

Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

Fragestunde Digitaltechnik

Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

Klausur Infos

■ Klausur am 15. März 2016 um 17 – 19 Uhr

- in insgesamt 8 Hörsälen (HS-Verteilung wird Ende nächster Woche auf ILIAS bekannt gegeben)
- Prüfungsdauer: 120 min
- Zugelassene Hilfsmittel:
 - Nichts außer zwei Seiten (**1 Blatt DIN-A4**, beidseitig beschriftet) **handgeschriebene** Formelsammlung zugelassen.
 - eine weitere Formelsammlung ist der Klausur angehängt. **Nur diese** ist zur Prüfung zugelassen. (Formelsammlung ist auf ILIAS einsehbar)
- **KEIN** Taschenrechner
- Prüfungsberechtigt sind nur Studierende, die sich vorher angemeldet haben (Bei Problemen Email an pistorius@kit.edu)
- Bei nicht Abmeldung **VOR** der Klausur (online oder direkt bei der Aufsicht) wird die Klausur mit 5,0 bewertet

■ Schon angemeldet?????

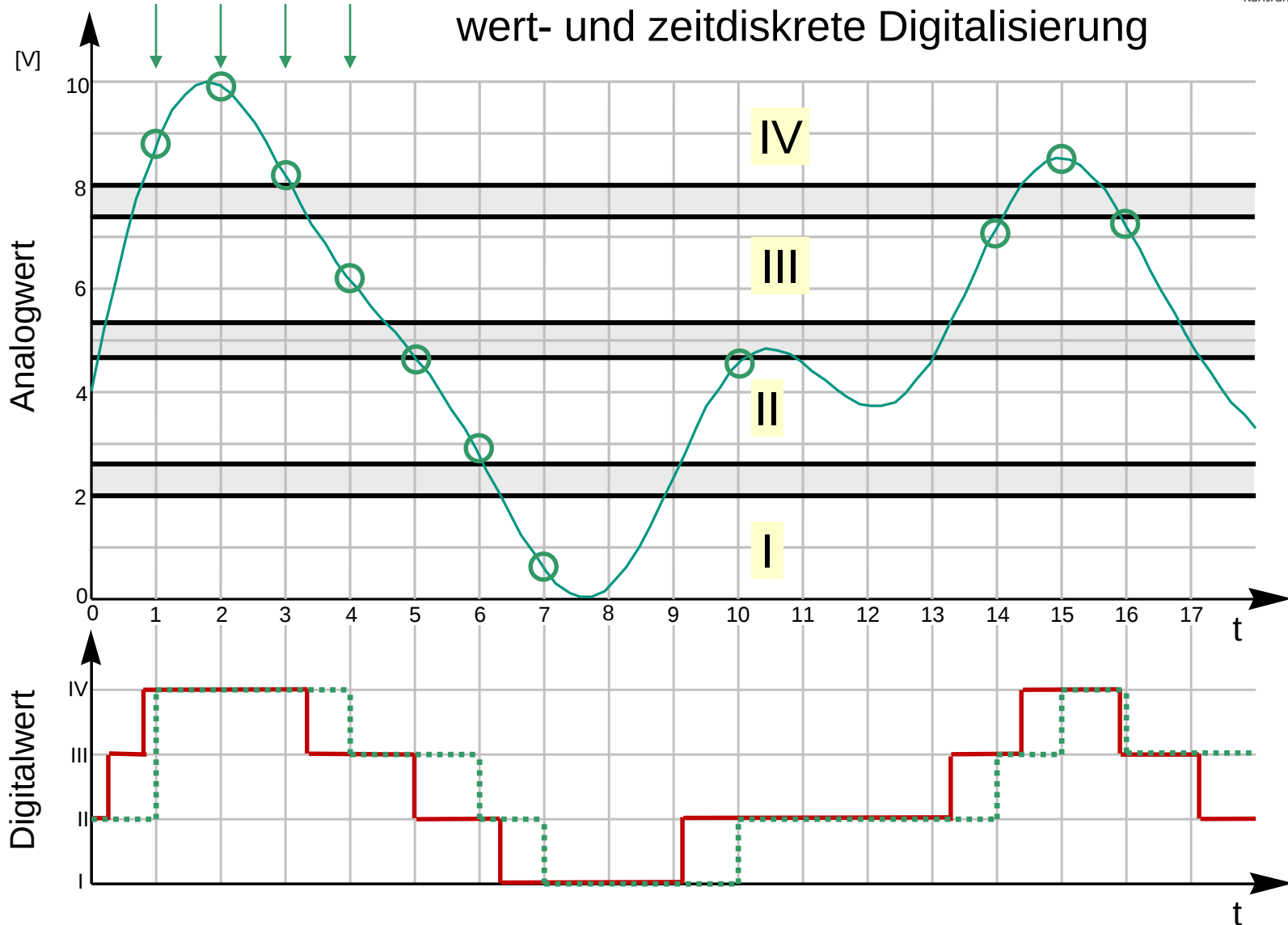
Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

Digitalisierung

Digitalisierung

wert- und zeitdiskrete Digitalisierung



Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

Informationsgehalt

Betrachtung nicht gleichwahrscheinlicher Zeichen:

- Informationsgehalt eines Zeichens e

$$H_e = \lg \frac{1}{p}$$

- In einer Folge mit L Zeichen ist die zu erwartende Auftrittshäufigkeit eines speziellen Zeichens i :

$$L \cdot p(i)$$

- **Alle** beobachteten **Zeichen i** liefern den gesamten **Informationsbeitrag**:

$$L \cdot p(i) \cdot H_{ei} = L \cdot p(i) \cdot \lg \frac{1}{p(i)}$$

- Alle N Zeichen zusammen betrachtet
→ **Ø Informationsgehalt** pro Zeichen:

$$H = \sum_{i=1}^N p(i) \cdot \lg \frac{1}{p(i)}$$

und nennt **H Entropie** der Quelle

- Optimale Codes versuchen die im Mittel auftretende durchschnittliche Codewortlänge $\bar{m}$ zu minimieren
- Der minimal erreichbare Idealwert ist dabei der durchschnittliche Informationsgehalt H des Zeichenvorrats $\rightarrow$ Entropie der Sendequelle
- Für einen Zeichenvorrat $\{x_1, x_2, x_3, \dots x_n\}$ gilt:

Ø Informationsgehalt H :
$$H = \sum_{i=1}^n p(x_i) \cdot H_e(x_i) = \sum_{i=1}^n p(x_i) \cdot \lg \frac{1}{p(x_i)}$$

Ø Codewortlänge m :
 $m(x_i)$: Länge CW für x_i

$$\bar{m} = \sum_{i=1}^n p(x_i) \cdot m(x_i)$$

Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

Fehlererkennung und -korrektur

■ Problem in der Praxis

Störeinflüsse können bei der Übertragung oder Speicherung von binär kodierten Informationen den Wert der zur Darstellung verwendeten physikalischen Größe verfälschen → Bits „kippen“

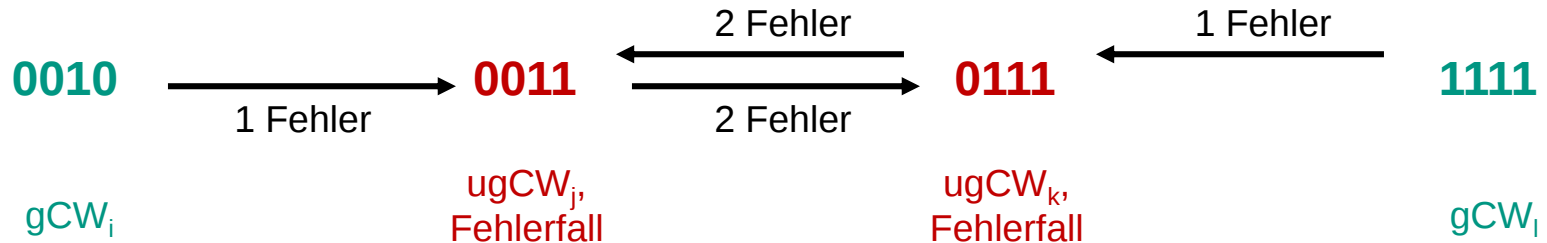
■ Lösung: unbenutzte Codewörter (CW) „geschickt“ nutzen



- einzelne Bitfehler (1 Bit “kippt”) sind erkennbar bei Codes mit minimaler Hammingdistanz $HD_{\min} = 2$
 - jeder denkbare Bitfehler führt zu ungültigem (d.h. unbenutztem) Codewort

Codes für Fehlererkennung / Fehlerkorrektur

■ 2-Fach Fehlererkennung:



■ 1-Fach Fehlerkorrektur:



■ Es gilt

Bei Codes mit $HD_{\min} = 3$:
→ Zweifachfehler erkennbar oder Einfachfehler korrigierbar

(minimale) Hamming-Distanz

- **Definition:** Hammingdistanz **HD** (auch Hammingabstand genannt)
 - HD_{ij} = Anzahl der Stellen, an denen sich CW_i und CW_j unterscheiden.
- **Definition:** Minimale Hammingdistanz **HD_{min}** eines Codes
 - $HD_{\min}(X) = \min\{HD_{ij} \mid CW_i, CW_j \in X \wedge CW_i \neq CW_j\}$
= minimale Hammingdistanz, die beim Vergleich aller Codewörter CW_i und CW_j (mit $CW_i \neq CW_j$) auftritt
- Die minimale Hammingdistanz ist eine entscheidende Eigenschaft eines Codes, um Übertragungsfehler erkennen und korrigieren zu können:
- Bei gegebenem $h = HD_{\min}(X)$ sind
 - $h - 1$ Bitfehler **erkennbar** und
 - $k < h / 2$ Bitfehler **korrigierbar**

Blocksicherung mittels Paritätsbits

- Zusammenfassung **mehrerer** Codeworte zu einem Block
- Hinzufügen eines **Prüfwortes** (Spaltenparitäten) an jeden Block

Ziffer	Codewörter mit gerader Parität				
5	0	1	0	1	0
4	0	1	0	0	1
1	0	0	1	1	1
3	0	0	1	1	0
9	1	0	0	1	0
8	1	0	0	0	1
Prüfwort	0	0	1	0	1

← Zeile mit Fehler (ungerade Parität)

↑ Spalte mit Fehler (ungerade Parität)

- Einfachfehlerkorrektur: Spalte und Zeile **eindeutig** ermittelbar
- Mehrfachfehler: ggf. nicht ermittelbar oder erkennbar

Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

Optimale Codes

- Konstruktionsvorschrift:
 - **Zeichenvorrat** nach **aufsteigender Wahrscheinlichkeit** linear sortieren → lineare Ordnung
 - **Zwei Teilmengen** so konstruieren, dass die **Summenwahrscheinlichkeiten** der beiden Teilmengen **möglichst** gleich groß sind, d.h. die addierten Auftrittswahrscheinlichkeiten der in beiden Teilmengen enthaltenen Zeichen ergeben möglichst die gleiche Summe
 - Die **erste Teilmenge** erhält das Codierungszeichen “0” und die **zweite Teilmenge** eine “1”
 - Mit den jeweils **resultierenden Teilmengen rekursiv** in gleicher Weise fortfahren
→ bis nur noch 1 Zeichen in resultierenden Teilmengen enthalten ist

- *Huffman's Kodierungs-Algorithmus (HKA)*
- HKA konstruiert einen Baum T , der die Kodierung der einzelnen Zeichen repräsentiert
- Eine Queue Q wird dafür benutzt, um die zwei Zeichen mit der niedrigsten Auftrittswahrscheinlichkeit zu finden
- Diese zwei Zeichen werden zu einem Objekt verschmolzen, dessen Auftrittswahrscheinlichkeit die Summe der Einzelwahrscheinlichkeiten ist

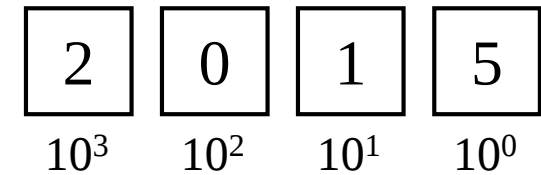
Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

Zahlensysteme

Wandlung der Dezimalzahl in andere Systemen

Dezimalzahl: $N_D = 2015$



Beispiel:
Wandlung der
Dezimalzahl 2015 in
eine Hexadezimalzahl.

$$2015 \bmod 16 = 15$$

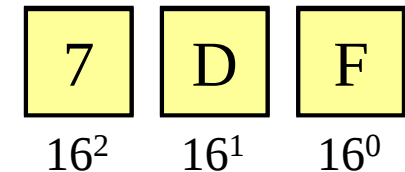
$$2015 \operatorname{div} 16 = 125$$

$$125 \bmod 16 = 13$$

$$125 \operatorname{div} 16 = 7$$

$$7 \bmod 16 = 7$$

Hexadezimalzahl: N_H



Binär → dezimal

■ Beispiel:

$$(11010110)_B$$

$$=(1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0)_D$$

$$=(128 + 64 + 0 + 16 + 0 + 4 + 2 + 0)_D$$

$$=214_D$$

oktal $\rightarrow$ binär $\rightarrow$ hex $\rightarrow$ dez

■ Beispiel:

$$1021_{\text{O}}$$

$$=(001\ 000\ 010\ 001)_{\text{B}}$$

$$=(\underbrace{0010}_{2}\ \underbrace{0001}_{1}\ \underbrace{0001}_{1})_{\text{B}}$$

$$=(2\ 1\ 1)_{\text{H}}$$

$$=(2*16^2 + 1*16^1 + 1*16^0)_{\text{D}}$$

$$=(512 + 16 + 1)_{\text{D}}$$

$$=529_{\text{D}}$$

Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

Zehnerkomplement

Bsp. Zehnerkomplement, positives Ergebnis

$$769_D - 341_D = 428_D$$

$$769 + (\underbrace{1000 - 341}_{\text{10er Komplement, K10(341)}}) - 1000 =$$

10er Komplement, K10(341)

$$769 + (\underbrace{999 - 341 + 1}_{\text{9er Komplement, K9(341)}}) - 1000 =$$

9er Komplement, K9(341)

$$\begin{aligned} 769 + (999 - 341) + 1 - 1000 &= \\ 769 + K9(341) + 1 - 1000 &= \\ 769 + 658 + 1 - 1000 &= \\ 1427 + 1 - 1000 &= \\ \mathbf{1428} - \mathbf{1000} &= \\ &= 428 \end{aligned}$$

Eigentliche Rechnung - mit positivem Ergebnis

Einfacher mathem. Trick: Ergänzung um 1000

→ (1000 - 341) ist das 10er Komplement der Zahl 341; leider ist dieses jedoch immer noch nicht leichter zu berechnen

weiterer Trick: Ausklammern einer 1

→ (999 - 341) ist das 9er Komplement der Zahl 341; Dieses hat den Vorteil, dass man es relativ leicht, nämlich Stellenweise berechnen kann (also keine Überträge bei der Berechnung auftreten:

$$K9(341) = 999 - 341 = 658$$

$$\{9 - 3 = 6; 9 - 4 = 5; 9 - 1 = 8\}$$

Das K10(341) ergibt sich also aus dem K9(341) + 1

Einziges Problem bleibt nun die abschließende Subtraktion von 1000; Wird aber (wie in diesem Fall) von einer größeren Zahl eine kleinere Abgezogen so entsteht IMMER ein **Übertrag von genau 1 in die nächst höhere Stelle**, und genau diese 1 muss abschließend abgezogen werden.

Man kann diesen Übertrag also einfach weglassen.

Bsp. Zehnerkomplement, negatives Ergebnis

$$341_D - 769_D = -428_D$$

$$341 + (\underbrace{1000 - 769}_{10\text{er Komplement; } K10(769)}) - 1000 =$$

$$341 + (\underbrace{999 - 769 + 1}_{9\text{er Komplement (} K9(769) = 230)}) - 1000 =$$

$$\begin{aligned} 341 + K9(769) + 1 - 1000 &= \\ 341 + 230 + 1 - 1000 &= \\ 571 + 1 - 1000 &= \\ \underline{572} - 1000 &= \end{aligned}$$

$$- (\underbrace{1000 - 572}_{10\text{er Komplement; } K10(572)}) =$$

10er Komplement; $K10(572)$

$$- (\underbrace{999 - 572 + 1}_{9\text{er Komplement (} K9(572) = 427)}) =$$

9er Komplement ($K9(572) = 427$)

$$- (427 + 1) = -428$$

Eigentliche Rechnung - mit negativem Ergebnis

Bilden des K10

Wieder wird das K9 gebildet, um das K10 leichter berechnen zu können ($K10(x) = K9(x) + 1$)

Nun tritt aber folgendes Problem auf:

→ Da hier eine größere Zahl von einer kleineren abgezogen wird, ergibt sich **kein Übertrag** auf die nächste Stelle womit die einfache Subtraktion der 1000 nicht mehr möglich ist.

Weiterer Trick: Ausklammern des Faktors -1

→ Das so entstandene Zwischenergebnis ist aber gerade das schon bekannte $K10(572)$, welches wiederum aus dem $K9(572) + 1$ leicht berechnet werden kann

Nicht zu vergessen ist natürlich die -1 die wir zwischendurch ausgeklammert haben, wodurch sich das korrekte negative Ergebnis ergibt

Tritt also bei der Berechnung kein Übertrag auf, so muss vom Zwischenergebnis der Subtraktion nochmals das K10 gebildet werden!

Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

Zweierkomplement

Bsp. Zweierkomplement, positives Ergebnis

$$1101_B - 1010_B = 11_B$$

$$1101 + (\underbrace{10000 - 1010}_{\text{2er Komplement, } K_2(1010)}) - 10000 =$$

$$1101 + (\underbrace{1111 - 1010 + 1}_{\text{1er Komplement, } K_1(1010)}) - 10000 =$$

$$\begin{aligned} & 1101 + (1111 - 1010) + 1 - 10000 \\ = & 1101 + K_1(1010) + 1 - 10000 \\ = & 1101 + 0101 + 1 - 10000 \\ = & 10010 + 1 - 10000 \\ = & 10011 - 10000 \\ = & 11 \end{aligned}$$

Eigentliche Rechnung – mit positivem Ergebnis

Trick: Ergänzung auf die nächst größere Stelle 10000_B

→ $(10000_B - 1010_B)$ ist das 2er Komplement der Zahl 1010_B ; leider ist dieses jedoch immer noch nicht leichter zu berechnen

weiterer Trick: Ausklammern einer 1

→ $(1111_B - 1010_B)$ ist das 1er Komplement der Zahl 1010_B ; Im Dualsystem ist das 1er Komplement einfach durch stellenweises Invertieren berechenbar:

$$K_1(1010_B) = 1111_B - 1010_B = 0101_B$$

$$\{1 - 1 = 0; 1 - 0 = 1; 1 - 1 = 0; 1 - 0 = 1\}$$

Das $K_2(1010)$ ergibt sich also aus dem $K_1(1010) + 1$

Einziges Problem bleibt nun die abschließende Subtraktion von 10000_B

Wird aber (wie in diesem Fall) von einer größeren Zahl eine kleinere Abgezogen so entsteht IMMER ein Übertrag von genau 1 in die nächst höhere Stelle, und genau diese 1 muss abschließend abgezogen werden. Man kann diesen Übertrag also einfach streichen.

Bsp. Zweierkomplement, negatives Ergebnis

$$1010_B - 1101_B = -11_B$$

$$1010 + (10000 - 1101) - 10000 =$$

2er Komplement; K2(1101)

$$1010 + (1111 - 1101 + 1) - 10000 =$$

1er Komplement (K1(1101) = 0010)

$$1010 + K1(1101) + 1 - 10000 =$$

$$1010 + 0010 + 1 - 10000 =$$

$$1100 + 1 - 10000 =$$

$$1101 - 10000 =$$

$$- (10000 - 1101) =$$

2er Komplement; K2(1101)

$$- (1111 - 1101 + 1) =$$

1er Komplement (K1(1101) = 0010)

$$- (10 + 1) = -11$$

Eigentliche Rechnung – mit negativem Ergebnis

Bilden des K2

Wieder wird das K1 gebildet um das K2 leichter berechnen zu können (K2(x) = K1(x) + 1)

Problem:

→ Da hier eine größere Zahl von einer kleineren abgezogen wird, ergibt sich kein Übertrag auf die nächste Stelle womit die einfache Subtraktion der 10000B nicht mehr möglich ist.

Trick: Ausklammern des Faktors -1

→ Das so entstandene Zwischenergebnis ist wieder gerade das schon bekannte K2(1101B), welches wiederum aus dem K1(1101B) + 1 leicht berechnet werden kann

Nicht zu vergessen ist natürlich wieder der Faktor -1 den wir zwischendurch ausgeklammert haben, wodurch sich das korrekte negative Ergebnis wieder ergibt

Tritt also bei der Berechnung kein Übertrag auf, so muss vom Zwischenergebnis der Subtraktion nochmals das K2 gebildet werden!

Addition im BCD - Code

- Bei der Addition zweier **BCD-Dekaden** sind drei Fälle zu beachten:
 - Tritt kein Übertrag und keine Pseudotetrade auf
 - Einfaches addieren ohne Korrektur
 - Tritt ein Übertrag auf
 - so muss „6_D“ zur Ergebnisdekade addiert werden.
 - ergibt sich als Resultat eine Pseudotetrade
 - so muss „6_D“ zur Ergebnisdekade addiert werden und eventuell eine „1_D“ zur Übertragungsdekade

Dezimal	BCD - Code
8	1000
+ 9	1001
	1
	0001 0001
+ Übertrag	0110
= 17	0001 0111

Dezimal	BCD - Code
7	0111
+ 6	0110
	1101
+ PSD	1 0110
= 13	0001 0011

Addition im Stibitz - Code

- Bei der Addition zweier **Stibitz-Dekaden** sind zwei Fälle zu beachten:
 - Tritt **kein Übertrag** auf
 - so muss „3_D“ abgezogen werden
 - dies entspricht einer Addition von „1101_B“ ohne Übertrag
 - Tritt **ein Übertrag** auf
 - so muss „3_D“ addiert werden
 - dies entspricht einer Addition von „0011_B“ ohne Übertrag

Dezimal	Stibitz - Code
4	0011 0111
+ 9	0011 1100
	1
	0111 0011
+	1 1101 0011
= 13	0100 0110

Dezimal	Stibitz - Code
3	0110
+ 4	0111
	1101
+ Ohne Ü.	1 1101
= 7	1010

Multiplikation im Dualsystem

- Führen Sie die folgenden Berechnungen im angegebenen Zahlensystem aus, ohne die Zahlen ins Dezimalsystem umzuwandeln:

					1	1	1	0	1	0	1	0	
					1	1	1	0	1	0	1	0	1
+				1	1	1	0	1	0	1	0		1
+			0	0	0	0	0	0	0	0			0
+		1	1	1	0	1	0	1	0				1
	1	1	1	1	1	1	1						
			1	1	1								
=	1	0	1	0	0	0	0	0	1	1	1	0	

Verknüpfung entspricht:

$\cdot 2^0$ UND - Operation

$\cdot 2^1$ UND + LShift um eine Stelle

$\cdot 2^2$ UND + LShift um zwei Stellen

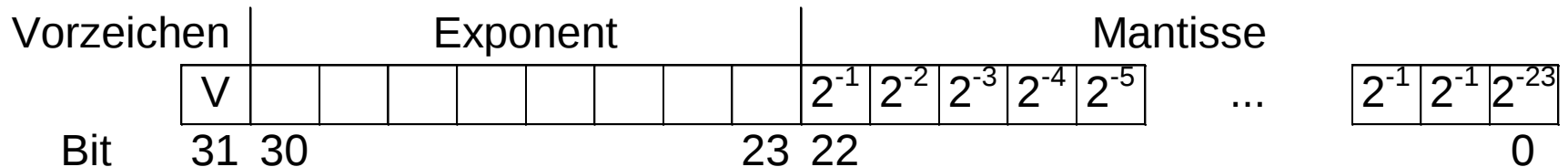
$\cdot 2^3$ UND + LShift um drei Stellen

Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

Fließkommazahlen

- Da **Gleitkommazahlen** auf sehr unterschiedliche Weise kodiert werden können, hat der **IEEE Richtlinien** festgelegt, wie Gleitkommazahlen dargestellt werden sollen
- **Gleitkomma-Zahlendarstellung gemäß IEEE-Standard:**



Exponent E	Mantisse M	Wert
255	$\neq 0$	ungültig (NaN)
255	0	$- 1^V \cdot \infty (\pm \text{unendlich})$
$0 < E < 255$	M	$- 1^V \cdot 2^{E-127} \cdot (1, M)$
0	$\neq 0$	$- 1^V \cdot 2^{-126} \cdot (0, M)$
0	0	$- 1^V \cdot 0$

Digitaltechnik: Zahlendarstellung

Gleitkomma-Zahlendarstellung gemäß IEEE-Standard:

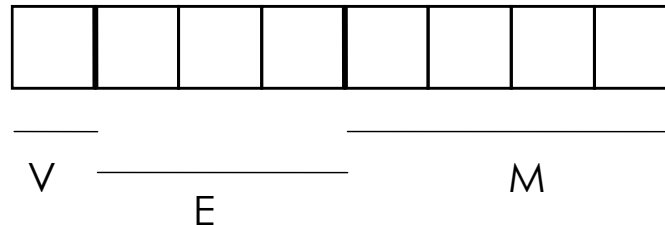
Die Zahl π mit 7 Dezimalstellen hinter dem Komma im **IEEE-Format** dargestellt:

$$\begin{aligned}
 \pi &\approx 3,1415926_D \approx 11,0010010000111110110\ 10_B \\
 &= (-1^0 \times 2^{128-127})_D \times 1,1001001000011111011\ 010_B \\
 &\quad \quad \quad / \quad \quad \quad | \quad \quad \quad /
 \end{aligned}$$

Ergebnis: $\pi \approx 0\ 10000000\ 1001001000011111011010$

Klausuraufgabe SS2009

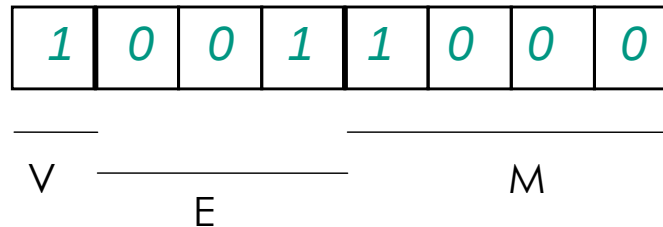
Zur Verwendung in einem Microcontroller wurde eine platzsparende Darstellung von Fließkommazahlen in einem einzigen Byte entwickelt. Das höchstwertige Bit stellt das Vorzeichen V dar, die vier niederwertigsten Bits die Mantisse M und die drei Bits in der Mitte den Exponenten E (siehe Abbildung).



Für alle möglichen binären Belegungen ergibt sich der Dezimalwert Z aus nachstehender Formel (vgl. IEEE-Fließkommazahl):

$$Z_D = (-1)^V \cdot 2^{E-3} \cdot (1, M)$$

A) Berechnen Sie den Dezimalwert der Belegung 10011000.



$$Z_D = (-1)^V \cdot 2^{E-3} \cdot (1, M)$$

$$\begin{aligned} Z &= (-1) \cdot 2^{(1-3)} \cdot 1,5 = (-1) \cdot 2^{(-2)} \cdot 1,5 \\ &= (-1) \cdot 0,25 \cdot 1,5 \\ &= -0,375 \end{aligned}$$

- B) Konvertieren Sie die Zahl $3,625_D$ aus dem Dezimalsystem in Fließkomma. (keine Klausuraufgabe)

V		E			M		

$$Z_D = (-1)^V \cdot 2^{E-3} \cdot (1, M)$$

Umwandeln in Dualzahl: $+3,625_D = +11,101_B$

Normalisieren: $11,101_B = 1,1101_B \cdot 2^1$

Exponent berechnen: $2^{E-3} = 2^1 \Rightarrow E = 3+1 = 4$

Fließkommazahl angeben:

0	1	0	0	1	1	0	1
V		E			M		

Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

Mengen, Relationen und Graphen

Bemerkung

- **Potenzmenge**: Menge aller Untermengen inkl. der Menge $\emptyset$
- **Kartesisches Produkt**: $S \times T$, Menge aller **geordneten** Paare (s,t) mit $s \in S$, $t \in T$, nicht kommutativ $\Rightarrow S \times T \neq T \times S$
- **Leere Menge** ist Untermenge jeder beliebigen Menge

- Auf der Grundmenge $G = \{ a, b, c, d, e, 1, 2, 3 \}$ seien zwei Mengen gegeben:
- $A = \{ x \mid x \in G, x = a \text{ oder } (x-3)^2 = 1 \text{ oder } x^3 = 27 \}$ und
- $B = \{ a, b, c, 2 \}$

3. Übung Aufgabe 4.2 Bilden Sie die Potenzmengen $P(A)$ und $P(B)$.

$$A = \{ a, 2, 3 \}$$

$$P(A) = \{ \{ \}, \{ a \}, \{ 2 \}, \{ 3 \}, \{ a, 2 \}, \{ a, 3 \}, \{ 2, 3 \}, \{ a, 2, 3 \} \}$$

$$B = \{ a, b, c, 2 \}$$

$$P(B) = \{ \{ \}, \{ a \}, \{ b \}, \{ c \}, \{ 2 \}, \{ a, b \}, \{ a, c \}, \{ a, 2 \}, \{ b, c \}, \{ b, 2 \}, \{ c, 2 \}, \{ a, b, c \}, \{ a, b, 2 \}, \{ a, c, 2 \}, \{ b, c, 2 \}, \{ a, b, c, 2 \} \}$$

- Auf der Grundmenge $G = \{ a, b, c, d, e, 1, 2, 3 \}$ seien zwei Mengen gegeben:
- $A = \{ x \mid x \in G, x = a \text{ oder } (x-3)^2 = 1 \text{ oder } x^3 = 27 \}$ und
- $B = \{ a, b, c, 2 \}$

3. Übung Aufgabe 4.3 Bilden Sie die **kartesischen Produkte** $A \times B$ und $B \times A$.

$$A = \{ a, 2, 3 \} \quad B = \{ a, b, c, 2 \}$$

$$A \times B = \{ (a, a), (a, b), (a, c), (a, 2), (2, a), (2, b), (2, c), (2, 2), (3, a), (3, b), (3, c), (3, 2) \}$$

$$B \times A = \{ (a, a), (a, 2), (a, 3), (b, a), (b, 2), (b, 3), (c, a), (c, 2), (c, 3), (2, a), (2, 2), (2, 3) \}$$

Zweistellige Relationen α auf einer Menge M

■ Eigenschaften:

- reflexiv: $x \alpha x, \forall x \in M$
- symmetrisch: aus $x \alpha y$ folgt $y \alpha x, \forall x, y \in M$
- antisymmetrisch: aus $x \alpha y$ und $y \alpha x$ folgt $x = y, \forall x, y \in M$
- transitiv: aus $x \alpha y$ und $y \alpha z$ folgt $x \alpha z, \forall x, y, z \in M$

■ Ordnungsrelation (Zeichen: $\leq$)

- reflexiv
- antisymmetrisch
- Transitiv

■ Äquivalenzrelation (Zeichen: $\equiv$)

- reflexiv
- symmetrisch
- transitiv

■ Verträglichkeitsrelation (Zeichen: $\sim$)

- reflexiv
- symmetrisch
- nicht transitiv

Eigenschaften von Graphen

Endlicher Graph

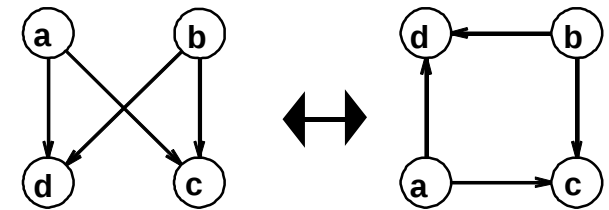
- Kanten- und Knotenmenge sind endlich.

Entarteter Graph

- Kantenmenge ist leer.

Isomorpher Graph

- “Graphen lassen sich ineinander überführen”



Vollständiger Graph

- Jeder Knoten ist über eine Kante mit jedem anderen Knoten verbunden.

Einfacher Graph

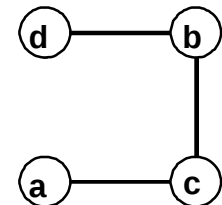
- Zwischen zwei Knoten gibt es höchstens eine Kante.

Gerichteter Graph

- Enthält mindestens eine gerichtete Kante.

Zusammenhängender Graph

- Es existiert vom jedem Knoten eine Kantenprogression (Kantenfolge) zu jedem anderen Knoten.



Eigenschaften von Graphen

streng zusammenhängender Graph

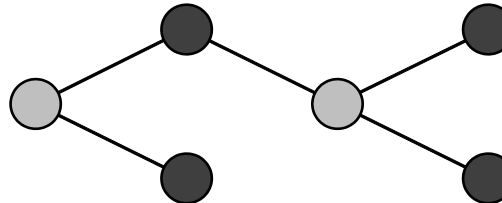
- Bei gerichteten Graphen: Jeder Knoten ist von jedem Knoten aus erreichbar (Kantenrichtung beachten).

zyklischer Graph

- gerichteter Graph mit mindestens einem Zyklus (auch Schleife).

bipartiter Graph

- zwei Knotenmengen, wobei keine Kante Knoten aus derselben Menge verbindet .



planarer Graph

- es existiert ein zugehöriger isomorpher Graph, der in der Ebene überschneidungsfrei ist.

gewichteter Graph

- enthält gewichtete Knoten oder Kanten (z.B.: Entfernungen, Kosten, ...)

■ Kanteneigenschaften

■ Schleife

- verbindet einen Knoten mit sich selbst.

■ Knoteneigenschaften

■ Grad $d(g)$

- Anzahl der inzidenten Kanten eines Knotens.

■ Ausgangsgrad $d^+(g)$

- Anzahl der abgehenden Kanten.

■ Eingangsgrad $d^-(g)$

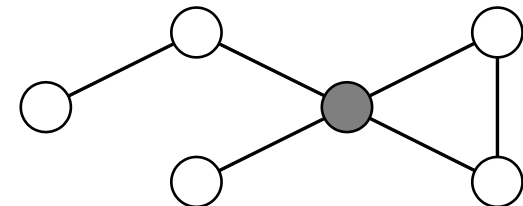
- Anzahl der ankommenden Kanten.

■ benachbarte Knoten $V'(g)$

- sind über eine Kante miteinander verbunden.
- bei gerichteten Graphen Unterscheidung in: Vorgänger: $V'_1(g)$; Nachfolger: $V'_2(g)$

■ Artikulation

- Knoten, nach dessen Entnahme der Graph in isolierte Teilgraphen zerfällt:



Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

Boolsche Algebra

Beziehungen der Schaltalgebra

Regeln für ein Element

$$\text{R5a} \quad a \vee 0 = a$$

$$\text{R6a} \quad a \vee 1 = 1$$

$$\text{R7a} \quad a \vee a = a$$

$$\text{R8a} \quad a \vee \bar{a} = 1$$

$$\text{R5b} \quad a \& 1 = a$$

$$\text{R6b} \quad a \& 0 = 0$$

$$\text{R7b} \quad a \& a = a$$

$$\text{R8b} \quad a \& \bar{a} = 0$$

$$\text{R9} \quad \overline{(\bar{a})} = \bar{\bar{a}} = a$$

Regeln für zwei oder mehr Elemente

$$\text{R10a} \quad a \vee (b \vee c) = (a \vee b) \vee c = a \vee b \vee c \quad \text{R10b} \quad a \& (b \& c) = (a \& b) \& c = a \& b \& c$$

(assoziative Gesetze)

$$\text{R11a} \quad a \vee (a \& b) = a \quad \text{R11b} \quad a \& (a \vee b) = a$$

(Absorptionsgesetze)

$$\text{R12a} \quad \overline{(a \vee b)} = \bar{a} \& \bar{b} \quad \text{R12b} \quad \overline{(a \& b)} = \bar{a} \vee \bar{b}$$

(De Morgansche Regeln)

Weitere Funktionen

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

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

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

Transitivität der Inklusion

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

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

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

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

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

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

$$\bar{x} = x \oplus 1, \quad x \oplus x = 0, \quad x \oplus 0 = x$$

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

- Zur Implementierung der Funktion stehen nur NAND-Gatter (mit mehreren Eingängen) zur Verfügung. Formen Sie die disjunktive Form mit Hilfe der DeMorganschen Regeln entsprechend um.

$$y = (x_1 \& \overline{x_3}) \vee (\overline{x_1} \& \overline{x_2})$$

$$\overline{\overline{(x_1 \& \overline{x_3}) \vee (\overline{x_1} \& \overline{x_2})}}$$

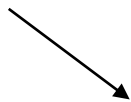
$$\overline{(x_1 \& \overline{x_3}) \vee (\overline{x_1} \& \overline{x_2})}$$

$$\overline{(x_1 \& \overline{x_3})} \& \overline{(\overline{x_1} \& \overline{x_2})}$$

$$= (x_1 \& (x_3 \& \overline{x_3})) \& ((\overline{x_1} \& \overline{x_1}) \& (\overline{x_2} \& \overline{x_2}))$$

mit $\overline{a} = \overline{a \& a}$

$$= a \& \overline{a}$$



Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

Symmetriediagramm

- **Konjunktion:** UND-Verknüpfung, &
- **Disjunktion:** ODER-Verknüpfung, ∨

- **KNF:** $y = \bigwedge_{j=0}^{2^n-1} (f_j \vee M_j)$ $\Rightarrow$ UND-Verknüpfung aller Maxterme (Nullblöcke)

Bsp.: $y = (\bar{a} \vee b \vee \bar{c}) \& (a \vee \bar{b} \vee c)$

a

1	1	0	1
0	1	1	1

b

c

- **DNF:** $y = \bigvee_{j=0}^{2^n-1} (f_j \& m_j)$ $\Rightarrow$ ODER-Verknüpfung aller Minterme (Einsblöcke)

Bsp.: $y = (\bar{a} \& b \& \bar{c}) \vee (a \& \bar{b} \& c)$

a

0	0	1	0
1	0	0	0

b

c

Hauptsatz der Schaltalgebra:

Satz: Jede beliebige Schaltfunktion $y = f(x_n, \dots, x_1)$ lässt sich als **Disjunktion** von **Mintermen** <Konjunktion von **Maxtermen**> **eindeutig** darstellen. In der **Disjunktion** <Konjunktion> treten genau diejenigen **Minterme** <Maxterme> auf, die zu den Einsstellen <Nullstellen> der Schaltfunktion gehören.

Beispiel: $y = f(x_4, x_3, x_2, x_1) = 1$, wenn die Oktalzahl durch 3 dividierbar ist

		x_1			
		0	1	0	0
x_2	0	0	0	0	0
	1	0	1	0	1
	2	0	0	1	0
	3	0	1	0	1
		x_3			
		0	1	0	1
		2	0	1	0
		3	1	0	1
		x_4			
		0	1	0	1
		2	0	1	0
		3	1	0	1

DNF: $y = (\bar{x}_4 \& \bar{x}_3 \& x_2 \& x_1) \vee (\bar{x}_4 \& x_3 \& x_2 \& \bar{x}_1) \vee (x_4 \& \bar{x}_3 \& \bar{x}_2 \& x_1) \vee (x_4 \& x_3 \& \bar{x}_2 \& \bar{x}_1) \vee (x_4 \& x_3 \& x_2 \& x_1)$

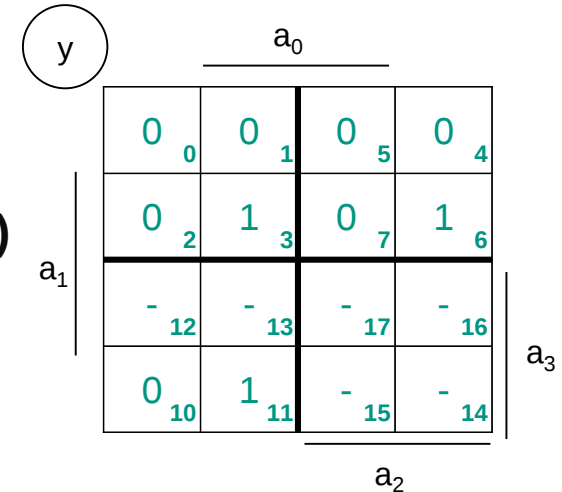
KNF: $y = (x_4 \vee x_3 \vee x_2 \vee x_1) \& (x_4 \vee x_3 \vee x_2 \vee \bar{x}_1) \& (x_4 \vee x_3 \vee \bar{x}_2 \vee x_1) \& (x_4 \vee \bar{x}_3 \vee x_2 \vee x_1) \& (x_4 \vee \bar{x}_3 \vee x_2 \vee \bar{x}_1) \& (x_4 \vee \bar{x}_3 \vee \bar{x}_2 \vee x_1) \& (\bar{x}_4 \vee x_3 \vee x_2 \vee x_1) \& (\bar{x}_4 \vee x_3 \vee x_2 \vee \bar{x}_1) \& (\bar{x}_4 \vee x_3 \vee \bar{x}_2 \vee x_1) \& (\bar{x}_4 \vee \bar{x}_3 \vee x_2 \vee x_1) \& (\bar{x}_4 \vee \bar{x}_3 \vee x_2 \vee \bar{x}_1) \& (\bar{x}_4 \vee \bar{x}_3 \vee \bar{x}_2 \vee x_1) \& (\bar{x}_4 \vee \bar{x}_3 \vee \bar{x}_2 \vee \bar{x}_1)$

→ nur mit den 3 Grundverknüpfungen (Operatoren) **Konjunktion**, **Disjunktion** und **Negation** ist es möglich **jede beliebige Schaltfunktion** darzustellen

→ $[\&, \vee, \neg]$ ist ein **Basissystem** der Schaltalgebra

Freistellen („don't cares“)

- „**unvollständig definierte Schaltfunktion**“: es ist mindestens eine **Don't-Care-Belegung (Freistelle)** vorhanden (*siehe Beispiel rechts*)
- Ergibt sich bei Eingangsbelegungen, die schaltungsbedingt **nie auftreten können**
- Daher: Zuordnung eines beliebigen Werts aus $f_j \in \{0, 1\}$ möglich
- Bei geschicktem Eins- oder Nullsetzen dieser Freistellen: mitunter erhebliche Vereinfachung des Ausdrucks (einfache Realisierung mit wenigen Logikelementen wird angestrebt)
- Die geschickte Wahl von Eins- und Nullstellen ist durch bloßes „Hinsehen“ nur für sehr kleine Ordnungen n möglich
- Beim **Nelson-Verfahren** werden Freistellen dazu verwendet, die Blockgröße zu erhöhen



y	a_0			
0	0	0	0	0
1	0	1	0	1
-	-	-	-	-
0	0	1	-	-

Minimierungsmethode

Minimierung am Symmetriediagramm

Beispiel (1):

- Gegeben: $y = f(x_4, x_3, x_2, x_1)$ durch die Angabe der **Eins-** und **Nullstellen** im **Symmetriediagramm**
- Gesucht: Kürzester algebraischer Ausdruck
→ **Disjunktive Minimalform (DMF)**
- **Primterme** zur Bildung einer **DMF** spezifizieren **Primeinsblöcke**
→ werden **Primimplikanten** genannt
(*Primimplikate* für Konjunktive Minimalform KMF)
- Die farbig dargestellten **Primimplikanten** überdecken **Eins-** und **Freistellen**

		x_1			
		0	1	5	4
x_2	0	0	1	7	6
	2	0	1	17	16
	12	0	1	15	14
	10	-	1	11	13
		x_3		x_4	

ITIV

Beispiel DMF

- Geben Sie die vollständige Blocküberdeckung in **disjunktiver** Form an.

		x_1			
x_2		1 ₀	1 ₁	0 ₅	1 ₄
		0 ₂	1 ₃	0 ₇	0 ₆
		x_3			

$$y = \overline{x_2} \& \overline{x_3} \vee \overline{x_1} \& \overline{x_3} \vee \overline{x_1} \& \overline{x_2}$$

$$y = \overline{x_1} \& \overline{x_3} \vee \overline{x_1} \& \overline{x_2}$$

Beispiel KMF

- Geben Sie die vollständige Blocküberdeckung in **konjunktiver** Form an.

	x_1			
	1 0	1 1	0 5	1 4
x_2	0 2	1 3	0 7	0 6
	x_3			

$$y = (\overline{x_1} \vee \overline{x_3}) \& (\overline{x_2} \vee \overline{x_3}) \& (\overline{x_1} \vee \overline{x_2})$$

$$y = (\overline{x_1} \vee \overline{x_3}) \& (\overline{x_1} \vee \overline{x_2})$$

Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

Nelson-Patrick-Verfahren

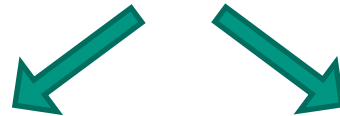
Formale algebraische Minimierung

→ Das Nelson/Petrick-Verfahren

- sehr leistungsfähiges algebraisches Verfahren
- Nelson-Verfahren: Bestimmung der Menge aller Primimplikanten (bzw. Primimplikaten)
- Petrick-Verfahren: Bestimmung der kostenminimalen Auswahl von Primimplikanten zur Einstellenüberdeckung (bzw. Primimplikaten zur Nullstellenüberdeckung)
Alternativ: graphisch mit Überdeckungstabelle
+ Regeln zur Abarbeitung/Vereinfachung

Ergebnis: kostenminimaler algebraischer Ausdruck der zu realisierenden digitalen Schaltfunktion

- Ziel: liefert zu einer (ggf. unvollständig definierten) Schaltfunktion f alle...



...Primimplikanten:

- Bildung der **Einstellenergänzung** f^E : Alle Freistellen werden zu „1“ deklariert
- Mittels bekannter algebraischer Verfahren wird die **KNF** [!] von f^E gebildet

...Primimplikate:

- Bildung der **Nullstellenergänzung** f^N : Alle Freistellen werden zu „0“ deklariert
- Mittels bekannter algebraischer Verfahren wird die **DNF** [!] von f^N gebildet



- Schrittweise ausdistribuierten, Termanteile, die **ausschließlich Freistellen** von f überdecken, direkt streichen
- Die Anzahl der Primblöcke kann anschließend mit dem **Petrick-Verfahren** minimiert werden

Minimierungsmethoden

Nelson-Verfahren:

Beispiel:

- Man nehme eine **Nullblocküberdeckung**:

$$f^E = (\overline{x_1} \vee x_2) \& (\overline{x_1} \vee x_3)$$

	x_1			
	1	0	0	1
x_2	-	0	1	-

- Durch **Ausdistribuierten** + **Umformung** erhält man:

$$\begin{aligned}
 f^E &= (\overline{x_1} \vee x_2) \& (\overline{x_1} \vee x_3) \\
 &= \overline{\overline{x_1} x_1} \vee \overline{\overline{x_1} x_3} \vee \overline{\overline{x_1} x_2} \vee \overline{x_2 x_3} \\
 &= \overline{\overline{x_1} x_1} \vee \overline{\overline{x_1} x_3} \vee \overline{\overline{x_1} x_2} \vee \overline{x_2 x_3} \\
 &= \overline{\overline{x_1} x_1} \vee \overline{\overline{x_1} x_2} \vee \overline{x_2 x_3} \\
 &= \overline{\overline{x_1} x_1} \vee \overline{x_2 x_3} \\
 &= \overline{\overline{x_1} x_1} \vee (\overline{x_2} \& \overline{x_3})
 \end{aligned}$$

Distributivgesetz

Regel R7

Absorption (R11)

Absorption (R11)

	x_1			
	1	0	0	1
x_2	-	0	1	-

Überdeckungsproblem (oder Auswahlproblem)

- Bei dem **Nelson-Verfahren** erfolgte nur die Ermittlung **aller Primterme**
- Weiterhin gesucht: **optimale Auswahl** der **Primterme**
 - eine **Auswahl** von **Primtermen**, die **alle Einsstellen** (Nullstellen) der gewünschten Funktion berücksichtigt wird **Überdeckung** genannt
- Lösung: graphische **Auswahl** der **Primterme** mit **Überdeckungstabelle**
 - für **jeden Primterm** wird angegeben, **welche Einsstellen** (bzw. Nullstellen) er **überdeckt**
 - zur **Bewertung** wird eine Spalte mit **Kosten c_k** angefügt
 - **Überdeckungstabelle** wird mit Hilfe bestimmter Regeln abgearbeitet (vereinfacht) -> **Kernermittlung** + **Dominanzregeln**
 - **algebraische Behandlung** der **optimalen Auswahl** von **Primtermen** auch möglich:
 - Bildung und Auswertung des **Petrick-Ausdrucks!**
- Ergebnis: **kostenminimale Überdeckung** der **Schaltfunktion**

Überdeckungsproblem:

Kernermittlung:

- Wenn eine **Einstelle** (Nullstelle) nur durch einen **einzigsten Primterm abgedeckt** wird, nennt man den **Primimplikanten** (Primimplikaten) **Kernimplikant** (*Kernimplikat*)
- **Kernimplikanten** (Kernimplikate) **müssen** auf jeden Fall in die **Überdeckungslösung** aufgenommen werden
- **Spalten von Einstellen** (Nullstellen) in der **Überdeckungstabelle**, die von **Kernimplikanten** (Kernimplikaten) **abgedeckt** werden, können **gestrichen** werden, und müssen in der weiteren Abarbeitung nicht mehr berücksichtigt werden

Minimierungsmethoden

Kernermittlung und Vereinfachung der Überdeckungstabelle

Beispiel:

k	PI	j							p_i	c_i
		0	2	4	11	12	14	16		
1	$x_4 x_2$					x		x	p_1	c_1
2	$\overline{x_3} x_2$		x			x			p_2	c_2
3	$x_4 x_3 \overline{x_1}$						x	x	p_3	c_3
4	$\overline{x_4} \overline{x_3} \overline{x_1}$				x				p_4	c_4
5	$\overline{x_4} \overline{x_2} \overline{x_1}$	x		x					p_5	c_5
6	$x_4 x_3 x_1$	x	x						p_6	c_6
7	$x_3 x_2 x_1$			x			x		p_7	c_7

Dominanzregeln und Ihre Anwendung (1)

Spaltendominanzregeln:

■ Spaltendominanz in der Überdeckungstabelle:

i_1	i_2	
X	X	p_1
		p_2
X	X	p_3
X		p_4
X		p_5

i_1	i_2	
X	X	p_1
		p_2
X	X	p_3
X		p_4
X		p_5

■ Wenn die Spalte i_2 durch einen der Terme p_1 oder p_3 *überdeckt* wird, so wird dadurch auch i_1 überdeckt

also: Einstelle der Spalte i_1 ist durch p_1 oder p_3 auch realisiert

■ Man sagt: ("Vektor" i_1) **dominiert** ("Vektor" i_2) und schreibt: $i_1 \geq i_2$

■ **Dominierende Spalten** (hier: i_1) können **gestrichen** werden

Dominanzregeln und Ihre Anwendung (2)

Zeilendominanzregeln:

■ Zeilendominanz in der **Überdeckungstabelle**:

i_1	X	X	X	X	p_1	c_1
i_2		X			p_2	c_2
...						
i_k	X		X	X	p_k	c_k

i_1	X	X	X	X	p_1	c_1
i_2		X			p_2	c_2
...						
i_k	X		X	X	p_k	c_k

■ Wenn eine **Zeile i_2** ($\cong$ **Primterm p_2**) nur **Spalten** ($\cong$ **Einstellen**) **überdeckt**, die auch von einer **anderen Zeile i_1** ($\cong$ **Primterm p_1**) **überdeckt** werden

→ i_2 wird von i_1 *dominiert*: $i_2 \leq i_1$, und

es gilt: (Kosten c_1) $\leq$ (Kosten c_2)

→ die Zeile i_2 kann gestrichen werden

Überdeckungsproblem (Auswahlproblem):

Petrick-Verfahren:

- Das **Petrick-Verfahren** ist eine **algebraische Methode**
 - dient zur **Bestimmung kostenminimaler Lösungen** von **Überdeckungsproblemen**
- Es kann **direkt nach** dem **Nelson-Verfahren** zur **kostenminimalen Auswahl** von **Primtermen** verwendet werden
- **Spaltendominanzen** und **Zeilendominanzen** können **algebraisch** mit Hilfe des Petrick-Verfahrens in Form von **Absorptionsregeln** bzw. **Ausdistribuierten abgearbeitet** bzw. **bewiesen** werden
- **Petrick-Verfahren** wird bei **zyklischen Resttabellen** verwendet

Minimierungsmethoden

Überdeckungsproblem (Auswahlproblem)

Beispiel:

k	PI	j							p_i	C_i
		0	2	4	11	12	14	16		
1	$x_4 x_2$					x		x	p_1	C_1
2	$\overline{x_3} x_2$		x			x			p_2	C_2
3	$x_4 x_3 \overline{x_1}$						x	x	p_3	C_3
4	$\overline{x_4} x_3 x_1$				x				p_4	C_4
5	$\overline{\overline{x_4} x_2 x_1}$	x		x					p_5	C_5
6	$\overline{\overline{x_4} x_3 x_1}$	x	x						p_6	C_6
7	$\overline{\overline{x_3} x_2 x_1}$			x			x		p_7	C_7

$$PA = (\overline{p_5} \vee \overline{p_6}) \& (\overline{p_2} \vee \overline{p_6}) \& (\overline{p_5} \vee \overline{p_7}) \& p_4 \& (p_1 \vee p_2) \& (p_3 \vee p_7) \& (p_1 \vee p_3) = 1$$

- Im **Petrick-Ausdruck** kann man erfassen (beweisen!), dass die **Präsenzvariablen** der **Kernimplikanten** (p_4) **immer eins** sein müssen

Überdeckungsproblem (Auswahlproblem)

- Auch der **Petrick-Ausdruck** wird durch **Distributionsregeln** und **Absorptionsregeln** vereinfacht

Beispiel:

$$PA = (p_5 \vee p_6) \& (p_2 \vee p_6) \& (p_5 \vee p_7) \& p_4 \& (p_1 \vee p_2) \& (p_3 \vee p_7) \& (p_1 \vee p_3) = 1$$

= ...

$$= p_1 p_4 p_6 p_7 \vee p_2 p_3 p_4 p_5 \vee p_1 p_3 p_4 p_5 p_6 \vee p_2 p_3 p_4 p_6 p_7 \vee p_1 p_2 p_4 p_5 p_7 = 1$$

$$= p_4 \& (p_1 p_6 p_7 \vee p_2 p_3 p_5 \vee p_1 p_3 p_5 p_6 \vee p_2 p_3 p_6 p_7 \vee p_1 p_2 p_5 p_7) = 1$$

$$PA = p_1 p_4 p_6 p_7 \vee p_2 p_3 p_4 p_5 \vee p_1 p_3 p_4 p_5 p_6 \vee p_2 p_3 p_4 p_6 p_7 \vee p_1 p_2 p_4 p_5 p_7 = 1$$

Überdeckungsproblem (Auswahlproblem)

- Damit der **Petrick-Ausdruck** '1' wird, muss **mindestens einer** der disjunktiv verknüpften **konjunktiven Teilterme** eins sein
- Jeder **konjunktive Teilterm** repräsentiert genau eine **mögliche Lösung** des **Auswahlproblems**
- Zur **Auswahl** der **optimalen Lösung**:
 - die **Kosten** c_k der **Primterme** p_k werden herangezogen
 - **Anzahl** der **Literale** von p_k
- **Vergleich** der **Kostensummen** K aller möglichen **Lösungen**
(K = Summe der Literale der jeweils notwendigen Primterme)
 - diejenige **Überdeckung** mit **optimalen Kosten** kann ermittelt werden

Minimierungsmethoden

Überdeckungsproblem (Auswahlproblem)

Beispiel:

$$PA = p_1 p_4 p_6 p_7 \vee p_2 p_3 p_4 p_5 \vee p_1 p_3 p_4 p_5 p_6 \vee p_2 p_3 p_4 p_6 p_7 \vee p_1 p_2 p_4 p_5 p_7 = 1$$

mit $c_1=c_2=2$; $c_3 \dots c_7=3$

$p_1 p_4 p_6 p_7$ $K = 11$

$p_2 p_3 p_4 p_5$ $K = 11$

$p_1 p_3 p_4 p_5 p_6$ $K = 14$

$p_2 p_3 p_4 p_6 p_7$ $K = 14$

$p_1 p_2 p_4 p_5 p_7$ $K = 13$

k	PI	j							p_i	c_i
		0	2	4	11	12	14	16		
1	$x_4 x_2$					x		x	p_1	c_1
2	$\overline{x_3 x_2}$		x			x			p_2	c_2
3	$x_4 x_3 \overline{x_1}$						x	x	p_3	c_3
4	$\overline{x_4 x_3 x_1}$				x				p_4	c_4
5	$\overline{x_4 x_2 x_1}$	x		x					p_5	c_5
6	$x_4 x_3 x_1$	x	x						p_6	c_6
7	$x_3 x_2 x_1$			x			x		p_7	c_7

- Damit ergeben sich **zwei kostenminimale Lösungen:**

$p_1 p_4 p_6 p_7$ und $p_2 p_3 p_4 p_5$

Maxterme

Bilden Sie alle Maxterme

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

b | d

Maxterme:

$$\begin{aligned}
 & (a + b + c + d), (\bar{a} + b + c + d), (\bar{a} + b + \bar{c} + d), \\
 & (\bar{a} + \bar{b} + c + d), (\bar{a} + \bar{b} + c + \bar{d})
 \end{aligned}$$

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

	1	4	5	14	16	
$\overline{b}c\overline{d}$		X	X			
bcd					X	
$\overline{a}c\overline{d}$				X	X	
$\overline{a}\overline{b}c$		X		X		
$a\overline{d}$	X		X			

Petrick-Ausdruck

Ordnen Sie den Primimplikanten eine Präsenzvariable zu und geben Sie den Petrickausdruck an.

	1	4	5	14	16	Präsenzvar.
$\overline{b}c\overline{d}$		X	X			p_1
bcd					X	p_2
$\overline{a}c\overline{d}$				X	X	p_3
$\overline{a}bc$		X		X		p_4
$a\overline{d}$	X		X			p_5

$$P_e = p_5 \cdot (p_1 + p_4) \cdot (p_1 + p_5) \cdot (p_3 + p_4) \cdot (p_2 + p_3)$$

$$= p_1 p_2 p_4 p_5 + p_1 p_3 p_5 + p_2 p_4 p_5 + p_3 p_4 p_5$$



Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

Entwicklungssatz

- Bezeichnet eine formale Methode, die es erlaubt, eine Funktion $f(x_n, \dots, x_3, x_2, x_1)$ in eine DNF (oder KNF) umzuwandeln
- Die Entwicklung der Funktion f nach der Variablen x_i
→ Aufteilung in zwei Fälle: $x_i = 1$ und $x_i = 0$

$$\begin{aligned} f(x_n, \dots, x_i, \dots, x_1) &= [x_i \& f(x_n, \dots, 1, \dots, x_1)] \vee [\bar{x}_i \& f(x_n, \dots, 0, \dots, x_1)] \\ &= [x_i \& f \Big|_{x_i=1}] \vee [\bar{x}_i \& f \Big|_{x_i=0}] \end{aligned}$$

- Die nach x_i entwickelten Funktionen $f \Big|_{x_i=1}$ und $f \Big|_{x_i=0}$ werden als Restfunktionen oder Kofaktoren bezeichnet

- Bei der Auswertung des Entwicklungssatzes für n Variablen: n Klammerebenen im Funktionsterm:

$$F(d, c, b, a) = b[c(1) \vee \bar{c}(\bar{a})] \vee \bar{b}[c(d(1) \vee \bar{d}(\bar{a})) \vee \bar{c}(d(a) \vee \bar{d}(0))]\dots$$



- Methode zur erleichterten Übersicht: Rechnen mit Restfunktionen (für Multiplexer-Realisierung hilfreich):

- Beispiel: $y = f(x_1, x_2, x_3, \dots)$

- Entwicklung nach x_1 :

$$f(1, x_2, x_3, \dots) = ?,$$

$$f(0, x_2, x_3, \dots) = ?$$

- Entwicklung nach x_2 :

$$f(1, 1, x_3, \dots) = ?,$$

$$f(0, 1, x_3, \dots) = ?,$$

$$f(1, 0, x_3, \dots) = ?,$$

$$f(0, 0, x_3, \dots) = ?$$

- Entwicklung nach x_3 : (...), usw.

- Weitere Auffächerung der Restfunktionen wird abgebrochen, sobald konstant wertig

Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

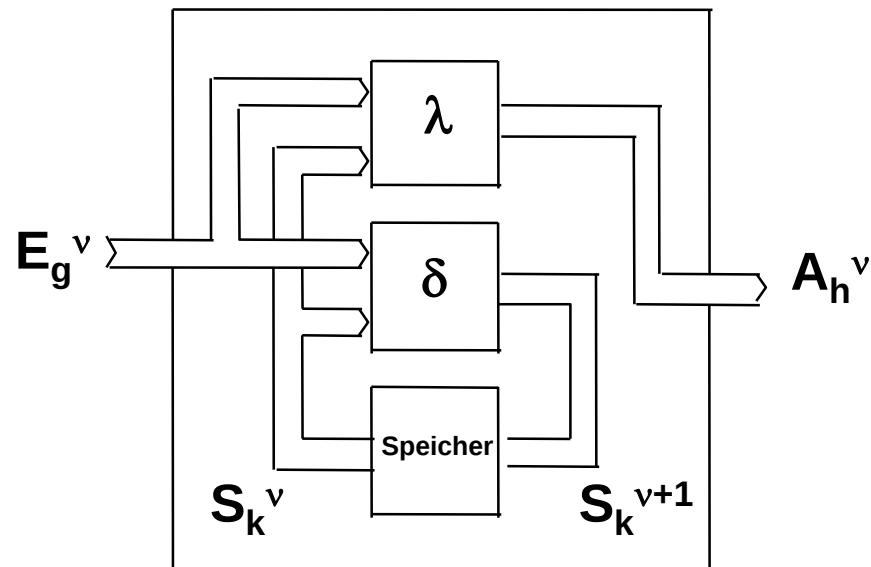
Automaten

Wichtige Typklassen von Automaten im Zusammenhang mit Digitalschaltungen:

- *endliche, diskrete* und *deterministische* Automaten
- Bezüglich der Ausgabefunktion unterscheidet man *drei Fälle*
- Mealy-Automat: mit $A_h^v = \lambda(E_g^v, S_k^v)$ als allgemeinster Fall:

- A_h^v : Ausgansvektor
- S_k^v : Zustandsvektor
- E_g^v : Eingangsvektor
- λ : Ausgabefunktion
- δ : Transitionsfunktion

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

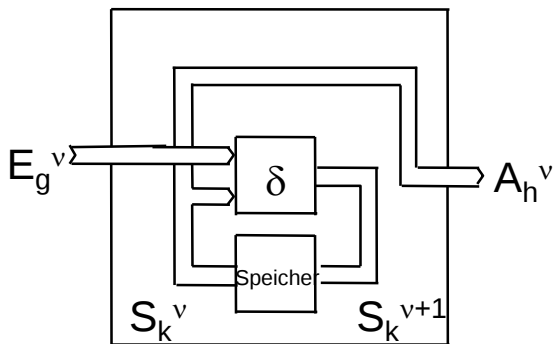
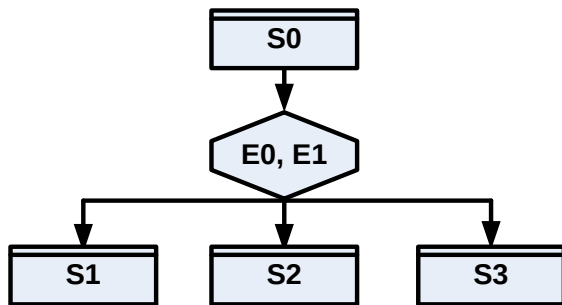


Automatentypen

Es gibt drei Typen von Automaten:

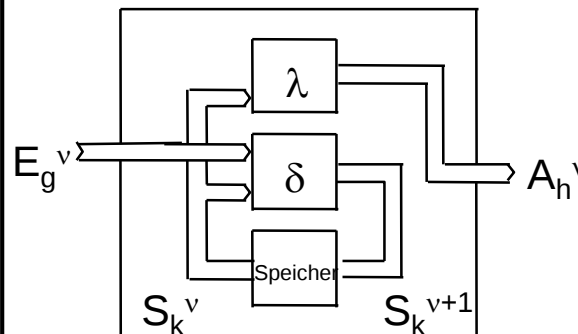
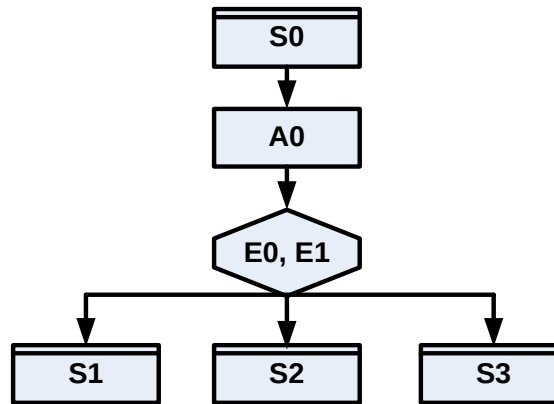
Medwedew-Automat

$$A_h^v = S_k^v$$



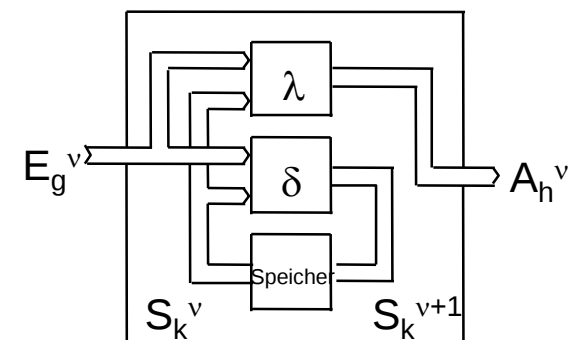
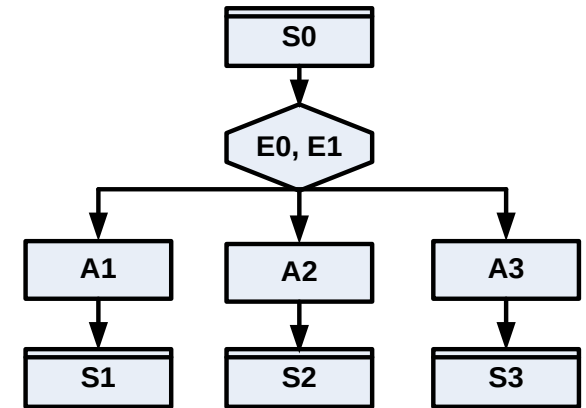
Moore-Automat

$$A_h^v = \lambda (S_k^v)$$



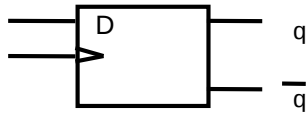
Mealy-Automat

$$A_h^v = \lambda (E_g^v, S_k^v)$$



Binärspeicher (FlipFlops)

Symbole



Charakteristische Gleichungen

RS-FlipFlop

$$q^{v+1} = S \vee (q^v \& \bar{R})$$

D-FlipFlop

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

T-FlipFlop

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

JK-FlipFlop

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

Ansteuerfunktionen

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

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

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

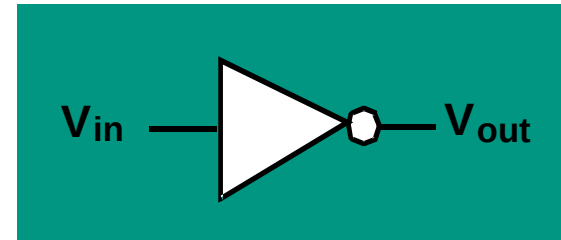
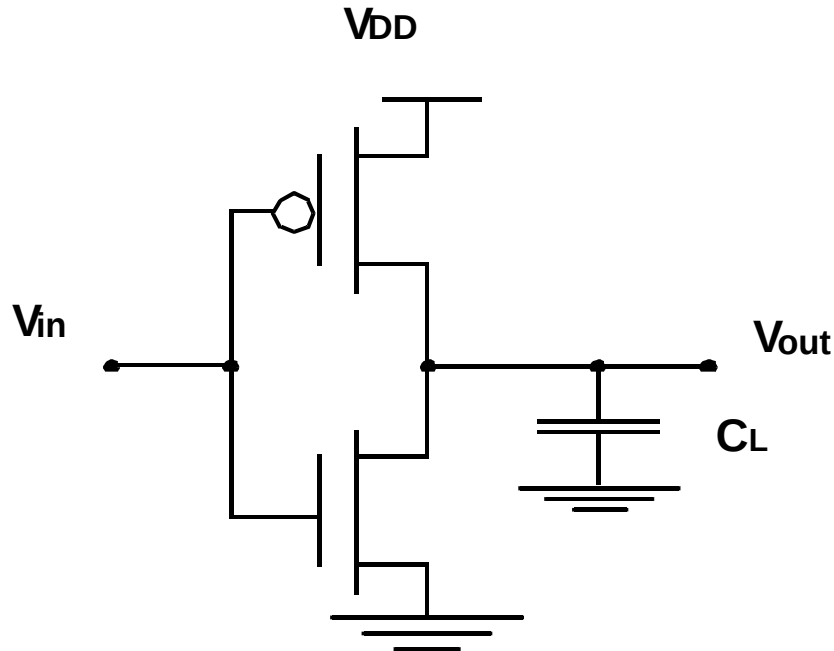
q^v	q^{v+1}	K	J
0	0	-	0
0	1	-	1
1	0	1	-
1	1	0	-

Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

CMOS

CMOS - Inverter



- Der **Inverter** besteht aus **zwei Transistoren**, einem NMOS und einem PMOS
- Ist V_{in} auf **high**, so **öffnet** der **NMOS-Transistor** und V_{out} wird auf **GND** gezogen, während dessen ist der **PMOS-Transistor gesperrt**
- Ist V_{in} hingegen auf **low**, so **öffnet** der **PMOS-Transistor** und V_{out} wird auf **V_{DD}** gezogen, der **NMOS-Transistor** wird hingegen **gesperrt**
- Strom zwischen V_{DD} und GND kann nur fließen, wenn die Transistoren umschalten und kurzzeitig beide offen sind

- P-Netz (v_1) stellt Verbindung zu V_{DD} her
- N-Netz (v_0) stellt Verbindung zu GND her
- Forderungen an fehlerfreies und wohldefiniertes Verhalten:

1. $v_0 \cdot v_1 = 0$ (kein Kurzschluss)
2. $v_0 + v_1 = 1$ (Vollständigkeit)

oder auch wenn Dualität gilt:

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

- Bedeutung im Symmetriediagramm:
 - Keine Überlappung von Einsen und Nullen (1)
 - Alle Felder im S-Diagramm sind belegt (2)
(kein undefiniertes Verhalten)

Punkteverteilung Klausur

- Auf dem Deckblatt der Klausur wird die Gewichtung der einzelnen Aufgaben angegeben:

Matrikelnummer: _____ Name: _____

Prüfung

ITV Prof. Dr.-Ing. J. Becker
Digitaltechnik
WS 2009/2010

Institut für Technik der Informationsverarbeitung, KIT

Klausur
Mo., 15.03.2010
Lösungsblätter

Hinweise zur Bearbeitung:
Hilfsmittel
Als Hilfsmittel zur Bearbeitung sind vier Seiten vorgegebene und ein **DIN A4 Blatt** selbst geschriebene Formeln und Formeln zugelassen. Nicht erlaubt hingegen sind die Verwendung eines Taschenrechners, zusätzlicher Unterlagen und jegliche Kommunikation mit anderen Personen.

Prüfungsdauer
Die Prüfungsdauer beträgt für die Aufgabenblätter 20 Minuten.
Prüfungsunterlagen
Die Prüfungsunterlagen bestehen aus insgesamt 24 Seiten: Aufgabenblättern (einschließlich diesem Titelblatt und zusätzlichen Lösungsblättern). 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 Ihre Matrikelnummer!

Auf jedes zusätzliche Lösungsblatt ist neben dem Namen und der Matrikelnummer 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 Rotstift!**

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.....	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		

Felix Pistorius
pistorius@kit.edu

Karlsruher Institut für Technologie (KIT)

Viel Erfolg!!!!