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. Juni 2023

Universität Freiburg, Lehrstuhl für Rechnerarchitektur

Gliederung

Aufgabe 1

Aufgabe 2

Aufgabe 3

Aufgabe 1

Aufgabe 1 I

Primimplikantentafel

Aufgabe 2

Aufgabe 2 I

Methode von Petrick

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

  - $Prim(xor_3) =$
  - *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

- $\bigvee\limits_{\substack{(b_1,\ldots,b_n)\in\mathbb{B}^n \\ \text{mit } \sum_{i=1}^n \text{ ungerade}}} (x_1^{b_1} \wedge \ldots \wedge x_n^{b_n})$, wobei $x_i^0 := x_i'$ 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* $b_1 := xor_i(x_1, ..., x_i), \quad b_2 := xor_{n-i}(x_{i+1}, ..., x_n)$
  1. *Fall* $b_1 = 0,\ b_2 = 0$:

     $$\text{Dann ist} \sum_{j=1}^{i} x_j \text{ gerade}, \sum_{j=i+1}^{n} x_j \text{ gerade} \Rightarrow \sum_{j=1}^{n} x_j \text{ gerade} \Rightarrow g(0,0) = 0 = xor_2(0,0)$$
  2. *Fall* $b_1 = 1,\ b_2 = 0$:

     $$\text{Dann ist} \sum_{j=1}^{i} x_j \text{ ungerade}, \sum_{j=i+1}^{n} x_j \text{ gerade} \Rightarrow \sum_{j=1}^{n} x_j \text{ ungerade} \Rightarrow g(1,0) = 1 = xor_2(1,0)$$
  3. *Fall* $b_0 = 1,\ b_2 = 1$:

     $$\text{Dann ist} \sum_{j=1}^{i} x_j \text{ gerade}, \sum_{j=i+1}^{n} x_j \text{ ungerade} \Rightarrow \sum_{j=1}^{n} x_j \text{ ungerade} \Rightarrow g(0,1) = 1 = xor_2(0,1)$$
  4. *Fall* $b_0 = 1,\ b_2 = 1$:

     $$\text{Dann ist} \sum_{j=1}^{i} x_j \text{ ungerade}, \sum_{j=i+1}^{n} x_j \text{ ungerade} \Rightarrow \sum_{j=1}^{n} x_j \text{ gerade} \Rightarrow g(1,1) = 1 = xor_2(1,1)$$

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

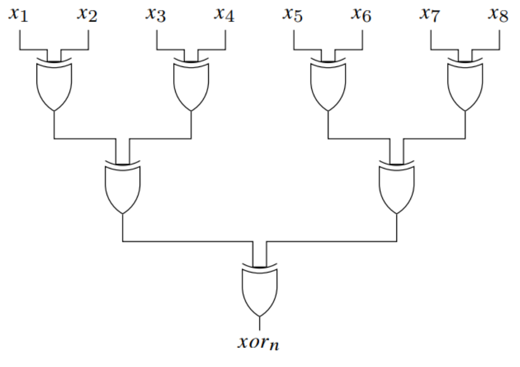
$\Rightarrow g = xor_2$, also $xor_n\left(x_1, \ldots, x_n\right) = xor_2\left(xor_i\left(x_1, \ldots, x_i\right), xor_{n-i}\left(x_{i+1}, \ldots, x_n\right)\right)$ für $1 \leq i \leq n-1$

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.*