



Prof. Dr. Rolf Drechsler, drechsler@informatik.uni-bremen.de, MZH 3510
Dr. Nicole Drechsler nd@informatik.uni-bremen.de, MZH 3485

10. Übungsblatt zur Vorlesung

Technische Informatik 1

Aufgabe 1

(4+2+2 Punkte)

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

shift : $\mathbf{B}^{n+2} \rightarrow \mathbf{B}^n$ 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 $\text{STD} \cup \text{MUX}_i, i \in \mathbf{N}$.

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

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

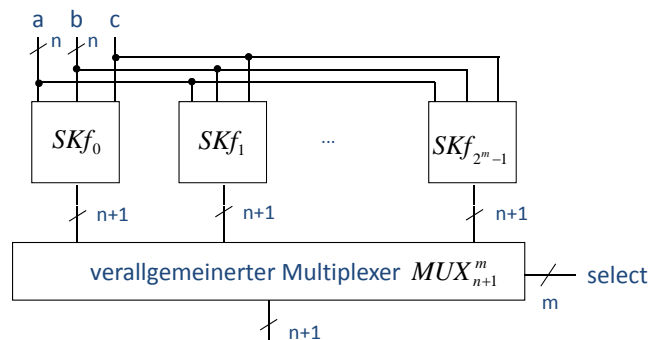


Abbildung 1: Realisierung der *n-Bit ALU* (Einzelfunktionen getrennt)

- 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 (Laufzeit) und des Platzbedarfs der in a) konstruierten Schaltung an.

Aufgabe 3

(4 Punkte)

Gib einen kombinatorischen Schaltkreis an, der zu einer positiven binären Zahl $a = (a_{n-1}, \dots, a_0)$ die Zahl $b = (b_{n-1}, \dots, b_0)$ mit $[b] = [a]/2$ berechnet. Betrachte hier eine ganzzahlige Division mit Rest.

Aufgabe 4

(2 Punkte)

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

Zusatzaufgabe

(4+2+2 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:

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

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

- a) 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 auf der Basis von STD.
- b) Gib eine Abschätzung der Schaltungstiefe (Laufzeit) und des Platzbedarfs der in a) konstruierten Schaltung an.
- c) Erweitere die Schaltung aus a) zu einer Implementierung von comp_n für Zahlen im Zweierkomplement.

Abgabetermin: zu Beginn der Vorlesung am 25.06.2009