Prüfung

Prof. Dr.-Ing. J. Becker

Digitaltechnik

SS 2006

Institut für Technik der Informationsverarbeitung, Universität Karlsruhe

Klausur

Mi., 04.10.2006

Aufgabenblätter

1
2
3
4
5
6
7
Σ

Hinweise zur Klausur

Hilfsmittel

Als Hilfsmittel zur Prüfung sind drei Seiten vorgegebene und zwei Seiten selbst geschriebene Formelsammlung zugelassen. Nicht erlaubt hingegen ist die Verwendung eines Taschenrechners, zusätzliche Unterlagen und jegliche Kommunikation mit anderen Personen.

Prüfungsdauer

Die Prüfungsdauer beträgt 120 Minuten.

Prüfungsunterlagen

Die Prüfungsunterlagen bestehen aus insgesamt 29 Seiten (einschließlich diesem Titelblatt).

Bitte vermerken Sie vor der Bearbeitung der Aufgaben auf jeder Seite oben Ihren Namen, auf der ersten Seite zusätzlich die Matrikelnummer!

Auf jedes zusätzliche Lösungsblatt ist neben dem Namen auch die Aufgaben- und die Seitennummer mit einzutragen. Vermeiden Sie das Beschreiben der Rückseiten.

Am Ende der Prüfung sind die 29 Seiten Aufgaben- und Lösungsblätter und alle verwendeten zusätzlichen Lösungsblätter abzugeben.

Aufgabe 1 Allgemeines

Aufgabe 1.1 Allgemeine Fragen

Beantworten Sie folgende Fragen:

A) Die Bitbreite des Datenpfades eines Prozessors – hierzu zählen Addierer, Multiplizierer etc. - beträgt heutzutage nicht selten 128 Bit.

I) Geben Sie den Wertebereich der Zahl *x* im Dezimalsystem an, welcher sich mit der oben genannten Bitbreite darstellen lässt, unter der Annahme, dass es sich um vorzeichenlose Integer Werte handelt.

$$0 \leq x \leq 2^{128} - 1$$

II) Geben Sie nun den Wertebereich der Zahl im Dezimalsystem unter der Annahme an, dass es sich um vorzeichenbehaftete Integer Werte in der Zweierkomplement Darstellung handelt.

$$-2^{128-1} \leq x \leq 2^{128-1} - 1$$

B) Erklären Sie, aus welchen Gründen bei der BCD Addition eine Korrektur des Ergebnisses durchgeführt werden muss. Geben Sie ferner den bei der Korrektur verwendeten Summand an und erklären Sie, weshalb kein anderer Summand hierfür in Frage kommt.

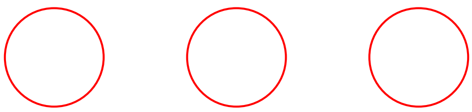
Für die Darstellung der Ziffern 0_D bis 9_D werden im Dualsystem 4 Bit benötigt. Mit 4 Bit wiederum lassen sich jedoch 16 unterschiedliche Kombinationen darstellen, weshalb sechs dieser Kombinationen ungenutzt bleiben. Diese werden dann auch als Pseudotetraden bezeichnet.

Bei der BCD-Addition zweier dezimaler Ziffern kann es nun vorkommen, dass das Ergebnis genau eine der sechs nicht erlaubten Kombinationen einnimmt. Um nun eine korrekte Überführung des Ergebnisses in das Dezimalsystem zu ermöglichen, muss das Ergebnis anhand einer weiteren Addition von 6_D korrigiert werden. Die Korrektur lässt sich anschaulich damit begründen, dass der Bereich der Pseudotetraden „übersprungen" werden muss, da diesen im Dezimalsystem keine Ziffer zugeordnet ist – im Dezimalsystem liegt hier bereits in Übertrag in die nächst höhere Stelle vor. Da dieser Bereich exakt 6 der 16 Kombination umfasst muss also mit der Ziffer 6_D korrigiert werden.

Sollte es gar zu einem Übertrag bei der BCD-Addition zweier dezimaler Ziffern kommen, so muss auch hier das Ergebnis korrigiert werden. Ähnlich wie zu vor erfolgt auch hier die Korrektur durch Addition von 6_D zur niederwertigsten BCD Zahl. Anschaulich lässt sich dies damit begründen, dass in der Darstellung des unkorrigierten BCD Ergebnisses der Bereich der Pseudotetraden enthalten ist. Da diese jedoch über keine Darstellung im Dezimalbereich verfügen muss auch hier dafür gesorgt werden, dass der Bereich nach der Korrektur nicht länger enthalten ist. Dies wird genau durch die Addition von 6_D bezweckt.

C) Wann nenn man einen Graphen entartet? Geben Sie ein Beispiel basierend auf exakt drei Knoten.

Einen Graphen nennt man entartet, wenn der Graph keine Kanten enthält, die Menge der Kanten also gleich der Leeren Menge ist.



Aufgabe 1.2 Boolesche Algebra

A) Zeigen oder widerlegen Sie die Gleichheit der nachfolgend angegebenen Booleschen Ausdrücke. Geben Sie für jeden durchgeführten Umformungsschritt die verwendete Regel an.

$$a \vee b \quad \Leftrightarrow \quad a \neq (b \neq (a \,\&\, b)) \qquad (\neq \text{ entspricht Antivalenz })$$

$a \neq (b \,\&\, \overline{(a \,\&\, b)} \vee \bar{b} \,\&\, (a \,\&\, b))$	Antivalenz aufgelöst
$a \neq (b \,\&\, \overline{(a \,\&\, b)} \vee \bar{b} \,\&\, a \,\&\, b)$	R10b
$a \neq (b \,\&\, \overline{(a \,\&\, b)})$	R8b, R6b
$a \neq (b \,\&\, (\bar{a} \vee \bar{b}))$	R12b
$a \neq (b \,\&\, \bar{a} \vee b \,\&\, \bar{b})$	H3
$a \neq (b \,\&\, \bar{a})$	R12b
$a \,\&\, \overline{(b \,\&\, \bar{a})} \vee \bar{a} \,\&\, (b \,\&\, \bar{a})$	Antivalenz aufgelöst
$a \,\&\, \overline{(b \,\&\, \bar{a})} \vee \bar{a} \,\&\, b$	R7b
$a \,\&\, (\bar{b} \vee a) \vee \bar{a} \,\&\, b$	R12b
$a \vee \bar{a} \,\&\, b$	R11b
$(a \vee \bar{a}) \,\&\, (a \vee b)$	H3
$a \vee b$	R8a, R5b

B) Die gegebene Schaltfunktion y = f(d, c, b, a) soll mit 2:1 Multiplexern realisiert werden. Dazu muß die Funktion nach jeder Variablen mit Hilfe des Entwicklungssatzes entwickelt werden.

$$y = \bar{a}\,\bar{c} \vee b \vee \bar{d}\,\bar{c} \vee a\,d\,c$$

Entwickeln Sie die Schaltfunktion nach der Variablen b. Geben Sie alle Zwischenschritte an.

$$y = \bar{b} \,\&\, (\bar{a}\,\bar{c} \vee \bar{d}\,\bar{c} \vee a\,d\,c) \,V\, b \,\&\, 1$$

$$= \bar{b} \,\&\, F_{Rest1} \vee b \,\&\, F_{Rest2}$$

Entwickelte Schaltfunktion:

$$y = \bar{b} \,\&\, (\bar{c}\,\bar{a} \vee \bar{c}\,\bar{d} \vee a\,d\,c) \vee b$$

Entwickeln Sie die Restfunktionen zuerst nach der Variablen c und dann, falls erforderlich, nach den verbleibenden Variablen, so daß als Restfunktionen nur noch Konstanten übrig bleiben. Geben Sie alle Zwischenschritte an.

$$F_{Rest1} = \bar{c}\,\bar{a} \vee \bar{c}\,\bar{d} \vee a\,d\,c$$

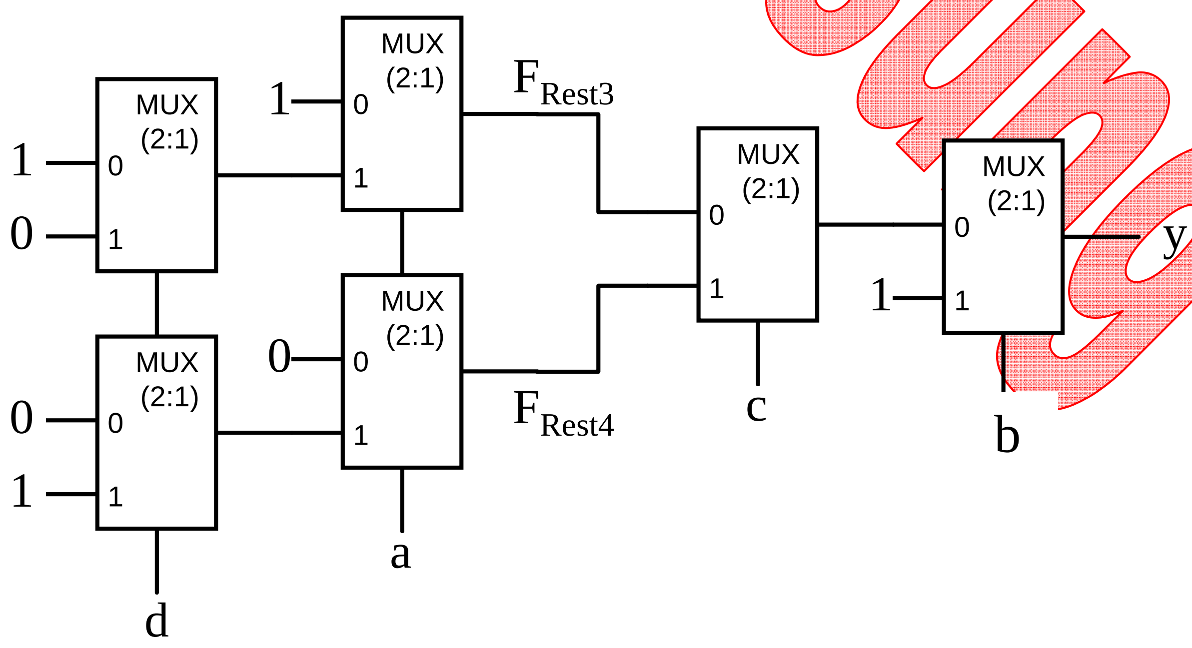
$$= \bar{c}\,(\bar{a} \vee \bar{d}) \vee c\,(a\,d) = \bar{c} \,\&\, F_{Rest3} \vee c \,\&\, F_{Rest4}$$

$$F_{Rest3} = \bar{a} \,\&\, 1 \vee a \,\&\, \bar{d} \qquad F_{Rest4} = \bar{a} \,\&\, 0 \vee a \,\&\, d$$

Entwickelte Restfunktion:

$$F_{Rest1} = \bar{c}\,(\bar{a} \,\&\, 1 \vee a \,\&\, \bar{d}) \vee c\,(\bar{a} \,\&\, 0 \vee a \,\&\, d)$$

C) Zeichen Sie die gesamte Schaltung unter ausschließlicher Verwendung von 2:1 Multiplexern.



Aufgabe 2 Minimierung

Aufgabe 2.1 Verfahren nach Nelson

Für eine unvollständig definierte Schaltfunktion ***G*** sei die Menge der Nullstellen (N) und die Menge der Freistellen (F) in **oktaler** Indizierung wie folgt gegeben:

N={0,1,2,3,4,12,16}
F={13,14}

Mit Hilfe des Nelson-Verfahrens sollen nun alle Primimplikanten der Funktion ermittelt werden.

A) Tragen Sie hierzu zunächst die Eins-, Null- und Freistellen in folgendes Symmetriediagramm ein.

x_1

	0_{0}	0_{1}	1_{5}	0_{4}
x_2	0_{2}	0_{3}	1_{7}	1_{6}
	0_{12}	$-_{13}$	1_{17}	0_{16}
	1_{10}	1_{11}	1_{15}	$-_{14}$

x_4

x_3

B) Bilden Sie die Nullblocküberdeckung τ_0 der Funktion ***G***. (Freistellen werden hierzu nicht genutzt)

$$\tau_0 = \{(0,0,-,-),(1,-,1,0),(0,-,0,0)\}$$

C) Bilden Sie nun die Einsvervollständigung g^E:

$$g^E = (x_4 + x_3)(\overline{x_4} + \overline{x_2} + x_1)(x_4 + x_2 + x_1)$$

D) Distribuieren Sie nun schrittweise den in Teil C) gefundenen Ausdruck aus. Formen Sie dabei geeignet um und streichen Sie alle redundanten Terme bzw. Termanteile. Geben Sie anschließend alle gefundenen Primimplikanten an. Verwendete Umformungsregeln müssen nicht angegeben werden.

$$g^E = (x_4 + x_3)(\overline{x_4} + \overline{x_2} + x_1)(x_4 + x_2 + x_1)$$

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

$$= x_4 \cancel{\overline{x_4} x_2} + x_4 \overline{x_2} \cancel{x_4} + x_4 x_1 + x_3 \overline{x_4} x_2 + \cancel{x_3 \overline{x_2} x_4} + x_3 x_1$$

$$= \underline{x_4 \overline{x_2} + x_4 x_1 + x_3 \overline{x_4} x_2 + x_3 x_1}$$

Aufgabe 2.2 Verfahren nach Petrick

Gegeben sei folgendes Symmetriediagramm der Schaltfunktion ***G***:

		a	
- 0	1 1	0 5	- 4
0 2	1 3	1 7	0 6
1 12	0 13	- 17	0 16
- 10	- 11	1 15	1 14

b, c, d

A) Das Nelson-Verfahren lieferte dabei die in der Tabelle 1 bereits eingetragenen Primterme. Vervollständigen Sie nun die folgende Überdeckungstabelle. Bilden Sie die Kostenfunktionswerte für gegebene Primterme, indem Sie die Variablen b und c mit „**1**" und die Variablen a und d mit „**2**" gewichten.

Präsenz-variablen	Primterme	Einsstellen (oktale Indizes)						Kosten
		1	3	7	12	14	15	
p_1	$\bar{b}\bar{c}$	X						2
~~p_2~~	~~$\bar{a}\bar{b}$~~					~~X~~		~~3~~
p_3	$a\bar{c}\bar{d}$	X	X					5
p_4	$ab\bar{d}$		X	X				5
p_5	abc			X				4
p_6	$\bar{b}d$					X	X	3
~~p_7~~	~~acd~~						~~X~~	~~5~~
p_8	$\bar{a}\bar{c}d$				X			5

Tabelle 1

B) Bestimmen Sie nun mögliche Kerne und ermitteln Sie anschließend durch die Ausnutzung der Zeilen- und der Spaltendominanzen, soweit anwendbar, die Resttabelle. Kennzeichnen Sie hierbei die verbliebenen Minterme.

C) Tragen Sie nun die im Aufgabenteil B) ermittelte Resttabelle in die Tabelle 2 ein (ordnen Sie dabei die verbleibenden oktalen Indizes wiederum aufsteigend an).

Präsenz-variablen	Primterme	Einsstellen (oktale Indizes)					Kosten
		1	3	7			
p_1	$\overline{b}\,\overline{c}$	X					2
p_3	$a\overline{c}\,\overline{d}$	X	X				5
p_4	$ab\overline{d}$		X	X			5
p_5	abc			X			4

Tabelle 2

D) Lösen Sie nun die Resttabelle mittels des Petrickausdrucks. Die Kosten der Primimplikanten sollen weiterhin als Entscheidungskriterium genutzt werden, um sich für eine Minimallösung zu entscheiden. Geben Sie die Lösung an.

$$
\begin{aligned}
PA' &= (p_1 + p_3)(p_3 + p_4)(p_4 + p_5) \\
&= (p_1p_3 + p_1p_4 + p_3p_3 + p_3p_4)(p_4 + p_5) \\
&= (p_1p_4 + p_3)(p_4 + p_5) \\
&= (p_1p_4p_4 + p_1p_4p_5 + p_3p_4 + p_3p_5) \\
&= \underline{p_1p_4 + p_3p_4 + p_3p_5}
\end{aligned}
$$

Die Realisierung der Resttabelle durch $\underline{p_1p_4}$ stellt die Minimallösungen dar.

E) Geben Sie nun die zur Realisierung benötigten Präsenzvariablen, die Kosten und die DMF der minimierten Funktion an.

benötigte Präsenzvariablen: p_6, p_8, p_1, p_4

Kosten der Realisierung: $3+5+2+5=15$

zugehörige DMF: $\overline{b}d + \overline{a}\,\overline{c}d + \overline{b}\,\overline{c} + ab\overline{d}$

Aufgabe 3 Zahlensysteme

A) Vervollständigen Sie die Tabelle 3, indem Sie die offenen Felder durch Konvertierung ergänzen. BCD-Zahlen sollen hierbei stets aus der Dezimaldarstellung abgeleitet werden.

Dezimal	Binär	Oktal	BCD
42_D	101010_B	52_O	0100 0010
782_D	1100001110_B	1416_O	0111 1000 0010
195_D	11000011_B	303_O	0001 1001 0101
2758_D	101011000110_B	5306_O	0010 0111 0101 1000

Tabelle 3

B) Addieren Sie die im Dezimalsystem gegebenen Zahlen 4862_D und 975_D im BCD Code. Stellen Sie ihren Lösungsweg – inklusive eventuell notwendiger Korrekturschritte - ausführlich dar.

BCD Code				Dezimalsystem
0100	1000	0110	0010	4862
0000	1001	0111	0101	975
1				
0101	0001	1101	0111	
	Übertrag	PT		
	0110	0110		
	1			
0101	1000	0011	0111	5837

Aufgabe 4 Optimale Codes

In Abbildung 1 sind die drei Räder eines einarmigen Banditen dargestellt. Jedes der drei Räder besteht aus acht gleich großen Sektoren, und beinhaltet eine unterschiedliche Anzahl der folgenden Symbole:

Symbol	Abkürzung
Raute	R
Kreuz	K
Kreis	KS
Dreieck	D
Quadrat	Q

Am Ende eines Spiels ist im Anzeigefenster des Spielautomaten immer genau einer der Sektoren pro Rad sichtbar, wobei jeder Sektor mit gleicher Wahrscheinlichkeit auftritt. Die Räder selbst drehen sich unabhängig voneinander, d.h. jede Sektorkombination der drei Räder besitzt die gleiche Auftrittswahrscheinlichkeit.

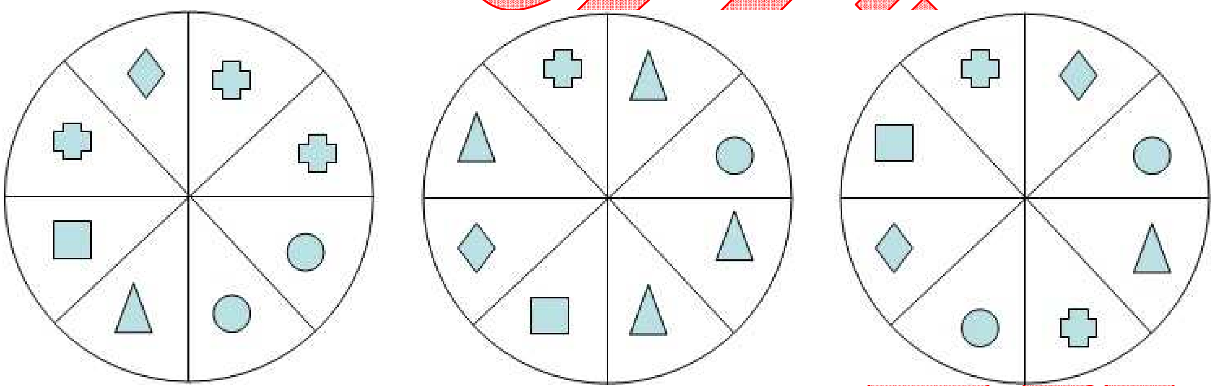


Abbildung 1

A) Berechnen Sie die Gewinnwahrscheinlichkeiten für die einzelnen Symbole, wenn zum Gewinnen auf jeder Scheibe das gleiche Symbol gezeigt werden muss.

Symbol	Rechnung	Gewinnwahrscheinlichkeit
KS	1/4*1/8*1/4	1/128
K	3/8*1/8*1/4	3/256
Q	1/8*1/8*1/8	1/512
D	1/8*1/2*1/8	1/128
R	1/8*1/8*1/4	1/256

B) Berechnen Sie nun die Wahrscheinlichkeit zu verlieren.

Die Wahrscheinlichkeit zu verlieren berechnet sich zu:

$$\begin{aligned} p_{verlieren} &= 1 - p_{gewinnen} \\ &= 1 - 1/128 - 3/256 - 1/512 - 1/128 - 1/256 \\ &= 1 - (4+6+1+4+2)/512 \\ &= 1 - 17/512 = 495/512 \end{aligned}$$

Da der Automat nur in größeren Zeitabständen vom Besitzer überprüft wird, sollen die bis dato gespielten Spiele nachvollziehbar und möglichst komprimiert gespeichert werden. Hierbei sollen sowohl die einzelnen Gewinne als auch verlorene Spiele mit einbezogen werden.

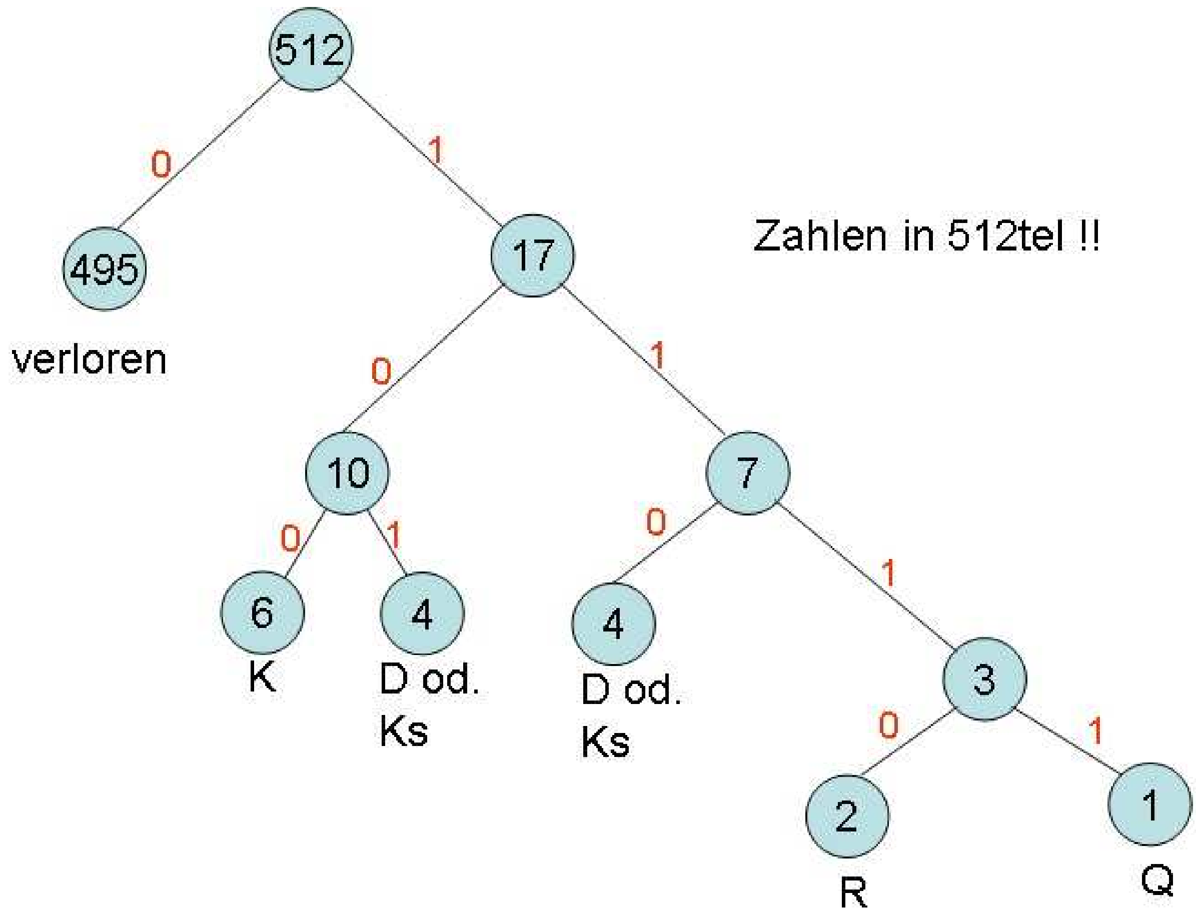
C) Entwickeln Sie hierzu eine Shannon-Fano Codierung. Ordnen Sie hierfür zunächst die oben ermittelten Wahrscheinlichkeiten in sinkender Reichenfolge von links nach rechts sortiert an. Zeichnen Sie anschließend, basierend auf dieser Ordnung, den Codebaum. Versehen Sie dabei die nach links gerichteten Äste mit „0“, die nach rechts gerichteten mit „1“.

495/512 | 6/512 + 4/512 + 4/512 +2/512+1/512

6/512 + 4/512 | 4/512+2/512+1/512

4/512 | 2/512 + 1/512

2/512 | 1/512



D) Ermitteln Sie die mittlere Codewortlänge der Shannon-Fano Codierung. Geben Sie ferner an, welche Anzahl an Bits für die Speicherung eines Ereignisses benötigt würde, wenn die Information lediglich unkomprimiert binär codiert gespeichert werden müsste?

$$\overline{m} = (1*495/512+3*6/512+3*4/512+3*4/512+4*2/512+4*1/512)$$
$$= (495+18+12+12+8+4)/512$$
$$= 549/512$$

Uncodiert: m=3

$$m = ld\lceil 6 \rceil = 3$$

E) Welche Eigenschaft der Shannon Codierung erlaubt die eindeutige Decodierung eines Shannon Codes?

Die Präfixfreiheit

Aufgabe 5 Fehlererkennung & Korrektur

Im Folgenden sei ein 5 Bit breiter Code gegeben .

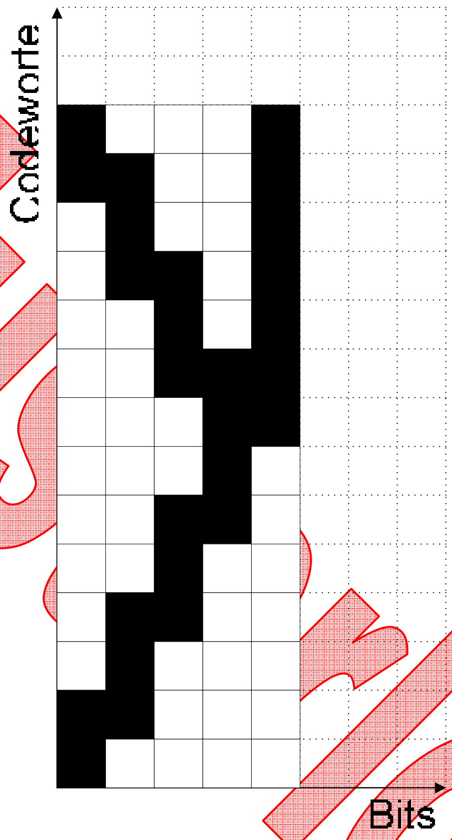


Abbildung 2

A) Nennen Sie alle Eigenschaften, die diesen Code auszeichnen.

Der Code ist zyklisch

Der Code ist einschrittig, Hd=1 zwischen zwei benachbarten Codeworten

B) Ist es möglich den Code so zu erweitern, dass er seine Eigenschaften behält und zusätzlich Einzelfehler korrigiert werden können? Falls ja, so tragen Sie die Erweiterung in Abbildung 2 ein. Begründen Sie Ihre Antwort.

Antwort: nein

Begründung: Der Code ist wie bereits erwähnt einschrittig, hat also ein HD_{min} von 1. Da diese Eigenschaft erhalten bleiben soll ist es nicht möglich einen Code zu generieren, der eine minimale HD von 3 aufweist, die benötigt wird, um einen Fehler korrigieren zu können.

Im Folgenden soll eine Schaltung entworfen werden die es erlaubt, einen Code zu generieren, der drei Nutzbits gegen das Auftreten eines Fehlers schützen soll. Hierbei soll ein Hamming-Code zum Einsatz kommen.

A) Welche Anzahl an Prüfbits wird benötigt, um drei Nutzbits gegen das Auftreten eines Fehlers derart zu schützen, so dass der Fehler korrigiert werden kann?

Zur Sicherung werden 3 weitere Bits benötigt. $HD_{min} = 3$, auslesen aus der Tabelle im Anhang ergibt für m=3 und HD=3 die Anzahl an Prüfbits.

B) Bestimmen Sie nun mit Hilfe der folgenden Tabelle, welche Datenbits x_1, x_2, x_3 anhand welcher Prüfbits geschützt werden:

lfd. Nr. duale Kennzahl	1 001	2 010	3 011	4 100	5 101	6 110
1. Stelle	y_1		x_1		x_2	
2. Stelle		y_2	x_1			x_3
3. Stelle				y_3	x_2	x_3

C) Basierend auf der Bedingung, dass die Prüfbits die zugehörigen Datenbits auf **gerade** Parität ergänzen, soll nun die nachfolgende Tabelle ausgefüllt werden.

x_3	x_2	x_1	y_3	y_2	y_1
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	0	1
0	1	1	1	1	0
1	0	0	1	1	0
1	0	1	1	0	1
1	1	0	0	1	1
1	1	1	0	0	0

D) Im Folgenden ist nun ein Schaltnetz zu entwerfen, das aus den Nutzbits die Paritätsbits erzeugt.

Übertragen Sie hierzu die Ansteuerfunktionen in die unten stehenden KV-Diagramme und bilden Sie daraus die jeweilige disjunktive Minimalform. Verfügen Sie evtl. enthaltene Freistellen zu „0“!!

y_1

		x_1	x_1	
	0	1	1	0
x_2	1	0	0	1
			x_3	x_3

$y_1 = x_1\bar{x}_2 \vee \bar{x}_1 x_2$

y_2

		x_1	x_1	
	0	1	0	1
x_2	0	1	0	1
			x_3	x_3

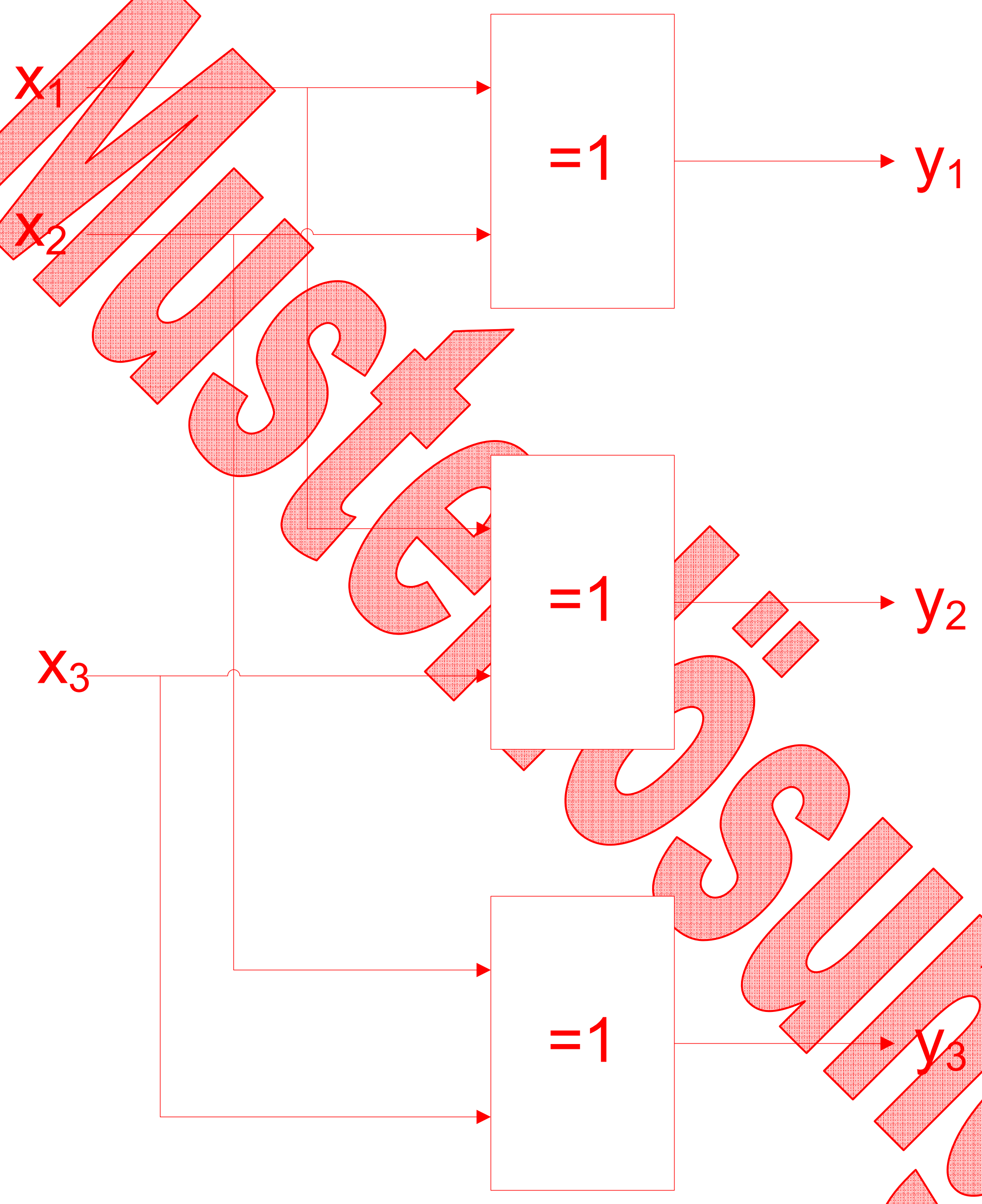
$y_2 = x_1\bar{x}_3 \vee \bar{x}_1 x_3$

y_3

		x_1	x_1	
	0	0	1	1
x_2	1	1	0	0
			x_3	x_3

$y_3 = x_2\bar{x}_3 \vee \bar{x}_2 x_3$

E) Entwerfen Sie nun basierend auf den ermittelten Disjunktiven Minimalformen das Schaltnetz, das mit einer minimalen Anzahl an Gattern realisiert werden kann. Hierfür haben Sie eine beliebige Anzahl an Invertern sowie UND, ODER und EXOR Gattern mit je zwei Eingängen zur Verfügung.



Aufgabe 6 Mengen & Relationen

A) Bestimmen Sie den Wahrheitscharakter einer jeden Aussage. Beachten Sie: für eine falsche Antwort wird ein Punkt abgezogen!

Aussage	Wahr	Falsch
Eine Menge kann mehrere gleiche Elemente enthalten	☐	☒
Eine Menge kann stets durch Auflistung aller ihrer Elemente vollständig beschrieben werden	☐	☒
Seien S und T Mengen. Es gilt $\|S\times T\|=\|T\times S\|$	☒	☐
Die Äquivalenzklassen zweier Elemente bzgl. einer Äquivalenzrelation sind entweder identisch oder disjunkt	☒	☐
Die leere Menge ist Element jeder Menge	☐	☒

B) Geben Sie die definierenden Eigenschaften einer Verträglichkeitsrelation an.

Reflexiv, symmetrisch, nicht transitiv

C) Ist die strikte Ordnung „<“ eine Ordnungsrelation? Begründen Sie Ihre Antwort!

Nein, da sie nicht reflexiv ist.

D) Wir betrachten einen zyklischen Binärcounter von 8 Bit Länge, das heißt bei Erhöhung des Counters im Zustand 1111 1111 ist der Folgezustand 0000 0000, ohne dass ein Übertrag oder Ähnliches berücksichtigt würde. Sei M die Menge der Zustände des Counters.

I) Was ist die Mächtigkeit |M| der Menge?

II) Wir nennen einen Zustand z0 „kleinergleich“ z1, wenn z1 = z0 oder aber wenn z1 von z0 aus in endlich vielen Zählererhöhungen erreicht werden kann. Ist „kleinergleich“ eine Ordnungsrelation? Begründen Sie ihre Antwort.

I) $2^8 = 256$

II) Nein „kleinergleich“ keine Ordnungsrelation, da sie nicht antisymmetrisch ist.

Aufgabe 7 Automaten

Gegeben ist das Ablaufdiagramm eines Automaten zur Steuerung eines elektrischen Fensterhebers im Automobil. Die Bewegung des Fensters wird durch einen mechanischen Taster gesteuert, der die beiden Steuersignale Fenster hoch (SH = 1) und Fester runter ausgibt (SR = 1), wenn dieser in die entsprechende Richtung bewegt wird. Weiterhin stehen ein oberer und unterer Endschalter (ESH, ESR) zur Verfügung, die jeweils zu eins gesetzt werden, sobald das Fenster die jeweilige Endposition erreicht.

Als Ausgänge werden zwei Steuersignale für den Motor verwendet mit denen das Fenster geschlossen (Motor_H = 1) oder geöffnet (Motor_R = 1) werden kann.

Bei Tastendruck bewegt sich das Fenster in die entsprechende Richtung bis der obere oder untere Anschlag erreicht ist. Bei losgelassenem Taster bewegt sich das Fenster weiter. Die Fahrt kann durch einen entgegen gesetzten Tastendruck jederzeit gestoppt werden.

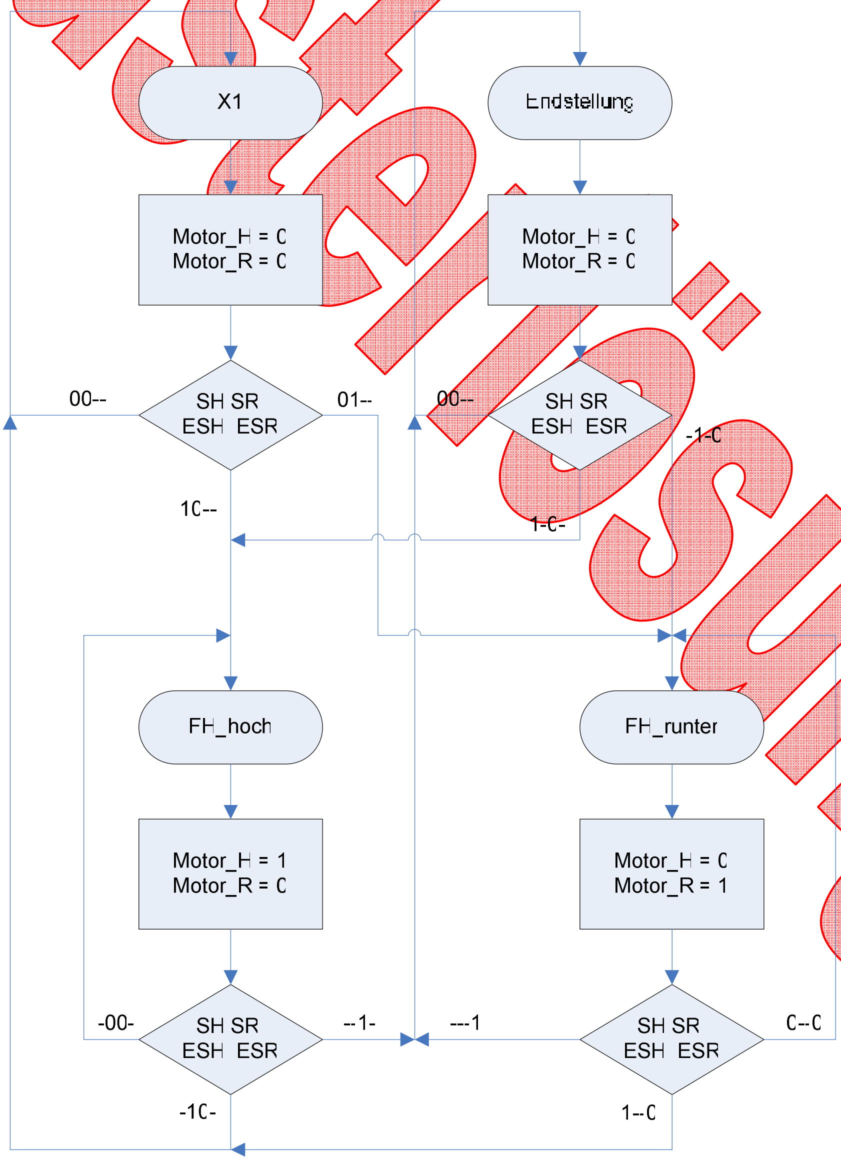


Abbildung 3

Hinweise:

- Die Tastersignale sind jeweils nur für einen Takt eins, also nur bei einem Zustandsübergang aktiv
- Es ist mechanisch ausgeschlossen, daß beide Tastersignale gleichzeitig eins sind (don't care)
- Die Endschalter können mechanisch ebenfalls nicht beide eins sein (don't care)
- Um Beschädigungen des Motors zu vermeiden ist darauf zu achten, dass immer nur eines der Steuersignale eins ist.

A) Wie viele FlipFlops sind zur Realisierung des Automaten mindestens nötig, wenn man entweder T-FlipFlops oder aber D-FlipFlops verwendet? Begründen Sie Ihre Antwort!

Antwort: Der Automat verfügt über 4 Zustände -> es werden 2 Flipflops benötigt, da $\lceil \mathrm{ld}\ 4 \rceil = 2$!

B) Um welchen Automatentyp handelt es sich in der obigen Abbildung 1. Begründen Sie Ihre Antwort.

Antwort: Es handelt sich um einen Moore-Automaten, da die Ausgabe nur vom Zustand abhängt.

C) Welche Bedeutung hat der Zustand X1 und an welcher Position steht das Fenster in diesem Zustand? Ist das Fenster in Bewegung?

Antwort: Das Fenster bewegt sich nicht und befindet sich nicht am oberen oder unteren Endschalter. Es ist damit weder komplett geöffnet oder geschlossen.

D) Welcher Vorteil ergibt sich durch die mechanische Einschränkung, daß beide Endschalter bzw. Tastersignale nicht gleichzeitig 1 sein können? Begründung!

Antwort: All die Fälle sind nicht relevant und können bei der Logikminimierung zu 0 oder 1 verfügt werden (don't care). Gewinn: kleineres Schaltnetz.

E) In Tabelle 6 ist für einige Fälle die Übertragungsfunktion vorgegeben. Bestimmen Sie daraus den Typ des FlipFlops und ergänzen Sie die Ansteuertabelle des verwendeten FlipFlop-Typs in Tabelle 4.

		Typ
Q	Q^{n+1}	DFF
0	0	0
0	1	1
1	0	0
1	1	1

Tabelle 4

Füllen Sie nun Tabelle 6 aus, indem Sie die Zustandsübergänge aus dem Ablaufdiagramm (Abbildung 3) in die Tabelle übertragen. Bestimmen Sie anschließend die Werte zur Ansteuerung der FlipFlops.
Hinweis: Zur Vereinfachung der Aufgabe wurden in Tabelle 6 einige Zustandsübergänge weggelassen.

Zustandscodierung:

Zustand	Q_1	Q_0
X1	0	0
FH_hoch	0	1
FH_runter	1	0
Endstellung	1	1

Tabelle 5

								Ansteuerung	
								Q_1	Q_0
Q_1	Q_0	SH	SR	ESH	ESR	Q_1^{n+1}	Q_0^{n+1}	TYP: D	TYP: D
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	1	0	1	0
0	0	0	1	0	1	1	0	1	0
0	0	0	1	1	0	1	0	1	0
0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	1	0	1	0	1
0	1	0	0	1	0	1	1	1	1
0	1	0	1	0	0	0	0	0	0
0	1	0	1	0	1	0	0	0	0
0	1	0	1	1	0	1	1	1	1
1	0	0	0	0	0	1	0	1	0
1	0	0	0	0	1	1	1	1	1
1	0	0	0	1	0	1	0	1	0
1	0	0	1	0	0	1	0	1	0
1	0	0	1	0	1	1	1	1	1
1	0	1	0	0	0	0	0	0	0
1	0	1	0	0	1	1	1	1	1
1	1	0	0	0	0	1	1	1	1

Tabelle 6

F) Übertragen Sie die Ansteuerfunktionen für das FlipFlop Q_0 in das unten stehenden KV-Diagramm. Bilden Sie anschließend die disjunktive Minimalform zur Ansteuerung des FlipFlops. Eventuell vorhandene Freistellen sind bereits so zu 0 oder 1 verfügt worden, dass sich ein optimaler Entwurf ergibt.

SR

ESR ESR

Q_0^{n+1}

0			0				1
	0	0			1	1	
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
		1	0	0	1	1	1
0	1	1	0	0	1	1	1
	1	1	0	0	1	1	1
				0	1	1	1

ESH

SH

ESH

Q_1^n

Q_0^n

Lösung:

Q_0^{r+1}

SR

ESR ESR

ESH, SH, ESH, Q_1^r, Q_0^r

0	0	0	0	0	0	1	1
0	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
0	1	1	0	0	1	1	1
0	1	1	0	0	1	1	1
0	1	1	0	0	1	1	1
0	1	1	0	0	1	1	1

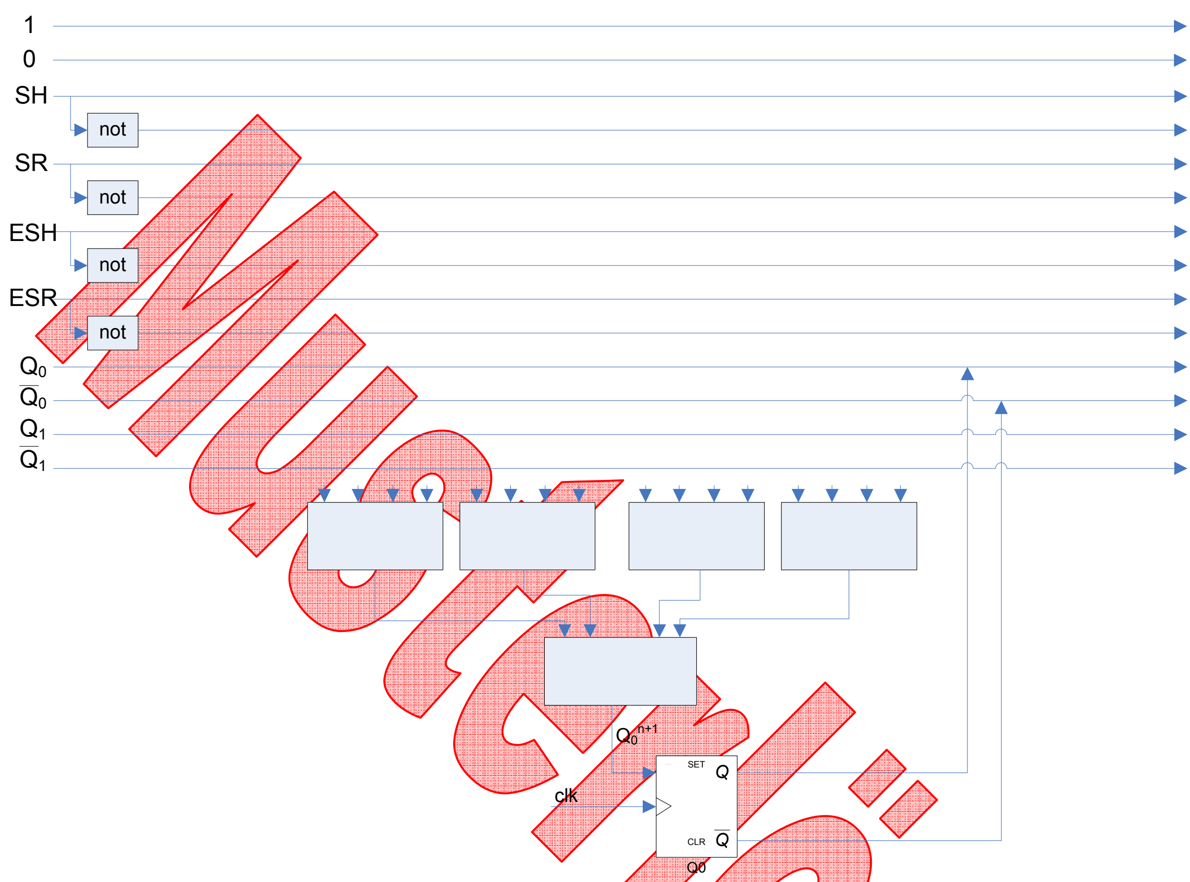
q_0^{n+1}: $(SH \,\&\, \overline{q_0}) \vee (\overline{SR} \,\&\, q_0) \vee (ESR \,\&\, q_1) \vee (ESH \,\&\, q_0 \,\&\, \overline{q_1})$

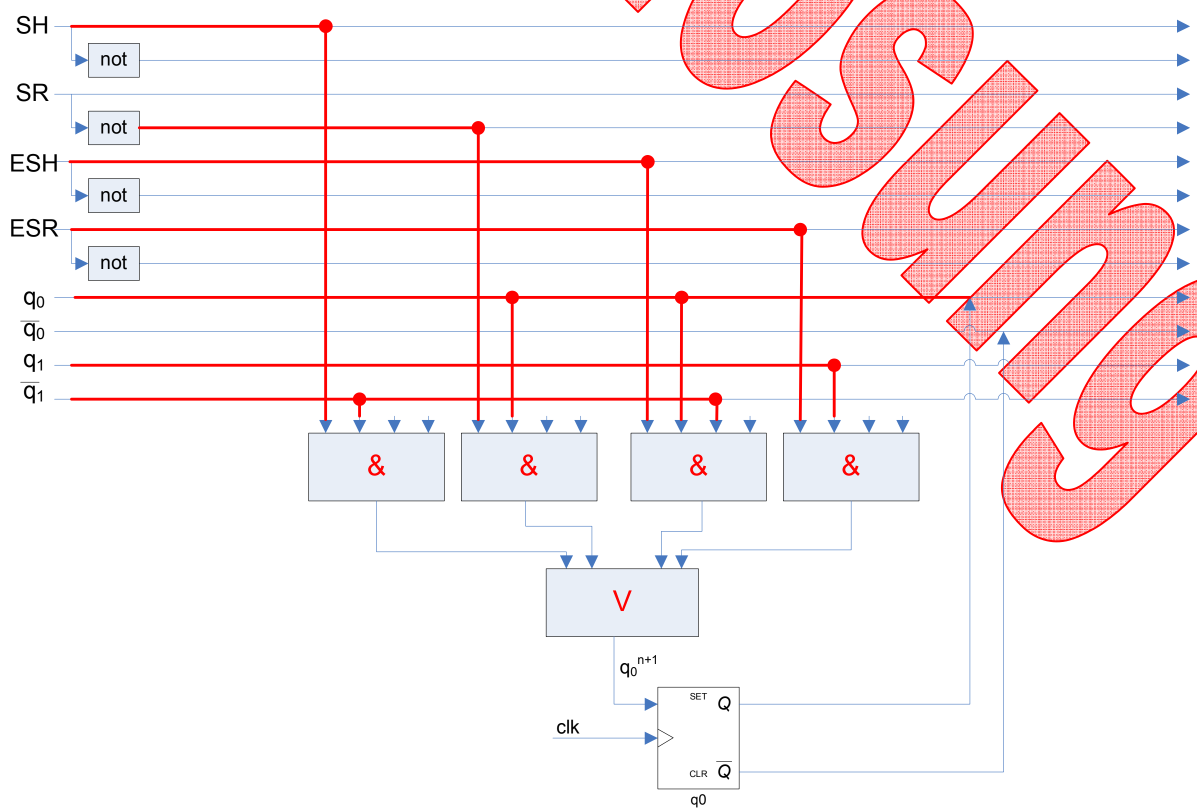
G) Leiten Sie, basierend auf der Zustandscodierung, die Ansteuerfunktion für die beiden Ausgänge Motor_H und Motor_R ab.

Motor_H: $q_0 \,\&\, \overline{q_1}$

Motor_R: $q_1 \,\&\, \overline{q_0}$

H) Zeichnen Sie basierend auf dem Ergebnis der Teilaufgabe E) die Ansteuerung des FF Q_0 in das vorgegebene Schaltnetz ein.





Zusätzliches Lösungsblatt 1:

Zusätzliches Lösungsblatt 2: