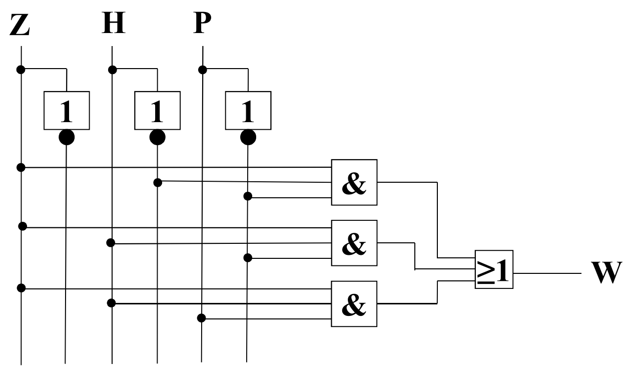
Beispiel Warnleuchte W im Auto: Leuchtet wann ?

Zündung: Z=0: Zündung aus, Z=1: Zündung an
Kühlwasser: H=0: Temperatur OK, H=1: Temperatur>95°
Wasser-Pegel: P=0: kein Wasser, P=1: ausreichend Wasser

Z	H	P	W	Minterm
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	0	
1	0	0	1	$Z\overline{H}\,\overline{P}$
1	0	1	0	
1	1	0	1	$ZH\overline{P}$
1	1	1	1	ZHP

Warnleuchten-Funktion in KDNF:

$$W = Z\overline{H}\,\overline{P} + ZH\overline{P} + ZHP$$



Ist die KDNF minimal bzgl. der Anzahl der Gatter?

Minimierung der KDNF der Warnleuchte

$$
\begin{aligned}
W &= Z\overline{H}\,\overline{P} + ZH\overline{P} + ZHP \\
&\overset{S3}{=} Z\overline{H}\,\overline{P} + ZH\overline{P} + ZH\overline{P} + ZHP \\
&\overset{A1,A1}{=} Z\overline{P}\,\overline{H} + Z\overline{P}H + ZH\overline{P} + ZHP \\
&\overset{A3,A3}{=} (Z\overline{P}\cdot\overline{H} + Z\overline{P}\cdot H) + (ZH\cdot\overline{P} + ZH\cdot P) \\
&\overset{A3,A3}{=} Z\overline{P}\cdot(\overline{H} + H) + ZH\cdot(\overline{P} + P) \\
&\overset{A4,A4}{=} Z\overline{P}\cdot 1 + ZH\cdot 1 \\
&\overset{A2,A2}{=} Z\overline{P} + ZH \qquad \text{(DMF)}
\end{aligned}
$$

Definition:

Ein Primterm von f ist eine Konjunktion von Variablen, die in f erfüllt ist, für die aber keine echte Teilkonjunktion in f erfüllt ist. Ein Primterm ist ein kleinster zusammenfassbarer Produktterm.

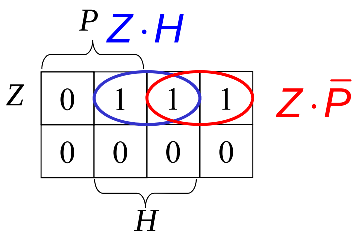
ZH ist ein Primterm der Funktion W.

Satz:

Die disjunktive Minimalform (DMF) einer Funktion ist eine Disjunktion von Primtermen.

Beispiel: Warnleuchte, Disjunktive Minimalform (DMF)

Z	H	P	W
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

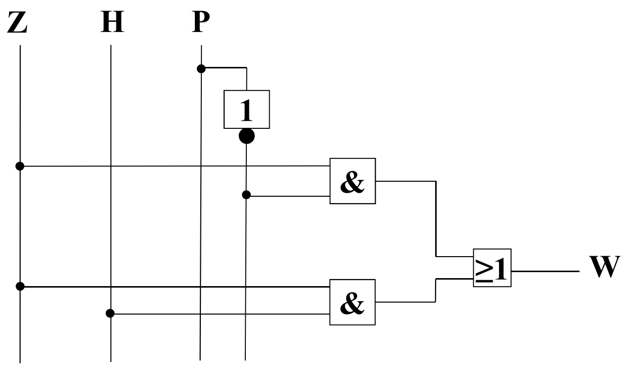


$$W = Z\overline{P} + ZH$$

67

Warnleuchten-Funktion in DMF:

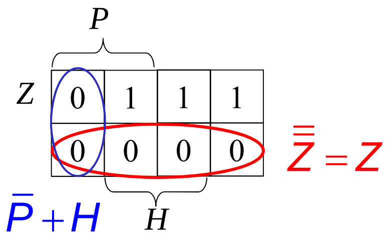
$$W = Z\overline{P} + ZH$$



68

Beispiel: Warnleuchte, Konjunktive Minimalform (KMF)

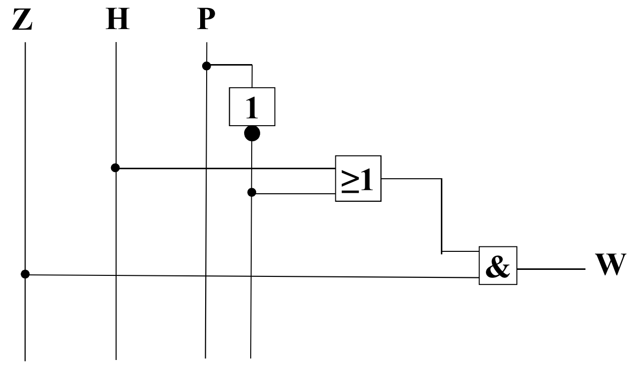
Z	H	P	W
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



$$W = Z(\bar{P} + H)$$

Warnleuchten-Funktion in KMF:

$$W = Z(\bar{P} + H)$$



Beispiel 2-Bit Multiplizierer:

n * m = p

2x2 Eingänge (0-3), 4 Ausgänge (0-9)

n	m	n*m
0	0	0
0	1	0
0	2	0
0	3	0
1	0	0
1	1	1
1	2	2
1	3	3
2	0	0
2	1	2
2	2	4
2	3	6
3	0	0
3	1	3
3	2	6
3	3	9

n_1	n_0	m_1	m_0	p_3	p_2	p_1	p_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

p_0

n_1n_0 \ m_1m_0	00	01	11	10
00				
01		1	1	
11		1	1	
10				

p_1

n_1n_0 \ m_1m_0	00	01	11	10
00				
01			1	1
11		1		1
10		1	1	

p_2

n_1n_0 \ m_1m_0	00	01	11	10
00				
01				
11				1
10			1	1

n_1n_0 \ m_1m_0	00	01	11	10
00				
01		1	1	
11		1	1	
10				

$$p_0 = n_0 \cdot m_0$$

n_1n_0 \ m_1m_0	00	01	11	10
00				
01			1	1
11		1		1
10		1	1	

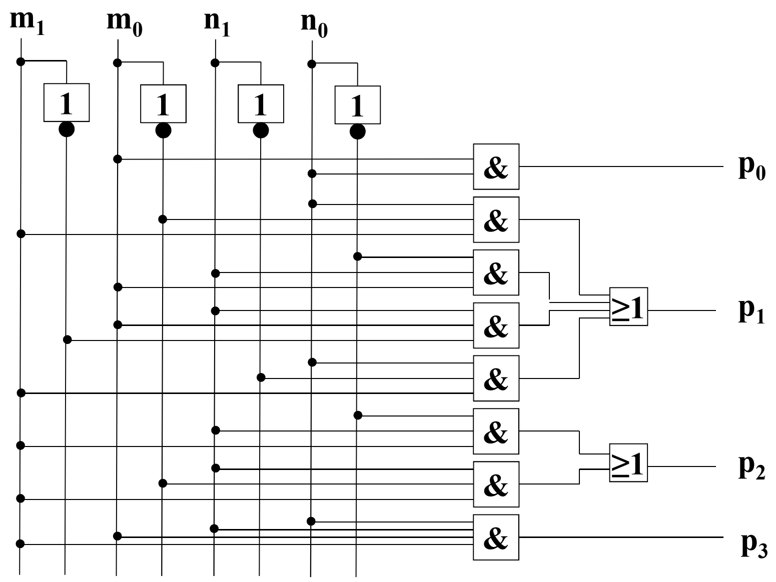
$$p_1 = n_0 \cdot \overline{m_0} \cdot m_1 + \overline{n_0} \cdot n_1 \cdot m_0 + n_1 \cdot m_0 \cdot \overline{m_1} + n_0 \cdot \overline{n_1} \cdot m_1$$

n_1n_0 \ m_1m_0	00	01	11	10
00				
01				
11				1
10			1	1

$$p_2 = \overline{n_0} \cdot n_1 \cdot m_1 + n_1 \cdot \overline{m_0} \cdot m_1$$

$$p_3 = n_0 \cdot n_1 \cdot m_0 \cdot m_1$$

Schaltnetz eines 2-Bit-Multiplizierers



73

Gray-Code zur Erzeugung der Belegung im KV-Diagramm

Gray-Code: Datenrepräsentation, bei dem sich zwischen benachbarten Einträgen einer Folge immer nur 1 Bit ändert. Das ändert die Reihenfolge der Zahlen. Beispiel einer Bitfolge von 2 Bit in binär und Gray Code:
2 Bit binär: n_2 = {00,01,10,11} , Dezimal n_{10} = {0,1,2,3}
Gray Code: n_2 = {00,01,11,10} , Dezimal n_{10} = {0,1,3,2}

Im 2D-KV-Diagramm mit (m+n) Variablen werden die Variablenbelegungen mit Graycodes entsprechend horizontal (n) und vertikal (m) angeordnet, so dass sich in benachbarten Zellen immer nur 1 Bit horizontal oder vertikal ändert. Dabei werden die niederwertigen n Bits horizontal und die höherwertigen m Bits vertikal angeordnet.

Vier Variablen:
n=m=2

x_3x_2 \ x_1x_0	00	01	11	10
00	0000	0001	0011	0010
01	0100	0101	0111	0110
11	1100	1101	1111	1110
10	1000	1001	1011	1010

x_2 x_3 x_0 x_1

74

KV-Diagramme mit 5 Variablen: n=3, m=2

Spiegelachse

x_4x_3 \ $x_2x_1x_0$	000	001	011	010	110	111	101	100
00	00000	00001	00011	00010	00110	00111	00101	00100
01	01000	01001	01011	01010	01110	01111	01101	01100
11	11000	11001	11011	11010	11110	11111	11101	11100
10	10000	10001	10011	10010	10110	10111	10101	10100

Spiegelsymmetrisches Zusammenfassen

3 Bit Gray-Code: (2 x 2-Bit Code, gespiegelt an der Mitte)
$x_2x_1x_0$ = (000,001,011,010,110,111,101,100), n=(0,1,3,2,6,7,5,4)

KV-Diagramme mit 5 Variablen, n=3,n=2

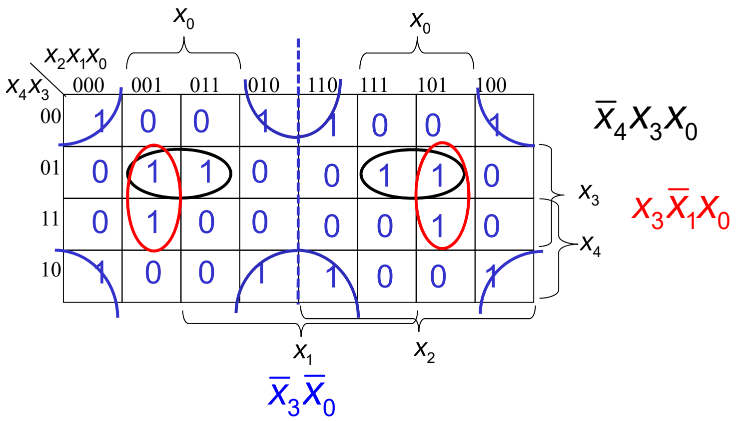
x_4x_3 \ $x_2x_1x_0$	000	001	011	010	110	111	101	100
00	00000	00001	00011	00010	00110	00111	00101	00100
01	01000	01001	01011	01010	01110	01111	01101	01100
11	11000	11001	11011	11010	11110	11111	11101	11100
10	10000	10001	10011	10010	10110	10111	10101	10100

x_0 x_0 x_3 x_4 x_1 x_2

Die Regeln des Zusammenfassens von Mintermen folgen denen der KV-Diagramme für 4 Variablen, aber zusätzlich zerfällt x0 in zwei Komponenten, die ebenfalls über die Spiegelachse horizontal zusammengefasst werden können.

Gray-Code für 5 Variablen: zerlege in Graycode für 3 und 2 Variablen
$x_2x_1x_0$ = (000,001,011,010,110,111,101,100), n=(0,1,3,2,6,7,5,4)
x_4x_3 = (00,01,11,10), n= (0,1,3,2)

KV-Diagramme mit 5 Variablen, n=3,n=2



$$f = \bar{x}_3\bar{x}_0 + x_3\bar{x}_1x_0 + \bar{x}_4x_3x_0$$

77

KV-Diagramme mit 6 Variablen

$x_5x_4x_3$ \ $x_2x_1x_0$	000	001	011	010	110	111	101	100
000	000000	000001	000011	000010	000110	000111	000101	000100
001	001000	001001	001011	001010	001110	001111	001101	001100
011	011000	011001	011011	011010	011110	011111	011101	011100
010	010000	010001	010011	010010	010110	010111	010101	011000
110	110000	110001	110011	110010	110110	110111	110101	111000
111	111000	111001	111011	111010	111110	111111	111101	111100
101	101000	101001	101011	101010	101110	101111	101101	101100
100	100000	100001	100011	100010	100110	100111	100101	100100

3 Bit Gray-Code: (2-Bit Code gespiegelt an der Mitte)
$x_2x_1x_0$ = (000,001,011,010,110,111,101,100), n=(0,1,3,2,6,7,5,4)

78

KV-Diagramme mit 6 Variablen

$x_5x_4x_3$ \ $x_2x_1x_0$	000	001	011	010	110	111	101	100
000	000000	000001	000011	000010	000110	000111	000101	000100
001	001000	001001	001011	001010	001110	001111	001101	001100
011	011000	011001	011011	011010	011110	011111	011101	011100
010	010000	010001	010011	010010	010110	010111	010101	011000
110	110000	110001	110011	110010	110110	110111	110101	111000
111	111000	111001	111011	111010	111110	111111	111101	111100
101	101000	101001	101011	101010	101110	101111	101101	101100
100	100000	100001	100011	100010	100110	100111	100101	100100

x_0 x_0 x_3 x_3 x_4 x_5 x_1 x_2

6 Variablen haben zwei 3-Bit-Graycodes. Die Variablen x_0 und x_3 zerfallen in zwei spiegelsymmetrische Komponenten, horizontal und vertikal.

KV-Diagramme mit 6 Variablen

$x_5x_4x_3$ \ $x_2x_1x_0$	000	001	011	010	110	111	101	100
000	0	0	0	0	0	0	0	0
001	0	1	1	0	0	1	1	0
011	0	1	1	0	0	1	1	0
010	0	1	1	0	0	0	0	0
110	0	1	1	0	0	0	0	0
111	0	1	1	0	0	1	1	0
101	0	1	1	0	0	1	1	0
100	1	0	0	0	0	0	0	0

x_3x_0

$x_4\bar{x}_2x_0$

$x_5\bar{x}_4\bar{x}_3\bar{x}_2\bar{x}_1\bar{x}_0$

x_0 x_0 x_3 x_3 x_4 x_5 x_1 x_2

DMF $f = x_5\bar{x}_4\bar{x}_3\bar{x}_2\bar{x}_1\bar{x}_0 + x_3x_0 + x_4\bar{x}_2x_0$

Das Verfahren von Quine und McCluskey

Mit KV-Diagrammen kommen wir nicht weiter, wenn die Anzahl der Eingabevariablen größer als 6 wird. In diesem Fall empfiehlt sich das Verfahren von Quine und McCluskey. Es beginnt mit der KDNF und besteht aus zwei Schritten:

Erstens: Das Verfahren von McCluskey erzeugt durch systematische Anwendung der Vereinfachungsregeln alle Primterme einer Funktion.

Zweitens: Das Verfahren von Quine wählt aus dieser Menge aller Primterme eine minimale Teilmenge aus, deren Oder-Verknüpfung die gesamte Funktion repräsentiert.

81

Das Verfahren von Quine und McCluskey

1. McCluskey:

Systematische Anwendung des Satz 14 auf Terme *m=xy*, wobei *m* ein beliebiger Produktterm aus *k* Variablen ist, der sich aus einem Teilterm *x* mit (*k-1*) Variablen und der einzelnen Variable *y* zusammen setzt.
Für zwei Terme $m_1 = x \cdot y$ und $m_2 = x \cdot \bar{y}$ gilt nach Satz 14:

$$m_1 + m_2 = x \cdot y + x \cdot \bar{y} = x$$

Der Gruppenterm x ist um eine Variable reduziert.
Wiederholte Zusammenfassung zu Gruppen minimaler Größe führt auf Primterme

2. Quine

Treffen einer minimalen Auswahl von Primtermen, deren Disjunktion die Funktion realisiert.

82

I) Algorithmus von McCluskey zur Findung der Primterme:

Starte bei KDNF und fasse zu Primtermen zusammen.

1. Stelle Minterme in einer Tabelle als binäre Belegungen (Binärdarstellung der Minterme entsprechend der Variablen) dar. Nummeriere die Minterme gemäß der Belegungen.
2. Fasse jeweils 2 Minterme (m_1,m_2) entsprechend Satz 14 zu einer Gruppe (Teilterm x) zusammen, wenn sich nur eine Stelle der Belegung ändert. Hake die bereits verwendeten Minterme ab. Wenn nicht zusammenfassbare Minterme übrig bleiben, sind sie Teil der Primterme.
3. Wiederhole die Zusammenfassung der Teilterme (Gruppen) wie in 2. angegeben zu größeren Einheiten zusammen (kürzere Terme), bis keine Zusammenfassung mehr möglich ist. Hake alle verwendeten Teilterme ab. Die übrig bleibenden Terme ohne Haken sind die Primterme.

83

Beispiel Algorithmus McCluskey:

Starte bei KDNF und fasse zu Primtermen zusammen:

$$f = abcd + a\overline{b}cd + \overline{a}bcd + abc\overline{d} + \overline{a}\,\overline{b}cd + \overline{a}bc\overline{d}$$

1. Stelle Minterme in einer Tabelle als binäre Belegungen dar. Nummeriere die Minterme gemäß der Belegungen dezimal.

Minterm	Dezimal	Binär $x_3x_2x_1x_0$	2 Terme zusammenfassen	Belegung 2 Terme	4 Terme zusammen-fassen (Runde 2)	Belegung 4 Terme	Evtl. weitere Gruppen zusammen-fassen	Prim-terme
$abcd$	15	1 1 1 1						
$a\overline{b}cd$	11	1 0 1 1					....	
$\overline{a}bcd$	7	0 1 1 1						
$abc\overline{d}$	14	1 1 1 0						
$\overline{a}\overline{b}cd$	3	0 0 1 1						
$\overline{a}bc\overline{d}$	6	0 1 1 0						

84

Beispiel zu McCluskey (2):

2. Fasse jeweils 2 Minterme zu einer Gruppe zusammen, wenn sich nur eine Stelle der Belegung ändert (Bedingung eines Gray Codes). Hake die bereits verwendeten Minterme ab. Wenn nicht zusammenfassbare Minterme übrig bleiben, sind sie Teil der Primterme. Markiere die eliminierten Variablen in der Belegung durch ein Leerzeichen „--“

Minterm	**Dezimal**	**Binär** $x_3x_2x_1x_0$	**2 Terme** zusammenfassen	**Belegung** 2 Terme (Runde 1)	**4 Terme** zusammen-fassen (Runde 2)	**Belegung** 4 Terme	Evtl. weitere Gruppen zusammen-fassen	**Prim-terme**
$abcd$	15✓	1 1 1 1	15,11	1 -- 1 1				
$a\bar{b}cd$	11✓	1 0 1 1	15,7	-- 1 1 1			....	
$\bar{a}bcd$	7✓	0 1 1 1	15,14	1 1 1 --				
$abc\bar{d}$	14✓	1 1 1 0	11,3	-- 0 1 1				
$\bar{a}\bar{b}cd$	3 ✓	0 0 1 1	7,3	0 -- 1 1				
$\bar{a}bc\bar{d}$	6 ✓	0 1 1 0	7,6	0 1 1 --				
			14,6	-- 1 1 0				

85

Beispiel zu McCluskey (3):

3. Wiederhole die Zusammenfassung der Gruppen wie in 2. zu größeren Einheiten (kürzere Terme) ggf. mehrfach, bis keine Zusammenfassung mehr möglich ist. Teste nur die Terme, bei denen die Stellen mit – übereinstimmen.
Die übrig bleibenden Terme ohne Häkchen sind die Primterme.

Minterm	**Dezimal**	**Binär** $x_3x_2x_1x_0$	**2 Terme** zusammenfassen	**Belegung** 2 Terme (Runde 1)	**4 Terme** zusammen-fassen (Runde 2)	**Belegung** 4 Terme	Evtl. weitere Gruppen zusammen-fassen	**Prim-terme**
$abcd$	15 ✓	1 1 1 1	15,11 ✓	1 -- 1 1	(15,11), (7,3)	-- -- 1 1		
$a\bar{b}cd$	11 ✓	1 0 1 1	15,7 ✓	-- 1 1 1	(15,7), (11,3)	-- -- 1 1	....	
$\bar{a}bcd$	7 ✓	0 1 1 1	15,14 ✓	1 1 1 --	(15,7), (14,6)	-- 1 1 --		
$abc\bar{d}$	14 ✓	1 1 1 0	11,3 ✓	-- 0 1 1	(15,14), (7,6)	-- 1 1 --		
$\bar{a}\bar{b}cd$	3 ✓	0 0 1 1	7,3 ✓	0 -- 1 1				
$\bar{a}bc\bar{d}$	6 ✓	0 1 1 0	7,6 ✓	0 1 1 --				
			14,6 ✓	-- 1 1 0				

86

Beispiel zu McCluskey (4):

Der Algorithmus terminiert jetzt, da keine weiteren Gruppen mehr zu Teiltermen zusammengefasst werden können.
Die übrig bleibenden Terme ohne Häkchen sind die Primterme. Die Disjunktion der Primterme ergibt eine Lösung, die aber nicht notwendigerweise bereits minimal ist. Es können noch Doubletten und redundante Primterme vorhanden sein. Eliminiere Doubletten mit Satz 3 und redundante Primterme mit Quine-Verfahren.

Minterm	**Dezimal**	**Binär** $x_3x_2x_1x_0$	**2 Terme** zusammenfassen	**Belegung** 2 Terme (Runde 1)	**4 Terme** zusammen-fassen (Runde 2)	**Belegung** 4 Terme	Alle Terme wurden zusammen-gefassst,	**Prim-terme**
$abcd$	15 ✓	1 1 1 1	15,11 ✓	1 -- 1 1	(15,11), (7,3)	-- -- 1 1		cd
$a\bar{b}cd$	11 ✓	1 0 1 1	15,7 ✓	-- 1 1 1	(15,7), (11,3)	-- -- 1 1	Jetzt Primterme ausgeben	$\cancel{cd}$
$\bar{a}bcd$	7 ✓	0 1 1 1	15,14 ✓	1 1 1 --	(15,7), (14,6)	-- 1 1 --		bc
$abc\bar{d}$	14 ✓	1 1 1 0	11,3 ✓	-- 0 1 1	(15,14), (7,6)	-- 1 1 --	Satz 3	$\cancel{bc}$
$\bar{a}\bar{b}cd$	3 ✓	0 0 1 1	7,3 ✓	0 -- 1 1				
$\bar{a}bc\bar{d}$	6 ✓	0 1 1 0	7,6 ✓	0 1 1 --				
			14,6 ✓	-- 1 1 0				

$$f = bc + cd$$

Zweites Beispiel zu Quine-McCluskey:

1. Berechne Primterme von f nach McCluskey

$$f = a\bar{b}\,\bar{c}\,d + ab\bar{c}d + abcd + \bar{a}bcd + \bar{a}bc\bar{d}$$

Minterm	**Dezimal**	**Binär** $x_3x_2x_1x_0$	**2 Terme** zusammenfassen	**Belegung** 2 Terme (Runde 1)	**4 Terme** Keine weitere Zusammenfassung	**Primterme**
$a\bar{b}\bar{c}d$	9 ✓	1 0 0 1	9,13	1 -- 0 1		$a\bar{c}d$
$ab\bar{c}d$	13 ✓	1 1 0 1	13,15	1 1 -- 1		abd
$abcd$	15 ✓	1 1 1 1	15,7	-- 1 1 1		bcd
$\bar{a}bcd$	7 ✓	0 1 1 1	7,6	0 1 1 --		$\bar{a}bc$
$\bar{a}bc\bar{d}$	6 ✓	0 1 1 0				

$$f = a\bar{c}d + abd + bcd + \bar{a}bc$$

Ist diese Form minimal oder enthält sie redundante Terme?
Test nach Quine!

II) Das Verfahren von Quine

Eine Primterm-Minterm-Tabelle wird aufgestellt: Die Minterme in der Zeile und die Primterme in der Spalte.

1. Alle Spalten, in denen eine 1 aus einer dominanten Zeile (Zeile mit nur einer 1) steht, werden markiert. Alle Zeilen, in denen 1 aus markierten Spalten stehen, werden gestrichen.
2. Wenn keine ungestrichene Zeile mehr vorhanden ist, wird das Verfahren beendet. Die markierten Spalten bilden die Minterme der DMF.
3. Wenn keine dominante Zeile mehr vorhanden ist, aber noch ungestrichene Zeilen existieren, wird eine Spalte mit den meisten ungestrichenen 1 markiert (wahlweise obligatorischer Primterm) und bei 1. fortgefahren.

Definitionen: Ein dominanter Minterm ist ein Term, der nur in einem einzigen Primterm enthalten ist. Der dazu gehörige Primterm heißt wesentlicher Primterm. Ein redundanter Primterm ist ein Term, der übrig bleibt, wenn alle Minterme schon in wesentlichen Primtermen enthalten sind. Er wird in der Minimalform nicht benötigt. Wenn eine Auswahl aus mehreren Primtermen besteht, dann heißen diese wahlweise obligatorische Primterme.

Quine: Finde dominante Minterme und wesentliche, wahlweise obligatorische, sowie redundante Primterme

$$f = a\bar{c}d + abd + bcd + \bar{a}bc$$

Eine Primterm-Minterm-Tabelle wird aufgestellt: Die Minterme (Nummer) in der Zeile und die Primterme (Nummern aller zusammengefassten Minterme) in der Spalte.

1. Markiere alle Minterme für jeden Primterm in der Tabelle mit 1. Im Primterm-Eintrag sind die Nummern aller zugehörigen Minterme aufgeführt.

	$a\bar{c}d$	abd	bcd	$\bar{a}bc$
Primterme:	**(9,13)**	**(13,15)**	**(15,7)**	**(7,6)**
Minterme:				
9	1			
13	1	1		
15		1	1	
7			1	1
6				1

Test nach Quine (2):

2. Wenn ein Minterm nur von einem Primterm abgedeckt wird (nur eine 1 in der Zeile), dann ist der Primterm wesentlich und der Minterm dominant. Markiere die wesentlichen Primterme (Häkchen und gestrichelte blaue Gerade) und streiche (blau) die Zeilen aller dominanten Minterme.

	$a\bar{c}d$	abd	bcd	$\bar{a}bc$
Primterme:	**(9,13)**	**(13,15)**	**(15,7)**	**(7,6)**
Minterme:	✓			✓
~~**9**~~	~~1~~			
13	1	1		
15		1	1	
7			1	1
~~**6**~~				~~1~~

91

Test nach Quine (3):

2. Streiche (rot) ebenfalls alle anderen Mintermzeilen, die bereits von den wesentlichen Primtermen abgedeckt werden.

Verfahre so weiter bis keine dominanten Minterme mehr auftreten.
In diesem Beispiel bleiben zwei Primterme und ein Minterm unmarkiert.
Diese Primterme sind wahlweise obligatorisch.

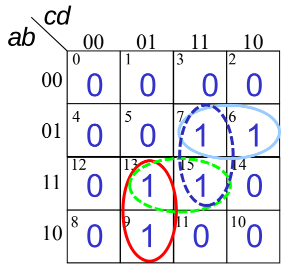
	$a\bar{c}d$	abd	bcd	$\bar{a}bc$
Primterme:	**(9,13)**	**(13,15)**	**(15,7)**	**(7,6)**
Minterme:	✓			✓
~~**9**~~	~~1~~			
~~**13**~~	~~1~~	~~1~~		
15		1	1	
~~**7**~~			~~1~~	~~1~~
~~**6**~~				~~1~~

92

Test nach Quine (4):

3. Falls noch Zeilen und Spalten übrig bleiben, dann kann einer der Primterme ausgewählt werden, der die meisten Minterme beinhaltet. Hier wahlweise obligatorisch (13,15) oder (15,7) wählen, beide decken Minterm 15 ab. Es gibt zwei wahlweise obligatorische gleichwertige Lösungen. Je einer der beiden Primterme (13,15) oder (15,7) ist redundant.

	$a\bar{c}d$	abd	bcd	$\bar{a}bc$
Primterme:	**(9,13)**	**(13,15)**	**(15,7)**	**(7,6)**
Minterme:	✓	✓	✓	✓
~~9~~	1			
~~13~~	1	1		
~~15~~		1	1	
~~7~~			1	1
~~6~~				1



$$f = a\bar{c}d + bcd + \bar{a}bc$$

oder

$$f = a\bar{c}d + abd + \bar{a}bc$$

Drittes Beispiel zu Quine-McCluskey (redundanter Term):

1. Berechne Primterme von f nach McCluskey

$$f = ab\bar{c} + abc + \bar{a}bc + \bar{a}\bar{b}c$$

Minterm	**Dezimal**	**Binär** $x_2x_1x_0$	**2 Terme** zusammen-fassen	**Belegung** 2 Terme (Runde 1)	**4 Terme** Keine weitere Zusammenfassung	**Primterme**
$ab\bar{c}$	6 ✓	1 1 0	6,7	1 1 --		ab
abc	7 ✓	1 1 1	7,3	-- 1 1		bc
$\bar{a}bc$	3 ✓	0 1 1	3,1	0 – 1		$\bar{a}c$
$\bar{a}\bar{b}c$	1 ✓	0 0 1				

$$f = ab + bc + \bar{a}c$$

Ist diese Form minimal oder enthält sie redundante Terme? Test nach Quine!

Test nach Quine (redundanter Term): $f = ab + bc + \bar{a}c$

Minterme 6 und 1 sind dominant und werden durch die Primterme (6,7) und (3,1) abgedeckt. Diese zwei Primterme sind wesentlich und decken alle Minterme ab. Ein Primterm (7,3) ist redundant und fällt weg (ist nicht markiert).

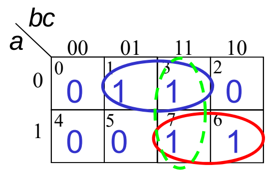
	ab	bc	$\bar{a}c$
Primterme:	**(6,7)**	**(7,3)**	**(3,1)**
Minterme:	✓		✓
6	1		
7	1	1	
3		1	1
1			1

95

Test nach Quine (redundanter Term): $f = ab + bc + \bar{a}c$

Minterme 6 und 1 sind dominant und werden durch die Primterme (6,7) und (3,1) abgedeckt. Diese zwei Primterme sind wesentlich und decken alle Minterme ab. Ein Primterm (7,3) ist redundant und fällt weg (ist nicht markiert).

	ab	bc	$\bar{a}c$
Primterme:	**(6,7)**	**(7,3)**	**(3,1)**
Minterme:	✓		✓
6	1		
7	1	1	
3		1	1
1			1



DMF: $f = ab + \bar{a}c$

96