

10 Übungsaufgaben mit Lösungen

Zu den einzelnen Kapiteln sind zahlreiche Übungsaufgaben mit ausführlichen Musterlösungen angegeben. Der Leser soll möglichst die Aufgaben selbständig lösen und anschließend sein Ergebnis mit der Musterlösung vergleichen.

| Aufg. | Kap. | Thema |
|-------|---------|---|
| 1, 2 | 2 | Minimieren logischer Gleichungen |
| 3, 4 | 2,4 | Minimieren logischer Gleichungen [VHDL]* zu 4 |
| 5 | 2,4,5 | Entwurf eines 2-Bit-Vergleichers [VHDL]* |
| 6 | 2,4,5 | Schaltnetz zur Wasserstandsregelung [VHDL]* |
| 7 | 3 | Widerstandsdimensionierung für Gatter mit offenem Kollektor |
| 8 | 2,3 | Ansteuerung von Leuchtdioden |
| 9 | 2,5 | Ansteuerung einer 7-Segment-Anzeige |
| 10 | 2,4,5 | Darstellung von Hexadezimalziffern auf einer 7-Segmentanz. [VHDL]* |
| 11 | 2,6 | Zustands- und flankengesteuertes D-Flipflop |
| 12 | 2,6 | Analyse eines Schaltwerks mit D-Flipflops |
| 13 | 6 | Entwurf eines JK- und eines T-Flipflops mit Hilfe eines D-Flipflops |
| 14 | 2,4,5,6 | Steuerung einer Ampelanlage [VHDL]* |
| 15 | 2,6 | Gray-Code-Zähler |
| 16 | 2,6 | Zählerentwurf unter Einsatz eines programmierbaren Zählers |
| 17 | 2,4,6 | Synchroner Modulo-5-Zähler [VHDL]* |
| 18 | 2,4,6 | Entwurf eines synchronen Schaltwerks (Moore-Automat) [VHDL]* |
| 19 | 2,4,6 | Entwurf eines synchronen Schaltwerks (Mealy-Automat) [VHDL]* |
| 20 | 2,4,6 | Entwurf eines synchronen Schaltwerks mit Registerausgabe [VHDL]* |
| 21 | 2,4,6 | Entwurf eines synchronen Schaltwerks mit Festwertspeicher |
| 22 | 2,4,6 | Synchrones Schaltwerk mit Multiplexer und Festwertspeicher |
| 23 | 1,2,7 | Entwurf eines Speichersystems mit 8-Bit-Wortbreite |
| 24 | 1,2,7 | Speichersystem mit 16-Bit-Wortbreite |
| 25 | 2,6,7 | Entwurf eines Schaltnetzes mit einem Festwertspeicher |
| 26 | 2,6 | Entwurf einer Testschaltung |
| 27 | 7,9 | Mikrocontrollersystem mit externer Speichererweiterung |
| 28 | 3,9 | Tastendecodierung mit dem Mikrocontroller 8051 |

Anmerkung:

Bilder und Tabellen in den Übungsaufgaben werden aufgabenweise durchnummeriert und zusätzlich mit "Ü" gekennzeichnet, um Verwechslungen mit den Bild- und Tabellennummern der Kap. 1 bis 9 und 11 zu vermeiden. Aufgaben, die mit [VHDL]* gekennzeichnet sind, enthalten entsprechende VHDL-Modelle.

Aufgabe 1: Minimieren logischer Gleichungen

Gegeben ist folgende logische Gleichung:

$$Y = A B C \vee \overline{A} \overline{B} \overline{C} \vee A B \overline{C} \vee A \overline{B} C \vee A \overline{B} \overline{C}$$

- Vereinfachen Sie die logische Gleichung mit der Booleschen Algebra und geben Sie die negierte und nichtnegierte disjunktive Minimalform an.
- Minimieren Sie die Gleichung mit dem KV-Diagramm und geben Sie die disjunktiven minimalen Gleichungen an. Entwerfen Sie die zugehörigen digitalen Schaltungen.

a) Lösung mit der Booleschen Algebra: Zwei Minterme, die sich nur in einer Variablen unterscheiden, werden zusammengefaßt und nach der Kürzungsregel 15.e) vereinfacht. Nach Rechenregel 7 dürfen dabei Terme mehrfach verwendet werden.

$$A B C \vee A B \overline{C} = A B; \overline{A} \overline{B} \overline{C} \vee A \overline{B} \overline{C} = \overline{B} \overline{C}; A \overline{B} C \vee A \overline{B} \overline{C} = A \overline{B}$$

$$\text{Zwischenergebnis: } Y = A B \vee \overline{B} \overline{C} \vee A \overline{B}$$

Das Verfahren läßt sich erneut auf $A B$ und $A \overline{B}$ anwenden.

$$\text{Endergebnis: } Y = A \vee \overline{B} \overline{C} \quad (\text{Gl.1})$$

Gleichung Gl.1 ist die nichtnegierte disjunktive Minimalform. Die minimale Schaltung ist in Bild Ü1.1 abgebildet. Mit Hilfe der Booleschen Algebra läßt sich aus Gl.1 auch die negierte disjunktive Minimalform herleiten.

$$\overline{Y} = \overline{A \vee \overline{B} \overline{C}} = \overline{A} \wedge (B \vee C) \rightarrow Y = \overline{\overline{A} C \vee \overline{A} B} \quad (\text{Gl.2})$$

Die entsprechende Schaltung zu der negierten disjunktiven Minimalform ist in Bild Ü1.2 abgebildet.

b) Lösung mit dem KV-Diagramm: Die logische Gleichung liegt als disjunktive Normalform vor. Somit läßt sich direkt das zugehörige KV-Diagramm aufstellen (Bild Ü1.1). Die Felder 4, 5, 6 und 7 sowie 4 und 0 lassen sich zu je einem Block zusammenfassen. Die zu den beiden Blöcken zugehörigen konjunktiven Terme werden verodert.

Nach dem KV-Diagramm in Bild Ü1.1 erhält man durch Blockbildung der 1-Felder die nichtnegierte disjunktive Minimalform:

$$Y = A \vee \overline{B} \overline{C} \quad (\text{Gl.1})$$

Faßt man im KV-Diagramm die 0-Felder zusammen (Bild Ü1.2), so erhält man die negierte disjunktive Minimalform:

$$\overline{Y} = \overline{A} C \vee \overline{A} B \quad \text{bzw.} \quad Y = \overline{\overline{A} C \vee \overline{A} B} \quad (\text{Gl.2})$$

| Wahrheitstabelle | | | | | KV-Diagramm | | | | | Minimale Schaltung | | | |
|------------------|---|---|---|---|-------------|--|-----------|--|--|--------------------|---|---|---|
| | A | B | C | Y | C | | \bar{C} | | | A | B | C | Y |
| 0 | 0 | 0 | 0 | 1 | | | | | | | | | |
| 1 | 0 | 0 | 1 | 0 | | | | | | | | | |
| 2 | 0 | 1 | 0 | 0 | | | | | | | | | |
| 3 | 0 | 1 | 1 | 0 | | | | | | | | | |
| 4 | 1 | 0 | 0 | 1 | | | | | | | | | |
| 5 | 1 | 0 | 1 | 1 | | | | | | | | | |
| 6 | 1 | 1 | 0 | 1 | | | | | | | | | |
| 7 | 1 | 1 | 1 | 1 | | | | | | | | | |

| | | C | | \bar{C} | | |
|---|-----------|---|---|-----------|---|---|
| B | \bar{B} | 0 | 1 | 1 | 1 | 0 |
| | | 1 | 0 | 1 | 1 | 1 |
| | | A | A | A | A | A |

$Y = A \vee \bar{B} \bar{C}$

Bild Ü1.1: Schaltungsrealisierung nach der nichtnegierten disjunktiven Minimalform (Gl.1)

Die entsprechende Schaltung ist in Bild Ü1.2 abgebildet.

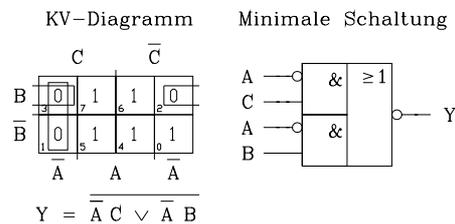


Bild Ü1.2: Schaltungsrealisierung nach der negierten disjunktiven Minimalform (Gl.2)

Aufgabe 2: Minimieren logischer Gleichungen

Gegeben ist folgende logische Gleichung mit den Zwischengrößen U und X.

$$Y = A B U \vee \bar{A} X C \vee [\bar{A} \wedge (B \vee U)] \quad \text{mit } U = B \bar{C} \quad \text{und} \quad X = B \vee \bar{A} C$$

Gesucht sind die disjunktiven Minimalformen. Geben Sie die zugehörigen Schaltungen an. Vergleichen Sie die negierte mit der nichtnegierten Form.

Lösung:

a) **Lösung mit der Booleschen Algebra:** In die logische Gleichung werden die Terme für die Zwischengrößen U und X eingesetzt.

$$Y = A B B \bar{C} \vee [\bar{A} C \wedge (B \vee \bar{A} C)] \vee [\bar{A} \wedge (B \vee B \bar{C})]$$

$$Y = A B \bar{C} \vee \bar{A} B C \vee \bar{A} C \vee \bar{A} B \vee \bar{A} B \bar{C} \quad (\text{nach 6, 11.b, 12.a})$$

$$A B \bar{C} \vee \bar{A} B \bar{C} = B \bar{C} \quad (\text{nach 15.e})$$

$$\bar{A} B C \vee \bar{A} C = \bar{A} C \wedge (B \vee 1) = \bar{A} C \quad (\text{nach 12.a, 2 und 4})$$

$$\text{Zwischenergebnis: } Y = B \bar{C} \vee \bar{A} C \vee \bar{A} B$$

Diese Gleichung scheint minimal zu sein. Erst nach einer weiteren Umformung läßt sich der Term $\neg A B$ noch eliminieren.

Es gilt: $\overline{A} B = \overline{A} B C \vee \overline{A} B \overline{C}$

Somit erhält man: $Y = B \overline{C} \vee \overline{A} C \vee \overline{A} B C \vee \overline{A} B \overline{C}$

$Y = [B \overline{C} \wedge (1 \vee \overline{A})] \vee [\overline{A} C \wedge (1 \vee B)]$ (nach 12.a)

Nach Regel 2 und 4 erhält man die nichtnegierte disjunktive Minimalform:

$Y = B \overline{C} \vee \overline{A} C$ (Gl.1)

Die entsprechende Schaltung ist in Bild Ü2.1 abgebildet.

Mit Hilfe der Booleschen Algebra läßt sich aus Gl.1 auch die negierte disjunktive Minimalform herleiten.

$\overline{Y} = \overline{B \overline{C} \vee \overline{A} C} = (\overline{B} \vee C) \wedge (A \vee \overline{C}) = A \overline{B} \vee A C \vee \overline{B} \overline{C} \vee 0$

$Y = \overline{A \overline{B} C \vee A \overline{B} \overline{C} \vee A C \vee \overline{B} \overline{C}}$

$Y = \overline{[A C \wedge (\overline{B} \vee 1)] \vee [\overline{B} \overline{C} \wedge (A \vee 1)]}$

Endergebnis: $Y = \overline{A C \vee \overline{B} \overline{C}}$ (Gl.2)

Die entsprechende Schaltung ist in Bild Ü2.2 abgebildet.

b) Lösung mit dem KV-Diagramm: Die Ausgangsgleichung wird soweit umgeformt, bis eine disjunktive Form vorliegt.

$Y = A B \overline{C} \vee \overline{A} B C \vee \overline{A} C \vee \overline{A} B \vee \overline{A} B \overline{C}$ (Gl.3)

Die Gleichung Gl.3 bildet die Grundlage für das KV-Diagramm. Im KV-Diagramm werden die "1"-Felder zu Blöcken zusammengefaßt und die nichtnegierte disjunktive Minimalform aufgestellt: $Y = B \overline{C} \vee \overline{A} C$ (Gl.1)

| Wahrheitstabelle | KV-Diagramm | Minimale Schaltung | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|----------------|--------------------|---|----------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|---|--|----------------|--|---|---|---|---|---|---|----------------|---|---|---|---|--|
| <table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>Y</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>2</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>3</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>4</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>5</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>6</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>7</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> </tbody> </table> | | A | B | C | Y | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 2 | 0 | 1 | 0 | 1 | 3 | 0 | 1 | 1 | 1 | 4 | 1 | 0 | 0 | 0 | 5 | 1 | 0 | 1 | 0 | 6 | 1 | 1 | 0 | 1 | 7 | 1 | 1 | 1 | 0 | <table border="1" style="border-collapse: collapse; text-align: center; margin: auto;"> <thead> <tr> <th colspan="2"></th> <th colspan="2">C</th> <th colspan="2">\overline{C}</th> </tr> </thead> <tbody> <tr> <th rowspan="2">B</th> <th>A</th> <td style="border: 1px solid black;">1</td> <td>0</td> <td style="border: 1px solid black;">1</td> <td style="border: 1px solid black;">1</td> </tr> <tr> <th>\overline{A}</th> <td style="border: 1px solid black;">1</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table> <p style="text-align: center;">$Y = B \overline{C} \vee \overline{A} C$</p> | | | C | | \overline{C} | | B | A | 1 | 0 | 1 | 1 | \overline{A} | 1 | 0 | 0 | 0 | |
| | A | B | C | Y | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 0 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | 0 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | 1 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 1 | 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | 1 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | 1 | 1 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | C | | \overline{C} | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| B | A | 1 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | \overline{A} | 1 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Bild Ü2.1: Schaltungsrealisierung nach der nichtnegierten disjunktiven Minimalform

Im KV-Diagramm (Bild Ü2.2) werden die "0"-Felder zu Blöcken zusammengefaßt und die negierte disjunktive Minimalform aufgestellt:

$Y = \overline{A C \vee \overline{B} \overline{C}}$ (Gl.2)

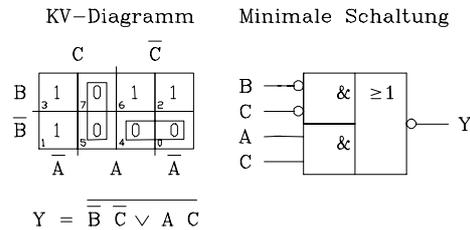


Bild Ü2.2: Schaltungsrealisierung nach der negierten disjunktiven Minimalform

Beide Schaltungen enthalten je zwei UND-Gatter mit zwei Eingängen, zwei Inverter und ein ODER- bzw. NOR-Gatter. Sie sind in der Realisierung gleichwertig.

Aufgabe 3: Minimieren logischer Gleichungen

Gegeben ist die Wahrheitstabelle Tabelle Ü3.1 mit den Eingangsvariablen X_1, X_2, X_3 und den Ausgangsvariablen Y_1 und Y_2 .

Tabelle Ü3.1: Wahrheitstabelle zur Aufgabenstellung

| | X_1 | X_2 | X_3 | Y_1 | Y_2 |
|---|-------|-------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 2 | 0 | 1 | 0 | 1 | 1 |
| 3 | 0 | 1 | 1 | 0 | 0 |
| 4 | 1 | 0 | 0 | 1 | 0 |
| 5 | 1 | 0 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 | 0 | 1 |
| 7 | 1 | 1 | 1 | 0 | 1 |

3.1 Bestimmen Sie aus der Wahrheitstabelle die logischen Gleichungen unter Anwendung der disjunktiven oder konjunktiven Normalform.

3.2 Geben Sie die disjunktiven Minimalformen an und skizzieren Sie die entsprechenden Schaltungen.

Lösung zu 3.1:

a) Disjunktive (DNF) und konjunktive Normalform (KNF) zu Y_1 :

$$\text{DNF: } Y_1 = \bar{X}_1 \bar{X}_2 X_3 \vee \bar{X}_1 X_2 \bar{X}_3 \vee X_1 \bar{X}_2 \bar{X}_3 \quad (\text{Gl.1})$$

$$\begin{aligned} \text{KNF: } Y_1 &= (X_1 \vee X_2 \vee X_3) \wedge (X_1 \vee X_2 \vee \bar{X}_3) \wedge (\bar{X}_1 \vee X_2 \vee \bar{X}_3) \wedge \\ &\quad \wedge (\bar{X}_1 \vee \bar{X}_2 \vee X_3) \wedge (\bar{X}_1 \vee \bar{X}_2 \vee \bar{X}_3) \quad (\text{Gl.2}) \end{aligned}$$

b) Disjunktive (DNF) und konjunktive Normalform (KNF) zu Y_2 :

$$\text{DNF: } Y_2 = \bar{X}_1 \bar{X}_2 X_3 \vee \bar{X}_1 X_2 \bar{X}_3 \vee X_1 \bar{X}_2 X_3 \vee X_1 X_2 \bar{X}_3 \vee X_1 X_2 X_3 \quad (\text{Gl.3})$$

$$\text{KNF: } Y_2 = (X_1 \vee X_2 \vee X_3) \wedge (X_1 \vee \overline{X_2} \vee \overline{X_3}) \wedge (\overline{X_1} \vee X_2 \vee X_3) \quad (\text{Gl.4})$$

Lösung zu 3.2:

Ausgehend von den disjunktiven Normalformen werden die Gleichungen Gl.1 und Gl.3 mit Hilfe des KV-Diagramms minimiert.

a) Disjunktive Minimalformen zu Y1: In Bild Ü3.1 wird die zu den markierten 1-Feldern zugehörige nichtnegierte disjunktive Minimalform aufgestellt. Die entsprechende Schaltung ist ebenfalls in Bild Ü3.1 dargestellt.

$$Y_1 = \overline{X_1} \overline{X_2} X_3 \vee \overline{X_1} X_2 \overline{X_3} \vee X_1 \overline{X_2} \overline{X_3} \quad (\text{Gl.5})$$

In Bild Ü3.2 werden die 0-Felder zu Blöcken zusammengefaßt; die zugehörige nichtnegierte disjunktive Minimalform aufgestellt. Die entsprechende Schaltung ist ebenfalls in Bild Ü3.2 dargestellt.

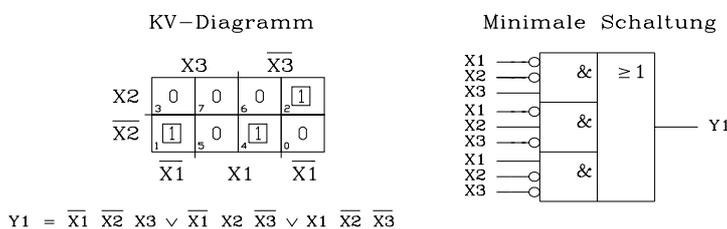


Bild Ü3.1: Lösung zu Y1 nach nichtnegierter disjunktiver Minimalform (Gl.5)

$$Y_1 = \overline{\overline{\overline{\overline{X_1} \overline{X_2} \overline{X_3}} \vee X_1 X_2} \vee X_2 X_3} \vee X_1 X_3 \quad (\text{Gl.6})$$

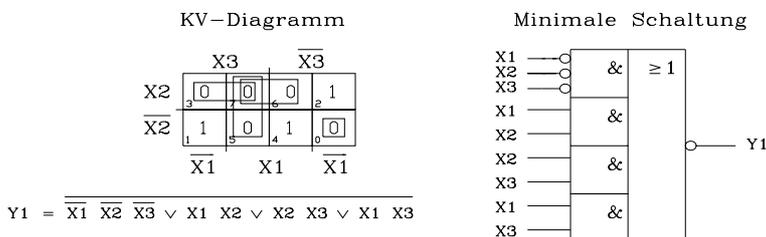
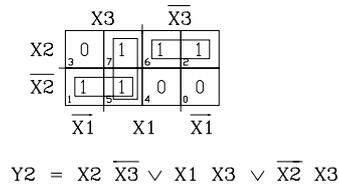


Bild Ü3.2: Lösung zu Y1 nach negierter disjunktiver Minimalform (Gl.6)

b) Disjunktive, minimale Gleichungen zu Y2: In Bild Ü3.3 werden die 1-Felder zu Blöcken zusammengefaßt und die zugehörige nichtnegierte disjunktive Minimalform aufgestellt.

$$Y_2 = X_2 \overline{X_3} \vee X_1 X_3 \vee \overline{X_2} X_3 \quad (\text{Gl.7})$$

Die entsprechende Schaltung ist ebenfalls in Bild Ü3.3 dargestellt.



Minimale Schaltung

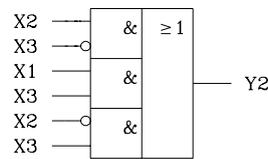
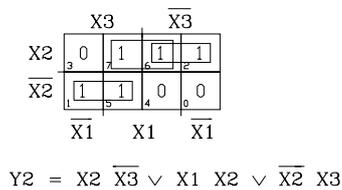


Bild Ü3.3: Lösung zu Y2 nach nichtnegierter disjunktiver Minimalform (Gl.7)

Alternativ zu der in Bild Ü3.3 hergeleiteten minimalen Gleichung ist eine gleichwertige Lösung in nichtnegierter Form möglich, die in Bild Ü3.4 angegeben ist.

$$Y2 = X2 \overline{X3} \vee X1 X2 \vee \overline{X2} X3 \quad (\text{Gl.8})$$



Minimale Schaltung

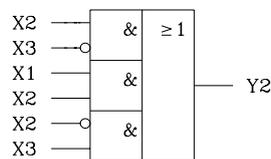
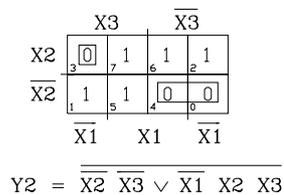


Bild Ü3.4: Gleichwertige Lösung zu Y2 nach nichtnegierter Minimalform (Gl.8)

In Bild Ü3.5 werden die 0-Felder zu Blöcken zusammengefaßt und die zugehörige negierte disjunktive Minimalform aufgestellt.

$$Y2 = \overline{X2} \overline{X3} \vee \overline{X1} X2 X3 \quad (\text{Gl.9})$$

Die entsprechende Schaltung ist ebenfalls in Bild Ü3.5 dargestellt.



Minimale Schaltung

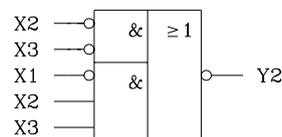


Bild Ü3.5: Lösung zu Y1 nach negierter disjunktiver Minimalform (Gl.9)

Bewertung:

- Für Y1 ist die Schaltungsrealisierung nach der nichtnegierten disjunktiven Minimalform (Gl.5) günstiger, da im Vergleich mit der negierten disjunktiven Minimalform (Gl.6) ein UND-Term weniger erforderlich ist.
- Ein Vergleich der beiden Lösungen für Y2 zeigt, daß die negierte disjunktive Minimalform (Gl.9) einen UND-Term weniger als die nichtnegierte disjunktive Minimalform (Gl.8) aufweist und somit günstiger für die Schaltungsrealisierung ist.

Aufgabe 4: Minimieren logischer Gleichungen

Bestimmen Sie aus der Wahrheitstabelle (Tabelle Ü4.1) mit Hilfe des KV-Diagramms die negierten und nichtnegierten Minimalformen für Y1, Y2 und Y3.

Tabelle Ü4.1: Wahrheitstabelle zur 4. Aufgabe

| | X1 | X2 | X3 | X4 | Y1 | Y2 | Y3 |
|----|----|----|----|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | * | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 4 | 0 | 1 | 0 | 0 | 1 | 1 | * |
| 5 | 0 | 1 | 0 | 1 | 0 | 0 | * |
| 6 | 0 | 1 | 1 | 0 | 0 | * | 0 |
| 7 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 8 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 9 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 10 | 1 | 0 | 1 | 0 | 0 | * | * |
| 11 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 12 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 13 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 14 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 15 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

Lösung:

KV-Diagramm für Y1

| | | | | | |
|-----------------|-----------------|----|-----------------|----|-----------------|
| | X4 | | $\overline{X4}$ | | |
| X3 | 3 | 7 | 6 | 2 | $\overline{X1}$ |
| | 11 | 15 | 14 | 10 | X1 |
| $\overline{X3}$ | 9 | 13 | 12 | 8 | $\overline{X1}$ |
| | 1 | 5 | 4 | 0 | X1 |
| | $\overline{X2}$ | | X2 | | $\overline{X2}$ |

$$Y1 = \overline{X3} \overline{X4} \vee X1 X2 \overline{X4}$$

KV-Diagramm für Y2

| | | | | | |
|-----------------|-----------------|----|-----------------|----|-----------------|
| | X4 | | $\overline{X4}$ | | |
| X3 | 3 | 7 | 6 | 2 | $\overline{X1}$ |
| | 11 | 15 | 14 | 10 | X1 |
| $\overline{X3}$ | 9 | 13 | 12 | 8 | $\overline{X1}$ |
| | 1 | 5 | 4 | 0 | X1 |
| | $\overline{X2}$ | | X2 | | $\overline{X2}$ |

$$Y2 = \overline{X3} \overline{X4} \vee X2 \overline{X4}$$

KV-Diagramm für Y3

| | | | | | |
|-----------------|-----------------|----|-----------------|----|-----------------|
| | X4 | | $\overline{X4}$ | | |
| X3 | 3 | 7 | 6 | 2 | $\overline{X1}$ |
| | 11 | 15 | 14 | 10 | X1 |
| $\overline{X3}$ | 9 | 13 | 12 | 8 | $\overline{X1}$ |
| | 1 | 5 | 4 | 0 | X1 |
| | $\overline{X2}$ | | X2 | | $\overline{X2}$ |

$$Y3 = X2 \overline{X3} \vee \overline{X3} \overline{X4} \vee X1 \overline{X4} \vee \overline{X1} X2 X4$$

Bild Ü4.1: Nichtnegierte disjunktive Minimalformen für Y1, Y2 und Y3

a) In den KV-Diagrammen (Bild Ü4.1) für die Ausgangsvariablen Y1, Y2 und Y3 werden die 1-Felder unter Berücksichtigung der redundanten Felder zu Blöcken zusammengefaßt und erhält die nichtnegierten disjunktiven Minimalformen:

$$Y1 = \overline{X3} \overline{X4} \vee X1 X2 \overline{X4} \quad (Gl.1)$$

$$Y2 = \overline{X3} \overline{X4} \vee X2 \overline{X4} \quad (Gl.2) \quad \text{alternativ: } Y2 = \overline{X3} \overline{X4} \vee X1 \overline{X4} \quad (Gl.2^*)$$

$$Y3 = X2 \overline{X3} \vee \overline{X3} \overline{X4} \vee X1 \overline{X4} \vee \overline{X1} X2 X4 \quad (Gl.3)$$

b) In den KV-Diagrammen (Bild Ü4.2) für die Ausgangsvariablen Y1, Y2 und Y3 werden die 0-Felder unter Berücksichtigung der redundanten Felder zu Blöcken zusammengefaßt. Somit erhält man die negierten disjunktiven Minimalformen:

$$Y1 = \overline{X4} \vee \overline{X1} X3 \vee \overline{X2} X3 \quad (Gl.4)$$

$$Y2 = \overline{X4} \vee \overline{X1} X3 \quad (Gl.5) \quad \text{alternativ: } Y2 = \overline{X4} \vee \overline{X2} X3 \quad (Gl.5^*)$$

$$Y3 = \overline{X2} X4 \vee X1 X3 X4 \vee \overline{X1} X3 \overline{X4} \quad (Gl.6)$$

KV-Diagramm für Y1

| | | | | |
|-----------------|----|-----------------|-----------------|-----------------|
| | | X4 | $\overline{X4}$ | |
| | | 0 | 0 | 0 |
| X3 | 3 | 7 | 6 | 2 |
| | | 0 | 0 | 1 |
| | | 0 | 0 | 0 |
| | | 0 | 0 | 0 |
| X1 | 11 | 15 | 14 | 10 |
| | | 0 | 0 | 1 |
| $\overline{X3}$ | 9 | 13 | 12 | 8 |
| | | 0 | 0 | 1 |
| | | 0 | 0 | 1 |
| | | 0 | 0 | 1 |
| | | 1 | 5 | 4 |
| | | 0 | 0 | 1 |
| | | $\overline{X2}$ | X2 | $\overline{X2}$ |

$$Y1 = \overline{X4} \vee \overline{X1} X3 \vee \overline{X2} X3$$

KV-Diagramm für Y2

| | | | | |
|-----------------|----|-----------------|-----------------|-----------------|
| | | X4 | $\overline{X4}$ | |
| | | 0 | 0 | * |
| X3 | 3 | 7 | 6 | 2 |
| | | 0 | 0 | 1 |
| | | 0 | 0 | * |
| | | 0 | 0 | 1 |
| X1 | 11 | 15 | 14 | 10 |
| | | 0 | 0 | 1 |
| $\overline{X3}$ | 9 | 13 | 12 | 8 |
| | | 0 | 0 | 1 |
| | | * | 0 | 1 |
| | | 1 | 5 | 4 |
| | | 0 | 0 | 1 |
| | | $\overline{X2}$ | X2 | $\overline{X2}$ |

$$Y2 = \overline{X4} \vee X3 \overline{X1}$$

KV-Diagramm für Y3

| | | | | |
|-----------------|----|-----------------|-----------------|-----------------|
| | | X4 | $\overline{X4}$ | |
| | | 0 | 1 | 0 |
| X3 | 3 | 7 | 6 | 2 |
| | | 0 | 0 | * |
| | | 0 | 0 | 1 |
| | | 0 | 0 | * |
| X1 | 11 | 15 | 14 | 10 |
| | | 0 | 1 | 1 |
| $\overline{X3}$ | 9 | 13 | 12 | 8 |
| | | 0 | 1 | 1 |
| | | 0 | * | * |
| | | 0 | * | * |
| | | 0 | * | 1 |
| | | 1 | 5 | 4 |
| | | 0 | 0 | 1 |
| | | $\overline{X2}$ | X2 | $\overline{X2}$ |

$$Y3 = \overline{X4} \overline{X2} \vee X1 X3 X4 \vee \overline{X1} X3 \overline{X4}$$

Bild Ü4.2: Negierte disjunktive Minimalformen für Y1, Y2 und Y3

Ein Vergleich der beiden Lösungen zeigt, daß die Gleichungen Gl.4, Gl.5 (Gl.5*) und Gl.6 etwas einfacher aufgebaut sind als die Gleichungen Gl.1, Gl.2 (Gl.2*) und Gl.3. Stehen beliebige Gatter für die Schaltungsrealisierung zur Verfügung, so wird der Entwurf mit der negierten Minimalform etwas günstiger. Beim Entwurf eines Schaltnetzes mit programmierbaren Bausteinen, z.B. mit einem PAL, ist der technische Aufwand für die Realisierung in beiden Fällen gleich groß. Es werden in beiden Fällen insgesamt 8 Produktterme benötigt. Setzt man einen PAL mit programmierbarer Makrozelle zur Realisierung der disjunktiven Minimalformen ein, läßt sich am Ausgang die gewünschte Polarität für die negierte/nichtneg. Minimalform einstellen.

VHDL-Modell: Entwurf zu vorgegebener Wahrheitstabelle Tab. Ü4.1

```

library ieee;
use ieee.std_logic_1164.all;

entity uebung_4 is port (
  x:    in std_logic_vector (1 to 4); -- Vektor x mit den Elementen x(1), x(2), x(3), x(4)
  y:    out std_logic_vector (1 to 3); -- Vektor y mit den Elementen y(1), y(2), y(3)
end uebung_4;

architecture verhalten of uebung_4 is begin
wahrheits_tab: process(x) begin
  case x is
    when "0000" => y <= "111";
    when "0001" => y <= "0-0";
    when "0010" => y <= "000";
    when "0011" => y <= "000";
    when "0100" => y <= "11-";
    when "0101" => y <= "00-";
    when "0110" => y <= "0-0";
    when "0111" => y <= "001";
    when "1000" => y <= "111";
    when "1001" => y <= "000";
    when "1010" => y <= "0--";
    when "1011" => y <= "000";
    when "1100" => y <= "111";
    when "1101" => y <= "001";
    when "1110" => y <= "111";
    when "1111" => y <= "000";
    when others => y <= "---";           -- alle anderen Faelle der neunwertigen Logik
  end case;
end process wahrheits_tab;
end verhalten;

```

Aufgabe 5: Entwurf eines 2-Bit-Vergleichers

Entwerfen Sie einen 2-Bit-Vergleicher. Es sollen die 2-Bit-Dualzahlen A und B in Betragsdarstellung miteinander verglichen und das Ergebnis des Vergleichs ausgegeben werden.

Es soll gelten:

| Vergleich | Ausgänge |
|-----------|----------------------------------|
| $A > B$ | $Y_A = 1, Y_B = 0$ und $Y_G = 0$ |
| $A < B$ | $Y_A = 0, Y_B = 1$ und $Y_G = 0$ |
| $A = B$ | $Y_A = 0, Y_B = 0$ und $Y_G = 1$ |

Stellen Sie die nichtnegierten und negierten disjunktiven Minimalformen für Y_A , Y_B und Y_G auf. Welche Gleichungen sind zur Realisierung der digitalen Schaltung nötig, wenn die Anzahl der UND-Verknüpfungen pro Gleichung minimal sein soll?

Lösung:

Tabelle Ü5.1: Wahrheitstabelle zu Aufgabe 5

| | A1 | A0 | B1 | B0 | YA | YB | YG |
|----|----|----|----|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 6 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 10 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 11 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 12 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 14 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 15 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

Hinweis: Die nichtnegierten disjunktiven Minimalformen lassen sich mittels des KV-Diagramms durch Zusammenfassen der 1-Felder aufstellen. Entsprechend werden die negierten disjunktiven Minimalformen durch Blockbildung der 0-Felder aufgestellt.

Nichtnegierte disjunktive Minimalformen:

$$Y_A = A_1 \overline{B_1} \vee A_1 A_0 \overline{B_0} \vee A_0 \overline{B_1} \overline{B_0}$$

$$Y_B = \overline{A_1} B_1 \vee \overline{A_0} B_1 B_0 \vee \overline{A_1} \overline{A_0} B_0$$

$$Y_G = \overline{A_1} \overline{A_0} \overline{B_1} \overline{B_0} \vee \overline{A_1} A_0 \overline{B_1} B_0 \vee A_1 \overline{A_0} B_1 \overline{B_0} \vee A_1 A_0 B_1 B_0$$

Negierte disjunktive Minimalformen:

$$Y_A = \overline{\overline{A_1} B_1} \vee \overline{\overline{A_0} B_1} \vee \overline{B_1 B_0} \vee \overline{\overline{A_1} \overline{A_0}} \vee \overline{\overline{A_1} B_0}$$

$$Y_B = \overline{\overline{B_1} \overline{B_0}} \vee \overline{A_1 \overline{B_1}} \vee \overline{A_1 \overline{B_0}} \vee \overline{A_1 A_0} \vee \overline{A_0 B_1}$$

$$Y_G = \overline{\overline{A_1} B_1} \vee \overline{A_1 \overline{B_1}} \vee \overline{\overline{A_0} B_0} \vee \overline{A_0 \overline{B_0}}$$

Es werden die nichtnegierten disjunktiven benutzt, da die Anzahl der Produktterme im Vergleich zu den negierten Minimalformen geringer ist.

VHDL-Modell: 2-Bit-Vergleicher

```
entity comp is port(
  a,b:    in bit_vector (1 downto 0); -- 2 Eingangsvektoren mit je 2 Elementen
  ya,yb,yg: out bit); -- 3 Ausgaenge
end comp;
```

```
architecture verhalten of comp is begin
vergleich: process (a,b) begin
  if a > b then ya <= '1'; yb <= '0'; yg <= '0';           -- Zahlenvergleich
  elsif a < b then ya <= '0'; yb <= '1'; yg <= '0';
  elsif a = b then ya <= '0'; yb <= '0'; yg <= '1';
  end if;
end process vergleich;
end verhalten;
```

Anmerkung:

Soll ein 4-Bit-Komparator entworfen werden, so wird nur der Datentyp der Vektoren a und b geändert: bit_vector (3 downto 0)

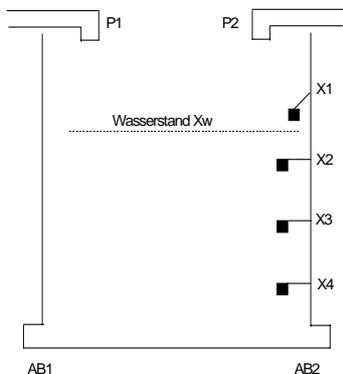
Aufgabe 6: Schaltnetz zur Wasserstandsregelung

Gegeben ist ein Wasserbehälter mit den beiden Abflüssen AB1 und AB2. Der Behälter kann über zwei Pumpen P1 und P2 gespeist werden. Am Behälterrand sind 4 Schwimmer (X1, X2, X3 und X4) angeordnet, über die der Wasserstand Xw überwacht werden kann. Der Wasserstand im Behälter soll über steuerbare Ventile an den Pumpen und Abflüssen geregelt werden (Bild Ü6.1).

Es gelte:

| | | | |
|----------|-----------------------|----------|-----------------------|
| P1 = 1: | Pumpe 1 eingeschaltet | P1 = 0: | Pumpe 1 ausgeschaltet |
| P2 = 1: | Pumpe 2 eingeschaltet | P2 = 0: | Pumpe 2 ausgeschaltet |
| AB1 = 1: | Abfluß 1 geöffnet | AB1 = 0: | Abfluß 1 geschlossen |
| AB2 = 1: | Abfluß 2 geöffnet | AB2 = 0: | Abfluß 2 geschlossen |

- 6.1 Falls eine unerlaubte Kombination der Eingangsvariablen auftritt, so soll eine Fehler-Variable F den Logik-Zustand "1" annehmen. Erstellen Sie die zugehörige Wahrheitstabelle.
- 6.2 Geben Sie die minimalen disjunktiven Gleichungen für alle Ausgangsvariablen in negierter und nichtnegierter Form an. Welche Gleichungen sind günstiger, wenn die Anzahl der Produktterme als Kriterium gewählt wird?



Xw unterhalb von X4:
Beide Pumpen ein; AB1 und AB2 zu

Xw zwischen X4 und X3:
Beide Pumpen ein; AB1 auf, AB2 zu

Xw zwischen X3 und X2:
P1 ein, P2 aus; AB1 zu, AB2 auf

Xw zwischen X2 und X1:
P1 aus, P2 ein; AB1 und AB2 auf

Xw oberhalb von X1:
P1 und P2 aus; AB1 und AB2 auf

Bild Ü6.1: Wasserbehälter zur Aufgabe 6

Lösung:

6.1 Zunächst wird die Wahrheitstabelle Tabelle Ü6.1 aufgestellt.

Tabelle Ü6.1: Wahrheitstabelle zur Aufgabe 6

| | X1 | X2 | X3 | X4 | P1 | P2 | AB1 | AB2 | F |
|----|----|----|----|----|----|----|-----|-----|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | * | * | * | * | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | * | * | * | * | 1 |
| 5 | 0 | 1 | 0 | 1 | * | * | * | * | 1 |
| 6 | 0 | 1 | 1 | 0 | * | * | * | * | 1 |
| 7 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 8 | 1 | 0 | 0 | 0 | * | * | * | * | 1 |
| 9 | 1 | 0 | 0 | 1 | * | * | * | * | 1 |
| 10 | 1 | 0 | 1 | 0 | * | * | * | * | 1 |
| 11 | 1 | 0 | 1 | 1 | * | * | * | * | 1 |
| 12 | 1 | 1 | 0 | 0 | * | * | * | * | 1 |
| 13 | 1 | 1 | 0 | 1 | * | * | * | * | 1 |
| 14 | 1 | 1 | 1 | 0 | * | * | * | * | 1 |
| 15 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |

6.2 Die minimalen disjunktiven Gleichungen werden mit Hilfe des KV-Diagramms aufgestellt.

| Nichtnegierte disjunktive Minimalform | Negierte disjunktive Minimalform |
|--|--|
| $P1 = \overline{X2}$ | $P1 = \overline{X2}$ |
| $P2 = \overline{X3} \vee \overline{X1} X2$ | $P2 = \overline{X1} \vee \overline{X2} X3$ |
| $AB1 = \overline{X3} X4 \vee X2$ | $AB1 = \overline{X2} X3 \vee \overline{X4}$ |
| $AB2 = X3$ | $AB2 = \overline{X3} = X3$ |
| $F = X3 \overline{X4} \vee X2 \overline{X3} \vee X1 \overline{X2}$ | $F = \overline{X1} \overline{X2} \overline{X3} \vee X2 X3 X4 \vee \overline{X1} X3 X4$ |

Die Gleichungen in negierter und nichtnegierter Form sind gleich günstig, da sie gleiche Anzahl an Produkttermen enthalten.

VHDL-Modell: Schaltnetz zur Wasserstandsregelung

```
library ieee;
use ieee.std_logic_1164.all;

entity wasser_beh is port(
  x:      in std_logic_vector (1 to 4); -- Eingangsvektor x: Elemente x(1), x(2), x(3), x(4)
  P1,P2,AB1,AB2,F: out std_logic);    5 Ausgaenge
end wasser_beh;

architecture verhalten of wasser_beh is begin
  schwimmer: process(x)
  begin
    case x is
      when "0000" => -- Wasserstand unterhalb von x4
        P1 <= '1'; P2 <= '1'; AB1 <= '0'; AB2 <= '0'; F <= '0';
      when "0001" => -- Wasserstand zwischen x4 und x3
        P1 <= '1'; P2 <= '1'; AB1 <= '1'; AB2 <= '0'; F <= '0';
      when "0011" => -- Wasserstand zwischen x3 und x2
        P1 <= '1'; P2 <= '0'; AB1 <= '0'; AB2 <= '1'; F <= '0';
      when "0111" => -- Wasserstand zwischen x2 und x1
        P1 <= '0'; P2 <= '1'; AB1 <= '1'; AB2 <= '1'; F <= '0';
      when "1111" => -- Wasserstand oberhalb von x1
        P1 <= '0'; P2 <= '0'; AB1 <= '1'; AB2 <= '1'; F <= '0';
      when others => -- Schwimmer defekt: Fehlermeldung
        P1 <= '1'; P2 <= '1'; AB1 <= '1'; AB2 <= '1'; F <= '1';
    end case;
  end process schwimmer;
end verhalten;
```

Aufgabe 7: Widerstandsdimensionierung für Gatter mit offenem Kollektor

Gegeben sei ein Gatter in Standard TTL mit Open-Kollektor-Ausgang und einem Pull-Up Widerstand R_L an $U_B = 5V$. Der Ausgang werde mit n digitalen Eingängen (Standard TTL) belastet.

Randbedingungen:

- Kollektorströme des Ausgangstransistors:
 - Reststrom des gesperrten Transistors: $I_{C_{Rest}} = 250 \mu A$
 - Max. Kollektorstrom des leitenden Transistors: $I_{C_{max}} = 16mA$
- Eingangsströme der digitalen Eingänge:
 - H-Pegel: $I_{IH} = 40 \mu A$
 - L-Pegel: $I_{IL} = -1,6 mA$

7.1 Gesucht ist der maximale ($R_{L_{max}}$) und minimale Wert ($R_{L_{min}}$) des Pull-Up-Widerstands für $n=5$.

7.2 Wie viele Gattereingänge n dürfen den Ausgang maximal belasten?

7.3 Erweitern Sie die Schaltung auf k kollektorseitig verbundene Ausgangstransistoren. Geben Sie für den Fall k=3 und n=5 den Wert für R_{Lmax} und R_{Lmin} an.

Lösung:

7.1 R_{Lmax} und R_{Lmin} für $n = 5$

a) Annahme: H-Pegel am Kollektor des Transistors. Für diesen Fall muß R_L möglichst klein sein, damit die untere Grenze für H-Pegel nicht unterschritten wird.

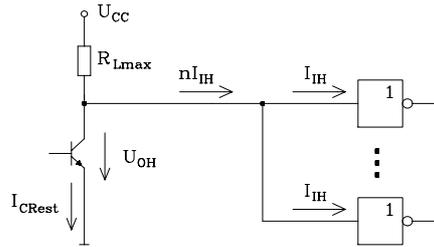


Bild Ü7.1: H-Pegel am Kollektor des Transistors

Es gilt: $U_{OH} \geq 2,4 \text{ V} \implies U_{OHmin} = 2,4 \text{ V}$

$$U_{CC} = (n E I_{iH} + I_{CRest}) E R_{Lmax} + U_{OHmin}$$

$$R_{Lmax} = \frac{U_{CC} - U_{OHmin}}{n \cdot I_{iH} + I_{CRest}} = \frac{5 \text{ V} - 2,4 \text{ V}}{5 \cdot 40 \mu\text{A} + 250 \mu\text{A}} = \frac{2,6 \text{ V}}{450 \mu\text{A}} = 5,77 \text{ k}\Omega$$

b) Annahme: L-Pegel am Kollektor des Transistors. Für diesen Fall muß R_L möglichst groß sein, damit die obere Grenze für L-Pegel nicht überschritten wird.

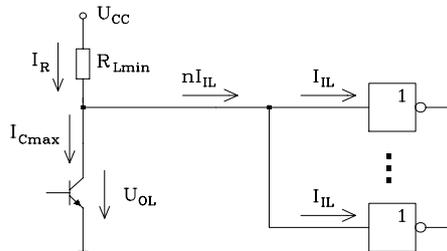


Bild Ü7.2: L-Pegel am Kollektor des Transistors

$$U_{OL} \leq 0,4 \text{ V} \rightarrow U_{OLmax} = 0,4 \text{ V}$$

$$I_{Cmax} = I_R - n E I_{iL}$$

$$U_{CC} = I_R E R_L + U_{OLmax}$$

$$U_{CC} = (n I_{IL} + I_{Cmax}) R_{Lmin} + U_{OLmax}$$

$$R_{Lmin} = \frac{U_{CC} - U_{OLmax}}{n \cdot I_{IL} + I_{Cmax}} = \frac{5 \text{ V} - 0,4 \text{ V}}{5 \cdot (-1,6 \text{ mA}) + 16 \text{ mA}} = \frac{4,6 \text{ V}}{8 \text{ mA}} = 575 \Omega$$

Der Wert des Lastwiderstandes muß zwischen den beiden Grenzwerten R_{Lmax} und R_{Lmin} liegen. Für zeitkritische Anwendungen ist die Gatterdurchlaufzeit zu berücksichtigen. Sie ist abhängig von der Zeitkonstante $R_L E C_L$, mit C_L als Lastkapazität. Je kleiner der Kollektorwiderstand ist, desto kürzer ist die Gatterdurchlaufzeit.

7.2 Maximale Anzahl der Gattereingänge für n

Grenzfall: $R_{Lmin} = R_{Lmax}$

Aus $n = 10$ folgt $R_{Lmin} \rightarrow \infty$, was nicht möglich ist. Die Anzahl der belastenden Gattereingänge darf maximal 9 sein. Für $n = 9$ ergeben sich folgende Werte für den Kollektorwiderstand: $R_{Lmax} = 4,2 \text{ k}\Omega$ und $R_{Lmin} = 2,88 \text{ k}\Omega$

7.3 Erweiterung auf k Ausgangstransistoren

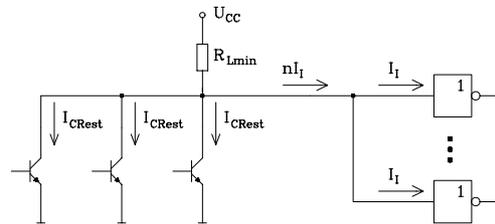


Bild Ü7.3: Parallelschaltung dreier Ausgangsstufen ($k = 3$ und $n = 5$)

$$R_{Lmax} = \frac{U_{CC} - U_{OHmin}}{n \cdot I_{IH} + k \cdot I_{CRest}} = \frac{5 \text{ V} - 2,4 \text{ V}}{5 \cdot 40 \mu\text{A} + 3 \cdot 250 \mu\text{A}} = 2,74 \text{ k}\Omega$$

Für die Berechnung von R_{Lmin} wird der ungünstigste Fall (worst case) angenommen, daß nur ein Transistor durchgeschaltet ist.

$$R_{Lmin} = \frac{U_{CC} - U_{OLmax}}{n \cdot I_{IL} + I_{Cmax}} = \frac{5 \text{ V} - 0,4 \text{ V}}{5 \cdot (-1,6 \text{ mA}) + 16 \text{ mA}} = 575 \Omega$$

Aufgabe 8: Ansteuerung von Leuchtdioden

Zur Anzeige der Logik-Pegel werden in der Digitaltechnik häufig Leuchtdioden (LEDs) eingesetzt, da sie wenig Strom aufnehmen und wenig Platz benötigen. Für die Ansteuerung der LEDs in TTL-Technik können z.B. Gatter mit Gegentakt-Endstufe (Bild Ü8.1 a und b) oder mit Open-Kollektor-Ausgang (Bild Ü8.1 c) eingesetzt werden. Der erforderliche LED-Strom wird mit Hilfe eines Vorwiderstandes eingestellt.

Im Fall a) soll die angeschlossene Leuchtdiode bei H-Pegel und in den Fällen b) und c) bei L-Pegel am Ausgang leuchten. Ergänzen Sie die in Bild Ü8.1 gegebenen Schaltungen und dimensionieren Sie die erforderlichen Vorwiderstände.

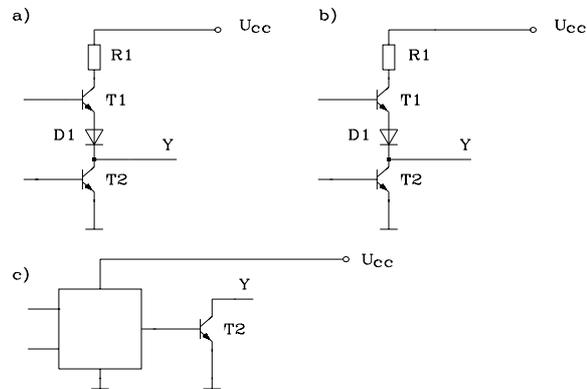


Bild Ü8.1: Gegebene Schaltungen zu Aufgabe 8

Anleitung:

Eine rote Leuchtdiode (GaAsP rot) leuchtet bei einem Strom $I_D \approx 10 \text{ mA}$ so hell, daß sie als Indikator zur Anzeige des Logik-Pegels eingesetzt werden kann. Die Spannung an der Leuchtdiode beträgt hierfür etwa $1,7 \text{ V}$. Ist die an der LED anliegende Spannung deutlich kleiner ($< 1,5 \text{ V}$), bleibt die Leuchtdiode dunkel. Falls die in Bild Ü8.1 eingesetzten Transistoren durchgeschaltet sind, beträgt die Kollektor-Emitter-Restspannung $U_{CE\text{sat}} = 0,3 \text{ V}$. An der Siliziumdiode fällt im durchgeschalteten Zustand eine Spannung von $0,6 \text{ V}$ ab. Der Kollektorwiderstand R_1 in Schaltung a) und b) hat einen Wert von 130Ω . Es gilt: $U_{CC} = 5 \text{ V}$. $I_{C\text{Rest}}$ ist vernachlässigt.

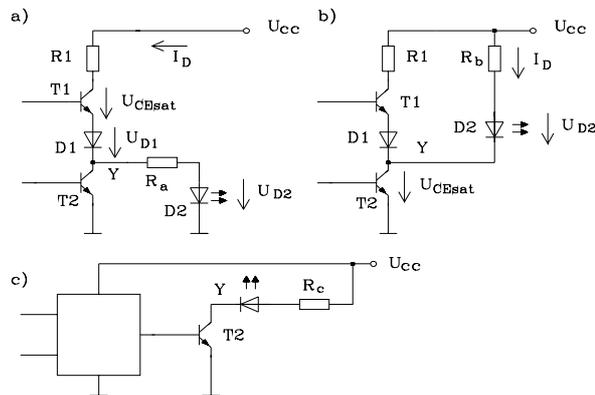


Bild Ü8.2: Gesuchte Schaltungen zu Aufgabe 8

Lösung:

Die gesuchten Schaltungen zur Ansteuerung der LEDs sind in Bild Ü8.2 abgebildet.

Fall a): In diesem Fall wird die LED über den Vorwiderstand R_a angesteuert. Der erforderliche Strom wird am Ausgang des TTL-Gatters zur Verfügung gestellt. Die LED leuchtet bei H-Pegel am Ausgang auf.

$$R_a = \frac{U_{CC} - I_D R_1 - U_{CEsat} - U_{D1} - U_{D2}}{I_D}$$

$$\text{mit } U_{CC} = 5 \text{ V}, R_1 = 130 \Omega, U_{CEsat} = 0,3 \text{ V}, U_{D1} = 0,6 \text{ V},$$

$$U_{D2} = 1,7 \text{ V} \text{ und } I_D = 10 \text{ mA}$$

$$R_a = \frac{5 \text{ V} - 1,3 \text{ V} - 0,3 \text{ V} - 0,6 \text{ V} - 1,7 \text{ V}}{10 \text{ mA}} = 110 \Omega$$

Fall b): In diesem Fall wird die LED über den Vorwiderstand R_b angesteuert. Der erforderliche Strom wird direkt von der Versorgungsspannung zur Verfügung gestellt. Die LED leuchtet bei L-Pegel am Ausgang auf.

$$R_b = \frac{U_{CC} - U_{CEsat} - U_{D2}}{I_D}$$

$$\text{mit } U_{CC} = 5 \text{ V}, U_{CEsat} = 0,3 \text{ V}, U_{D2} = 1,7 \text{ V} \text{ und } I_D = 10 \text{ mA}$$

$$R_b = \frac{5 \text{ V} - 0,3 \text{ V} - 1,7 \text{ V}}{10 \text{ mA}} = 300 \Omega \text{ (gewählt } 270 \Omega)$$

Fall c): In diesem Fall wird die LED am Kollektor eines Gatters mit offenem Kollektorausgang über einen Vorwiderstand R_c angesteuert. Der erforderliche Strom wird direkt von der Versorgungsspannung zur Verfügung gestellt. Die LED leuchtet bei L-Pegel am Ausgang auf. Es gelten für R_c die gleichen Dimensionierungsbedingungen wie im Fall b).

$$R_c = \frac{5 \text{ V} - 0,3 \text{ V} - 1,7 \text{ V}}{10 \text{ mA}} = 300 \Omega \text{ (gewählt } 270 \Omega)$$

Ein Vergleich der drei Treiberschaltungen zeigt, daß im Fall a) die im treibenden Gatter entstehende Verlustleistung am größten ist. Sie beträgt für die oben angegebenen Werte 22 mW. Für die beiden anderen Fälle b) und c) ist die im treibenden Ausgangstransistor entstehende Verlustleistung mit 3 mW verhältnismäßig klein.

$$P_{Va} = I_D^2 * R_1 + (U_{CEsat} + U_{D1})I_D = 13 \text{ mW} + 9 \text{ mW} = 22 \text{ mW}$$

$$P_{Vb} = P_{Vc} = I_D * U_{CEsat} = 3 \text{ mW}$$

Aufgabe 9: Ansteuerung einer 7-Segment-Anzeige

Mit Hilfe einer 7-Segment-Anzeige sollen die in der Tabelle angegebenen Zeichen in Abhängigkeit der Eingangsvariablen U, V und W dargestellt werden. Es sei bekannt, daß nur diese Eingangsvariablenbelegungen vorkommen.

Entwerfen Sie hierfür eine minimale Schaltung.

| Tabelle | | | Zeichen | 7-Segment-Anzeige | Es gilt folgende Zuordnung: Ein Segment der Anzeige leuchtet, wenn die zugehörige Steuervariable (Sa, Sb, Sc, ...) den Logik-Zustand 1 annimmt. |
|---------|---|---|---------|-------------------|--|
| U | V | W | | | |
| 0 | 0 | 0 | □ | | |
| 0 | 0 | 1 | □ | | |
| 0 | 1 | 0 | □ | | |
| 1 | 0 | 0 | □ | | |

Bild Ü9.1: Darstellung von Sonderzeichen auf einer 7-Segment-Anzeige

Lösung:

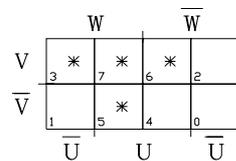
Zur Lösung der gestellten Aufgabe wird eine Wahrheitstabelle aufgestellt, in der die Logik-Zustände der Ausgangsvariablen Sa bis Sg in Abhängigkeit von den Eingangsvariablen U, V und W dargestellt sind. Für die nicht auftretenden Kombinationen der Eingangsvariablen werden redundante Terme (*) auf der Ausgangsseite eingetragen.

Tabelle Ü9.1: Wahrheitstabelle zur Ermittlung der Steuervariablen

| | U | V | W | Sa | Sb | Sc | Sd | Se | Sf | Sg |
|---|---|---|---|----|----|----|----|----|----|----|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 2 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 3 | 0 | 1 | 1 | * | * | * | * | * | * | * |
| 4 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 5 | 1 | 0 | 1 | * | * | * | * | * | * | * |
| 6 | 1 | 1 | 0 | * | * | * | * | * | * | * |
| 7 | 1 | 1 | 1 | * | * | * | * | * | * | * |

Mit Hilfe des KV-Diagramms (Bild Ü9.2) werden die disjunktiven, minimalen Gleichungen ermittelt.

KV-Diagramm



$$S_a = (0) \vee (1) \vee (2) = \bar{U}$$

$$S_b = (0) \vee (1) \vee (4) = \bar{V}$$

$$S_c = (0) \vee (1) \vee (4) = \bar{V}$$

$$S_d = (0) = \bar{U} \bar{V} \bar{W}$$

$$S_e = S_f = (0) \vee (1) \vee (2) \vee (4) = 1$$

$$S_g = (1) \vee (2) \vee (4) = U \vee V \vee W = \bar{S_d}$$

Bild Ü9.2: KV-Diagramm zur Bestimmung der minimalen Gleichungen

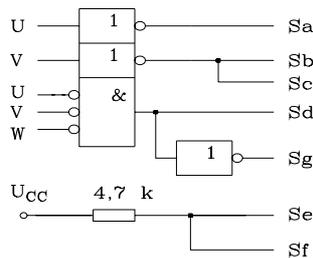


Bild Ü9.3: Gesuchte digitale Schaltung

Aufgabe 10: Darstellung von Hexadezimalziffern auf einer 7-Segment-Anzeige

Mittels einer 7-Segment-Anzeige sollen die in Tabelle Ü10.1 gegebenen hexadezimalen Ziffern in Abhängigkeit der Eingangsvariablen X_1, X_2, X_3 und X_4 dargestellt werden. Entwerfen Sie dafür eine minimale Schaltung und geben Sie die logischen Gleichungen in disjunktiver Minimalform an. Wählen Sie jeweils die Gleichung mit der geringsten Anzahl an UND-Verknüpfungen (Produkttermen) aus.

Tabelle Ü10.1: Zuordnung zwischen den Eingangsvariablen $X_1 \dots X_4$ und den Hex-Ziffern

| X_1 | X_2 | X_3 | X_4 | Zeichen |
|-------|-------|-------|-------|---------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 1 | 1 | 3 |
| 0 | 1 | 0 | 0 | 4 |
| 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 1 | 0 | 6 |
| 0 | 1 | 1 | 1 | 7 |
| 1 | 0 | 0 | 0 | 8 |
| 1 | 0 | 0 | 1 | 9 |
| 1 | 0 | 1 | 0 | A |
| 1 | 0 | 1 | 1 | b |
| 1 | 1 | 0 | 0 | C |
| 1 | 1 | 0 | 1 | d |
| 1 | 1 | 1 | 0 | E |
| 1 | 1 | 1 | 1 | F |

Lösung:**Tabelle Ü10.2:** Wahrheitstabelle bezüglich der einzelnen Segmente

| X1 | X2 | X3 | X4 | Sa | Sb | Sc | Sd | Se | Sf | Sg |
|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

Nichtnegierte disjunktive Minimalform:

$$S_a = X_1 \bar{X}_4 \vee X_2 X_3 \vee \bar{X}_2 \bar{X}_4 \vee \bar{X}_1 X_3 \vee X_1 \bar{X}_2 \bar{X}_3 \vee \bar{X}_1 X_2 X_4$$

$$S_b = \bar{X}_1 \bar{X}_2 \vee \bar{X}_2 \bar{X}_4 \vee \bar{X}_1 \bar{X}_3 \bar{X}_4 \vee X_1 \bar{X}_3 X_4 \vee \bar{X}_1 X_3 X_4$$

$$S_c = X_1 \bar{X}_2 \vee \bar{X}_1 X_2 \vee \bar{X}_1 \bar{X}_3 \vee \bar{X}_1 X_4 \vee \bar{X}_3 X_4$$

$$S_d = X_1 \bar{X}_3 \vee X_2 \bar{X}_3 X_4 \vee X_2 X_3 \bar{X}_4 \vee \bar{X}_2 X_3 X_4 \vee \bar{X}_1 \bar{X}_2 \bar{X}_4$$

$$S_e = \bar{X}_2 \bar{X}_4 \vee X_3 \bar{X}_4 \vee X_1 X_3 \vee X_1 X_2$$

$$S_f = \bar{X}_3 \bar{X}_4 \vee X_2 \bar{X}_4 \vee X_1 X_3 \vee X_1 \bar{X}_2 \vee \bar{X}_1 X_2 \bar{X}_3$$

$$S_g = X_1 X_4 \vee \bar{X}_2 X_3 \vee X_3 \bar{X}_4 \vee X_1 \bar{X}_2 \vee \bar{X}_1 X_2 \bar{X}_3$$

Negierte disjunktive Minimalform:

$$\bar{S}_a = \bar{X}_1 \bar{X}_2 \bar{X}_3 X_4 \vee \bar{X}_1 X_2 \bar{X}_3 \bar{X}_4 \vee X_1 \bar{X}_2 X_3 X_4 \vee X_1 X_2 \bar{X}_3 X_4$$

$$\bar{S}_b = \bar{X}_1 X_2 \bar{X}_3 X_4 \vee X_1 X_3 X_4 \vee X_1 X_2 \bar{X}_4 \vee X_2 X_3 \bar{X}_4$$

$$\bar{S}_c = X_1 X_2 \bar{X}_4 \vee \bar{X}_1 \bar{X}_2 X_3 \bar{X}_4 \vee X_1 X_2 X_3$$

$$\bar{S}_d = \bar{X}_1 \bar{X}_2 \bar{X}_3 X_4 \vee \bar{X}_1 X_2 \bar{X}_3 \bar{X}_4 \vee X_2 X_3 X_4 \vee X_1 \bar{X}_2 X_3 \bar{X}_4$$

$$\bar{S}_e = \bar{X}_1 X_4 \vee \bar{X}_1 X_2 \bar{X}_3 \vee \bar{X}_2 \bar{X}_3 X_4$$

$$\bar{S}_f = X_1 X_2 \bar{X}_3 X_4 \vee \bar{X}_1 \bar{X}_2 X_3 \vee \bar{X}_1 X_3 X_4 \vee \bar{X}_1 \bar{X}_2 X_4$$

$$\bar{S}_g = \bar{X}_1 \bar{X}_2 \bar{X}_3 \vee \bar{X}_1 X_2 X_3 X_4 \vee X_1 X_2 \bar{X}_3 \bar{X}_4$$

Die Gleichungen in negierter Minimalform enthalten weniger Produktterme und sind somit günstiger.

VHDL-Modell: Hexadezimalziffern auf 7-Segment

```

library ieee;
use ieee.std_logic_1164.all;

entity uebung_10 is port(
    x:                in std_logic_vector (1 to 4);
    sa,sb,sc,sd,se,sf,sg: out std_logic);
end uebung_10;

architecture verhalten of uebung_10 is
    signal y: std_logic_vector (6 downto 0);    -- y ist Zwischengroesse
begin
    wahrheits_tab: process(x) begin
        case x is
            when "0000" => y <= "1111110";
            when "0001" => y <= "0110000";
            when "0010" => y <= "1101101";
            when "0011" => y <= "1111001";
            when "0100" => y <= "0110011";
            when "0101" => y <= "1011011";
            when "0110" => y <= "1011111";
            when "0111" => y <= "1110000";
            when "1000" => y <= "1111111";
            when "1001" => y <= "1111011";
            when "1010" => y <= "1110111";
            when "1011" => y <= "0011111";
            when "1100" => y <= "1001110";
            when "1101" => y <= "0111101";
            when "1110" => y <= "1001111";
            when "1111" => y <= "1000111";
            when others => y <= "-----";    -- neunwertige Logik
        end case;
    end process wahrheits_tab;
    sa <= y(6); sb <= y(5); sc <= y(4); sd <= y(3); -- nebenlaufige Anweisungen
    se <= y(2); sf <= y(1); sg <= y(0);
end verhalten;

```

Aufgabe 11: Zustands- und flankengesteuertes D-Flipflop

Ein zustands- und ein positiv flankengesteuertes D-Flipflop werden von einem Signal X am D-Eingang und ϵ am Takteingang angesteuert. Der Ausgang des zustandsgesteuerten D-Flipflops sei Qz und der des flankengesteuerten Qf. Vervollständigen Sie den skizzierte Signalzeitplan und geben Sie die Unterschiede der beiden Flipflops an.

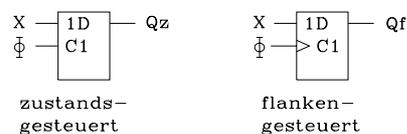


Bild Ü11.1: Ansteuerung der beiden D-Flipflops

Anmerkung: Zu Beginn der Betrachtungen seien beide Flipflops zurückgesetzt. Die Verzögerungszeiten der Flipflops seien vernachlässigbar.

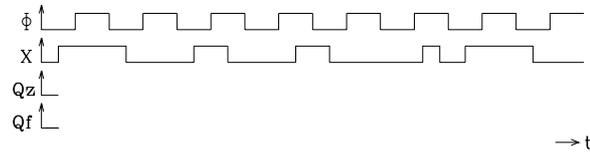


Bild Ü11.2: Vorgegebener Signalverlauf des Taktes und Eingangssignals

Lösung:

Ein zustandsgesteuertes D-Flipflop ist für $\epsilon = 1$ transparent, d. h. eine Änderung am D-Eingang wirkt sich direkt am Ausgang aus (Verzögerungszeiten seien vernachlässigbar). Während der Takt ϵ im Logik-Zustand 0 ist, speichert das zustandsgesteuerte D-Flipflop den Wert.

Dagegen übernimmt das flankengesteuerte D-Flipflop den Logik-Zustand am D-Eingang mit der positiven Taktflanke und speichert ihn bis zur nächstfolgenden positiven Taktflanke.

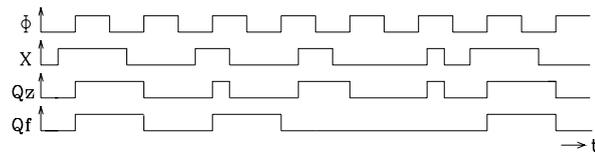


Bild Ü11.3: Vollständiger Signalzeitplan zu Aufgabe 11

Aufgabe 12: Analyse eines Schaltwerks mit D-Flipflops

In einer digitalen Schaltung sollen 3 flankengesteuerte D-Flipflops eingesetzt werden.

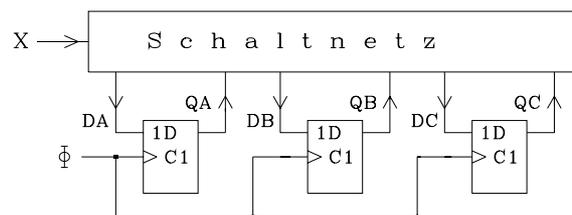


Bild Ü12.1: Blockschaltbild zu Aufgabe 12

Die logischen Gleichungen für die D-Eingänge lauten:

$$DA = \overline{QA}$$

$$DB = \overline{X} \overline{QC} \overline{QB} QA \vee \overline{X} QB \overline{QA} \vee X QB QA \vee X QC \overline{QA}$$

$$DC = X \overline{QC} \overline{QB} \overline{QA} \vee \overline{X} QB QA \vee \overline{X} QC \overline{QA} \vee X QC QA$$

QA, QB, QC sind die Ausgänge der drei D-Flipflops und X ist eine Eingangsvariable (Bild Ü12.1 und Ü12.2). Die Ausgänge QA, QB und QC haben die Wertigkeit 2^0 , 2^1 und 2^2 . Vervollständigen Sie den in Bild Ü12.2 gegebenen Signalzeitplan und charakterisieren Sie die Schaltung.

Für $t = 0$ sollen die drei D-Flipflops rückgesetzt sein.

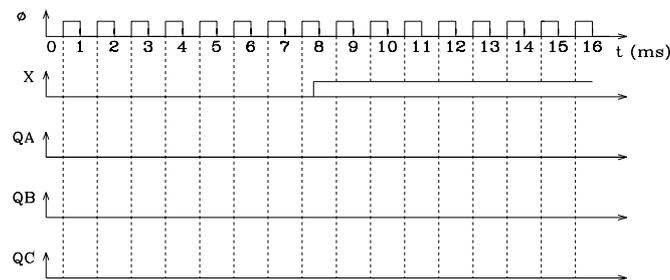


Bild Ü12.2: Signalzeitplan zu Aufgabe 12

Lösung:

Das erste D-Flipflop mit dem Ausgang QA ist als Frequenzteiler (1:2) geschaltet. Der zeitliche Verlauf der Ausgangsgröße QA kann unabhängig von X und den Ausgängen QB und QC bestimmt werden (Bild Ü12.3). Da QB und QC von X, QA, QB und QC abhängen, kann der zeitliche Verlauf dieser beiden FlipflopAusgänge nur für eine Taktperiode, zwischen zwei benachbarten positiven Taktflanken, bestimmt werden. Dann werden die neu ermittelten Logik-Zustände wieder in die Gleichungen eingesetzt und der Signalverlauf für die nächste Periode gewonnen, usw. (Bild Ü12.3).

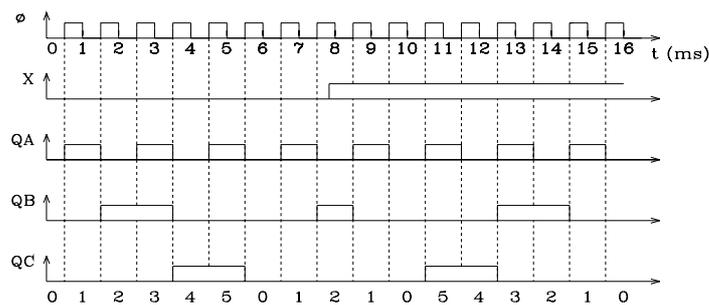


Bild Ü12.3: Vollständiger Signalzeitplan zu Aufgabe 12

Die Schaltung stellt einen synchronen Vorwärts-/Rückwärts-Modulo-6-Zähler dar, der über das Eingangssignal X in der Zählrichtung umgeschaltet werden kann. Der Zähler zählt für X = 0 vorwärts und für X = 1 rückwärts.

Aufgabe 13: Entwurf eines JK- und eines T-Flipflops mit Hilfe eines D-Flipflops

Entwerfen Sie mit Hilfe eines flankengesteuerten D-Flipflops und eines Schaltnetzes ein flankengesteuertes JK- und T-Flipflop.

Lösung:

Es gelten folgende Übergangsbedingungen:

| D-Flipflop | JK-Flipflop | T-Flipflop |
|------------------|---|---|
| $Q^* = D$ (Gl.1) | $Q^* = J \bar{Q} \vee \bar{K} Q$ (Gl.2) | $Q^* = T \bar{Q} \vee \bar{T} Q$ (Gl.3) |

- a) Entwurf des JK-Flipflops durch Gleichsetzen der Übergangsbedingungen nach Gl.1 und Gl.2: $D = J \bar{Q} \vee \bar{K} Q$
- b) Entwurf des T-Flipflops durch Gleichsetzen von Gl.1 und Gl.3:
 $D = T \bar{Q} \vee \bar{T} Q$

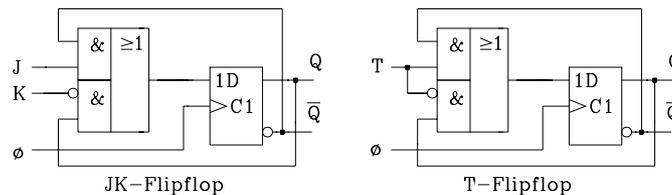


Bild Ü13.1: Realisierung eines JK- und eines T-Flipflops mit Hilfe eines D-Flipflops

Aufgabe 14: Steuerung einer Ampelanlage

Eine Straßenkreuzung mit Haupt- und Nebenstrecke soll von einer Ampelanlage gesteuert werden.

Für die Lichtsignale der Hauptstrecke soll gelten: 14 s grün, 2 s gelb, 14 s rot, 2 s rotgelb, usw.. Für die Lichtsignale der Nebenstrecke soll gelten: 6 s grün, 2 s gelb, 22 s rot, 2 s rotgelb, usw.. Beim Umschalten sollen sich die Rotphasen beider Ampelanlagen für 2 s überlappen. Daraus folgt, daß innerhalb eines Ampelzyklus beide Ampelanlagen zweimal für je 2 s gleichzeitig rot geschaltet sind.

Zum Entwurf werde ein Zähler eingesetzt. Dessen Ausgänge sollen über ein Schaltnetz die sechs Lampen der beiden Ampelanlagen ansteuern.

- 14.1 Geben Sie den Zählertyp und die erforderliche Taktfrequenz an.
- 14.2 Geben Sie in Form einer Wahrheitstabelle die Abhängigkeit der sechs Steuervariablen vom Zählerstand an, beginnen Sie beim Zählerstand 0 mit der Grünphase der Hauptstrecke.
- 14.3 Stellen Sie die logischen Gleichungen für die sechs Steuervariablen auf. Zum Entwurf soll ein PAL mit nichtnegierten Ausgängen eingesetzt werden. Minimieren Sie die Gleichungen, so daß die Anzahl der erforderlichen Produktterme (UND-Verknüpfungen) des PALs minimal wird.

Lösung zu 14.1: Da ein Lichtsignalzyklus der Haupt- und Nebenstrecke 32 s benötigt und die kleinste Zeiteinheit 2 s beträgt, ist eine Zählerkapazität von $32 \text{ s} / 2 \text{ s} = 16$ nötig. Gewählt wird ein asynchroner Vorwärts-Dualzähler mit einer Taktfrequenz von 0,5 Hz. Der Zähler kann asynchron über $\neg R$ rückgesetzt werden (Bild Ü14.3).

Lösung zu 14.2:

Tabelle Ü14.1: Wahrheitstabelle mit den geforderten Steuervariablen

| | QD | QC | QB | QA | GRÜN_H | ROT_H | GELB_H | GRÜN_N | ROT_N | GELB_N |
|----|----|----|----|----|--------|-------|--------|--------|-------|--------|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 5 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 6 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 8 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 9 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 11 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 12 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 14 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 15 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |

Lösung zu 14.3:

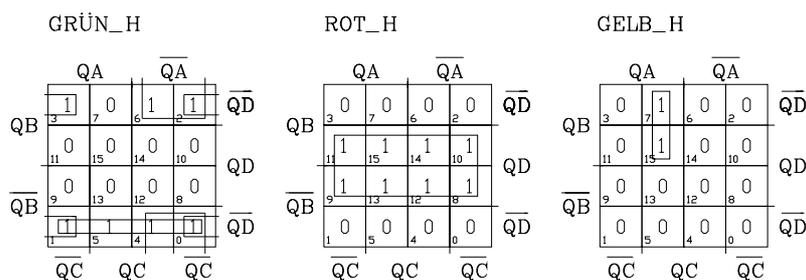


Bild Ü14.1: KV-Diagramme zur Minimierung der logischen Gleichungen GRÜN_H, ROT_H und GELB_H

$$\text{GRÜN_H} = (0) \vee (1) \vee (2) \vee (3) \vee (4) \vee (5) \vee (6) = \overline{QD} \overline{QB} \vee \overline{QD} \overline{QC} \vee \overline{QD} \overline{QA}$$

$$\text{ROT_H} = (8) \vee (9) \vee (10) \vee (11) \vee (12) \vee (13) \vee (14) \vee (15) = QD$$

$$\text{GELB_H} = (7) \vee (15) = QC \overline{QB} \overline{QA}$$

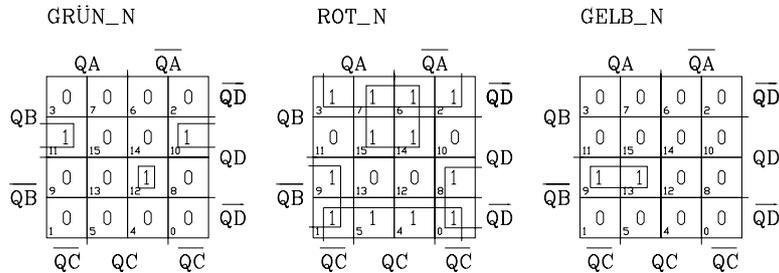


Bild Ü14.2: KV-Diagramme zur Minimierung der logischen Gleichungen GRÜN_N, ROT_N und GELB_N

$$\text{GRÜN_N} = (10) \vee (11) \vee (12) = QD \overline{QB} \overline{QC} \vee QD \overline{QC} \overline{QB} \overline{QA}$$

$$\text{ROT_N} = (0) \vee (1) \vee (2) \vee (3) \vee (4) \vee (5) \vee (6) \vee (7) \vee (8) \vee (9) \vee (14) \vee (15) = \overline{QD} \vee \overline{QC} \overline{QB} \vee QC \overline{QB}$$

$$\text{GELB_N} = (9) \vee (13) = QD \overline{QB} \overline{QA}$$

In Bild Ü14.3 ist das Blockschaltbild der Ampelsteuerung abgebildet. Der eingesezte Zähler kann über $\neg R = 0$ rückgesetzt werden. Dadurch wird die Grünphase für die Hauptstrecke eingeleitet.

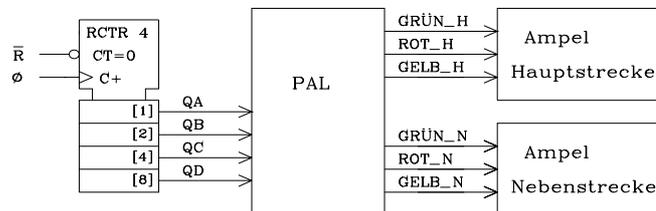


Bild Ü14.3: Blockschaltbild der Ampelsteuerung

VHDL-Modell: Ampelanlage

library ieee; **use** ieee.std_logic_1164.all; **use** work.std_arith.all;

```
entity ampel is port (
    clk, r_neg:          in std_logic;    -- r_neg = reset nicht
    gruen_h, rot_h, gelb_h, gruen_n, rot_n, gelb_n: out std_logic);
end ampel;
```

```

architecture verhalten of ampel is    -- Architektur mit 2 Prozessen
    signal zaehl: std_logic_vector(3 downto 0);    -- interner Zaehler
begin
    zaehler: process (r_neg, clk)    -- Prozess zur Modellierung des Zaehlers
    begin
        if r_neg='0' then zaehl <= "0000";    -- asynchroner Reset
        elsif (clk'event and clk='1') then zaehl <= zaehl +1;
        end if;
    end process zaehler;
    ausgabe: process (zaehl)    -- Prozess fuer die Ausgabe
    begin
        if (zaehl < "0111") then    gruen_h <= '1'; rot_h <= '0'; gelb_h <= '0';
            gruen_n <= '0'; rot_n <= '1'; gelb_n <= '0';
        elsif (zaehl = "0111") then gruen_h <= '0'; rot_h <= '0'; gelb_h <= '1';
            gruen_n <= '0'; rot_n <= '1'; gelb_n <= '0';
        elsif (zaehl = "1000") then gruen_h <= '0'; rot_h <= '1'; gelb_h <= '0';
            gruen_n <= '0'; rot_n <= '1'; gelb_n <= '0';
        elsif (zaehl = "1001") then gruen_h <= '0'; rot_h <= '1'; gelb_h <= '0';
            gruen_n <= '0'; rot_n <= '1'; gelb_n <= '1';
        elsif (zaehl > "1001" and zaehl < "1101") then
            gruen_h <= '0'; rot_h <= '1'; gelb_h <= '0';
            gruen_n <= '1'; rot_n <= '0'; gelb_n <= '0';
        elsif (zaehl = "1101") then gruen_h <= '0'; rot_h <= '1'; gelb_h <= '0';
            gruen_n <= '0'; rot_n <= '0'; gelb_n <= '1';
        elsif (zaehl = "1110") then gruen_h <= '0'; rot_h <= '1'; gelb_h <= '0';
            gruen_n <= '0'; rot_n <= '1'; gelb_n <= '0';
        elsif (zaehl = "1111") then gruen_h <= '0'; rot_h <= '1'; gelb_h <= '1';
            gruen_n <= '0'; rot_n <= '1'; gelb_n <= '0';
        end if;
    end process ausgabe;
end verhalten;

```

Aufgabe 15: Gray-Code-Zähler

Entwerfen Sie einen synchronen *Dezimalzähler*, der im *Graycode* vorwärts zählt. Geben Sie die nichtnegierten disjunktiven Minimalformen und die Schaltung an.

Lösung:

Tabelle Ü15.1: Wahrheitstabelle für den Entwurf mit T-Flipflops

| MT | Zahl | QD | QC | QB | QA | QD* | QC* | QB* | QA* | TD | TC | TB | TA |
|----|------|----|----|----|----|-----|-----|-----|-----|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 3 | 2 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 2 | 3 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 6 | 4 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 7 | 5 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 5 | 6 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 4 | 7 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |

| MT | Zahl | QD | QC | QB | QA | QD* | QC* | QB* | QA* | TD | TC | TB | TA |
|----|------|----|----|----|----|-----|-----|-----|-----|----|----|----|----|
| 12 | 8 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 13 | 9 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 15 | PT | 1 | 1 | 1 | 1 | * | * | * | * | * | * | * | * |
| 14 | PT | 1 | 1 | 1 | 0 | * | * | * | * | * | * | * | * |
| 10 | PT | 1 | 0 | 1 | 0 | * | * | * | * | * | * | * | * |
| 11 | PT | 1 | 0 | 1 | 1 | * | * | * | * | * | * | * | * |
| 9 | PT | 1 | 0 | 0 | 1 | * | * | * | * | * | * | * | * |
| 8 | PT | 1 | 0 | 0 | 0 | * | * | * | * | * | * | * | * |

Anmerkungen zur Wahrheitstabelle:

Falls beim Inkrementieren des Zählers der T-Flipflop Ausgang sich ändern soll, muß der T-Eingang "1" gesetzt sein. Soll der Ausgang sich nicht ändern, wird der T-Eingang "0" gesetzt. Für die Feldnumerierungen im KV-Diagramm sind die Dezimalzahlen der Minterme (Spalte MT) maßgeblich.

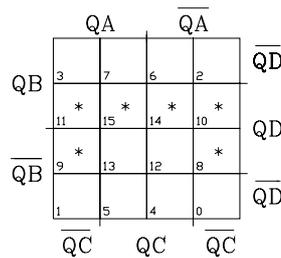
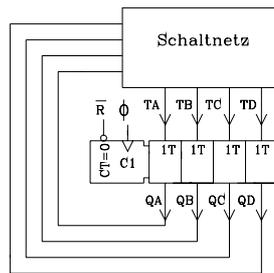
Nichtnegierte disjunktive Minimalform:

$$TD = \overline{QD} QC \overline{QB} \overline{QA} \vee QD QA$$

$$TC = \overline{QC} QB \overline{QA} \vee QD QA$$

$$TB = \overline{QC} \overline{QB} QA \vee QC QB QA$$

$$TA = QD \vee \overline{QC} \overline{QB} \overline{QA} \vee QC \overline{QB} QA \vee \overline{QC} QB QA \vee QC QB \overline{QA}$$

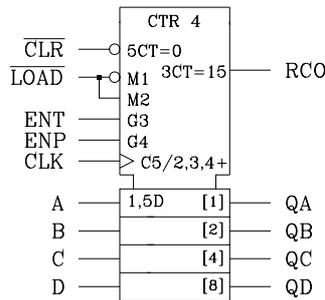


Aufgabe 16: Zählerentwurf unter Einsatz eines programmierbaren Zählers

Entwerfen Sie zwei synchrone Zähler mit den Zählfolgen a) und b):

a) 0,1,2,3,4,5,6,7,8,9,10,11,0, ... b) 3,4,5,6,7,8,9,10,11,12,3, ...

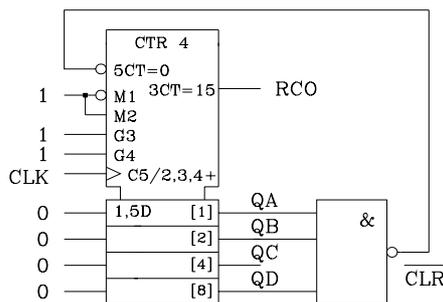
Anleitung: Setzen Sie den integrierten synchronen Dualzähler vom Typ 74LS163 (Kap. 6.2.2, Bild 6.40) ein. Durch eine entsprechende Rückkopplung läßt sich der Dualzähler auf die angegebenen Zählfolgen einstellen.



Besondere Eigenschaften des Zählers 74LS163:
Synchrones Rücksetzen und Laden

Bild Ü16.1: Programmierbarer synchroner Dualzähler

Lösung zu a): Der Zähler soll von 0 bis 11 aufwärts zählen und mit der nächsten positiven Taktflanke wieder bei 0 beginnen. Dazu wird der synchrone Dualzähler vom Typ 74LS163 eingesetzt, der synchron rückgesetzt wird, wenn am Rücksetzeingang eine "0" liegt. Mit Hilfe des KV-Diagramms (Bild Ü16.2) läßt sich die minimale Gleichung für das Rücksetzsignal $\overline{\text{CLR}} = \overline{(\text{QA} \text{ QB} \text{ QD})}$ aufstellen. Über ein NAND-Gatter mit drei Eingängen wird die Rückkopplung von den Zählerausgängen auf den Rücksetzeingang aufgebaut (Bild Ü16.2). Alternativ könnte der NAND-Ausgang auch auf den Ladeeingang rückgekoppelt werden. Dann müßte der Rücksetzeingang an "1" gelegt werden.



| | | | | |
|----|--|----|----|----|
| | QA | QA | | |
| | 0 | 0 | 0 | 0 |
| QB | 3 | 7 | 6 | 2 |
| | 0 | 0 | * | 0 |
| | 1 | 15 | 14 | 10 |
| | 0 | * | * | 0 |
| QD | 9 | 13 | 12 | 8 |
| | 0 | 0 | 0 | 0 |
| QC | 1 | 5 | 4 | 0 |
| | 0 | 0 | 0 | 0 |
| | QC | QC | QC | |
| | $\overline{\text{CLR}} = \overline{\text{QA} \text{ QB} \text{ QD}}$ | | | |

Bild Ü16.2: Gesuchte Schaltung zu a)

Lösung zu b): In diesem Fall soll der Zähler aufwärts von 3 bis 12 zählen und dann wieder bei 3 beginnen. Mit Hilfe eines NAND-Gatters in der Rückkopplung von den Zählerausgängen auf den Ladeeingang wird beim Zählerstand 12 eine "0" an den Ladeeingang gelegt, und mit der nächsten positiven Taktflanke wird die an den Dateneingängen anliegende Dualzahl 0011 als neuer Anfangswert in den Zähler geladen.

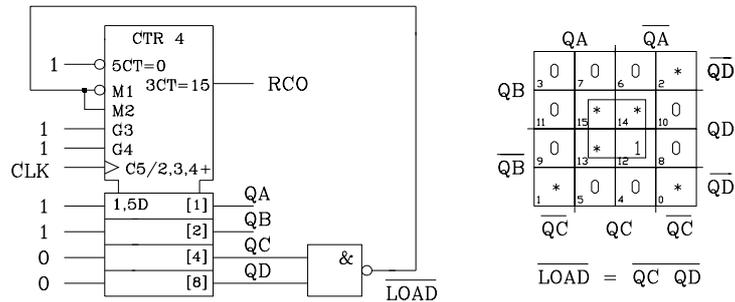


Bild Ü16.3: Gesuchte Schaltung zu b)

Aufgabe 17: Synchroner Modulo-5-Zähler

Geben Sie zu den drei Unterpunkten jeweils die nichtnegierten disjunktiven Minimalformen an.

- 17.1 Entwerfen Sie mit Hilfe von D-Flipflops einen synchronen Modulo-5 Zähler. Der Zähler soll vorwärts zählen.
- 17.2 Entwerfen Sie mit Hilfe von D-Flipflops einen synchronen Modulo-5 Zähler. Der Zähler soll rückwärts zählen.
- 17.3 Entwerfen Sie mit Hilfe von D-Flipflops einen umschaltbaren synchronen Modulo-5 Zähler. Mit $UM = 0$ soll vorwärts und mit $UM = 1$ soll rückwärts gezählt werden.

Lösung:

17.1 Vorwärtszähler

Tabelle Ü17.1: Wahrheitstabelle für den synchronen Modulo-5-Vorwärtszähler

| QC | QB | QA | Zahl | QC* | QB* | QA* |
|----|----|----|------|-----|-----|-----|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 2 | 0 | 1 | 1 |
| 0 | 1 | 1 | 3 | 1 | 0 | 0 |
| 1 | 0 | 0 | 4 | 0 | 0 | 0 |
| 1 | 0 | 1 | 5 | * | * | * |
| 1 | 1 | 0 | 6 | * | * | * |
| 1 | 1 | 1 | 7 | * | * | * |

Übergangsbedingung für das D-Flipflop: $D = Q^*$

$$DA = QA^* = (0) \vee (2) = \overline{QA} \overline{QC}$$

$$DB = QB^* = (1) \vee (2) = QA \overline{QB} \vee \overline{QA} QB$$

$$DC = QC^* = (3) = QA QB$$

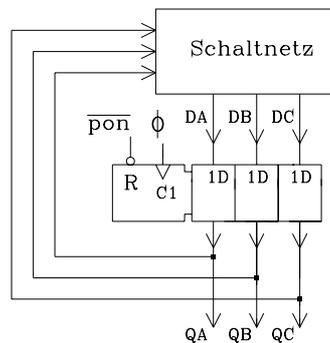


Bild Ü17.1: Schaltung des Modulo-5-Vorwärtszählers

17.2 Rückwärtszähler

Tabelle Ü17.2: Wahrheitstabelle für den synchronen Modulo-5-Rückwärtszähler

| QC | QB | QA | Zahl | QC* | QB* | QA* |
|----|----|----|------|-----|-----|-----|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 2 | 0 | 0 | 1 |
| 0 | 1 | 1 | 3 | 0 | 1 | 0 |
| 1 | 0 | 0 | 4 | 0 | 1 | 1 |
| 1 | 0 | 1 | 5 | * | * | * |
| 1 | 1 | 0 | 6 | * | * | * |
| 1 | 1 | 1 | 7 | * | * | * |

Übergangsbedingung: $D = Q^*$

$$DA = QA^* = (2) \vee (4) = QC \vee \overline{QA} QB$$

$$DB = QB^* = (3) \vee (4) = QC \vee QA QB$$

$$DC = QC^* = (0) = \overline{QA} \overline{QB} \overline{QC}$$

Es gilt die Schaltung nach Bild Ü17.1.

17.3 Umschaltbarer Vor-/Rückwärtszähler

Die Umschaltvariable UM wird in der Wahrheitstabelle eine zusätzliche Eingangsgröße berücksichtigt.

Übergangsbedingung: $D = Q^*$

$$DA = QA^* = (0) \vee (2) \vee (10) \vee (12) = QC UM \vee \overline{QA} QB \vee \overline{QA} \overline{QC} \overline{UM}$$

$$DB = QB^* = (1) \vee (2) \vee (11) \vee (12) =$$

$$= QC UM \vee QA QB UM \vee \overline{QA} QB \overline{UM} \vee QA \overline{QB} \overline{UM}$$

$$DC = QC^* = (3) \vee (8) = QA \overline{QB} \overline{UM} \vee \overline{QA} \overline{QB} \overline{QC} UM$$

Tabelle Ü17.2: Wahrheitstabelle für Aufgabe 17.3

| UM | QC | QB | QA | Zahl | QC* | QB* | QA* |
|----|----|----|----|------|-----|-----|-----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 2 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 3 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 4 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 5 | * | * | * |
| 0 | 1 | 1 | 0 | 6 | * | * | * |
| 0 | 1 | 1 | 1 | 7 | * | * | * |
| 1 | 0 | 0 | 0 | 8 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 9 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 10 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 11 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 12 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 13 | * | * | * |
| 1 | 1 | 1 | 0 | 14 | * | * | * |
| 1 | 1 | 1 | 1 | 15 | * | * | * |

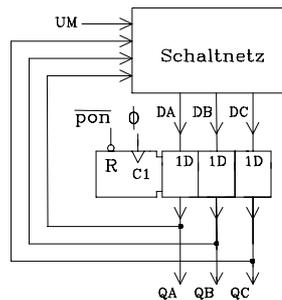


Bild Ü17.2: Schaltung des umschaltbaren Modulo-5-Vorwärts-/Rückwärtszählers

VHDL-Modell: Umschaltbarer Modulo-5-Zähler

```
library ieee;
use ieee.std_logic_1164.all;
```

```
entity mod_5_um is port (
    clk, pon, um: in std_logic;
    q: out std_logic_vector (2 downto 0));
end mod_5_um;
```

```
architecture verhalt_mod_5 of mod_5_um is -- Realisierung mit einer Zustandsmaschine
    type zustand_type is (S0, S1, S2, S3, S4);
    signal zustand: zustand_type;
```

```

begin
zustand_machine: process (clk, pon)           --Registerausgabe
begin
  if pon = '0' then q <= "000"; zustand <= S0;   -- Anfangszustand
  elsif clk'event and clk = '1' then
    case zustand is                          -- Zustandsfolge ist von "um" abhaengig
      when S0 =>
        if um='1' then zustand <= S4; q <= "100";
        elsif um='0' then zustand <= S1; q <= "001";
        end if;
      when S1 =>
        if um='1' then zustand <= S0; q <= "000";
        elsif um='0' then zustand <= S2; q <= "010";
        end if;
      when S2 =>
        if um='1' then zustand <= S1; q <= "001";
        elsif um='0' then zustand <= S3; q <= "011";
        end if;
      when S3 =>
        if um='1' then zustand <= S2; q <= "010";
        elsif um='0' then zustand <= S4; q <= "100";
        end if;
      when S4 =>
        if um='1' then zustand <= S3; q <= "011";
        elsif um='0' then zustand <= S0; q <= "100";
        end if;
      when others => null;
    end case;
  end if;
end process zustand_machine;
end verhalt_mod_5;

```

Aufgabe 18: Entwurf eines synchronen Schaltwerks (Moore-Automat)

Gegeben ist ein Takt ϕ mit der Periodendauer $T_p = 1\mu\text{s}$ und ein zum Takt asynchrones Eingangssignal X1 (Bild Ü18.1). Die Eingangsimpulse sind breiter als T_p und der Abstand zwischen den Eingangsimpulsen, gemessen von negativer Flanke bis zur nächstfolgenden positiven, ist größer als $3 T_p$.

Entwerfen Sie einen digitalen Differenzierer, der sowohl nach der positiven als auch nach der negativen Flanke eines Eingangsimpulses je einen zur positiven Taktflanke synchronen Impuls der Breite T_p ausgibt. Auch als Reaktion auf schmale Eingangsimpulse sollen an Y1 zwei Impulse in einem Abstand der Taktperiode T_p ausgegeben werden. Die synchrone Ausgabe an Y1 soll so schnell wie möglich erfolgen (Bild Ü18.1).

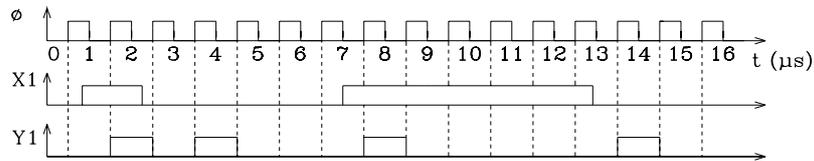


Bild Ü18.1: Signalzeitplan des digitalen Differenzierers

Anleitung:

Setzen Sie zur Lösung der Aufgabe ein synchrones Schaltwerk vom Typ Moore-Automat ein. Stellen Sie das entsprechende Zustandsdiagramm und die Zustandsfolgetabelle auf und reduzieren Sie die Zustände soweit wie möglich. Entwerfen Sie anhand der Zustandsfolgetabelle ein synchrones Schaltwerk mit D-Flipflops als Zustandsvariablenspeicher.

Lösung:

Zur Lösung der Aufgabe wird zunächst ein Zustandsdiagramm entworfen. Anhand der Aufgabenstellung wird schrittweise das Zustandsdiagramm a) in Bild Ü18.2 entwickelt. Nach dem Einschalten der Versorgungsspannung (pon) wird der Anfangszustand 0 erreicht. In diesem Zustand gibt das Schaltwerk am Ausgang Y1 den Logik-Zustand 0 aus und wartet (Warteschleife für $\neg X1 = 1$) bis das Eingangssignal X1 den Logik-Zustand 1 annimmt. Für $X1 = 1$ erfolgt mit der nächsten positiven Taktflanke der Übergang in den Zustand 1.

Im Zustand 1 wird an Y1 für eine Taktperiode der Logik-Zustand 1 ausgegeben. Mit der nächsten positiven Taktflanke erfolgt für $\neg X1 = 1$ (kurzer Eingangsimpuls) der Übergang nach Zustand 4, während für $X1 = 1$ (breiter Impuls) der Folgezustand 2 erreicht wird. In beiden Zuständen (2 und 4) wird $Y1 = 0$ ($\neg Y1 = 1$) ausgegeben.

Von Zustand 4 ausgehend wird für $X1 = 0$ ($\neg X1 = 1$) zunächst der Zustand 5 erreicht, in dem $Y1 = 1$ wird (zweiter Ausgabeimpuls) und anschließend der Anfangszustand 0. Sowohl im Zustand 4 als auch im Zustand 5 kann aufgrund der Randbedingung "Minimaler Abstand zwischen zwei Eingangsimpulsen ist größer als $3 T_p$ " das Eingangssignal X1 nicht 1 werden.

Falls ein breiter Eingangsimpuls vorliegt, wartet das Schaltwerk im Zustand 2 das Impulsende (Warteschleife für $X1 = 1$) ab und geht dann für $X1 = 0$ ($\neg X1 = 1$) mit der nächsten positiven Flanke in den Zustand 3 über. Im Zustand 3 wird für eine Taktperiode $Y1 = 1$ ausgegeben, und anschließend erfolgt der Übergang in den Anfangszustand 0. Aufgrund des minimalen Abstands der Eingangsimpulse von größer $3 T_p$ kann im Zustand 3 die Eingangsvariable X1 nicht "1" werden.

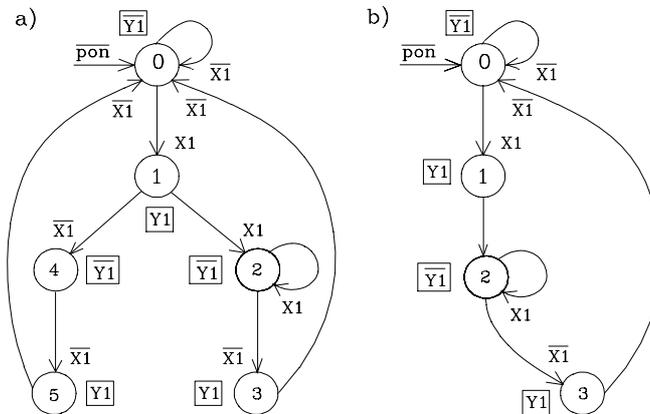


Bild Ü18.2: Zustandsdiagramm des digitalen Differenzierers

Die zu dem Zustandsdiagramm in Bild Ü18.2 a) zugehörige Zustandsfolgetabelle ist in Tabelle Ü18.1 a) dargestellt. Die Zustände 3 und 5 in Tabelle Ü18.1 a) sind äquivalent und werden zu dem Zustand 3 zusammengefaßt. Wenn man die Zustandsnummer 5 durch 3 ersetzt, erkennt man, daß die Zustände 2 und 4 ebenfalls äquivalent sind. Sie werden zu dem Zustand 2 zusammengefaßt. Die sich daraus ergebende reduzierte Zustandsfolgetabelle ist in Tabelle Ü18.1 b) abgebildet und das zugehörige Zustandsdiagramm ist in Bild Ü18.2 b) skizziert.

Tabelle Ü18.1: Gegenüberstellung der nichtreduzierten (links) und der reduzierten Zustandsfolgetabelle (rechts)

| Eingang | Zustand | | Ausgang |
|---------|---------|-----|---------|
| | m | m+1 | m |
| X | Z | Z* | Y |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 4 | 1 |
| 1 | 1 | 2 | 1 |
| 0 | 2 | 3 | 0 |
| 1 | 2 | 2 | 0 |
| 0 | 3 | 0 | 1 |
| 1 | 3 | * | 1 |
| 0 | 4 | 5 | 0 |
| 1 | 4 | * | 0 |
| 0 | 5 | 0 | 1 |
| 1 | 5 | * | 1 |

| Eingang | Zustand | | Ausgang |
|---------|---------|-----|---------|
| | m | m+1 | m |
| X | Z | Z* | Y |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 2 | 1 |
| 1 | 1 | 2 | 1 |
| 0 | 2 | 3 | 0 |
| 1 | 2 | 2 | 0 |
| 0 | 3 | 0 | 1 |
| 1 | 3 | * | 1 |

Tabelle Ü18.2: Ausführliche Form der reduzierten Zustandsfolgetabelle

| Eing. (dez.) | Zustand (dez.) | | Ausg. (dez.) | Eingangs- variablen | Zustands- variablen | | Ausgangs- variablen | |
|-----------------|----------------|-----|-----------------|------------------------|------------------------|----|------------------------|-----|
| | m | m+1 | | | m | m | | m+1 |
| X | Z | Z* | Y | X1 | Z1 | Z2 | Z1* Z2* | Y1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 2 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 2 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 2 | 3 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 2 | 2 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 3 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 3 | * | 1 | 1 | 1 | 1 | * | * |

Anhand der ausführlichen Zustandsfolgetabelle in Tabelle Ü18.2 werden mit Hilfe der KV-Diagramme (Bild Ü18.3) die Gleichungen D1 und D2 für die D-Flipflops sowie die Ausgangsgleichung Y1 bestimmt.

Für die KV-Diagramme gilt folgende Zuordnung: 2^2 2^1 2^0
X1 Z1 Z2

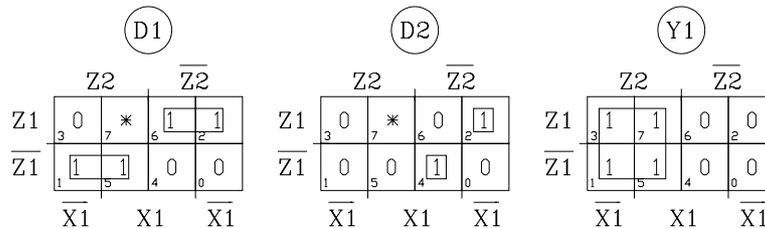


Bild Ü18.3: KV-Diagramme für D1, D2 und Y1

$$Z1^* = D1 = (1) \vee (5) \vee (2) \vee (6) = Z1 \overline{Z2} \vee \overline{Z1} Z2$$

$$Z2^* = D2 = (4) \vee (2) = X1 \overline{Z1} \overline{Z2} \vee \overline{X1} Z1 \overline{Z2}$$

$$Y1 = (1) \vee (5) \vee (3) \vee (7) = Z2$$

Das gesuchte Schaltwerk ist in Bild Ü18.4 abgebildet. Für die technische Realisierung kann ein PAL mit Registerausgang eingesetzt werden. In diesem Fall wird zur Lösung der Aufgabe nur ein integrierter Baustein benötigt.

Das Schaltwerk in Bild Ü18.4 ist ein Moore-Automat mit synchroner Ausgabe $Y1 = Z2$. Die im Zustandsdiagramm (Bild Ü18.2) gekennzeichnete Möglichkeit über einen Einschaltimpuls ($\neg\text{pon}$) den Anfangszustand 0 zu erreichen, wird im Schaltwerk über den negierten Rücksetzeingang am D-Register mit $\neg\text{pon} = 0$ realisiert.

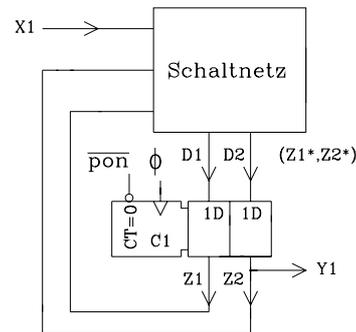


Bild Ü18.4: Schaltung des digitalen Differenzierers

VHDL-Modell: Digitaler Differenzierer mit zwei Ausgabeimpulsen

```
library ieee;
use ieee.std_logic_1164.all;
```

```
entity differ_2 is port (
  clk, x1, pon:   in std_logic;
  y1:            out std_logic);
end differ_2;
```

```
architecture differ_2_arch of differ_2 is
```

```
  type zustand_type is (S0, S1, S2, S3);
  signal zustand: zustand_type;
```

```
begin
```

```
  zustand_machine: process (clk, pon)
```

```
  begin
```

```
    if pon='0' then zustand <= S0;      -- mit pon = 0 in den Anfangszustand
```

```
    elsif clk'event and clk = '1' then
```

```
      case zustand is
```

```
        when S0 =>                                -- Zustand 0 (S0)
```

```
          if x1='1' then zustand <= S1;
```

```
          elsif x1='0' then zustand <= S0;
```

```
          end if;
```

```
        when S1 =>                                -- Zustand 1 (S1)
```

```
          zustand <= S2;
```

```
        when S2 =>                                -- Zustand 2 (S2)
```

```
          if x1='1' then zustand <= S2;
```

```
          elsif x1='0' then zustand <= S3;
```

```
          end if;
```

```
        when S3 =>                                -- Zustand 3 (S3)
```

```
          if x1='0' then zustand <= S0;
```

```
          end if;
```

```
      end case;
```

```
    end if;
```

```
  end process;
```

```

y1_zuweisung:                -- Signal-Zuweisung fuer kombinatorische Ausgabe
y1 <= '1' when (zustand = S1 or zustand = S3)
    else '0';
end differ_2_arch;

```

Aufgabe 19: Entwurf eines synchronen Schaltwerks (Mealy-Automat)

Gegeben ist ein Takt ϕ mit der Periodendauer $T_p = 1\mu\text{s}$ und ein zum Takt asynchrones Eingangssignal $X1$ (Bild Ü19.1). Die Eingangsimpulse sind breiter als T_p und der Abstand zwischen den Eingangsimpulsen, gemessen von negativer Flanke bis zur nächstfolgenden positiven, ist größer als $3 T_p$.

Entwerfen Sie einen digitalen Differenzierer, der unmittelbar nach der positiven Flanke eines Eingangsimpulses einen H-Impuls am Ausgang ausgibt. Verzögerungszeiten der im Schaltwerk eingesetzten Gatter und Flipflops seien vernachlässigbar. Der Ausgabeimpuls $Y1$ soll breiter als $2 T_p$, jedoch schmaler als $3 T_p$ sein und synchron mit der positiven Taktflanke beendet werden (Bild Ü19.1).

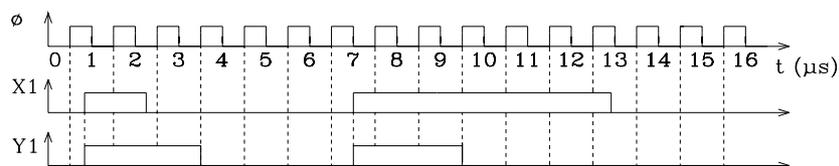


Bild Ü19.1: Signalzeitplan des digitalen Differenzierers

Anleitung:

Setzen Sie zur Lösung der Aufgabe ein synchrones Schaltwerk vom Typ Mealy-Automat ein. Stellen Sie das entsprechende Zustandsdiagramm und die Zustandsfolgetabelle auf und reduzieren Sie die Zustände soweit wie möglich. Entwerfen Sie anhand der Zustandsfolgetabelle ein synchrones Schaltwerk mit D-Flipflops als Zustandsvariablenpeicher.

Lösung:

Zur Lösung der Aufgabe wird zunächst ein Zustandsdiagramm entworfen. Anhand der Aufgabenstellung wird schrittweise das Zustandsdiagramm a) in Bild Ü19.2 entwickelt. Nach dem Einschalten der Versorgungsspannung (pon) wird der Anfangszustand 0 erreicht. In diesem Zustand gibt das Schaltwerk am Ausgang $Y1$ für $X1 = 0$ den Logik-Zustand 0 und für $X1 = 1$ den Logik-Zustand 1 aus. Solange $X1 = 0$ ist, bleibt das Schaltwerk in einer Warteschleife. Für $X1 = 1$ erfolgt mit der nächsten positiven Taktflanke der Übergang in den Zustand 1.

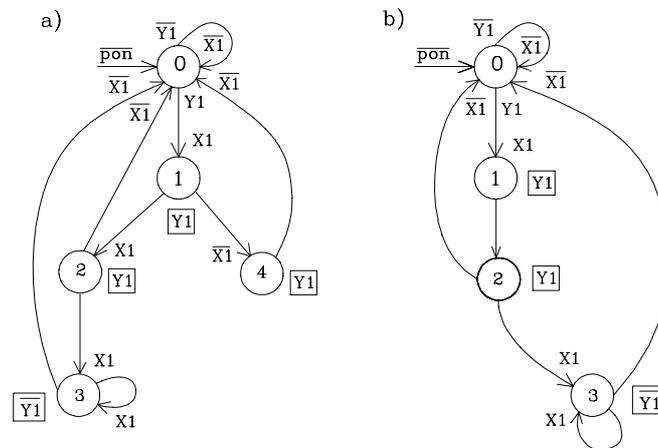


Bild Ü19.2: Zustandsdiagramm des digitalen Differenzierers

Im Zustand 1 wird an $Y1$ für eine Taktperiode der Logik-Zustand 1 ausgegeben. Mit der nächsten positiven Taktflanke erfolgt für $\bar{X1} = 1$ (kurzer Eingangsimpuls) der Übergang nach Zustand 4, während für $X1 = 1$ (breiter Impuls) der Folgezustand 2 erreicht wird. In beiden Zuständen (2 und 4) wird $Y1 = 1$ ausgegeben. Da im Zustand 0 für $X1 = 1$ und in zwei aufeinander folgenden Zuständen (1 und 2 bzw. 1 und 4) am Ausgang $Y1 = 1$ ausgegeben wird, sind die Ausgangsimpulse breiter als $2 T_p$ und schmäler als $3 T_p$.

Vom Zustand 4 ausgehend wird für $X1 = 0$ ($\bar{X1} = 1$) der Anfangszustand 0 erreicht. Im Zustand 4 kann aufgrund der Randbedingung „Minimaler Abstand zwischen zwei Eingangsimpulsen ist größer als $3 T_p$ “ das Eingangssignal $X1$ nicht 1 werden. Im Zustand 2 verzweigt sich der Signalfluß. Für $X1 = 0$ ($\bar{X1} = 1$) wird der Anfangszustand und für $X1 = 1$ der Zustand 3 erreicht.

Im Zustand 3 wartet das Schaltwerk das Impulsende (Warteschleife für $X1 = 1$) ab und geht dann für $X1 = 0$ ($\bar{X1} = 1$) mit der nächsten positiven Flanke in den Zustand 0 über. Im Zustand 3 wird $Y1 = 0$ ($\bar{Y1} = 1$) ausgegeben.

Die zu dem Zustandsdiagramm in Bild Ü19.2 a) zugehörige Zustandsfolgetabelle ist in Tabelle Ü19.1 a) dargestellt.

Die Zustände 2 und 4 in Tabelle Ü19.1 a) sind äquivalent und werden zu dem Zustand 2 zusammengefaßt. Die sich daraus ergebende reduzierte Zustandsfolgetabelle ist in Tabelle Ü19.1 b) abgebildet und das zugehörige Zustandsdiagramm ist in Bild Ü19.2 b) skizziert.

Tabelle Ü19.1: Gegenüberstellung der nichtreduzierten und der reduzierten Zustandsfolgetabelle

a) Zustandsfolgetabelle

| Eingang | Zustand | | Ausgang |
|---------|---------|-----|---------|
| | m | m+1 | m |
| X | Z | Z* | Y |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 4 | 1 |
| 1 | 1 | 2 | 1 |
| 0 | 2 | 0 | 1 |
| 1 | 2 | 3 | 1 |
| 0 | 3 | 0 | 0 |
| 1 | 3 | 3 | 0 |
| 0 | 4 | 0 | 1 |
| 1 | 4 | * | 1 |

b) Reduzierte Zustandsfolgetabelle

| Eingang | Zustand | | Ausgang |
|---------|---------|-----|---------|
| | m | m+1 | m |
| X | Z | Z* | Y |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 2 | 1 |
| 1 | 1 | 2 | 1 |
| 0 | 2 | 0 | 1 |
| 1 | 2 | 3 | 1 |
| 0 | 3 | 0 | 0 |
| 1 | 3 | 3 | 0 |

Tabelle Ü19.2: Ausführliche Form der reduzierten Zustandsfolgetabelle

| Eing. (dez.) | Zustand (dez.) | | Ausg. (dez.) | Eigangs- variablen | Zustands- variablen | | Ausgangs- variablen |
|-----------------|-------------------|-----|-----------------|-----------------------|------------------------|---------|------------------------|
| | m | m+1 | | | m | m+1 | |
| X | Z | Z* | Y | X1 | Z1 Z2 | Z1* Z2* | Y1 |
| 0 | 0 | 0 | 0 | 0 | 0 0 | 0 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 0 | 0 1 | 1 |
| 0 | 1 | 2 | 1 | 0 | 0 1 | 1 0 | 1 |
| 1 | 1 | 2 | 1 | 1 | 0 1 | 1 0 | 1 |
| 0 | 2 | 0 | 1 | 0 | 1 0 | 0 0 | 1 |
| 1 | 2 | 3 | 1 | 1 | 1 0 | 1 1 | 1 |
| 0 | 3 | 0 | 0 | 0 | 1 1 | 0 0 | 0 |
| 1 | 3 | 3 | 0 | 1 | 1 1 | 1 1 | 0 |

Anhand der ausführlichen Zustandsfolgetabelle in Tabelle Ü19.2 werden mit Hilfe der KV-Diagramme (Bild Ü19.3) die Gleichungen D1 und D2 für die D-Flipflops sowie die Ausgangsgleichung Y1 bestimmt.

Für die KV-Diagramme gilt folgende Zuordnung: 2^2 2^1 2^0
 $X1$ $Z1$ $Z2$

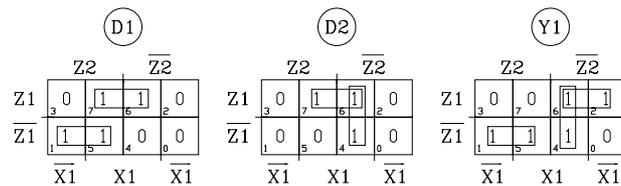


Bild Ü19.3: KV-Diagramme für D1, D2 und Y1

$$Z1^* = D1 = (1) \vee (5) \vee (6) \vee (7) = \overline{Z1} Z2 \vee X1 Z1$$

$$Z2^* = D2 = (4) \vee (6) \vee (7) = X1 \overline{Z2} \vee X1 Z1$$

$$Y1 = (4) \vee (1) \vee (5) \vee (2) \vee (6) = Z1 \overline{Z2} \vee \overline{Z1} Z2 \vee X1 \overline{Z2}$$

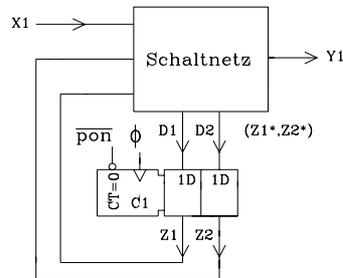


Bild Ü19.4: Schaltung des digitalen Differenzierers

Das gesuchte Schaltwerk ist in Bild Ü19.4 abgebildet. Für die technische Realisierung kann ein PAL mit zusätzlichen Registerausgängen eingesetzt werden. In diesem Fall wird zur Lösung der Aufgabe nur ein integrierter Baustein benötigt.

Das Schaltwerk in Bild Ü19.4 ist ein Mealy-Automat mit asynchroner Ausgabe am Ausgang Y1. Die im Zustandsdiagramm (Bild Ü19.2) markierte Möglichkeit über einen Einschaltimpuls ($\neg\text{pon}$) den Anfangszustand 0 zu erreichen, wird im Schaltwerk über den negierten Rücksetzeingang am D-Register mit $\neg\text{pon} = 0$ realisiert.

VHDL-Modell: Digitaler Differenzierer als Mealy-Automat

```

library ieee;
use ieee.std_logic_1164.all;

entity diff_mealy is port (
  clk, pon, x1:    in std_logic;
  y1:             out std_logic);
end diff_mealy;

architecture diff_mealy_arch of diff_mealy is
  type zustand_type is (S0, S1, S2, S3);
  signal zustand: zustand_type;
begin
  zustand_machine: process (clk, pon)
  begin
    if pon='0' then zustand <= S0;      -- pon = 0 --> Anfangszustand
    elsif clk'event and clk = '1' then
      case zustand is
        when S0 =>
          if x1='1' then zustand <= S1;
          elsif x1='0' then zustand <= S0;
          end if;

```

```

        when S1 =>
            zustand <= S2;
        when S2 =>
            if x1='0' then zustand <= S0;
            elsif x1='1' then zustand <= S3;
            end if;
        when S3 =>
            if x1='1' then zustand <= S3;
            elsif x1='0' then zustand <= S0;
            end if;
    end case;
end if;
end process;

y1_assignment:          -- Kombinatorische Ausgabe fuer Mealy-Automat
y1 <= '1' when (zustand = S0 and x1='1') else
    '0' when (zustand = S0 and x1='0') else
    '1' when (zustand = S1) else
    '1' when (zustand = S2)
    else '0';

end diff_mealy_arch;

```

20. Entwurf eines synchronen Schaltwerks mit Registerausgabe

Für die Steuerung der Datenübergabe von einem Rechner an zwei Meßgeräte soll ein synchrones Schaltwerk entworfen werden. Es werden nacheinander Datenwörter gesendet, die von den Meßgeräten empfangen und quittiert werden. Ein neues Datenwort wird mit der positiven Flanke am Ausgang Y1 des Schaltwerks gesendet und bleibt bis zur negativen Flanke von Y1 gültig. Der Ausgang Y1 soll solange im 1-Zustand bleiben, bis beide Meßgeräte das Datenwort empfangen und durch einen 1-Impuls quittiert haben. Nach einer kurzen Pause wird anschließend das nächste Datenwort mit einem neuen 1-Impuls am Ausgang Y1 übergeben (s. Bild Ü20.1).

Es gilt:

- Die Übernahme der Daten am Meßgerät ist mit der negativen Flanke des Quittungssignals X1 bzw. X2 abgeschlossen.
- Impulsbreite von X1 bzw. X2 ist größer als 1,2 μs und kleiner als 10 μs .
- Impulspause am Ausgang Y1 ist größer als 0,5 μs .

Zur Lösung der Aufgabenstellung soll ein synchroner Moore-Automat mit Registerausgabe eingesetzt werden. Bestimmen Sie die erforderliche Taktfrequenz und begründen Sie Ihre Wahl. Geben Sie das Zustandsdiagramm und die Zustandsfolgetafel in ausführlicher Form an. Reduzieren Sie die Anzahl der Zustände, falls es möglich ist. Die digitale Schaltung ist *nicht* erforderlich.

Geben Sie ein geeignetes VHDL-Modell an, das eine Registerausgabe ermöglicht.

Hinweis: Es ist sichergestellt, daß immer beide Meßgeräte Quittungssignale senden.

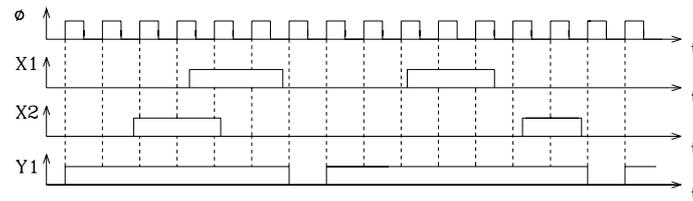


Bild Ü20.1: Beispiel für den Signalverlauf an den Eingängen und am Ausgang des Schaltwerks

Lösung:

Es soll gelten: $0,5\mu\text{s} \leq T_\phi \leq 1,2\mu\text{s}$, damit ein Impuls einerseits sicher erfasst wird und andererseits die Impulspause an Y groß genug ist. Gewählt wird $T_\phi = 1\mu\text{s}$.

Tabelle Ü20.1: Zustandsfolgetabelle für einen Moore-Automaten mit Registerausgabe

| X1 | X2 | Z1 | Z2 | Z3 | Z1* | Z2* | Z3* | Y1* |
|----|----|----|----|----|-----|-----|-----|-----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | * | * | * | 1 |
| 1 | 0 | 0 | 0 | 0 | * | * | * | 1 |
| 1 | 1 | 0 | 0 | 0 | * | * | * | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

VHDL-Modell: Moore-Automat mit Registerausgabe

```

library ieee;
use ieee.std_logic_1164.all;

entity daten_ueber is port (
    pon,clk: in std_logic;
    x:       in std_logic_vector (1 to 2);
    y1:     out std_logic);
end daten_ueber;

```

```

architecture arch_ueber of daten_ueber is
    type states is (S0,S1,S2,S3,S4); -- Typdeklaration
    signal zustand: states;
begin

reg_aus: process(clk, pon)
    begin
        if pon='1'then                                -- pon = 1 ---> Anfangszustand
            zustand <= S0;
            y1 <= '0';
        elsif (clk'event and clk = '1') then
            y1 <= '1';                                -- y1 = 1 ist die Voreinstellung
            case zustand is                            -- wenn IST-Zustand gleich ...
                when S0 =>
                    zustand <= S1;
                when S1 =>
                    case x is                            -- Folgezustand ist abhaengig von x
                        when "00" => zustand <= S1;
                        when "01" => zustand <= S4;
                        when "10" => zustand <= S2;
                        when others => zustand <= S3;
                    end case;
                when S2 =>
                    if (x = "00" or x = "10") then -- X=0 oder X=2
                        zustand <= S2;
                    else                                -- X=1 oder X=3
                        zustand <= S3;
                    end if;
                when S3 =>
                    if x="00" then
                        zustand <= S0;
                        y1 <= '0';                    -- Y1=0 gilt fuer den Zustand S0
                    else
                        zustand <= S3;
                    end if;
                when S4 =>
                    if (x = "00" or x = "01") then
                        zustand <= S4;
                    else
                        zustand <= S3;
                    end if;
                end case;
            end if;
        end process reg_aus;

end arch_ueber;

```

Aufgabe 21: Entwurf eines synchronen Schaltwerks mit Festwertspeicher

In einem Stahlwerk sind drei leistungsstarke Elektrolichtbogenöfen installiert. Mit dem zuständigen Energieversorgungsunternehmen ist vertraglich eine maximale Leistungsaufnahme P_m festgelegt worden. Aus dem Grunde darf die maximale Leistung P_m auf keinen Fall überschritten werden.

Leistungsaufnahme der Öfen in Prozent der Maximalleistung P_m :

Ofen A 65%; Ofen B 50%; Ofen C 25%

Die Schaltwarten der einzelnen Öfen sind örtlich getrennt und jeder Ofenwart kann von seiner Schaltwarte die Anforderung OFEN-EIN bzw. OFEN-AUS stellen.

Mit Hilfe einer automatischen Steuerung soll gewährleistet werden, daß folgende Bedingungen eingehalten werden:

- Ein Ofen wird eingeschaltet, wenn die Anforderung OFEN-EIN besteht und noch kein Ofen eingeschaltet war.
- Ein zweiter Ofen wird zugeschaltet, wenn die Anforderung besteht und die Summe der beiden Ofenleistungen kleiner als die maximale Leistung P_m ist.
- Zwei Öfen dürfen nicht gleichzeitig eingeschaltet werden. Zuerst wird der leistungsstärkere Ofen eingeschaltet werden; Bedingung b) muß dabei beachtet werden.
- Soll Ofen A (B) ein- und Ofen B (A) ausgeschaltet werden, so wird zunächst Ofen B (A) ausgeschaltet und im nächsten Zyklus Ofen A (B) eingeschaltet.
- Falls es möglich ist, sollen 2 Öfen in Betrieb sein.
- Ein Ofen darf erst abgeschaltet werden, wenn die Anforderung OFEN-AUS von der Schaltwarte erfolgt.

Entwerfen Sie die Zustandsfolgetabelle und das entsprechende synchrone Schaltwerk.

Lösung:

Die Anforderungen der Schaltwarten sind Eingangsgrößen für das Schaltwerk. Es wird folgende Zuordnung festgelegt:

Eingänge:

| | |
|---------------------------------|-------------------------------|
| Ofen A einschalten: $X_A = 1$; | Ofen A ausschalten: $X_A = 0$ |
| Ofen B einschalten: $X_B = 1$; | Ofen B ausschalten: $X_B = 0$ |
| Ofen C einschalten: $X_C = 1$; | Ofen C ausschalten: $X_C = 0$ |

Ausgänge:

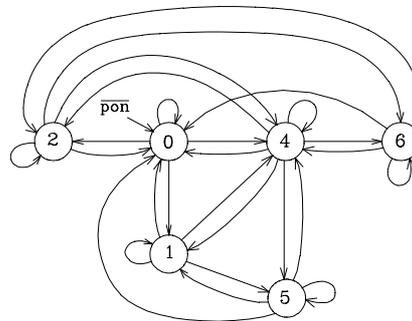
| | |
|-----------------------------------|----------------------------------|
| $Y_A = 1$: Ofen A eingeschaltet; | $Y_A = 0$: Ofen A ausgeschaltet |
| $Y_B = 1$: Ofen B eingeschaltet; | $Y_B = 0$: Ofen B ausgeschaltet |
| $Y_C = 1$: Ofen C eingeschaltet; | $Y_C = 0$: Ofen C ausgeschaltet |

Aufgrund der Randbedingungen c) und d) wird für die Musterlösung der Moore-Automat gewählt. Mit Hilfe der angegebenen Zuordnung (s. Tabelle Ü21.1) läßt sich eine synchrone Ausgabe über die Zustandsvariablen erreichen.

Tabelle Ü21.1: Die gewählte Zuordnung ermöglicht eine Ausgabe über Zustandsvariablen

| Z | 2^2 | 2^1 | 2^0 | Eingeschaltete Öfen |
|---|-------|-------|-------|---------------------|
| Y | Z1 | Z2 | Z3 | |
| Y | YC | YB | YA | |
| 0 | 0 | 0 | 0 | --- |
| 1 | 0 | 0 | 1 | Ofen A |
| 2 | 0 | 1 | 0 | Ofen B |
| 4 | 1 | 0 | 0 | Ofen C |
| 5 | 1 | 0 | 1 | Ofen A und Ofen C |
| 6 | 1 | 1 | 0 | Ofen B und Ofen C |

Im Zustandsdiagramm (Bild Ü21.1) sind die Übergänge zwischen den einzelnen Zuständen dargestellt. Damit das Zustandsdiagramm noch übersichtlich bleibt, sind die Übergangsbedingungen und die Ausgabegleichungen nicht innerhalb des Zustandsdiagramms angegeben, sondern in einer Vereinbarung aufgelistet.

**Bild Ü21.1:** Zustandsdiagramm (Moore-Automat)

Vereinbarung: $X(m,m+1)$ $m =$ derzeitiger Zustand, $m+1 =$ Folgezustand
 $Y(m)$ $m =$ derzeitiger Zustand (Moore-Automat)

a) Zustand 0 (Anfangszustand)

$$\begin{aligned} X(0,0) &= \overline{XC} \overline{XB} \overline{XA} & X(0,2) &= \overline{XC} \overline{XB} \overline{XA} \vee XC \overline{XB} \overline{XA} \\ X(0,1) &= \overline{XC} \overline{XB} \overline{XA} \vee \overline{XC} \overline{XB} \overline{XA} \vee XC \overline{XB} \overline{XA} \vee XC \overline{XB} \overline{XA} \\ X(0,4) &= XC \overline{XB} \overline{XA} & Y(0) &= \overline{YC} \overline{YB} \overline{YA} \end{aligned}$$

b) Zustand 1

$$\begin{aligned} X(1,0) &= \overline{XC} \overline{XB} \overline{XA} \vee \overline{XC} \overline{XB} \overline{XA} \vee XC \overline{XB} \overline{XA} & X(1,4) &= XC \overline{XB} \overline{XA} \\ X(1,1) &= \overline{XC} \overline{XB} \overline{XA} \vee \overline{XC} \overline{XB} \overline{XA} & X(1,5) &= XC \overline{XB} \overline{XA} \vee XC \overline{XB} \overline{XA} \\ Y(1) &= \overline{YC} \overline{YB} \overline{YA} \end{aligned}$$

c) Zustand 2

$$\begin{aligned} X(2,0) &= \overline{XC} \overline{XB} \overline{XA} \vee \overline{XC} \overline{XB} \overline{XA} \vee XC \overline{XB} \overline{XA} & X(2,4) &= XC \overline{XB} \overline{XA} \\ X(2,2) &= \overline{XC} \overline{XB} \overline{XA} \vee \overline{XC} \overline{XB} \overline{XA} & X(2,6) &= XC \overline{XB} \overline{XA} \vee XC \overline{XB} \overline{XA} \\ Y(2) &= \overline{YC} \overline{YB} \overline{YA} \end{aligned}$$

d) Zustand 4

$$X(4,0) = \overline{XC} \overline{XB} \overline{XA}$$

$$X(4,1) = \overline{XC} \overline{XB} \overline{XA} \vee \overline{XC} XB XA$$

$$X(4,2) = \overline{XC} \overline{XB} \overline{XA}$$

$$Y(4) = YC \overline{YB} \overline{YA}$$

$$X(4,4) = XC \overline{XB} \overline{XA}$$

$$X(4,5) = XC \overline{XB} \overline{XA} \vee XC XB XA$$

$$X(4,6) = XC \overline{XB} \overline{XA}$$

e) Zustand 5

$$X(5,0) = \overline{XC} \overline{XB} \overline{XA} \vee \overline{XC} XB \overline{XA}$$

$$X(5,1) = \overline{XC} \overline{XB} \overline{XA} \vee \overline{XC} XB XA$$

$$Y(5) = YC \overline{YB} YA$$

$$X(5,4) = XC \overline{XB} \overline{XA} \vee XC XB \overline{XA}$$

$$X(5,5) = XC \overline{XB} \overline{XA} \vee XC XB XA$$

f) Zustand 6

$$X(6,0) = \overline{XC} \overline{XB} \overline{XA} \vee \overline{XC} \overline{XB} XA$$

$$X(6,2) = \overline{XC} \overline{XB} \overline{XA} \vee \overline{XC} XB XA$$

$$Y(6) = YC YB \overline{YA}$$

$$X(6,4) = XC \overline{XB} \overline{XA} \vee XC \overline{XB} XA$$

$$X(6,6) = XC \overline{XB} \overline{XA} \vee XC XB XA$$

Tabelle Ü21.2: Zustandsfolgetabelle für einen Moore-Automaten

| Eingangsvariablen | | | Zustandsvariablen | | | | | | Ausgangsvariablen | | |
|-------------------|----|----|-------------------|----|----|-----|-----|-----|-------------------|----|----|
| m | | | m | | | m+1 | | | m | | |
| XC | XB | XA | Z1 | Z2 | Z3 | Z1* | Z2* | Z3* | YC | YB | YA |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |

| Eingangsvariablen | | | Zustandsvariablen | | | | | | Ausgangsvariablen | | |
|-------------------|---|---|-------------------|---|---|-----|---|---|-------------------|---|---|
| m | | | m | | | m+1 | | | m | | |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |

Für die Schaltungsrealisierung werden ein Festspeicher und ein D-Register eingesetzt. Das D-Register dient zur Speicherung der Zustandsvariablen und zur Aufsynchonisierung der Eingangsvariablen. Für den Takt Φ wird eine Periodendauer von 100 ms gewählt.

Aus Gründen der Übersichtlichkeit ist auf eine besondere Unterscheidung im Signalnamen zwischen den Eingangsvariablen und den synchronisierten Eingangsvariablen verzichtet worden. In der zugehörigen Programmiertabelle (Tabelle Ü21.3) werden die Adressen durch die Zustandsvariablen Z (Z^m) und die synchronisierten Eingangsvariablen gebildet. Jede Speicheradresse enthält die zugehörigen Logik-Zustände der Zustandsvariablen Z^* (Z^{m+1}) und der Ausgangsvariablen. Die Speicherung der Ausgangsvariablen ist nicht unbedingt erforderlich; sie sind hier aus Übungszwecken aufgenommen worden, um synchrone und asynchrone Ausgabe beim Moore-Automaten gegenüberzustellen.

Tabelle Ü21.3: Programmiertabelle für den Festwertspeicher

| XC A5 | XB A4 | XA A3 | Z1 A2 | Z2 A1 | Z3 A0 | - D7 | - D6 | Z1* D5 | Z2* D4 | Z3* D3 | YC D2 | YB D1 | YA D0 |
|----------|----------|----------|----------|----------|----------|---------|---------|-----------|-----------|-----------|----------|----------|----------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |

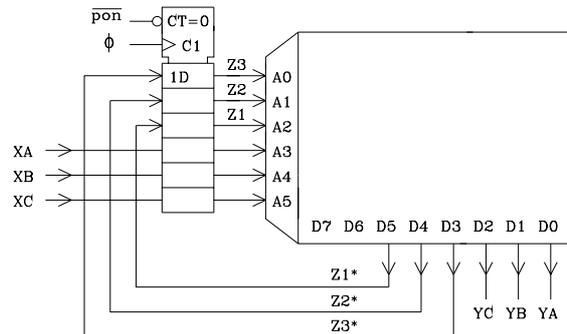


Bild Ü21.2: Hardware des gesuchten Schaltwerks

Anmerkung:

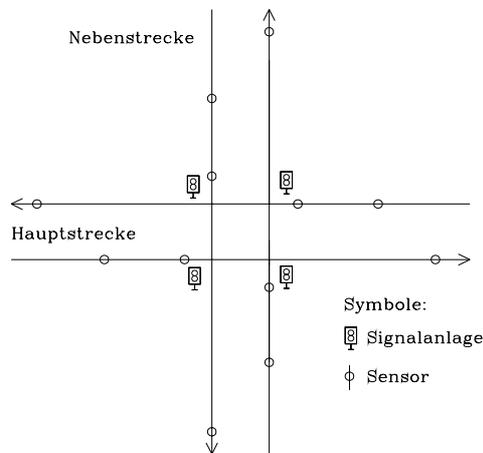
Das Schaltwerk in Bild Ü21.2 ist ein Moore-Automat mit synchroner Ausgabe an den Registerausgängen über die Zustandsvariablen Z1, Z2 und Z3. Alternativ können die Öfen auch asynchron über die Ausgänge YC, YB und YA ein- und ausgeschaltet werden. Die im Zustandsdiagramm gekennzeichnete Möglichkeit über einen Einschaltimpuls ($\neg\text{pon}$) den Anfangszustand 0 zu erreichen, wird im Schaltwerk über den negierten Rücksetzeingang am D-Register mit $\neg\text{pon} = 0$ realisiert. Die am eingesetzten Festwertspeicher vorhandenen Steueranschlüsse $\neg\text{CS}$, $\neg\text{OE}$ sowie alle weiteren Adreßanschlüsse werden an "0" angeschlossen.

Aufgabe 22: Synchrones Schaltwerk mit Multiplexer und Festwertspeicher

Für eine Eisenbahnkreuzung, bestehend aus einer Haupt- und einer Nebenstrecke, ist eine Lichtsignalanlage in Abhängigkeit von den auf vier Gleisen fahrenden Zügen (Bild Ü22.1) zu steuern. Jede Strecke besteht aus zwei Gleisen, so daß Zugverkehr in beiden Richtungen möglich ist.

Auf jedem Gleis befinden sich in unmittelbarer Nähe der Signalanlage ein Sensor und in genügend großem Abstand von der Signalanlage ein zweiter Sensor (Bild Ü22.1). Die Signale der beiden Sensoren, die in Fahrtrichtung vor der Kreuzung sind, werden verodert und als ein Signal zum Schaltwerk geführt. Ein dritter Sensor befindet sich in so großem Abstand hinter der Kreuzung, daß alle Züge, die von diesem Sensor erfaßt werden, außerhalb des Kreuzungsbereichs sind. Der Abstand zweier Züge auf einem Gleis ist so groß, daß sie nie gleichzeitig im Kreuzungsbereich sind. Der Sensor erfaßt fahrende und stehende Züge und gibt an seinem Ausgang H-Pegel (1-Zustand) aus, solange der Zug im Sensorbereich ist.

An allen Gleisen befinden sich Signalanlagen mit einer grünen und einer roten Lampe. Entwerfen Sie ein synchrones Schaltwerk zur Signalsteuerung der Eisenbahnkreuzung (Bild Ü22.1)



Randbedingung a: Befindet sich kein Zug in der Kreuzung, sollen die Signale für die Hauptstrecke grün und für die Nebenstrecke rot sein. Falls auf der Haupt- und Nebenstrecke gleichzeitig Züge in die freie Kreuzung einfahren, so soll der Zug (sollen die Züge) auf der Hauptstrecke Vorfahrt haben.

Randbedingung b: Ist ein Gleis der Hauptstrecke (Nebenstrecke) im Kreuzungsbereich belegt, so müssen Züge auf der Nebenstrecke (Hauptstrecke) warten. Ein Zug in Gegenrichtung auf dem anderen Gleis der Strecke soll freie Fahrt haben.

Bild Ü22.1: Eisenbahnkreuzung mit Sensoren und Lichtsignalanlage

Lösung:

Im ersten Schritt werden die Sensoren vor und hinter der Kreuzung gekennzeichnet (Bild Ü22.2). Danach wird mit Hilfe eines Streckenbelegungsplanes (Tabelle Ü22.1) eine Zuordnung zwischen Streckenbelegung und Zustand vorgenommen. Hierbei wird unterschieden, ob die Haupt- oder Nebenstrecke nur in einer oder in beiden Richtungen befahren wird.

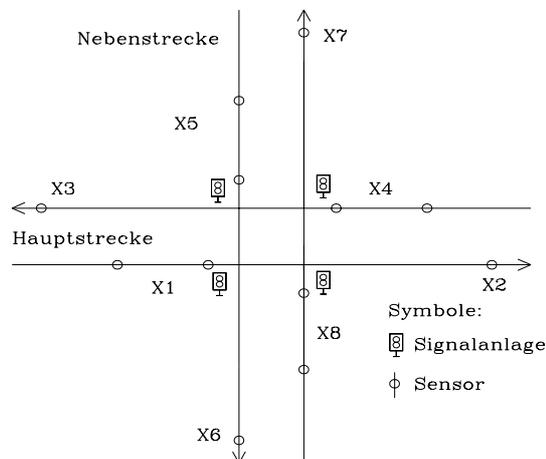


Bild Ü22.2: Kennzeichnung der Eingangsgrößen

Tabelle Ü22.1: Streckenbelegungsplan

| Streckenbelegung | Richtung des Zuges | Zustand | Lichtsignal -Ausgabe | | | | | |
|--------------------------|--------------------|---------|----------------------|----|----|--------------|----|----|
| | | | Nebenstrecke | | | Hauptstrecke | | |
| | | | rot | NR | NG | grün | HR | HG |
| Kreuzung frei | --- | 0 | rot | 1 | 0 | grün | 0 | 1 |
| Nebenstrecke (Ein Zug) | X8 → X7 | 1 | grün | 0 | 1 | rot | 1 | 0 |
| Nebenstrecke (Ein Zug) | X5 → X6 | 2 | grün | 0 | 1 | rot | 1 | 0 |
| Nebenstrecke (Zwei Züge) | X8 → X7 X5 → X6 | 3 | grün | 0 | 1 | rot | 1 | 0 |
| Hauptstrecke (Ein Zug) | X4 → X3 | 4 | rot | 1 | 0 | grün | 0 | 1 |
| Hauptstrecke (Ein Zug) | X1 → X2 | 5 | rot | 1 | 0 | grün | 0 | 1 |
| Hauptstrecke (Zwei Züge) | X4 → X3 X1 → X2 | 6 | rot | 1 | 0 | grün | 0 | 1 |

Zur Lösung der Aufgabenstellung könnte sowohl der Mealy- als auch der Moore-Automat zum Einsatz kommen. Im folgenden wird die Lösung mit dem Moore-Automat vorgestellt. Damit das Zustandsdiagramm (Bild Ü22.3) noch übersichtlich bleibt, sind die Übergänge und Ausgänge nicht im Zustandsdiagramm gekennzeichnet, sondern in einer zusätzlichen Vereinbarung angegeben. Parallel zu dem Zustandsdiagramm wird die Zustandsfolgetabelle (Tabelle Ü22.2) entwickelt. Alle nicht relevanten Eingangsvariablen werden durch das Symbol "-" gekennzeichnet. Es werden maximal vier der acht Eingangsvariablen in einem Zustand benötigt. In einer Multiplexer-Zuordnungstabelle (Tabelle Ü22.3) werden die Eingangsvariablen in Abhängigkeit von den Zustandsvariablen den maskierten Eingangsvariablen F1 bis F4 zugeordnet. Die Zustandsvariablen übernehmen hier die Rolle der Maskierungsvariablen. Da nur im Zustand 0 vier Eingangsvariablen relevant sind, kann $F1 = X1$ und $F2 = X4$ gesetzt werden. Dadurch werden zwei Multiplexer eingespart. Bild Ü22.5 zeigt die entsprechende Multiplexerschaltung.

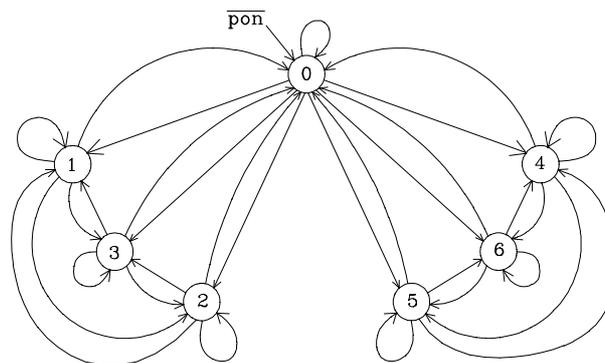


Bild Ü22.3: Zustandsdiagramm (Moore-Automat)

Vereinbarung der Kennzeichnung von Übergangs- und Ausgabebedingungen:

$X(m,m+1)$ m = derzeitiger Zustand, $m+1$ = Folgezustand

$Y(m)$ m = derzeitiger Zustand (Moore-Automat)

a) Zustand 0 (Anfangszustand)

$$X(0,0) = \overline{X1} \overline{X4} \overline{X5} \overline{X8}$$

$$X(0,1) = \overline{X1} \overline{X4} \overline{X5} X8$$

$$X(0,2) = \overline{X1} \overline{X4} X5 \overline{X8}$$

$$X(0,3) = \overline{X1} \overline{X4} X5 X8$$

$$Y(0) = \overline{NR} \overline{NG} \overline{HR} \overline{HG}$$

$$X(0,4) = \overline{X1} X4$$

$$X(0,5) = X1 \overline{X4}$$

$$X(0,6) = X1 X4$$

b) Zustand 1

$$X(1,0) = \overline{X5} X7; X(1,2) = X5 X7$$

$$X(1,1) = \overline{X5} \overline{X7}; X(1,3) = X5 \overline{X7}$$

$$Y(1) = \overline{NR} \overline{NG} \overline{HR} \overline{HG}$$

c) Zustand 2

$$X(2,0) = X6 \overline{X8}; X(2,2) = \overline{X6} \overline{X8}$$

$$X(2,1) = X6 X8; X(2,3) = \overline{X6} X8$$

$$Y(2) = \overline{NR} \overline{NG} \overline{HR} \overline{HG}$$

d) Zustand 3

$$X(3,0) = X6 X7; X(3,2) = \overline{X6} X7$$

$$X(3,1) = X6 \overline{X7}; X(3,3) = \overline{X6} \overline{X7}$$

$$Y(3) = \overline{NR} \overline{NG} \overline{HR} \overline{HG}$$

e) Zustand 4

$$X(4,0) = \overline{X1} X3; X(4,5) = X1 X3$$

$$X(4,4) = \overline{X1} \overline{X3}; X(4,6) = X1 \overline{X3}$$

$$Y(4) = \overline{NR} \overline{NG} \overline{HR} \overline{HG}$$

f) Zustand 5

$$X(5,0) = X2 \overline{X4}; X(5,5) = \overline{X2} \overline{X4}$$

$$X(5,4) = X2 X4; X(5,6) = \overline{X2} X4$$

$$Y(5) = \overline{NR} \overline{NG} \overline{HR} \overline{HG}$$

g) Zustand 6

$$X(6,0) = X2 X3; X(6,5) = \overline{X2} X3$$

$$X(6,4) = X2 \overline{X3}; X(6,6) = \overline{X2} \overline{X3}$$

$$Y(6) = \overline{NR} \overline{NG} \overline{HR} \overline{HG}$$

Für den Entwurf des Schaltwerks wird ein Festwertspeicher mit vorgeschaltetem Multiplexer ausgewählt (Bild Ü22.4). Durch die Maskierung der Eingangsvariablen wird der Adreßumfang von 2^{11} auf 2^7 reduziert. Als Zustandsvariablenpeicher wird ein D-Register mit Rücksetzeingang eingesetzt, so daß mit $\neg \text{pon} = 0$ der Anfangszustand eingestellt werden kann. Ein weiteres D-Register synchronisiert die Eingangsvariablen auf. Aus Gründen der Übersichtlichkeit ist auf eine besondere Unterscheidung im Signalnamen zwischen den Eingangsvariablen und den synchronisierten Eingangsvariablen verzichtet worden. In Tabelle Ü22.4 ist die Programmier­tabelle für den Festwertspeicher abgebildet. Man beachte, daß für die direkt angeschlossenen Eingangsvariablen X1 und X4 in den Zuständen 1 bis 6 stets alle mögliche Kombinationen berücksichtigt werden müssen.

Tabelle Ü22.2: Zustandsfolgetabelle für einen Moore-Automaten

| Eingangsvariablen | | | | | | | | Zustandsvariablen | | | | | | Ausgangsvariablen | | | |
|-------------------|----|----|----|----|----|----|----|-------------------|----|----|-----|-----|-----|-------------------|----|----|----|
| m | | | | | | | | m | | | m+1 | | | m | | | |
| X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | Z1 | Z2 | Z3 | Z1* | Z2* | Z3* | NR | NG | HR | HG |
| 0 | - | - | 0 | 0 | - | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | - | - | 0 | 0 | - | - | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | - | - | 0 | 1 | - | - | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | - | - | 0 | 1 | - | - | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | - | - | 1 | - | - | - | - | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | - | - | 0 | - | - | - | - | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | - | - | 1 | - | - | - | - | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| - | - | - | - | 0 | - | 0 | - | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| - | - | - | - | 0 | - | 1 | - | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| - | - | - | - | 1 | - | 0 | - | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| - | - | - | - | 1 | - | 1 | - | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| - | - | - | - | - | 0 | - | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| - | - | - | - | - | 1 | - | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| - | - | - | - | - | 0 | - | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| - | - | - | - | - | 1 | 0 | - | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| - | - | - | - | - | 1 | - | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | - | 0 | - | - | - | - | - | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | - | 1 | - | - | - | - | - | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | - | 0 | - | - | - | - | - | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | - | 1 | - | - | - | - | - | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| - | 0 | - | 0 | - | - | - | - | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| - | 1 | - | 0 | - | - | - | - | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| - | 0 | - | 1 | - | - | - | - | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| - | 1 | - | 1 | - | - | - | - | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| - | 0 | 0 | - | - | - | - | - | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| - | 0 | 1 | - | - | - | - | - | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| - | 1 | 0 | - | - | - | - | - | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| - | 1 | 1 | - | - | - | - | - | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

Tabelle Ü22.3: Multiplexer-Zuordnungstabelle

| Z | Z1 | Z2 | Z3 | F1 | F2 | F3 | F4 |
|---|----|----|----|----|----|----|----|
| 0 | 0 | 0 | 0 | X1 | X4 | X5 | X8 |
| 1 | 0 | 0 | 1 | - | - | X5 | X7 |
| 2 | 0 | 1 | 0 | - | - | X6 | X8 |
| 3 | 0 | 1 | 1 | - | - | X6 | X7 |
| 4 | 1 | 0 | 0 | - | - | X1 | X3 |
| 5 | 1 | 0 | 1 | - | - | X2 | X4 |
| 6 | 1 | 1 | 0 | - | - | X2 | X3 |
| 7 | 1 | 1 | 1 | - | - | - | - |

F1 = X1 und F2 = X4

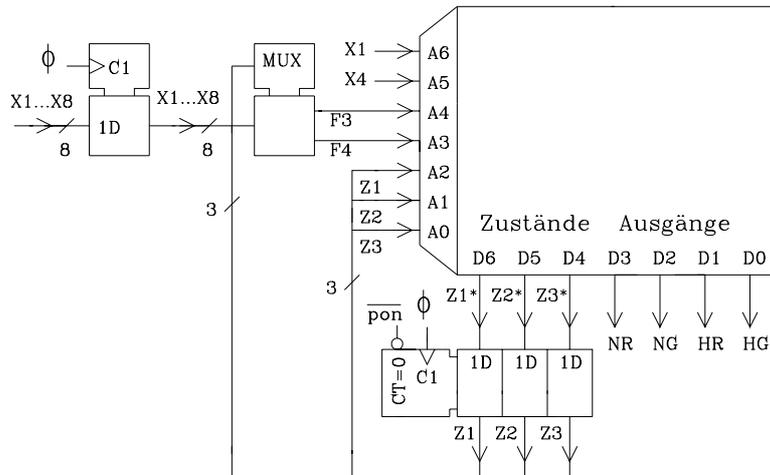


Bild Ü22.4: Schaltwerk mit maskierten Eingangsvariablen und kombinatorischer Ausgabe

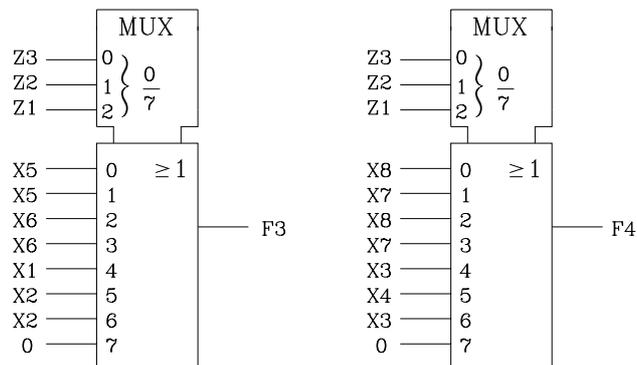


Bild Ü22.5: Multiplexer zur Maskierung der Eingangsvariablen

Tabelle Ü22.4: Programmiertabelle für den Festwertspeicher

| X1 | X4 | F3 | F4 | Z1 | Z2 | Z3 | - | Z1* | Z2* | Z3* | NR | NG | HR | HG |
|----|----|----|----|----|----|----|----|-----|-----|-----|----|----|----|----|
| A6 | A5 | A4 | A3 | A2 | A1 | A0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 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 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |

| X1 A6 | X4 A5 | F3 A4 | F4 A3 | Z1 A2 | Z2 A1 | Z3 A0 | - D7 | Z1* D6 | Z2* D5 | Z3* D4 | NR D3 | NG D2 | HR D1 | HG D0 |
|----------|----------|----------|----------|----------|----------|----------|---------|-----------|-----------|-----------|----------|----------|----------|----------|
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

Aufgabe 23: Entwurf eines Speichersystems mit 8-Bit-Wortbreite

Entwerfen Sie ein digitales Speichersystem mit folgenden Eigenschaften:

- 16 Adreßleitungen und 8 Datenleitungen (Wortbreite: 8 Bit)
- Adreßbereich des RAMs: 0 ... 3FFFH (hexadezimal)
- Adreßbereich des ROMs: 8000H ... BFFFH (hexadezimal)

Zur Verfügung stehen statische RAMs der Speicherkapazität 4K x 8 Bit und EPROMs der Speicherkapazität 8K x 8 Bit. Beide Speichertypen haben je einen \bar{CS} - und einen \bar{OE} -Anschluß. Die RAMs haben zusätzlich noch einen Schreibeingang \bar{WE} .

- 23.1 Geben Sie ein Blockschaltbild des gesamten Speichersystems an.
- 23.2 Geben Sie die Gleichungen für vollständige und unvollständige Decodierung an. Nennen Sie Vor- und Nachteile der beiden Decodierungsarten.
- 23.3 Im RAM-Bereich soll nun wahlweise der Speicher im Adreßbereich von 3800H ... 3FFFH per Schalter gesperrt werden können. Geben Sie für den Fall der vollständigen Decodierung die Änderung der Gleichungen und die entsprechende Schaltung an.

Lösung zu 23.1: Die Adressen eines Speicherbereichs werden häufig als Hexadezimalzahlen angegeben, während die Speicherkapazität der Schreib-/Lese- und Festwertspeicher mit Potenzzahlen zur Basis 2 bezeichnet wird. Aus dem Grund werden in einer Tabelle die für die Aufgabe wichtigen Umrechnungen zwischen Potenzzahlen zur Basis 2, Hexadezimalzahlen und Dezimalzahlen angegeben.

Tabelle Ü23.1: Umrechnungstabelle zwischen Potenzzahlen zur Basis 2, Hexadezimalzahlen und Dezimalzahlen

| Potenzzahl zur Basis 2 | Abkürzung | Hexadezimalzahl | Dezimalzahl |
|------------------------|-----------|-----------------|-------------|
| 2^{10} | 1K | 400H | 1024 |
| 2^{11} | 2K | 800H | 2048 |
| 2^{12} | 4K | 1000H | 4096 |
| 2^{13} | 8K | 2000H | 8192 |
| 2^{14} | 16K | 4000H | 16384 |
| 2^{15} | 32K | 8000H | 32768 |

Für die erforderliche RAM-Speicherkapazität von 4000H x 8 Bit werden nach Tabelle Ü23.1 vier statische RAMs der Kapazität 4K x 8 Bit benötigt. Der geforderte Festwertspeicher von ebenfalls 4000H x 8 Bit läßt sich mit zwei EPROMs des zur Verfügung stehenden Typs (Speicherkapazität = 8K x 8 Bit) realisieren. In Bild Ü23.1 ist das geforderte Blockschaltbild dargestellt.

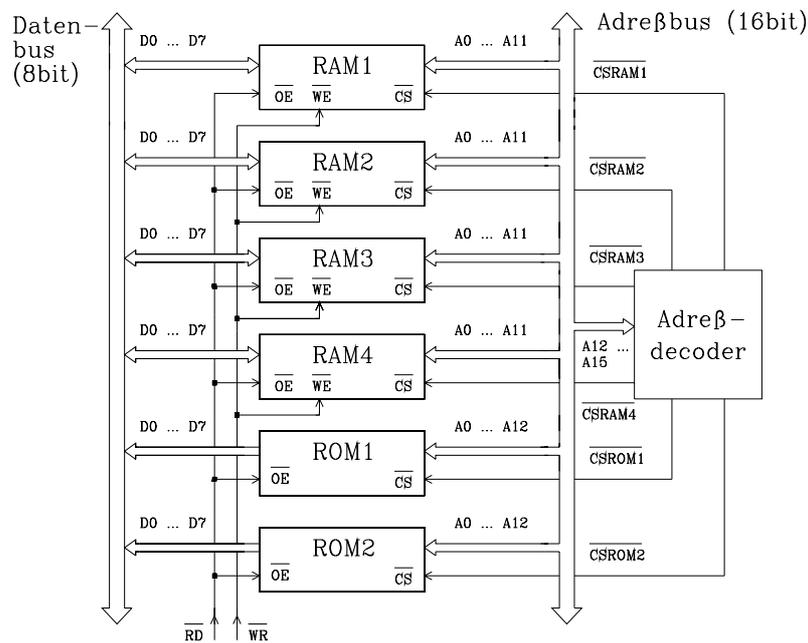


Bild Ü23.1: Blockschaltbild des gesuchten Speichersystems

Lösung zu 23.2: In einer Tabelle (Tabelle Ü23.2) ist die Aufteilung des gesamten Adreßbereichs dargestellt. Für spätere Erweiterungen sind noch zwei Bereiche reserviert. Mit Hilfe der angegebenen Anfangs- und Endadressen der einzelnen Speicherbausteine lassen sich die Gleichungen für vollständige und unvollständige Adreßdecodierung aufstellen.

Bei der vollständigen Adreßdecodierung werden alle im System zur Verfügung stehenden Adreßbits, die nicht schon am Speicherbaustein angeschlossen sind, zur Decodierung mit herangezogen. Für die unvollständige Adreßdecodierung werden nur die zur Unterscheidung der einzelnen Speicherbausteine unbedingt notwendigen Adreßbits zur Decodierung verwendet.

Der Hardwareaufwand ist bei der vollständigen Adreßdecodierung größer als bei der unvollständigen. Vorteilhaft ist bei der vollständigen Adreßdecodierung die eindeutige Beziehung zwischen Adresse und Speicherplatz, während bei der unvollständigen ein Speicherplatz unter mehreren Adressen angesprochen werden kann. Eine Speichererweiterung ist bei vollständiger Adreßdecodierung leicht möglich, während bei unvollständiger alle Gleichungen überprüft und ggf. korrigiert werden müssen.

Tabelle Ü23.2: Speicherbelegungsplan für die gestellte Aufgabe

| Speicher Adresse- | Adresse (hex.) | Adresse (binär) | | | | | | | Speichertyp | –Chip Se- lect-Signal |
|----------------------|-------------------|-----------------|-----|-----|-----|-----|-----|----|-------------|--------------------------|
| | | A15 | A14 | A13 | A12 | A11 | ... | A0 | | |
| Anfang | 0000 | 0 | 0 | 0 | 0 | 0 | ... | 0 | RAM1 | –CSRAM1 |
| Ende | 0FFF | 0 | 0 | 0 | 0 | 1 | ... | 1 | | |
| Anfang | 1000 | 0 | 0 | 0 | 1 | 0 | ... | 0 | RAM2 | –CSRAM2 |
| Ende | 1FFF | 0 | 0 | 0 | 1 | 1 | ... | 1 | | |
| Anfang | 2000 | 0 | 0 | 1 | 0 | 0 | ... | 0 | RAM3 | –CSRAM3 |
| Ende | 2FFF | 0 | 0 | 1 | 0 | 1 | ... | 1 | | |
| Anfang | 3000 | 0 | 0 | 1 | 1 | 0 | ... | 0 | RAM4 | –CSRAM4 |
| Ende | 3FFF | 0 | 0 | 1 | 1 | 1 | ... | 1 | | |
| Anfang | 4000 | 0 | 1 | 0 | 0 | 0 | ... | 0 | nichtbelegt | |
| Ende | 7FFF | 0 | 1 | 1 | 1 | 1 | ... | 1 | | |
| Anfang | 8000 | 1 | 0 | 0 | 0 | 0 | ... | 0 | ROM1 | –CSROM1 |
| Ende | 9FFF | 1 | 0 | 0 | 1 | 1 | ... | 1 | | |
| Anfang | A000 | 1 | 0 | 1 | 0 | 0 | ... | 0 | ROM2 | –CSROM2 |
| Ende | BFFF | 1 | 0 | 1 | 1 | 1 | ... | 1 | | |
| Anfang | C000 | 1 | 1 | 0 | 0 | 0 | ... | 0 | nichtbelegt | |
| Ende | FFFF | 1 | 1 | 1 | 1 | 1 | ... | 1 | | |

Tabelle Ü23.3: Gleichungen für die vollständige und unvollständige Adreßdecodierung*Vollständige Adreßdecodierung*

$$\overline{\text{CSRAM1}} = \overline{\text{A15}} \overline{\text{A14}} \overline{\text{A13}} \overline{\text{A12}}$$

$$\overline{\text{CSRAM2}} = \overline{\text{A15}} \overline{\text{A14}} \overline{\text{A13}} \overline{\text{A12}}$$

$$\overline{\text{CSRAM3}} = \overline{\text{A15}} \overline{\text{A14}} \overline{\text{A13}} \overline{\text{A12}}$$

$$\overline{\text{CSRAM4}} = \overline{\text{A15}} \overline{\text{A14}} \overline{\text{A13}} \overline{\text{A12}}$$

$$\overline{\text{CSROM1}} = \overline{\text{A15}} \overline{\text{A14}} \overline{\text{A13}}$$

$$\overline{\text{CSROM2}} = \overline{\text{A15}} \overline{\text{A14}} \overline{\text{A13}}$$

Unvollständige Adreßdecodierung

$$\overline{\text{CSRAM1}} = \overline{\text{A15}} \overline{\text{A13}} \overline{\text{A12}}$$

$$\overline{\text{CSRAM2}} = \overline{\text{A15}} \overline{\text{A13}} \overline{\text{A12}}$$

$$\overline{\text{CSRAM3}} = \overline{\text{A15}} \overline{\text{A13}} \overline{\text{A12}}$$

$$\overline{\text{CSRAM4}} = \overline{\text{A15}} \overline{\text{A13}} \overline{\text{A12}}$$

$$\overline{\text{CSROM1}} = \overline{\text{A15}} \overline{\text{A13}}$$

$$\overline{\text{CSROM2}} = \overline{\text{A15}} \overline{\text{A13}}$$

Lösung zu 23.3: Der zu sperrende Adreßbereich liegt vollständig im Bereich des Bausteins RAM4. Folglich muß auch nur die Gleichung –CSRAM4 geändert werden. In die Adreßdecodierung wird nun zusätzlich das Adreßbit A11 und der vom Schalter einstellbare Logik-Zustand S einbezogen.

Es gilt folgende Zuordnung (Bild Ü23.2):

Schalter geöffnet (S = 1): Speicher RAM4 kann im Adreßbereich von 3000H bis 3FFFH angesprochen werden.

Schalter geschlossen (S = 0): Speicher RAM4 kann nur im Adreßbereich von 3000H bis 37FFH angesprochen werden.

$$\text{CSRAM4}^* = \overline{\text{A15}} \overline{\text{A14}} \overline{\text{A13}} \overline{\text{A12}} \overline{\text{A11}} \text{S} \vee \overline{\text{A15}} \overline{\text{A14}} \overline{\text{A13}} \overline{\text{A12}} \overline{\text{A11}} \overline{\text{S}} \vee \overline{\text{A15}} \overline{\text{A14}} \overline{\text{A13}} \overline{\text{A12}} \overline{\text{A11}} \overline{\text{S}}$$

$$\overline{\text{CSRAM4}}^* = \overline{\text{A15}} \overline{\text{A14}} \text{A13} \text{A12} \text{S} (\text{A11} \vee \overline{\text{A11}}) \vee \overline{\text{A15}} \overline{\text{A14}} \text{A13} \text{A12} \overline{\text{A11}} (\text{S} \vee \overline{\text{S}})$$

$$\overline{\text{CSRAM4}}^* = \overline{\text{A15}} \overline{\text{A14}} \text{A13} \text{A12} \text{S} \vee \overline{\text{A15}} \overline{\text{A14}} \text{A13} \text{A12} \overline{\text{A11}}$$

Negiert man beide Seiten, so erhält man:

$$\text{CSRAM4}^* = \overline{\overline{\text{A15}} \overline{\text{A14}} \text{A13} \text{A12} \text{S} \vee \overline{\text{A15}} \overline{\text{A14}} \text{A13} \text{A12} \overline{\text{A11}}}$$

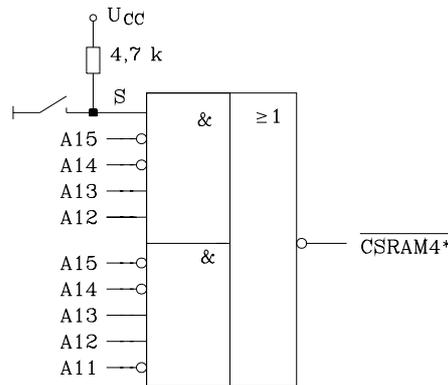


Bild Ü23.2: Mit Hilfe eines Schalters läßt sich der Speicher RAM4 im Adreßbereich 3800H bis 3FFFH sperren

Aufgabe 24: Speichersystem mit 16-Bit-Datenbus

Entwerfen Sie ein Speichersystem mit den angegebenen Speicherkapazitäten.

- Gesamter RAM-Bereich: 128k * 16 Bit
- Gesamter Festwertspeicher-Bereich: 256k * 16 Bit

Es gelten folgende Randbedingungen:

- 20-Bit-Adreßbus und 16-Bit-Datenbus
- Anfangsadresse RAM-Bereich: 00000H
- Anfangsadresse ROM-Bereich: 80000H

Zur Verfügung stehen folgende Speicherbausteine:

Statische RAMs der Kapazität 32K * 8 Bit und EPROMs der Kapazität 128K * 8 Bit

Gesucht sind die Gleichungen für die vollständige und unvollständige Adreßdecodierung.

Zusatzfrage:

Wie lauten die Gleichungen für die vollständige Adreßdekodierung, falls die Anfangsadresse für den ROM-Bereich 88000H ist?

Lösung:

Es werden 8 SRAMs der Kapazität 32K * 8 Bit und 4 EPROMs der Kapazität 128K * 8 Bit gewählt.

| Adresse (hex.) | Adresse (binär) | | | | | | | | Chip Se- lect-Signal |
|-------------------|-----------------|-----|-----|--------------------|-------|--------------------|----------|---------|-------------------------|
| | A19 | A18 | A17 | A16 | A15 | A14...A 0 | D15...D8 | D7...D0 | |
| 0000 07FFF | 0 | 0 | 0 | 0 | 0 | 0.....0 1.....1 | RAM 2 | RAM 1 | CSRAM1-2 |
| 08000 0FFFF | 0 | 0 | 0 | 0 | 1 | 0.....0 1.....1 | RAM 4 | RAM 3 | CSRAM3-4 |
| 10000 17FFF | 0 | 0 | 0 | 1 | 0 | 0.....0 1.....1 | RAM 6 | RAM 5 | CSRAM5-6 |
| 18000 1FFFF | 0 | 0 | 0 | 1 | 1 | 0.....0 1.....1 | RAM 8 | RAM 7 | CSRAM7-8 |
| 20000 7FFFF | 0 | 0 | 1 | 0.....0 1.....1 | frei | frei | | | |
| 80000 9FFFF | 1 | 0 | 0 | 0.....0 1.....1 | ROM 2 | ROM 1 | CSROM1-2 | | |
| A0000 BFFFF | 1 | 0 | 1 | 0.....0 1.....1 | ROM 4 | ROM 3 | CSROM3-4 | | |
| C0000 FFFFF | 1 | 1 | 0 | 0.....0 1.....1 | frei | frei | | | |

*Adreßdekodierung:**vollständig*

$$\overline{\text{CSRAM1-2}} = \overline{\text{A19 A18 A17 A16 A15}}$$

$$\overline{\text{CSRAM3-4}} = \overline{\text{A19 A18 A17 A16 A15}}$$

$$\overline{\text{CSRAM5-6}} = \overline{\text{A19 A18 A17 A16 A15}}$$

$$\overline{\text{CSRAM7-8}} = \overline{\text{A19 A18 A17 A16 A15}}$$

$$\overline{\text{CSROM1-2}} = \overline{\text{A19 A18 A17}}$$

$$\overline{\text{CSROM3-4}} = \overline{\text{A19 A18 A17}}$$

unvollständig

$$\overline{\text{CSRAM1-2}} = \overline{\text{A19 A16 A15}}$$

$$\overline{\text{CSRAM3-4}} = \overline{\text{A19 A16 A15}}$$

$$\overline{\text{CSRAM5-6}} = \overline{\text{A19 A16 A15}}$$

$$\overline{\text{CSRAM7-8}} = \overline{\text{A19 A16 A15}}$$

$$\overline{\text{CSROM1-2}} = \overline{\text{A19 A17}}$$

$$\overline{\text{CSROM3-4}} = \overline{\text{A19 A17}}$$

Zur Zusatzfrage:

| Adresse (hex.) | Adresse (binär) | | | | | | | | Chip Se- lect-Signal |
|-------------------|-----------------|-----|-----|-----|-----|--------------------|----------|---------|-------------------------|
| | A19 | A18 | A17 | A16 | A15 | A14...A 0 | D15...D8 | D7...D0 | |
| 88000 A7FFF | 1 | 0 | 0 | 0 | 1 | 0.....0 1.....1 | ROM 2 | ROM 1 | CSROM1-2 |
| A8000 C7FFF | 1 | 0 | 1 | 0 | 1 | 0.....0 1.....1 | ROM 4 | ROM 3 | CSROM3-4 |

Vollständige Adreßdekodierung:

$$\overline{\text{CSROM1-2}} = \overline{\text{A19 A18 A17 A16 A15}} \vee \overline{\text{A19 A18 A17 A16 A15}}$$

$$\vee \overline{\text{A19 A18 A17 A16 A15}} \vee \overline{\text{A19 A18 A17 A16 A15}}$$

$$\overline{\text{CSROM3-4}} = \overline{\text{A19 A18 A17 A16 A15}} \vee \overline{\text{A19 A18 A17 A16 A15}}$$

$$\vee \overline{\text{A19 A18 A17 A16 A15}} \vee \overline{\text{A19 A18 A17 A16 A15}}$$

Die Gleichungen lassen sich wie folgt vereinfachen:

$$\overline{\text{CSROM1-2}} = \overline{\text{A19 A18 A17 A15}} \vee \overline{\text{A19 A18 A17 A16}}$$

$$\vee \overline{\text{A19 A18 A17 A16 A15}}$$

$$\overline{\text{CSROM3-4}} = \overline{\text{A19 A18 A17 A15}} \vee \overline{\text{A19 A18 A17 A16}}$$

$$\vee \overline{\text{A19 A18 A17 A16 A15}}$$

Aufgabe 25: Entwurf eines Schaltnetzes mit einem Festwertspeicher

Gegeben ist ein logisches Gleichungssystem:

$$Y1 = \overline{X5} X3 X1 \vee \overline{X4} X2 \overline{X1} \vee X5 \overline{X3} X2 \vee \overline{X2} \overline{X1}$$

$$Y2 = X5 X4 X3 X2 X1 \vee \overline{X5} X4 \overline{X1} \vee \overline{X3} X1$$

$$Y3 = \overline{X2} \overline{X1} \vee X3 X2 \vee X5 X4 X1$$

$$Y4 = \overline{X4} X3 \overline{X1} \vee \overline{X5} X3 X2 \vee X5 \overline{X2}$$

$$Y5 = \overline{X3} X2 \vee \overline{X4} X3 X1 \vee X5 \overline{X4} X1$$

$$Y6 = \overline{X5} X4 \overline{X2} \vee \overline{X3} X2 \overline{X1} \vee X2 X1$$

$$Y7 = \overline{X4} X3 \overline{X2} \vee \overline{X5} X3 X1 \vee X4 \overline{X2}$$

$$Y8 = X5 \overline{X3} X2 \vee X2 \overline{X1}$$

Entwerfen Sie das zugehörige Schaltnetz unter Verwendung eines Festwertspeichers. Geben Sie den Typ des Festwertspeichers und die Programmier­tabelle an.

Lösung:

Da ein Festwertspeicher eingesetzt wird, muß sowohl die Hardware (Bild Ü25.1) als auch die Software in Form einer Programmier­tabelle angegeben werden (Tabelle Ü25.1). Man beachte, daß die Zuordnung zwischen Eingangs- und Ausgangsgrößen einerseits und Adreßeingängen und Datenausgängen andererseits sowohl für die Hardware als auch für die Programmier­tabelle gleich gewählt wird. Der -CS-Anschluß des PROMs liegt an "0".

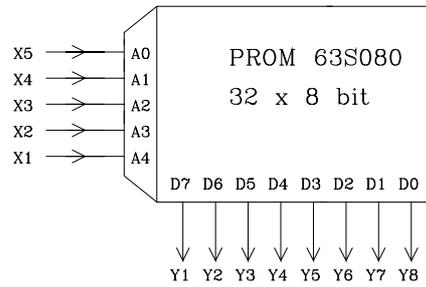


Bild Ü25.1: Hardware des gesuchten Schaltnetzes

Tabelle Ü25.1: Programmier­tabelle für das gesuchte Schaltnetz

| Adr. hex. | X1 A4 | X2 A3 | X3 A2 | X4 A1 | X5 A0 | Y1 D7 | Y2 D6 | Y3 D5 | Y4 D4 | Y5 D3 | Y6 D2 | Y7 D1 | Y8 D0 | Daten hex. |
|--------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|---------------|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | A0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | B0 |
| 2 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | E6 |
| 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | B2 |
| 4 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | B2 |
| 5 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | B2 |
| 6 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | E6 |
| 7 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | B2 |
| 8 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 8D |
| 9 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 8D |
| A | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 4D |
| B | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 8D |
| C | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | B1 |
| D | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | B1 |
| E | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 71 |
| F | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 21 |
| 10 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 40 |
| 11 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 58 |
| 12 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 46 |
| 13 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 72 |
| 14 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 8A |
| 15 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1A |
| 16 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 86 |
| 17 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 32 |
| 18 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 4C |
| 19 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | CD |
| 1A | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 4C |
| 1B | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | ED |
| 1C | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | BE |
| 1D | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 2C |
| 1E | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | B6 |
| 1F | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 64 |

Aufgabe 26: Entwurf einer Testschaltung

Entwerfen Sie eine digitale Schaltung zum Testen von NAND-Gattern mit drei Eingängen. Setzen Sie einen Zähler zum Stimulieren des Testobjekts ein und werten Sie das Ergebnis mit Hilfe eines PROMs aus. Das Schaltnetz soll zwei Ausgänge (ERG_R und ERG_F) erhalten, die jeweils über LEDs angezeigt werden. Erweitern Sie die Schaltung so, daß der Zähler im Fehlerfall angehalten wird. Der Start zu einem neuen Testlauf soll per Tastendruck über den Reseteingang des Zählers erfolgen.

Testergebnis:

| ERG_R | ERG_F | Funktionstest |
|-------|-------|-----------------------|
| 1 | 0 | positiv (kein Fehler) |
| 0 | 1 | negativ (Fehler) |

Geben Sie die vollständige Schaltung mit geeignetem Zähler, PROM inklusive Programmierstabelle und die Ansteuerung der LEDs an.

Lösung:

Tabelle Ü26.1: Wahrheitstabelle des NAND-Gatters mit drei Eingängen

| X1 | X2 | X3 | XA |
|----|----|----|----|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Tabelle Ü26.2: Programmierstabelle des PROMs für die Testschaltung

| XA | X1 | X2 | X3 | ERG_R | ERG_F |
|----|----|----|----|-------|-------|
| A3 | A2 | A1 | A0 | D1 | D0 |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 |

Zur Erzeugung des Bitmusters wird ein synchroner 3-Bit-Dualzähler eingesetzt, der vorwärts zählt und über einen negierten Rücksetzeingang mit "0" rückgesetzt werden kann. Die Zählerausgänge sind sowohl mit dem Testobjekt (NAND-Gatter mit drei Eingängen) als auch mit den Adreßeingängen des Festwertspeichers verbunden. Weiterhin ist der Ausgang des Testobjekt an einen Adreßanschluß des PROMs angeschlossen. Das PROM ist so programmiert, daß der Ausgang $ERG_R = 1$ und $ERG_F = 0$ werden, falls das Testobjekt fehlerfrei ist. Im Fehlerfall werden die Signale ERG_R und ERG_F ihre Polarität ändern.

1.Fall: Betrieb ohne Rückkopplung: Der Takt ϕ mit einer Periodendauer von ca. 1 Hz wird direkt an den Takteingang des Zählers gelegt. Über den Taster kann der Zähler auf 0 gesetzt werden und der Test beginnt beim Loslassen des Tasters. Falls das NAND fehlerfrei arbeitet, wird LED_R aufleuchten, andernfalls LED_F .

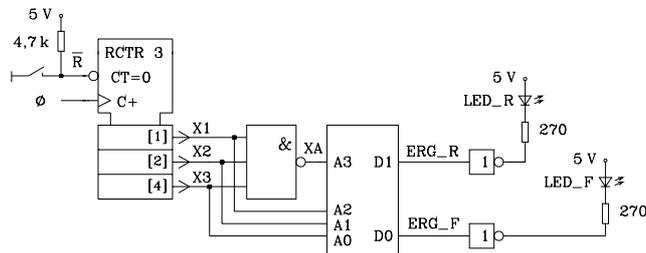


Bild Ü26.1: Blockschaltbild der Testschaltung ohne Rückkopplung

2.Fall: Betrieb mit Rückkopplung: Falls ein Fehler beim Testen festgestellt wird soll der Zähler angehalten werden, so daß die Bitkombination, bei der der Fehler aufgetreten ist, festgestellt werden kann. Mit Hilfe eines UND-Gatters als Torschaltung läßt sich der Zählertakt in Abhängigkeit eines PROM-Ausgangs sperren.

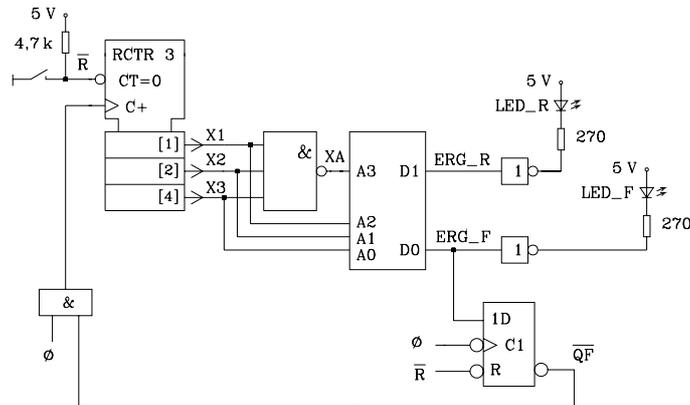


Bild Ü26.2: Blockschaltbild der Testschaltung mit Rückkopplung

Erfolgt eine direkte Rückkopplung eines PROM-Ausgangs auf den zweiten UND-Eingang, so könnten aufgrund von Signaleinbrüchen an den Datenausgängen bei der Adressenumschaltung mehrere Zähltaktflanken erzeugt werden. Deshalb wird der Ausgang ERG_F mit der negativen Flanke des Taktes ϕ in einem D-Flipflop zwischengespeichert, und der negierte Flipflopausgang wird rückgekoppelt. Das D-Flipflop wird zu Beginn des Testes mit dem Tastersignal $\neg R$ rückgesetzt. Im Fehlerfall wird der Zähler rechtzeitig angehalten.

Aufgabe 27: Mikrocontrollersystem mit externer Speichererweiterung

Es soll ein Mikrocontrollersystem auf der Basis des 8-Bit-Mikrocontrollers 87C52 mit externem Festwert- und Schreib-/Lesespeicher aufgebaut werden. Der 87C52 enthält ein 8-KByte-EPROM und ein 256-Byte-RAM sowie drei Timer, ansonsten hat er die gleichen Anschlüsse und Eigenschaften wie der 80C51.

Das Mikrocontrollersystem soll auf einen zusammenhängenden Festwertspeicher (Programmspeicher) der Kapazität 40 KByte erweitert werden. Außerdem soll ein externer Datenspeicher (RAM) von 24 KByte vorgesehen werden.

Es stehen EPROMs mit den Speicherkapazitäten 8 KByte, 16 KByte und 32 KByte zur Verfügung. Für den Aufbau des geforderten 24-KByte-Datenspeichers in einem zusammenhängenden Adreßbereich stehen SRAMs der Kapazität 8 KByte, 16 KByte und 32 KByte zur Verfügung. Die EPROMs haben die Steuereingänge $\neg CS$ und $\neg OE$ und die SRAMs $\neg CS$, $\neg WE$ und $\neg RD$.

Wählen Sie sowohl für den Festwert- als auch für den Datenspeicher geeignete Bausteine aus. Die *Anzahl* der Speicherbausteine soll *minimal* werden. Legen Sie sinnvolle Anfangsadressen fest und begründen Sie Ihre Wahl.

Geben Sie die minimalen Gleichungen für vollständige Adreßdecodierung an. Zur Verfügung steht ein PAL mit *negierten* Ausgängen.

Lösung:

Entwurf des Mikrorechnersystems. Die Verteilung der vorgesehenen Speicherbausteine im adressierbaren Bereich des Mikrocontrollers erfolgt tabellarisch gemäß Tabelle Ü27.1.

Das Blockschaltbild des Mikrocontrollersystems entspricht der Schaltung in Bild 9.28 (Kap. 9), wenn folgende Änderungen durchgeführt werden: Statt 8051 muß der Mikrocontroller 87C52 eingesetzt werden. Da der interne Programmspeicher verwendet wird, muß der Anschluß $\neg EA$ an H-Pegel angeschlossen werden.

Tabelle Ü27.1: Speicherbelegungsplan für die gestellte Aufgabe

| Adreß-Bereich | Adresse hex. | Adresse binär | | | | | Speichertyp | CS-Signal |
|----------------|--------------------|---------------|-----|-----|-----|--------------------|----------------------------|---------------------------------------|
| | | A15 | A14 | A13 | A12 | A11...A0 | | |
| Anfang Ende | 0 0 0 0 1 F F F | 0 | 0 | 0 | 0 | 0.....0 1.....1 | EPROM 8 KByte in- | --- |
| Anfang Ende | 2 0 0 0 3 F F F | 0 | 0 | 1 | 0 | 0.....0 1.....1 | EPROM 32 KByte | Bereich 1 $\overline{A15} A14 A13$ |
| Anfang Ende | 4 0 0 0 5 F F F | 0 | 1 | 0 | 0 | 0.....0 1.....1 | extern, in 4 Teilberei- | Bereich 2 $\overline{A15} A14 A13$ |
| Anfang Ende | 6 0 0 0 7 F F F | 0 | 1 | 1 | 0 | 0.....0 1.....1 | chen mit je 8 KByte | Bereich 3 $\overline{A15} A14 A13$ |
| Anfang Ende | 8 0 0 0 9 F F F | 1 | 0 | 0 | 0 | 0.....0 1.....1 | | Bereich 4 $A15 A14 A13$ |
| Anfang Ende | A 0 0 0 F F F F | 1 | 0 | 1 | 0 | 0.....0 1.....1 | nicht belegt | --- |
| Anfang Ende | 0 0 0 0 3 F F F | 0 | 0 | 0 | 0 | 0.....0 1.....1 | SRAM 24 KByte, in | Bereich 1 $A15 A14$ |
| Anfang Ende | 4 0 0 0 5 F F F | 0 | 1 | 0 | 0 | 0.....0 1.....1 | 2 Bereichen | Bereich 2 $\overline{A15} A14 A13$ |
| Anfang Ende | 6 0 0 0 F F F F | 0 | 1 | 1 | 0 | 0.....0 1.....1 | nicht belegt | --- |

Vollständige Adreßdecodierung

$$\overline{CSEPROM} = \overline{A15} A14 A13 \vee \overline{A15} A14 \overline{A13} \vee \overline{A15} A14 A13 \vee A15 \overline{A14} \overline{A13}$$

$$= \overline{A15} A13 \vee \overline{A15} A14 \vee A15 \overline{A14} \overline{A13}$$

$$\overline{CSRAM} = \overline{A15} A14 \vee \overline{A15} A14 \overline{A13} = \overline{A15} A14 \vee \overline{A15} \overline{A13}$$

Anmerkung:

Die Aufgabenstellung fordert eine minimale Anzahl von Speicherbausteinen. Daher wurde für das SRAM ein 32-KByte-Baustein gewählt, der Zugriff aber nur für die geforderten 24 KByte freigeben. Falls Adreßkonflikte im Rechner ausgeschlossen sind, kann der Adreßdecoder die restlichen 8 KByte des SRAMs ebenfalls freigeben.

Aufgabe 28: Tastendecodierung mit dem Mikrocontroller 8051

Gegeben ist ein Tastaturfeld mit 64 Tasten, die matrixförmig angelegt sind (Bild Ü28.1). Zur Entkopplung der Spaltenleitungen bei Mehrfachbetätigung sind in Reihe zu den Tastern Dioden geschaltet. Mit Hilfe eines Mikrocontrollers vom Typ 8051 soll eine Tastendecodierung aufgebaut werden. Es soll der Tastendruck jedes einzelnen Tasters eindeutig erkannt und das Ergebnis gespeichert werden.

Anleitung:

Verwenden Sie die Ports 0 und 1 zur Tastendecodierung. Berücksichtigen Sie für die Taster eine Prellzeit von maximal 9 ms. Geben Sie die entsprechende Hardware an und entwickeln Sie ein C51-Programm, das die Tastenwerte übernimmt und softwaremäßig entprellt.

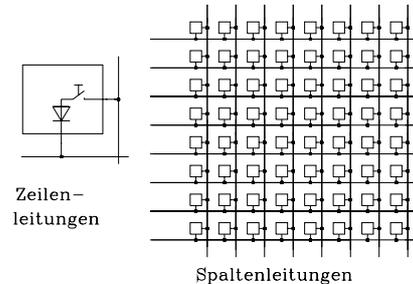


Bild Ü28.1: Matrixförmige Anordnung eines Tastaturfeldes

Lösung:**28.1 Entwurf der Hardware**

Zur Lösung der Aufgabe wird Port 0 auf Eingabe und Port 1 auf Ausgabe eingestellt. An Port 0 werden die Zeilenleitungen und an Port 1 die Spaltenleitungen angeschlossen. Jede Zeilenleitung wird mit einem Pull-Down-Widerstand R_Z und jede Spaltenleitung mit einem Pull-Up-Widerstand R_S abgeschlossen. Über die Pull-Down-Widerstände sind die Eingänge P0.0 bis P0.7 auf L-Pegel (entspricht dem 0-Zustand bei positiver Logik) voreingestellt (Bild Ü28.2).

Mit Hilfe der Software wird in einem Programmmodul ein 8-Bit-Schieberegister realisiert, das eine "1" im Kreise schiebt, alle anderen Bits sind "0". Das Bitmuster wird für eine Zeitspanne von 10 ms an Port 1 ausgegeben. Danach wird die "1" um eine Stelle weitergeschoben usw..

Wird z.B. der H-Pegel an P1.0 ausgegeben, so werden die betätigten Taster der linken Spalte an die entsprechenden Eingänge von Port 0 eine "1" legen. In einer Einleseroutine werden die Anschlüsse von Port 0 ständig abgefragt, und die Datenwörter werden abgespeichert. Da die Taster maximal 9 ms prellen, wird innerhalb der vorgegebenen Zeitspanne zweimal in zeitlichen Abständen von 10 ms das an Port 0 anliegende Datenwort eingelesen und ausgewertet. Erhält man in beiden Fällen das gleiche Ergebnis, ist der Taster sicher geöffnet oder geschlossen, ansonsten prellt er.

Falls an P1.0 H-Pegel ausgegeben wird, liegen die anderen Port-1-Ausgänge und die entsprechenden Spaltenleitungen auf L-Pegel. Die an diese Spaltenleitungen angeschlossenen Taster werden somit nicht abgefragt.

Der Mikrocontroller wird hier ohne Speicherweiterung betrieben. Über die serielle Schnittstelle wird eine Verbindung zu einem Personalcomputer (PC) hergestellt. Vom

PC können Anweisungen an den Mikrocontroller gegeben werden, und der Mikrocontroller kann den erkannten Tastencode an den PC weitergeben.

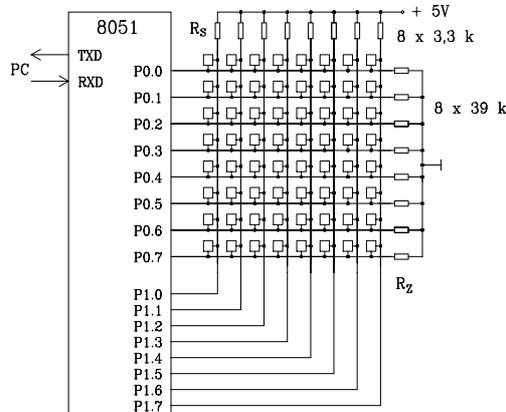


Bild Ü28.2: Einsatz eines Mikrocontrollers zur Tastendecodierung

Dimensionierung der Pull-Up-Widerstände R_S

Jeder Ausgang von Port 1 kann maximal 4 TTL-LS-Lasten ($I_{OL,max} = 1,6 \text{ mA}$) treiben (Kap. 9, Tabelle 9.5). Für den Fall, daß am entsprechenden Port-1-Ausgang L-Pegel ausgegeben wird und in der angeschlossenen Spalte kein Taster geschlossen ist, muß der Ausgang den über den Pull-Up-Widerstand R_S fließenden Strom aufnehmen. Da die Ausgangsspannung für L-Pegel vernachlässigbar klein ist, erhält man folgende Bedingung für R_S :

$$R_S \geq \frac{5V}{I_{OL,max}} = \frac{5V}{1,6 \text{ mA}} = 3,1 \text{ k}\Omega \quad \text{gewählt: } R_S = 3,3 \text{ k}\Omega$$

Dimensionierung der Pull-Down-Widerstände R_Z :

Der Pull-Down-Widerstand wird für den ungünstigsten Fall, der auftreten kann, dimensioniert. Falls an einem Ausgang von Port 1 H-Pegel anliegt und alle Taster der zugehörigen Spalte betätigt sind, sind die acht Pull-Down-Widerstände parallel geschaltet. Es ergibt sich ein Spannungsteiler, bestehend aus R_S , der Diode und $R_Z/8$. Die am Widerstand R_Z anliegende Spannung muß für diesen Fall noch im erlaubten H-Pegelbereich für TTL liegen. Für die Berechnung können die Eingangswiderstände der Portanschlüsse (P0.0 bis P0.7) sowie der Strom des Ausgangsanschlusses von Port 1 vernachlässigt werden. Die Durchlaßspannung der Diode wird mit 0,6 V berücksichtigt. Für die minimale Eingangsspannung bei TTL-Pegel wird unter Berücksichtigung des Störabstandes 2,4 V angenommen.

$$\frac{0,125 \cdot R_Z}{R_S + 0,125 \cdot R_Z} \geq \frac{2,4V}{4,4V} \rightarrow R_Z \geq 31,7 \text{ k}\Omega \quad \text{gewählt: } 33 \text{ k}\Omega$$

Anmerkung:

Als Eingangsport darf nicht anstelle von Port 0 ein anderer Port, z.B. Port 2 gewählt werden. Aufgrund der internen Pull-Up-Widerstände der Port-2-Anschlüsse mit 10 - 40 kΩ ist eine rein passive Schaltung zur Tastendecodierung nicht mehr möglich. Es müssen zusätzliche Treiber verwendet werden.

28.2 Entwurf der Software*C51-Programm für den Mikrocontroller 8051*

```
#include <reg51.h>          /* define 8051 registers */

/*****
/***** KURZBESCHREIBUNG *****/
/* Software zur Tastendecodierung mit dem 8051. Eingabe ueber P0; Ausgabe ueber */
/* P.0=1, P1.1 =1, u.s.w. (10 ms-Zyklus). Abfrage innerhalb von 9 ms zweimal: Ent- */
/* prellen per Software. Nach 10 ms: naechste Spalte der Tastatur */
/* Erforderlich ist ein 100us-Takt, der ueber den Timer0 in Verbindung mit einer Pro- */
/* grammschleife zur Verfuegung gestellt wird. Es wird vorausgesetzt, dass ein Oszilla- */
/* tortakt von 12 MHz zur Verfuegung steht. Timer0 soll als Zeitgeber in der Betriebs- */
/* art 2 (automatisches Rueckladen des Zaehleranfangswertes) arbeiten: Anfangswert = */
/* 250 ---> 250 * 1 us = 0,25 ms. Zaehler bis 4 gezaehlt ---> 1 ms./ millisek <= 80 ms */
/* Daraus folgt: TH0 = -250; TL0 = -250; TMOD = 2; */
/*****/

/*****/
/* Globale Variablen: zaehler, millisek, taste_neu. Array: spalte[8] */
*/
/*****/
unsigned char data zaehler = 0;
unsigned char data millisek = 0;
unsigned char data taste_neu = 0;
unsigned char data spalte[8] = {0,0,0,0,0,0,0,0};

void init();          /* Prototyp der Initialisierungsroutine */

/*****/
/* Funktion: main () */
/* Beschreibung: Hauptprogramm mit Portein- und -ausgabe fuer Tastendecodierung */
/*****/

void main ()
{
    init();
    while(1)          /* Endlosschleife */
    {
        switch (millisek) /* Schleifenbeginn */
        {
            case 0:      /*Tasteneubernahme fuer Spalte 0*/
                P1 = 1;
                taste_neu = P0;
                break;
        }
    }
}
```

```
case 9:
    if (taste_neu == P0) spalte[0] = taste_neu;
    break;

case 10:
    /*Tasteneubernahme fuer Spalte 1*/
    P1 = 2;
    taste_neu = P0;
    break;

case 19:
    if (taste_neu == P0) spalte[1] = taste_neu;
    break;

case 20:
    /*Tasteneubernahme fuer Spalte 2*/
    P1 = 4;
    taste_neu = P0;
    break;

case 29:
    if (taste_neu == P0) spalte[2] = taste_neu;
    break;

case 30:
    /*Tasteneubernahme fuer Spalte 3*/
    P1 = 8;
    taste_neu = P0;
    break;

case 39:
    if (taste_neu == P0) spalte[3] = taste_neu;
    break;

case 40:
    /*Tasteneubernahme fuer Spalte 4*/
    P1 = 16;
    taste_neu = P0;
    break;

case 49:
    if (taste_neu == P0) spalte[4] = taste_neu;
    break;

case 50:
    /*Tasteneubernahme fuer Spalte 5*/
    P1 = 32;
    taste_neu = P0;
    break;

case 59:
    if (taste_neu == P0) spalte[5] = taste_neu;
    break;

case 60:
    /*Tasteneubernahme fuer Spalte 6*/
    P1 = 64;
    taste_neu = P0;
    break;

case 69:
    if (taste_neu == P0) spalte[6] = taste_neu;
    break;

case 70:
    /*Tasteneubernahme fuer Spalte 7*/
```

```
        P1 = 128;
        taste_neu = P0;
        break;
    case 79:
        if (taste_neu == P0) spalte[7] = taste_neu;
    }
}

/*****
/* Funktion: init()
/* Beschreibung: Initialisierung des Timer0, Start und Interruptfreigabe.
/* Anfangswerte festlegen.
*****/

void init()
{
    TH0 = -250; /* Rueckladewert des Timer0 */
    TL0 = TH0; /* Startwert des Timer0 */
    TMOD = 2; /* Timer 0, Betriebsart 2 */
    EA = 1; /* generelle Interrupt-Freigabe */
    ET0 = 1; /* Interrupt Timer0 freigeben */
    TR0 = 1; /* Timer0 starten */
}

/*****
/* Funktion: int_timer0 ()
/* Interruptroutine, wird alle 250 us mit dem Ueberlauf von Timer0 erreicht
*****/

void int_timer0() interrupt 1
{
    zaehler++; /* zaehler zaehlt bis 1 ms */
    if (zaehler == 4)
    {
        zaehler = 0;
        millisek++; /* millisek zaehlt die Millisekunden */
    }
    if (millisek == 80) millisek = 0;
}
```