Digitaltechnik

Andreas König
Professur Technische Informatik
Fakultät Informatik
Technische Universität Chemnitz

Wintersemester 2001/2002

Rekapitulierung zu Kapitel 3

- **Einführung in die Boolesche Algebra**
- **Darstellung von Schaltfunktionen:**
  - **Funktionstabellen**
  - **KV-Diagramme**
  - **Algebraische Darstellung**
  - **Gatterdarstellung**
  - **Würfel**
  - **BDDs und OBDDs**
- **Regelwerk zur algebraischen Umformung**
- **Vorstellung verschiedener Basissysteme**
- **Rüstzeug zur eindeutigen Konstruktion und Darstellung einer beliebigen Schaltfunktion (Bündelfunktionen)**
- **Vollständige bzw. unvollständige Schaltfunktionen**
- **Entwurf von Schaltfunktionen mit negativer und positiver Logik**

Vorlesungsgliederung:

1. **Einführung**
2. **Kodierung und Arithmetik**
3. **Grundlagen der Booleschen Algebra**
4. **Entwurf zweistufiger kombinatorischer Logik**
5. **Zieltechnologien und Technologieanpassung**
6. **Zeitliches Verhalten kombinatorischer Schaltnetze**
7. **Entwurf sequentieller Schaltwerke**
8. **Funktionsblöcke digitaler Rechner und Systeme**
9. **Entwurf von Systemen der Digitaltechnik**
10. **Ausblick**

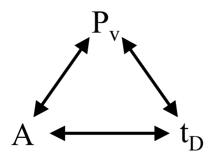
Kapitelgliederung:

4. Entwurf zweistufiger kombinatorischer Logik

4.1 Aufgabenstellung
4.2 Begriffsdefinitionen
4.3 Graphische Minimierung im KV-Diagramm
4.4 Tabellarische Minimierung (Quine/McCluskey-Verfahren)
4.5 Überdeckungsproblem
4.6 Funktionsbündel
4.7 Exakte und heuristische Lösung
4.8 Synthesewerkzeuge

Aufgabenstellung

- Der Entwurf von Systemen der Digitaltechnik unterliegt einer Reihe harter wirtschaftlicher und technischer Randbedingungen, wie z.B. den Kosten, der Entwicklungszeit, der Geschwindigkeit, oder dem Leistungsverbrauch P_v bei **garantierter Systemfunktionalität**
- Die Anzahl benötigter Logikbausteine beeinflusst auf PCB-Ebene (**P**rinted **C**ircuit **B**oard) die Kosten
- Bei einer mikroelektronischen Lösung bestimmt die Gatterzahl und die benötigte Gattertiefe den Chipflächenbedarf A und damit die Kosten
- Art der Gatter und die Tiefe (Stufenzahl) bestimmen Geschwindigkeit (t_D)
- Insgesamt Optimierungsdreieck im Digitalentwurf:



- Beispiel einer Mehrzieloptimierung, Trade-off für jeweilige Lösung

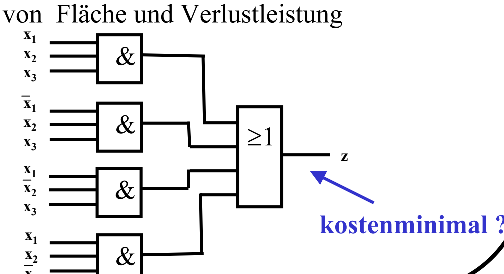
Aufgabenstellung

- Die Betrachtung und Optimierung dieses Gesamtaspekts ist äußerst komplex
- Im folgenden erfolgt die Beschränkung auf die Optimierung kombinatorischer zweistufiger Logik
- **Aufgabenstellung:** Zweistufige Logikminimierung
  - Gegeben ist eine beliebige Boolesche Funktion (Schaltfunktion)
  - Gesucht ist ein Schaltwerk in Form einer zweistufigen Und/Oder-Realisierung mit minimalen Kosten !
- Erkennbar steckt in der Forderung minimaler Kosten ein Bezug zum Realisierungsaufwand in Form von Fläche und Verlustleistung

x_1

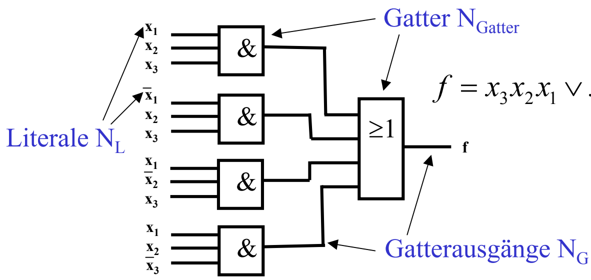
	0	0	1	0
x_2	0	1	1	1

x_3



Aufgabenstellung

- Ansatz einer Kostenfunktion:



$$f = x_3x_2x_1 \vee x_3x_2\overline{x_1} \vee x_3\overline{x_2}x_1 \vee \overline{x_3}x_2x_1$$

- Anzahl der Literale als Kostenfunktion: $N_E = N_L = 12$
- Anzahl der Literale und Produkterme als Kostenfunktion: $N_E = N_L + N_G - 1 = 12 + (5-1) = 16$
- Anzahl der Produkterme als Kostenfunktion: $N_E = N_G - 1 = 4$
- Ziel der Logikminimierung: $\mathbf{N_E(f) \overset{!}{=} Minimum}$

Aufgabenstellung

- Erste Überlegungen haben gezeigt, dass algebraische Zusammenfassungen bzw. graphische Zusammenfassungen im KV-Diagramm bereits Möglichkeiten in Richtung gewünschter Minimierung bieten

x_2x_1 \ x_4x_3	00	01	11	10
00	1	0	0	1
01	0	1	0	0
11	1	1	1	1
10	1	1	1	1

x_3, x_1, x_2, x_4

$$f = \bigvee_{j \in \{0,2,3,5,6,7,8,10,11,14,15\}} (m_j)$$

DNF: $N_E = N_L + N_G - 1 = 44 + (12-1) = 55$

- Visuelle Zusammenfassungen im KV-Diagramm durch Verschmelzung benachbarter Felder (Feldwechsel entspricht Änderung einer Variablen !)

Aufgabenstellung

➢ Die blau eingezeichnete erste Zusammenfassung bezieht sich auf:

$$\overline{x}_4\overline{x}_3x_2x_1 \vee \overline{x}_4x_3x_2x_1 = \overline{x}_4x_2x_1(\overline{x}_3 \vee x_3) = \overline{x}_4x_2x_1$$

x_2x_1 \ x_4x_3	00	01	11	10
00	1	0	0	1
01	0	1	0	0
11	1	1	1	1
10	1	1	1	1

$$f = \bigvee_{j\in\{0,2,3,5,6,7,8,10,11,14,15\}} (m_j)$$

DNF: $N_E = N_L + N_G - 1 = 44 + (12-1) = 55$

➢ Die Verschmelzung lässt sich in Zweierpotenzen zu immer größeren Belegungsblöcken sinkender Literalzahl fortsetzen

Aufgabenstellung

➢ Die lila eingezeichnete zweite Zusammenfassung ergibt nun:

$$\boxed{x_2x_1}$$

x_2x_1 \ x_4x_3	00	01	11	10
00	1	0	0	1
01	0	1	0	0
11	1	1	1	1
10	1	1	1	1

$$f = \bigvee_{j\in\{0,2,3,5,6,7,8,10,11,14,15\}} (m_j)$$

DNF: $N_E = N_L + N_G - 1 = 44 + (12-1) = 55$

➢ Eine weitere, grün eingezeichnete Zusammenfassung ergibt nun: $\boxed{x_2}$

Aufgabenstellung

- Der noch nicht verwendete Minterm 5 kann nun ebenfalls mit einem (bereits verwendeten !) Nachbarfeld verschmolzen werden:

$\boxed{\overline{x_4}x_3x_1}$

	x_2x_1 \ x_4x_3	00	01 (x_3)	11 (x_3, x_4)	10 (x_4)	
	00	1	0	0	1	
x_1	01	0	1	0	0	
x_1	11	1	1	1	1	x_2
	10	1	1	1	1	x_2

$$f = \bigvee_{j \in \{0,2,3,5,6,7,8,10,11,14,15\}} (m_j)$$

DNF: $N_E = N_L + N_G - 1 = 44+(12-1)=55$

- Nun bleiben noch die Minterme 0 und 8. Diese unterscheiden sich nur in der Variablen x_4 und können somit auch zusammengefasst werden !

Aufgabenstellung

- Damit ergibt sich zunächst: $\boxed{\overline{x_3}\,\overline{x_2}\,\overline{x_1}}$

	x_2x_1 \ x_4x_3	00	01 (x_3)	11 (x_3, x_4)	10 (x_4)	
	00	1	0	0	1	
x_1	01	0	1	0	0	
x_1	11	1	1	1	1	x_2
	10	1	1	1	1	x_2

$$f = \bigvee_{j \in \{0,2,3,5,6,7,8,10,11,14,15\}} (m_j)$$

DNF: $N_E = N_L + N_G - 1 = 44+(12-1)=55$

- Jedoch können aus Symmetriegründen die verbleibenden zwei Ecken, die auch zur Einstellenmenge gehören, mit einbezogen werden !

Aufgabenstellung

- Damit ergibt sich insgesamt: $\boxed{\overline{x_3}\,\overline{x_1}}$

$x_2x_1 \backslash x_4x_3$	00	01	11	10
00	1	0	0	1
01	0	1	0	0
11	1	1	1	1
10	1	1	1	1

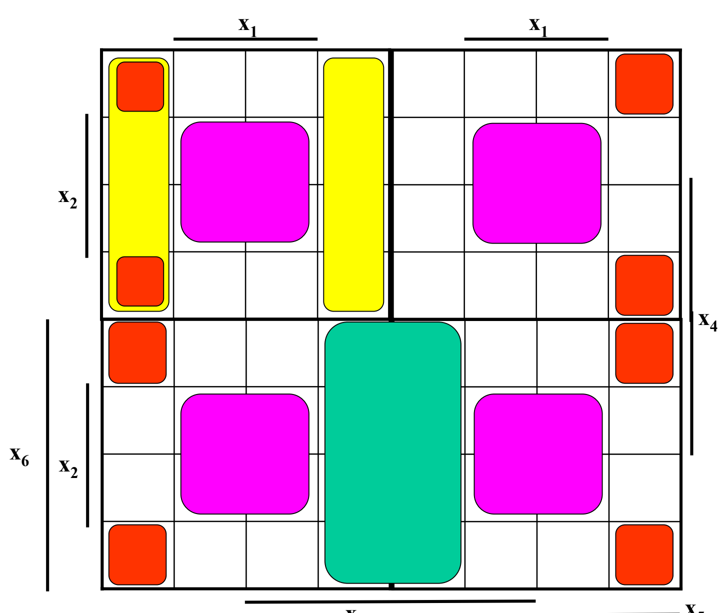
x_3, x_1, x_2, x_4

$$f = \overline{x_3}\,\overline{x_1} \vee x_2 \vee \overline{x_4}x_3x_1$$

$$N_E = N_L + N_G - 1 = 6+(3-1)=8$$

- Die Betrachtung kann aus Gründen der Dualität auch für die KNF und Maxterme äquivalent vorgenommen werden !

Aufgabenstellung



x_2x_1

$x_6x_3\overline{x_1}$

$\overline{x_6}\,\overline{x_5}\,\overline{x_1}$

$\overline{x_3}\,\overline{x_2}\,\overline{x_1}$

Aufgabenstellung

- Behandlung von Don´t-care-Feldern unvollständig spezifizierter Funktionen bei der Zusammenfassung

x_2x_1 \ x_4x_3	00	01	11	10
00	0	0	d	0
01	1	1	d	1
11	1	1	0	0
10	0	d	0	0

$\boxed{\overline{x_4}x_1}$

$\boxed{\overline{x_2}x_1}$

- Die d-Felder so Nullstellen- bzw. Einsstellenmenge zuordnen, dass sich maximale Belegungsblöcke ergeben

Aufgabenstellung

- Entsprechende Zusammenfassung von Maxtermen:

x_2x_1 \ x_4x_3	00	01	11	10
00	0	0	d	0
01	1	1	d	1
11	1	1	0	0
10	0	d	0	0

$\boxed{x_1}$

$\boxed{\overline{x_4} \vee \overline{x_2}}$

- Die bisherigen Zusammenfassung führen zu Verbesserungen
- Um das wirkliche (globale) Minimum zu finden, muss eine systematischere Herangehensweise herangezogen werden

Begriffsdefinitionen

Definition 4.1: Ein **Implikant** einer Funktion f ist ein Produktterm c für den gilt c≤ f, d.h. der Implikant überdeckt mindestens einen oder maximal alle Einsstellen der Funktion f. Ein Minterm ist damit ein Implikant.

x_2x_1 \ x_4x_3	00	01	11	10
00	1	1	0	0
01	0	0	0	1
11	0	1	1	0
10	0	1	1	0

Kein Implikant

Ein Implikant: $x_4\overline{x}_3\overline{x}_2x_1$

Ein Implikant: $\overline{x}_4x_3x_2$

Begriffsdefinitionen

Definition 4.2: Ein **Primimplikant** einer Funktion f ist ein Produktterm p für den gilt p≤ f, d.h. der selbst ein Implikant ist, und für den kein anderer Implikant c von f existiert für den p≤c gilt. D.h., ein Primimplikant wird von keinem anderen Implikanten überdeckt und besitzt damit mindestens eine Einsstelle mehr als die existierenden Implikanten von f.

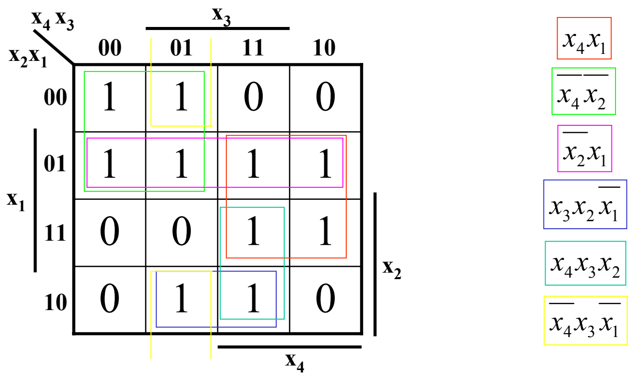
x_2x_1 \ x_4x_3	00	01	11	10
00	1	1	0	0
01	0	0	0	1
11	0	1	1	0
10	0	1	1	0

Kein Primimplikant: $x_4x_3x_2$

Primimplikant: x_3x_2

Begriffsdefinitionen

- Beispiel zur Auffindung aller Primimplikanten einer gegebenen Funktion:

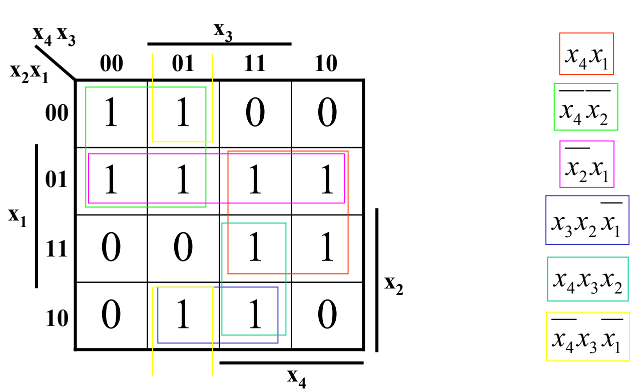


- Alle im KV-Diagramm hervorgehobenen Implikanten sind Primimplikanten von f
- Im KV-Diagramm ist die Menge aller Primimplikanten hervorgehoben

Begriffsdefinitionen

- Die Disjunktion aller Primimplikanten heißt Blakesche Normalform
- Sie ist ebenfalls eine kanonische Normalform:

$$f = x_4 x_1 \vee \overline{x_4}\,\overline{x_2} \vee \overline{x_2}\,x_1 \vee x_3 x_2 \overline{x_1} \vee x_4 x_3 x_2 \vee \overline{x_4}\,x_3\,\overline{x_1}$$



➢ Erkennbar werden nicht alle Primimplikanten zur Funktionsdarstellung benötigt !

$$f = x_4 x_1 \vee \overline{x_4}\,\overline{x_2} \vee x_3 x_2 \overline{x_1}$$

x_2x_1 \ x_4x_3	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	1	1
10	0	1	1	0

x_3 (Spalten 01, 11), x_4 (Spalten 11, 10), x_1 (Zeilen 01, 11), x_2 (Zeilen 11, 10)

$\boxed{x_4 x_1}$

$\boxed{\overline{x_4}\,\overline{x_2}}$

$\boxed{x_3 x_2 \overline{x_1}}$

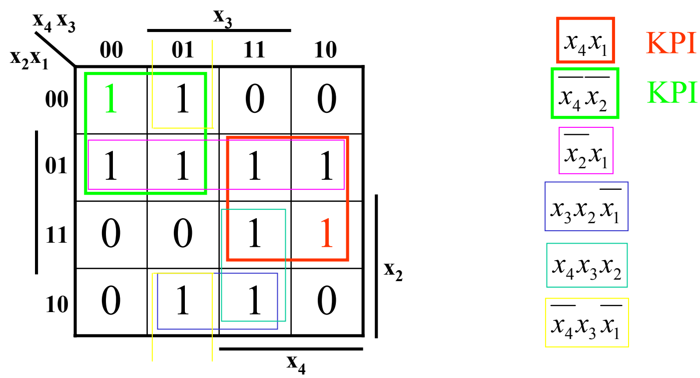
Satz 4.1 Sei P die Menge aller Primimplikanten der Funktion f. Dann setzt sich die Disjunktive Minimalform DMF aus einer Disjunktion von Primimplikanten P* zusammmen für die gilt $P^* \subseteq P$. Anschaulich ließe sich ja jeder Implikant durch einen Primimplikant mit gleichen oder geringeren Kosten ersetzen und dieser würde daher keinen Eingang in die DMF finden.

➢ Welche der Primimplikanten aus der Blakeschen Normalform werden zur Bildung der DMF benötigt ?

➢ Nach welchen Kriterien kann die Auswahl verschiedener Typen von Primimplikanten zur Minimierung der Kostenfunktion erfolgen ?

Begriffsdefinitionen

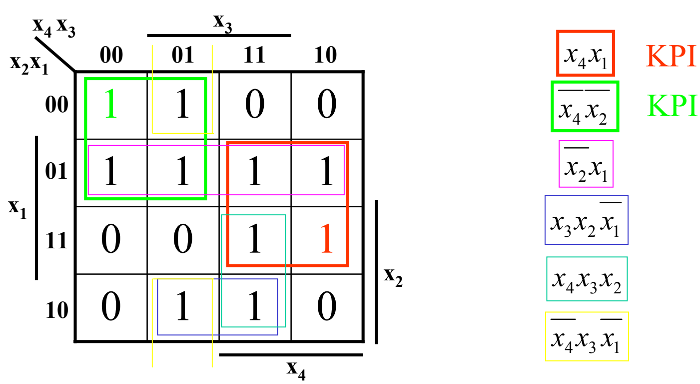
Definition 4.3: Ein Primimplikant p einer Funktion f heißt **Kernprimimplikant** (KPI) wenn er von der Disjunktion aller anderen Primimplikanten der Menge P´=P \ p nicht überdeckt wird. Anschaulich heißt dies, ein KPI oder ein essentieller Primimplikant besitzt mindestens ein Einselement (Minterm) von f, das (der) auch durch die Zusammenfassung aller anderen Primimplikanten sonst nicht dargestellt wird.



Begriffsdefinitionen

- Die KPI werden also unabdingbar zur Bildung der DMF von benötigt
- Manche Minterme werden von mehreren Primimplikanten überdeckt !

$$f = \boxed{x_4 x_1} \vee \boxed{\bar{x}_4 \bar{x}_2} \vee \bar{x}_2 x_1 \vee x_3 x_2 \bar{x}_1 \vee x_4 x_3 x_2 \vee \bar{x}_4 x_3 \bar{x}_1$$



Begriffsdefinitionen

Definition 4.4: Ein Primimplikant p einer Funktion f heißt a**bsolut eliminierbarer Primimplikant** (API) wenn er von der Disjunktion aller Kernprimimplikanten der Menge P_K überdeckt wird. Anschaulich heißt dies, ein API besitzt kein Einselement (keinen Minterm) von f, das (der) nicht bereits durch einen der Kernprimimplikanten überdeckt bzw. dargestellt wird.

x_2x_1 \ x_4x_3	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	1	1
10	0	1	1	0

x_3 x_2 x_4

x_4x_1	KPI
$\overline{x_4}\,\overline{x_2}$	KPI
$\overline{x_2}x_1$	API
$x_3x_2\overline{x_1}$	
$x_4x_3x_2$	
$\overline{x_4}x_3\overline{x_1}$	

Begriffsdefinitionen

- Die KPI werden also unabdingbar zur Bildung der DMF von benötigt
- Die API sind redundant und können entfernt werden

$$f = \boxed{x_4x_1} \vee \boxed{\overline{x_4}\,\overline{x_2}} \vee \cancel{\overline{x_2}}x_1 \vee x_3x_2\overline{x_1} \vee x_4x_3x_2 \vee \overline{x_4}x_3\overline{x_1}$$

x_2x_1 \ x_4x_3	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	1	1
10	0	1	1	0

x_3 x_1 x_2 x_4

x_4x_1	KPI
$\overline{x_4}\,\overline{x_2}$	KPI
$\overline{x_2}x_1$	API
$x_3x_2\overline{x_1}$	?
$x_4x_3x_2$	?
$\overline{x_4}x_3\overline{x_1}$	?

Definition 4.5: Ein Primimplikant p einer Funktion f heißt **relativ eliminierbarer Primimplikant** (RPI) wenn er von der Disjunktion aller Primimplikanten der Menge P \ P_A überdeckt wird. Anschaulich heißt dies, ein RPI besitzt mindestens ein Einselement (einen Minterm) von f, das (der) nicht durch einen der Kernprimimplikanten überdeckt bzw. dargestellt wird aber von mehreren RPI überdeckt wird.

x_2x_1 \ x_4x_3	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	1	1
10	0	1	1	0

x_3: Spalten 01, 11; x_4: Spalten 11, 10; x_1: Zeilen 01, 11; x_2: Zeilen 11, 10

$x_4 x_1$	KPI
$\overline{x_4}\,\overline{x_2}$	KPI
$\overline{x_2} x_1$	API
$x_3 x_2 \overline{x_1}$	RPI
$x_4 x_3 x_2$	RPI
$\overline{x_4} x_3 \overline{x_1}$	RPI

Satz 4.2 Sei P die Menge aller Primimplikanten der Funktion f. Dann setzt sich die Disjunktive Minimalform DMF aus einer Disjunktion aller Kernprimimplikanten KPI aus P_K und einer geeigneten Auswahl relativ eliminierbarer Primimplikanten RPI aus P_R zusammen, so dass kein Primimplikant p dieser Disjunktion von der Menge aller Primimplikanten P*\p überdeckt wird. Eine Normalform, für die kein Produktterm weggelassen werden kann, ohne dass die Funktion verändert wird, heißt irredundant.

- Erfüllen mehrere DNF die Bedingung des Satzes 4.2 (irredundante DNF) so ist die DNF mit geringsten Kosten als DMF auszuwählen

Begriffsdefinitionen

- Beispiel der RPI-Auswahl zur Bildung einer irredundanten NF:

$$f = x_4 x_1 \vee \overline{x_4}\,\overline{x_2} \vee \cancel{\overline{x_2} x_1} \vee g_i$$

$$g_1 = x_3 x_2 \overline{x_1} \vee x_4 x_3 x_2 \vee \overline{x_4} x_3 \overline{x_1}$$

$$g_2 = x_3 x_2 \overline{x_1} \vee \cancel{x_4 x_3 x_2} \vee \cancel{\overline{x_4} x_3 \overline{x_1}}$$

$$g_3 = \cancel{x_3 x_2 \overline{x_1}} \vee x_4 x_3 x_2 \vee \overline{x_4} x_3 \overline{x_1}$$

$$g_4 = x_3 x_2 \overline{x_1} \vee x_4 x_3 x_2 \vee \cancel{\overline{x_4} x_3 \overline{x_1}}$$

$$g_5 = x_3 x_2 \overline{x_1} \vee \cancel{x_4 x_3 x_2} \vee \overline{x_4} x_3 \overline{x_1}$$

$x_2 x_1$ \ $x_4 x_3$	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	1	1
10	0	1	1	0

Begriffsdefinitionen

- Erkennbar sind die beiden Varianten mit g1 und g2 irredundant

$$f^1 = x_4 x_1 \vee \overline{x_4}\,\overline{x_2} \vee x_3 x_2 \overline{x_1}$$

$$f^2 = x_4 x_1 \vee \overline{x_4}\,\overline{x_2} \vee x_4 x_3 x_2 \vee \overline{x_4} x_3 \overline{x_1}$$

- Die Kosten für f^1 sind $N_E=10$
- Die Kosten für f^2 sind $N_E=14$
- Die kostengünstigere irredundante NF wurde ins KV-Diagramm eingetragen:
- DMF ist durch f^1 gegeben

$x_2 x_1$ \ $x_4 x_3$	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	1	1
10	0	1	1	0

Begriffsdefinitionen

- Vorgehensweise bei unvollständig spezifizierten Funktionen:
  - Zuschlagen der d-Felder zur Einstellen- bzw. Nullstellenmenge
  - **Aber:** Ein Produktterm ist nur PI, wenn er mindestens eine 1 enthält
  - Entsprechendes gilt für KPI/API
  - Beispiel:

KPI: $\boxed{\overline{x_4}\,\overline{x_3}}$

API: $\boxed{\overline{x_4} x_1}$

kein PI: $\boxed{\overline{x_2} x_1}$

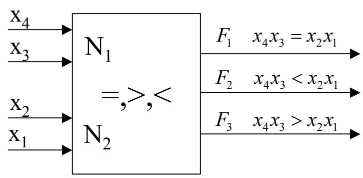
x_2x_1 \ x_4x_3	00	01	11	10
00	1	0	0	0
01	d	d	d	d
11	1	d	0	0
10	d	0	0	0

x_3 (Spalten 01, 11), x_4 (Spalten 11, 10), x_1 (Zeilen 01, 11), x_2 (Zeilen 11, 10)

Graphische Minimierung im KV-Diagramm

- Die Durchführung zweistufiger Logiksynthese beruht auf der Auffindung und geeigneten Auswahl aller Primimplikanten zur aufwandsgünstigsten Realisierung der darzustellenden Funktion f
- **Lösungsmöglichkeit 1:** Darstellung der Funktion f im KV-Diagramm und visuelles, interaktives bestimmen der KPI, API, RPI und einer darauf basierenden minimalen Lösung
- Erkennbare Limitierung durch Variablenzahl ($n \leq 6$) und Übersichtlichkeit bei großer Zahl von Primimplikanten
- Dieser Ansatz wird im folgenden anhand eines Beispiels dargestellt
- **Lösungsmöglichkeit 2:** Tabellarische Verfahren zur Bestimmung aller Primimplikanten (ggf. rekursiv), gefolgt von der Lösung eines sogenannten Überdeckungsproblems
- Bessere rechnergestützte Umsetzung dieser exakten Herangehensweise
- Derartige Ansätze werden z.T. nachfolgenden dargestellt und angewandt

- Anwendungsbeispiel 2 bit Komparator:



$x_4x_3x_2x_1$	$F_1F_2F_3$
0 0 0 0	1 0 0
0 0 0 1	0 1 0
0 0 1 0	0 1 0
0 0 1 1	0 1 0
0 1 0 0	0 0 1
0 1 0 1	1 0 0
0 1 1 0	0 1 0
0 1 1 1	0 1 0
1 0 0 0	0 0 1
1 0 0 1	0 0 1
1 0 1 0	1 0 0
1 0 1 1	0 1 0
1 1 0 0	0 0 1
1 1 0 1	0 0 1
1 1 1 0	0 0 1
1 1 1 1	1 0 0

- Entwurf der Teilfunktion F_1:

x_2x_1 \ x_4x_3	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

$x_4x_3x_2x_1$	$F_1F_2F_3$
0 0 0 0	1 0 0
0 0 0 1	0 1 0
0 0 1 0	0 1 0
0 0 1 1	0 1 0
0 1 0 0	0 0 1
0 1 0 1	1 0 0
0 1 1 0	0 1 0
0 1 1 1	0 1 0
1 0 0 0	0 0 1
1 0 0 1	0 0 1
1 0 1 0	1 0 0
1 0 1 1	0 1 0
1 1 0 0	0 0 1
1 1 0 1	0 0 1
1 1 1 0	0 0 1
1 1 1 1	1 0 0

$$\boxed{F_1 = x_4x_3x_2x_1 \vee x_4\overline{x_3}x_2\overline{x_1} \vee \overline{x_4}x_3\overline{x_2}x_1 \vee \overline{x_4}\,\overline{x_3}\,\overline{x_2}\,\overline{x_1}}$$

- Nur Minterme die damit gleichzeitig KPI sind

- Entwurf der Teilfunktion F_2:

x_4x_3 / x_2x_1	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

x_3 (columns 01, 11); x_4 (columns 11, 10); x_1 (rows 01, 11); x_2 (rows 11, 10)

$$\boxed{F_2 = \overline{x}_4 x_2 \vee \overline{x}_4\,\overline{x}_3 x_1 \vee \overline{x}_3\,\overline{x}_2 x_1}$$

$x_4x_3x_2x_1$	$F_1\ F_2\ F_3$
0 0 0 0	1 0 0
0 0 0 1	0 1 0
0 0 1 0	0 1 0
0 0 1 1	0 1 0
0 1 0 0	0 0 1
0 1 0 1	1 0 0
0 1 1 0	0 1 0
0 1 1 1	0 1 0
1 0 0 0	0 0 1
1 0 0 1	0 0 1
1 0 1 0	1 0 0
1 0 1 1	0 1 0
1 1 0 0	0 0 1
1 1 0 1	0 0 1
1 1 1 0	0 0 1
1 1 1 1	1 0 0

- Drei KPI, keine API oder RPI

- Entwurf der Teilfunktion F_3:

x_4x_3 / x_2x_1	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

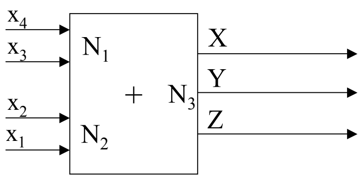
x_3 (columns 01, 11); x_4 (columns 11, 10); x_1 (rows 01, 11); x_2 (rows 11, 10)

$$\boxed{F_3 = x_4\overline{x}_2 \vee x_3\overline{x}_2\,\overline{x}_1 \vee x_4 x_3 \overline{x}_1}$$

$x_4x_3x_2x_1$	$F_1\ F_2\ F_3$
0 0 0 0	1 0 0
0 0 0 1	0 1 0
0 0 1 0	0 1 0
0 0 1 1	0 1 0
0 1 0 0	0 0 1
0 1 0 1	1 0 0
0 1 1 0	0 1 0
0 1 1 1	0 1 0
1 0 0 0	0 0 1
1 0 0 1	0 0 1
1 0 1 0	1 0 0
1 0 1 1	0 1 0
1 1 0 0	0 0 1
1 1 0 1	0 0 1
1 1 1 0	0 0 1
1 1 1 1	1 0 0

- Drei KPI, keine API oder RPI

- Anwendungsbeispiel 2 bit Addierer:



$x_4x_3x_2x_1$	X Y Z
0 0 0 0	0 0 0
0 0 0 1	0 0 1
0 0 1 0	0 1 0
0 0 1 1	0 1 1
0 1 0 0	0 0 1
0 1 0 1	0 1 0
0 1 1 0	0 1 1
0 1 1 1	1 0 0
1 0 0 0	0 1 0
1 0 0 1	0 1 1
1 0 1 0	1 0 0
1 0 1 1	1 0 1
1 1 0 0	0 1 1
1 1 0 1	1 0 0
1 1 1 0	1 0 1
1 1 1 1	1 1 0

- Entwurf der Teilfunktion X:

x_2x_1 \ x_4x_3	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	1	1	1
10	0	0	1	1

$$\boxed{X = x_4x_2 \vee x_3x_2x_1 \vee x_4x_3x_1}$$

$x_4x_3x_2x_1$	X Y Z
0 0 0 0	0 0 0
0 0 0 1	0 0 1
0 0 1 0	0 1 0
0 0 1 1	0 1 1
0 1 0 0	0 0 1
0 1 0 1	0 1 0
0 1 1 0	0 1 1
0 1 1 1	1 0 0
1 0 0 0	0 1 0
1 0 0 1	0 1 1
1 0 1 0	1 0 0
1 0 1 1	1 0 1
1 1 0 0	0 1 1
1 1 0 1	1 0 0
1 1 1 0	1 0 1
1 1 1 1	1 1 0

- Drei KPI, keine API oder RPI

- Entwurf der Teilfunktion Y:

x_2x_1 \ x_4x_3	00	01	11	10
00	0	0	1	1
01	0	1	0	1
11	1	0	1	0
10	1	1	0	0

x_3: Spalten 01, 11; x_4: Spalten 11, 10; x_1: Zeilen 01, 11; x_2: Zeilen 11, 10

$$\boxed{\begin{aligned} Y = {} & x_4\overline{x_3}\,\overline{x_2} \vee x_4\overline{x_2}\,\overline{x_1} \vee \overline{x_4}x_3\overline{x_2}x_1 \\ & \vee x_4x_3x_2x_1 \vee \overline{x_4}\,\overline{x_3}x_2 \vee \overline{x_4}x_2\overline{x_1} \end{aligned}}$$

$x_4x_3x_2x_1$	X Y Z
0 0 0 0	0 0 0
0 0 0 1	0 0 1
0 0 1 0	0 1 0
0 0 1 1	0 1 1
0 1 0 0	0 0 1
0 1 0 1	0 1 0
0 1 1 0	0 1 1
0 1 1 1	1 0 0
1 0 0 0	0 1 0
1 0 0 1	0 1 1
1 0 1 0	1 0 0
1 0 1 1	1 0 1
1 1 0 0	0 1 1
1 1 0 1	1 0 0
1 1 1 0	1 0 1
1 1 1 1	1 1 0

- Sechs KPI (davon zwei Minterme), keine API oder RPI

- Entwurf der Teilfunktion Z:

x_2x_1 \ x_4x_3	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	0

x_3: Spalten 01, 11; x_4: Spalten 11, 10; x_1: Zeilen 01, 11; x_2: Zeilen 11, 10

$$\boxed{Z = \overline{x_3}x_1 \vee x_3\overline{x_1}}$$

$x_4x_3x_2x_1$	X Y Z
0 0 0 0	0 0 0
0 0 0 1	0 0 1
0 0 1 0	0 1 0
0 0 1 1	0 1 1
0 1 0 0	0 0 1
0 1 0 1	0 1 0
0 1 1 0	0 1 1
0 1 1 1	1 0 0
1 0 0 0	0 1 0
1 0 0 1	0 1 1
1 0 1 0	1 0 0
1 0 1 1	1 0 1
1 1 0 0	0 1 1
1 1 0 1	1 0 0
1 1 1 0	1 0 1
1 1 1 1	1 1 0

- Zwei KPI, keine API oder RPI

➢ DMF gesucht für folgende Funktion f:

x_1 x_1

0	0	0	d	1	0	0	0
1	d	0	1	1	1	d	1
1	d	0	d	1	0	d	1
d	1	0	d	1	0	0	0

x_2 x_4 x_3 x_5

➢ PI, KPI, API, RPI ?

Graphische Minimierung im KV-Diagramm

Digitaltechnik
Entwurf zweistufiger Logik

➢ DMF gesucht für folgende Funktion f:

x_1 x_1

0	0	0	d	1	0	0	0
1	d	0	1	1	1	d	1
1	d	0	d	1	0	d	1
d	1	0	d	1	0	0	0

x_2 x_4 x_3 x_5

➢ KPI: $\boxed{x_3\overline{x_1}}$ $\boxed{x_5\overline{x_4}x_2}$ $\boxed{\overline{x_5}x_4\overline{x_3}}$

➢ RPI: $\boxed{x_2\overline{x_1}}$ $\boxed{\overline{x_3}x_2}$

- DMF gesucht für folgende Funktion f:

		x_1	x_1			x_1	x_1		
	0	0	0	d	1	0	0	0	
x_2	1	d	0	1	1	1	d	1	
x_2	1	d	0	d	1	0	d	1	x_4
	d	1	0	d	1	0	0	0	x_4
			x_3	x_3	x_3	x_3			
					x_5	x_5	x_5	x_5	

$$f^1 = x_3\overline{x_1} \vee x_5\overline{x_4}x_2 \vee \overline{x_3}x_2 \vee \overline{x_5}x_4\overline{x_3}$$

$$f^2 = x_3\overline{x_1} \vee x_5\overline{x_4}x_2 \vee x_2\overline{x_1} \vee \overline{x_5}x_4\overline{x_3}$$

- Gesucht: DMF für ein Schaltwerk zur Umsetzung von ASCII-Kode nach BCD-Darstellung für die Zahlen 0 bis 9:

Zeichen	**0**	**1**	**2**	**3**	**4**
ASCII	**0110000**	**0110001**	**0110010**	**0110011**	**0110100**
BCD	**0000**	**0001**	**0010**	**0011**	**0100**
Zeichen	**5**	**6**	**7**	**8**	**9**
ASCII	**0110101**	**0110110**	**0110111**	**0111000**	**0111001**
BCD	**0101**	**0110**	**0111**	**1000**	**1001**

- Die Bestimmung aller Primimplikanten ist die Ausgangsbasis für exakte Minimierungsverfahren
- Im zweiten Schritt wird aus diesen gefundenen Primimplikanten eine der kostengünstigsten bzw. die kostengünstigste Überdeckung ausgewählt
- Es existieren eine Reihe von Verfahren zu dieser exakten Herangehensweise:
  - Graphisches Verfahren unter Nutzung des KV-Diagramms
  - Algebraisches Verfahren nach Nelson [Nelson 55, Lipp 99]
  - Tabellarische Verfahren nach Quine-McCluskey [McCluskey 56, Katz 94] und das sogenannte Consensus-Verfahren nach Mott [Mott 60, Eschermann 92]
- Ein Problem des ersten Schritts in der exakten Minimierung ist, dass es Funktionen gibt, die bei n Variablen $3^n/n$ Primimplikanten besitzen
- Graphische und manuelle Herangehensweise nur für kleines n !
- **Zusätzlich:** Überdeckungsproblem des zweiten Schritts NP-vollständig
- Exakte Minimierung erfordert effiziente Implementierung beider Schritte, z.B. rekursive Primimplikantenbestimmung
- **Zunächst:** Tabellarisches Verfahren nach Quine-McCluskey

Tabellarisches Verfahren nach Quine-McCluskey:

- In dem Verfahren wird von der Minterm-Darstellung ausgehend eine tabellarische Verschmelzung ´´benachbarter Felder´´ in Analogie zur KV-Darstellung vorgenommen
- Überprüft werden alle Minterme, die sich nur in einer Variablen unterscheiden
- Zu diesem Zweck wird eine erste Tabelle erstellt, in der die Minterme nach Gruppen sortiert eingetragen werden
- Die Sortierung der Gruppen erfolgt nach der Anzahl der Einsen in der Belegung aufsteigend in der Tabelle
- Überprüft wird nun für jede mögliche Kombination zweier benachbarter Gruppen, ob eine Unterscheidung genau in einer Stelle gegeben ist und damit eine Zusammenfassung durch auslassen (-) der Variablen im Produktterm möglich ist
- Konnte ein Tabelleneintrag in einer Zusammenfassung verwendet werden, so wird er mit √ gekennzeichnet

Fortsetzung: Tabellarisches Verfahren nach Quine-McCluskey:

- Konnte ein Tabelleneintrag in keiner Zusammenfassung verwendet werden, so wird er mit * gekennzeichnet, d.h. er ist bereits ein Primimplikant
- Im Falle einer Zusammenfassung wird der entstandene Belegungsausdruck aus 0, 1, und – in die nächste Tabelle zur Weiterverarbeitung übernommen. Die Indizes der zugehörigen, überdeckten Minterme werden in einer eigenen Tabellenspalte mitgeführt
- Es erfolgt eine Sortierung der neugebildeten Belegungsausdrücke nach der Anzahl der Einsen in Gruppen
- Erneut erfolgt die Prüfung auf mögliche Zusammenfassung, deren Durchführung und Aufstellung neuer Tabellen
- Abbruchkriterium: Lassen sich in einer Tabelle keine Zusammenfassungen mehr finden, so kann keine weitere Tabelle mehr aufgestellt werden.
- Alle mit * gekennzeichneten Tabelleneinträge sind Primimplikanten
- Auffindung der irredundanten Normalform DMF durch Aufstellung und Auswertung der Primimplikantentabelle

- **Veranschaulichung der Primimplikantenbestimmung** nach Quine-McCluskey
- Beispielfunktion: $f = \left(\bigvee_{j \in \{4,5,6,8,9,10,13\}} (m_j) \right) \vee \left(\bigvee_{i \in \{0,7,15\}} (d_i) \right)$
- Auflistung aller Minterme (Eins- und d-Stellenmenge)

j	Belegung	
4	0100	
5	0101	
6	0110	
8	1000	
9	1001	
10	1010	
13	1101	
0	0000	
7	0111	
15	1111	

Sortierung

j	Belegung	
0	0000	
4	0100	
8	1000	
5	0101	
6	0110	
9	1001	
10	1010	
7	0111	
13	1101	
15	1111	

- **Veranschaulichung der Primimplikantenbestimmung** nach Quine-McCluskey (Fortsetzung)
- Gruppenweise Überprüfung aller möglichen Paare auf Zusammenfassung:

0000
0100

0000
1000

1. Tabelle

j	Belegung	
0	0000	√
4	0100	√
8	1000	√
5	0101	
6	0110	
9	1001	
10	1010	
7	0111	
13	1101	
15	1111	

Zusammenfassung
(1. und 2. Gruppe)

2. Tabelle

j	Belegung	
0, 4	0-00	
0, 8	-000	

- **Veranschaulichung der Primimplikantenbestimmung** nach Quine-McCluskey (Fortsetzung)
- Gruppenweise Überprüfung aller möglichen Paare auf Zusammenfassung:

0100
0101

0100
0110

1. Tabelle

j	Belegung	
0	0000	√
4	0100	√
8	1000	√
5	0101	√
6	0110	√
9	1001	√
10	1010	√
7	0111	
13	1101	
15	1111	

Zusammenfassung
(2. und 3. Gruppe)

2. Tabelle

j	Belegung	
0, 4	0-00	
0, 8	-000	
4, 5	010-	
4, 6	01-0	
8, 9	100-	
8, 10	10-0	

- **Veranschaulichung der Primimplikantenbestimmung** nach Quine-McCluskey (Fortsetzung)
- Gruppenweise Überprüfung aller möglichen Paare auf Zusammenfassung:

1. Tabelle

j	Belegung	
0	0000	√
4	0100	√
8	1000	√
5	0101	√
6	0110	√
9	1001	√
10	1010	√
7	0111	√
13	1101	√
15	1111	

Zusammenfassung
(3. und 4. Gruppe)

2. Tabelle

j	Belegung	
0, 4	0-00	
0, 8	-000	
4, 5	010-	
4, 6	01-0	
8, 9	100-	
8, 10	10-0	
5, 7	01-1	
5, 13	-101	
6, 7	011-	
9, 13	1-01	

- **Veranschaulichung der Primimplikantenbestimmung** nach Quine-McCluskey (Fortsetzung)
- Gruppenweise Überprüfung aller möglichen Paare auf Zusammenfassung:

1. Tabelle

j	Belegung	
0	0000	√
4	0100	√
8	1000	√
5	0101	√
6	0110	√
9	1001	√
10	1010	√
7	0111	√
13	1101	√
15	1111	√

Zusammenfassung
(4. und 5. Gruppe)

2. Tabelle

j	Belegung	
0, 4	0-00	
0, 8	-000	
4, 5	010-	
4, 6	01-0	
8, 9	100-	
8, 10	10-0	
5, 7	01-1	
5, 13	-101	
6, 7	011-	
9, 13	1-01	
7, 15	-111	
13, 15	11-1	

- **Veranschaulichung der Primimplikantenbestimmung** nach Quine-McCluskey (Fortsetzung)
- Gruppenweise Überprüfung aller möglichen Paare auf Zusammenfassung:

2. Tabelle

j	Belegung	
0, 4	0-00	*
0, 8	-000	*
4, 5	010-	√
4, 6	01-0	√
8, 9	100-	*
8, 10	10-0	*
5, 7	01-1	√
5, 13	-101	√
6, 7	011-	√
9, 13	1-01	*
7, 15	-111	√
13, 15	11-1	√

Zusammenfassung

3. Tabelle

j	Belegung	
4,5,6,7	01--	
5,7,13,15	-1-1	

- **Veranschaulichung der Primimplikantenbestimmung** nach Quine-McCluskey (Fortsetzung)
- Gruppenweise Überprüfung aller möglichen Paare auf Zusammenfassung:

3. Tabelle

j	Belegung	
4,5,6,7	01--	*
5,7,13,15	-1-1	*

Zusammenfassung

Abbruch !

4. Tabelle

j	Belegung	

- **Veranschaulichung der Primimplikantenbestimmung** nach Quine-McCluskey (Fortsetzung)
- Gegenprobe der gefundenen Primimplikanten über KV-Diagramm:

$\boxed{0-00 = \overline{x_4}\,\overline{x_2}\,\overline{x_1}}$

$\boxed{-000 = \overline{x_3}\,\overline{x_2}\,\overline{x_1}}$

$\boxed{100- = x_4\,\overline{x_3}\,\overline{x_2}}$

$\boxed{10-0 = x_4\,\overline{x_3}\,\overline{x_1}}$

$\boxed{1-01 = x_4\,\overline{x_2}\,x_1}$

$\boxed{01-- = \overline{x_4}\,x_3}$

$\boxed{-1-1 = x_3 x_1}$

x_2x_1 \ x_4x_3	00	01	11	10
00	d	1	0	1
01	0	1	1	1
11	0	d	d	0
10	0	1	0	1

x_3 (columns 01, 11); x_4 (columns 11, 10); x_1 (rows 01, 11); x_2 (rows 11, 10)

- Auffindung der irrendundanten Normalform oder irredundanten Hülle nach Quine-McCluskey durch Aufstellung einer Primimplikantentabelle
- **Primimplikantentabelle:** Primimplikanten über Minterme (ohne d-Menge)

	4	5	6	8	9	10	13
0,4 (0-00)	X						
0,8 (-000)				X			
8,9 (100-)				X	X		
8,10 (10-0)				X		X	
9,13 (1-01)					X		X
4,5,6,7 (01--)	X	X	X				
5,7,13,15 (-1-1)		X					X

- Bestimmung der KPI in der Primimplikantentabelle
- Erkennbar ist ein KPI gegeben, wenn in einer Mintermspalte nur eine Markierung zu finden ist

	4	5	6	8	9	10	13
0,4 (0-00)	X						
0,8 (-000)				X			
8,9 (100-)				X	X		
8,10 (10-0)				X		X	
9,13 (1-01)					X		X
4,5,6,7 (01--)	X	X	X				
5,7,13,15 (-1-1)		X					X

- Die gefundenen KPI gehen alle in die irredundante Normalform ein und überdecken typisch weitere Minterme
- In der Primimplikantentabelle können Spalten gestrichen werden, die durch die gefundenen KPI überdeckt werden

	4	5	6	8	9	10	13
0,4 (0-00)	X						
0,8 (-000)				X			
8,9 (100-)				X	X		
8,10 (10-0)				X		X	
9,13 (1-01)					X		X
4,5,6,7 (01--)	X	X	X				
5,7,13,15 (-1-1)		X					X

- Im gegebenen Beispiel sind nur noch die Minterme 9 und 13 nicht durch die gefundenen KPI abgedeckt
- Aus den drei verbleibenden RPI ist die Lösung 1-01 die kostengünstigste

	4	5	6	8	9	10	13
0,4 (0-00)	X						
0,8 (-000)				X			
8,9 (100-)				X	X		
8,10 (10-0)				X		X	
9,13 (1-01)					X		X
4,5,6,7 (01--)	X	X	X				
5,7,13,15 (-1-1)		X					X

- Damit ergibt sich die exakte Lösung des gegebenen Minimierungsproblems zu:

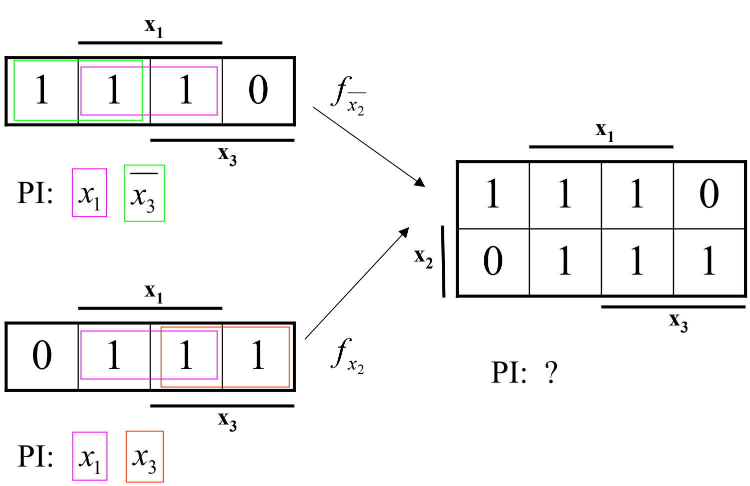
$$f = \boxed{\overline{x_4} x_3} \vee \boxed{x_4 \overline{x_3}\,\overline{x_1}} \vee \boxed{x_4 \overline{x_2} x_1}$$

x_2x_1 \ x_4x_3	00	01	11	10
00	0	1	0	1
01	0	1	1	1
11	0	1	0	0
10	0	1	0	1

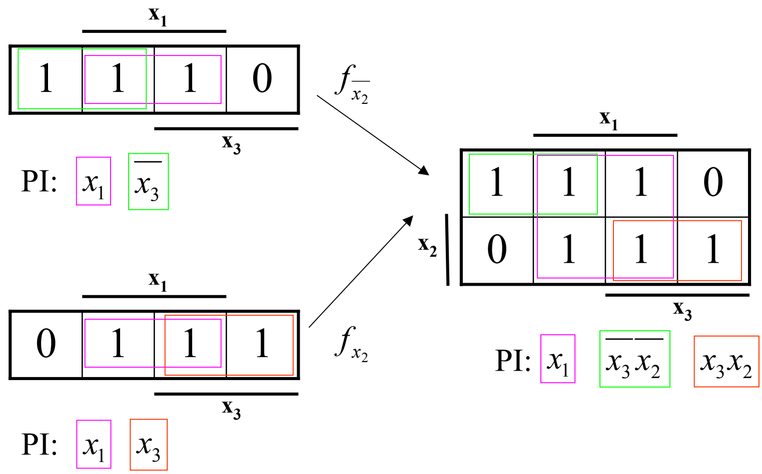
x_3 (Spalten 01, 11), x_1 (Zeilen 01, 11), x_2 (Zeilen 11, 10), x_4 (Spalten 11, 10)

- In der Literatur werden zur effizienten Durchführung der exakten Logikminimierung Verfahren zur rekursiven Bestimmung der Primimplikanten vorgestellt [Eschermann 92]
- Im zweiten Schritt folgt für die Menge der gefundenen PI dann die Lösung des sogenannten Überdeckungsproblems
- Der rekursive Ansatz beruht auf dem Entwicklungssatz und der Bestimmung der PI einer Funktion f aus den PI ihrer beiden Kofaktoren
- Die Zerlegung in Kofaktoren kann soweit durchgeführt werden, bis diese nur noch aus einem Würfel bestehen
- Kann eine Funktion mit nur einem Würfel repräsentiert werden, so ist dieser Würfel analog zu einem PI ein Primwürfel [Eschermann 92]
- Hier soll nur kurz das Prinzip der Bildung der PI von f aus den PI der Kofaktoren aufgezeigt werden
- Divide-et-impera Schema zur effizienteren Durchführung

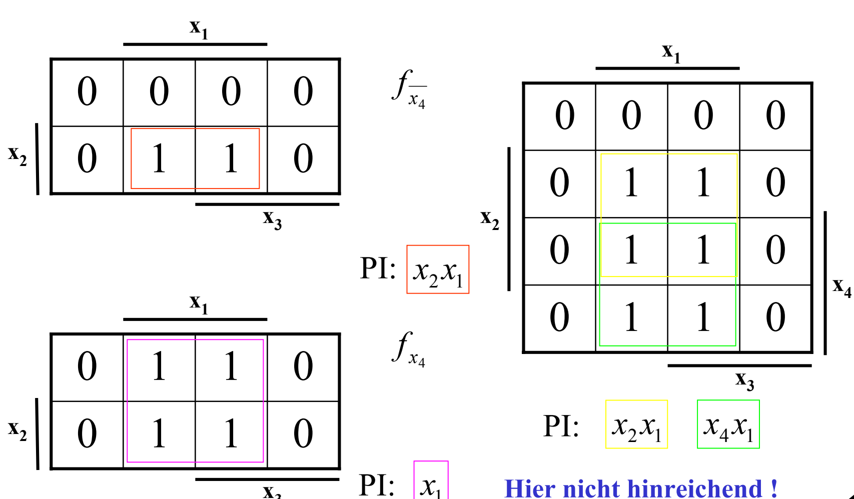
- PI der Kofaktoren zu einem Zerlegungspunkt seien bekannt. Aus ihnen seien die PI der Funktion f zu bestimmen !



- Die neuhinzukommende Variable muss mit den PI der Kofaktoren verknüpft werden (**Ausnahme:** PI die in beiden Kofaktoren vorkommen)



- Sind die vorliegenden PI der beiden Kofaktoren allein hinreichend zur Bestimmung der PI von f ?



Hier nicht hinreichend !

Rekursive Primimplikantenbestimmung

- Die aus der Und-Verknüpfung der Kofaktoren resultierenden PI bleiben als gemeinsamer Teil für f erhalten

x_1 (columns 2–3), x_3 (columns 3–4), x_2 (row 2)

0	0	0	0
0	1	1	0

$f_{\bar{x}_4} \wedge f_{x_4}$

PI: $x_2 x_1$

x_1 (columns 2–3), x_3 (columns 3–4), x_2 (row 2)

0	0	0	0
0	1	1	0

$f_{\bar{x}_4}$

PI: $x_2 x_1 \bar{x}_4$

x_1 (columns 2–3), x_3 (columns 3–4), x_2 (row 2)

0	1	1	0
0	1	1	0

f_{x_4}

PI: $x_4 x_1$

x_1 (columns 2–3), x_3 (columns 3–4), x_2 (rows 2–3), x_4 (rows 3–4)

0	0	0	0
0	1	1	0
0	1	1	0
0	1	1	0

PI: $x_2 x_1$ $x_4 x_1$

Rekursive Primimplikantenbestimmung

- Allgemeines Vorgehen für eine Funktion f mit:

$$f = x \wedge f_x \vee \bar{x} \wedge f_{\bar{x}}$$

- Die Menge PI(f) der Primimplikanten von f wird gebildet durch:

$$PI(f) = PI(f_x \wedge f_{\bar{x}}) \cup \bar{x} \circ (PI(f_{\bar{x}}) \setminus PI(f_x \wedge f_{\bar{x}})) \cup$$
$$x \circ (PI(f_x) \setminus PI(f_x \wedge f_{\bar{x}}))$$

- In diesem Kontext bedeutet $x \circ PI$ die Menge von Produkttermen, die durch Und-Verknüpfung von x mit allen Elementen von PI entsteht, z.B.:

$$x_4 \circ \{\bar{x}_2 x_1, x_2\} = \{x_4 \bar{x}_2 x_1, x_4 x_2\}$$

$$x_4 \circ \{\ \} = \{\ \}$$

$$\{\bar{x}_2 x_1, x_2\} \setminus \{\bar{x}_2 x_1\} = \{x_2\}$$

- Vorgehen für Funktion f aus dem (Gegen)Beispiel:

$$f = x_4 \wedge f_{x_4} \vee \overline{x_4} \wedge f_{\overline{x_4}}$$

- Die Menge PI(f) der Primimplikanten von f wird gebildet durch:

$$PI(f) = PI(f_{x_4} \wedge f_{\overline{x_4}}) \cup \overline{x_4} \circ (PI(f_{\overline{x_4}}) \setminus PI(f_{x_4} \wedge f_{\overline{x_4}})) \cup$$
$$x_4 \circ (PI(f_{x_4}) \setminus PI(f_{x_4} \wedge f_{\overline{x_4}}))$$

$$PI(f) = \{x_2 x_1\} \cup \overline{x_4} \circ (\{x_2 x_1\} \setminus \{x_2 x_1\}) \cup$$
$$x_4 \circ (\{x_1\} \setminus \{x_2 x_1\})$$

$$PI(f) = \{x_2 x_1\} \cup \overline{x_4} \circ \{\ \} \cup x_4 \circ \{x_1\}$$

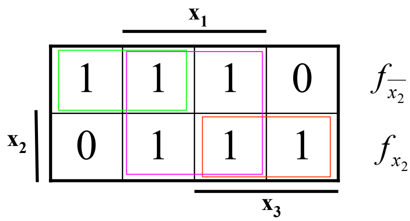
$$PI(f) = \{x_2 x_1,\ x_4 x_1\}$$

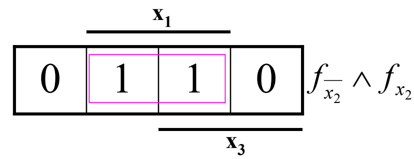
- Damit ist die Bestimmung der PI einer Funktion in n Variablen zurückführbar auf die Bestimmung der PI dreier Funktionen in n-1 Variablen
- Die Zerlegung kann rekursiv dem Divide-et-impera Schema nach durchgeführt werden
- Abbruchbedingungen für die Rekursion:
  - PI(0)={}
  - PI(1)={1}
  - $PI(\bar{x})=\{\bar{x}\}$
  - PI(x)={x}

- Mögliche Verfahrensvereinfachung für sogenannte unate Funktionen
- Weitere Detaillierung in fachlicher Vertiefungsveranstaltung
- Abschliessendes Beispiel:

Rekursive Primimplikantenbestimmung

- Die neuhinzukommende Variable muss mit den PI der Kofaktoren verknüpft werden (**Ausnahme:** PI die in beiden Kofaktoren vorkommen)





$$PI(f_{\overline{x}_2}) = \{x_1, \overline{x}_3\}$$

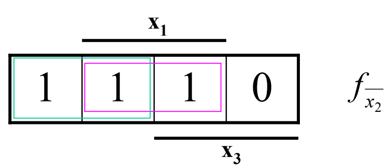
$$PI(f_{x_2}) = \{x_1, x_3\}$$

$$PI(f_{\overline{x}_2} \wedge f_{x_2}) = PI(x_1) = \{x_1\}$$

$$\boxed{\begin{aligned} PI(f) &= \{x_1\} \cup \overline{x}_2 \circ \overline{x}_3 \cup x_2 \circ x_3 \\ &= \{x_1, \overline{x}_2\overline{x}_3, x_2 x_3\} \end{aligned}}$$

Rekursive Primimplikantenbestimmung

- Betrachtung der tieferen Zerlegungsebene und der Abbruchkriterien:



$$\left.\begin{aligned} &PI(f_{\overline{x}_2}) = \{x_1, \overline{x}_3\} \\ &PI(f_{\overline{x}_2 x_3}) = \{x_1\} \\ &PI(f_{\overline{x}_2 \overline{x}_3}) = \{1\} \\ &PI(f_{\overline{x}_2 x_3} \wedge f_{\overline{x}_2 \overline{x}_3}) = \{x_1\} \end{aligned}\right\} \quad \begin{aligned} &PI(f_{\overline{x}_2}) = \{x_1\} \cup \overline{x}_3 \circ \{1\} \cup x_3 \circ \{\ \} \\ &PI(f_{\overline{x}_2}) = \{x_1, \overline{x}_3\} \end{aligned}$$

- Durch geeignete Methoden, z.B. durch aufwandsgünstige rekursive Bestimmung, können die PI einer Funktion sämtlich gefunden werden
- Die nachfolgende Auswahl relevanter PI mit dem Ziel der Gewinnung einer irredundanten und kostenminimalen DMF wurde bislang noch weitgehend pragmatisch angegangen
- Im folgenden soll nun das dahinterstehende sogenannte Überdeckungs-problem genauer betrachtet werden
- Es sollen systematischere Wege zur Auffindung einer DMF dargelegt und begangen werden
- Allgemeine Formulierung: Für eine Menge M sowie eine Menge T von m Teilmengen von M mit $T=\{M_1, ..., M_m\}$ und eine monotone Kostenfunktion, die jeder Teilmenge von T Kosten zuordnet, ist eine Teilmenge T* von T so zu bilden, dass

$$\bigcup M_i \in T^* = M$$

d.h. M bei minimalen Kosten vollständig überdeckt wird

- Repräsentation der Problems durch die Überdeckungsmatrix (s. Quine-McCluskey Methode) für willkürliche Beispielfunktion:

	0	2	4	9	10	12	14	p_k	c_k
10,11,14,15 (1-1-)					X		X	p_1	c_1
2,3,10,11 (-01-)		X			X		X	p_2	c_2
13,15 (11-1)						X		p_3	c_3
9,11 (10-1)				X				p_4	c_4
0,4 (0-00)	X		X					p_5	c_5
0,2 (00-0)	X	X						p_6	c_6
4,12 (-100)			X			X		p_7	c_7

3,7,11,15 gehören zur d-Stellenmenge !

- Eine Möglichkeit der Bestimmung einer Lösung wird durch das Verfahren von Petrick geboten
- Hierbei werden sogenannte Existenz- bzw. Präsenzvariablen e_i für jeden Primimplikanten p_i eingeführt, für die gilt:

$$e_i = \begin{cases} 1 & wenn\ p_i \in T^* \\ 0 & wenn\ p_i \notin T^* \end{cases}$$

- Alternativen in einer Spalte werden durch die Alternativen bzgl. der Präsenzvariablen abgebildet
- KPI müssen natürlich immer in der Lösung präsent sein ($e_i^{KPI}=1$)
- Für gegebenes Beispiel bestimmt sich der Petrickausdruck PA zu

$$PA = (e_5 \vee e_6) \wedge (e_2 \vee e_6) \wedge (e_5 \vee e_7) \wedge e_4 \wedge$$
$$(e_1 \vee e_2) \wedge (e_3 \vee e_7) \wedge (e_1 \vee e_3) \equiv 1$$

- Die Bearbeitung des Ausdrucks läuft prinzipiell wieder auf die gleiche Problematik, wie z.B. beim Nelson-Verfahren, hinaus
- **Wesentlicher Unterschied:** Variable kommen nur nicht negiert vor und bei PA =1 handelt es sich um eine Boolesche Gleichung für die eine (optimale) Lösung zu bestimmen ist
- Durch Umformen des Ausdrucks von PA lässt sich eine günstigere Darstellung zur Lösung der Aufgabe finden:

$$PA = e_1e_4e_6e_7 \vee e_2e_3e_4e_5 \vee e_1e_3e_4e_5e_6 \vee e_2e_3e_4e_6e_7 \vee e_1e_2e_4e_5e_7$$

- Jeder Produktterm stellt eine mögliche Kombination der PI der ursprünglichen Minimierungsaufgabe dar, die eine irredundante Überdeckung repräsentiert
- Zur Auswahl der günstigsten Lösungen müssen nun die Kosten herangezogen werden
- Dafür erfolgt zunächst die Zuweisung von Kosten an die PI durch die Anzahl ihrer Literale. Damit ist c_1=c_2=2, c_3 bis c_7=3

Überdeckungsproblem

- Daraus lassen sich die Kosten für die möglichen Lösungen errechnen:

$e_1e_4e_6e_7$	K = 11
$e_2e_3e_4e_5$	K = 11
$e_1e_3e_4e_5e_6$	K = 14
$e_2e_3e_4e_6e_7$	K = 14
$e_1e_2e_4e_5e_7$	K = 13

- Erkennbar existieren zwei Lösungen mit gleichen, minimalen Kosten:

$$y_1 = x_4\overline{x_3}x_1 \vee x_4x_2 \vee \overline{x_4x_3x_1} \vee x_3\overline{x_2x_1}$$

$$y_2 = x_4\overline{x_3}x_1 \vee \overline{x_3}x_2 \vee x_4x_3\overline{x_1} \vee \overline{x_4x_3x_1}$$

Überdeckungsproblem

- Die Aufstellung und Abarbeitung des Petrick-Ausdrucks lässt sich durch Ausnutzung von Information aus der Überdeckungstabelle von der Durchführung her vereinfachen
- So sind KPI unverzichtbarer Bestandteil der Lösung
- Sie zeichnen sich dadurch aus, dass es nur einen einzigen Eintrag in einer entsprechenden Spalte gibt
- Über die Auswertung aller Spalten sind die z KPI schnell zu finden und der Petrick-Ausdruck kann dargestellt werden als:

$$PA = e_1^{KPI} \wedge e_2^{KPI} \wedge \ldots \wedge e_z^{KPI} \wedge PA' \equiv 1$$

- Für das betrachtete Beispiel wird dann PA= e_4 & PA´ mit

$$PA' = (e_5 \vee e_6) \wedge (e_2 \vee e_6) \wedge (e_5 \vee e_7) \wedge$$
$$(e_1 \vee e_2) \wedge (e_3 \vee e_7) \wedge (e_1 \vee e_3) \equiv 1$$

- Bei gegebener Kostenannahme ergeben sich zwei Terme $e_1e_6e_7$ bzw. $e_2e_3e_5$, die noch um e_4 zur minimalen Lösung ergänzt werden müssen

Überdeckungsproblem

- Bei größeren Aufgaben und entsprechenden Überdeckungstabellen ist die manuelle Bearbeitung mittels des Petrick-Ausdrucks schwierig
- Einsatz von entsprechender CAD-Unterstützung
- Für kleinere Tabellen existiert graphische Methode zur Reduktion der Tabellen unter Ausnutzung sogenannter Dominanzverhältnisse
- Dabei existieren zwei Typen von Dominanzen:
  - Spaltendominanz (Spaltenüberdeckung)
  - Zeilendominanz (Zeilenüberdeckung)
- Diese werden im folgenden Verfahren zur Tabellenreduktion eingesetzt:
  - Bestimmung aller essentiellen Spalten und Streichung aller davon überdeckten Größen; leergewordene Zeilen streichen
  - Prüfung aller Spaltenpaare auf Dominanz; dominierende Spalten streichen; leergewordene Zeilen streichen
  - Prüfung aller Zeilenpaare auf Dominanz; dominierte Zeilen streichen; leergewordene Zeilen streichen, falls dadurch keine kostengünstigere Lösung entfällt (Streichung gleichwertiger Zeilen kann zu Verlust gleichwertiger Lösungen führen)

Überdeckungsproblem

- Fortsetzung des Verfahrens zur Tabellenreduktion:
  - Die drei Schritte solange wiederholen, bis keine Änderung mehr eintritt
  - Überprüfung, ob Tabelle abgearbeitet wurde:
    - Falls ja, Minimallösung(en) ermitteln
    - Falls nein, die vorliegende sogenannte zyklische Resttabelle mit dem Petrick-Verfahren abarbeiten
- Zunächst Erläuterung des Begriffs Spaltendominanz:

Kann gestrichen werden → i_1

i_1	i_2	p_k
X	X	p_1
X		p_2
X	X	p_3
X		p_4

$$PA' = (e_1 \vee e_2 \vee e_3 \vee e_4) \wedge (e_1 \vee e_3) = (e_1 \vee e_3)$$

Überdeckungsproblem

- Erläuterung des Begriffs Zeilendominanz:
- Dominiert eine Zeile i_1 eine Zeile i_2 bei geringeren oder gleichen Kosten, so kann die dominierte Zeile gestrichen werden
- **Fall 1:** Es sei $i_1 > i_2$ und $c_{i1} \leq c_{i2}$

					p_k	c_k
i_1	X	X	X	X	p_1	c_{i1}
i_2		X		X	p_2	c_{i2}
					p_3	c_{i3}
i_k	X		X		p_k	c_{ik}

- Zeile i_1 überdeckt mehr Spalten als Zeile i_2 bei gleichen oder geringeren Kosten

$$PA' = e_1 \vee e_2 e_k \quad \text{mit} \quad c_{i1} < c_{i2} + c_{ik}$$

Überdeckungsproblem

- **Fall 2:** Es sei $i_1 = i_2$ und $c_{i1} \leq c_{i2}$

					p_k	c_k
i_1	X	X		X	p_1	c_{i1}
i_2	X	X		X	p_2	c_{i2}

- Die beiden Zeilen haben gleichen Wert bezüglich der erzielten Überdeckung
- Bei $c_{i1} < c_{i2}$ wird die kostengünstigere Lösung gewählt, bei $c_{i1} = c_{i2}$ wird eine gleichwertige Lösung durch das Streichen verworfen

$$PA' = e_1 \vee e_2 \quad \text{mit} \quad c_{i1} = c_{i2}$$

Überdeckungsproblem

- **Beispiel** einer zyklischen Resttabelle:

	X		X
	X	X	
		X	X

- Diese kann mit den präsentierten Methoden nicht weiter vereinfacht werden
- Petrick-Verfahren ist eine effiziente Möglichkeit zur Abarbeitung der zyklischen Resttabelle und Generierung einer Lösung
- Anwendungsbeispiele zur Veranschaulichung der Methode

Überdeckungsproblem

- Anwendungsbeispiel aus Quine-McCluskey-Methode:

	4	5	6	8	9	10	13
0,4 (0-00)	X						
0,8 (-000)				X			
8,9 (100-)				X	X		
8,10 (10-0)				X		X	
9,13 (1-01)					X		X
4,5,6,7 (01--)	X	X	X				
5,7,13,15 (-1-1)		X					X

Überdeckungsproblem

- Bestimmung der KPI in der Überdeckungstabelle (essentielle Spalten)

	4	5	6	8	9	10	13
0,4 (0-00)	X						
0,8 (-000)				X			
8,9 (100-)				X	X		
8,10 (10-0)				X		X	
9,13 (1-01)					X		X
4,5,6,7 (01--)	X	X	X				
5,7,13,15 (-1-1)		X					X

Überdeckungsproblem

- Streichung aller durch die Kerngrößen (KPI) überdeckten Größen; leergewordenene Zeilen streichen

	4	5	6	8	9	10	13
0,4 (0-00)	X						
0,8 (-000)				X			
8,9 (100-)				X	X		
8,10 (10-0)				X		X	
9,13 (1-01)					X		X
4,5,6,7 (01--)	X	X	X				
5,7,13,15 (-1-1)		X					X

Überdeckungsproblem

- Streichung aller durch die Kerngrößen (KPI) überdeckten Größen; leergewordenene Zeilen streichen

	9	13
0,4 (0-00)		
0,8 (-000)		
8,9 (100-)	X	
8,10 (10-0)		
9,13 (1-01)	X	X
4,5,6,7 (01--)		
5,7,13,15 (-1-1)		X

Überdeckungsproblem

- Keine Spaltendominanz für gegebene Resttabelle
- Prüfung auf Zeilendominanz: Zweite Zeile dominiert erste und dritte Zeile !
- Prüfung der Kosten (Zusatzspalte)

	9	13	c_k
8,9 (100-)	X		3
9,13 (1-01)	X	X	3
5,7,13,15 (-1-1)		X	2

- Dritte Zeile hat die niedrigsten Kosten aber erfordert Hinzunahme der ersten Zeile, um gleiche Abdeckung zu erreichen
- **Daher:** Streichen der ersten und dritten Zeile

Überdeckungsproblem

- Tabelle abgearbeitet:

	9	13	c_k
9,13 (1-01)	X	X	3

- Minimale Lösung:

$$y = x_4 \overline{x_3}\,\overline{x_1} \vee \overline{x_4} x_3 \vee x_4 \overline{x_2} x_1$$

- **Weitere Beispiele:**

Digitaltechnik
Entwurf zweistufiger Logik

Überdeckungsproblem

- Primplikantentabelle des Beispiels 4.2.1:
  Bestimmung der KPI in der Überdeckungstabelle (essentielle Spalten)

		0	1	3	8	9	13	14	15	16	17	19	24	25	27	31
0,1,8,9,16,17,24,25	p_0	X	X		X	X				X	X		X	X		
17,19,25,27	p_1										X	X		X	X	
1,3,17,19	p_2		X	X							X	X				
27,31	p_3														X	X
15,31	p_4								X							X
14,15	p_5							X	X							
13,15	p_6						X		X							
9,13	p_7					X	X									

Überdeckungsproblem

- Streichung aller durch die Kerngrößen (KPI) überdeckten Größen; leergewordenene Zeilen streichen

		0	1	3	8	9	13	14	15	16	17	19	24	25	27	31
0,1,8,9,16,17,24,25	p_0	X	X		X	X				X	X		X	X		
17,19,25,27	p_1										X	X		X	X	
1,3,17,19	p_2		X	X							X	X				
27,31	p_3														X	X
15,31	p_4								X							X
14,15	p_5							X	X							
13,15	p_6						X		X							
9,13	p_7					X	X									

Überdeckungsproblem

- Ergänzung um eine Kostenspalte für die weiteren Reduktionsschritte
- Liegt Spaltendominanz vor ?

		13	27	31	c_k
17,19,25,27 (1-0-1)	p_1		X		3
27,31 (11-11)	p_3		X	X	4
15,31 (-1111)	p_4			X	4
13,15 (011-1)	p_6	X			4
9,13 (01-01)	p_7	X			4

Überdeckungsproblem

- Prüfung auf Zeilendominanz
- **Erkennbar:** Zeile 2 dominiert Zeile 3 bei gleichen Kosten
- Streichung von Zeile 3 !

		13	27	31	c_k
17,19,25,27 (1-0-1)	p_1		X		3
27,31 (11-11)	p_3		X	X	4
13,15 (011-1)	p_6	X			4
9,13 (01-01)	p_7	X			4

Überdeckungsproblem

- Erneute Prüfung auf Zeilendominanz
- **Erkennbar:** Zeile 2 dominiert Zeile 1, allerdings bei größeren Kosten
- Allerdings: Um die notwendige Abdeckung, die durch Zeile 2 geboten wird, zu erreichen, braucht Zeile 1 einen zusätzlichen Term, z.B. die vorab gestrichene Zeile 3
- Damit addieren sich die Kosten zu 3+4=7 > 4 von Zeile 2
- **Daher:** Streichung von Zeile 1 !

		13	27	31	c_k
17,19,25,27 (1-0-1)	p_1		X		3
27,31 (11-11)	p_3		X	X	4
13,15 (011-1)	p_6	X			4
9,13 (01-01)	p_7	X			4

Überdeckungsproblem

- Erkennbar liegt keine Spaltendominanz in der verbleibenden Tabelle vor
- Erneute Prüfung auf Zeilendominanz
- **Erkennbar:** Die beiden Zeilen 2 und 3 überdecken sich bzw. sind exakt gleich bei gleichen Kosten
- Streichen einer der beiden Zeilen führt zum Verlust einer gleichwertigen Lösung !

		13	27	31	c_k
27,31 (11-11)	p_3		X	X	4
13,15 (011-1)	p_6	X			4
9,13 (01-01)	p_7	X			4

$$y_1 = \overline{x_2}\,\overline{x_1} \vee \overline{x_3}\,\overline{x_2}x_0 \vee \overline{x_4}x_3x_1x_1 \vee \boxed{x_4x_3x_1x_0} \vee \boxed{\overline{x_4}x_3\overline{x_1}x_0}$$

$$y_2 = \overline{x_2}\,\overline{x_1} \vee \overline{x_3}\,\overline{x_2}x_0 \vee \overline{x_4}x_3x_1x_1 \vee \boxed{x_4x_3x_1x_0} \vee \boxed{\overline{x_4}x_3x_2x_0}$$

Überdeckungsproblem

- Primplikantentabelle des Beispiels 4.2.1:
  Bildung des Petrick-Ausdrucks für das Überdeckungsproblem

		0	1	3	8	9	13	14	15	16	17	19	24	25	27	31
0,1,8,9,16,17,24,25	p_0	X	X		X	X				X	X		X	X		
17,19,25,27	p_1										X	X		X	X	
1,3,17,19	p_2		X	X							X	X				
27,31	p_3														X	X
15,31	p_4								X							X
14,15	p_5							X	X							
13,15	p_6						X		X							
9,13	p_7					X	X									

$$PA = e_0(e_0 \vee e_2)e_2(e_0 \vee e_7)(e_6 \vee e_7)e_5(e_4 \vee e_5 \vee e_6)(e_0 \vee e_1 \vee e_2)$$
$$(e_1 \vee e_2)(e_0 \vee e_1)(e_1 \vee e_3)(e_3 \vee e_4)$$

Überdeckungsproblem

- Vereinfachung des Petrick-Ausdrucks für das Überdeckungsproblem
- Regel R11´ (Absorbtion) erlaubt die Zusammenfassung bzw. Streichung aller Terme, die einen KPI enthalten:

$$x(x \vee y) = x$$

$$e_0(e_0 \vee e_2) = e_0$$

- Ursprünglicher Ausdruck:

$$PA = e_0(e_0 \vee e_2)e_2(e_0 \vee e_7)(e_6 \vee e_7)e_5(e_4 \vee e_5 \vee e_6)(e_0 \vee e_1 \vee e_2)$$
$$(e_1 \vee e_2)(e_0 \vee e_1)(e_1 \vee e_3)(e_3 \vee e_4)$$

- Zusammenfassung nach e_0:

$$PA = e_0 e_2(e_6 \vee e_7)e_5(e_4 \vee e_5 \vee e_6)(e_1 \vee e_2)(e_1 \vee e_3)(e_3 \vee e_4)$$

- Zusammenfassung nach weiteren KPI:

$$PA = e_0 e_2 e_5(e_6 \vee e_7)(e_1 \vee e_3)(e_3 \vee e_4)$$

Überdeckungsproblem

- Weitergehende Vereinfachung des Petrick-Ausdrucks für das Überdeckungsproblem durch Aufteilung in essentiellen Teil und PA´

$$PA = e_0 \wedge e_2 \wedge e_5 \wedge PA'$$

- Abarbeiten des Ausdrucks PA´:

$$PA' = (e_6 \vee e_7)(e_1 \vee e_3)(e_3 \vee e_4)$$

- Ausdistribuieren:

$$PA' = (e_6 e_1 \vee e_7 e_1 \vee e_6 e_3 \vee e_7 e_3)(e_3 \vee e_4)$$
$$= e_6 e_3 e_1 \vee e_7 e_3 e_1 \vee e_6 e_3 e_3 \vee e_7 e_3 e_3 \vee e_6 e_4 e_1 \vee e_7 e_4 e_1 \vee e_6 e_4 e_3 \vee e_7 e_4 e_3$$
$$= \underline{e_6 e_3 e_1} \vee \underline{e_7 e_3 e_1} \vee \underline{e_6 e_3} \vee \underline{e_7 e_3} \vee e_6 e_4 e_1 \vee e_7 e_4 e_1 \vee \underline{e_6 e_4 e_3} \vee \underline{e_7 e_4 e_3}$$

- Damit existieren zunächst acht (vier) mögliche Lösungen, aus denen bezüglich der Kosten die günstigste Variante gefunden werden muss:

- Aufstellung der Kosten durch Summation der Kosten in den beitragenden Literalen der Produktterme:

Hierbei haben anhand der Literalzahl $e_{3\text{-}7}$ Kosten von K=4, $e_{1,2}$ Kosten von K=3 und e_0 Kosten von K=2

$e_6e_3e_1$	K = 4+4+3=11
$e_7e_3e_1$	K = 4+4+3=11
e_6e_3	K = 4+4=8
e_7e_3	K = 4+4=8
$e_6e_4e_1$	K = 4+4+3=11
$e_7e_4e_1$	K = 4+4+3=11
$e_6e_4e_3$	K = 4+4+4=12
$e_7e_4e_3$	K = 4+4+4=12

- Damit existieren für PA´zwei gleichwertige Lösungen
- Diese müssen mit dem essentiellen Anteil zu PA zusammengefügt werden, z.B. zu

$$y_1 = \bar{x}_2\bar{x}_1 \vee \bar{x}_3\bar{x}_2x_0 \vee \bar{x}_4x_3x_1x_1 \vee \underbrace{\boxed{x_4x_3x_1x_0}}_{e_3} \vee \underbrace{\boxed{\bar{x}_4x_3\bar{x}_1x_0}}_{e_7}$$

- Primplikantentabelle des Beispiels zur Quine-McCluskey-Methode: Bildung des Petrick-Ausdrucks für das Überdeckungsproblem

	4	5	6	8	9	10	13
0,4 (0-00)	X						
0,8 (-000)				X			
8,9 (100-)				X	X		
8,10 (10-0)				X		X	
9,13 (1-01)					X		X
4,5,6,7 (01--)	X	X	X				
5,7,13,15 (-1-1)		X					X

$$PA = (e_0 \vee e_5)(e_5 \vee e_6)e_5(e_1 \vee e_2 \vee e_3)(e_2 \vee e_4)e_3(e_4 \vee e_6)$$

Überdeckungsproblem

- Vereinfachung des Petrick-Ausdrucks für das Überdeckungsproblem
- Regel R11´ (Absorbtion) erlaubt die Zusammenfassung bzw. Streichung aller Terme, die einen KPI enthalten:

$$x(x \vee y) = x$$

$$e_0(e_0 \vee e_2) = e_0$$

- Ursprünglicher Ausdruck:

$$PA = (e_0 \vee e_5)(e_5 \vee e_6)e_5(e_1 \vee e_2 \vee e_3)(e_2 \vee e_4)e_3(e_4 \vee e_6)$$

- Zusammenfassung nach e_3:

$$PA = (e_0 \vee e_5)(e_5 \vee e_6)e_5(e_2 \vee e_4)e_3(e_4 \vee e_6)$$

- Zusammenfassung nach weiteren KPI:

$$PA = e_3 e_5 (e_2 \vee e_4)(e_4 \vee e_6)$$

Überdeckungsproblem

- Weitergehende Vereinfachung des Petrick-Ausdrucks für das Überdeckungsproblem durch Aufteilung in essentiellen Teil und PA´

$$PA = e_3 \wedge e_5 \wedge PA'$$

- Abarbeiten des Ausdrucks PA´:

$$PA' = (e_2 \vee e_4)(e_4 \vee e_6)$$

- Ausdistribuieren:

$$PA' = (e_4 e_2 \vee e_4 e_4 \vee e_6 e_2 \vee e_6 e_4)$$

$$PA' = e_4 \vee e_6 e_2$$

- Damit existieren zunächst vier (zwei) mögliche Lösungen, aus denen bezüglich der Kosten die günstigste Variante gefunden werden muss:

Überdeckungsproblem

- Aufstellung der Kosten durch Summation der Kosten in den beitragenden Literalen der Produktterme:

Hierbei haben anhand der Literalzahl $e_{0\text{-}4}$ Kosten von K=3, $e_{5,6}$ Kosten von K=2

$e_4 e_2$	K = 3+3=6
e_4	K = 3
$e_6 e_2$	K = 2+3=5
$e_6 e_4$	K = 2+3=5

- Damit existiert für PA´ nur eine Lösung
- Diese muss mit dem essentiellen Anteil zu PA zusammengefügt werden:

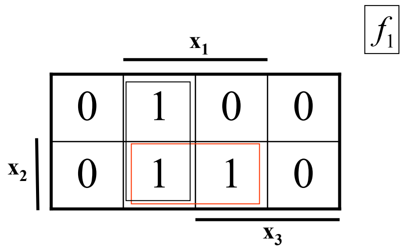
$$y = x_4 \overline{x_3 x_1} \vee \overline{x_4} x_3 \vee \overset{e_4}{x_4 \overline{x_2} x_1}$$

Funktionsbündel

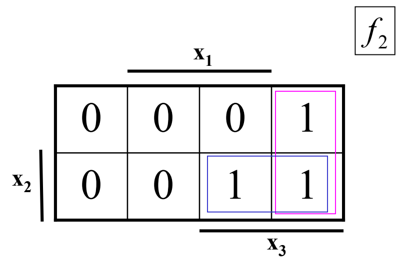
- Bislang wurden Boolesche Funktionen für mehrere Eingangsvariable und einen bzw. mehrere Ausgänge und ihre Schaltwerkrealisierung betrachtet
- Die bisher vorgestellten Methoden hatten zum Ziel jeweils eine exakte Lösung des Minimierungsproblems für jede mit einem Ausgang assoziierte Funktionzu finden
- Bislang wurde das Optimierungspotenzial durch funktionsübergreifende Einsparungsmöglichkeiten nicht genutzt
- Mögliche Einsparungen sind durch die Identifikation und gemeinsame Nutzung von (Produkt)Termen, die in allen DMF vorliegen gegeben
- **Weiterhin:** Gegebenfalls die Verwendung anderer Terme als die der irredundanten DMF der Einzelfunktion zu einer besseren Gesamtlösung für das Funktionsbündel führen
- Wie könnte ein systematischer Ansatz zur Optimierung von Funktionsbündeln aussehen ?

Funktionsbündel

- Bei der Realisierung eines Schaltwerks mit einem Funktionsbündel ist es für eine bündelbezogene minimale Gesamtlösung nicht hinreichend die PI und die DMF der einzelnen Funktionen des Bündels zu bestimmen
- Anschauungsbeispiel:



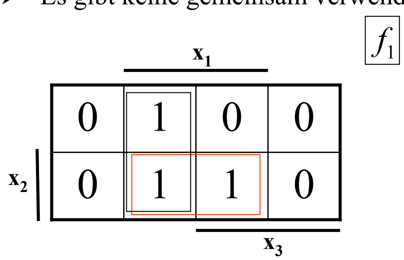
PI: x_2x_1 $\overline{x_3}x_1$

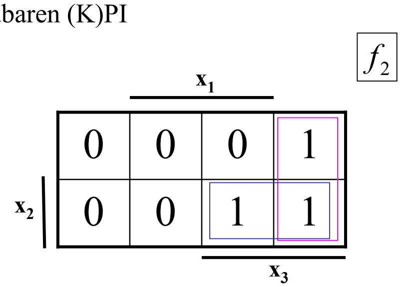


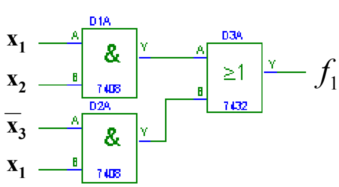
PI: $x_3\overline{x_1}$ x_3x_2

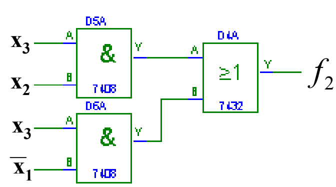
Funktionsbündel

- Erkennbar besteht jede DMF aus zwei KPI und damit vier Produktterme für beide Funktionen
- Es gibt keine gemeinsam verwendbaren (K)PI



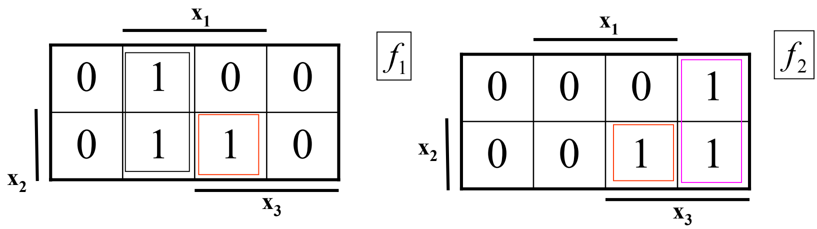


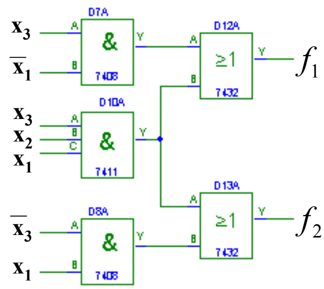




Funktionsbündel

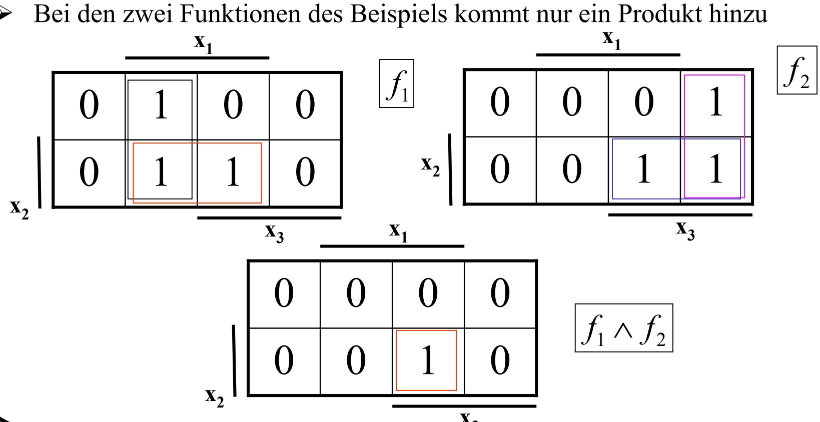
- Die Identifizierung eines gemeinsamen Produktterms, der kein PI ist, und zwei KPI ergeben ebenfalls eine vollständige und minimale Darstellung





Funktionsbündel

- Die systematische Bestimmung eines minimalen Schaltwerks für ein Funktionsbündel erfordert die Bestimmung der PI der einzelnen Funktionen sowie aller Produkte (Und-Verknüpfungen) zwischen diesen
- D.h., bei m Funktionen im Bündel müssen PI für $k=(2^m-m-1)$ Funktionen bestimmt werden !
- Bei den zwei Funktionen des Beispiels kommt nur ein Produkt hinzu



Funktionsbündel

- Für die gefundenen PI der k Funktionen muss nun eine Variante des Überdeckungsproblems gelöst werden
- Dabei wird ein Petrick-Ausdruck aus der Und-Verknüpfung der Ausdrücke für die einzelnen Funktionen gebildet

f_1

		x_1	x_1	
	0	1	0	0
x_2	0	1	1	0
			x_3	x_3

f_2

		x_1	x_1	
	0	0	0	1
x_2	0	0	1	1
			x_3	x_3

$e_1 : \overline{x_3} x_1$

$e_2 : x_2 x_1$

$e_3 : x_2 x_2 x_1$

$e_4 : x_3 \overline{x_1}$

$e_5 : x_3 x_2$

$e_3 : x_2 x_2 x_1$

$$\boxed{PA = [e_1(e_2 \vee e_3)][e_4(e_3 \vee e_5)] = 1}$$

x_3

Funktionsbündel

- Übliche Herangehensweise zur Abarbeitung des Petrick-Ausdrucks:

$$PA = [e_1(e_2 \vee e_3)][e_4(e_3 \vee e_5)] = 1$$

$$PA = [e_1 e_2 \vee e_1 e_3)][e_4 e_3 \vee e_4 e_5)] = 1$$

$$PA = e_4 e_3 e_2 e_1 \vee e_4 e_3 e_1 \vee e_5 e_4 e_2 e_1 \vee e_5 e_4 e_3 e_1 = 1$$

$$PA = e_4 e_3 e_1 \vee e_5 e_4 e_2 e_1 \vee e_5 e_4 e_3 e_1 = 1$$

Kostenbetrachtung:

Kostenbetrachtung:	$e_4 e_3 e_1$	K = 2+2+3=7
	$e_5 e_4 e_2 e_1$	K = 2+2+2+2=8
	$e_5 e_4 e_3 e_1$	K = 2+2+2+2=8

$$\boxed{f_1 = \overline{x_3} x_1 \vee x_2 x_2 x_1} \qquad \boxed{f_2 = x_3 \overline{x_1} \vee x_2 x_2 x_1}$$

x_3

- Gemeinsamer Entwurf der Teilfunktion f_1 und f_2 aus Beispiel 4.1.1:

x_2x_1 \ x_4x_3	00	01	11	10
00	0	0	1	0
01	1	1	1	1
11	1	1	1	1
10	0	0	1	1

e_3 e_1 e_2

$$f_1 = x_4x_2 \vee x_1 \vee x_4x_3$$

x_2x_1 \ x_4x_3	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	1	1	1	1
10	1	1	1	1

e_5 $e_6 = e_2$

$$f_2 = x_2 \vee x_4x_3$$

- Für beide Funktionen nur KPI, Bildung des Produkts von f_1 und f_2

- Bestimmung der PI des Produkts von f_1 und f_2 :

x_2x_1 \ x_4x_3	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	1	1	1	1
10	0	0	1	1

e_3 e_4 e_2

$$f_1 \wedge f_2 = x_4x_2 \vee x_2x_1 \vee x_4x_3$$

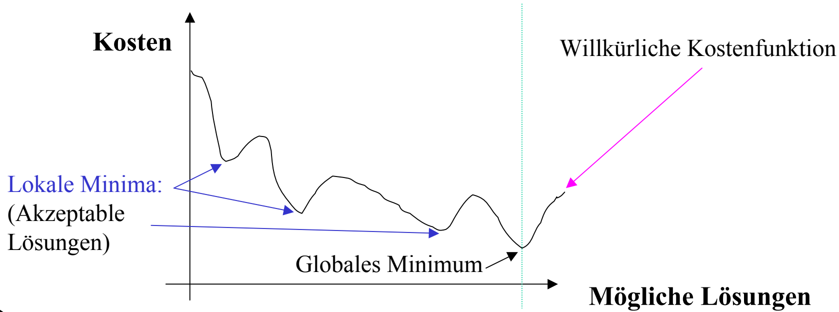
- Drei KPI für das Produkt von f_1 und f_2
- Gemeinsamer KPI sichtbar
- Formaler Petrick-Ausdruck: (unter Ausnutzung möglicher Zusammenfassungen)

$$[e_1e_2e_3(e_1 \vee e_4)][e_5e_2(e_5 \vee e_4)(e_5 \vee e_3)] = 1$$

$$[e_1\underline{e_2}e_3][e_5\underline{e_2}] = e_5e_3\underline{e_2}e_1 = 1$$

Exakte und heuristische Lösung

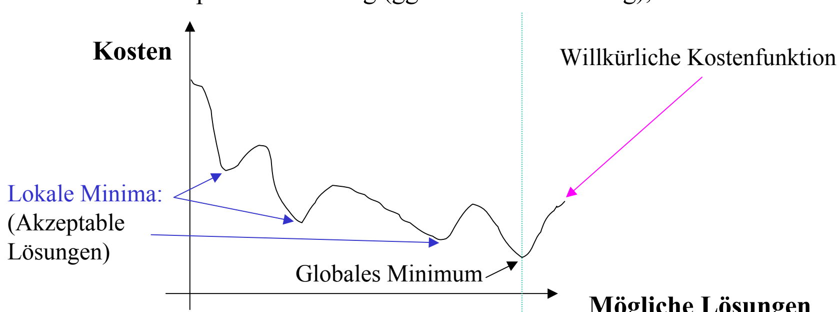
- Die bisher vorgestellten Methoden hatten zum Ziel eine exakte Lösung des Minimierungsproblems zu finden
- **Überlegung:** Ist dieses Ziel in allen Fällen mit beherrschbarem Aufwand erreichbar ?
- Damit stellt sich die Frage, wie der Aufwand der Lösungsfindung zu messen ist
- **Ebenso:** Was ist eine exakte Lösung und welche Alternativen gäbe es ?



Digitaltechnik
Entwurf zweistufiger Logik

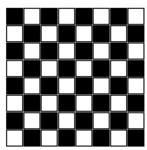
Exakte und heuristische Lösung

- Die exakte Lösung des Minimierungsproblems entspricht dem globalen Minimum der Kostenfunktion zu finden
- Um dieses **garantiert** zu finden, ist eine vollständige Durchsuchung aller möglichen Lösungen erforderlich
- Über heuristische Annahmen kann der zu durchsuchende Bereich des Lösungsraums eingeschränkt werden
- **Konsequenz:** Es kann nur noch die Auffindung eines lokalen Minimums, d.h. einer suboptimalen Lösung (ggf. die exakte Lösung), erwartet werden



Exakte und heuristische Lösung

- Die bisher betrachteten Minimierungsverfahren bestehen aus zwei Schritten:
  - Bestimmung aller Primimplikanten
  - Lösung des Überdeckungsproblems für alle gefundenen PI
- **Zur Erinnerung:**
  - Ein Problem des ersten Schritts in der exakten Minimierung ist, dass es Funktionen gibt, die bei n Variablen $3^n/n$ Primimplikanten besitzen
  - **Zusätzlich:** Überdeckungsproblem des zweiten Schritts NP-vollständig, d.h. es gibt kein Verfahren mit polynomiellen Aufwand zur Lösungsfindung
- Erkennbar sind beide Schritte damit in ihrem erforderlichem Aufwand in exponentieller Abhängigkeit von der Problemgröße n
- **Wirkung:** Die Geschichte mit dem Schachbrett und den Reiskörnern .....



- Ein Reiskorn auf das erste Feld
- Verdoppelung für jedes weitere Feld
- Letztes Feld: 2^{63} Körner

Exakte und heuristische Lösung

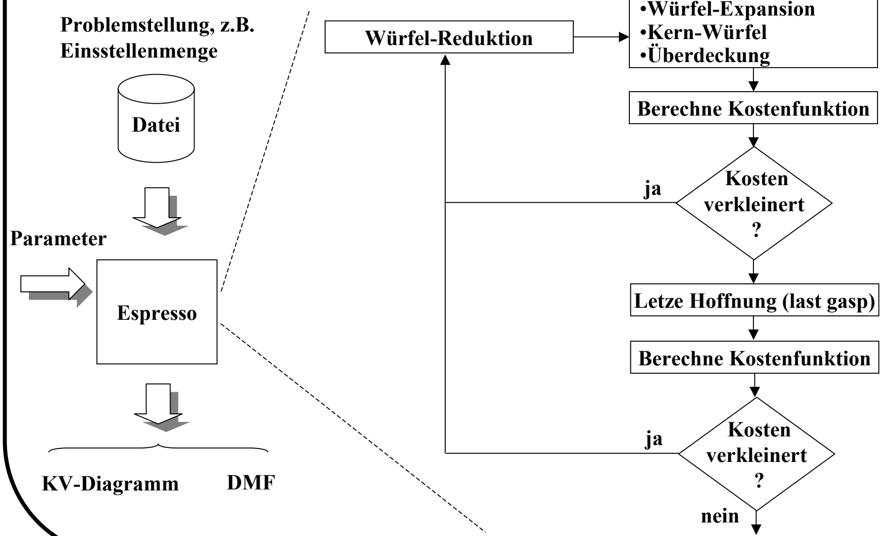
- Vergleichbare Wirkung auf Entwurfszeiten und damit Entwurfskosten sowie Time-to-market etc.
- Für kleine Variablenzahl n ist der resultierende Aufwand beherschbar, d.h. die Diskrepanz eines polynomiellen bzw. exponentiellen Aufwands vernachlässigbar
- Für wachsendes n wächst der Aufwand in unbeherrschbare Größen-ordnungen
- Daher sind Lösungen, die (akzeptable) lokale Minima zu erschwinglichem, d.h. polynomiellen Aufwand mit möglichst geringem Polynomgrad liefern, in praktischem Einsatz akzeptabel
- Entsprechende Heuristiken stellen oft die einzig praktikable Möglichkeit zur Lösungsfindung dar
- Im folgenden wird ein effizientes heuristisches Verfahren und seine rechnergestützte Umsetzung zu Logiksynthese betrachtet und angewendet

ESPRESSO

- Einfache heuristische Minimierungsverfahren, z.B. auf Basis von Funktion-zerlegung bzw. –entwicklung und Monotonieprüfung [Eschermann 92]
- Iterative heuristische Minimierung in rechnergestützter Implementierung
- Werkzeug Espresso (EspressoII) der University of California at Berkeley
- Grundlage der meisten in den neunziger Jahren kommerziell eingesetzten Werkzeuge zur zweistufigen Logiksynthese
- Das Programm setzt besonders effektive Heuristiken iterativ ein
- Die Minimierung von Bündelfunktion ist möglich
- Eine detaillierte Betrachtung der zugrunde liegenden Algorithmen überschreitet den Rahmen einer Grundstudiumsvorlesung
- Zunächst erfolgt prinzipielle Betrachtung der Vorgehensweise und die elementaren Schritte in der Iteration
- Das Werkzeug (empfohlene freiverfügbare Version !) wird dann anhand einiger Beispiele für die Minimierung von Schaltwerken in zweistufiger Realisierung demonstriert

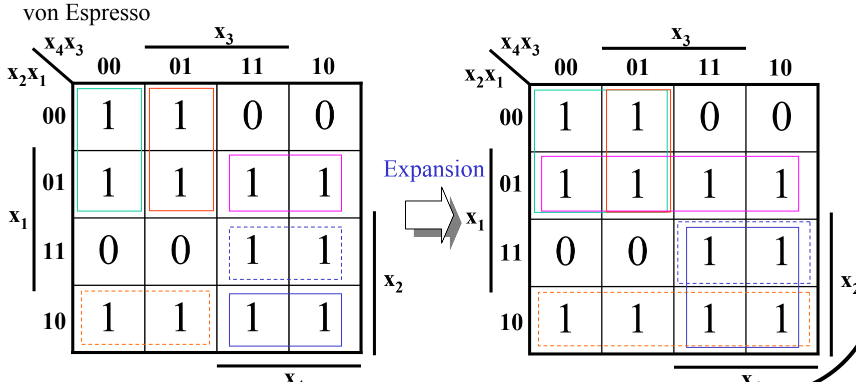
- Beschreibungen zu Espresso sind zusätzlich zu den Originalquellen der Autoren Brayton, Hachfeld, McMullen, Sangiovanni-Vincentelli [BHMS 84] den Literaturempfehlungen [Eschermann 92] sowie [Katz 94] zu entnehmen
- Anwenderinformation sind in den On-Line Dokumenten im Package von Chang [*Chang 94*] zu finden
- Auf der folgenden Folie wird ein Flussdiagramm eines vereinfachten Grundalgorithmus zu EspressoII vorgestellt [Eschermann 92]
- Das Verfahren setzt auf einer initialen Hülle von Würfeln bzw. Implikanten auf
- Die Optimierung wird durch elementare Schritte vorgenommen:
  - Expansion vorliegender Würfel bis zur maximal möglichen Größe
  - Eliminierung von überdeckten Würfel: Resultat Kern-Würfel bzw. PI
  - Bestimmung einer irredundanten Hülle (Überdeckungsproblem)
  - Reduktion vorliegender Kern-Würfel, wobei die zu realisierende Funktion vollständig überdeckt bleiben muß
- Iteratives Vorgehen erlaubt prinzipiell Überwindung lokaler Minima

- Flussdiagramm zu vereinfachtem Grundalgorithmus von EspressoII:



Logiksynthese
Digitaltechnik
Entwurf zweistufiger Logik

- Einzelne Schritte des EspressoII-Grundalgorithmus: Initialisierung und Expansion
- Espresso versucht nicht alle Implikanten zu finden und daraus die PI zu bestimmen, sondern die vorliegenden Implikanten einer Initialisierung werden mit dem Ziel der Maximierung expandiert
- Durch einen expandierten Implikanten überdeckte Implikanten werden gestrichen
- Die erreichbare Güte hängt erkennbar von der Reihenfolge und Richtung der Implikantenexpansion ab.
- In den entsprechenden Methoden begründet sich großteils die Leistungsfähigkeit von Espresso



- Einzelne Schritte des EspressoII-Grundalgorithmus: Irredundante Hülle
- Aus den initialen und expandierten Implikanten werden die PI einer irredundanten Hülle (DMF) bestimmt (KPI werden zur Aufwandsreduktion herausgenommen und erst am Ende der Minimierung wieder zur Hülle hinzugefügt; ihre Minterme gehen zur d-Stellenmenge über)

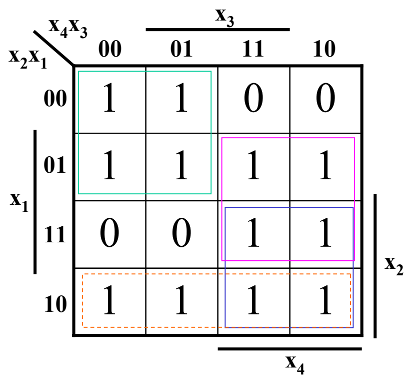
x_2x_1 \ x_4x_3	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	1	1
10	1	1	1	1

- Einzelne Schritte des EspressoII-Grundalgorithmus: Reduktion
- Die gefundene irredundanten Hülle kann zwar bereits eine gute Lösung darstellen, aber ggf. gibt es weitere irredundanten Hülle mit geringerer Termanzahl oder gleicher Termanzahl mit geringerer Literalzahl
- Zur Auffindung einer günstigeren Hülle reduziert Espresso die PI auf eine Größe, die die Funktion gerade noch abdecken

x_2x_1 \ x_4x_3	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	1	1
10	1	1	1	1

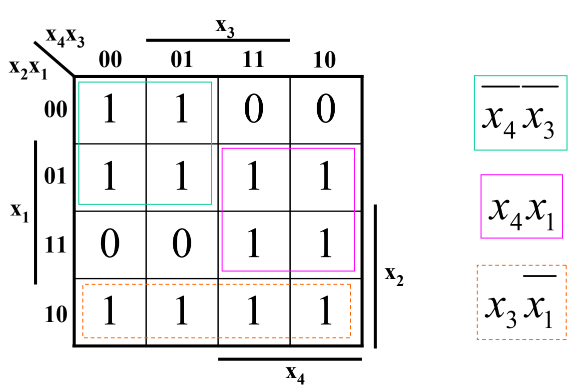
Logiksynthese

- Einzelne Schritte des EspressoII-Grundalgorithmus: Iteration
- Nach der Reduktion kann ist die resultierende Hülle typischerweise nicht mehr prim
- Ein iteratives Durchlaufen der Schritte Expansion und Bestimmung einer neuen irredundanten Hülle



Logiksynthese

- Einzelne Schritte des EspressoII-Grundalgorithmus: Iteration
- Nach der Reduktion kann ist die resultierende Hülle typischerweise nicht mehr prim
- Ein iteratives Durchlaufen der Schritte Expansion und Bestimmung einer neuen irredundanten Hülle



- Ein letzter Schritt überprüft, ob das Hinzufügen eines weiteren PI die Eliminierung von zwei anderen PI aus der gegenwärtigen Hülle erlauben würde
- Kann ein solcher Austausch erfolgen (Kostenreduktion), so wird ein neuer Iterationsschritt durchgeführt, sonst der Optimierungsvorgang abgebrochen

- Die Ein- und Ausgabe von Daten in Espresso erfolgt über ASCII-Dateien
- Die Eingabe erfolgt in Form einer Funktionstabelle, wobei zur Spezifikation die:
  - Einstellenmenge
  - Einstellen und d-Stellenmenge
  - Nullstellenmenge
  - Nullstellenmenge und d-Stellenmenge
- Ausgaben können je nach Optionsvorgaben KV-Diagramme oder die PI der irredundanten Hülle für die gegebene(n) Funktion(en) sein
- Details sind den Dateien Manual und Format der Demoversion zu finden

- Beispiel einer Eingabedatei anhand der zur Veranschaulichung im KV-Diagramm verwendeten Funktion:

$x_4x_3x_2x_1$

```
.i 4
.o 1
0000 1
0001 1
0100 1
0101 1
0010 1
0110 1
1110 1
1010 1
1101 1
1001 1
1111 1
1011 1
```

```
Output space # 0
1..1
1111
.11.
1111
```

			x_4		
	1	.	.	1	
x_1	1	1	1	1	
	.	1	1	.	x_2
	1	1	1	1	
			x_3		

- Minimierungsresultat für das gegebene Beispiel:

$\overline{x_4}\,\overline{x_1}$

$\overline{x_2}x_1$

x_4x_2

```
.i 4
.o 1
.p 3
0--0 1
--01 1
1-1- 1
.e
```

```
.i 4
.o 1
0000 1
0001 1
0100 1
0101 1
0010 1
0110 1
1110 1
1010 1
1101 1
1001 1
1111 1
1011 1
```

$x_4x_3x_2x_1$

		x_4	x_4		
	1	.	.	1	
x_1	1	1	1	1	
x_1	.	1	1	.	x_2
	1	1	1	1	x_2
			x_3	x_3	

			x_3	x_3	
	1	1	.	.	
x_1	1	1	1	1	
x_1	.	.	1	1	x_2
	1	1	1	1	x_2
			x_4	x_4	

- Minimierungsresultat für das Beispiel mit phase=0:

$x_4\overline{x_2}\,\overline{x_1}$

$\overline{x_4}x_2x_1$

```
.i 4
.o 1
#.phase 0
.p 2
1-00 1
0-11 1
.e
```

```
.i 4
.o 1
.phase 0
0000 1
0001 1
0100 1
0101 1
0010 1
0110 1
1110 1
1010 1
1101 1
1001 1
1111 1
1011 1
```

$x_4x_3x_2x_1$

		x_4	x_4		
	.	1	1	.	
x_1	.	.	.	.	
x_1	1	.	.	1	x_2
	.	.	.	.	x_2
			x_3	x_3	

			x_3	x_3	
	.	.	1	1	
x_1	.	.	.	.	
x_1	1	1	.	.	x_2
	.	.	.	.	x_2
			x_4	x_4	

Resultat: Maxterm-Lösung hier günstiger (9 vs. 8)

- Beispiel 4.2.1 (Quine-McCluskey, Funktion in fünf Variablen):

```
Output space # 0
1111 1111
.... ...1
...1 ..11
.... 111.
```

Zugehöriges KV-Diagramm in fünf Variablen

$x_4x_3x_2x_1x_0$

```
.i 5
.o 1
.phase 1
00000 1
00001 1
01000 1
10000 1
00011 1
01001 1
10001 1
11000 1
01101 1
01110 1
10011 1
11001 1
01111 1
11011 1
11111 1
```


- Minimierungsresultat für das Beispiel 4.2.1:

$$\overline{x_4}x_3\overline{x_1}x_0$$

$$x_4x_3x_1x_0$$

$$\overline{x_4}x_3x_2x_1$$

$$\overline{x_2}\,\overline{x_1}$$

$$\overline{x_3}\,\overline{x_2}x_0$$

```
.i 5
.o 1
#.phase 1
.p 5
01-01 1
11-11 1
0111- 1
--00- 1
-00-1 1
.e
```

$$\boxed{\begin{aligned} y_1 = {} & \overline{x_2}\,\overline{x_1} \vee \overline{x_3}\,\overline{x_2}x_0 \vee \overline{x_4}x_3x_1x_1 \\ & \vee x_4x_3x_1x_0 \vee \overline{x_4}x_3\overline{x_1}x_0 \end{aligned}}$$

$x_4x_3x_2x_1x_0$

```
.i 5
.o 1
.phase 1
00000 1
00001 1
01000 1
10000 1
00011 1
01001 1
10001 1
11000 1
01101 1
01110 1
10011 1
11001 1
01111 1
11011 1
11111 1
```

Logiksynthese

- Beispiel eines Funktionsbündels (Bsp. 4.1.1, ASCII nach BCD Ziffernumkodierung, 4 LSB)

```
Output space # 0  Output space # 2
.11.              ..1.
.11.              1111
.11.              1111
.11.              ..1.

Output space # 1  Output space # 3
..11              ..1.
.111              .11.
.111              1111
..11              1111
```

$x_4x_3x_2x_1$

```
.i 4
.o 4
0000 0000
0001 0001
0010 0010
0011 0011
0100 0100
0101 0101
0110 0110
0111 0111
1000 1000
1001 1001
1010 1111
1011 1111
1100 1111
1101 1111
1110 1111
1111 1111
```


Logiksynthese

- Beispiel eines Funktionsbündels (Bsp. 4.1.1, ASCII nach BCD Ziffernumkodierung, 4 LSB)

x_4

x_3 x_4x_2

x_2 x_4x_3

x_1 x_4x_3 x_4x_2

```
.i 4
.o 4
.p 6
1--- 1000
-1-- 0100
--1- 0010
---1 0001
1-1- 0101
11-- 0011
.e
```

$f_4f_3f_2f_1$

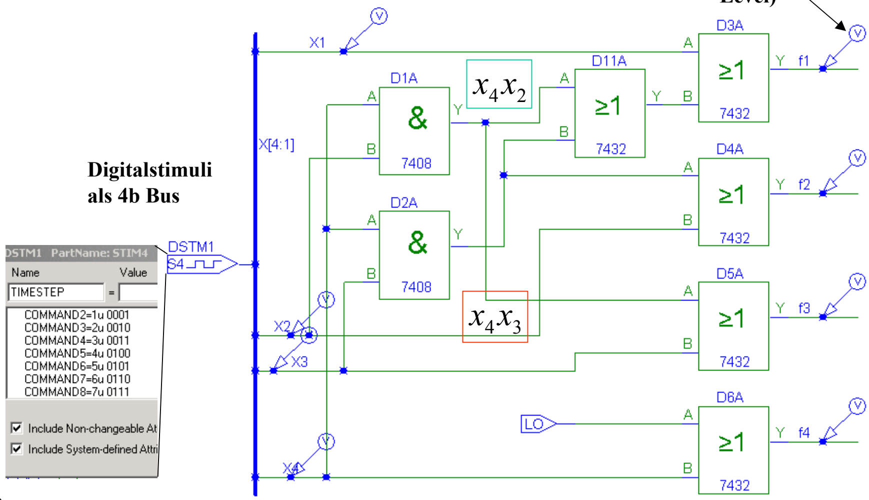
$x_4x_3x_2x_1$

```
.i 4
.o 4
0000 0000
0001 0001
0010 0010
0011 0011
0100 0100
0101 0101
0110 0110
0111 0111
1000 1000
1001 1001
1010 1111
1011 1111
1100 1111
1101 1111
1110 1111
1111 1111
```

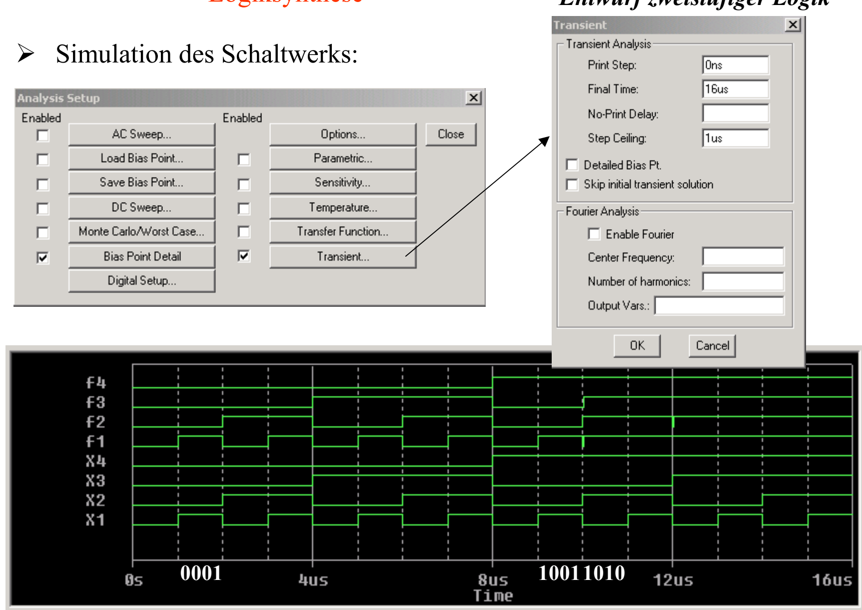
- Beispiel eines Funktionsbündels (Bsp. 4.1.1, ASCII nach BCD Ziffernumkodierung, 4 LSB)

Marker (Voltage Level)

Digitalstimuli als 4b Bus



- Simulation des Schaltwerks:



Zusammenfassung

- Alle betrachteten Minimierungsüberlegungen gelten aus Dualitätsgründen für einen Ansatz mit Maxtermen genauso wie mit Mintermen !
- Eine Funktion kann ggf. über die KNF (KMF) günstiger realisierbar sein als über die DNF (DMF) und umgekehrt; Probe z.B. durch Minimierung der invertierten Funktion !
- Die mit Espresso behandelbaren Problemgrößen sind nicht unbegrenzt (einige hundert bis tausend PI)
- Es existieren mittlerweile aktuellere, deutlich verbesserte Verfahren !

- Bislang Einschränkung auf zweistufige Und/Oder- bzw. Oder/Und-Realisierungen mit (nicht)negierten Literalen
- Weitere Basissysteme sowie mehrstufige Logik erfordern Erweiterung der Betrachtung
- Bislang nur logische Verknüpfung betrachtet; Zeitliche Effekte wurden vernachlässigt
- Beide Punkte besitzen starke Abhängigkeit zur konkreten Technologie