Tutorat 8

Primimplikantentafel, Methode von Petrick, XOR, Quine-McCluskey, Primimplikanten, Tiefe

Gruppe 9

Präsentator:
Jürgen Mattheis
(juergmatth@gmail.com)

Vorlesung von:
Prof. Dr. Scholl

Übungsgruppenbetreuung:
Tobias Seufert

21. August 2023

Universität Freiburg, Lehrstuhl für Rechnerarchitektur

Gliederung

Aufgabe 1

Aufgabe 1 I

Primimplikantentafel

Aufgabe 1.1

- On-Menge: $ON(f) = \{0000, 0001, 0100, 0101, 0110, 0111, 1000, 1010, 1100, 1110, 1111\}$
- Primimplikanten: $Prim(f) = \{- - 00, -1 - 0, -11-, 1 - -0, 0 - 0-, 01 - -\}$

Lösung 1.1

	0000	0001	0100	0101	0110	0111	1000	1010	1100	1110	1111
–00	1		1				1		1		
-1-0			1		1				1	1	
-11-					1	1				1	1
1–0							1	1	1	1	
0-0-	1	1	1	1							
01–			1	1	1	1					

Aufgabe 1 I

Primimplikantentafel - Anwenden der Reduktionsregeln in richtiger Reihenfolge

Lösung 1.1

	0001	0110	0111	1000	1010	1100	1110	1111
–00				1		1		
-1-0		1				1	1	
-11-		1	1				1	1
1–0				1	1	1	1	
0-0-	1							
01–								

Aufgabe 1 II

Primimplikantentafel - Anwenden der Reduktionsregeln in richtiger Reihenfolge

Lösung 1.1

	0001	0110	0111	1000	1010	1100	1110	1111
–00				1		1		
-1-0		1				1	1	
-11-		1	1				1	1
1–0				1	1	1	1	
0-0-	1							
01–		1	1					

- *1. Regel: 0-0-, 1–0 und -11- sind wesentlich*

Aufgabe 1 III

Primimplikantentafel - Anwenden der Reduktionsregeln in richtiger Reihenfolge

Lösung 1.1

–00	∅
-1-0	
-11-	
1–0	
0-0-	
01–	

- 1. Regel: *0-0-, 1–0 und -11- sind wesentlich*
- *Leere Tabelle, kein Petrick notwendig*
- $f_{min} = \bar{x}_1\bar{x}_3 + x_1\bar{x}_4 + x_2x_3$

Aufgabe 1 I

Primimplikantentafel - Anwenden der Reduktionsregeln in falscher Reihenfolge

Lösung 1.1

	0001	0110	0111	1000	1010	1100	1110	1111
–00				1		1		
-1-0		1				1	1	
-11-		1	1				1	1
1–0				1	1	1	1	
0-0-	1							
01–		1	1					

- *2. Regel:* *0000, 0100, 0101 dominieren 0001 ⇒ Lösche 0000, 0100 und 0101*

Aufgabe 1 II

Primimplikantentafel - Anwenden der Reduktionsregeln in falscher Reihenfolge

Lösung 1.1

	0001	1000	1010	1100	1111
–00		1		1	
-1-0				1	
-11-					1
1–0		1	1	1	
0-0-	1				
01–					

- 2. Regel: *0110, 0111 und 1110 dominieren 1111 ⇒ Lösche 0110, 0111 und 1110*

Aufgabe 1 III

Primimplikantentafel - Anwenden der Reduktionsregeln in falscher Reihenfolge

Lösung 1.1

	0001	1010	1111
–00			
-1-0			
-11-			1
1–0		1	
0-0-	1		
01–			

- 2. Regel: *1000 und 1100 dominieren 1010 ⇒ Lösche 1000 und 1100*

Aufgabe 1 IV

Primimplikantentafel - Anwenden der Reduktionsregeln in falscher Reihenfolge

Lösung 1.1

–00	∅
-1-0	
-11-	
1–0	
0-0-	
01–	

- *1. Regel: 0-0-, 1–0 und -11- sind wesentlich*
- *Leere Tabelle, kein Petrick notwendig*
- $f_{min} = \bar{x}_1\bar{x}_3 + x_1\bar{x}_4 + x_2x_3$

Aufgabe 2

Aufgabe 2 I

Methode von Petrick

Voraussetzungen 2.1

- *Absorption:* $A + AB = A$
- *Korollar:* $AA = A$

Aufgabe 2 II

Methode von Petrick

Aufgabe 2.1

	a	b	c	d	e	f
A		1		1		
B	1		1	1		
C			1	1		1
D		1			1	
E			1	1	1	
F	1	1				1

- Minterme: $\{a, b, c, d, e, f\}$
- Primimplikanten: $\{A, B, C, D, E, F\}$

Aufgabe 2 III

Methode von Petrick

Lösung 2.1

	a	b	c	e	f
A		1			
B	1		1		
C			1		1
D		1		1	
E			1	1	
F	1	1			1

- *Regel 2:* *d dominiert c* $\Rightarrow$ *Lösche d.*

Aufgabe 2 IV

Methode von Petrick

Lösung 2.1

	a	b	c	e	f
B	1		1		
C			1		1
D		1		1	
E			1	1	
F	1	1			1

- *Regel 3:* *D (oder F) dominiert A* $\Rightarrow$ *Lösche A*

Aufgabe 2 V

Methode von Petrick

Lösung 2.1

	a	b	c	e	f
B	1		1		
C			1		1
D		1		1	
E			1	1	
F	1	1			1

- *Keine weitere Reduktionsregel mehr anwendbar* $\Rightarrow$ *Petrick*

Aufgabe 2 VI

Methode von Petrick

Lösung 2.1

- *Petrick:*

$$\underbrace{(B+F)}_{\text{überdecken } a}\ \underbrace{(D+F)}_{\text{überdecken } b}\ \underbrace{(B+C+E)}_{\text{usw.}}(D+E)(C+F)$$

$$= (BD + BF + FD + F) \cdot (BD + BE + CD + CE + ED + E) \cdot (C + F)$$

$$= (BD + BF + FD + F) \cdot (BDC + BEC + CD + CE + EDC + BDF + BEF + CDF + CEF + EDF + EF)$$

$$= (BD + BF + FD + F) \cdot (CD + CE + BDF + EF)$$

$$= BDC + BDCE + BDF + BDEF + BFCD + BFCE + BFD + BFE + FDC + FDCE + FDB + FDE + FCD + FCE + FBD + FE$$

$$= BDC + BDF + BFD + FDC + FBD + \underline{FE}$$

- *Produkt mit den wenigsten Termen auswählen. Wenn es zwei solche Produkte gibt, dann wählt man die Terme mit den wenigsten Literalen*
- *Minimalpolynom* $f_{min} = F + E$

Aufgabe 3

Aufgabe 3 I

XOR, Quine-McCluskey, Primimplikanten, Tiefe

Lösung 3.1

- *Minterme:* $x_1'x_2'x_3,\ x_1'x_2x_3',\ x_1x_2'x_3',\ x_1x_2x_3$
- *Quine-McCluskey:*

$L_0^{x_1,x_2,x_3}$
001
010
100
111

  - $Prim(xor_3) = \emptyset$
  - *Es gibt keine Kombinationsmöglichkeiten der Minterme aus L_0. Daher sind alle L_1-Mengen leer und die Primmenge im nächsten Schritt enthält alle vier Minterme. Der Grund ist, dass sich alle Minterme in (mindestens) zwei Literalen unterscheiden, d.h. alle Minterme sind Primimplikanten.*

Aufgabe 3 II

XOR, Quine-McCluskey, Primimplikanten, Tiefe

Lösung 3.1

	1	*2*	*4*	*7*
$x_1'x_2'x_3$	*1*			
$x_1'x_2x_3'$		*1*		
$x_1x_2'x_3'$			*1*	
$x_1x_2x_3$				*1*

- *Es ist direkt ersichtlich, dass alle Primimplikanten wesentlich sind (Regel 1). Die Disjunktion aller Minterme ist also das einzige Minimalpolynom für xor_3*

Aufgabe 3 III

XOR, Quine-McCluskey, Primimplikanten, Tiefe

Lösung 3.2

- $$\bigvee_{\substack{(b_1,\ldots,b_n)\in\mathbb{B}^n \\ \text{mit } \sum_{i=1}^n b_i \text{ungerade}}} (x_1^{b_1} \wedge \ldots \wedge x_n^{b_n}), \text{ wobei } x_i^0 := x_i' \text{ und } x_i^1 := x_i$$
- *Der angegebene Ausdruck ist die Disjunktion aller Minterme für xor_n. In Aufgabenteil a) haben wir schon gesehen, dass alle Minterme wesentlich sind um xor_n darzustellen, da weder durch Quine-McCluskey noch Primimplikantentafel Reduktionen erreicht werden können. Deswegen beschreibt auch hier die Disjunktion der Minterme das Minimalpolynom. Es sind 2^{n-1} Primimplikanten, je mit Länge n.*

Voraussetzungen 3.3

- $xor_n(x_1, \ldots, x_n) = g(xor_i(x_1, \ldots, x_i), xor_{n-i}(x_{i+1}, \ldots, x_n))$ für $1 \leq i \leq n-1$

Aufgabe 3 IV

XOR, Quine-McCluskey, Primimplikanten, Tiefe

Lösung 3.3

- *Sei* $b_1 := xor_i(x_1, \ldots, x_i), \quad b_2 := xor_{n-i}(x_{i+1}, \ldots, x_n)$
  1. *Fall* $b_1 = 0,\ b_2 = 0$:
     Dann ist $\sum_{j=1}^{i} x_j$ *gerade*, $\sum_{j=i+1}^{n} x_j$ *gerade* $\Rightarrow \sum_{j=1}^{n} x_j$ *gerade* $\Rightarrow g(0,0) = 0 = xor_2(0,0)$
  2. *Fall* $b_1 = 1,\ b_2 = 0$:
     Dann ist $\sum_{j=1}^{i} x_j$ *ungerade*, $\sum_{j=i+1}^{n} x_j$ *gerade* $\Rightarrow \sum_{j=1}^{n} x_j$ *ungerade* $\Rightarrow g(1,0) = 1 = xor_2(1,0)$
  3. *Fall* $b_1 = 0,\ b_2 = 1$:
     Dann ist $\sum_{j=1}^{i} x_j$ *gerade*, $\sum_{j=i+1}^{n} x_j$ *ungerade* $\Rightarrow \sum_{j=1}^{n} x_j$ *ungerade* $\Rightarrow g(0,1) = 1 = xor_2(0,1)$
  4. *Fall* $b_1 = 1,\ b_2 = 1$:
     Dann ist $\sum_{j=1}^{i} x_j$ *ungerade*, $\sum_{j=i+1}^{n} x_j$ *ungerade* $\Rightarrow \sum_{j=1}^{n} x_j$ *gerade* $\Rightarrow g(1,1) = 0 = xor_2(1,1)$

Aufgabe 3 V

XOR, Quine-McCluskey, Primimplikanten, Tiefe

Lösung 3.3

- *Zusammengefasst:*

b_1	b_2	g
0	0	0
0	1	1
1	0	1
1	1	0

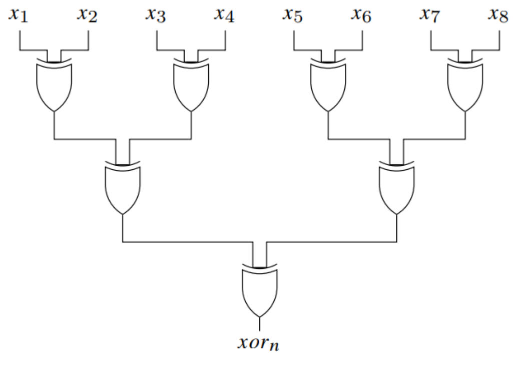
$\Rightarrow g = xor_2$, also $xor_n(x_1, \ldots, x_n) = xor_2(xor_i(x_1, \ldots, x_i), xor_{n-i}(x_{i+1}, \ldots, x_n))$ für $1 \leq i \leq n-1$

Aufgabe 3 VI

XOR, Quine-McCluskey, Primimplikanten, Tiefe

Lösung 3.4

- *Ein balancierter Baum hat die geringste Tiefe. D.h. wähle* $i = \lceil \frac{n}{2} \rceil$



Aufgabe 3 VII

XOR, Quine-McCluskey, Primimplikanten, Tiefe

Lösung 3.5

- *Für einen balancierten Baum ist Kosten* $= \#XOR{-}Gatter = n - 1$ *und längster Pfad* $= \lceil log(n) \rceil$. *Bzw. da vorgegeben, dass* $n = 2^k$ *gilt, ist auch* $log(n)$ *korrekt.*

Appendix

Aufgabe 4 I

Primimplikantentafel, 1te Regel

- da es nur darum geht, dass alle Minterme vom Anfang abgedeckt sind und die wesentlichen Minterme auf jeden Fall genommen werden müssen, werden auch die Minterme, die sie abdecken auf jeden Fall genommen. Aus diesem Grund ist es nicht mehr notwendig diese Minterme in der Primimplikantentafel miteinzubeziehen, da man Minterme nur in der Primimplikantentafel miteinbezieht, wenn man noch eine Abdeckung für diese Minterme sucht.

Aufgabe 4 I

Primimplikantentafel, 2te Regel

	010	100	110
~~$ab\overline{c}$~~			~~1~~
~~$a\overline{c}$~~		~~1~~	~~1~~
$a\overline{c}$		1	~~1~~

	010	100	110
$b\overline{c}$	1		~~1~~
~~$a\overline{c}$~~		~~1~~	~~1~~
$a\overline{c}$		1	~~1~~

- eines der Momome, das den Minterm abdeckt der dominiert wird, wird irgendwann aufgrund weiterer Reduktionen mit der ersten und dritten Reduktionsregel wesentlich und wird dann genommen. Sobald dieses Monom genommen wird ist es aufgrund der Teilmengenbeziehung sicher, dass dieses Monom auch den Minterm der den ersteren Minterm dominiert hat abdeckt. Der zweitere Minterm ist somit nicht mehr von Interesse für die Primimplikantentafel, weil man bereits eine Abdeckung für diesen Minterm gefunden hat und es nur darum geht, dass man alle Minterme vom Anfang abgedeckt hat.

Aufgabe 4 I

Primimplikantentafel, 3te Regel

- Wenn die On-Menge eines Monoms Teilmenge der On-Menge eines anderen Monoms ist, dann nimmt das Monom mit der größeren On-Menge, da größere ON-Menge weniger Literale bedeutet und somit weniger Kosten. Mit einer größeren On-Menge kann man mehr Minterme auf einmal abdecken und es geht nur darum, dass man am Ende alle Minterme vom Anfang irgendwie abgedeckt hat.

Aufgabe 4 I

Primimplikantentafel am Hypercube

	0000	0001	0100	0101	0110	0111	1000	1010	1100	1110	1111
0-0-	1	1	1	1							
–00	1		1				1		1		
01–			1	1	1	1					
-11-			1		1				1	1	
-1-0					1	1				1	1
1–0							1	1	1	1	

Aufgabe 4 II

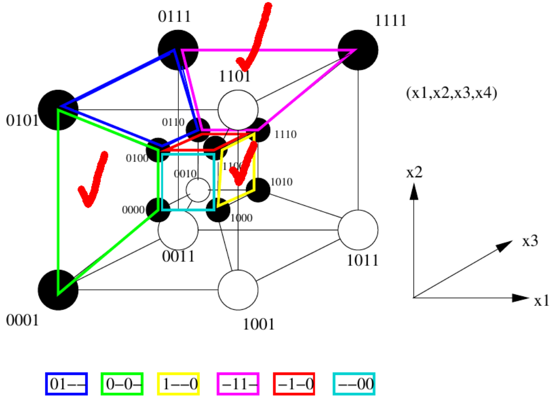
Primimplikantentafel am Hypercube

	0000	0001	0100	0101	0110	0111	1000	1010	1100	1110	1111
0-0-	1	1	1	1							
–00	1		1				1		1		
01–			1	1	1	1					
-11-			1		1				1	1	
-1-0					1	1				1	1
1–0							1	1	1	1	

- $(\overline{x_0}\,\overline{x_2}) \vee (x_1\overline{x_3}) \vee (x_0\overline{x_3})$, $\{0{-}0{-}, {-}1{-}0, 1{-}{-}0\}$

Aufgabe 4 III

Primimplikantentafel am Hypercube



Aufgabe 4 I

Binärbaum, wobei $n = 2^k$

- Anzahl Blätter (vollständiger Binärbaum): $n = 2^d = 2^3 = 8$
- Anzahl Knoten: $k = \sum_{i=0}^{d} 2^i = \frac{2^{d+1} - 1}{2 - 1} = 2^{d+1} - 1 = 2^{3+1} - 1 = 1 + 2 + 4 + 8 = 15$
- Tiefe mithilfe Anzahl Blätter: $d = log_2(n) = log_2(8) = 3$
- Tiefe mithilfe Anzahl Knoten (vollständiger Baum): $d = log_2(k+1) - 1 = log_2(15+1) - 1 = 4 - 1 = 3$
  - Herleitung: $2^{d+1} - 1 = k \Leftrightarrow 2^{d+1} = k + 1 \Leftrightarrow log_2(2^{d+1}) = log_2(k+1) \Leftrightarrow d + 1 = log_2(k+1) \Leftrightarrow d = log_2(k+1) - 1$
- Tiefe mithilfe Anzahl Knoten: $d = \lfloor(log_2(k_{real}))\rfloor = \lfloor log_2(10)\rfloor = 3$
  - $d = \lfloor(log_2(k_{real}))\rfloor = \lfloor log_2(8)\rfloor = 3$
  - $d = \lfloor(log_2(k_{real}))\rfloor = \lfloor log_2(7)\rfloor = 2$