7.2 Schaltnetze für grundlegende Operationen

Natürlich wird bei Komplexitätsbestimmungen nicht jeder Algorithmus bis auf Schaltnetzebene heruntergebrochen. Statt dessen bestimmt man ein- für allemal die Schaltnetz-Komplexität von (z. B. arithmetischen) Grundoperationen und Basisalgorithmen und führt kompliziertere Algorithmen darauf zurück.

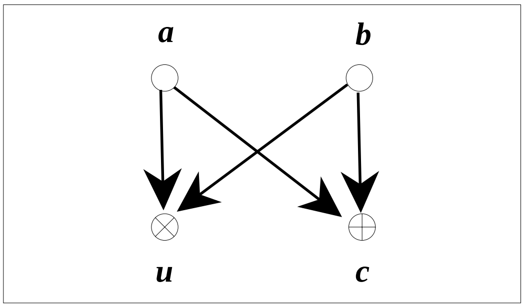

Abbildung 1: Schaltnetz zur Addition zweier Einbit-Zahlen

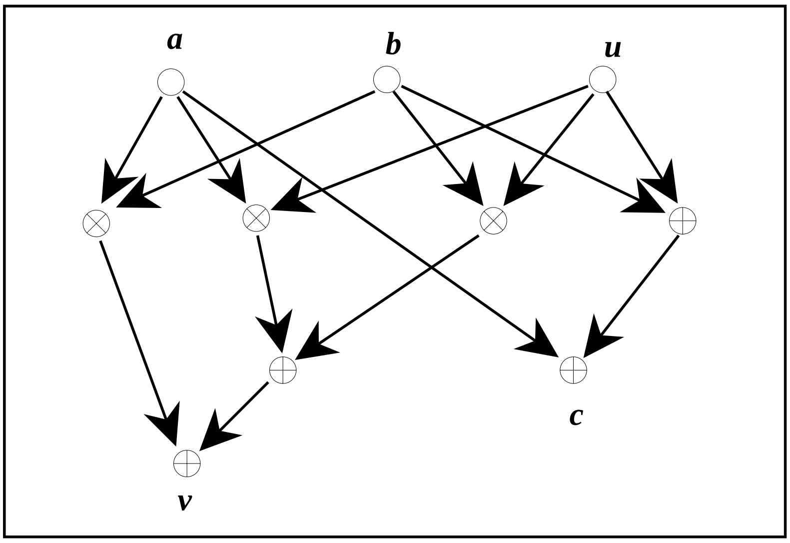

Abbildung 2: Schaltnetz zur Addition dreier Einbit-Zahlen

1. Die Addition zweier Bits a, b mit $\bmod 2$-Summe c und Übertrag u, also die „primitive" Addition in $\mathbb{N}$ von Ziffern zur Basis 2, ist eine Funktion $C : \mathbb{F}_2^2 \longrightarrow \mathbb{F}_2^2$. Ein passendes Schaltnetz ist in Abbildung 1 wiedergegeben.

2. Die analoge Addition dreier Bits a, b, u mit mod 2-Summe c und Übertrag v ist der Grundbaustein, mit dem die Addition beliebiger Ganzzahlen zur Basis 2 aufgebaut wird. Sie wird durch eine Funktion $C : \mathbb{F}_2^3 \longrightarrow \mathbb{F}_2^2$ mit dem Schaltnetz aus Abbildung 2 beschrieben. Eine Formel für v ist nämlich:

$$v = (a \otimes b) \oplus (a \otimes u) \oplus (b \otimes u) = \begin{cases} 1, & \text{wenn mindestens 2 Eingaben 1 sind,} \\ 0 & \text{sonst.} \end{cases}$$

3. Die Addition einer Einbit-Zahl zu einer s-Bit-Zahl verläuft nach einem ähnlichen, umfangreicheren Schema und lässt sich durch ein Schaltnetz mit $3s + 1$ Knoten, davon $s + 1$ Ausgängen, erledigen.

4. Die Multiplikation zweier Bits ist trivial. Interessanter ist es, die Multiplikation zweier Zweibit-Zahlen $2a_1 + a_0$ und $2b_1 + b_0$ mit Vierbit-Ergebnis $8c_3 + 4c_2 + 2c_1 + c_0$ durch ein Schaltnetz zu beschreiben. Der klassische Algorithmus ergibt die Formeln

$$\begin{array}{llll} c_0 = a_0 \otimes b_0, & & & \\ t_1 = a_0 \otimes b_1, & t_2 = a_1 \otimes b_0, & c_1 = t_2 \oplus t_1, & u_1 = t_2 \otimes t_1, \\ t_3 = a_1 \otimes b_1, & c_2 = t_3 \oplus u_1, & c_3 = t_3 \otimes u_1 & \end{array}$$

mit Hilfsbits t_1, t_2, t_3 und u_1. Sie lassen sich in das Schaltnetz aus Abbildung 3 übersetzen.

5. Auch logische Operationen und Verzweigungen kann man auf Additionen und Multiplikationen in $\mathbb{F}_2$ reduzieren. Für die logischen Operationen sieht man das an der Tabelle 1. Eine Verzweigung wird etwa so beschrieben: Gegeben seien drei Schaltnetze $C_0, C_1 : \mathbb{F}_2^r \longrightarrow \mathbb{F}_2^s$ und $E : \mathbb{F}_2^r \longrightarrow \mathbb{F}_2$. Gesucht ist ein Schaltnetz $C : \mathbb{F}_2^r \longrightarrow \mathbb{F}_2^s$ mit

$$C(x) = \begin{cases} C_0(x), & \text{falls } E(x) = 0, \\ C_1(x), & \text{falls } E(x) = 1. \end{cases}$$

Dieses konstruiert man mit Hilfe der Formeln

$$y \otimes v = \begin{cases} 0, & \text{falls } y = 0, \\ v, & \text{falls } y = 1. \end{cases}$$

$$(\neg y) \otimes u = \begin{cases} u, & \text{falls } y = 0, \\ 0, & \text{falls } y = 1. \end{cases}$$

$$[y \otimes v] \oplus [(1 \oplus y) \otimes u] = \begin{cases} u, & \text{falls } y = 0, \\ v, & \text{falls } y = 1. \end{cases}$$

Setzt man hier C_0, C_1 und E ein, so erhält man als Lösung

$$C(x) = [E(x) \otimes C_1(x)] \oplus [(1 \oplus E(x)) \otimes C_0(x)].$$

Dieses Schaltnetz ist in Abbildung 4 dargestellt. Es hat auch einen konstanten Eingang.

6. Der Vergleich $x \geq y$ zweier Bits ist eine der 16 zweistelligen Bitoperationen, nämlich $1 \oplus y \oplus x \otimes y$. Allgemeiner lässt sich ein Vergleich von $n+1$-Bit-Zahlen $x = (x_n \dots x_0)$ und $y = (y_n \dots y_0)$ so zusammenbasteln:

$$C(x,y) = \begin{cases} 1, & \text{wenn } x_n > y_n \\ & \text{oder } x_n = y_n \text{ und } x' \geq y', \\ 0, & \text{wenn } x_n = y_n \text{ und } x' < y' \\ & \text{oder } x_n < y_n \end{cases}$$

mit $x' = (x_{n-1} \dots x_0)$ und $y' = (y_{n-1} \dots y_0)$.

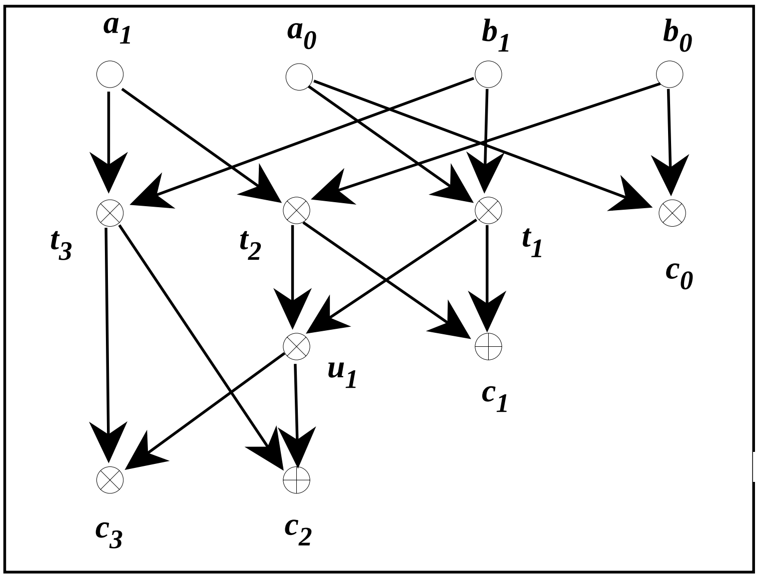

Abbildung 3: Schaltnetz zur Multiplikation zweier Zweibit-Zahlen

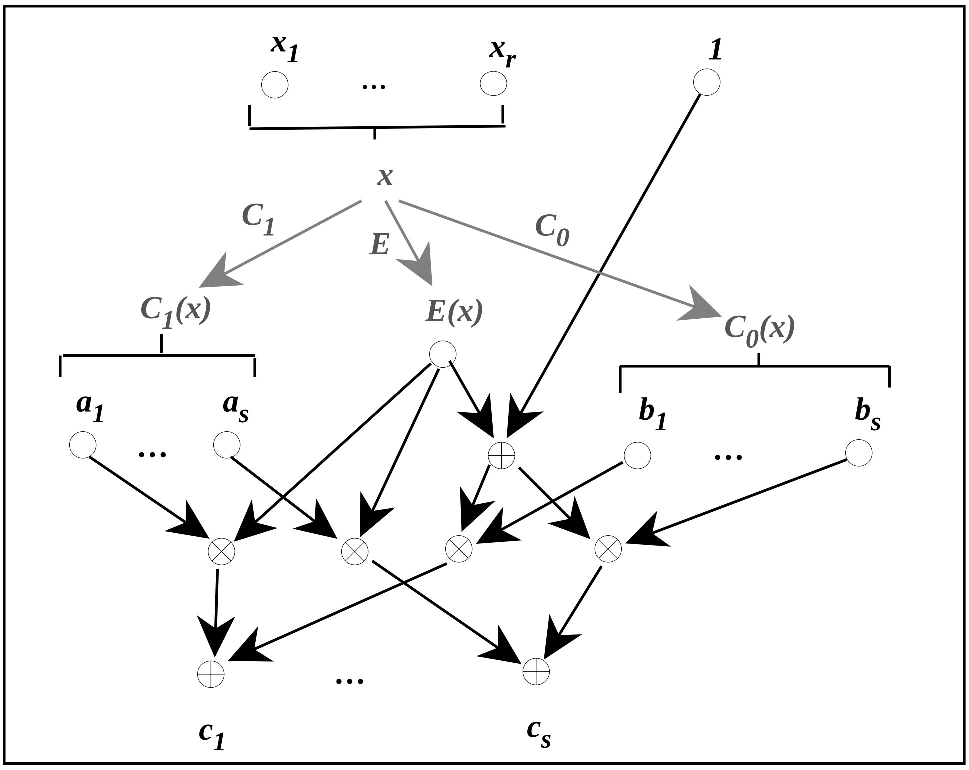

Abbildung 4: Schaltnetz für eine Verzweigung