UNIVERSITÄT LEIPZIG

Institut für Informatik

Abt. Technische Informatik

Dr. rer.nat. Hans-Joachim Lieske
Tel.: [49]-0341-97 32213
Zimmer: So 04-47
e-mail: lieske@informatik.uni-leipzig.de
www: http://www.informatik.uni-leipzig.de/~lieske

Studentenmitteilung

2. Semester - SS 2008

Montag, 16. Juni 2008

Aufgaben zu Übung
Grundlagen der Technischen Informatik 2

4. Aufgabenkomplex - 1. Aufgabe

Minimierung logischer Schaltungen mittels des Verfahrens von Quine-Mc-Cluskey

Gegeben ist die nebenstehende vollständige Funktionstabelle:

Aufgaben:

Minimieren Sie die Schaltung nach Quine-Mc-Cluskey.

1. Bestimmen Sie die Anzahl der Einsen für jeden Minterm
2. Bestimmen Sie die 1. "Quine'sche" Tabelle
3. Bestimmen Sie die 2. "Quine'sche" Tabelle
4. Lösen Sie das Überdeckungsproblem mittels der Überdeckungsfunktion $ü_f$

5. Minimieren Sie die Schaltung und bestimmen Sie die Lösungen Q_{D1}-min(Kosten=.....)=, Q_{D2}min(Kosten=.....)=, ... mit den geringsten Kosten
6. Zeichnen Sie den Schaltplan **einer** der minimierten Booleschen Funktionen mit den geringsten Kosten Q_1-min(Kosten=.....)=, Q_2-min(Kosten=.....)=, ... nach der Gleichung (streng)

Bemerkungen:
Im günstigsten Fall existiert nur eine Funktion mit minimalen Kosten, es können aber auch mehr sein.
Es sollen keine Reduktionsregeln für die 2. Quinesche Tabelle benutzt, sondern die Überdeckungsfunktion bestimmt werden.

Vollständige Funktionstabelle			
Nr.	**Eingangsvariablen** x_4, x_3, x_2, x_1, x_0	**Q**	**Anzahl Einsen**
0	**00000**	**1**	
1	**00001**	**1**	
2	**00010**		
3	**00011**	**1**	
4	**00100**	**1**	
5	**00101**		
6	**00110**	**1**	
7	**00111**	**1**	
8	**01000**	**1**	
9	**01001**	**1**	
10	**01010**	**1**	
11	**01011**	**1**	
12	**01100**		
13	**01101**		
14	**01110**		
15	**01111**		
16	**10000**	**1**	
17	**10001**		
18	**10010**	**1**	
19	**10011**		
20	**10100**		
21	**10101**		
22	**10110**		
23	**10111**		
24	**11000**	**1**	
25	**11001**	**1**	
26	**11010**	**1**	
27	**11011**	**1**	
28	**11100**	**1**	
29	**11101**		
30	**11110**	**1**	
31	**11111**		

Hilfen:

Vollständige Funktionstabelle			
Nr.	**Eingangsvariablen** x_4, x_3, x_2, x_1, x_0	**Q**	**Anzahl Einsen**
0	**00000**		
1	**00001**		
2	**00010**		
3	**00011**		
4	**00100**		
5	**00101**		
6	**00110**		
7	**00111**		
8	**01000**		
9	**01001**		
10	**01010**		
11	**01011**		
12	**01100**		
13	**01101**		
14	**01110**		
15	**01111**		
16	**10000**		
17	**10001**		
18	**10010**		
19	**10011**		
20	**10100**		
21	**10101**		
22	**10110**		
23	**10111**		
24	**11000**		
25	**11001**		
26	**11010**		
27	**11011**		
28	**11100**		
29	**11101**		
30	**11110**		
31	**11111**		

1. "Quine'sche" Tabelle (1.Teil)					
0. Ordnung					
Nr.	$x_4x_3x_2x_1x_0$	Primim plikant	Nr.	$x_4x_3x_2x_1x_0$	Primim plikant

Wenn der Term der mit den Termen der niederen- und der höheren Gruppe nicht vereinfacht werden kann, dann Primimplikant

1. "Quine'sche" Tabelle (2.Teil)					
1. Ordnung					
Nr.	$x_4x_3x_2x_1x_0$	Primim plikant	Nr.	$x_4x_3x_2x_1x_0$	Primim plikant

Wenn der Term der mit den Termen der niederen- und der höheren Gruppe nicht vereinfacht werden kann, dann Primimplikant

1. "Quine'sche" Tabelle (3.Teil)					
2. Ordnung					
Nr.	$x_4x_3x_2x_1x_0$	Primim plikant	Nr.	$x_4x_3x_2x_1x_0$	Primim plikant

Wenn der Term der mit den Termen der niederen- und der höheren Gruppe nicht vereinfacht werden kann, dann Primimplikant

1. "Quine'sche" Tabelle (4.Teil)					
3. Ordnung					
Nr.	$x_4x_3x_2x_1x_0$	Primim-Pli-kant	Nr.	$x_4x_3x_2x_1x_0$	Primim-Pli-kant

Wenn der Term der mit den Termen der niederen- und der höheren Gruppe nicht vereinfacht werden kann, dann Primimplikant

1. "Quine'sche" Tabelle (5.Teil)		
4. Ordnung		
Nr.	$x_4x_3x_2x_1x_0$	**Primim -plikant**

Wenn der Term der mit den Termen der niederen- und der höheren Gruppe nicht vereinfacht werden kann, dann Primimplikant

1. "Quine'sche" Tabelle			
Übersicht Primimplikanten			
Nr.	**Funktion**	$x_4x_3x_2x_1x_0$	**Primim -plikant**

Wenn der Term der mit den Termen der niederen- und der höheren Gruppe nicht vereinfacht werden kann, dann Primimplikant

2. "Quine'sche" Tabelle																								
	Minterme																							**Kosten**
Prim-implkant																								

Bemerkung:

Sind zwischen den Variablen keine Operatoren, so ist das als UND-Verknüpfung zu lesen.
Beispiel: $abc \equiv a \wedge b \wedge c$

Für bestimmte Fälle wird x_0 mit $2^0=1$, x_1 mit $2^1=2$, x_2 mit $2^2=4$ und später x_3 mit $2^3=8$ u.s.w. gewichtet, so das man sie als eine Zahl ansehen kann.
Bei den Schaltungen können die Gatter beliebig viele Eingänge haben, ausgenommen der Inverter. Es sind, wenn nicht ausdrücklich anders gefordert, nur AND-, OR- und NOT-Gatter zu verwenden.
Leere Felder in Karnaugh-Veitch-Diagrammen sind immer null.
Bei den Konversionen sind Inverter als Spezialfall der NAND- und NOR - Gatter auf der untersten Ebene erlaubt. Die Konversionen sind, wenn nicht anders angegeben, aus den kanonischen Normalformen zu erstellen.
Streng in Zusammenhang mit der Schaltung bedeutet, daß alle Inverter gezeichnet werden müssen! Es existiert jeweils nur ein Draht für die nicht invertierten Variablen.
Zum Beispiel gilt für die Implikanten 1. Ordnung (1,5) und (2,6) I(1)={(1,5),(2,6)}
2. Ordnung (4,5,6,7) I(2)={(4,5,6,7)}. Für die Primimplikanten z.B: PI(1)= und die Kernimplikanten z.B: KPI(1)= . Entsprechend gilt für Implikate Ik, Primimplikate PIk und Kernprimimplikate KPIk.
Bei der Baumdarstellung geht man zweckmäßiger Weise von der kanonisch disjunktiven Normalform oder einer disjunktiven Form aus.
Die Kosten sind entsprechend der Kostenbestimmung im Quine-McCluskey Verfahren aus der Vorlesung zu berechnen. Für n-Variablen hat der (Prim)implikant 0. Ordnung (Minterm) die Kosten n, der (Prim)implikant 1. Ordnung (2er Block) die Kosten n-1 usw.

Analog gilt es auch für die (Prim)implikate

Es kann mehrere minimale Funktionen mit minimalen Kosten geben.