Prüfung

Prof. Dr.-Ing. J. Becker

Digitaltechnik

WS 2009/2010

Institut für Technik der Informationsverarbeitung, KIT

Klausur

Mo., 15.03.2010

Lösungsblätter

Hinweise zur Klausur

Hilfsmittel

Als Hilfsmittel zur Prüfung sind vier Seiten vorgegebene und **<u>ein DIN A4 Blatt</u>** selbst geschriebene Formelsammlung zugelassen. Nicht erlaubt hingegen sind die Verwendung eines Taschenrechners, zusätzlicher Unterlagen und jegliche Kommunikation mit anderen Personen.

Prüfungsdauer

Die Prüfungsdauer beträgt für die Klausur 120 Minuten.

Prüfungsunterlagen

Die Prüfungsunterlagen bestehen aus insgesamt 24 bedruckten Aufgabenblättern (einschließlich diesem Titelblatt und zusätzlicher Lösungsblätter). Weiterhin sind 4 zusätzliche Seiten Formelsammlung enthalten.

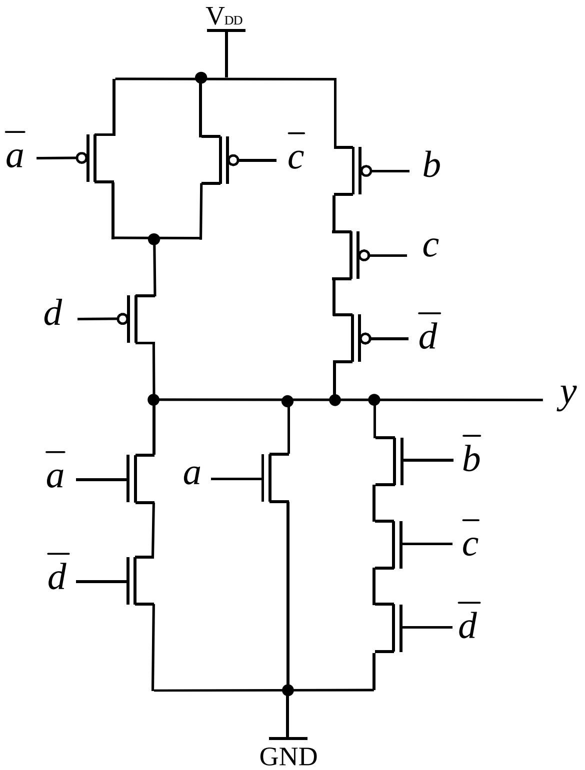
Bitte vermerken Sie vor der Bearbeitung der Aufgaben auf jeder Seite oben Ihren Namen, auf der ersten Seite zusätzlich die Matrikelnummer!

Auf jedes zusätzliche Lösungsblatt ist neben dem Namen auch die Aufgabennummer mit einzutragen. Vermeiden Sie das Beschreiben der Rückseiten.
Am Ende der Prüfung sind die 24 Seiten Aufgaben- und Lösungsblätter und alle verwendeten zusätzlichen Lösungsblätter abzugeben.
Verwenden Sie zum Bearbeiten der Aufgaben lediglich dokumentenechte Schreibgeräte – keinen Bleistift sowie Rotstifte!

Aufgabe 1	CMOS-Schaltnetze	2		~6%
Aufgabe 2	Information und Codierung	5		~14%
Aufgabe 3	Fehlererkennung/Fehlerkorrektur	8		~9%
Aufgabe 4	Automaten	10		~16%
Aufgabe 5	Mengen, Relationen, Graphen	13		~8%
Aufgabe 6	Boolsche Algebra & Zahlensysteme	15		~11%
Aufgabe 7	Minimierung	17		~17%
Aufgabe 8	Schaltnetze	20		~20%
			Σ	

Aufgabe 1 CMOS-Schaltnetze

Gegeben sei folgender CMOS-Schaltkreis:



A) Geben Sie für die vorliegende Schaltung sowohl die pull-down-Funktion G als auch die pull-up-Funktion F in disjunktiver Minimalform an.

$$G = \overline{a}\,\overline{d} + a + \overline{b}\,\overline{c}\,\overline{d}$$

$$F = (a+c)\overline{d} + \overline{b}\,\overline{c}d = a\overline{d} + c\overline{d} + \overline{b}\,\overline{c}d$$

B) Gegeben sind folgende pull-down bzw. pull-up-Funktionen F bzw. G. Stellen Sie fest, ob die durch die beiden Funktionen definierte CMOS Schaltung, wohldefiniert ist. Prüfen sie dazu auf Kurzschlüsse und vollständige Definition.

$$G = \bar{a}\bar{c} + bd + a\bar{c}\bar{d}$$

$$F = \bar{b}c + a\bar{b}d + c\bar{d}$$

Prüfen auf Kurzschlüsse:

$$F \cdot G \quad (\overset{!}{=} 0)$$

$$= \left(\bar{a}\bar{c} + bd + a\bar{c}\bar{d}\right) \cdot \left(\bar{b}c + a\bar{b}d + c\bar{d}\right)$$

$$= \cancel{\bar{a}\bar{c}\bar{b}c} + \cancel{bd\bar{b}c} + \cancel{a\bar{c}\bar{d}\bar{b}c}$$

$$+ \cancel{\bar{a}\bar{c}a\bar{b}d} + \cancel{bda\bar{b}d} + \cancel{a\bar{c}\bar{d}a\bar{b}d}$$

$$+ \cancel{\bar{a}\bar{c}c\bar{d}} + \cancel{bdc\bar{d}} + \cancel{a\bar{c}\bar{d}c\bar{d}}$$

$$= 0$$

→ Kein Kurzschluss!

Prüfen auf vollständige Definition:

$$F + G \quad (\overset{!}{=} 1)$$

$$= \left(\bar{a}\bar{c} + bd + a\bar{c}\bar{d}\right) + \left(\bar{b}c + a\bar{b}d + c\bar{d}\right)$$

$$= c\left(bd + \bar{b} + a\bar{b}d + \bar{d}\right) + \bar{c}\left(\bar{a} + bd + a\bar{d} + a\bar{b}d\right)$$

$$= c\left(b\left(d + \bar{d}\right) + \bar{b}\right) + \bar{c}\left(\bar{a} + a\left(bd + \bar{d} + \bar{b}d\right)\right)$$

$$= c\left(b(1) + \bar{b}\right) + \bar{c}\left(\bar{a} + a(1)\right)$$

$$= c(1) + \bar{c}(1)$$

$$= 1$$

→ vollständig definiert!

C) Gegeben ist folgende pull-up-Funktion F. Bestimmen Sie die minimale Anzahl an Transistoren, die für die Implementierung der Funktion F innerhalb einer CMOS Realisierung notwendig sind. Stellen Sie den Lösungsweg dar und begründen Sie ihre Antwort.

$$F = \left(c + \overline{d}\left(a + \overline{b}\right)\right)\left(b\overline{c} + \overline{a}\left(c + d\right)\right)$$

$$F = \left(c + \overline{d}\left(a + \overline{b}\right)\right)\left(b\overline{c} + \overline{a}\left(c + d\right)\right)$$

$$= \left(c + a\overline{d} + \overline{b}\overline{d}\right)\left(b\overline{c} + \overline{a}c + \overline{a}d\right)$$

$$= \overline{a}c + \cancel{\overline{a}cd} + ab\overline{c}\overline{d} + \cancel{\overline{a}\overline{b}c\overline{d}}$$

$$= \overline{a}c + ab\overline{c}\overline{d}$$

→ 6 Transistoren (12 Transistoren wenn <u>PullUp & Pulldown</u> realisiert werden)

→ Jede Variable benötigt einen Transistor

Aufgabe 2 Information und Codierung

Aufgabe 2.1 Codierung

Sie werden als Entwickler eines Modellspielzeugherstellers eingestellt. Ihre erste Aufgabe besteht darin, die Datenübertragung zur Steuerung eines Radio-Controlled-Helikopters (RC-Heli) zu optimieren.

Die Firma vertreibt zwei Typen von Helikoptern (grün und rot), **die über den selben Funkkanal gesteuert werden**. Auf diesem Kanal müssen die Daten für beide Heli-Typen übertragen werden. Um die Steuerkommandos an einen Heli zu übertragen, wird für jeden Heli-Typ jeweils ein Befehlssatz kodiert. Neben dem Typ (Rot/Grün) beinhaltet ein solcher Befehlssatz 5 Steuerbefehle („Steigen“, „Sinken“, „Linksdrehen“, „Rechtsdrehen“ und „Alle Rotoren STOP“)

Aufgrund der Farben werden die beiden Heli-Typen in unterschiedlichen Stückzahlen produziert und verkauft:

Rot: 7 500 000 Einheiten

Grün: 2 500 000 Einheiten

Die verschiedenen Kommandos zur Steuerung der Helis treten mit unterschiedlichen Wahrscheinlichkeiten auf:

Steigen (Up) : 28 %

Sinken (Down): 12 %

Linksdrehen (Left): 32 %

Rechtsdrehen (Right): 24 %

Alle Rotoren Stop (Stop) : 4 %

A) Geben Sie die Auftrittswahrscheinlichkeit der möglichen Kommandos für jeden der beiden Heli-Typen an. Verwenden sie die nachfolgende Tabelle 2-1. **Geben Sie alle Wahrscheinlichkeiten in „%“ (Prozent) an!**

Rot Up (RU)	Grün Up (GU)	Rot Down (RD)	Grün Down (GD)	Rot Left (RL)
¾ * 0,28 = 0,21 = 21 %	¼ * 0,28 = 0,07 = 7 %	¾ * 0,12 = 0,09 = 9 %	¼ * 0,12 = 0,03 = 3 %	¾ * 0,32 = 0,24 = 24 %

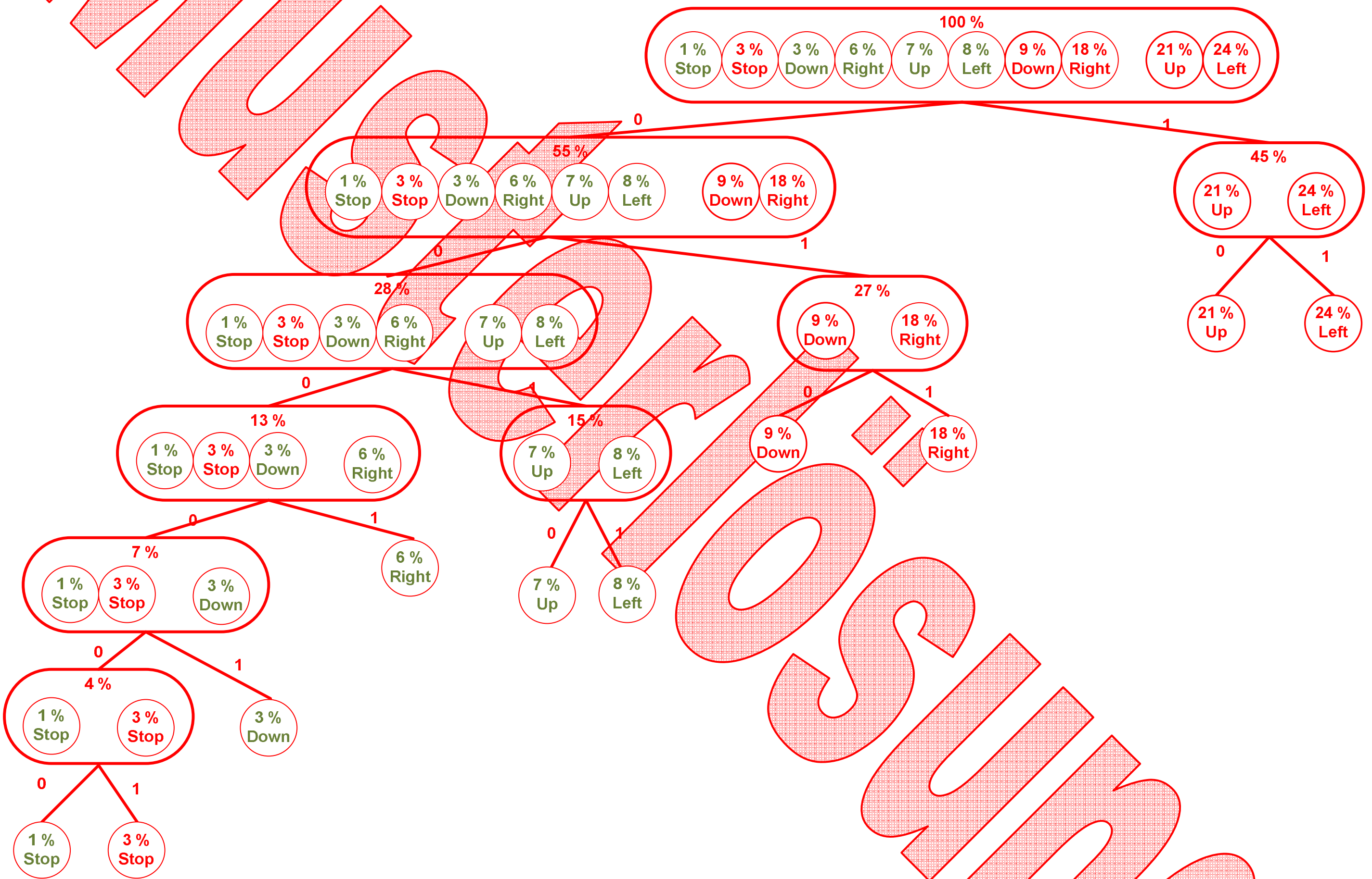
Grün Left (GL)	Rot Right (RR)	Grün Right (GR)	Rot Stop (RS)	Grün Stop (GS)
¼ * 0,32 = 0,08 = 8 %	¾ * 0,24= 0,18 = 18 %	¼ * 0,24 = 0,06 = 6 %	¾ * 0,04 = 0,03 = 3 %	¼ * 0,04 = 0,01 = 1 %

Tabelle 2-1: Auftrittswahrscheinlichkeiten

B) Um die durchschnittliche Codewortlänge zu minimieren, sollen die verschiedenen Steuerkommandos nach Shannon-Fanø codiert werden. Entwickeln Sie für alle Kommandos eine Shannon-Fanø-Codierung und tragen Sie die Codes in Tabelle 2-2 ein.

Hinweise:

- Sortieren sie die Knoten nach aufsteigenden Wahrscheinlichkeiten von links nach rechts. Wenn identische Wahrscheinlichkeiten auftreten darf die Reihenfolge frei gewählt werden.
- Weisen Sie den linken Ästen des entstehenden Baumes die „0" zu, den rechten Ästen die „1".
- Verwenden Sie die Partitionierungskonvention aus der Digitaltechnikvorlesung.



Rot Up (RU)	Grün Up (GU)	Rot Down (RD)	Grün Down (GD)	Rot Left (RL)
10	0010	010	00001	11

Grün Left (GL)	Rot Right (RR)	Grün Right (GR)	Rot Stop (RS)	Grün Stop (GS)
0011	011	0001	000001	000000

Tabelle 2-2: Steuerbefehlcodierung

C) Ein Datenpaket besteht aus 5 Steuerkommandos. Pro Paket darf <u>maximal ein Stop-Befehl</u> enthalten sein. Die Länge der Nutzdaten (Steuerkommandos) pro Paket darf 30 Bit nicht übersteigen. Können alle erlaubten Kombinationen an Befehlen mit diesen Vorgaben übertragen werden? (Die Länge des Pakets ergibt sich aus der Länge der Kodierung der Befehle mit Shannon-Fanø). Begründen Sie Ihre Antwort mit einer Rechnung!

Worst Case: Grüner Heli, 4 x Down, 1 x Stop

a) 4 * 5 Bit + 1 * 6 Bit = 26 Bit

b) 4 * 6 Bit + 1 * 6 Bit = 30 Bit

=> Ja, alle möglichen Kombinationen können übertragen werden.

D) Geben Sie die mittlere Codewortlänge der gewählten Shannon-Fanø-Codierung an.

Berechnung der durchschnittlichen Codewortlänge für Shannon-Fanø:

24% * 2 Bit + 21% * 2 Bit + 18% * 3 Bit + 9% * 3 Bit + 8% * 4 Bit + 7 % * 4 Bit + 6% * 4 Bit + 3% * 5 Bit + 3% * 6 Bit + 1% * 6 Bit = 2,94 Bit

E) Geben Sie die mittlere Länge der Steuerkommandos an, falls alle Kommandos nicht mittels eines optimalen Codes, sondern rein binär kodiert würden?

10 Steuerkommandos => Anzahl der Bits pro Codewort = ceiling [ld(10)] = 4 Bit

Aufgabe 3 Fehlererkennung/Fehlerkorrektur

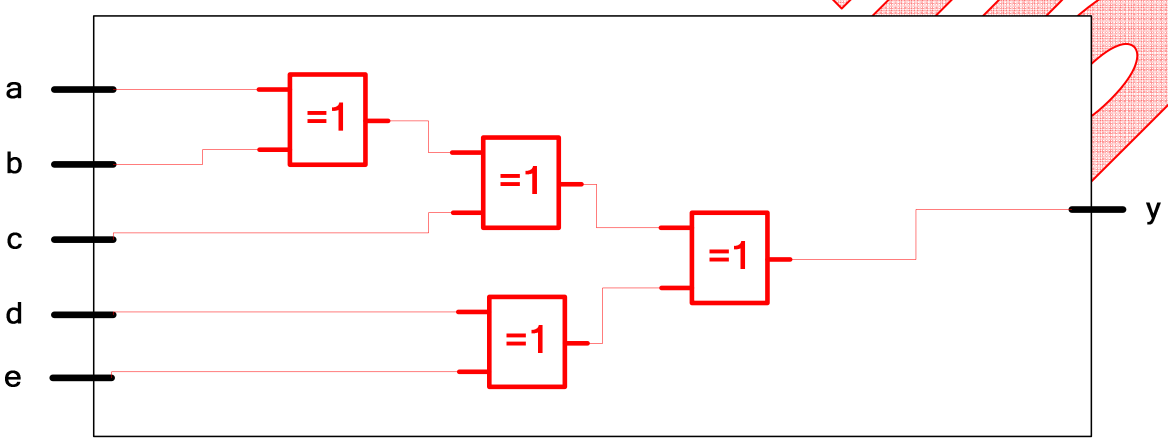
Aufgabe 3.1 Fehlererkennung durch Parität

Auf einem seriellen Datenübertragungskanal treten Bitfehler auf. Deshalb soll der Kanal so gesichert werden, dass pro Datenwort (4 Bit) ein Bitfehler erkannt werden kann.

A) Sie entscheiden sich dazu jedes Codewort mit einem Paritätsbit zu sichern. Die zu übertragenden Codewörter sollen auf <u>ungerade Parität</u> ergänzt werden. Geben Sie für folgende Codewörter (Dezimale Darstellung) die Binäre Repräsentation, sowie die Paritätsbits an.

Dezimaldarstellung	Binäres Datenwort e d c b	Paritätsbit a
0	0 0 0 0	1
3	0 0 1 1	1
7	0 1 1 1	0
8	1 0 0 0	0
13	1 1 0 1	0
15	1 1 1 1	1

B) Zur Prüfung der Parität soll ein Schaltnetz zum Einsatz kommen. Wenn ein Paritätsfehler vorliegt soll y='0' sein, sonst soll y='1' sein. Realisieren Sie die Paritätserkennung mit einer <u>minimalen Anzahl von XOR2-Gattern</u>. Zeichnen Sie das resultierende Schaltnetz wenn Codeworte auf ungerade Parität ergänzt wurden..



Aufgabe 3.2 Fehlerkorrektur durch Hamming-Codes

Eine serielle Datenübertragung ist mit einem Hamming-Code gesichert. Jedes Codewort besteht aus 4 Datenbits (x0, x1, x2, x3) gefolgt von 3 Prüfbits (y0, y1, y2). Abbildung 3-1 zeigt den Aufbau des verwendeten Hamming-Codes.

X3	X2	X1	X0	Y2	Y1	Y0

Abbildung 3-1: Aufbau des Hamming-Codes

A) Welche Hammingdistanz hat der beschriebene Code? Wie viele Bitfehler pro Codewort können damit korrigiert werden?

Hammingdistanz: $HD_{min} = 3$

Damit kann 1 Fehler pro Codewort korrigiert werden

Konstruieren Sie den 1-F-korrigierbaren Hammingcode. Gegeben ist dazu ein fehlerfreier Datenstrom unter Verwendung gerader Parität: **0100101 0001111 0010110 1000011**
Die Reihenfolge der übertragenen Bits entspricht Abbildung 3-1.

B) Ergänzen Sie die untenstehende Tabelle. Bestimmen Sie die korrekte Zuordnung von Prüf- und Datenbits entsprechend der in der Tabelle gegebenen dualen Kennzahlen. Tragen Sie in der folgenden Tabelle ein, welche Datenbits x0, x1, x2, x3 durch welche Prüfbits y0, y1, y2 geschützt werden.

lfd. Nr. **duale Kennzahl**	1 **001**	2 **010**	3 **011**	4 **100**	5 **101**	6 **110**	7 **111**
1. Stelle	y0		x3		x2		x0
2. Stelle		y1	x3			x1	x0
3. Stelle				y2	x2	x1	x0

C) Es wurde eine zweite Folge aus Codewörtern empfangen. Es kam die gleiche Codierung wie in Aufgabenteil B) gefordert ist zum Einsatz. Wieviele Fehler sind bei der Übertragung mindestens aufgetreten? Markieren Sie die korrigierbaren Bits in der unten gegebenen Folge.

Codewort Folge: 0100011 1110000 0001000

Zwei Fehler,

Korrektur: 0110 011 1110 000 0000 000

Aufgabe 4 Automaten

Aufgabe 4.1 Analyse eines Automaten

A) Über das Ablaufdiagramm in Abbildung 4–1 (siehe nachfolgende Seite) sei ein Automat vollständig beschrieben. Welcher Automatentyp ist in Abbildung 4–1 dargestellt? Begründen Sie Ihre Antwort.

Mealyautomat. Die Ausgabe ist von Zustand UND Eingabe abhängig.

B) Welches generelle Problem tritt bei der Realisierung über den in Abbildung 4–1 beschriebenen Automatentyp auf (im Vergleich der anderen Automatentypen aus der Vorlesung)?

Änderungen am Eingang des Automaten schlagen immer sofort auf die Ausgabe durch..

C) Wieviele Flipflops werden zur Realisierung des Automaten aus Abbildung 4–1 minimal benötigt, wenn ausschliesslich JK Flipflops verwendet werden? Begründen Sie Ihre Antwort.

Es müssen 3 Zustände kodiert werden. Somit ergibt sich aus $\lceil ld\,3 \rceil = 2$, dass minimal 2 Flipflops notwendig sind (unabhängig vom Typ der Flipflops).

D) Welche Funktion weißt der in Abbildung 4–1 gegebene Automat auf? Welche Funktion haben die Signale „d“ und „en“?

Auf-/Ab-**Binärzähler** (von 1_D bis 3_D, Überlauf nach 3_D auf 1_D)

en: **enable Signal**, wenn en=0 wird der aktuelle Zustand gehalten

d: **Zählrichtung** wenn d=1 dann hochzählen (Addition mit 1) sonst runterzählen (Subtraktion von 1)

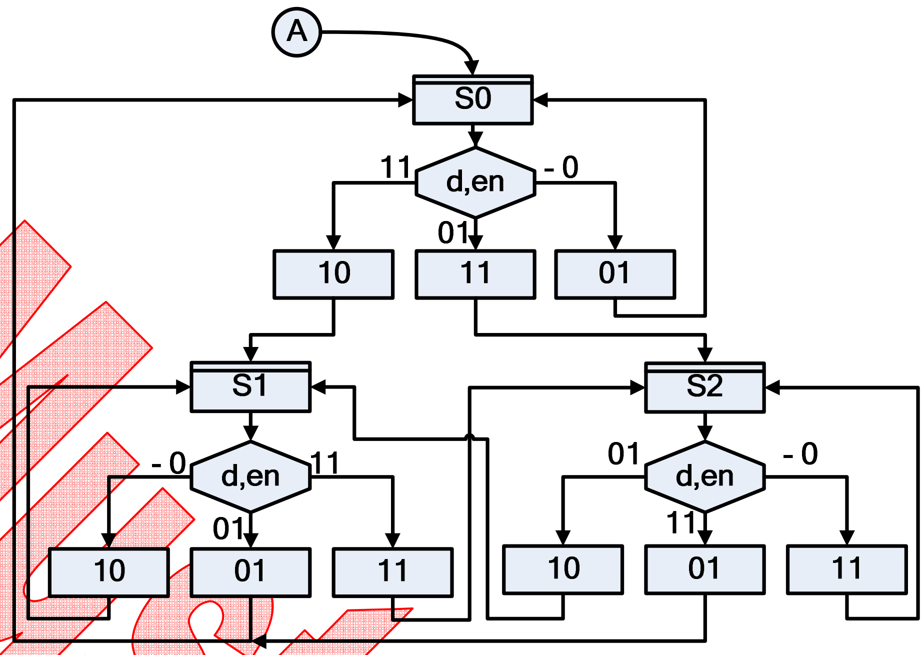


Abbildung 4–1: Ablaufdiagramm eines Automaten

E) Das Ablaufdiagramm aus Abbildung 4–1 soll in eine Ablauftabelle übertragen werden. Vervollständigen Sie die Felder in Tabelle 4-1 und streichen Sie eventuell nicht notwendige Spalten (Q: Zustandskodierung, X: Eingangsgröße, Y: Ausgangssignale, Zustandsspeicherung über Toggel Flipflops). Verwenden Sie eine minimale Anzahl von Flip Flops.

	Q^t			X^t		Q^{t+1}			T-FF			Y^t		
	q_2	q_1	q_0	d	en	q_2	q_1	q_0	t_2	t_1	t_0	y_2	y_1	y_0
S2		1	0	-	0		1	0		0	0		1	1
				0	1		0	1		1	1		1	0
				1	1		0	0		1	0		0	1
S1		0	1	-	0		0	1		0	0		1	0
				0	1		0	0		0	1		0	1
				1	1		1	0		1	1		1	1
S0		0	0	-	0		0	0		0	0		0	1
				0	1		1	0		1	0		1	1
				1	1		0	1		0	1		1	0

Tabelle 4-1: Ablauftabelle

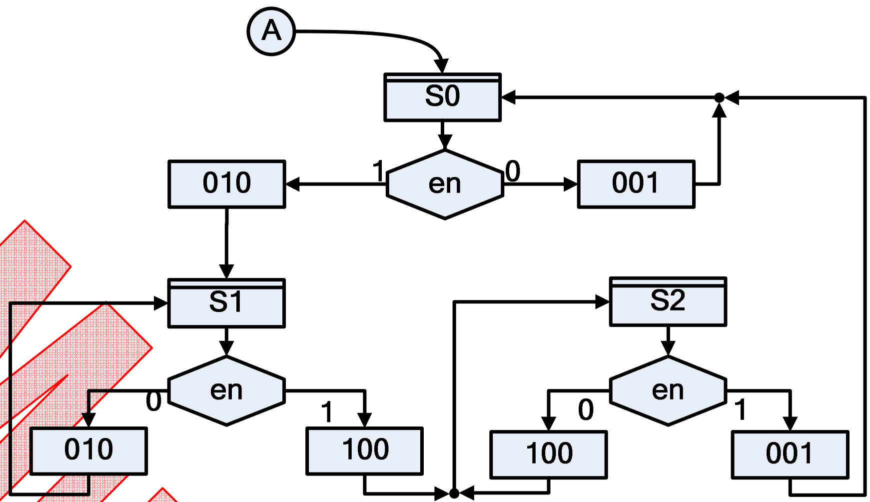


Abbildung 4–2: Vereinfachtes Ablaufdiagramm

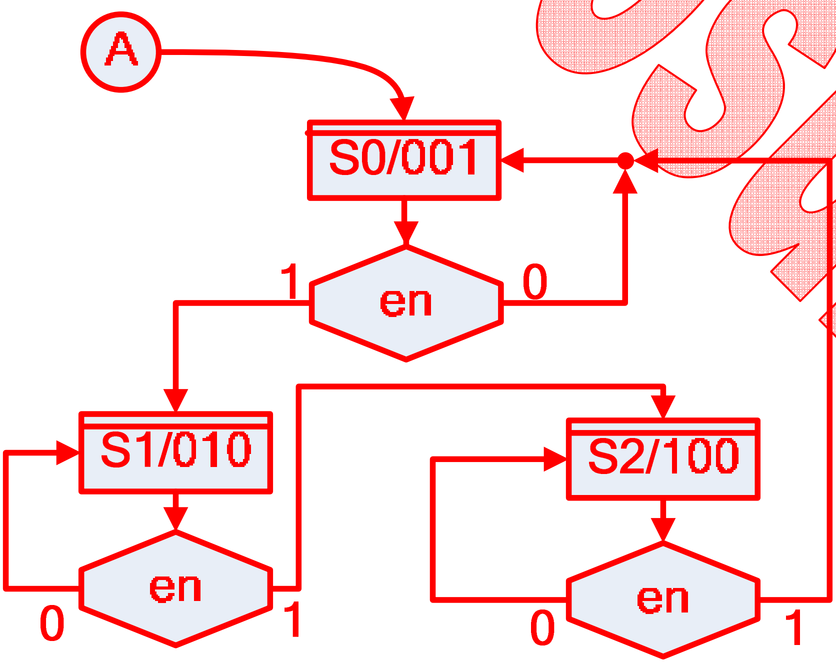
F) In Abbildung 4–2 ist eine vereinfachte Version eines Ablaufdiagramms gegeben. Die darin beschriebene Funktionalität soll als Medwedewautomat realisiert werden. Geben Sie die Kodierung der minimal notwendigen Zustände an wie sie sich aus dem Ablaufdiagramm oben ableiten lässt.

S0: 001

S1: 010

S2: 100

G) Zeichnen Sie den vereinfachten Automaten aus Abbildung 4–2 als Medwedewautomat. Verwenden Sie dabei eine minimale Anzahl von Zuständen.



Aufgabe 5 Mengen, Relationen, Graphen

Für die folgenden Teilaufgaben interpretieren wir Mengen als Knoten eines Graphen und Teilmengenbeziehungen als gerichtete Kanten (Pfeile) wie folgt:

$\subset \;\hat{=}\; \leftarrow$,

$\supset \;\hat{=}\; \rightarrow$,

$= \;\hat{=}\; \leftrightarrow$.

Gegeben seien des Weiteren die neun folgende Mengen:

{0}, {7}, {10}, {11},

N = Menge der natürlichen Zahlen nach Peano, d.h. 0 ist keine natürliche Zahl.

$N_0 = N \cup \{0\}$

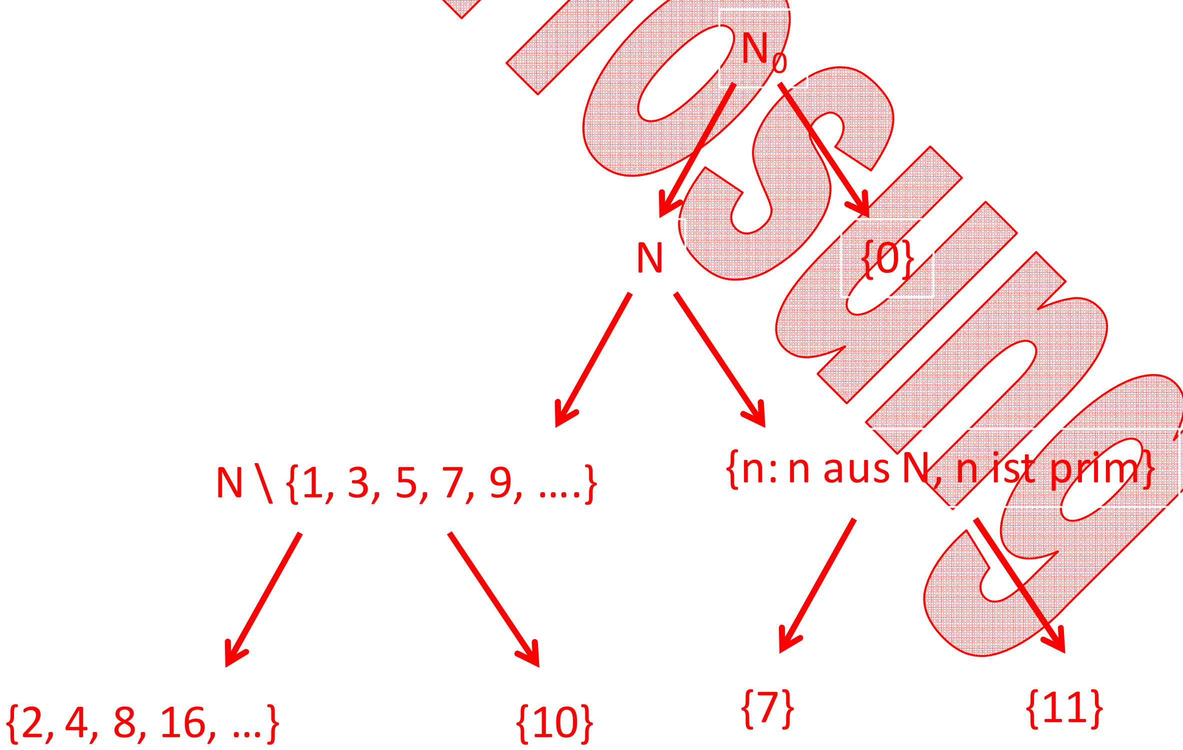
N \ {1, 3, 5, 7, 9, ….}

{n: n $\in$ N, n ist prim}

{2, 4, 8, 16, …}

A) Erstellen Sie aus den gegebenen Mengen als Knotenmenge durch Auswahl geeigneter Beziehungspfeile einen Binärbaum.

<u>Hinweis:</u> Nicht alle Teilmengenbeziehungen müssen verwendet werden.



B) Was ist die Wurzel W des Baumes aus Teilaufgabe A?

Die Wurzel ist N_0.

C) Könnte man aus den gegebenen Mengen auch einen Binärbaum erstellen, bei dem das in Teilaufgabe B) gewählte W keine Wurzel ist? Begründen Sie Ihre Aussage oder geben Sie ein Beispiel.

N_0 ist immer Wurzel, da keine andere der gegebenen Mengen Obermenge von N_0 ist und sie somit keine Vorgänger haben kann.

D) Fügen Sie Menge $\{\pi\}$ in den Graphen der Teilaufgabe A) ein. Verwenden Sie die oben angegebenen Regeln. Es reicht die Angabe (Zeichnung) des neuen Knotens mit allen Knoten aus Teilaufgabe A), die direkt mit dem einzufügenden Knoten durch eine Kante verbunden sind.

$\{\pi\}$

E) Ist das Ergebnis noch ein Binärbaum? Falls nein – warum nicht?

Das Ergebnis ist kein Binärbaum, da nicht zusammenhängend.

Die Teilbarkeitsrelation „|“ auf N sei definiert als:
Seien a, b ∈ N. Es gilt: a | b genau dann wenn a Teiler von b ist.

F) Geben Sie die Relationseigenschaften der so definierten Teilbarkeitsrelation an. Welcher speziellen Relation aus der Vorlesung entspricht die gegebene Teilbarkeitsrelation?

reflexiv (jedes a teilt sich selbst),

antisymmetrisch (aus a | b folgt a < b oder a=b, wenn zusätzlich gilt: b | a folgt immer a=b)

transitiv (wenn a | b und b | c dann auch a | c, da b=ax, c=by=axy mit x, y ∈ N).

Daraus folgt: Ordungsrelation

Aufgabe 6 Boolsche Algebra & Zahlensysteme

Aufgabe 6.1 Boolsche Algebra

Mit Hilfe der ÄQUIVALENZ, der ODER-Operation und einer Konstanten kann ein Basissystem aufgebaut werden.

A) Wie kann in diesem Basissystem die NEGATION dargestellt werden?

$$\bar{a} = (a \equiv 0)$$

Für das Operatorensystem Konjunktion, Disjunktion und Negation ist mit der Disjunktiven bzw. der Konjunktiven Normalform (DNF, KNF) eine kanonische Darstellung möglich. Für das Operatorensystem Antivalenz, Konjunktion und Konstante existiert ebenfalls eine kanonische Darstellung, die für zwei Variablen a und b folgende Form besitzt:

$$y = f(b,a) = k_0 \oplus (k_1 \,\&\, a) \oplus (k_2 \,\&\, b) \oplus (k_3 \,\&\, a \,\&\, b)$$

B) Bestimmen Sie für die Koeffizientenwerte $k_0 = k_1 = k_3 = 1; \quad k_2 = 0$ die zugehörige Funktion. Geben Sie das Ergebnis als DNF an. Hinweis: Die Antivalenzoperation ist assoziativ!

$$y = 1 \oplus (1 \,\&\, a) \oplus (0 \,\&\, b) \oplus (1 \,\&\, a \,\&\, b) = 1 \oplus a \oplus 0 \oplus (a \,\&\, b)$$

$$= 1 \oplus a \oplus (a \,\&\, b)$$

$$= 1 \oplus \left(\bar{a}(a \,\&\, b) \vee a\overline{(a \,\&\, b)}\right) = 1 \oplus \left((\bar{a}a \,\&\, \bar{a}b) \vee a(\bar{a} \vee \bar{b})\right) = 1 \oplus \left((0 \,\&\, \bar{a}b) \vee (\bar{a}a \vee a\bar{b})\right)$$

$$= 1 \oplus \left(0 \vee a\bar{b}\right) = 1 \oplus \left(a\bar{b}\right) = \overline{a\bar{b}}$$

$$= \bar{a} \vee b$$

$$= \bar{a} \,\&\, 1 \vee b \,\&\, 1 = \bar{a}(b \vee \bar{b}) \vee b(a \vee \bar{a}) = \bar{a}b \vee \bar{a}\bar{b} \vee ab \vee \bar{a}b$$

$$= \bar{a}b \vee \bar{a}\bar{b} \vee ab$$

Aufgabe 6.2 Konvertierung

Vervollständigen Sie die Tabelle 6-1, indem Sie die offenen Felder durch Konvertierung ergänzen.

Stibitz	Binär	Hexadezimal	Dezimal
0100 0101 1000$_{Stibitz}$	1111101 $_B$	7D $_H$	125 $_{Dez}$
0101 1000 1010$_{Stibitz}$	100000001$_B$	101 $_H$	257 $_{Dez}$

Tabelle 6-1: Konvertierungstabelle

Aufgabe 6.3 BCD

Addieren Sie die im unten als BCD codierte Zahlen direkt im BCD Code. Stellen Sie ihren Lösungsweg – inklusive aller notwendiger Korrekturschritte -ausführlich dar. Geben Sie das Ergebnis R als BCD codierte Zahl und als Dezimalzahl an.

		0111	**0100**	**0110**	**1000**
+		**0011**	**0101**	**0101**	**1001**
		111 -		1111	
=		1010	1001	1100	0001
Korr. wegen:.		Ps		Ps.	Übertrag.
Zwischen-Ergebnis:		1010	1001	1100	0001
+		0110		0110	0110
		11---	--11	1- - -	
=	0001	0000	1010	0010	0111
Korr. wegen:			Ps		
=	0001	0000	1010	0010	0111
+			0110		
		---1	11--		
=	0001	0001	0000	0010	0111

R$_{BCD}$ = 0001 0001 0000 0010 0111 **R$_{Dez}$** = 11027

Aufgabe 7 Minimierung

Aufgabe 7.1 Petrickausdruck

Ohne Streichungsregeln anzuwenden, hat ein Entwickler aus der Überdeckungstabelle einer Schaltfunktion den folgenden Petrickausdruck gebildet:

$$PA = a \& (a \vee c \vee f) \& (b \vee c \vee f) \& (b \vee c \vee e \vee f) \& (d \vee e) \& (d \vee f)$$

Um den Ausdruck nicht vollständig ausdistribuieren zu müssen, soll zunächst die Überdeckungstabelle wiedergewonnen werden.

A) Ergänzen Sie die untenstehende Überdeckungstabelle entsprechend des gegebenen Petrickausdrucks, ohne diesen zu vereinfachen. Die überdeckenden Größen sind durch die Präsenzvariablen a, b, c, d, e und f gegeben. Die zu überdeckenden Größen E_i werden entsprechend der Reihenfolge wie Sie im Petrickausdruck auftauchen (links nach rechts), aufsteigend von E_1 bis E_n indiziert.

$p_i \backslash E_i$	E_1	E_2	E_3	E_4	E_5	E_6	
a	X	X					
b			X	X			
c		X	X	X			
d					X	X	
e				X	X		
f		X	X	X		X	

Tabelle 7-1: Überdeckungstabelle 1

Es sei nun folgende Überdeckungstabelle (Tabelle 7-2) gegeben:

$p_i \backslash E_i$	E_1	E_2	E_3	E_4	E_5	E_6	Kosten
a				X	X		1 €
b	X					X	2 €
c	X			X		X	3 €
d		X	X				4 €
e	X		X				5 €
f	X	X		X		X	6 €

Tabelle 7-2: Überdeckungstabelle 2

B) Bestimmen Sie in Tabelle 7-2 alle Kernspalten und markieren Sie die entsprechende(n) Zelle(n).

Kernspalte: E5

Streichbaren Spalten: (E5), E4

C) Wenden Sie nun die Spaltendominanzregeln an. Welche Spalte(n) können gestrichen werden? Geben Sie die entsprechenden Spalte(n) unten an.

Dominierende Spalte(n): E1

<u>Dominierte</u> Spalte(n): E6

Streichbare Spalte(n): E1

D) Wenden Sie nun die Zeilendominanzregeln an. Welche Zeile(n) können gestrichen werden? Geben Sie die entsprechenden Zeile(n) unten an.

Dominierende Zeile(n): d, (b) (f)

<u>Dominierte</u> Zeile(n): e, (c) (b)

Streichbare Zeile(n): e, (c) (b nicht streichbar)

E) Wenn neue Kerne entstanden sind geben sie die entsprechende Kernspalte an und markieren die entsprechende(n) Zelle(n) in Tabelle 7-2. Sind keine Kerne entstanden, so geben Sie den resultierenden Petrickausdruck an.

Kernspalte: E3

Streichbaren Spalten: (E3), E2

Petrickausdruck: b v (c) v f (Spalte E6)
Vollständiger Ausdruck wäre: (d v e)&(a)&(b v c v f) | (E3 v E5 v E6)

F) Geben Sie die kostenkünstigste Realisierung der in Tabelle 7-2 dargestellten Funktion an. Stellen Sie dazu den minimal notwendigen Petrickausdruck auf. Wie hoch sind die Kosten?

{a, b, d}, Die Kosten betragen 1€ + 2€ + 4€ = 7€

Aufgabe 7.2 KV Diagramm

Gegeben sei eine unvollständige Schaltfunktion y(d,c,b,a) .

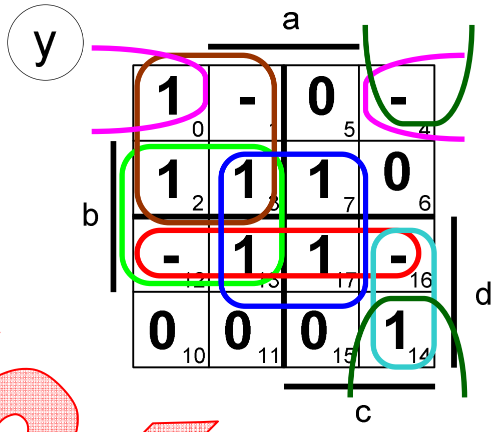


Abbildung 7–1: Symmetriediagramm

A) Geben Sie für die in Abbildung 7–1 gegebene Schaltfunktion die drei Belegungsmengen an. Verwenden Sie dazu die **oktale** Indizierung.

Einstellenmenge: E={0,2,3,7,13, 14, 17}

Nullstellenmenge: N={5, 6, 10. 11, 15}

Freistellenmenge: F={1, 4, 12, 16}

B) Ermitteln Sie nun die vollständige Einsblocküberdeckung τ_1 **aller** Prim-Einsblöcke der **Einsvervollständigung** f^E der Schaltfunktion y(d,c,b,a).

τ_1 (d, c, b, a) = { (- , 0 , 1 , -) (- , - , 1 , 1) (0 , 0 , - , -) (1 , - , 1 , -) (1 , 1 , - , 0) (- , 1 , 0 , 0) (0 , - , 0 , 0) }

C) Geben Sie nun alle Primterme an, die den in Teilaufgabe B) gefundenen Primblöcken entsprechen

Primimplikanten: $(\bar{c}b)$, (ba) , $(\bar{d}\,\bar{c})$, (db) , $(dc\bar{a})$, $(c\bar{b}\bar{a})$, $(\bar{d}\,\bar{b}\bar{a})$

D) Geben Sie nun eine vollständige Minimalüberdeckung der Funktion y(d,c,b,a) in disjunktiver Form an.

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

Zweite Lösungsmöglichkeit: $ab \vee \bar{c}\bar{d} \vee \bar{a}cd$

Aufgabe 8 Schaltnetze

Aufgabe 8.1 ISA Bussystem

Der Steuer-PC einer Industrieanlage soll um ein zusätzliches Touchscreen Display erweitert werden. Das neue Display wird, parallel zum alten Display des Steuer-PCs, über den vorhandenen ISA Bus angesprochen (siehe Abbildung 8–1). Die 16 Adressleitungen des ISA Bus(A15 – A0) stellen Adressen in binärer Form dar.
Beide Displays reagieren auf die Adressen 0x03C0-0x03DF, so dass ein einfaches Einbauen der neuen Hardware zu einem Adresskonflikt zwischen den beiden Displays führen würde. Das neue Display soll nun vom PC aus mit den Adressen 0x11C0 - 0x11DF angesprochen werden.

Dazu wird ein Schaltnetz vor das neue Display geschaltet. Diese kodiert den Adressbereich 0x11C0 - 0x11DF auf den Bereich 0x03C0-0x03DF um.

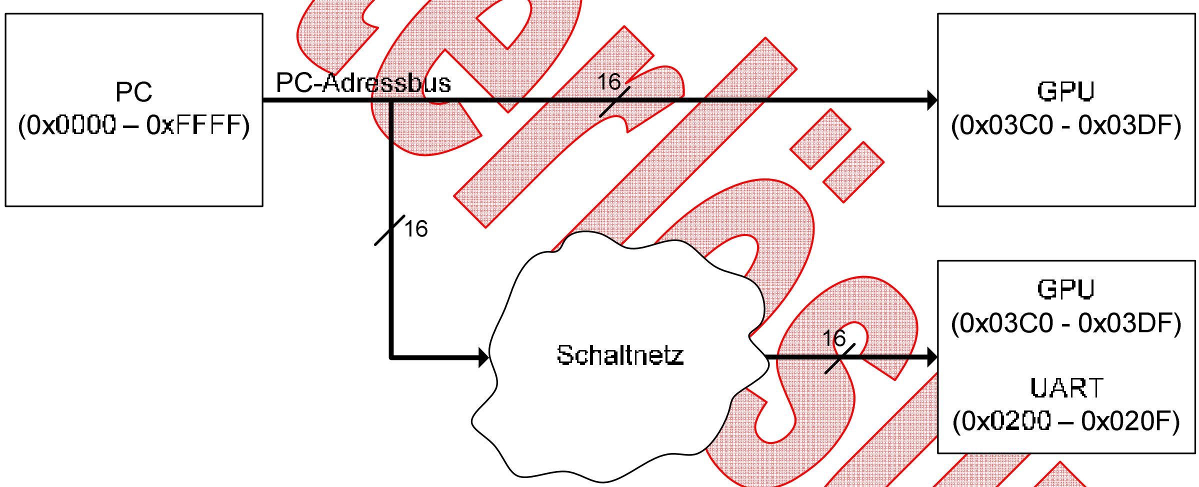


Abbildung 8–1: Displayansteuerung

A) Geben Sie die Adressen 0x11C0, 0x11DF, 0x03C0, 0x03DF in binärer Darstellung an.

0x11C0 = 0001 0001 1100 0000

0x11DF = 0001 0001 1101 1111

0x03C0 = 0000 0011 1100 0000

0x03DF = 0000 0011 1101 1111

B) Geben Sie die minimal mögliche Umverdrahtung der Adressleitungen an, mit der die Adressen von 0x11C0 - 0x11DF in den Bereich 0x03C0-0x03DF überführt werden können. (Bemerkung: Sie können davon ausgehen, dass die Karte Zugriffe außerhalb ihres Adressbereichs ignoriert.)

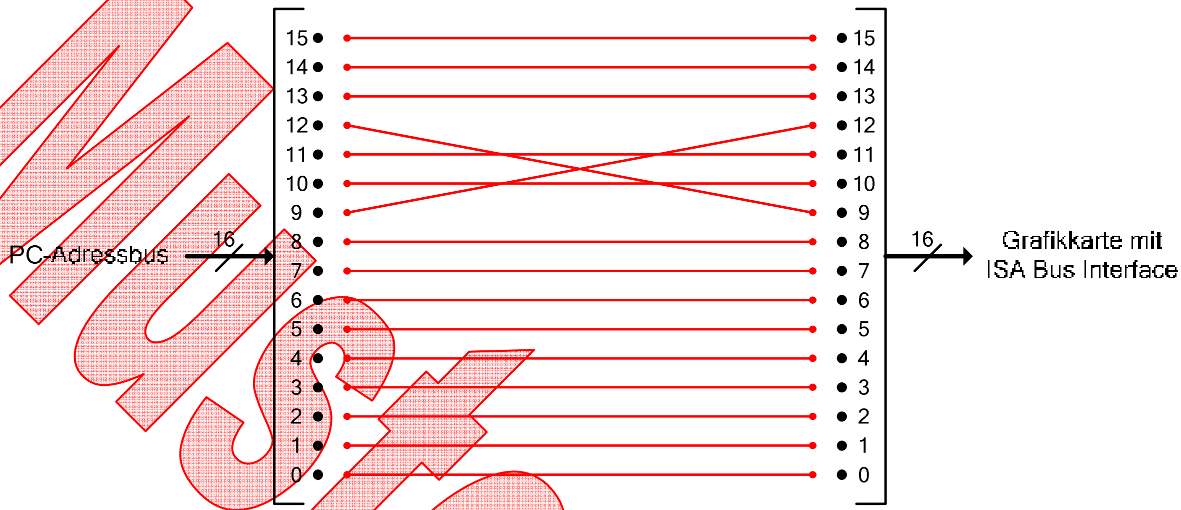


Abbildung 8–2: Verdrahtungsschema

Nach dem Kauf der neuen Grafikkarte stellt sich heraus, dass sich neben dem Grafikprozessor (GPU) noch weitere Bausteine zum Auslesen des Touchpanels auf der selben Platine befinden, deren Register sich im Bereich 0x3000 - 0x3FFF befinden und aufgrund des verwendeten Softwaretreibers nicht umgemappt werden dürfen.

C) Entwerfen Sie ein Schaltfunktion die über die 4 höchstwertigsten Bits (A15, A14, A13, A12) der binären Darstellung der Adresse eine logische ‚1‘ ausgibt wenn die anliegende Adresse im Bereich 0x3000 - 0x3FFF liegt. Liegt die die Adresse nicht in diesem Bereich soll die Funktion f(A15,A14,A13,A12) = sel eine logische ‚0‘ ausgeben. Füllen Sie das unten stehende KV Diagramm entsprechend aus.

sel

A12 (columns 2–3), A13 (rows 2–3), A14 (columns 3–4), A15 (rows 3–4)

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

D) Geben Sie die Schaltfunktion f(A15,A14,A13,A12) = sel an. Formen Sie die Schaltfunktion so um, dass nur NAND2 Gatter mit zwei Eingängen verwendet werden können.

$$sel = A12 \,\&\, A13 \,\&\, \overline{A14} \,\&\, \overline{A15}$$

$$= \overline{\overline{(A12 \,\&\, A13) \,\&\, \left(\overline{A14} \,\&\, \overline{A15}\right)}} = \overline{\overline{(A12 \,\overline{\&}\, A13)} \,\overline{\&}\, \overline{\left(\overline{A14} \,\overline{\&}\, \overline{A15}\right)}}$$

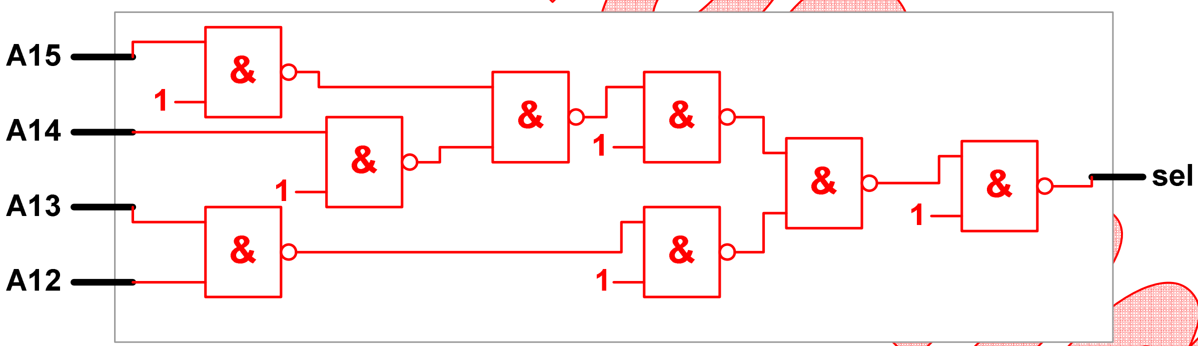
$$= \overline{\overline{(A12 \,\overline{\&}\, A13)} \,\overline{\&}\, \left(\overline{\overline{A14} \,\overline{\&}\, \overline{A15}}\right)} \quad ; \text{ mit } \overline{a} = a \overline{\&} 1$$

$$= \overline{\left((A12 \,\overline{\&}\, A13) \,\overline{\&}\, 1\right) \,\overline{\&}\, \left(\overline{(A14 \overline{\&} 1) \,\overline{\&}\, (A15 \overline{\&} 1)}\right)}$$

$$= \overline{\left((A12 \,\overline{\&}\, A13) \,\overline{\&}\, 1\right) \,\overline{\&}\, \left(\left((A14 \overline{\&} 1) \,\overline{\&}\, (A15 \overline{\&} 1)\right) \overline{\&} 1\right)}$$

$$= \left[\left((A12 \,\overline{\&}\, A13) \,\overline{\&}\, 1\right) \,\overline{\&}\, \left(\left((A14 \overline{\&} 1) \,\overline{\&}\, (A15 \overline{\&} 1)\right) \overline{\&} 1\right)\right] \overline{\&} 1$$

E) Zeichnen Sie das resultierende Schaltnetz aus NAND2 Gattern mit zwei Eingängen.



Aufgabe 8.2 PAL Realisierung

Die Gesamtschaltung in Abbildung 8–3 soll durch PAL Realisierungen ersetzt werden. Wenn das Signal sel = ‚1'ist, soll di Signalleitung A9 l mit der Signalleitung $A12_{out}$ und A12 mit $A9_{out}$ verbunden sein.
Gilt sel = ‚0', dann soll die Signalleitung A9 direkt mit der Signalleitung $A9_{out}$ und A12 direkt mit $A12_{out}$ verbunden sein. Die Realisierung des sel Signals ist durch $SN_{sel} = A15 \& A14 \& \overline{A13} \& \overline{A12}$ gegeben.

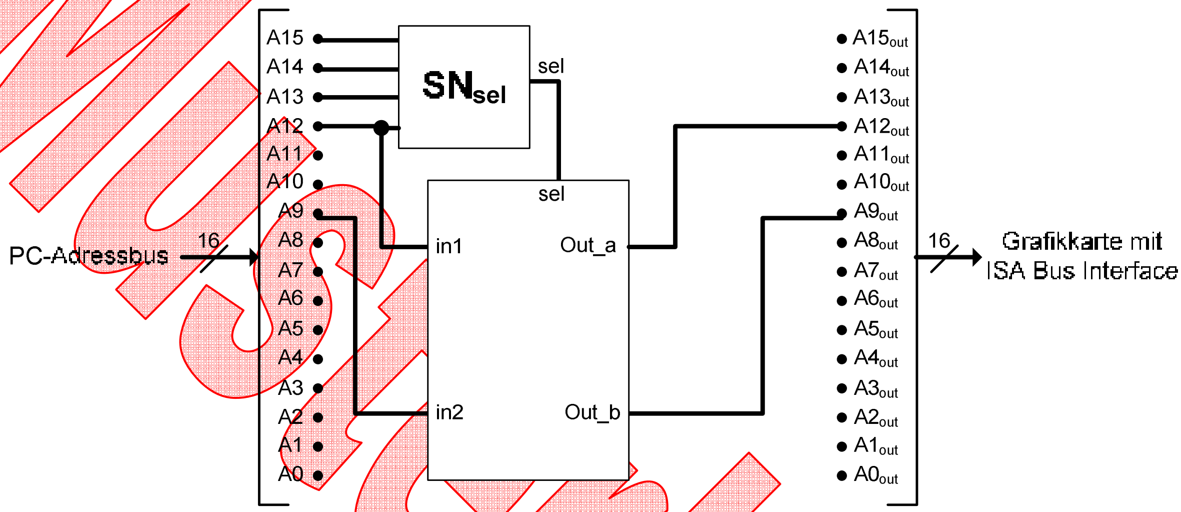


Abbildung 8–3: Umschaltschema

A) Stellen Sie die boolschen Gleichungen zur Bestimmung von $A9_{out}$ und $A12_{out}$ als disjunktive Minimalform (DMF) auf. Stellen Sie Ihren Lösungsweg ausführlich dar.

$$sel = A15 \,\&\, A14 \,\&\, \overline{A13} \,\&\, \overline{A12}$$

$$\overline{sel} = \overline{A15} \vee \overline{A14} \vee A13 \vee A12$$

$$A9_{out} = (sel \,\&\, A12) \vee (\overline{sel} \,\&\, A9)$$

$$= \left(A15 \,\&\, A14 \,\&\, \overline{A13} \,\&\, \overline{A12} \,\&\, A12\right) \vee \left(\left(\overline{A15} \vee \overline{A14} \vee A13 \vee A12\right) \,\&\, A9\right)$$

$$= (0) \vee \left(\overline{A15} \,\&\, A9\right) \vee \left(\overline{A14} \,\&\, A9\right) \vee (A13 \,\&\, A9) \vee (A12 \,\&\, A9)$$

$$= \left(\overline{A15} \,\&\, A9\right) \vee \left(\overline{A14} \,\&\, A9\right) \vee (A13 \,\&\, A9) \vee (A12 \,\&\, A9)$$

$$A12_{out} = (sel \,\&\, A9) \vee (\overline{sel} \,\&\, A12)$$

$$= \left(A15 \,\&\, A14 \,\&\, \overline{A13} \,\&\, \overline{A12} \,\&\, A9\right) \vee \left(\left(\overline{A15} \vee \overline{A14} \vee A13 \vee A12\right) \,\&\, A12\right)$$

$$= \left(A15 \,\&\, A14 \,\&\, \overline{A13} \,\&\, \overline{A12} \,\&\, A9\right) \vee \left(\overline{A15} \,\&\, A12 \vee \overline{A14} \,\&\, A12 \vee A13 \,\&\, A12 \vee A12\right)$$

$$= \left(A15 \,\&\, A14 \,\&\, \overline{A13} \,\&\, \overline{A12} \,\&\, A9\right) \vee A12$$

$$= \left(A15 \,\&\, A14 \,\&\, \overline{A13} \,\&\, A9\right) \vee A12$$

B) Die beiden Schaltnetze aus Aufgabenteil A) sollen nun mit Hilfe von PAL Bausteinen realisiert werden. Zeichnen Sie die Verbindungspunkte in das unten stehende Diagramm ein.