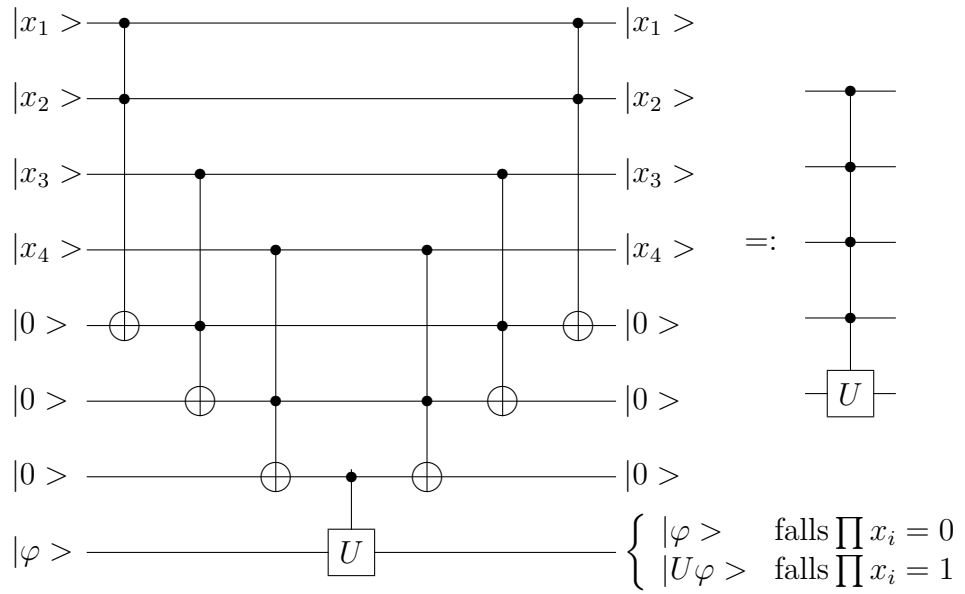
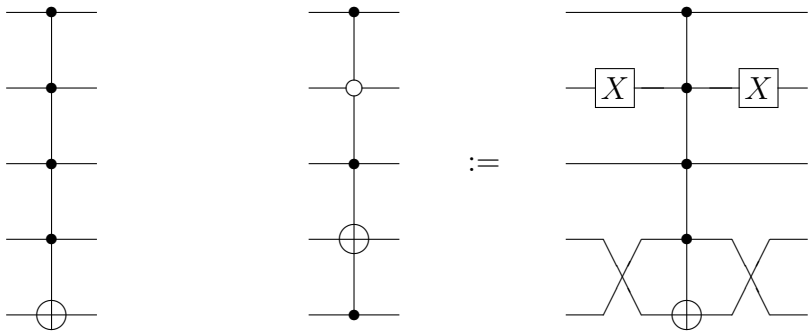
Quantum Computation:
Zusammenfassung der 5. Vorlesung (27.05.2011)

1.2.1.3 *N-Qubitgatter (Fortsetzung)*

Mit dem Toffoli Gatter lassen sich n-fach gesteuerte Gatter konstruieren, wobei $n-1$ Hilsbits benötigt werden, die zur Erzeugung des Produkts der Steuerbits dienen. Der Fall $n=4$ ist im folgenden Bild skizziert.



Speziell lassen sich für $U=X$ die n-fach gesteuerten Verneinungen in verschiedenen Versionen konstruieren, von denen zwei Beispiele im folgenden Bild gezeigt werden.



Klassich betrachtet erzeugen diese Gatter Transpositionen der 2^N Wörter, die mit N Bits gebildet werden können, hier der $2^5 = 32$ Wörter, die mit 5 Bits gebildet werden können. Als $(2^N \times 2^N)$-Permutationsmatrizen dargestellt bewirken sie Transpositionen der Elemente der Computerbasis (durch Multiplikation von trcchts) bzw. der Komponenten eines Spaltenvektors (durch Multiplikation von links), hier im $\mathbf{C}^{32}$, denn offenbar gilt für das Gatter rechts im letzten Bild

$$|x_1x_2x_3x_4, x_5 > \longmapsto \begin{cases} |x_1x_2x_3(x_4 \oplus 1)x_5 > & \text{falls } (x_1x_2x_3x_4x_5) = (101x_41), \\ |x_1x_2x_3x_4x_5 > & \text{sonst.} \end{cases}$$

Die Transposition vertauscht also die Plätze der Wörter (10101) und (10111) miteinander. Auf diese Weise kann man Transpositionen von solchen Dualzahlen (Wörtern) erzeugen, die sich nur in einer Dualstelle unterscheiden. Zur Erzeugung beliebiger Transpositionen dient der *Gray code*: Man gelangt zur Transposition von einer gegebenen Dualzahl mit einer anderen, indem man eine Spalte bildet, die mit der einen gegebenen Dualzahl beginnt und mit der anderen endet, und in der sich aufeinanderfolgende Dualzahlen nur in einer Stelle unterscheiden, etwa

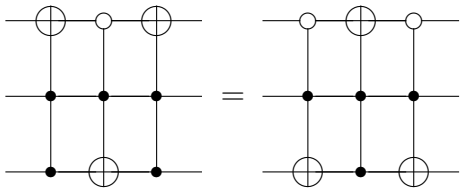
$$\left.\begin{array}{ccccc} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{array}\right\} = \left\{\begin{array}{c} k \\ l \\ m \\ n \end{array}\right. .$$

Rechts stehen der besseren Übersicht halber die zugehörigen Dezimalzahlen, die sich in der entsprechenden Zweierpotenz unterscheiden. Die Komposition der Transpositionen, die von einer Zeile zur nächsten führen und von der oben betrachteten Art sind, ergeben die gewünschte Transposition. Ist T_{ij} die Permutationsmatrix für die Vertauschung von i und j, dann gilt

$$T_{kn} = T_{mn}T_{lm}T_{kl}T_{lm}T_{mn} = T_{kl}T_{lm}T_{mn}T_{lm}T_{kl}.$$

Bei der Multiplikation kann man von der Regel gebrauch machen, dass der linke Faktor die entsprechenden Zeilen des rechten Faktors vertauscht und der rechte die entsprechenden Spalten des linken. Für drei Qubits lässt sich z.B. T_{27} mit

$$\left.\begin{matrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}\right\} = \left\{\begin{matrix} 2 \\ 3 \\ 7 \end{matrix}\right. \quad \text{mit}$$



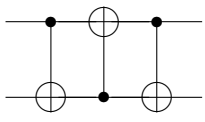
implementieren, und es gilt

$$T_{27} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

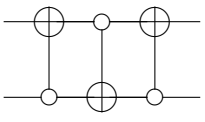
$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} = T_{37}T_{23}T_{37}.$$

Ein weiteres Beispiel ist das $SWAP$-Gatter, das wir auf viererlei Weise mit zwei verschiedenen Gray codes

$$\left.\begin{matrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{matrix}\right\} = \left\{\begin{matrix} 1 \\ 3 \\ 2 \end{matrix}\right.$$

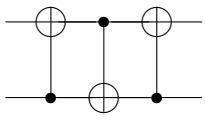


$=$

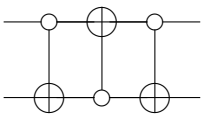


$$\left.\begin{matrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{matrix}\right\} = \left\{\begin{matrix} 1 \\ 0 \\ 2 \end{matrix}\right.$$

$=$



$=$



gemäß

$$T_{12} = T_{23}T_{13}T_{23} = T_{13}T_{23}T_{13} = T_{02}T_{01}T_{02} = T_{01}T_{02}T_{01}$$

betrachten. Da eine beliebige Permutation durch Produkte einer geraden oder ungeraden Anzahl von Transpositionen darghestellt werden kann, sind somit alle Permutationen implementierbar.

Wir hatten schon gezeigt, dass die 1-Qubitgatter zusammen mit dem $CNOT$-Gatter für die 2-Qubitgatter eine universelle Menge bilden, nachdem wir

(i) die Permutationen der S_4, und damit

(ii) die $1 \oplus \mathrm{U}_{(ij)}(3) -$ Matrizen $\begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \alpha & \beta \\ & & \gamma & \delta \end{pmatrix}, \begin{pmatrix} 1 & & & \\ & \alpha & 0 & \beta \\ & 0 & 1 & 0 \\ & \gamma & 0 & \delta \end{pmatrix},$

$\begin{pmatrix} 1 & & & \\ & \alpha & \beta & \\ & \gamma & \delta & \\ & & & 1 \end{pmatrix}$, und damit

(iii) die $1 \oplus \mathrm{U}(3) -$ Matrizen $\begin{pmatrix} 1 & \\ & U \end{pmatrix}$

implementieren konnten und im Abschnitt 1.2.1.2. den 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.

bewiesen hatten. Für N-Qubitgatter haben wir nun gezeigt, dass wir

(i) die Permutationen der S_{2^N}, und damit

(ii) die $1^{\otimes k} \oplus \mathrm{U}_{(ij)}(2^N - k) -$ Matrizen

$(k = 0, 1, 2, \ldots, (2^N - 2))$ und damit

(iii) die $1^{\otimes(2^N - (2^{N-1}+l))} \oplus \mathrm{U}(2^{N-1} + l) -$ Matrizen $\begin{pmatrix} 1^{\otimes l} & \\ & U \end{pmatrix}$

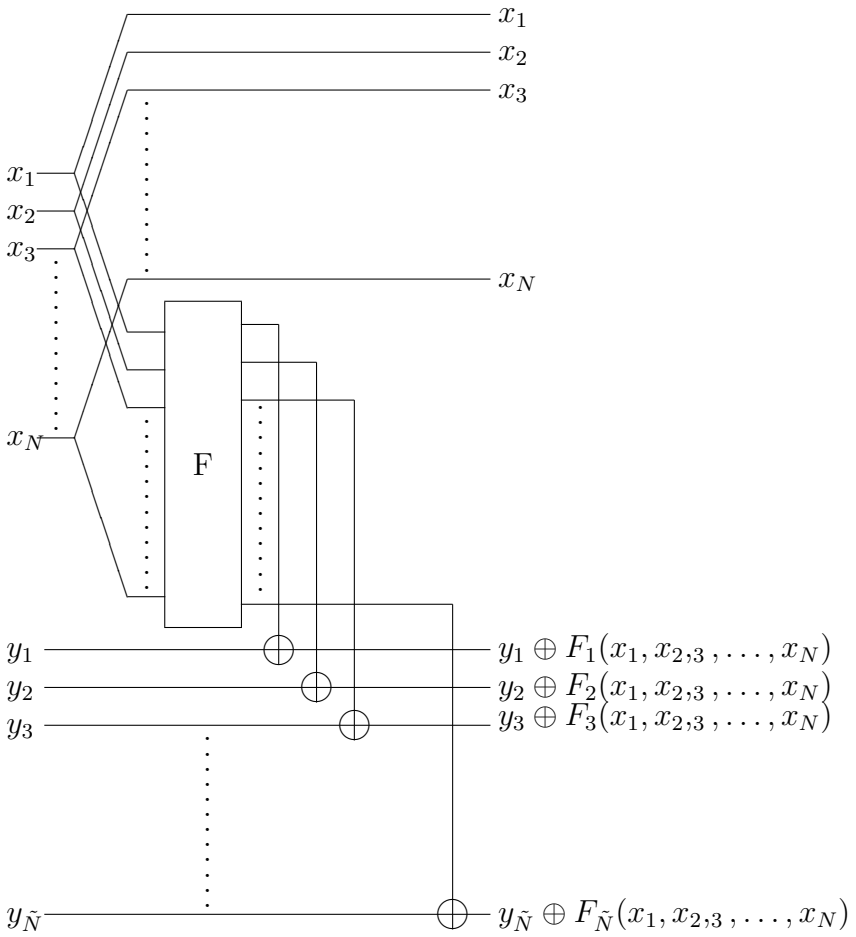
$(l = 1, 2, \ldots, (2^N - 2^{N-1}) - 1)$

implementieren können. Der zitierte Satz erlaubt damit, die Implementierbarkeit von N-Qubitgattern aus 1-Qubitgattern und $CNOT$-Gattern auf die von $(N-1)$.Gattern zurückzuführen. Mit der Implementierbarkeit von 2-Qubitgattern als Induktionsanfang und der von $(N-1)$-Qubitgattern als Induktionsannnahme ist damit gezeigt:

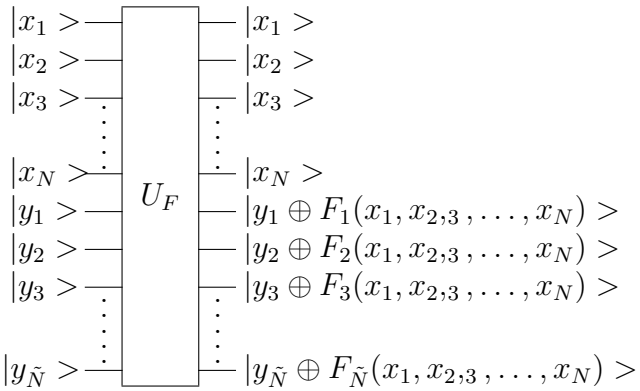
Satz: Die 1-Qubitgatter bilden zusammen mit dem $CNOT$-Gatter eine universelle Menge für unitäre N-Qubitgatter.

1.2.1.4 *Unitäre Versionen klassischer Algorithmen*

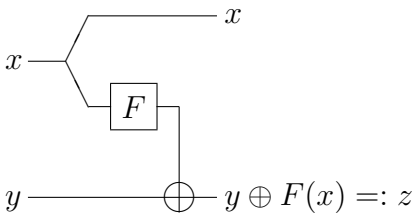
Wir hatten im Zusammenhang mit dem Toffoligatter schon bemerkt, dass sich jeder klassische Algorithmus auch unitär implementieren lässt. Ein radikales Verfahren, einen irreversiblen klassischen Funktor $F : \{0,1\}^N \to \{0,1\}^{\tilde{N}}$ reversibel zu machen zeigt die folgende Konstruktion:



Dieses Gatter ist von zweiter Ordnung, zweimal hintereinander geschaltet erzeugt es die Identität. Damit ist das Gatter reversibel und kann mit einer Permutationsmatrix aus $S_{2^{N+\tilde{N}}}$ als unitäre Transtormation eines Quantenalgorithmus dargestellt werden. Hier werden $N + \tilde{N}$ Stellen benötigt. Dies ist im Allgemeinen nicht optimal. Bei der reversiblen Implementation des Halbaddierers kamen wir mit drei Bits anstelle von fünf Bits aus, die das radikale Verfahren erfordert, beim reversiblen Volladdierer mit vier anstelle von sechs. Ein Vorteil des betrachteten Verfahrens ist, dass man den klassischen Algorithmus nicht analysieren muss, er kann sogar unbekannt sein. Zu jedem klassischen Algorithmus gibt es also einen Quantenalgorithmus mit der unitären Matrix U_F und kann mit den $U(2)$- und CNOT-Gattern in einem Schaltkreis implementiert werden:



Speziell betrachten wir die vier monatischen Wahrheitswertfunktoren, von denen zwei reversibel, nämlichdie *Identität*: $(F(0), F(1)) = (0, 1)$) und die *Negation*: $(F(0), F(1)) = (1, 0)$), und zwei irreversibel, nämlich die *Antilogie*: $(F(0), F(1)) = (0, 0)$) und die *Tautologie*: $(F(0), F(1)) = (1, 1)$), sind. Mit dem radikalen Verfahren werden alle vier reversibel durch Permutationen dargestellt.



$(F(0)F(1))$		(00)	(01)	(10)	(11)
xy		xz	xz	xz	xz
00	0	0	0	1	1
01	1	1	1	0	0
10	2	2	3	2	3
11	3	3	2	3	2

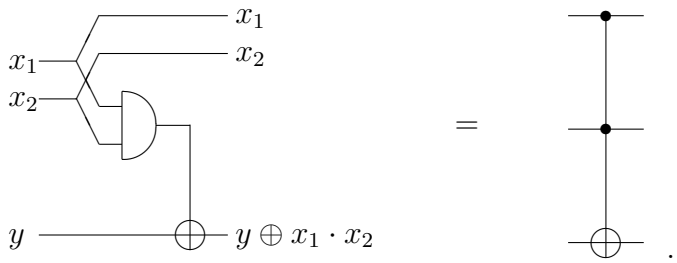
Der besseren Übersicht halber wurde in der Wahrheitswerttabelle von der Dezimalzählung Gebrauch gemacht. Die unitäre Implementation führt auf die Matrix

$$U_F = \begin{pmatrix} \delta_{0F(0)}\mathbf{1} + \delta_{1F(0)}\sigma_x & \\ & \delta_{0F(1)}\mathbf{1} + \delta_{1F(1)}\sigma_x \end{pmatrix}.$$

Der Deutschalgorithmus, den wir im nächsten Kapitel betrachten werden,

dient der Untersuchung eines unbekannten monadischen Wahrheitswertfunktors auf Reversibilität.

Ein weiteres Beispiel ist die reversible Implementation der Konjunktion, die auf das Toffoli Gatter führt:



.