot c)} \cdot (a + \overline{b}) \cdot \overline{(b + d)}$$

**Lösung:**



### 2. Aufgabe (50 Prozent): Richtungsanzeiger

Es soll ein Dekoder für eine Richtungsanzeige entworfen werden. Jeder der 6 Ausgänge treibt dabei ein Leuchtsegment, das entsprechend der Bitkombination am Eingang des Dekoders an- oder ausgeschaltet werden soll:



**2.1. Leuchtsegment a (Tut)** Entwerft eine Schaltung für das Leuchtsegment a. Geht dabei nach untenstehendem Schema vor:

1. Erstellt eine Wertetabelle für a.
2. Überträgt die Werte in eine KV-Tafel.
3. Ermittelt die minimale DNF aus der KV-Tafel.
4. Zeichnet die Schaltung (alle Gatter sind zugelassen).

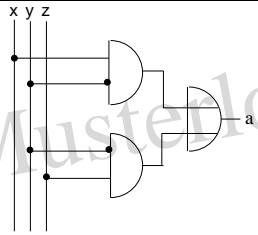
**Lösung:**

x	y	z	a
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	*

0	1	0	0
1	1	*	0

$$a = x \cdot \overline{y} + z \cdot \overline{y}$$



2.2. Wertetabelle (15 Prozent) Stellt die Wertetabelle für die Eingänge x,y,z und die Ausgänge b bis f auf!

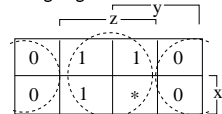
Lösung:

x	y	z	b	c	d	e	f
0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0
0	1	0	0	0	0	1	1
0	1	1	1	1	0	1	0
1	0	0	0	0	1	1	0
1	0	1	1	0	0	0	1
1	1	0	0	1	1	0	1
1	1	1	*	*	*	*	*

2.3. KV-Diagramm (20 Prozent) Ermittelt für die Ausgänge b und e mit je einem KV-Diagramm eine minimale DNF und eine minimale KNF!

Lösung:

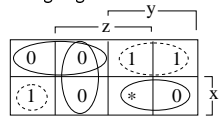
Ausgang b



DNF:  $b = z$

KNF:  $b = z$

Ausgang e



DNF:  $e = y \cdot \bar{x} + x\bar{y} \cdot \bar{z}$

KNF:  $e = (\bar{z} + y) \cdot (x + y) \cdot (\bar{x} + \bar{y})$

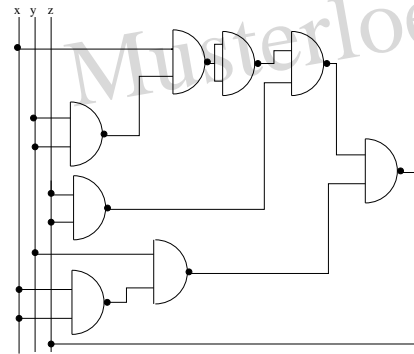
2.4. Schaltentwurf (15 Prozent) Zeichnet für die Ausgänge b und e die entsprechenden Schaltnetz! Es sind nur NAND-Gatter mit 2 Eingängen zu verwenden. Welche Normalform eignet sich am besten zur Realisierung des Schaltnetzes?

Lösung:

Bei NAND-Gattern eignet sich die DNF am besten, da die einzelnen Terme bereits konjugiert sind.

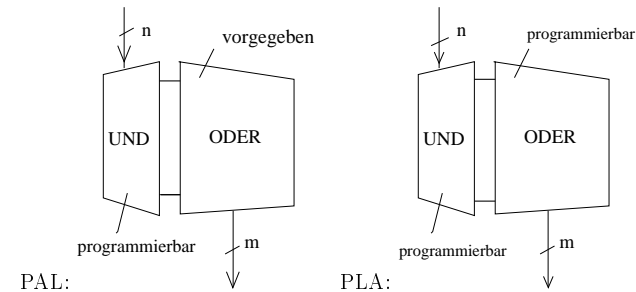
Umformung:

$$e = y \cdot \bar{x} + x \cdot \bar{y} \cdot \bar{z} = \overline{\overline{y \cdot \bar{x} + x \cdot \bar{y} \cdot \bar{z}}} = \overline{\overline{y \cdot \bar{x}} \cdot \overline{x \cdot \bar{y} \cdot \bar{z}}}$$



3. Aufgabe (30 Prozent): ROM, Logikfelder: PAL, PLA

3.1. Wiederholung ROM, PAL, PLA (Tut) Welches sind die Merkmale der einzelnen Bausteine?

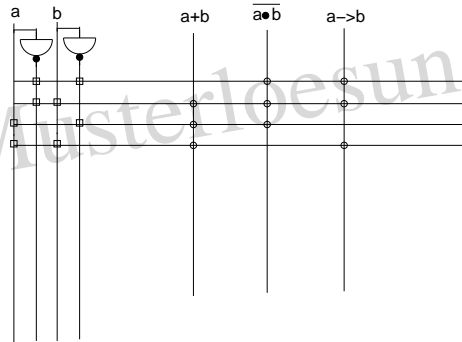


Lösung:

1. ROM steht für Read Only Memory. Diese Bausteine realisieren Schaltnetze mit festen Inhalten, die vom Hersteller durch Maskenprogrammierung festverdrahtet wurden.
2. PAL steht für Programmable Array Logic. Diese Bausteine haben eine programmierbare UND-Matrix und eine feste ODER-Matrix.
3. PLA steht für Programmable Logical Array. PLA-Bausteine haben eine programmierbare UND-Matrix und eine programmierbare ODER-Matrix, deshalb werden sie auch als universelle Bausteine bezeichnet.

3.2. Schaltnetz ROM (Tut) Wie sieht eine Schaltung für einen ROM-Baustein aus? Entwerft eine ROM-Schaltung für die Ausgangssignale:  $a + b$ ,  $a \cdot b$ ,  $a \rightarrow b$

Lösung:



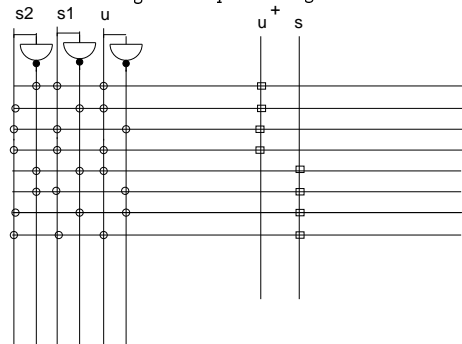
3.3. Schaltnetz PAL (Tut) Zeichnet die Schaltung eines PAL mit der Funktion eines Volladdierers!

$x$	$y$	$u$	$u_+$	$s$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

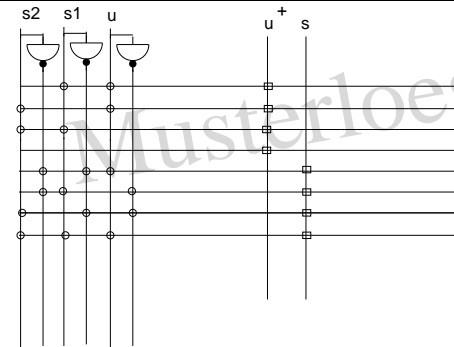
Hinweis:

Lösung:

PAL-Schaltung ohne Optimierung:

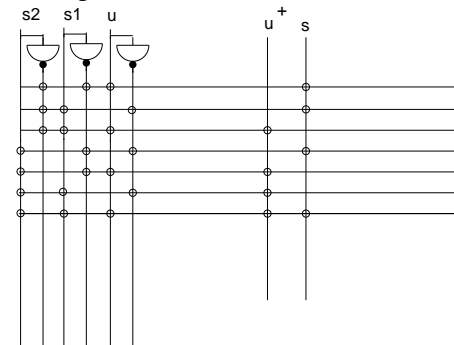


PAL-Schaltung mit Optimierung (mDNF):



3.4. Schaltnetz PLA (30 Prozent) Zeichnet die Schaltung eines Volladdierers mit Hilfe eines PLA. Welche Vorteile hat ein PLA gegenüber einem PAL? Begründet eure Antwort.

Lösung:



Der Vorteil liegt darin, dass sowohl die UND-Matrix als auch die ODER-Matrix programmierbar sind.

Die letzte Zeile der Wertetabelle musste beim PAL in 2 Zeilen untergebracht werden (je ein ODER-Gatter)

Beim PLA koennen mehrfach gebrauchte Konjunktionsterme in einer Zeile zusammengefasst werden.

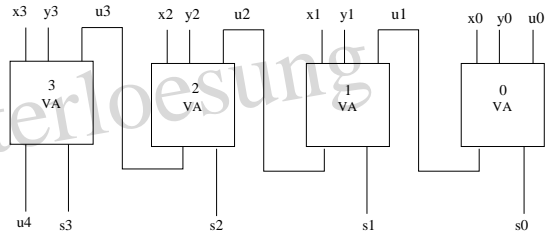
Die folgenden Aufgaben sind zur Bearbeitung in den Tutorien am 11./12.02.2002 gedacht.

#### 4. Aufgabe: Addiertechniken

4.1. Ripple-Carry-Addierer (Tut) Skizziert die Struktur eines 4-Bit Ripple-Carry-Addierers! Welche Schwachstelle des Addierwerkes wird sichtbar?

Lösung:

Der Addierer ist aus vier einzelnen Volladdierern aufgebaut, deren Übertragsausgänge kaskadiert sind.



Das Melden eines Übertragungssignal bis zur Stelle  $n-1$  ist durch die Kaskadierung langsam. Nimmt man an, dass ein Übertrag erst nach  $2ns$  gültig ist, so benötigt ein 4-Bit Carry-Ripple-Addierer für jede Addition  $t = 4 \cdot 2ns = 8ns$ , bis der auslaufende Übertrag  $u_4$  gültig ist. Der Zeitaufwand erhöht sich allerdings noch geringfügig um die Zeit, die der 3. VA für die Berechnung des höchstwertigen Bit braucht. Alternativen für den Ripple-Carry-Addierer sind: Carry Look Ahead, Carry Select und Carry Save.

#### 4.2. Carry-Look-Ahead-Addierer (Tut) .

**Hinweis:** Die Idee des Carry-Look-Ahead-Addierers ist es, durch eine Vorausberechnung des Übertrages die Rechenzeit zu verkürzen.

Für den  $i$ -ten Volladdierer gilt:  $u_{i+1} = x_i \cdot y_i + (x_i + y_i) \cdot u_i = G_i + P_i \cdot u_i$ , wobei

1.  $G_i = x_i \cdot y_i$  angibt, ob in der Stelle  $i$  ein Übertragungssignal erzeugt wird (,Generate'),
2.  $P_i = x_i + y_i$  angibt, ob die Stelle  $i$  das Übertragungssignal der vorangehenden Stelle weiterleitet (,Propagate').

Gebt die Gleichungen aller Überträge eines 4-Bit-Carry-Look-Ahead-Addierers an! Worin besteht der Vorteil gegenüber dem Ripple-Carry-Addierer?

**Lösung:**

Für einen 4-Bit-Carry-Look-Ahead-Addierers ergeben sich folgende Gleichungen für  $u_i$ :

$$u_1 = x_0 \cdot y_0 + (x_0 + y_0) \cdot u_0 := G_0 + P_0 \cdot u_0$$

$$u_2 = G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot u_0$$

$$u_3 = G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot u_0$$

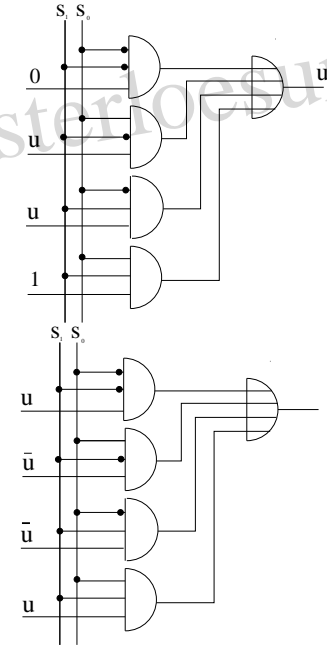
$$u_4 = G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0 + P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot u_0$$

pine

#### 5. Aufgabe: Multiplexer als Volladdierer

Zeichnet die Schaltung eines Volladdierers mit Hilfe je eines Multiplexers für Summe und Übertrag!

**Lösung:**



**Hinweis:** Hier wird der Multiplexer nicht zur Dateneübertragung, sondern zur Datenverarbeitung benutzt, indem von den Eingängen verschiedene Terme zum Ausgang transportiert werden.