



Prof. Dr. Rolf Drechsler, drechsler@informatik.uni-bremen.de, MZH 3510
Dr. Robert Wille, rwille@informatik.uni-bremen.de, MZH 3485
Oliver Keszöcze, keszocze@informatik.uni-bremen.de, MZH 3440

10. Übungsblatt zur Vorlesung Technische Informatik 1

Aufgabe 1

(5+2+1 Punkte)

Ein *1-Bit Shifter* ist ein Schaltkreis, der die folgende Boolesche Funktion berechnet:

$\text{shift} : \mathbb{B}^{n+2} \rightarrow \mathbb{B}^n$ mit

$$\text{shift}(a_{n-1}, \dots, a_0, s_l, s_r) = \begin{cases} (a_{n-2}, \dots, a_0, 0), & \text{falls } s_l = 1 \wedge s_r = 0 \\ (0, a_{n-1}, \dots, a_1), & \text{falls } s_l = 0 \wedge s_r = 1 \\ (a_{n-1}, \dots, a_0) & \text{sonst} \end{cases}$$

- a) Konstruiere einen kombinatorischen Schaltkreis, der *shift* realisiert. Verwende dabei Schaltelemente aus $\text{STD} \cup \text{MUX}_i, i \in \mathbb{N}$.

Hinweis: Betrachte zunächst einen *1-Bit Shifter* für $n = 4$.

- b) Gib eine Abschätzung der Schaltungstiefe und der Kosten der in a) konstruierten Schaltung an.
c) Wie kann die Schaltung angepasst werden, damit sie einen *k-Bit Shifter* ($k \in \mathbb{N}$) realisiert?

Aufgabe 2

(4+2 Punkte)

In der Vorlesung wurde eine Realisierung der *n-Bit ALU* vorgestellt, wobei die Einzelfunktionen getrennt dargestellt werden, wie es auch in Abbildung 1 zu sehen ist.

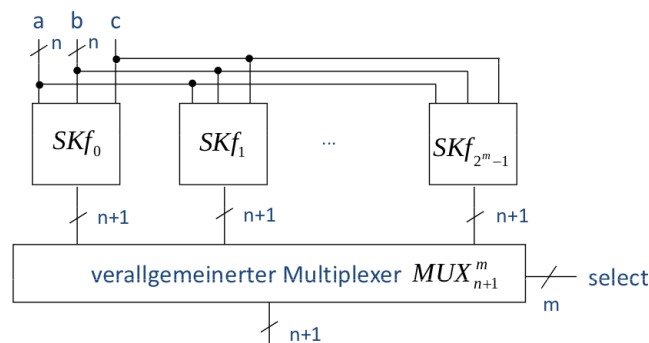


Abbildung 1: Realisierung der *n-Bit ALU* (Einzelfunktionen getrennt)

- a) Gib eine detaillierte Konstruktion des verallgemeinerten Multiplexers MUX_{n+1}^m auf der Basis von Multiplexer-Gattern an.
b) Gib eine Abschätzung der Schaltungstiefe und der Kosten der in a) konstruierten Schaltung an.

Aufgabe 3

(1+3+1 Punkte)

Betrachte die Boolesche Funktion $f = x_1x_2 + x_3 + \bar{x}_4$.

- Konstruiere direkt aus der Funktionsbeschreibung eine Schaltung, welche f realisiert. Verwende dabei ausschließlich Schaltelemente aus STD.
- Konstruiere zu f das reduzierte OBDD mit Variablenordnung $x_1 < x_2 < x_3 < x_4$. Erzeuge mit Hilfe dieses Entscheidungsdiagramms eine Schaltung, welche f realisiert und dabei ausschließlich Schaltelemente aus MUX₁ enthält.
- Gib die Schaltungstiefe und die Kosten der in a) und b) konstruierten Schaltungen an.

Aufgabe 4

(1 Punkte)

Was leistet die *Symbolische Simulation* eines kombinatorischen Schaltkreises? Wo liegen die Vorteile gegenüber der „einfachen“ Schaltkreissimulation?

Zusatzaufgabe

(8 Punkte)

Ein *Komparator* ist ein Logikbaustein, der die numerischen Vergleichsrelationen $<$, $>$ und $=$ implementiert. Der Schaltkreis bekommt als Eingabe zwei positive binäre Zahlen x und y der Länge n über die Eingänge (x_{n-1}, \dots, x_0) und (y_{n-1}, \dots, y_0) und berechnet daraus die Wertebelegung des Ausgangs z mit $z_2 = (x > y)$, $z_1 = (x = y)$ und $z_0 = (x < y)$. Je nachdem ob x größer, gleich oder kleiner y ist, nimmt genau eine der Ausgangsleitungen (z_2, z_1, z_0) den Wert 1 an. Ein Komparator berechnet also die folgende Boolesche Funktion:

$\text{comp}_n : \mathbb{B}^{2n} \rightarrow \mathbb{B}^3$ mit

$$\text{comp}_n(x_{n-1}, \dots, x_0, y_{n-1}, \dots, y_0) = \begin{cases} (1, 0, 0), & \text{falls } (x_{n-1} > y_{n-1}) \\ & \vee (x_{n-1} = y_{n-1}) \wedge (x_{n-2} > y_{n-2}) \\ & \vdots \\ & \vee \\ & \vdots \\ & \vee (x_{n-1} = y_{n-1}) \wedge \dots \wedge (x_1 = y_1) \wedge (x_0 > y_0) \\ (0, 1, 0), & \text{falls } \forall_i : x_i = y_i \\ (0, 0, 1), & \text{falls } (x_{n-1} < y_{n-1}) \\ & \vee (x_{n-1} = y_{n-1}) \wedge (x_{n-2} < y_{n-2}) \\ & \vdots \\ & \vee \\ & \vdots \\ & \vee (x_{n-1} = y_{n-1}) \wedge \dots \wedge (x_1 = y_1) \wedge (x_0 < y_0) \end{cases}$$

- Gib einen kombinatorischen Schaltkreis an, der comp_n implementiert, wobei die Zahlen x und y als vorzeichenlose Zahlen interpretiert werden sollen.
Hinweis: Betrachte zunächst eine Implementierung von comp_1 .
- Gib eine Abschätzung der Schaltungstiefe (Laufzeit) und der Kosten der in a) konstruierten Schaltung an.
- Erweitere die Schaltung aus a) zu einer Implementierung von comp_n für Zahlen im Zweierkomplement.

Abgabetermin: vor Beginn der Vorlesung am **20. Juni 2013**