Digitaltechnik

Institut für Technik der Informationsverarbeitung – ITIV
Prof. Dr.-Ing. Dr. h. c. Jürgen Becker

Digitaltechnik
Datum: 15.03.2016
Name: Chuck Norris
Matr. Nr.: 10.03.1940
ID: ∞

Hörsaal: Ryan, Oklahoma
Platz: $\oint$

Hinweise zur Klausur

Hilfsmittel:

- Ein DIN A4 Blatt selbst geschriebene Formelsammlung.
- Verwenden Sie zum Bearbeiten der Aufgaben nur dokumentenechte Schreibgeräte – keinen Bleistift, keine Rotstifte!
- Alle nicht genannten Hilfsmittel sind untersagt. Dies beinhaltet jegliche Kommunikation mit anderen Personen sowie die Benutzung eines Taschenrechners.

Klausurdauer:
Die Prüfungsdauer für die Klausur beträgt 120 Minuten.

Klausurunterlagen:
Die Klausurunterlagen bestehen aus insgesamt 27 Seiten Aufgabenblättern (inklusive dieses Titelblatt, 8 Aufgabenblöcke und 1 zusätzliches Lösungsblatt) und 3 Seiten Formelsammlung. **Bitte prüfen Sie vor der Bearbeitung der Aufgaben auf jeder Seite oben Ihre Matrikelnummer sowie Ihre ID. Überprüfuen Sie zusätzlich Ihren Namen auf der ersten Seite.**
Falls Sie zusätzliche Blätter zur Lösung der Aufgaben benötigen, fragen Sie nach zusätzlichem Lösungspapier bei der Aufsicht. Vermeiden Sie generell das Beschreiben der Rückseiten. Die Verwendung von eigenen Blättern ist nicht erlaubt. Geben Sie zu jeder Aufgabe ein detaillierten Rechenweg an. Lösungen ohne Rechenweg können trotz richtigem Ergebnis zu Punktabzug führen.

Klausurabgabe:
In den letzten 30 Minuten der Klausur ist eine vorzeitige Abgabe der Klausur nicht möglich. Am Ende der Klausur bleiben Sie bitte sitzen. Alle Aufgaben- und Lösungsblätter sowie dieses Deckblatt sind in den ausgehändigten Umschläge abzugeben. Diese werden von der Aufsicht eingesammelt.

Chapter 1: Codierung

Aufgabe 1.1: Blocksicherung

Eine wichtige Nachricht im 7-Bit-ASCII-Code soll für die Übertragung über eine fehleranfällige Strecke mit einer Blocksicherung mit doppelter Paritätsprüfung versehen werden.

A) Vorbereitend wurden die Codeworte der Nachricht ohne Paritätsbits und Prüfsumme in den Datenblock in Tabelle 1.1 eingetragen. Dekodieren Sie die zu übermittelnden Zeichen unter Zuhilfenahme einer 7-Bit-ASCII-Code-Tabelle und vervollständigen Sie somit die Spalte Zeichen in der Tabelle.

Zeichen	**Codeworte**							**Parität**
R	1	0	1	0	0	1	0	
	1	1	0	1	0	0	1	
	1	1	0	0	0	1	1	
	1	1	0	1	0	0	0	
	1	1	1	0	1	0	0	
	1	1	0	1	0	0	1	
	1	1	0	0	1	1	1	

Tabelle 1.1: Blocksicherung: Zu versendende Daten

B) Sender und Empfänger haben sich auf ungerade (odd) Parität und eine Blocklänge von 7 Zeichen geeinigt. Ergänzen Sie den zu versendenden Datenblock in Tabelle 1.1 entsprechend um Paritätsbits und Prüfwort.

C) Dieser Block wird nun spaltenweise (scrambling) mit einer Übertragungsrate von 32.000 bit/s übertragen. Wie lang darf eine Störung maximal dauern, damit sie sicher erkannt werden kann? Geben Sie die Lösung und Rechnung an.

D) Im weiteren Verlauf der Kommunikation zwischen Sender und Empfänger hat der Empfänger den in Tabelle 1.2 dargestellten Block empfangen. Überprüfen Sie den empfangenen Block. Sind bei der Übertragung Fehler aufgetreten? **Gehen Sie davon aus, dass das Prüfwort korrekt übermittelt wurde.** Falls Sie Übertragungsfehler finden, geben Sie die Anzahl der Übertragungsfehler an und markieren Sie die falsch empfangenen Bits in der Tabelle.

Zeichen	**Codeworte**							**Parität**
	1	0	0	0	1	1	0	0
	1	1	1	0	0	0	1	0
	1	1	0	1	1	0	0	1
	1	1	1	0	0	1	1	0
	1	1	0	0	0	1	1	1
	1	1	0	1	0	0	0	0
	0	1	0	0	0	0	1	1
Prüfwort	1	1	0	1	1	0	1	0

Tabelle 1.2: Blocksicherung: Empfangene Daten

E) Welche minimale Hamming-Distanz (HDmin) wird mindestens benötigt, um einen Code mit 3-facher Fehlerkorrektur zu erhalten?

F) Wie viele Fehler ließen sich mit der in Aufgabenteil E) bestimmten minimalen Hamming-Distanz (HDmin) erkennen?

Aufgabe 1.2: BCD-Code

A) 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.

Aufgabe 1.3: Digitalisierung

A) Ein analoges Spannungssignal im Wertebereich 0 V bis 5 V soll digitalisiert werden, wobei das digitale Signal eine Auflösung (Intervallgröße) von 1 V haben soll. Welche minimale Bitbreite (Anzahl der Bits für die Codierung der unterschiedlichen Digitalwerte) müsste ein Analog/Digital-Wandler hierfür haben.

B) Nennen Sie einen Vorteil, den undefinierte Bereiche bei der Digitalisierung von analogen Signalen bieten.

Chapter 2: Mengen, Relationen und Graphen

Aufgabe 2.1: Mengen

Gegeben seien folgende Mengen

$$M_1 = \{2, 6\}$$
$$M_2 = \{x | x \text{ Primzahl, } x > 2, x < 10\}$$
$$M_3 = \left\{\frac{x}{2} | x \text{ durch 4 teilbar, }, x > 0, x < 21\right\}$$

A) Geben Sie die Elemente der Mengen M2 und M3 an

B) Zeichnen Sie für jede Menge M ein Gebiet und tragen in jedes Gebiet die zugehörigen Elemente ein.

C) Bestimmen Sie die folgenden Mengen und Größen:

$$M_1 \cap M_2 =$$
$$|M_1 \cup M_2| =$$
$$|M_1 \times M_2 \times M_3| =$$

Aufgabe 2.2: Relationen

A) Modifizieren Sie den Graphen in Abbildung 2.1 derart, dass dieser reflexiv ist.

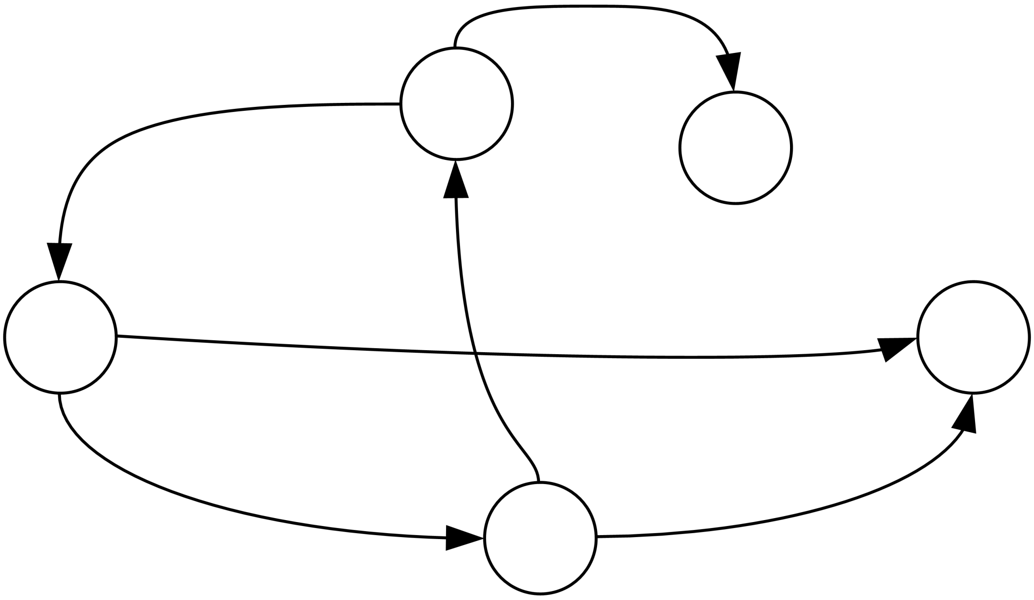


Abbildung 2.1: Reflexiver graph

B) Ist der Graph in Abbildung 2.1 symmetrisch? Begründen Sie Ihre Antwort.

Aufgabe 2.3: Graphentheorie

Gegeben sei der Abbildung 2.2 gezeigte Graph

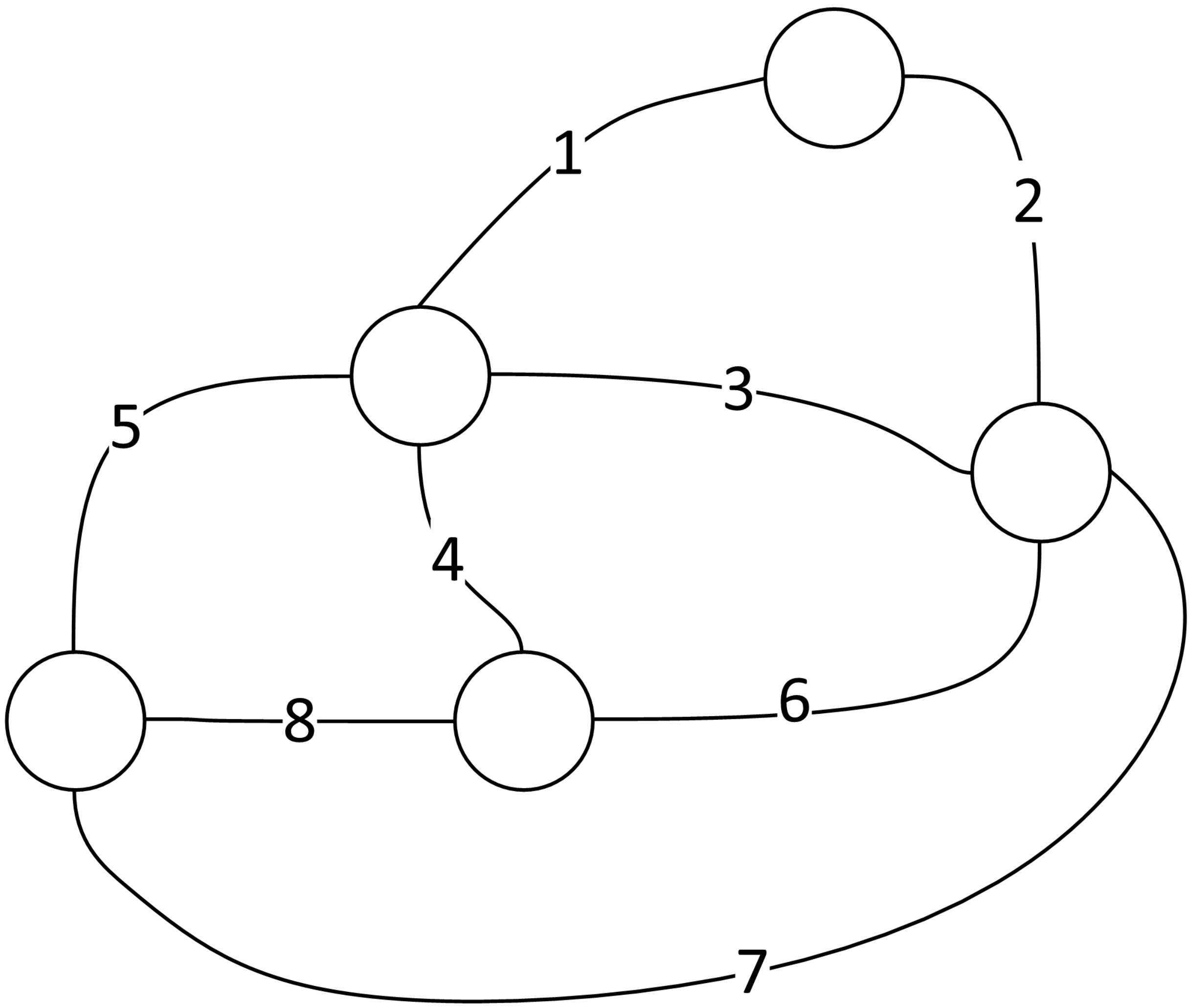


Abbildung 2.2: Kettenprogression und dualer Graph

A) Zeigen Sie, dass es eine Kettenprogression gibt, die sämtliche Kanten enthält.

B) Konstruieren Sie zum Graphen in Abbildung 2.2 einen dualen Graphen. Benennen/kennzeichnen Sie die entsprechenden Gebiete direkt in Abbildung 2.2 und zeichnen Sie in Abbildung 2.2 den dualen Graphen ein.

Chapter 3: Boolesche Algebra

A) Zeigen Sie mit Boolescher Algebra, dass Ausdruck A gleich Ausdruck B ist.

$A = (a\overline{b}c\overline{d}) \vee (a\overline{b}cd) \vee (abcd) \vee (\overline{\overline{a} \vee \overline{b} \vee \overline{c} \vee d})$

$B = \overline{(\overline{a} \vee \overline{c})}$

B) Zeigen Sie, dass das Absorptionsgesetz gilt: $a(a \vee b) = a$

C) Gegeben sei folgende boolesche Funktion: $Y = (ab\overline{c}d) \vee (a\overline{b}d) \vee (\overline{a}c\overline{d}) \vee (a\overline{b}\overline{c}d)$
Entwickeln Sie den Ausdruck Y mit Hilfe des Booleschen Entwicklungssatzes in der Reihenfolge d, c, b, a. Geben Sie alle Zwischenergebnisse an und tragen sie das Ergebnis in die unten stehende Tabelle 3.1 ein. Eine Umformung des Ausdrucks ist nicht gestattet.

(d, c, b, a)	Y	(d, c, b, a)	Y
$(0, 0, 0, 0)$		$(1, 0, 0, 0)$	
$(0, 0, 0, 1)$		$(1, 0, 0, 1)$	
$(0, 0, 1, 0)$		$(1, 0, 1, 0)$	
$(0, 0, 1, 1)$		$(1, 0, 1, 1)$	
$(0, 1, 0, 0)$		$(1, 1, 0, 0)$	
$(0, 1, 0, 1)$		$(1, 1, 0, 1)$	
$(0, 1, 1, 0)$		$(1, 1, 1, 0)$	
$(0, 1, 1, 1)$		$(1, 1, 1, 1)$	

Tabelle 3.1: Ergebnis des Entwicklungssatzes

D) Leiten Sie aus der folgenden Wahrheitstabelle die dazugehörige DNF für $Z(a, b, c)$ ab.

a	b	c	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

E) Für die Realisierung der Funktion $Z = (\overline{a}bc) \vee (a\bar{b}\overline{c})$ stehen in der Praxis nur NAND-Gatter mit zwei Eingängen zur Verfügung. Geben Sie die umgeformte Funktion $Z(a, b, c)$ an, bei der nur NAND-Gatter mit zwei Eingängen verwendet werden.

Chapter 4: Zahlensysteme und Codierung

Aufgabe 4.1: Umrechnung von Zahlensystemen

A) Vervollständigen Sie die Tabelle 4.1, indem Sie die offenen Felder durch die entsprechende Konvertierung ergänzen.

Dezimal	**Binär**	**Oktal**	**Hexadezimal**
1077_D			
	$101\ 0011\ 0110_B$		
		1275_O	
			$AB3_H$

Tabelle 4.1: Umrechnung von Zahlensystemen

B) Wandeln Sie die Zahl $A6_{\ 15}$ (Basis 15) in das Duodezimalsystem (Basis 12) um. Geben Sie die Zwischenschritte an.

Aufgabe 4.2: Rechenoperationen im Binärsystem

A) Abbildung 4.1 zeigt zwei normierte 16-Bit Fließkommazahlen. Das höchstwertige Bit stellt dabei das Vorzeichen V dar, die nachfolgenden sechs Bits den Exponenten E und die niederwertigsten neun Bits die Mantisse M. Addieren Sie die zwei in der normierten 16-Bit-Fließkommazahlendarstellung gegebenen Zahlen miteinander. Geben Sie das Ergebnis ebenfalls in dieser Darstellung an. Geben Sie die Zwischenschritte bei der Umrechnung an.

	V	E_5	E_4	E_3	E_2	E_1	E_0	M_8	M_0	M_6	M_5	M_4	M_3	M_2	M_1	M_0
	1	0	1	1	1	1	0	0	0	0	1	1	0	0	0	1
+	1	1	0	0	0	1	0	1	1	1	0	0	0	0	1	0
=																

Abbildung 4.1: Addition von zwei 16-Bit-Fließkommazahlen

B) Führen Sie eine Subtraktion der Zahl 16_D von der Zahl 53_D ($53_D - 16_D$) im STIBITZ-Code durch. Stellen Sie Ihren Lösungsweg ausführlich dar.

Chapter 5: Minimierung

Aufgabe 5.1: Symmetriediagramm

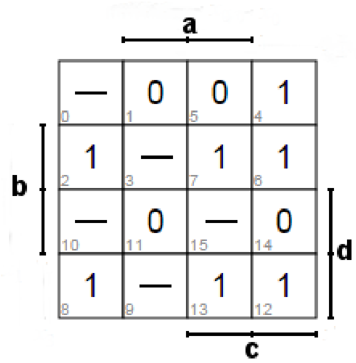


Abbildung 5.1: Symmetriediagramm

A) Bestimmen Sie die konjunktive Normalform (KNF) aus dem Symmetriediagramm in Abbildung 5.1.

B) Bestimmen Sie aus dem Symmetriediagramm in Abbildung 5.2 (identisch zu Aufgabe A)) zwei unterschiedliche disjunktive Minimalformen (DMF) der Schaltfunktion, die sich mindestens in einem Primimplikanten unterscheiden. Zur grafischen Hilfestellung wurde das Symmetriediagramm zweimal abgebildet.

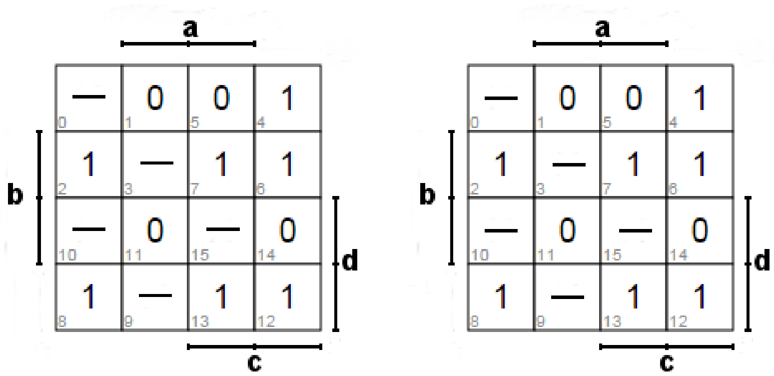


Abbildung 5.2: Symmetriediagramm

Aufgabe 5.2: Nelson/Petrick Verfahren

A) Gegeben ist der folgende Petrickausdruck:
$PA = (b \vee d \vee g)\ \&(a \vee e \vee g)\ \&(b \vee d)\ \&(e \vee g)\ \&(d)\ \&(a \vee f)\ \&(c \vee d \vee g)\ \&(a \vee e \vee f)$

Ergänzen Sie die untenstehende Überdeckungstabelle (Tabelle 5.1) entsprechend des gegebenen Pertickausdruck, ohne diesen zu vereinfachen. Die überdeckenden Größen sind durch die Präsenzvariablen a, b, c, d, e, f und g 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_8 indiziert. Bestimmen Sie anschließend alle Kerne und markieren Sie die entsprechende(n) Zelle(n). Geben Sie die streichbaren Spalte(n) an.

p_i/E_i	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8
a								
b								
c								
d								
e								
f								
g								

Tabelle 5.1: Überdeckungstabelle 1

B) Wenden Sie die Spaltendominanzregel auf Tabelle 5.2 an. Welche Spalte(n) können gestrichen werden? Streichen Sie die entsprechende(n) Spalte(n) und geben Sie die dominierte(n) und zugehörigen dominierende(n) Spalte(n), sowie die streichbare(n) Spalte(n) in der Lösungstabelle 5.3 an.

p_i/E_i	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9
a		X		X	X		X		
b	X				X				
c			X	X		X	X		
d	X			X	X				X
e		X							
f		X	X			X		X	
g					X	X		X	X

Tabelle 5.2: Überdeckungstabelle 2

Dominierende Spalte(n):						
Dominierte Spalte(n):						
Streichbare Spalte(n):						

Tabelle 5.3: Lösungstabelle

C) Erläutern Sie kurz die jeweiligen Ziele der Verfahren nach Nelson und Petrick. Welches Ergebnis liefert die Anwendung beider Verfahren?

Chapter 6: Optimale Codes

Zur Auswertung und Optimierung einer bekannten Smartphone-App, die es ermöglicht, anonyme Nachrichten an Personen im Umkreis zu versenden, sollen die Themen der Nachrichten analysiert werden. Hierzu werden die Themen der Nachrichten entsprechend gespeichert. Um diese Daten möglichst effizient speichern und übertragen zu können, soll hierfür eine optimale Codierung entwickelt werden. In Tabelle 6.1 ist die Anzahl der Nachrichten pro Thema angegeben.

Thema	**Anzahl Nachrichten pro Tag**	**Ermittelte Codierung**
A: Mittagessen	55	
B: Witze	75	
C: DT Vorlesung	85	
D: DT Übung	1	
E: Partys	80	
F: KVV Kontrollen	5	
G: Leben in Karlsruhe	25	
H: Weisheiten	8	
Gesamt	334	-

Tabelle 6.1: Nachrichtenaufkommen pro Thema

A) Zunächst soll untersucht werden, welche mittlere Codewortlänge sich für eine Codierung ergibt, falls alle Codewörter binär und mit gleicher Länge codiert werden. Geben Sie dafür die Berechnungsvorschrift sowie die ermittelte minimale mittlere Codewortlänge an.

B) Bestimmen Sie eine optimale Codierung nach dem Huffman Verfahren für die Auftrittshäufigkeiten aus Tabelle 6.1 und tragen Sie diese anschließend in Tabelle 6.1 (Spalte „Ermittelte Codierung“) ein. Der Lösungsweg muss nachvollziehbar sein und der vollständige Codierbaum angegeben werden! (Hinweis: **linke Äste mit ’0’, rechte Äste mit ’1’ codieren; Knoten mit höherer Wahrscheinlichkeit links**)

C) Geben Sie die Formel zur Berechnung der mittleren Codewortlänge an. Geben Sie anschließend den vollständigen Rechenweg für die mittlere Codewortlänge für die im Aufgabenteil B) entwickelte Codierung an. Das Endergebnis muss nicht ausgerechnet werden.

D) Anhand welcher Quelleneigenschaft kann die Effizienz der gefundenen Codierung beurteilt werden? Geben Sie deren Namen, die Formel zur Berechnung sowie den vollständigen Rechenweg an, das Endergebnis muss nicht ausgerechnet werden.

E) Die gesammelten Daten werden täglich an ein zentrales System übermittelt. Berechnen Sie das durchschnittliche Datenvolumen dieser Übertragung pro Woche in Byte. Gehen Sie dabei von einer mittleren Codewortlänge von 3 Bit aus. Geben Sie den vollständigen Rechenweg an, das Endergebnis muss nicht ausgerechnet werden.

Chapter 7: Automaten

Aufgabe 7.1: Automatenentwurf

In dieser Aufgabe soll ein Zustandsautomat für die Ansteuerung einer selbstschließenden Omnibus-Türe entworfen werden. Das Öffnen und Schließen der Tür kann jeweils durch das Drücken eines entsprechenden Tasters ausgelöst werden. Um zu erkennen, ob die Türe vollständig geöffnet oder geschlossen ist, gibt es je einen entsprechenden Sensor. Um das Einklemmen von Personen und Gegenständen zu verhindern verfügen die Aktuatoren zum Schließen der Tür über eine Kraftbegrenzung. Für den Zustandsautomaten sind insgesamt die folgenden Eingangs- und Ausgangsbits vorhanden:

Eingänge:

TO**:** Der Taster zum Öffnen der Tür wird gedrückt.

TC**:** Der Taster zum Schließen der Tür wird gedrückt.

OP**:** Die Tür ist vollständig geöffnet

CL**:** Die Tür ist vollständig geschlossen

FM**:** Die Kraftbegrenzung der Aktuatoren ist aktiv

Ausgänge:

AO**:** Bei gesetztem Bit wird der Aktuator zum Öffnen der Tür aktiv.

AC**:** Bei gesetztem Bit wird der Aktuator zum Schließen der Tür aktiv.

Zustände:

$S_{closing}$**:** Der Schließvorgang der Türe läuft

$S_{opening}$**:** Der Öffnungsvorgang der Türe läuft

S_{idle}**:** Die Aktuatoren der Türe sind nicht aktiv

Es soll nun ein geeigneter Zustandsautomat entworfen werden, der folgende Anforderungen erfüllt

- Im Zustand S_{idle} soll ein Tastendruck die Türe je nach gedrücktem Taster öffnen oder schließen.
- Die Aktuatoren sollen abgeschaltet werden sobald die Tür nach einem Öffnungs- oder Schließvorgang den jeweiligen Endzustand (vollständig geschloßen bzw. geöffnet) erreicht.
- Wird die Kraftbegrenzung beim Schließvorgang aktiv, so soll von eingeklemmten Personen/Gegenständen ausgegangen werden.
- Beim Erkennen eingeklemmter Personen/Gegenstände soll sich die Türe wieder vollständig öffnen.

A) Die Ansteuerung soll als Medwedew-Automat realisiert werden. Was zeichnet einen Medwedew-Automaten aus? Wie müssen die vorgegebenen Zustände $S = (S_0, S_1)$ binär kodiert werden um die korrekte Ausgabe $A = (A_0, A_1) := (AO, AC)$ bestehend aus den Ansteuerbits AO, AC für die Aktuatoren zu generieren? Tragen sie die Kodierung in die vorgegebene Tabelle ein.

Zustand	Zustandsbits	
	S_0	S_1
$S_{closing}$	0	1
$S_{opening}$	1	0
S_{idle}	0	0

B) Entwerfen Sie das Ablaufdiagramm für den gesuchten Zustandsautomaten entsprechend der vorgegebenen Anforderungen.

Aufgabe 7.2: Realisierung von Automaten mit FlipFlops

A) Der Zustandsspeicher des Automaten aus Tabelle 7.1 soll mit einem *RS-FlipFlop* (mit den Eingängen s_0 und r_0) für das erste Bit S_0 und einem *JK-FlipFlop* (mit den Eingängen j_0 und k_0) für das zweite Bit S_1 realisiert werden.
Ergänzen Sie in der Ablauftabelle die fehlenden Ansteuerbits für die Eingänge r_0 und k_0 der FlipFlops. Verwenden Sie nach Möglichkeit „dont't care" Stellen.

Zustand $S^v = (S_0^v, S_1^v)$	Eingabe E^v	Folgezustand S^{v+1}	Ausgabe A	FlipFlop Ansteuerung s_0	r_0	j_0	k_0
00	00	00	1	0	-	0	-
	01	01	1	0	-	1	-
	10	10	1	1	0	0	-
	11	11	0	1	0	1	-
01	00	10	1	1	0	-	1
	01	01	1	0	-	-	0
	10	11	0	1	0	-	0
	11	00	1	0	-	-	1
10	00	11	0	-	0	1	-
	01	01	1	0	1	1	-
	10	10	1	-	0	0	-
	11	00	1	0	1	0	-
11	00	00	0	0	1	-	1
	01	01	0	0	1	-	0
	10	10	0	-	0	-	1
	11	11	1	-	0	-	0

Tabelle 7.1: Ablauftabelle eines Zustandsautomaten

s_0 (Symmetriediagramm: S_0^v oben, S_1^v links, E_0 rechts, E_1 unten)

1	0	0	0
1	-	-	0
1	-	0	1

j_0 (Symmetriediagramm: S_0^v oben, S_1^v links, E_0 rechts, E_1 unten)

-	-	-	-
-	-	-	-
0	0	0	1

B) Die Ansteuerfuntionen für die FlipFlops sollen nun minimiert werden. Geben Sie mit Hilfe der vorgegebenen Symmetriediagramme (Seite 23) jeweils eine minimale Ansteuerfunktion für die beiden Eingänge s_0 und j_0 an.

C) Welchen Vorteil hat das JK-FlipFlop bei der Minimierung der Ansteuerfunktion gegebüber dem RS-FlipFlop? Begründen Sie ihre Antwort.

Chapter 8: CMOS und Gatter

Aufgabe 8.1: Gatter-Schaltungen

A) Geben sie die Funktion der gegebenen Schaltung in boolescher Algebra an.

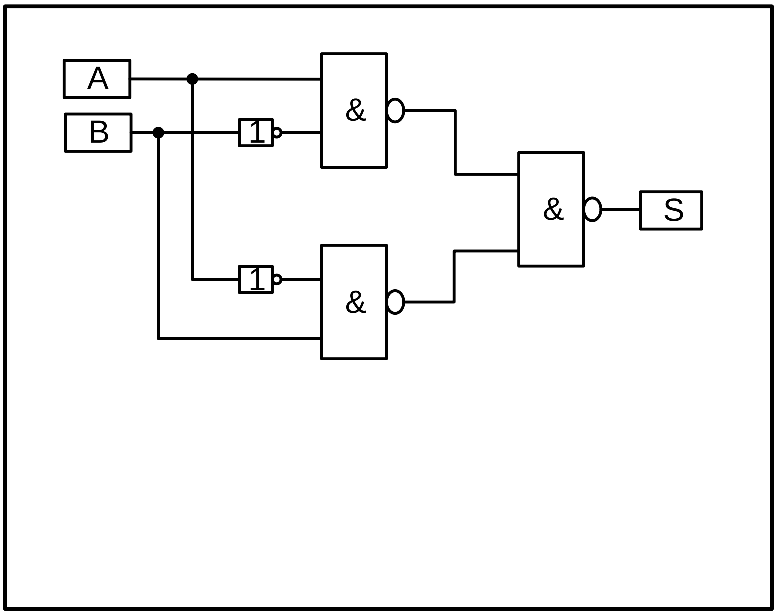


Abbildung 8.1: Gatter-Schaltung

B) Erweitern sie die Gatterschaltung aus Abbildung.8.1 zu einem Halbaddierer. Ein Halbaddierer gibt an Ausgang *Cout* eine 1 aus wenn bei der Addition ein Übertrag entsteht, ansonsten eine 0. Verwenden sie dazu nur NAND- und Inverter-Gatter. Geben sie zusätzlich die Funktion der hinzugefügten Schaltung in boolescher Algebra an.

Aufgabe 8.2: Analyse einer CMOS-Schaltungen

Gegeben sei das nicht wohldefinierte CMOS-Schaltnetz:

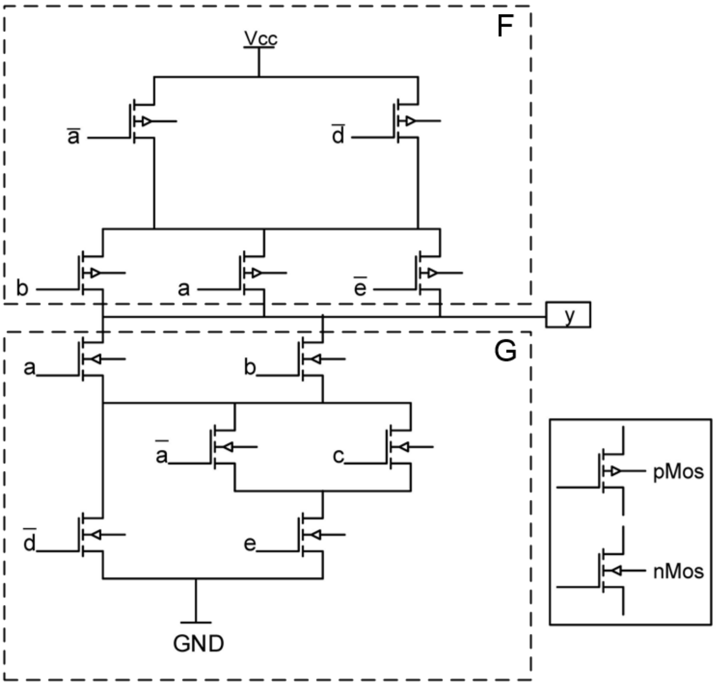


Abbildung 8.2: CMOS-Schaltnetz

A) Geben Sie für die vorliegende Schaltung aus Abbildung 8.2 sowohl die pull-down-Funktion G als auch die pull-up-Funktion F an.

B) Gegeben sind folgende pull-up- bzw. pull-down-Funktionen F und G. Stellen Sie mit Hilfe boolescher Algebra fest, ob die durch die beiden Funktionen definierte CMOS Schaltung einen Kurzschluss enthält. Geben Sie die Eingangsbelegungen an, welche zu Kurzschlüssen führen.

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

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

Aufgabe 8.3: Realisierung einer CMOS-Schaltungen

Gegeben Sei nun die pull-up Funktion eines CMOS-Schaltnetzes: $F = (\bar{a}d + cd)(\bar{c} + b)$

A) Geben Sie die für eine wohldefinierte CMOS-Schaltung nötige Abhängigkeit zwischen G und F an. Geben Sie die pull-down-Funktion G so an, dass diese als Transistorschaltnetz realisiert werden kann.

B) Zeichnen Sie das pull-down Schaltnetz der Funktion G aus Aufgabenteil A) mit minimaler Anzahl an Transistoren in Abbildung.8.3 ein.

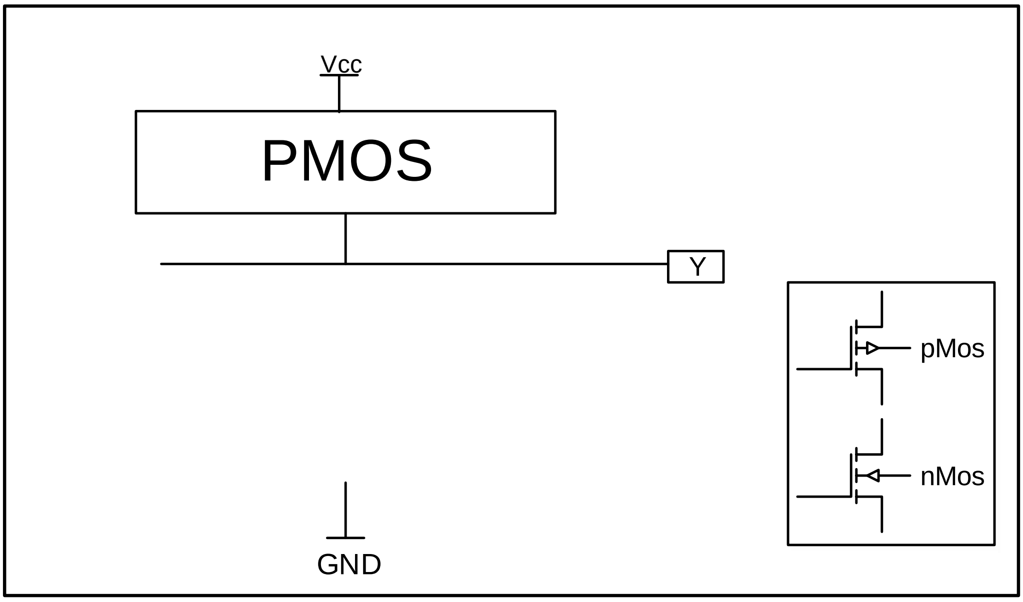


Abbildung 8.3: CMOS-Schaltnetz

Zusätzliches Lösungsblatt:

Formelblatt Digitaltechnik

Huntingtonschen Axiome für alle $a,b,\mathrm{I},\mathrm{O} \in K\,;\, \overline{a} = k \in K$

Abgeschlossenheit:	(H1)	$a \top b \in K$	$a \perp b \in K$
Kommutativgesetz:	(H2)	$a \top b = b \top a$	$a \perp b = b \perp a$
Distributivgesetz:	(H3)	$(a \top b) \perp c = (a \perp c) \top (b \perp c)$	$(a \perp b) \top c = (a \top c) \perp (b \top c)$
Neutrales Element:	(H4)	$\mathrm{I} \top a = a$	$\mathrm{O} \perp a = a$
Komplement:	(H5)	$a \top k = \mathrm{O}$	$a \perp k = \mathrm{I}$

Abgeleitete Regeln

R1a:	$\overline{0} = 1$	R1b:	$\overline{1} = 0$
R2a:	$0 \vee 0 = 0$	R2b:	$1 \,\&\, 1 = 1$
R3a:	$1 \vee 1 = 1$	R3b:	$0 \,\&\, 0 = 0$
R4a:	$1 \vee 0 = 1$	R4b:	$1 \,\&\, 0 = 0$
R5a:	$a \vee 0 = a$	R5b:	$a \,\&\, 0 = 0$
R6a:	$a \vee 1 = 1$	R6b:	$a \,\&\, 1 = a$
R7a:	$a \vee a = a$	R7b:	$a \,\&\, a = a$
R8a:	$a \vee \overline{a} = 1$	R8b:	$a \,\&\, \overline{a} = 0$
R9:	$\overline{(\overline{a})} = \overline{\overline{a}} = a$		

Assoziative Gesetze:

R10a: $(a \vee b) \vee c = a \vee (b \vee c) = a \vee b \vee c$

R10b: $(a \,\&\, b) \,\&\, c = a \,\&\, (b \,\&\, c) = a \,\&\, b \,\&\, c$

Absorptionsgesetze:

R11a: $(a \vee b) \,\&\, a = a$

R11b: $(a \,\&\, b) \vee a = a$

De Morgan:

R12a: $\overline{(a \vee b)} = \overline{a} \,\&\, \overline{b}$

R12b: $\overline{(a \,\&\, b)} = \overline{a} \vee \overline{b}$

Umwandlung DNF ↔ KNF

$$OR\left(AND(l_i)\right) \xleftrightarrow{R9} OR\left(AND(\overline{\overline{l_i}})\right) \xleftrightarrow{R12a} OR\left(\overline{OR(\overline{l_i})}\right) \xleftrightarrow{R12b} \overline{AND\left(OR(\overline{l_i})\right)} \xleftrightarrow{H3}$$

$$\overline{OR\left(AND(\overline{l_k})\right)} \xleftrightarrow{R12a} AND\left(\overline{AND(\overline{l_k})}\right) \xleftrightarrow{R12b} AND\left(OR(\overline{\overline{l_k}})\right) \xleftrightarrow{R9} AND\left(OR(l_k)\right)$$

Weitere Funktionen

Implikation $a \rightarrow b = \overline{a} + b$

Äquivalenz $a \equiv b = a \cdot b + \overline{a} \cdot \overline{b}$

Inklusion $f \leq g \Rightarrow f \cdot \overline{g} = 0$

Transitivität der Inklusion

$$(f \leq g) \cdot (g \leq h) \Rightarrow f \leq h$$

Tautologie $f = g \Leftrightarrow f \equiv g = 1$

$$f \leq g \Leftrightarrow \overline{f} + g = 1$$

XOR-Regeln $x \oplus y = \overline{x} \cdot y + x \cdot \overline{y}$

$$x + y = x \cdot y \oplus x \oplus y$$

$$x \cdot (y \oplus z) = x \cdot y \oplus x \cdot z$$

$$\overline{x} = x \oplus 1,\ x \oplus x = 0,\ x \oplus 0 = x$$

Multiplexer $f = x \cdot a + \overline{x} \cdot b$

Entwicklungssatz

Boolescher Entwicklungssatz $f = x \cdot f_x + \overline{x} \cdot f_{\overline{x}}$

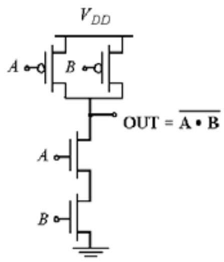
Dualer Entwicklungssatz $f = (\overline{x} + f_x) \cdot (x + f_{\overline{x}})$

CMOS Schaltungen

CMOS (wohl definiert) $v_1 = \overline{v_0}$

CMOS (kein Kurzschluss) $v_1 \cdot v_0 = 0$

CMOS (Vollständigkeit) $v_1 + v_0 = 1$



Addierer

Halbaddierer $s_i = a_i \oplus b_i$ $c_{i+1} = a_i \cdot b_i$

Volladdierer $s_i = a_i \oplus b_i \oplus c_i$

$$c_{i+1} = a_i \cdot b_i + (a_i \oplus b_i) \cdot c_i$$

Carry-Look-ahead $c_{i+1} = g_i + p_i \cdot c_i$

Generate $g_i = a_i \cdot b_i$

Propagate $p_i = a_i \oplus b_i$

Informationsgehalt

Informationsgehalt H_e eines Zeichens: $H_e = \mathrm{ld}\,\frac{1}{p}$

Informationsgehalt H einer Quelle: $H = \sum_{i=1}^{N} p(i) \cdot \mathrm{ld}\,\frac{1}{p(i)}$

mit der Auftrittswahrscheinlichkeit $p(i)$ und $\sum_{i=1}^{N} p(i) = 1$

a,b,x,y,z : boolesche Variablen l : Literal f,g : boolesche Funktionen v_1 : p-Netz v_0 : n-Netz
s : Summe/Zustand c : Carry i : Eingang δ : Transitionsfunktion λ : Ausgabefunktion

Codierung

Allgemeiner Aufbau einer polyadischen Zahl:

$$N \;=\; dn * Rn + \ldots + d1 * R1 \;+\; d0 * R0$$

mit *N* Zahl im Zahlensystem; *R* Basis; R^i Wertigkeit; d_i Ziffer der i-ten Stelle; *Z* Menge der Ziffer $d_i \in \{0, 1, 2, \ldots, R-1\}$

ASCII-Tabelle

LSB	MSB							
Binär	000	001	010	011	100	101	110	111
	Steuerzeichen			Großbuchstaben			Kleinbuchstaben	
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	„	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	´	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Anzahl Codewörter im (*k* aus m)-Code:

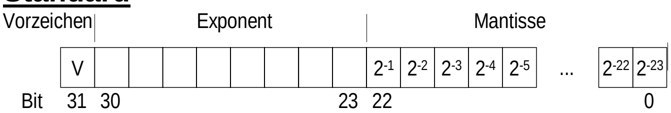
$$\binom{m}{k} = \frac{m!}{k!(m-k)!}$$

Korrigierbare Fehleranzahl bei ungerader HD:

$$F_k = \frac{(d-1)}{2}$$

Erkennbare Fehleranzahl: $F_e = d - 1$

Gleitkommadarstellung gemäß IEEE 754-Standard



Exponent E	Mantisse M	Wert
255	≠0	ungültig (NAN)
255	0	$-1^V \cdot \infty$ (±unendlich)
$0 < E < 255$	M	$-1^V \cdot 2^{E-127} \cdot (1{,}M)$
0	≠ 0	$-1^V \cdot 2^{-126} \cdot (0{,}M)$
0	0	$-1^V \cdot 0$

Minimierung - Allgemeine Vorgehensweise:

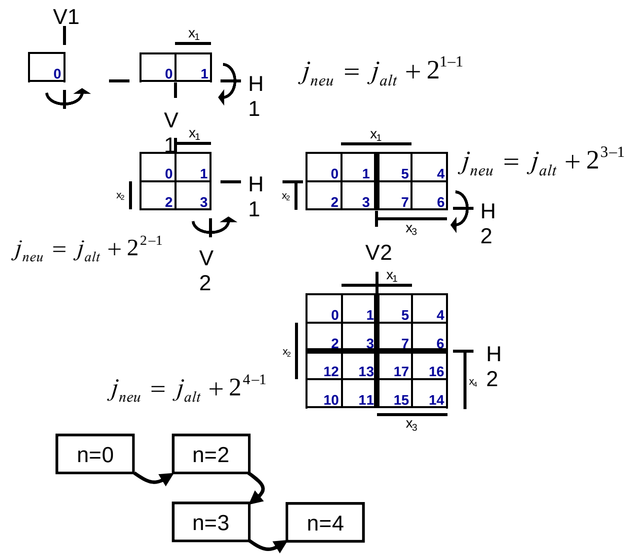
1) **Kerne** bestimmen und **Streichen aller überdeckten Spalten** (@ Einsstellen) („leergewordene" Zeilen können auch gestrichen werden)

2) **Spaltendominanzen** finden und **dominierende Spalten streichen**

3) **Zeilendominanzen** finden und **dominierte Zeilen streichen**, nach Möglichkeit (-> **Kosten ci beachten**!)

4) Schritte **1-3 wiederholen**, bis **Überdeckungstabelle nicht reduzierbar** -> keine Kerne und Dominanzen mehr (ggf. noch zyklische Resttabelle auflösen)

Entwicklung eines Symmetriediagramms



FlipFlops (charakteristische Gleichungen)

RS-FlipFlop:

$$q_k^{v+1} = S^v \vee (q_k^v \,\&\, \overline{R}^v)$$

q^v	q^{v+1}	R	S
0	0	-	0
0	1	0	1
1	0	1	0
1	1	0	-

D-FlipFlop:

$$q_k^{v+1} = D^v$$

q^v	q^{v+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

JK-FlipFlop:

$$q^{v+1} = (\overline{K} \,\&\, q^v) \vee (J \,\&\, \overline{q}^v)$$

q^v	q^{v+1}	K	J
0	0	-	0
0	1	-	1
1	0	1	-
1	1	0	-

T-FlipFlop:

$$q^{v+1} = (T \,\&\, \overline{q}^v) \vee (\overline{T} \,\&\, q^v)$$

q^v	q^{v+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Automaten

A_h^v Ausgangsvektor;

S_k^v Zustandsvektor;

E_g^v Eingangsvektor

Transitionsgleichungen

$$s_i^{v+1} = \delta_i\left(S_k^v, E_g^v\right), mit\ s_i^{v+1} \in S_k^{v+1}$$

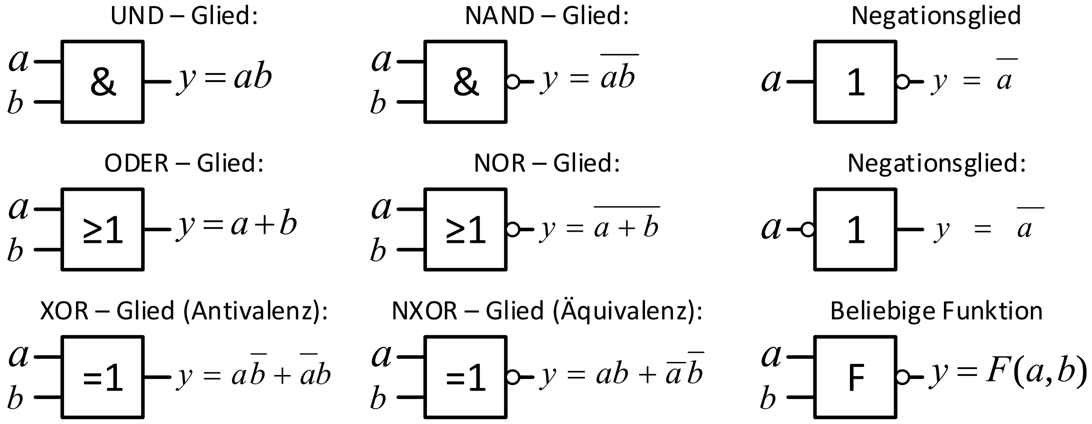
Ausgabefunktionen Medwedew $A_h^v = S_k^v$

Ausgabefunktionen Moore $A_h^v = \lambda_i\left(S_k^v\right)$

Ausgabefunktionen Mealy $A_h^v = \lambda_i\left(S_k^v, E_g^v\right)$

a, b, x, y, z : boolesche Variablen l : Literal f, g : boolesche Funktionen v_1 : p-Netz v_0 : n-Netz
s : Summe/Zustand c : Carry i : Eingang δ : Transitionsfunktion λ : Ausgabefunktion

Schaltsymbole

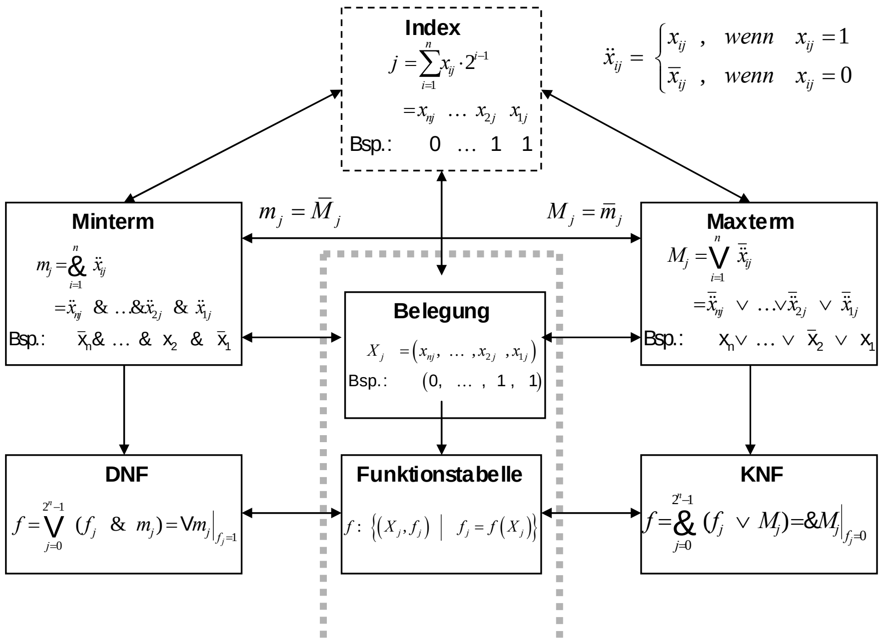


Zusammenstellung der wichtigsten Basissysteme

Zahl der Operatoren	Namen	Zeichen	Darstellung mit UND, ODER, NICHT	Darstellung von $\bar a$	$a\&b$	$a\vee b$
3	NICHT UND ODER	$y=\bar a$ $y=a\&b$ $y=a\vee b$	-	$\bar a$ - -	- $a\&b$ -	- - $a\vee b$
2	NICHT UND	$y=\bar a$ $y=a\&b$	-	$\bar a$ -	- $a\&b$	$a\vee b=\overline{\overline{a\vee b}}=\overline{\bar a\&\bar b}$
2	NICHT ODER	$y=\bar a$ $y=a\vee b$	-	$\bar a$ -	$a\&b=\overline{\overline{a\&b}}$ $=\overline{\bar a\vee\bar b}$	- $a\vee b$
2	UND ANTIVALENZ (Konstante 1)	$y=a\&b$ $y=a\oplus b$	$y=a\&b$ $y=\bar a\&b\vee a\&\bar b$	$\bar a=\bar a\&1\vee a\&\bar 1$ $=a\oplus 1$	$a\&b$ -	$a\vee b$ $=a\oplus b\oplus(a\&b)$
1	NAND	$y=a\overline{\&}b$	$y=\overline{a\&b}$ $=\bar a\vee\bar b$	$\bar a=\overline{a\&a}=a\overline{\&}a$ oder $\bar a=\overline{a\&1}=a\overline{\&}1$	$a\&b=\overline{\overline{a\&b}}$ $=\overline{a\overline{\&}b}$ $=(a\overline{\&}b)\overline{\&}(a\overline{\&}b)$	$a\vee b=\overline{\bar a\&\bar b}$ $=\bar a\overline{\&}\bar b$ $=(a\overline{\&}a)\overline{\&}(b\overline{\&}b)$
1	NOR	$y=a\overline{\vee}b$	$y=\overline{a\vee b}$ $=\bar a\&\bar b$	$\bar a=\overline{a\vee a}=a\overline{\vee}a$ oder $\bar a=\overline{a\vee 0}=a\overline{\vee}0$	$a\&b=\overline{\bar a\vee\bar b}$ $=\bar a\overline{\vee}\bar b$ $=(a\overline{\vee}a)\overline{\vee}(b\overline{\vee}b)$	$a\vee b=\overline{\overline{a\vee b}}$ $=\overline{a\overline{\vee}b}$ $=(a\overline{\vee}b)\overline{\vee}(a\overline{\vee}b)$

Hauptsatz der Schaltalgebra

Beziehungen zwischen den Begriffen



a,b,x,y,z : boolesche Variablen l : Literal f,g : boolesche Funktionen v_1 : p-Netz v_0 : n-Netz
s : Summe/Zustand c : Carry i : Eingang δ : Transitionsfunktion λ : Ausgabefunktion