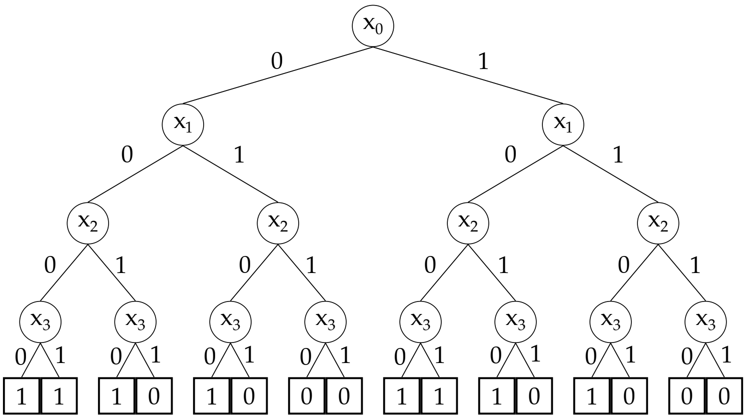
UNIVERSITÄT LEIPZIG

Grundlagen der Technischen Informatik 2
Sommersemester 25

Übungsblatt 3

Aufgabe 1: Binäre Entscheidungsdiagramme

Gegeben sei die Funktion f durch das folgende geordnete binäre Entscheidungsdiagramm (OBDD):



1. Reduzieren Sie den OBDD so weit wie möglich und zeichnen Sie den rOBDD. Geben Sie bei jedem Schritt die angewandte Regel an.

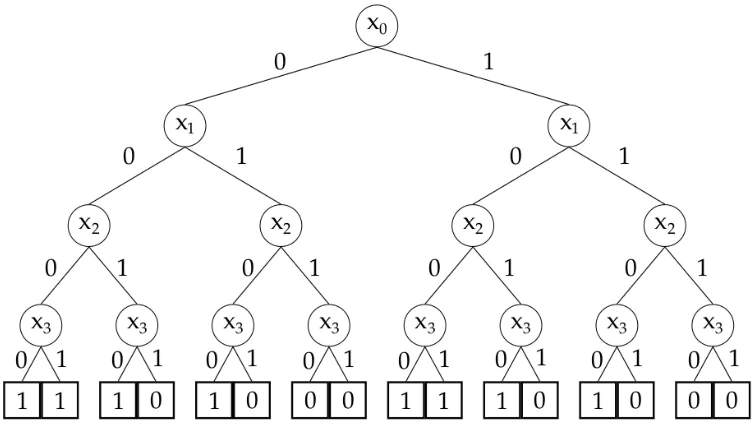
   Regel 1: Eliminierung von Knoten mit gleichen Nachfolgern.
   Regel 2: Gemeinsame Nutzung gleicher Teilbäume.

2. Leiten Sie aus dem rOBDD die Funktion f in disjunktiver Form ab.

3. Die minimierte Funktion von f lautet:

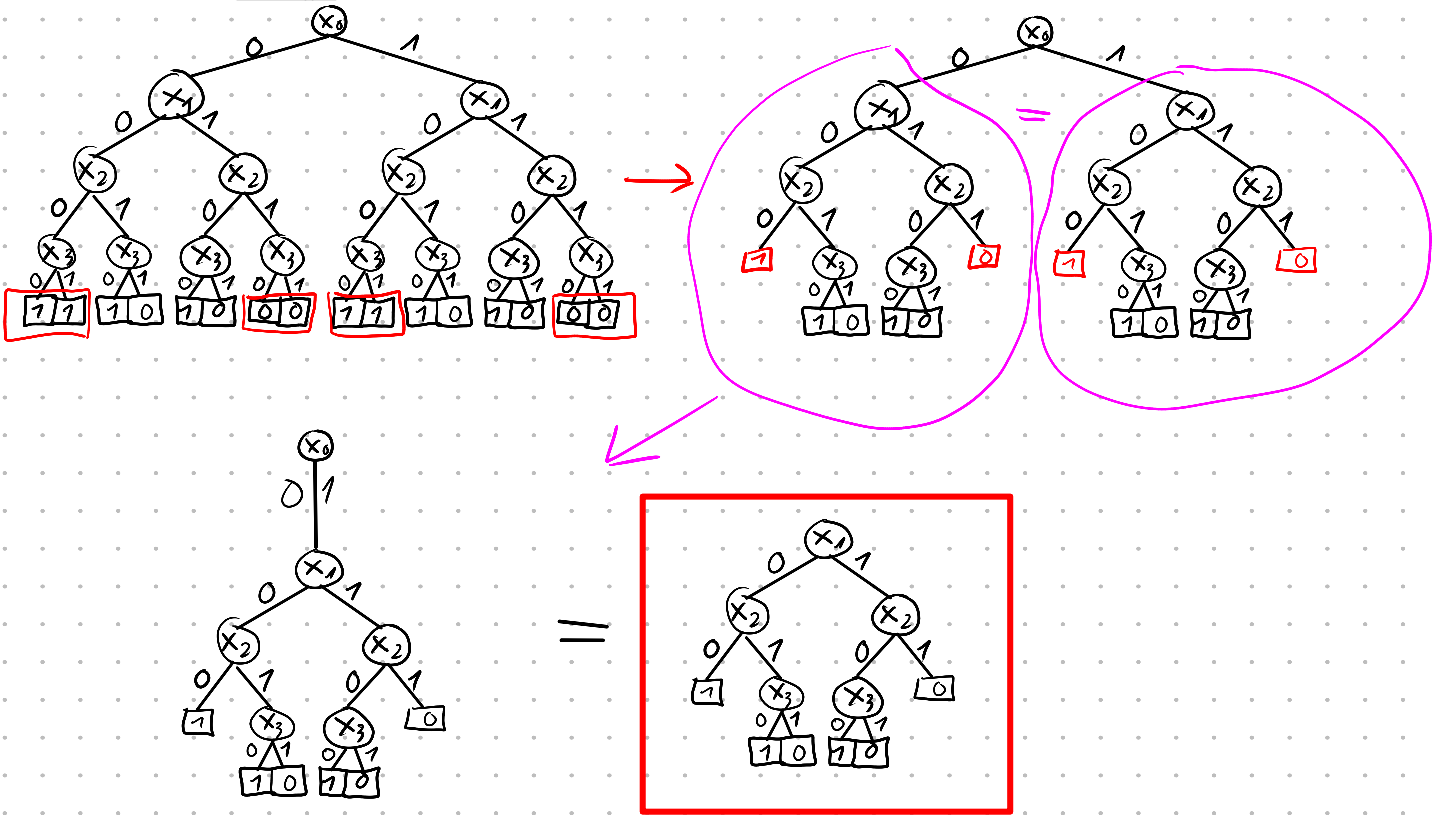
$$f_{min} = \overline{x_2}\,\overline{x_1} \vee \overline{x_3}\,\overline{x_1} \vee \overline{x_3}\,\overline{x_2}$$

   Ist es somit möglich die Reduzierung des OBDDs als Minimierungsverfahren zu nutzen? Begründen Sie Ihre Entscheidung.



1. Reduzieren Sie den OBDD so weit wie möglich und zeichnen Sie den rOBDD. Geben Sie bei jedem Schritt die angewandte Regel an.

   Regel 1: Eliminierung von Knoten mit gleichen Nachfolgern.
   Regel 2: Gemeinsame Nutzung gleicher Teilbäume.



2. Leiten Sie aus dem rOBDD die Funktion f in disjunktiver Form ab.

Aufgabe 2: Maschinenzahlen

1. Wandeln Sie die folgenden Binärzahlen in Dezimalzahlen um.

   (a) 10001_2 $16 + 1 = 17_{10}$ (b) 1010111_2 $64+16+4+2+1 = 87$

2. Wandeln Sie die folgenden Dezimalzahlen in Binärzahlen um.

   (a) 144_{10} (b) 413_{10}

3. Gegeben sei die Hexadezimalzahl $18A32D_{16}$. Wandeln Sie diese in eine Binärzahl um.
4. Gegeben sei die Binärzahl 11010101001001001111_2. Wandeln Sie diese in eine Hexadezimalzahl um.
5. Berechnen Sie das Zweierkomplement der folgenden 8-Bit Integer.

   (a) `0x00001100` (b) `0x11111100`

6. Wandeln Sie die folgenden Dezimalzahlen in IEEE754 16-bit half-precision floating-point Zahlen um.

   (a) 10000_{10} (b) 16.16_{10}

7. Wandeln Sie die folgende IEEE754 32-bit floating-point Zahlen in eine Dezimalzahl um.

   (a) `0 10001000 11110011100000000000000`

Aufgabe 3: Schaltnetze

1. Seien $A =$`0b00010101` und $B =$`0b00111011` als zwei signed 8-bit Integer gegeben.
   (a) Berechnen Sie $A + B$. Führen Sie dafür binäre Addition durch. (Das Ergebnis soll ebenfalls ein signed 8-bit Integer sein.)
   (b) Berechnen Sie $B - A$. Führen Sie dafür binäre Subtraktion durch. (Das Ergebnis soll ebenfalls ein signed 8-bit Integer sein.)
2. Konstruieren Sie analog zum Adder (siehe Vorlesung) einen Subtracter. (Ein Schaltnetz, welches die binäre Subtraktion durchführen kann.)
   (a) Entwerfen Sie einen Half-Subtracter. (Eine Schaltung, die zwei Bits subtrahieren kann.)
   (b) Erweitern Sie diese Schaltung zu einem Full-Subtracter. (Eine Schaltung, die drei Bits subtrahieren kann.)
   (c) Wie kann eine solche Schaltung auf 8 Bit erweitert werden? Beschreiben Sie das theoretische Vorgehen.
3. Entwerfen Sie analog zum Multiplexer (siehe Vorlesung) eine Schaltung, welche einen Input e, abhängig vom Steuersignal s_0 an Output o_0 oder Output o_1 weiterleitet.

2. Wandeln Sie die folgenden Dezimalzahlen in Binärzahlen um.

(a) 144_{10} (b) 413_{10}

a)
144/2 = 77 R0
77/2 = 36 R1
36/2 = 18 R0
18/2 = 9 R0
9/2 = 4 R1
4/2 = 2 R0
2/2 = 1 R0
1/2 = 0 R1

$= 10010010_2$

b) $413_{10} =$

413/2 = 206 R1
206/2 = 103 R0
103/2 = 51 R1
51/2 = 25 R1
25/2 = 12 R1
12/2 = 6 R0
6/2 = 3 R0
3/2 = 1 R1
1/2 = 0 R1

110011101_2

3. Gegeben sei die Hexadezimalzahl $18A32D_{16}$. Wandeln Sie diese in eine Binärzahl um.

$18A32D_{16}$ = 0001 1000 1010 0011 0010 1101

0001
1000
1010
0011
0010
1101

4. Gegeben sei die Binärzahl $1101010100100 1001111_2$. Wandeln Sie diese in eine Hexadezimalzahl um.

D 5 2 4 F

$D524F_{16}$

5. Berechnen Sie das Zweierkomplement der folgenden 8-Bit Integer.

(a) 00001100 = 12 (b) 11111100 = −4

(a) b: „+", 12

+12

(b) b: „−" → Zahl invertieren +1

1111100 → (Inv) 0000011

→ (+1) 0000100

= 4

= −4

6. Wandeln Sie die folgenden Dezimalzahlen in IEEE754 16-bit half-precision floating-point Zahlen um.

(a) 10000_{10} (b) 16.16_{10}

1 (Vz), 5 (exp), 10 (Mantisse)

a) $10000_{10} \triangleq 10011100010000$

Die „1" am Anfang kann weggelassen werden

1,0011100010000,... (13 stellen)

Mantisse 10 stellen ⇒ 3 stellen verlieren

1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 000

Immer ne 1 also egal

Passt nicht rein

bias „b" bei 16 float = 15

e = komma verschoben = 13

exp = e + b

= 13 + 15

$= 28_{10} = 11100_2$ 11100

Vz + 0

0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0

b) $16,16_{10}$

$+$ exp

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

$16_{10} \hat{=} 10000_2$

$0,16_{10} \hat{=} 0,001010...$

$16,16_{10} = 1\underbrace{0000,}_{4\ \text{Stelle(n)}}001010...$

$0,16 \cdot 2 = 0,32$ R 0

$0,32 \cdot 2 = 0,64$ R 0

$0,64 \cdot 2 = 1,28$ R 1

$0,28 \cdot 2 = 0,56$ R 0

$0,56 \cdot 2 = 1,12$ R 1

$0,12 \cdot 2 = 0,24$ R 0

$0,24 \cdot 2 = 0,48$ R 0

$0,48 \cdot 2 = 0,96$ R 0

⋮

00101000..

$4_{10} = e$

$b = 15$

$Exp = e + b$

$= 4 + 15$

$= 19 \hat{=} 10011$

7. Wandeln Sie die folgende IEEE754 32-bit floating-point Zahlen in eine Dezimalzahl um.

(a) 0 $\overbrace{10001000}^{7}$ 11110011100000000000000

0000000000000000

$$b = 2^{(|exp|-1)} - 1 = 2^{7-1} - 1 = 2^{6} - 1 = 128 - 1 = 127$$

$0 = +$

$10001000_2 = 136$
$128 + 8$

$136 - 127 = \boxed{9}$

$1,11110011100000000000000 \cdot 2^9$

$1111100111,0...$
512 128 32
64 2 1
256 4

$=$

512
+256
+128
+64
+32
+4
+2
+1
$+999_{10}$

Aufgabe 2: Maschinenzahlen

1. Wandeln Sie die folgenden Binärzahlen in Dezimalzahlen um.

   (a) 10001_2 (b) 1010111_2

   $16 + 1 = 17_{10}$ $64 + 16 + 4 + 2 + 1 = 87$

2. Wandeln Sie die folgenden Dezimalzahlen in Binärzahlen um.

   (a) 144_{10} (b) 413_{10}

3. Gegeben sei die Hexadezimalzahl $18A32D_{16}$. Wandeln Sie diese in eine Binärzahl um.
4. Gegeben sei die Binärzahl 11010101001001001111_2. Wandeln Sie diese in eine Hexadezimalzahl um.
5. Berechnen Sie das Zweierkomplement der folgenden 8-Bit Integer.

   (a) `0x00001100` (b) `0x11111100`

6. Wandeln Sie die folgenden Dezimalzahlen in IEEE754 16-bit half-precision floating-point Zahlen um.

   (a) 10000_{10} (b) 16.16_{10}

7. Wandeln Sie die folgende IEEE754 32-bit floating-point Zahlen in eine Dezimalzahl um.

   (a) `0 10001000 11110011100000000000000`

Aufgabe 3: Schaltnetze

1. Seien $A =$`0b00010101` und $B =$`0b00111011` als zwei signed 8-bit Integer gegeben.
   (a) Berechnen Sie $A + B$. Führen Sie dafür binäre Addition durch. (Das Ergebnis soll ebenfalls ein signed 8-bit Integer sein.)
   (b) Berechnen Sie $B - A$. Führen Sie dafür binäre Subtraktion durch. (Das Ergebnis soll ebenfalls ein signed 8-bit Integer sein.)
2. Konstruieren Sie analog zum Adder (siehe Vorlesung) einen Subtracter. (Ein Schaltnetz, welches die binäre Subtraktion durchführen kann.)
   (a) Entwerfen Sie einen Half-Subtracter. (Eine Schaltung, die zwei Bits subtrahieren kann.)
   (b) Erweitern Sie diese Schaltung zu einem Full-Subtracter. (Eine Schaltung, die drei Bits subtrahieren kann.)
   (c) Wie kann eine solche Schaltung auf 8 Bit erweitert werden? Beschreiben Sie das theoretische Vorgehen.
3. Entwerfen Sie analog zum Multiplexer (siehe Vorlesung) eine Schaltung, welche einen Input e, abhängig vom Steuersignal s_0 an Output o_0 oder Output o_1 weiterleitet.

1. Seien $A =$ `0b00010101` und $B =$ `0b00111011` als zwei signed 8-bit Integer gegeben.
   (a) Berechnen Sie $A + B$. Führen Sie dafür binäre Addition durch. (Das Ergebnis soll ebenfalls ein signed 8-bit Integer sein.)
   (b) Berechnen Sie $B - A$. Führen Sie dafür binäre Subtraktion durch. (Das Ergebnis soll ebenfalls ein signed 8-bit Integer sein.)

a) $A = 10101$

$B = 111011$

$$A + B = \begin{array}{r} \scriptstyle 1111\ \ \\ 10101 \\ +\ 111011 \\ \scriptstyle 1\qquad\quad \\ \hline 1010000 \end{array}$$

b) $B - A = B + (\neg A + 1)$

$\neg A = 01010$

$\neg A + 1 = 1011$

$$B - A = \begin{array}{r} 111011 \\ -\ 10101 \\ \scriptstyle 1\quad\ \ \\ \hline 100110 \end{array}$$

✓

$\neq$

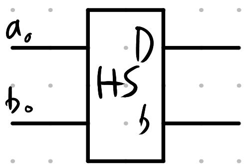
$$B + (\neg A + 1) = \begin{array}{r} 111011 \\ +\ \ 1011 \\ \scriptstyle 11\ \ 11\ \ \\ \hline 1000110 \end{array}$$

X ?

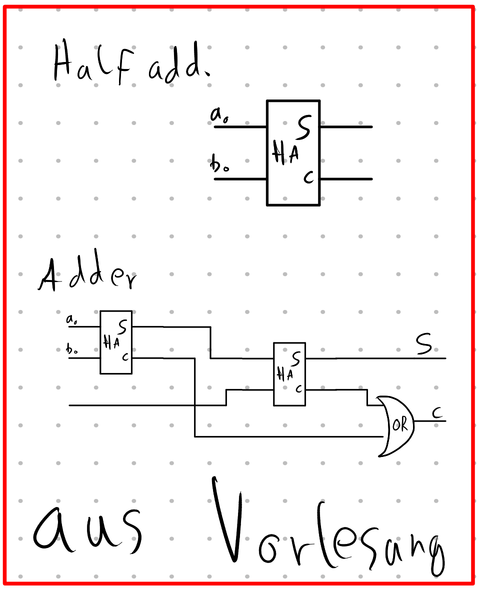
?

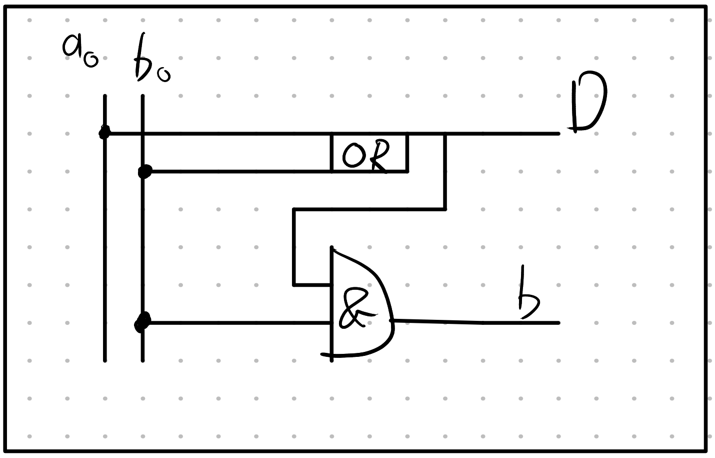
2. Konstruieren Sie analog zum Adder (siehe Vorlesung) einen Subtracter. (Ein Schaltnetz, welches die binäre Subtraktion durchführen kann.)
   (a) Entwerfen Sie einen Half-Subtracter. (Eine Schaltung, die zwei Bits subtrahieren kann.)
   (b) Erweitern Sie diese Schaltung zu einem Full-Subtracter. (Eine Schaltung, die drei Bits subtrahieren kann.)
   (c) Wie kann eine solche Schaltung auf 8 Bit erweitert werden? Beschreiben Sie das theoretische Vorgehen.

a) Half subtr.

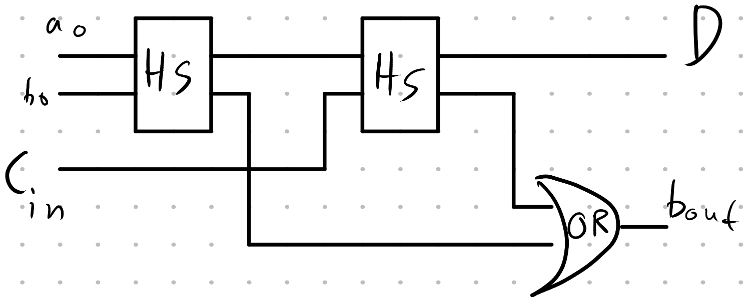


a_0	b_0	D	b
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



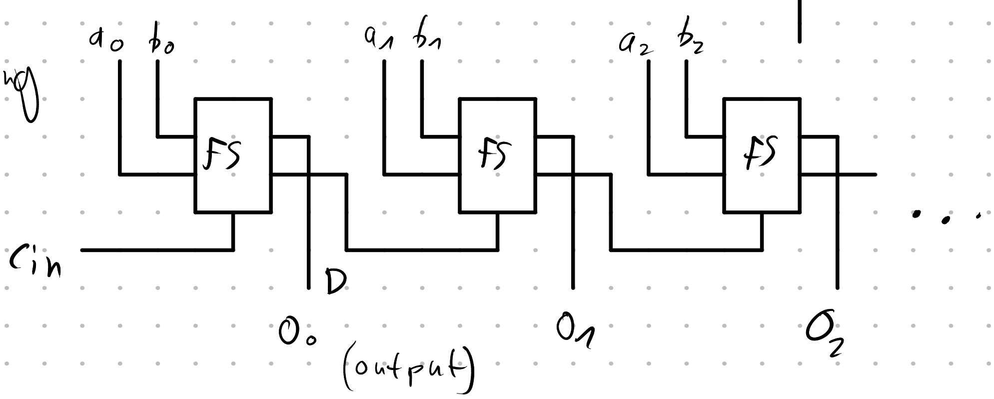


b) Subtractor



a_0	b_0	c_{in}	D	b_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Skalierung



3. Entwerfen Sie analog zum Multiplexer (siehe Vorlesung) eine Schaltung, welche einen Input e, abhängig vom Steuersignal s_0 an Output o_0 oder Output o_1 weiterleitet.