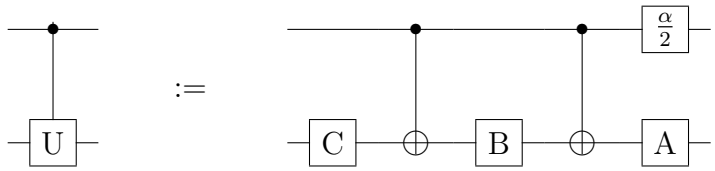
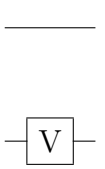
Quantum Computation:
Zusammenfassung der 4. Vorlesung (07.11.08)

1.2.1.2 *2-Qubitgatter (Fortsetzung)*

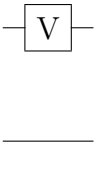
Wir hatten gezeigt [3. Vorlsesung, 1.2.1.2, Seite 4], dass jede $U(2)$-Matrix in der Form $U = e^{i\alpha} A\sigma_x B\sigma_x C$ mit $SU(2)$-Matrizen A; B, C geshrieben werden kann, wobei $ABC = \mathbf{1}$ ist. Mit Hilfe dieser Aussage konstruieren wir nun Gatter, die die Steuerung beliebiger 1-Qubitgatter bewirken:



Mit den von 1-Qubitgattern erzeugten unitären Matrizen



$$=(\mathbf{1} \otimes V) = \begin{pmatrix} V & 0 \\ 0 & V \end{pmatrix},$$



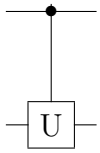
$$= (V \otimes \mathbf{1}) = \begin{pmatrix} v_{00}\mathbf{1} & v_{01}\mathbf{1} \\ v_{10}\mathbf{1} & v_{11}\mathbf{1} \end{pmatrix},$$

ergibt sich

$$\begin{pmatrix} \mathbf{1} & 0 \\ 0 & e^{i\alpha}\mathbf{1} \end{pmatrix} \begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix} \begin{pmatrix} \mathbf{1} & 0 \\ 0 & \sigma_x \end{pmatrix} \begin{pmatrix} B & 0 \\ 0 & B \end{pmatrix} \begin{pmatrix} \mathbf{1} & 0 \\ 0 & \sigma_x \end{pmatrix} \begin{pmatrix} C & 0 \\ 0 & C \end{pmatrix}$$
$$= \begin{pmatrix} ABC & 0 \\ 0 & e^{i\alpha} A\sigma_x B\sigma_x C \end{pmatrix} = \begin{pmatrix} \mathbf{1} & 0 \\ 0 & U \end{pmatrix}$$

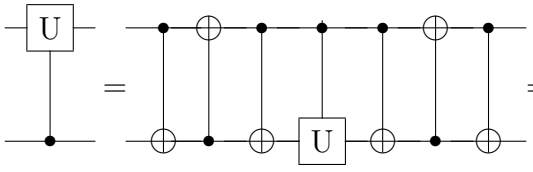
für das gesteuerte 1-Qubitgatter.

Neben

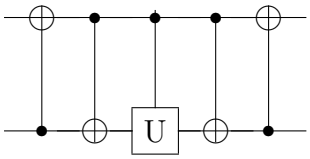


$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{pmatrix} =: U_{(23)},$$

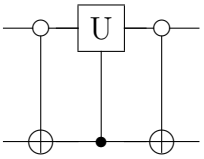
können mit den gesteuerten Verneinungen auch



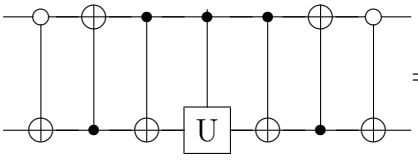
$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & u_{00} & 0 & u_{01} \\ 0 & 0 & 1 & 0 \\ 0 & u_{10} & 0 & u_{11} \end{pmatrix} =: U_{(13)},$$



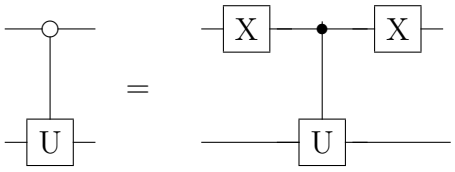
$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & u_{00} & u_{01} & 0 \\ 0 & u_{10} & u_{11} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} =: U_{(12)},$$



$$= \begin{pmatrix} u_{00} & 0 & 0 & u_{01} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ u_{10} & 0 & 0 & u_{11} \end{pmatrix} =: U_{(03)},$$



$$= \begin{pmatrix} u_{00} & 0 & u_{01} & 0 \\ 0 & 1 & 0 & 0 \\ u_{10} & 0 & u_{11} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} =: U_{(02)},$$



$$= \begin{pmatrix} u_{00} & u_{01} & 0 & 0 \\ u_{10} & u_{11} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} =: U_{(01)}$$

erzeugt werden. Die $U(4)$ Matrizen der Form $V_{(ij)}$, $i \neq j$ $(i,j = 0,1,2,3)$, bilden die zur $U(2)$ isomorphe Untergruppe, die in span$\{|i>, |j>\}$ operiert und das Orthokomplement im $\mathbf{C}^4$ invariant lässt. Wir wollen diese Untergruppe $U_{(ij)}(4)$ nennen.

Allgemein nennen wir die in $\mathbf{C}^N$ wirkende Untergruppen der $U(N)$, die unitär in span$\{|i>, |j>\}$,$i \neq j$ $(i,j = 0,1,2,\ldots,(N-1))$, operieren und die Orthokomplemente in $\mathbf{C}^N$ invariant lassen, $U_{(ij)}(N)$. Ferner bezeichnen wir die zu $U(N-1)$ isomorphe Untergruppe der $U(N)$, deren Elemente den Vektor $|0>$ invariant lässt, mit $1 \oplus U(N-1)$. Es gilt nun folgender Satz:

Satz: Sei $U \in U(N)$, dann gibt es $V_{(0k)} \in U_{(0k)}(N)$ $(k = 1,2,\ldots,(N-1))$, so dass

$$V_{(0(N-1))}V_{(0(N-2))}\ldots V_{(02)}V_{(01)}U \in 1 \oplus U(N-1)$$

gilt.

Beweis: Der Satz gilt für $N = 3$. Sei

$$U = \begin{pmatrix} a & * & * \\ b & * & * \\ * & * & * \end{pmatrix},$$

dann setze mit $r := \sqrt{|a|^2 + |b|^2}$

$$V_{(01)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ falls } b = 0, \qquad V_{(01)} = \frac{1}{r}\begin{pmatrix} \overline{a} & \overline{b} & 0 \\ b & -a & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ falls } b \neq 0.$$

In beiden Fällen ist

$$V_{(01)}V_{(01)}^{+} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad V_{(01)}U = \begin{pmatrix} r & * & * \\ 0 & * & * \\ c' & * & * \end{pmatrix}.$$

Nun setze mit $r' := \sqrt{r^2 + |c'|^2}$

$$V_{(02)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ falls } c' = 0, \qquad V_{(02)} = \frac{1}{r'} \begin{pmatrix} r & 0 & \overline{c'} \\ 0 & r' & 0 \\ c' & 0 & -r \end{pmatrix} \text{ falls } c' \neq 0.$$

In beiden Fällen ist

$$V_{(02)} V_{(02)}^{+} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad V_{(02)} V_{(01)} U = \begin{pmatrix} 1 & * & * \\ 0 & * & * \\ 0 & * & * \end{pmatrix},$$

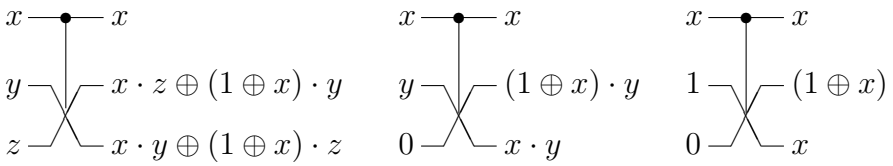
wobei im zweiten Fall in der letzten Matrix anstelle der 1 zunächst r' steht. Da $V_{(02)}V_{(01)}U$ unitär ist, muss der erste Spaltenvektor und der erste Zeilenvektor ein Einheitsvektor sein, also ist $r' = 1$ und

$$V_{(02)} V_{(01)} U = \begin{pmatrix} 1 & 0 & 0 \\ 0 & w_{11} & w_{12} \\ 0 & w_{21} & w_{22} \end{pmatrix} = 1 \oplus W \in 1 \oplus U(2).$$

Damit ist der Satz für $N = 3$ bewiesen, Bei einer gegebenen $(N \times N)$-Matrix führt das Verfahren auch für $N > 3$ nach $N - 1$ Schritten auch dazu, dass die Zeilen 1 bis $N - 1$ mit 0 beginnen. Ist die Matrix unitär, dann beginnt die 0-te Spalte mit 1 und man kann auf die behauptete Gleichung schließen.

Dieser Satz zeigt, dass die 1-Qubitgatter zusammen mit dem CNOT-Gatter eine universelle Menge für die 2-Qubitgatter bilden. Im $\mathbf{C}^4$ lassen sich aus den implementierbaren Gattern $1 \oplus U_{(12)}(4)$ und $1 \oplus U_{(13)}(4)$, d.h. den oben skizzierten Gattern der Form $U_{(12)}(4)$ und $U_{(13)}(4)$, die Gatter $1 \oplus U(3)$ bilden, aus denen wiederum alle $U(4)$ Gatter gebildet werden können.

Von besonderer Bedeutung ist auch das gesteuerte $SWAP$-Gatter:

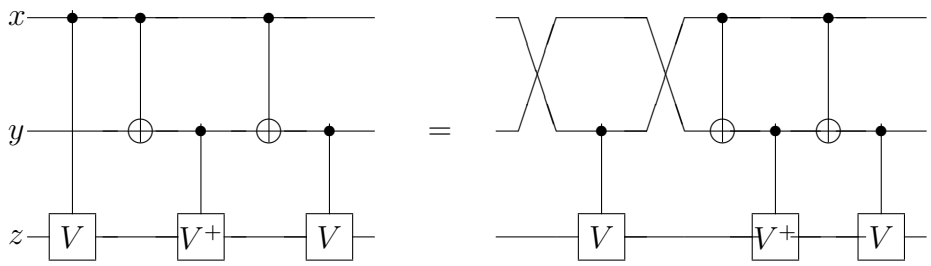


Dieses Gatter heißt Fredkingatter. Es ist offenbar reversibel und erlaubt es, die klassische Konjunktion und die klassische Negation reversibel zu implementieren. Als reversibles Gatter erzeugt es eine Permutation der acht

Wörter, die mit 3 Bits gebildet werden können. Mit der entsprechenden Permutationsmatrix lässt sich das Fredkingatter unitär im $\mathbf{C}^8$ implementieren. Es folgt, dass alle klassischen Algorithmen unitär implementiert werden können.

1.2.1.3 *N-Qubitgatter*

Hier ist der Hilbertraum der $\mathbf{C}^{2^N}$, und (2^N-1)-fach gesteuerte Verneinungen erzeugen die Transpositionsmatrizen der S_{2^N}. Wir konstruieren zunächst das zweifach gesteuerte 1-Qubit U-Gatter. Es wird durch den folgenden Schaltkreis erzeugt, wobei $V^2 = U$ ist. Anhand der "Wahrheitswerte" xy überlegt man sich "klassisch" für $|xyz>$ leicht, dass im Fall $xy = 0$ der Zustand $|z>$ des dritten Qubits unverändert bleibt. Im Fall $xy = 01$ wirkt V^+V auf $|z>$ und im Fall $xy = 10$ wirkt VV^+. Nur im Fall $xy = 11$ wirkt $V^2 = U$ auf $|z>$. Quantenmechanisch treten natürlich auch Überlagerungen der Basiselemente auf, die man nicht so leicht mit klassischen Argumenten behandeln kann .

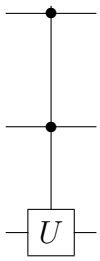


Die Gatter des rechten Schaltbildes sind uns schon bekannt und können sofort hingeschrieben und ausmultipliziert werden. Es ergibt sich

$$(\mathbf{1} \otimes \begin{pmatrix} \mathbf{1} & 0 \\ 0 & V \end{pmatrix})(\begin{pmatrix} \mathbf{1} & 0 \\ 0 & \sigma_x \end{pmatrix} \otimes \mathbf{1})(\mathbf{1} \otimes \begin{pmatrix} \mathbf{1} & 0 \\ 0 & V^+ \end{pmatrix})(\begin{pmatrix} \mathbf{1} & 0 \\ 0 & \sigma_x \end{pmatrix} \otimes \mathbf{1})$$

$$\cdot(\begin{pmatrix} 1 & 0 & 0 \\ 0 & \sigma_x & 0 \\ 0 & 0 & 1 \end{pmatrix} \otimes \mathbf{1})(\mathbf{1} \otimes \begin{pmatrix} \mathbf{1} & 0 \\ 0 & V \end{pmatrix})(\begin{pmatrix} 1 & 0 & 0 \\ 0 & \sigma_x & 0 \\ 0 & 0 & 1 \end{pmatrix} \otimes \mathbf{1}) =$$

$$= \begin{pmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & U \end{pmatrix} \qquad =$$

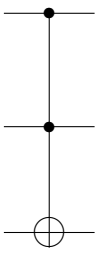


.

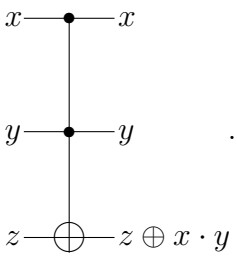
Rechts unten befindet sich das Symbol für das Schaltelement. Die zweifach gesteuerte Verneinung ergibt sich für $U = \sigma_x$, also für

$$V = \frac{1}{2}(1 \mp i)(\mathbf{1} \pm i\sigma_x), \qquad \text{denn} \qquad V^+V = \mathbf{1}, \qquad V^2 = \sigma_x.$$

Das resultierende Gatter trägt den Namen Toffoligatter:

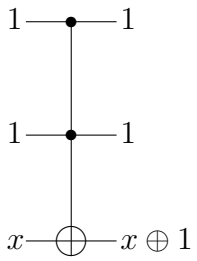


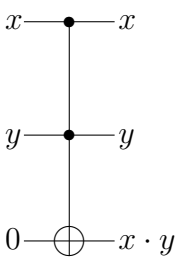
$$= \begin{pmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & \sigma_x \end{pmatrix}, \quad \text{klassisch:}$$

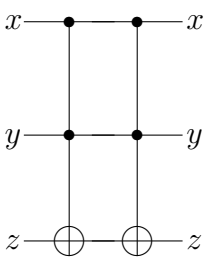


.

Das klassische Toffoli Gatter ist universell und reversibel.

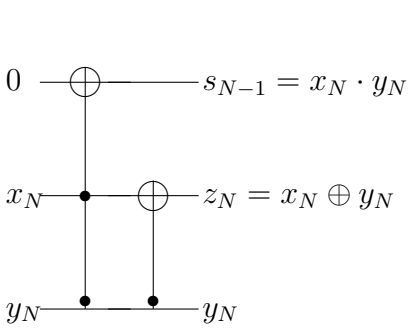


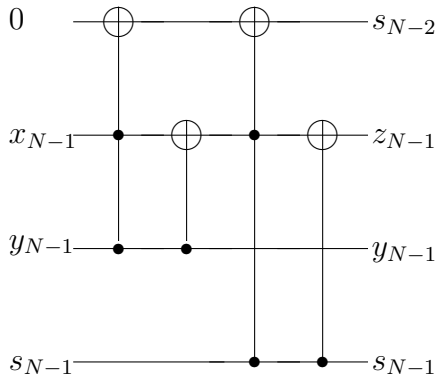




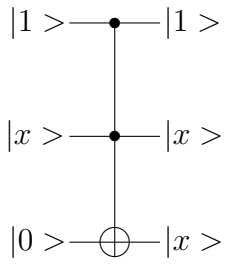
Mit dem Toffoligatter und seinen durch Vor- und Nachschalten von X oder $SWAP$ erzeugten Versionen lassen sich deshalb alle klassischen Algorithmen auch unitär implementieren, so zum Beispiel der Halbaddierer und der Vol-

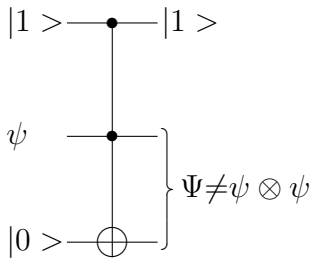
laddierer:





Dabei ist $s_{N-2} = x_{N-1} \cdot y_{N-1} \oplus s_{N-1} \cdot (x_{N-1} \oplus y_{N-1})$ und $z_{N-1} = x_{N-1} \oplus y_{N-1} \oplus s_{N-1}$. Bei der unitären Implementierung wird der Zweck des klassischen Algorithmus für Überlagerungen der Basiszustände im Allgemeinen entfremdet. So geht die Kopierfunktion des Toffoli Gatters bei Überlagerungen verloren:

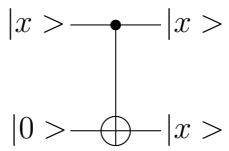


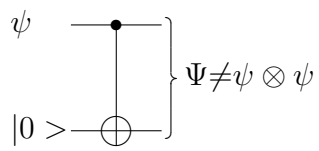


Anstelle einer Kopie des Zustands ψ wird der verschränkte Zustand

$$\Psi = |00>< 0|\psi> + |11>< 1|\psi>$$

erzeugt, der im Falle $| < 0|\psi > | = | < 1|\psi > | = \frac{1}{\sqrt{2}}$ maximal verschränkt ist. Für die eben beschriebenen Funktionen muss man allerdings nicht das Toffoligatter heranziehen, die einfach gesteuerte Verneinung





tut es auch.