

Synthese digitaler Schaltungen

Aufgabensammlung

1. Boolesche Algebra

1.1 Schalterrealisierungen von Schaltungsfunktion

a) $y = (x_0 + \overline{x_1}) \cdot (x_0 + x_1) \cdot x_2$

b) $y = (\overline{x_0} + x_1 + x_2) \cdot (\overline{x_2} + x_3) \cdot (\overline{x_0} + \overline{x_1})$

1.2 Überprüfen von Booleschen Gleichungen

a) $\overline{\overline{x_3 \sim x_2 \sim x_1 \sim x_0}} = x_3 \sim x_2 \sim x_1 \sim x_0$

b) $\overline{x_3 \uparrow x_2 \uparrow x_0} = \overline{x_3 \uparrow x_2 \uparrow x_0}$

c) $x_1 + \overline{x_1} \cdot x_0 = x_1 + x_0$

d) $\overline{x_3 x_2 x_0} + \overline{x_2 x_0} + \overline{x_3 x_1} + x_3 x_2 x_1 = x_1 \cdot (\overline{x_3 x_2 x_0} + \overline{x_2 x_0} + x_3 x_2) + \overline{x_1} \cdot (\overline{x_3 x_2 x_0} + \overline{x_2 x_0} + \overline{x_3})$

e) $\overline{x_4 x_1} \cdot (x_3 x_2 + x_3 \overline{x_2}) + x_3 x_1 \cdot (x_4 x_2 + x_4 \overline{x_2}) = x_3 x_1$

f) $\overline{x_1 x_2} + \overline{x_3 x_1} + x_3 \overline{x_2} = \overline{x_2 x_1} + \overline{x_3 x_2} + \overline{x_1 x_3}$

g) $x_1 \uparrow x_0 = \overline{\overline{x_1 \sim x_0}}$

h) $\overline{x_1 x_2 x_0} + x_2 x_1 x_0 + \overline{x_2 x_1 x_0} + \overline{x_2 x_1 x_0} = x_2 \sim x_1 \uparrow x_0$

i) $(x_1 + x_0) \cdot (x_1 + x_2) = x_1 + x_0 x_2$

j) $x_{n-1} \uparrow \dots \uparrow x_i \uparrow \dots \uparrow x_0 = \overline{\overline{x_{n-1} \sim \dots \sim x_i \sim \dots \sim x_0}} \quad (n \text{ gerade})$

k) $x_{n-1} \uparrow \dots \uparrow x_i \uparrow \dots \uparrow x_0 = x_{n-1} \sim \dots \sim x_i \sim \dots \sim x_0 \quad (n \text{ ungerade})$

1.3 Realisierung von Schaltfunktionen mit NAND und NOR

a) $y = x_2 x_1 + x_3 \overline{x_1} + x_3 \overline{x_2} + x_4 x_2 + x_4 x_3$

b) $y = \overline{x_2 x_1 x_0} + \overline{x_2 x_1 x_0} + \overline{x_2 x_1 x_0}$

c) $y = (x_2 + x_1 + \overline{x_0}) \cdot (x_2 + \overline{x_1} + x_0) \cdot (x_2 + x_1 + x_0)$

d) $y = x_4 x_2 x_1 + x_4 \overline{x_3 x_1} + x_4 x_1 x_0$

e) $y = \overline{x_0} \cdot (x_3 + x_2 + x_1)$

f) $y = \overline{x_3 x_1 x_0} + x_0$

1.4 Auslesen von KDNF und KKNF aus der Schaltbelegungstabelle

x_2	x_1	x_0	y_1	y_2	y_3	y_4	y_5
0	0	0	0	1	1	0	0
0	0	1	0	1	1	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	0	1	0
1	0	0	0	0	0	0	1
1	0	1	0	0	1	0	0
1	1	0	1	1	1	0	0
1	1	1	0	1	1	0	0

1.5 Umformen in die KDNF

a) $y = (x_1 + \overline{x_0}) \cdot (x_2 + \overline{x_1} + x_0) \cdot \overline{x_2 x_1}$

b) $y = (x_2 + x_1) \cdot (\overline{x_1} + x_0) \cdot (\overline{x_3 x_2} + x_1)$

c) $y = (x_2 + x_0) \cdot (x_1 x_0 + 1) \cdot \overline{x_3 x_2 x_0}$

d) $y = x_1 x_0 + \overline{x_3 x_1 x_0} + \overline{x_4 x_2 x_1 x_0} + x_2 x_0$

e) $y = \overline{x_3 x_2 x_0}$

$$\text{f) } y = x_1 + 1$$

$$\text{g) } y = x_1 + (x_0 \sim x_1)$$

$$\text{h) } y = (x_1 + x_0) \cdot \overline{x_2 x_1 x_0} + \overline{x_3 x_0}$$

$$\text{i) } y = (x_1 \sim x_0) + (x_2 \sim x_0)$$

$$\text{j) } y = (x_1 \sim x_0) \cdot (x_1 + x_0)$$

$$\text{k) } y = (x_1 + x_0) + (x_1 \sim x_0)$$

1.6 Umformen in die KKNF

$$\text{a) } y = x_2 + x_1 + x_3 x_0 + \overline{x_2 x_0} + x_2 x_1 \overline{x_0}$$

$$\text{b) } y = x_3 x_2 \overline{x_1} + x_2 + x_1 x_0 + x_3 \overline{x_0}$$

$$\text{c) } y = x_0 + x_2 \overline{x_1} + x_3 \overline{x_0} + x_3 x_2 \overline{x_1}$$

$$\text{d) } y = x_3 x_2 \overline{x_0} + \overline{x_3 x_2 x_1} + \overline{x_3 x_1 x_0} + x_5$$

$$\text{e) } y = x_3 x_0 + x_1$$

$$\text{f) } y = x_2 \overline{x_1} + x_3 x_4$$

$$\text{g) } y = x_5 \overline{x_2 x_1} + x_5 \overline{x_4 x_2} + x_5 \overline{x_2 x_0}$$

$$\text{h) } y = (x_3 x_2 + x_1) + (\overline{x_0} + x_1)$$

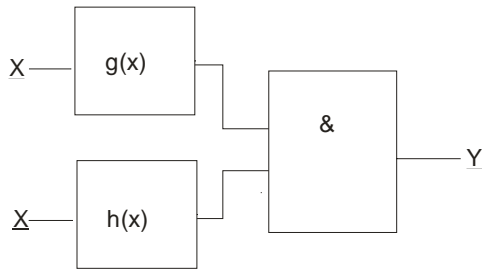
$$\text{i) } y = (x_2 x_1 + 1) + ((x_4 + \overline{x_1}) + \overline{x_0})$$

$$\text{j) } y = x_2 \overline{x_1} + x_3 x_0 + x_1$$

2 Minimierung von Schaltfunktionen

2.1 Minimierung mit Nebenbedingungen

Minimieren Sie $g(x)$ unter Beachtung der Tatsache, dass für bestimmte Eingangsbelegungen x_e der Ausgang $g(x)$ nicht abgefragt wird.



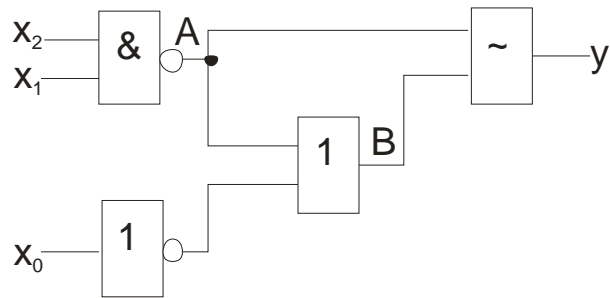
- a) $g(x) = x_2 x_1 + x_3 x_2 x_0 + x_3 x_0$
 $h(x) = x_1 x_0$
- b) $g(x) = \overline{x_4 x_3 x_1} + \overline{x_4 x_3 x_2 x_0} + x_4 x_3 x_2 x_0 + x_3 x_2 x_1 x_0$
 $h(x) = \overline{x_4 x_3} + x_4 x_0 + x_3 x_1 x_0$
- c) $g(x) = x_4 x_1$
 $h(x) = x_4 x_1$

2.2 Minimierung aus dem Karnaugh-Plan

- a) $f(x) = \sum \{0, 1, 3, 7, 12, 14\}, k = 4$
- b) $f(x) = \sum \{2, 3, 4, 5, 6, 7, 11, 12, 13, 14, 15\}, k = 4$
- c) $f(x) = \sum \{0, 1, 2, 3, 4, 5, 10, 11, 12, 13, 14, 15\}, k = 4$
- d) $f(x) = \sum \{0, 1, 2, 3, 4, 8, 12\}, k = 4$
- e) $f(x) = \sum \{0, 11, 12\}, k = 4$
 $D = \{4, 5, 13, 15\}$
- f) $f(x) = \sum \{0, 3, 4, 7, 8, 12, 14, 16, 20, 22, 24, 28\}, k = 5$
 $D = \{6, 10, 11, 15, 19, 23, 26, 27, 30, 31\}$

g) $f(x) = \sum \{0, 3, 4, 6, 8, 11, 12, 14, 16, 19, 20, 22, 24, 27, 28, 30\}, k = 5$
 $D = \{10, 26\}$

2.3 Ermittlung der Schaltfunktion aus einer gegebenen Schaltung



2.4 Verfahren nach Quine/McCluskey und Tafelauswahlverfahren

- a) $f(x) = \prod 1, 4, 6, \quad k = 3$
- b) $f(x) = \prod 1, 2, 6, 7, 9, 10, 14 \quad k = 4$
- c) $f(x) = \prod 5, 10, 12, 13, 14, \quad k = 4$
- d) $f(x) = \sum 2, 3, 4, 5, 6, 7, \quad k = 3$
- e) $f(x) = \sum 0, 2, 3, 5, 7, \quad k = 3$
- f) $f(x) = \sum 0, 1, 2, 3, 4, 5, 10, 11, 12, 13, 14, 15, \quad k = 4$

2.5 Faktorisierung aus dem Karnaugh-Plan

Entwickeln Sie je drei mehrstufige NAND- und NOR- Schaltungen für:

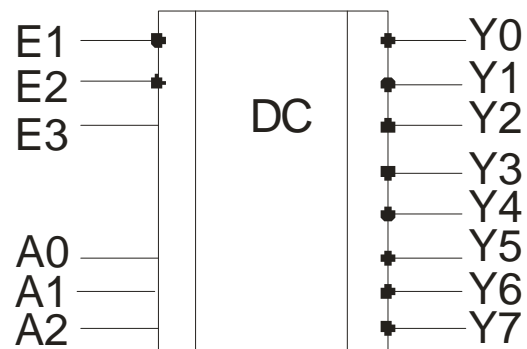
$f(x) = \sum 0, 2, 4, 5, 7, \quad k = 4$

3. Kombinatorische Grundschaltungen

3.1 Multiplexer und Demultiplexer

- a) Entwerfen Sie einen 4 auf 1 Multiplexer mit Binäradresse und Tristate- Ausgang.
- Vollständige Wahrheitstafel
 - Entwicklung der Schaltung im NAND-Basissystem
 - Schaltungen für die MOS-Treiberstufen
- b) Realisieren Sie unter Verwendung eines 8205 einen 8 auf 1 Multiplexer, der die in der Tafel dargestellte Funktion realisiert.

STB	a_2	a_1	a_0	y	\overline{y}
0	d	d	d	0	1
1	0	0	0	x_0	$\overline{x_0}$
1	0	0	1	x_1	$\overline{x_1}$
1	0	1	0	x_2	$\overline{x_2}$
1	0	1	1	x_3	$\overline{x_3}$
1	1	0	0	x_4	$\overline{x_4}$
1	1	0	1	x_5	$\overline{x_5}$
1	1	1	0	x_6	$\overline{x_6}$
1	1	1	1	x_7	$\overline{x_7}$



- c) Entwerfen Sie einen 32 auf 1 Multiplexer unter Verwendung von 4 auf 1 Multiplexern und 1 aus 8 Dekodern
- d) Gegeben sei die Schaltfunktion $y = x_1x_0 + x_3x_2x_1 + x_3x_2x_1x_0$
- Realisieren Sie die Schaltfunktion durch einen 16 auf 1 Multiplexer
 - Realisieren Sie die Schaltfunktion durch einen 8 auf 1 Multiplexer
 - Realisieren Sie die Schaltfunktion durch einen 4 auf 1 Multiplexer
- e) Realisieren Sie $f(x) = \sum 0, 2, 3, 4, 8, 9, 13, 17, 21, 25, 29, 31$, $k = 5$ mittels 4 auf 1 Multiplexern.
- f) Realisieren Sie unter Verwendung von Multiplexern einen byteweise organisierten 64bit-ROM. Erläutern Sie dessen Einsatz als Codewandler.
- g) Entwerfen Sie die Schaltung eines 2 x 8-Bit-Busumschalters mit Tristateausgängen auf Gattenniveau.

3.2 Dekodierschaltungen

- a) Entwerfen Sie einen 1 aus 4 Dekoder, der binär angesteuert wird.
 - Wie können Sie die Schaltung erweitern, so dass wahlweise negierte Ausgangssignale auftreten?
 - Wie können Sie die Ausgänge wahlweise sperren?
- b) Entwerfen Sie einen BCD in 3-Excesscode-Dekoder.
- c) Entwerfen Sie einen BCD in Aikencode-Dekoder.
- d) Entwerfen Sie einen BCD in 7-Segment-Dekoder (ohne Segment h).
- e) Entwerfen Sie einen Gray in BCD-Dekoder
 - auf Gatterniveau
 - als PLA
 - mit Antivalenzgattern
- f) Entwerfen Sie einen BDC in Gray-Dekoder
 - auf Gatterniveau
 - mit Antivalenzgattern
- g) Entwerfen Sie einen Aiken in Gray-Dekoder.

3.3 Addierer

- a) Geben Sie die NAND-Realisierung und ein PLA-Schema für einen 1-Bit-Volladdierer an.
- b) Entwerfen Sie aus 1-Bit-Volladdierern eine Schaltung, die zwei 4-Bit –Zahlen addiert. Die Schaltung ist so zu erweitern, dass die Summe zweier BCD-Zahlen (0-9) gebildet wird.
- c) Entwerfen Sie einen Paritätsgenerator für 4 Variable
 - auf Gatterniveau
 - mit Halbaddierern

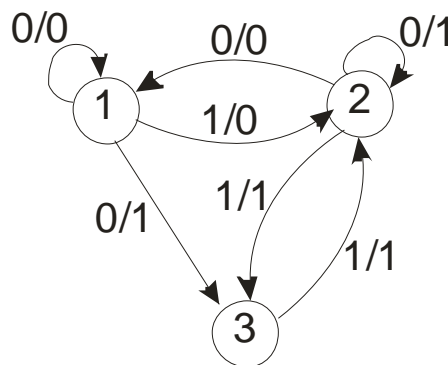
3.4 Digitalkomparatoren

- a) Entwerfen Sie einen Komparator (low- aktiv) zum Vergleich zweistelliger Variabler
 - auf Gatterniveau
 - mit einstelligen Komparatoren
- b) Ein Messwertgeber liefert zwei zweistellige Ausgangssignale. Zur Auswertung sind beide Signale zu addieren. Bei einer Summer größer als 4 ist ein Signal abzugeben. Alle Adress- und Datenleitungen sind binär zu kodieren.
 - Entwerfen Sie das Schaltbild, wenn nacheinander 4 Messstellen abgefragt werden.
Bauelemente: (4 auf 1 MUX, 1 Bit-Volladdierer, 2 x 4-Bit-Komparator)
 - Entwerfen Sie die zur Erfüllung der Komparatorfunktion notwendige Schaltung.

4 Sequentielle Schaltungen

4.1 Grundlagen der Automatentheorie

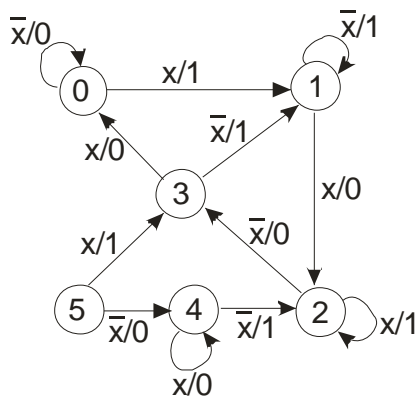
- Entwerfen Sie ein dreistufiges Schieberegister, das aus dem Initialzustand $Z = (0,0,0)$ in drei Takten mit „1“ gefüllt wird!
 - Automatengraph
 - Folgezustandstabelle
 - Schaltung
- Entwerfen Sie ein dreistufiges Schieberegister, das aus dem Initialzustand $Z = (0,0,0)$ abwechselnd mit „0“ und „1“ gefüllt wird!
 - Automatengraph
 - Folgezustandstabelle
 - Schaltung
 - Untersuchung auf parasitäre Zyklen
- Geben Sie den vollständigen Automatengraphen für ein dreistufiges Schieberegister an, das aus einem beliebigen Anfangszustand heraus mit „1“ gefüllt wird!
- Wandeln Sie den gegebenen MEALY-Automaten in den entsprechenden MOORE-Automaten um und zeichnen Sie den Automatengraphen.



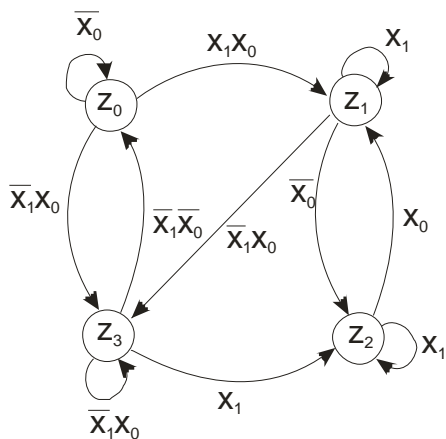
- Wandel Sie den gegebenen MEALY-Automaten in den entsprechenden MOORE-Automaten um und zeichnen Sie beide Automatengraphen.

	0	1	2	3	
	0	0	1	1	x_1
Z_μ	0	1	0	1	x_0
0	0/0	0/0	3/1	1/1	
1	2/1	2/1	1/1	1/1	
2	2/1	2/1	1/1	1/1	
3	1/0	1/0	2/0	3/1	

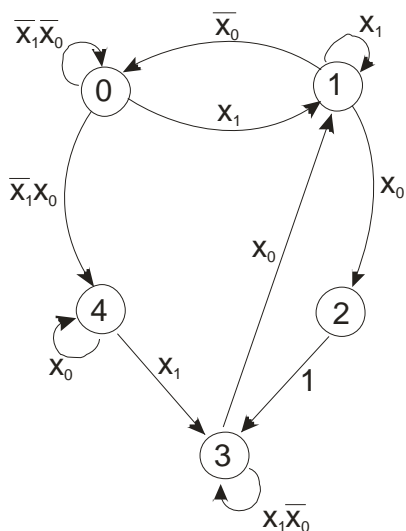
- f) Wandeln Sie den gegebenen MEALY-Automaten in den entsprechenden MOORE-Automaten um und zeichnen Sie den Automatengraphen.



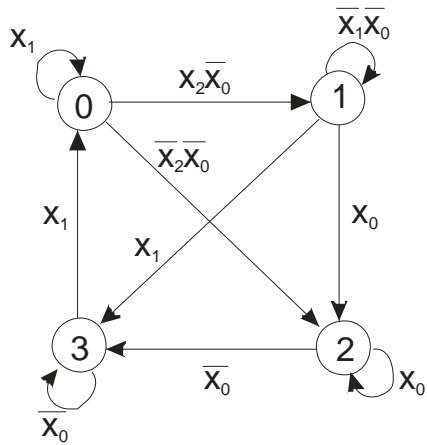
- g) Überprüfen Sie die Vollständigkeit, Widerspruchsfreiheit und Stabilität des folgenden Automaten.



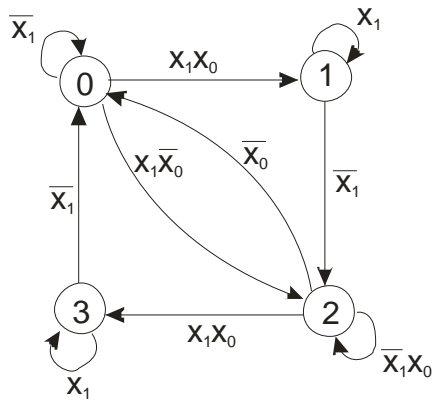
- h) Überprüfen Sie Vollständigkeit, Widerspruchsfreiheit und Stabilität des folgenden Automaten.



- i) Überprüfen Sie Vollständigkeit, Widerspruchsfreiheit und Stabilität des folgenden Automaten.

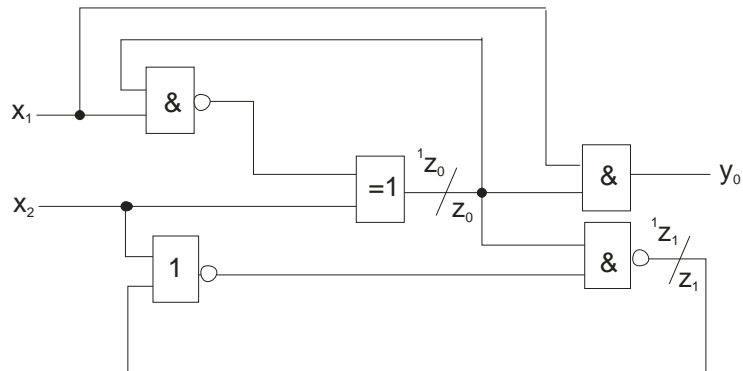


- j) Überprüfen Sie die Stabilität des folgenden Automaten.

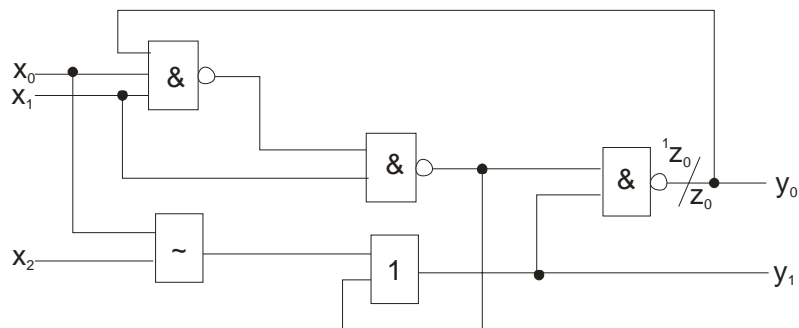


4.2 Analyse sequentieller Schaltungen- Schnittmethode

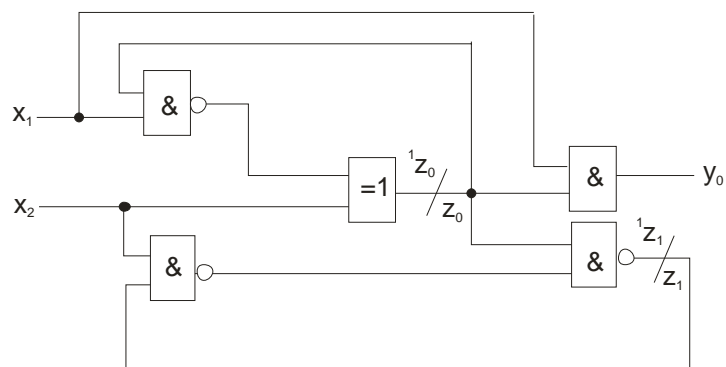
a)



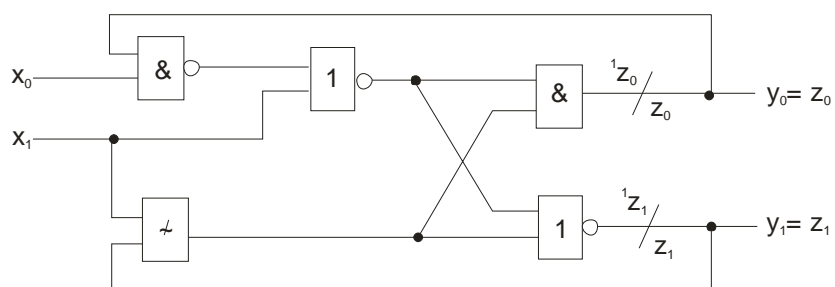
b)



c)



d)



- Ermitteln Sie aus den gegebenen Schaltungen Z_1, Z_0 und y_0 bzw. y_1 und stellen Sie die Zustandstabelle sowie den dazugehörigen Zustandsgraphen auf.

- Überprüfen Sie an Hand des Zustandsgraphen, ob die Schaltung stabil ist und begründen Sie ihre Aussage.

4.3 Flip-Flops

a) Klären Sie folgende Fragen:

- astabile Kippschaltung
- monostabile Kippschaltung
- bistabile Kippschaltung
- kombinatorische Kippschaltung
- sequentielle Schaltung
- Welchen Kategorien gehören Flip-Flops an?

b) RS-Flip-Flop

- Geben Sie die Schaltung für ein RS-Flip-Flop an, das aus NAND-Gattern aufgebaut ist.
- Stellen Sie die Funktionstabelle für dieses Flip-Flop auf. Benutzen Sie die Symbole $R_n, S_n, Q_n, Q_{n+1}, \overline{Q_{n+1}}$
- Zeichnen Sie den reduzierten und vollständigen Folgezustandsgraph.
- Warum ist die Eingangsbelegung $R = S = 0$ verboten? Was verstehen Sie unter dem Speicherzustand?
- Erweitern Sie Ihre Schaltung so, dass Sie mit einem Signal festlegen können, wann Ihr Flip-Flop schalten kann, und klären Sie in diesem Zusammenhang die Begriffe „asynchrones System“ und synchrones System“.
- Zeichnen Sie die Taktdiagramme für verschiedene Taktungsarten: einstufiges positiv-taktzustandsgetriggertes RS-FF, einstufiges negativ-taktzustandsgetriggertes RS-FF, positiv-taktzustandsgetriggertes RS-MS-FF, negativ-taktzustandsgetriggertes RS-MS-FF.
- Entwerfen Sie mit Hilfe eines RS-FF's auf Gatterniveau eine Entprellschaltung! Geben Sie die Schaltungen für NAND- und NOR-Basissysteme an!

c) Master-Slave Flip-Flop

- Geben Sie das Prinzipschaltbild eines MS-Flip-Flops (MS-FF) an.
- Geben Sie eine Schaltung für ein JK-MS-FF an.

d) Umwandlung von Flip-Flops

Wandeln Sie die folgenden Flip-Flop-Typen ineinander um.

- RS-FF in T-FF
- T-FF in JK-FF
- JK-FF in RS-FF
- RS-FF in D-FF
- D-FF in RS-FF

4.4 Zähler und Teiler

- a) Nach welchen Kriterien werden Zähler unterschieden? Was sind die wesentlichen Vor- und Nachteile verschiedener Zählerarten?
- b) Wie viele Flip-Flops benötigt man für einen Zähler modulo n ?
- c) Entwerfen Sie unter Verwendung von JK-MS-FF einen synchronen zyklischen BCD-Vorwärtszähler von 0 bis 7.
- d) Entwerfen Sie unter Verwendung von JK-MS-FF einen synchronen zyklischen Vorwärts-/Rückwärtszähler von 0 bis 15 (Binärcode).
- e) Entwerfen Sie unter Verwendung von RS-MS-FF einen synchronen zyklischen Binärvorwärtszähler, der abwechselnd von 0 bis 3 und von 0 bis 5 zählt.
- f) Entwerfen Sie unter Verwendung von JK-MS-FF einen synchronen zyklischen BCD-Vorwärtszähler, der wahlweise von 0 bis 3 und von 0 bis 5 zählt (Abfrage des Signals zum Zählerumfang nur am Zyklusende).
- g) Entwerfen Sie unter Verwendung von JK-MS-FF einen asynchronen Vorwärtszähler von 0 bis 3 (binär).
- h) Welche Möglichkeiten bestehen, den in Aufgabe g) entworfenen Zähler in einen Rückwärtszähler umzuwandeln?
- i) Entwerfen Sie unter Verwendung von JK-MS-FF einen asynchronen Vorwärts-/Rückwärtszähler von 0 bis 3 (binär). Die Zählrichtung soll über ein externes Signal R bestimmt werden ($R = 1 \rightarrow$ rückwärts)
- j) Entwerfen Sie unter Verwendung von JK-MS-FF einen asynchronen BCD-Zähler von 0 bis 8. Überprüfen Sie die Initialisierung und versehen Sie ihn gegebenenfalls mit einem RESET.
- k) Entwerfen Sie einen asynchronen Zähler von 0 bis 9 im Aiken-Code. Verwenden Sie negativ-taktflankengetriggerte JK-MS-FF's.
- l) Entwerfen Sie unter Verwendung von negativ-taktflankengetriggerten JK-MS-FF einen asynchronen zyklischen Binärrückwärtszähler von 6 bis 3.
- m) Entwerfen Sie unter Verwendung von 4 negativ-taktflankengetriggerten JK-MS-FF einen asynchronen zyklischen Binärvorwärtszähler von 7 bis 10.
- n) Entwerfen Sie unter Verwendung von negativ-taktflankengetriggerten JK-MS-FF einen asynchronen zyklischen Binärvorwärtszähler von 7 bis 10. Minimieren Sie dabei die Anzahl der FF's. Führen Sie ggf. eine zusätzliche Kombinatorik ein.
- o) Realisieren Sie mit einem integrierten 4-Bit Binärzähler einen BCD-Rückwärtszähler von 7 bis 3.
- p) Realisieren Sie mit einem integrierten BCD-Zähler einen BCD-Vorwärtszähler von 2 bis 5.

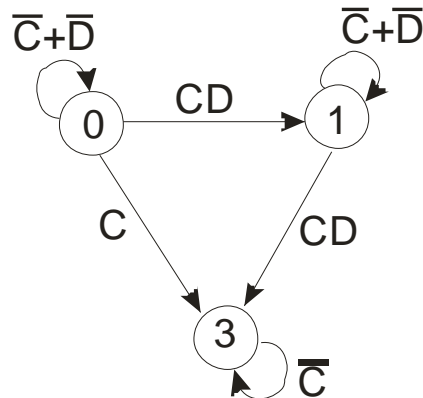
4.5 Impulsfolgeerkennung

- a) Entwerfen Sie einen Moore-Automaten, der nach der vollständigen Erkennung der Impulsfolge „11010111“ eine „1“ ausgibt (sonst „0“). Welche Änderungen wären bei einer Notation als Mealy-Automat erforderlich?

- b) Entwerfen Sie einen Moore-Automaten, der nach der vollständigen Erkennung der Impulsfolge „1120312“ eine „1“ ausgibt (sonst „0“). Nennen Sie mögliche Anwendungsgebiete eines solchen Automaten!

4.6 Realisierung beliebiger Automaten

- a) Entwerfen Sie mit D-FF eine Schaltung, die folgenden Automaten realisiert:



4.7 Schieberegister

- a) Welche FF-Typen können zur Realisierung von SR genutzt werden? Ist ein asynchroner Aufbau möglich (Begründung)?
- b) Entwerfen Sie ein rechtsschiebendes 4-Bit-Schieberegister mit serieller Ein- und Ausgabe. Nutzen Sie JK-MS-FF's.
- Erweitern Sie die Schaltung so, dass die Schieberichtung umschaltbar ist und dass parallel ausgelesen werden kann.
 - Können die JK-MS-FF durch taktflankengesteuerte D-FF ersetzt werden?
 - Schalten Sie das Schieberegister als Ringzähler.