

Info2: 1 Nachrichten

Alphabet: Menge an Zeichen

Wort: Folge an Zeichen aus Alphabet

Kardinalität: # Wörter A^m der Länge m , die möglich sind

Optimalcode: min. mittlere Wortlänge

Prefix-Bedingung: kein Wort ist Anfang anderes Wortes

Blockcode: feste Wortlänge

Informationsgehalt: $I(a_j) = \log\left(\frac{1}{p_j}\right) = -\log(p_j)$ $[I] = \text{bit}$

Entropie: Informationsgehalt einer Nachricht / benötigte Bits

$$H = \sum_{j=1}^n p_j \cdot \log\left(\frac{1}{p_j}\right) = -\sum_{j=1}^n p_j \cdot \log(p_j) \quad [H] = \text{bit}$$

Max Entropie: $H_0 = -\sum_{j=1}^n \frac{1}{n} \cdot \log\left(\frac{1}{n}\right) = \log(n)$ $[H_0] = \text{bit}$

Redundanz: $R = H_0 - H$ $[R] = \text{bit}$

relative Redundanz: $r = R/H_0$ (einheitenlos)

n Stelle k -facher Fehler, m Wortlänge, p Stelle flipped bit

$$P(k) = \binom{m}{k} \cdot p^k \cdot (1-p)^{m-k}$$

Hamming-Distanz: # verschiedene Bits

erkennbare Fehler: $e = (d_m - 1)$ $d_m = \text{min. Hamming Dist.}$

korrigierbare Fehler: $k = \left\lfloor \frac{d_m - 1}{2} \right\rfloor = \left\lfloor \frac{e}{2} \right\rfloor$

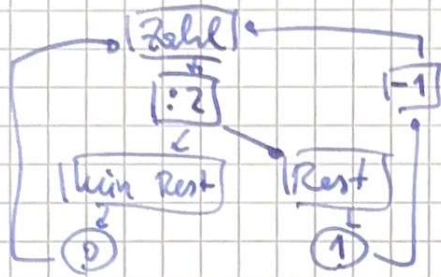
BCD-Code: Ziffern einzeln gespeichert,

masked codes = Pseudo/extra Codes

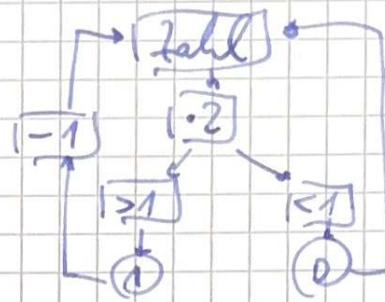
2 Binärzahlen

$-10 \rightarrow -2$:

vor Komma:



nach Komma:



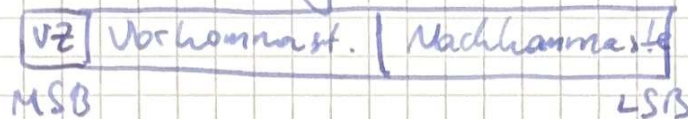
Negative Darstellung:

- VZ-Bit vorne (0+ 1-)
- Einerskomplement ($\neg$ Pos)
- Zweierskomplement ($\neg$ Pos +1)

Subtraktion = Addition des Zweierskomplements

3 Floats

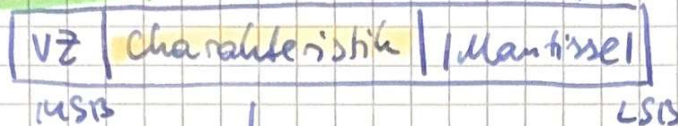
Festkommadarstellung:



Gleitkommadarstellung:

$$Z = M \cdot B^E$$

M: Mantisse, E: Exponent, B: Basis₋₂



$$C = E + \left(\frac{1}{2} B^c - 1\right)$$

C = Länge d. Charakteristik

IEEE 754:

$$C = E + \left(\frac{1}{2} 2^8 - 1\right) = E + 127$$

M: 1, Mantisse
implizit

C = M = 0: Null

C = 255: NaN

+/-

- ▷ kleinerer Exponent zu größerem angleichen
- ▷ Mantissen addieren
- ▷ Normalisieren

*/÷

- ▷ Mantissen verrechnen
- ▷ Exponenten addieren/subtrahieren
- ▷ VZ setzen
- ▷ Normalisieren

4) Boolesche Algebra

- Kommutativ
- Distributiv
- Neutrales Element
- Komplement

$$\begin{aligned} \cdot &= \wedge \\ + &= \vee \end{aligned}$$

Absorption: $a \cdot (a + b) = a$, $a + (b \cdot a) = a$

Morgan: $\overline{a \cdot b} = \bar{a} + \bar{b}$, $\overline{a + b} = \bar{a} \cdot \bar{b}$
Shannon

DNF: $(a \cdot b) + (a \cdot \bar{c}) + (b \cdot \bar{c})$ (Minterme)

KNF: $(a + b) \cdot (a + c) \cdot (b + \bar{c})$ (Maxterme)

DNF:

a	b	f(a,b)
0	0	0
0	1	1
1	0	0
1	1	1

$$f(a,b) = (\bar{a} \cdot b) + (a \cdot b)$$

KNF:

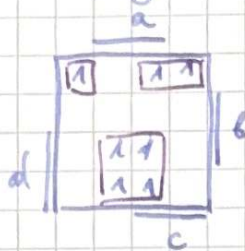
a	b	f(a,b)
0	0	0
0	1	1
1	0	0
1	1	1

$$f(a,b) = (a + b) \cdot (\bar{a} + b)$$

Variablen reduziert!

5) Minimierung

Karnaugh-Veitch- / KV-Diagramme



6) Quine-McCluskey

- 1) Äquivalenzklassen (# nicht-negative vars)
- 2) Dezimalwert jedes Minterms notieren
- 3) Klassen zusammenfassen
 - wenn nur eine Variable Unterschied → auslassen
 - neue Klassen
- 4) wiederholen
- 5) Matrix (Überdeckungsmatrix)
 - Spalten = Dezimalwerte d. ursprünglichen Terme
 - Zeilen = Reduzierte Terme (Primimplikanten)
- 6) wesentliche Primimplikanten auswählen
 - die, die auf einen Term abdecken
- 7) Restmatrix → wiederholen

(7) Logische Schaltungen

 AND

 NOR

 OR

 XOR

 NOT

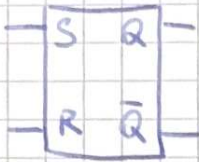
 $\bar{a}b$

 NAND

...

[8] Flip-Flops

RS-Flip-Flop



S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	/

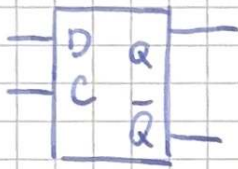
Mit Zustands-
änderung

= nur Update,
wenn Clock = 1

Mit Rückflanken-
triggerung

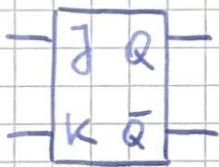
= Update, wenn
Clock 1 → 0

D-Flip-Flop



C	D	Q_{n+1}
0	0	Q_n
0	1	Q_n
1	0	0
1	1	1

JK-Flip-Flop



J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\bar{Q}_n$

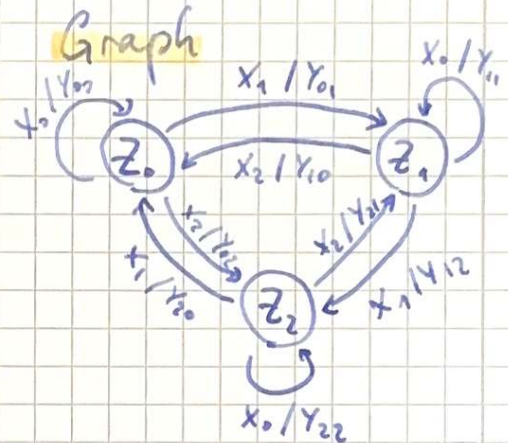
9 Automaten

$$A = (X, Y, Z, f, g)$$

$g(Y^{n+1} = g(X^{n+1}, Z^n))$
 Ausgangsfkt
 $f(Z^{n+1} = f(X^{n+1}, Z^n))$
 Übergangsfkt
 Zustandsmenge
 Ausgangsmenge
 Eingangsmenge

Tabelle

n	n+1		
	X_0	X_1	X_2
Z_0	Z_0/Y_{00}	Z_1/Y_{01}	Z_2/Y_{02}
Z_1	Z_1/Y_{11}	Z_2/Y_{12}	Z_0/Y_{10}
Z_2	Z_2/Y_{22}	Z_0/Y_{20}	Z_1/Y_{21}



- Mealy-Automat
- Moore-Automat: Ausgabe hängt nur von Zustand ab, nicht von Eingabe
- Zuordner: nur ein Zustand
- autonomer: nur eine Eingabe
- Halb-Aut: nur eine Ausgabe

Moore → Mealy

n	n+1			n
	X_0	X_1	X_2	
Z_0	Z_0/Y_{00}	Z_1/Y_{01}	Z_2/Y_{02}	Y_0
Z_1	Z_1/Y_{11}	Z_2/Y_{12}	Z_0/Y_{10}	Y_1
Z_2	Z_2/Y_{22}	Z_0/Y_{20}	Z_1/Y_{21}	Y_2

Mealy → Moore

n	n+1		n
	X_0	X_1	
$Z_{00} = (Z_0 \times X_0)$	Z_0/Y_0	Z_1/Y_1	Y_0
$Z_{01} = (Z_0 \times X_1)$	Z_1/Y_1	Z_2/Y_2	Y_1
$Z_{10} = (Z_1 \times X_0)$	Z_2/Y_2	Z_0/Y_0	Y_2
$Z_{11} = (Z_1 \times X_1)$	Z_0/Y_0	Z_1/Y_1	Y_0
$Z_{20} = (Z_2 \times X_0)$	Z_1/Y_1	Z_2/Y_2	Y_1
$Z_{21} = (Z_2 \times X_1)$	Z_2/Y_2	Z_0/Y_0	Y_2

9a) Automaten

Zustände äquivalent $\rightarrow$ auf beliebige Eingabefolge reagiert gleich egal bei welchem Zustand angefangen

k-äquivalent: gleiche Reaktion auf Eingabefolge der Länge k

Moore-Reduktion

	x_0	x_1	x_2	
z_1	$z_{10} - U_1^0$	$z_1 - U_2^0$	$z_6 - U_2^0$	Y_2
z_2	$z_{11} - U_1^0$	$z_2 - U_3^0$	$z_6 - U_2^0$	Y_2
z_3	$z_1 - U_2^0$	$z_3 - U_1^0$	$z_5 - U_2^0$	$Y_1 \rightarrow K_1^0$
z_4	$z_9 - U_1^0$	$z_2 - U_2^0$	$z_3 - U_1^0$	Y_2
z_5	$z_{10} - U_1^0$	$z_1 - U_2^0$	$z_6 - U_2^0$	Y_1
z_6	$z_8 - U_1^0$	$z_2 - U_3^0$	$z_{11} - U_1^0$	$Y_2 \rightarrow K_2^0$
z_7	$z_8 - U_3^0$	$z_2 - U_3^0$	$z_{11} - U_1^0$	Y_2
z_8	$z_6 - U_2^0$	$z_{11} - U_1^0$	$z_5 - U_1^0$	$Y_2 \rightarrow K_3^0$
z_9	$z_1 - U_2^0$	$z_3 - U_1^0$	$z_5 - U_1^0$	Y_1
z_{10}	$z_3 - U_1^0$	$z_5 - U_3^0$	$z_7 - U_2^0$	Y_2
z_{11}	$z_1 - U_3^0$	$z_6 - U_3^0$	$z_1 - U_1^0$	Y_1

1) 0-Äquivalenzklassen (gleiche Ausgabe)

	x_0	x_1	x_2	
z_3	$z_1 - U_2^0$	$z_2 - U_3^0$	$z_6 - U_2^0$	Y_1
z_5	$z_{10} - U_1^0$	$z_1 - U_2^0$	$z_6 - U_2^0$	Y_1
z_{11}	$z_1 - U_3^0$	$z_6 - U_3^0$	$z_1 - U_1^0$	$Y_1 \rightarrow K_1^1$
z_1	$z_{10} - U_1^0$	$z_1 - U_2^0$	$z_6 - U_2^0$	$Y_1 \rightarrow K_2^1$
z_2	$z_{11} - U_1^0$	$z_2 - U_3^0$	$z_6 - U_2^0$	$Y_2 \rightarrow K_3^1$
z_4	$z_9 - U_1^0$	$z_2 - U_2^0$	$z_3 - U_1^0$	Y_2
z_6	$z_8 - U_1^0$	$z_2 - U_3^0$	$z_{11} - U_1^0$	$Y_2 \rightarrow K_4^1$
z_7	$z_8 - U_3^0$	$z_2 - U_3^0$	$z_{11} - U_1^0$	Y_2
z_{10}	$z_3 - U_1^0$	$z_5 - U_3^0$	$z_7 - U_2^0$	Y_2
z_8	$z_6 - U_2^0$	$z_{11} - U_1^0$	$z_5 - U_1^0$	$Y_1 \rightarrow K_5^1$
z_9	$z_1 - U_2^0$	$z_3 - U_1^0$	$z_5 - U_1^0$	Y_1

2) 1-Äquivalenzklassen (gleiche Klasse der Übergangszustände)

3) wieder teilen bis keine Aufteilung mehr möglich

minimierten Automaten ablesen:

$\rightarrow$ jede Klasse ist ein Zustand

$\rightarrow$ Übergänge in die Klassen der alten Zustände (= neue Zustände)

10 Realisierung von Automaten

- 1) Automaten erstellen (Graph/Tabelle)
- 2) Codierung festlegen $\{x_0=0, x_1=1, z_0=00, z_1=01, z_2=10, \dots\}$
- 3) Bauteil aussuchen (J-K-Flip-Flop)

J	K	q_{n+1}
0	0	q_n
0	1	0
1	0	1
1	1	q_n

$q_i \rightarrow q_{i+1}$	J	K
0 $\rightarrow$ 0	0	*
0 $\rightarrow$ 1	1	*
1 $\rightarrow$ 0	*	1
1 $\rightarrow$ 1	*	0

4) Produktionstabelle

x^{n+1}	z^n	z^{n+1}							
x_0	q_1	q_0	q_1	q_0	J_1	K_1	J_0	K_0	
0	0	0	1	0	1	*	0	*	

Eingabe Zustand Übergang zu Flip-Flop-Funktion

5) Minimieren (jeden Eingang, J_1, K_1, J_0, K_0)

→ definiert über x_0, q_1, q_0

q_1
1 * * 0
0 * * 1
x_0

→ viele Don't-Care-Felder, weil nicht jede Kombi vorkommt

6) Minimale Funktion als Schaltung zeichnen

11 ALU

Flags: $C \rightarrow \text{Carry} \rightarrow \text{übertrag beim letzten bit}$
 $N \rightarrow \text{Negative} \rightarrow \text{erstes bit} = 1$
 $V \rightarrow \text{overflow} \rightarrow \text{schon ein bit bei letzten zwei überträgen} = 1$
 $Z \rightarrow \text{zero} = 0$

Vergleich von R_1 u. R_2 :

► $K_2(R_2)$, Addition

► Flags: $Z = 1$ $R_1 = R_2$
 $Z = 0$ $N \oplus V$ $R_1 < R_2$
 $Z = 0$ $N = V$ $R_1 > R_2$

12 Assembler

LDI rd, K

lade K in rd

MOV rd, rr

$rr \rightarrow rd$

ADD/SUB/MUL rd, rr

Mathe $\rightarrow rd$, bei MUL $\rightarrow R1:R0$

CP rd, rr / CPI rd, K

BREQ / BRLO K

springe um $K+1$, falls Z bzw. C gesetzt

RJMP

springe um $K+1$

PUSH rr / POP rd

Stack