Quantenschaltkreise

Seminar: Quantenrechner ~ Sommersemester 2004

Dozenten: Prof. Johannes Köbler und Olaf Beyersdorff

Vortrag: Jens Kleine ~ jkleine@informatik.hu-berlin.de

Vortag vom 12.05.2004 ~ Humboldt Universität zu Berlin

Inhaltsverzeichnis:

Rückblick Qubits.....3

Unitäre Matrizen.....4

No-Cloning-Satz.....5

Quantengatter.....6

1-Qubit Bausteine.....6

Quantenbausteine für mehrere Qubits.....7

Quantenschaltkreise.....9

Quantenprogramme.....10

Simulation klassischer Schaltkreise.....11

Simulation probabilistischer Schaltkreise..15

Literaturverzeichnis.....16

Rückblick Qubits

Qubits sind eine Zustandsbeschreibung in der Quantenmechanik mit Hilfe eines Vektors mit Norm 1 im Hilbertraum. Eine Standardbasis für Qubits ist: $\{|0\rangle,|1\rangle\}$ wobei $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ und $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ ist.

Ein Qubit lässt sich darstellen als $|y\rangle = a|0\rangle + b|1\rangle$. Dabei werden *a* und *b* Amplituden genannt und es muss gelten $|a|^2 + |b|^2 = 1$. Eine Messung des Qubits $|y\rangle$ liefert nun mit einer Wahrscheinlichkeit von $|a|^2$ das Ergebnis $|y\rangle = 0$ und mit einer Wahrscheinlichkeit von $|b|^2$ das Ergebnis $|y\rangle = 1$. Anschließend befindet sich das Qubit im Zustand $|0\rangle$ oder $|1\rangle$.

Analog gibt es für 2-Qubits eine Standardbasis aus den Vektoren:

$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, $|01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$, $|10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$ und $|11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$.

Operationen werden mathematisch durch Matrixmultiplikation mit unitären Matrizen realisiert. Unitäre Matrizen sind stets umkehrbar und somit können aus den Ausgabewerten eines Schaltkreises die Eingabewerte rekonstruiert werden.

Unitäre Matrizen

Die Wiederherstellbarkeit der Eingabewerte ist ein fundamentaler Unterschied zur klassischen Logik und deren Schaltkreisen. Nach der Anwendung eines UND-Gatters auf zwei Werte *a* und *b* kann nicht mehr rekonstruiert werden, ob bei dem Ausgabewert *false* nun *a* oder *b* oder gar beide Eingabewerte mit dem Wert *false* belegt waren.

a	b	Ausgabe	Eingabe
0	0	0	a? b? ?
1	0	0	a? b? ?
0	1	0	a? b? ?
1	1	1	a & b

Die Verwendung von unitären Matrizen und damit reversibler Schaltkreise ist laut [BAU02], [RAT03] und [SAU03] notwendig auf Grund des Landauer Prinzips (1961), das besagt, dass zur Löschung eines Bits an physischer Information mindestens Energie der Menge $k_B T \ln 2$ an die Umgebung verloren geht.

k_B = Stefan-Boltzmann-Konstante = $1.3806 * 10^{-23} \frac{J}{K}$

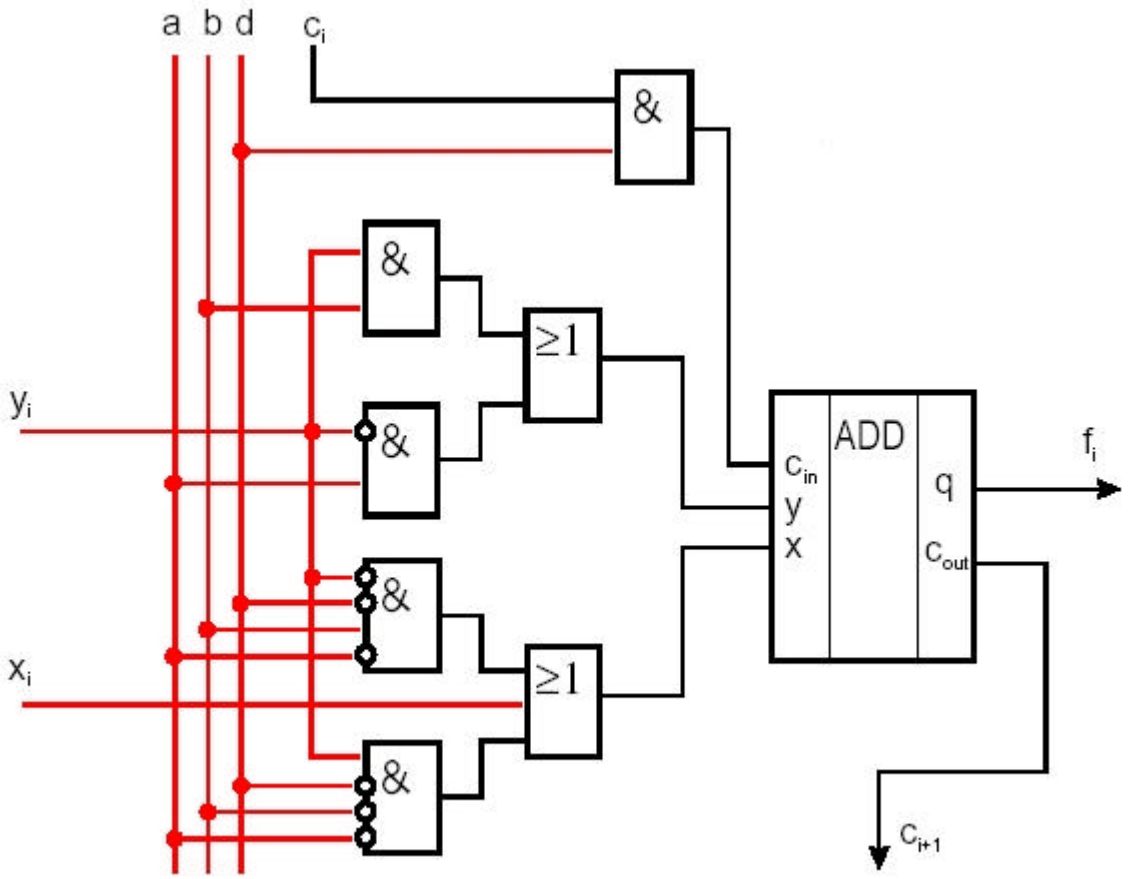
T = Temperatur (in Kelvin) des Rechners

In klassischen Computern geht ständig Energie in Form von Wärme verloren. Bei Quantenrechnern darf dies jedoch nicht geschehen, denn die Superposition der Quantenzustände muss über die gesamte Berechnung beibehalten werden und die Kohärenz der Zustände würde verloren gehen, wenn innerhalb des Computers unkontrolliert Energie in Form von Wärme abgegeben würde.

Allerdings folgt nach [TOR03] durch die Forderung der Verwendung unitärer Matrizen und der damit verbundenen Reversibilität der Quantengatter, dass die Anzahl der Eingänge der Anzahl der Ausgänge entsprechen muss. Eine Einschränkung, die bei klassischen und probabilistischen Schaltkreisen nicht gefordert wird, die jedoch nicht schwierig zu realisieren ist. Die klassischen boolschen Funktionen lassen sich ebenfalls als Matrizenmultiplikation darstellen.

Hierbei stellt man jedoch fest, dass sich nicht allen boolschen Funktionen als unitäre Matrizen darstellen lassen.
Zusätzlich ist mit unitären Matrizen auch noch folgender Satz verbunden.

No-cloning-Satz
Es gibt keine Unitärtransformation U, die ein bisher unbestimmtes Qubit kopieren kann.
Mathematisch lässt sich beweisen, dass das Kopieren von Qubits nur möglich ist, wenn diese den Wert 0 oder 1 haben (siehe [TOR03]), also ihr Zustand bereits durch eine Messung festgelegt wurde. Hiermit können zwar klassische Schaltkreise simuliert werden, jedoch können die Zustände von Qubits, die nicht den Wert 0 oder 1 haben nicht kopiert werden. Ein Fanout der Leitungen ist in einem Quantenschaltkreis somit ebenfalls nicht möglich. Es kann keinen Bus geben, von denen alle Komponenten dieselben Daten mehrfach abgreifen. Es ist allerdings möglich einen Quantenzustand auf einen anderen Ort zu übertragen, jedoch wird dabei das Original zerstört.



Quantengatter

In [RAT03] wird ein Quantenschaltkreis definiert als eine Menge von azyklisch verbundenen Quantengattern. Ein Quantengatter mit *n* Inputs und *n* Outputs wird durch eine unitäre $n \times n$-Matrix repräsentiert. Die Quantenverdrahtung der Quantengatter ist dabei nicht mit einer physischen Verbindung bei klassischen Schaltkreisen vergleichbar. Effekte wie Feldinteraktionen zwischen einzelnen Qubits erzeugen eine Verbindung zwischen den Quantengattern, die im Vergleich mit klassischen Schaltkreisen als Verdrahtung verstanden werden kann. Wie bei klassischen Gattern können die Funktionen eines Quantengatters durch eine Wahrheitstabelle der Zustände beschrieben werden.

1-Qubit Bausteine

Bei der Erstellung von klassischen Schaltkreisen kann man lediglich auf zwei Bausteine zurückgreifen, die auf einem Bit arbeiten. Dies sind die Identität und die Negation. Die Belegungstabellen dieser Schaltkreise kommen daher mit nur einem Eingangs- und einem Ausgangswert aus.

Identität:

I	O
0	0
1	1

Negation:

I	O
0	1
1	0

Bei der Negation ergibt sich die Frage, was mit Qubits geschehen soll, die nicht einen der Werte 0 oder 1 haben. Da die Transformationen bei Qubits linear sind, wird auch die Negation über die Zwischenwerte linear erfolgen. Benötigt wird also ein Baustein, der $a|0\rangle + b|1\rangle$ in $a|1\rangle + b|0\rangle$ überführt. Es ergibt sich für die Transformation die Matrixdarstellung:

$$NOT := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Hierbei handelt es sich offensichtlich um eine unitäre Transformation, denn die zweifache Hintereinanderausführung ergibt die Identität:

$$NOT^t * NOT = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Nun lassen sich im Gegensatz zur klassischen Logik noch mehr Quantenbausteine definieren, die auf einem Bit arbeiten. In [SAU03] werden einige dieser 1-Qubit Bausteine beschrieben. Hier nun noch als Beispiel der Hadamard-Baustein, da dieser später bei der Simulation probabilistischer Schaltkreise Verwendung findet. Definiert wird er in Matrixdarstellung wie folgt:

$$H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Es gilt nun:

$$H|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$
$$H|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

Somit ergibt sich, dass der Hadamard-Baustein mit einer Wahrscheinlichkeit von jeweils $\frac{1}{2}$ als Ergebnis 0 oder 1 liefert. Da auch der Hadamard-Baustein eine unitäre Transformation realisiert, liefert die zweifache Anwendung die Identität. Dies bedeutet jedoch, dass die einmalige Anwendung auf einen festen Wert ***x*** ein zufälliges Ergebnis ***y=0*** oder ***y=1*** erzeugt und nach der zweimaligen Anwendung der Ausgangszustand ***x*** wieder herbeigeführt wird, statt den neuen, ebenfalls festen, Wert ***y*** erneut in einen zufälligen Wert zu überführen.

Quantenbausteine für mehrere Qubits

Wie bereits am Beispiel des UND-Gatters gezeigt, schränkt die Forderung nach unitären Transformationen die Möglichkeiten der Bildung von Quantenbausteinen ein. Man wünscht sich einen universellen Bausteinsatz, mit dem sich alle boolschen Funktionen darstellen lassen. Aus den Ausgangswerten von UND,- XOR- und NAND-Gattern lassen sich die Eingangswerte nicht rekonstruieren. Daher bedient man sich bei der Erstellung von Quantenbausteinen einer unitären XOR-Variante, die Controlled NOT oder auch CNOT

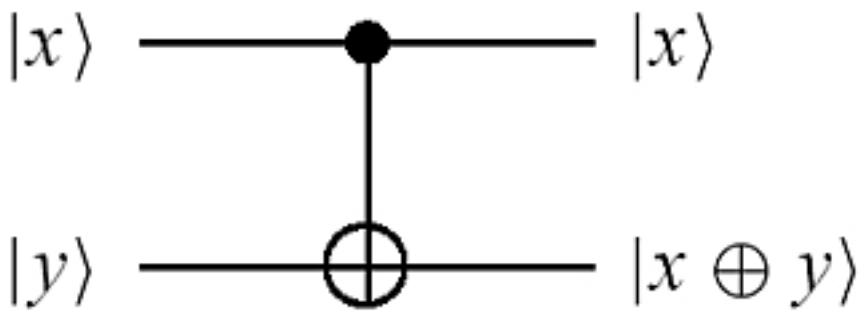
genannt wird. Wie ein gewöhnliches XOR hat dieser Baustein 2 Eingänge, wie für einen Quantenbaustein üblich, hat er dieselbe Anzahl an Ausgängen. Der erste Eingang trägt den Namen *Steuerbit* und der zweite den Namen *Zielbit* oder auch *gesteuertes Bit*. Ist nun das *Steuerbit* aktiv, also im Zustand $|1\rangle$, wird das *Zielbit* geflippt, ansonsten passiert nichts. Es ergibt sich somit folgende Transformation:

$|00\rangle \rightarrow |00\rangle$
$|01\rangle \rightarrow |01\rangle$
$|10\rangle \rightarrow |11\rangle$
$|11\rangle \rightarrow |10\rangle$

Die Matrixdarstellung ist:

$$CNOT := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Dieser Quantenbaustein hat ein eigenes Symbol erhalten, bei dem sich das *Steuerbit* oben und das *Zielbit* unten befindet:

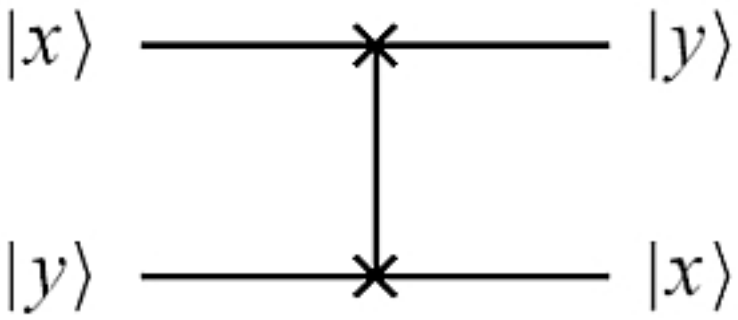


Zusammen mit den 1-Qubit Bausteinen bildet der CNOT Baustein einen universellen Bausteinsatz. Der Beweis dazu findet sich in Kapitel 3 von [SAU03].

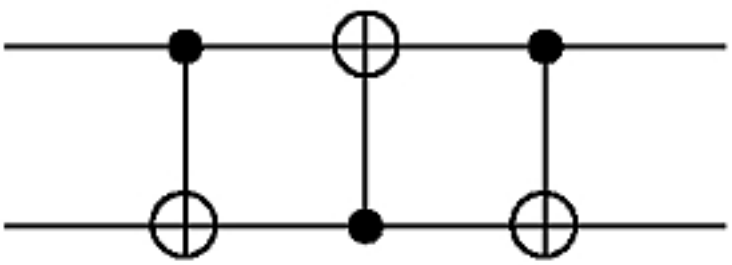
Quantenschaltkreise

Quantenschaltkreise werden ähnlich dargestellt, wie klassische Schaltkreise, jedoch haben die Quantenschaltkreise, wie die Quantenbausteine aus denen sie bestehen, für *n* Eingänge immer auch *n* Ausgänge. Sie führen Zustandstransformationen durch, die sich durch die Hintereinanderausführung der Quantenbausteine ergibt. Die Leitungen in einem Quantenschaltkreis lassen sich nach [SAU03] als die einzelnen Qubits interpretieren. Der Zustand einer Leitung entspricht daher dem Zustand eines Qubits. Der Gesamtzustand aller Leitungen ergibt sich jedoch auf Grund der Besonderheiten der Quantenschaltkreise nicht unbedingt aus den Teilzuständen der Leitungen.

Da die Quantenschaltkreise durch die Hintereinanderausführung von Quantenbausteinen arbeiten, die unitäre Transformationen darstellen, muss das Produkt von unitären Matrizen wieder eine unitäre Matrix ergeben. Dass dem so ist, wird in Kapitel 1.4 von [SAU03] bewiesen. Hier nun als Beispiel für einen Quantenschaltkreis, der die Bezeichnung SWAP trägt und zwei Eingangssignale vertauscht. Somit realisiert er eine Überkreuzung von Drähten, die in Quantenschaltkreisen sonst nicht erlaubt ist. Daher erhält der Schaltkreis wegen seiner Wichtigkeit ein eigenes Symbol:



Er wird aus drei CNOT-Gattern zusammengesetzt:



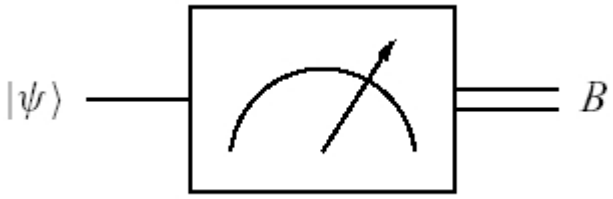
Die Auswirkungen des Schaltkreises lassen sich wie folgt aus den drei Einzelauswirkungen der CNOT-Gatter berechnen:

$$\begin{aligned}|a,b\rangle &\rightarrow |a,a\oplus b\rangle \\ &\rightarrow |a\oplus(a\oplus b),a\oplus b\rangle = |b,a\oplus b\rangle \\ &\rightarrow |b,(a\oplus b)\oplus b\rangle = |b,a\rangle\end{aligned}$$

Die dazugehörige unitäre Matrix ist:

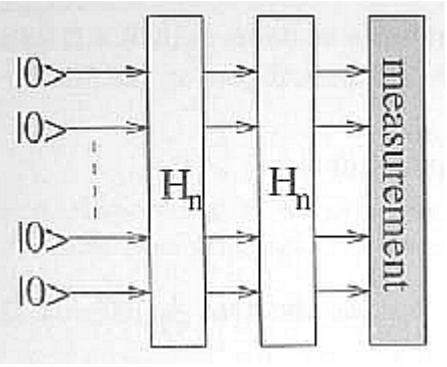
$$SWAP := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ein letzter besonderer Baustein führt Messungen durch und weißt somit den Qubits Werte zu. Angewendet auf einen Zustand mit der Eigenschaft $|y\rangle = a|0\rangle + b|1\rangle$ liefert der Baustein mit der Wahrscheinlichkeit $|a^2|$ für den Wert 0 und mit $|b^2|$ für den Wert 1 ein klassisches Bit B. Leitungen in Quantenschaltkreisen, die klassische Bits leiten, werden meist durch doppelte Linien dargestellt. Damit ergibt sich für die Messung dieser Baustein:



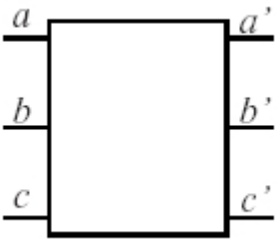
Quantenprogramme

Quantenprogramme werden laut [RAT03] definiert als ein Quantenschaltkreis mit Messgattern am Ende.



Simulation klassischer Schaltkreise

Zur Simulation klassischer Schaltkreise verwendet man im Normalfall das Fredkin-Gatter, das folgenden Aufbau



und die Regeln

$$f(a,b,0) = (a,b,0)$$
$$f(a,b,1) = (b,a,1)$$

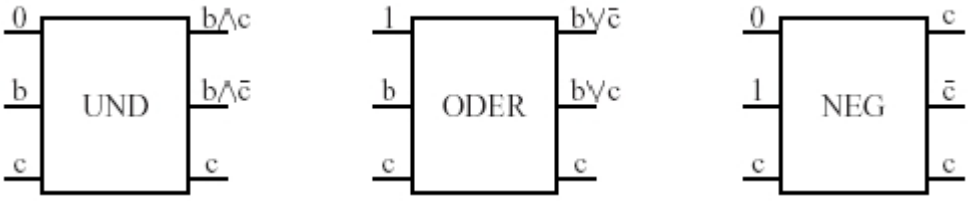
hat. Das Fredkin-Gatter vertauscht die Eingänge a und b, falls der Steuerungseingang c den Wert 1 hat. Eine zweifache Hintereinanderausführung des Fredkin-Gatters ergibt die Identitätsfunktion.

a_1	b_1	c_1	a_1'	b_1'	c_1'	a_2	b_2	c_2	a_2'	b_2'	c_2'
x	**y**	**0**	x	y	0	x	y	0	**x**	**y**	**0**
x	**y**	**1**	y	x	1	y	x	1	**x**	**y**	**1**

Konstanten, wie 0 und 1 werden als *ancillas* bezeichnet. Als Matrixdarstellung des Fredkin-Gatters ergibt sich somit:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Mit Hilfe dieses Gatters und unter zur Hilfenahme von ancillas lassen sich nun die elementaren Funktionen UND, ODER und NEG simulieren, wie die folgenden Belegungstabellen zeigen.



UND:

a	**b**	c	**a'**	b'	c'
0	**0**	0	**0**	0	0
0	**0**	**1**	**0**	0	1
0	**1**	0	**0**	1	0
0	**1**	**1**	**1**	0	1

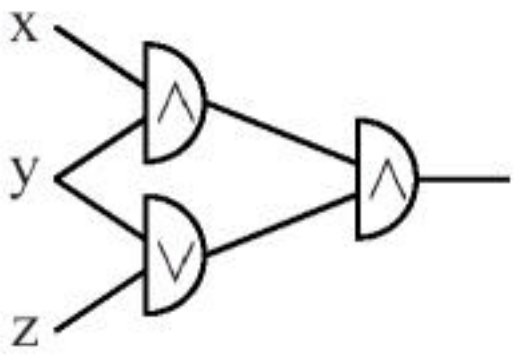
ODER:

a	**b**	c	a'	**b'**	c'
1	**0**	0	1	**0**	0
1	**0**	**1**	0	**1**	1
1	**1**	0	1	**1**	0
1	**1**	**1**	1	**1**	1

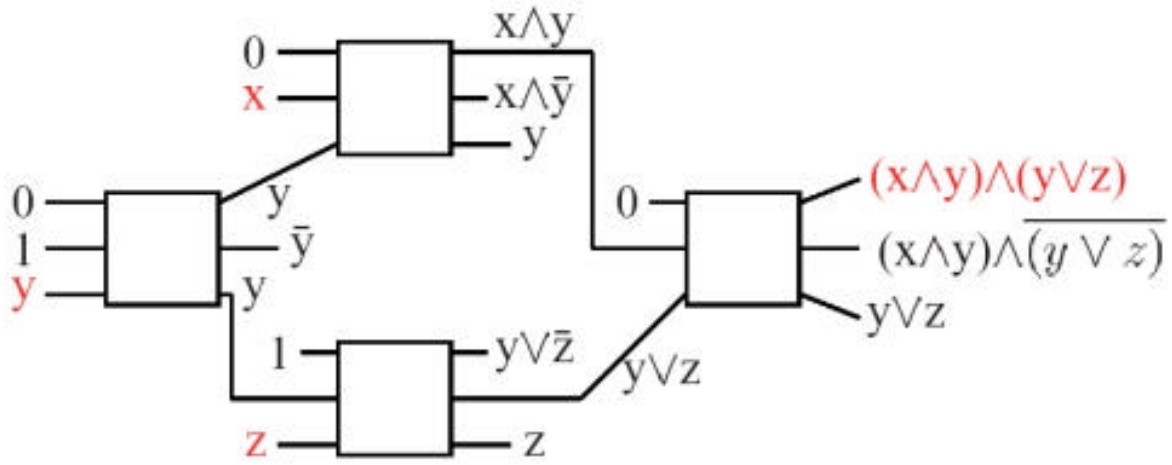
NEG:

a	b	**c**	a'	**b'**	c'
0	1	**0**	0	**1**	0
0	1	**1**	1	**0**	1

UND, ODER und NEG bilden einen vollständigen Satz an Bausteinen für klassische Schaltkreise. Somit lassen sich die gerade aufgeführten drei Bausteine zu beliebigen klassischen Schaltkreisen zusammensetzen. Hier ein Beispiel für einen klassischen Schaltkreis, der aus den Eingabewerten x, y und z die Funktion $(x \wedge y) \wedge (y \vee z)$ berechnet.



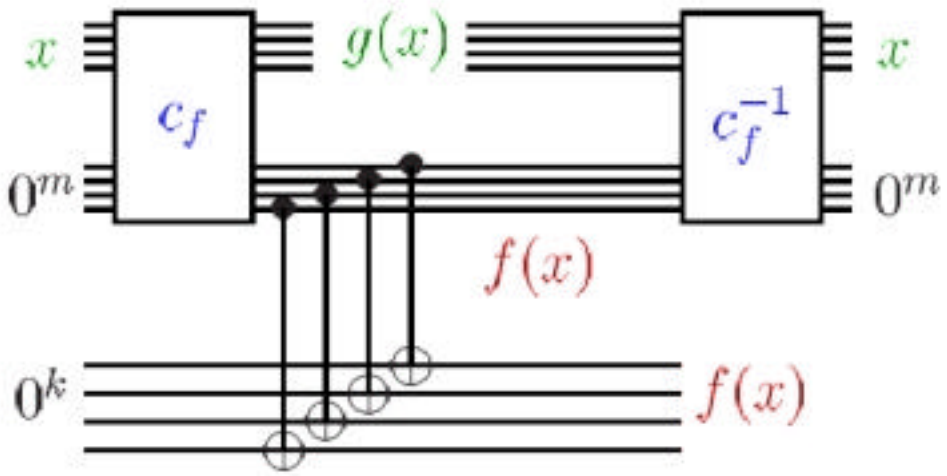
Als Quantenschaltkreis könnte die Funktion so realisiert werden:



Zu beachten ist hier, dass das NEG-Gatter als Ersatz für einen Fanout verwendet wird. Da hier ein klassischer Schaltkreis simuliert wird, und somit nur die boolschen Werte 0 und 1 für die Eingänge angenommen werden, kann der Eingangswert von *y* kopiert werden.

Bei der tatsächlichen Konstruktion von Quantenschaltkreisen würde man jedoch die zusätzlichen Ausgänge aus ancillas und unnötigen Zwischenergebnissen, die zusammen als *garbage* oder *g(x)* bezeichnet werden, durch die Umkehrung der Rechenschritte wieder in ihre Anfangswerte überführen. Zudem würden die ancillas mit dem Wert 1 durch Negation von ancillas mit dem Wert 0 ersetzt, so dass in der Eingabe nur noch *m* ancillas mit dem Wert 0 vorkommen. Für den Ausgabewert der berechneten Funktion *f(x)* verwendet man

zusätzliche *k* ancillas, da bei Quantenschaltkreisen die Eingänge dieselbe Dimension wie die Ausgänge haben müssen. Somit ergibt sich als allgemeiner Bauplan für die Berechnung einer Funktion *f(x)*:



Man erhält nun also außer der Eingabe nur noch das Ergebnis *f(x)* zurück und es ergibt sich für die Schaltung: $(x,0^m,0^k) \rightarrow (x,0^m, f(x))$
Vernachlässigt man nun die ancillas ergibt sich: $(x) \rightarrow (x, f(x))$
Ersetzt man nun die 0^k ancillas durch eine *k*-dimensionale Variable *y* unter Verwendung der CNOT-Verknüpfung ergibt sich final für die Schaltung die Gleichung: $(x, y) \rightarrow (x, y \oplus f(x))$
Damit ist gezeigt, dass sich klassische Schaltkreise durch Quantenschaltkreise simulieren lassen.

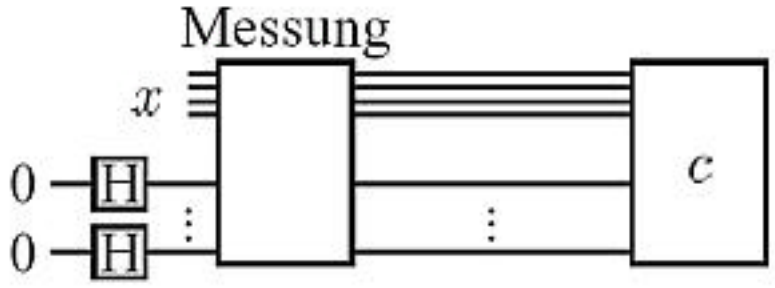
Simulation probabilistischer Schaltkreise

Probabilistische Schaltkreise unterscheiden sich von klassischen darin, dass sie zusätzlich einige zufällige Konstanten in ihre Berechnung mit einbeziehen. Klassische Schaltkreise lassen sich wie oben gezeigt mit Quantenschaltkreisen simulieren und auf Grund der Natur von Qubits lassen sich Zufallswerte einfach erzeugen, indem man an Stellen innerhalb des Schaltkreises Messungen durchführt, die die Werte von Qubits festlegen. Verwendet man die auf Seite 7 vorgestellte Hadamard-Transformation, die mit der Wahrscheinlichkeit $\frac{1}{2}$ als Ergebnis 0 oder 1 zurückgibt, auf *n* Konstanten mit dem Wert 0, so gilt:

$$\left|0^n\right\rangle \rightarrow^H \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} |i\rangle$$

Eine Messung erzeugt nun einen Vektor mit neuen Konstanten, deren Werte eine zufällige Gleichverteilung über $\{0, 1\}^n$ darstellen, da sich die Wahrscheinlichkeit für einen konkreten Zustand $|i\rangle$ zu $\left|\frac{1}{\sqrt{2^n}}\right|^2 = \frac{1}{2^n}$ ergibt.

Die Simulation eines probabilistischen Schaltkreises durch Quantenschaltkreise kann dann so aussehen:



Literaturverzeichnis

Vorlesungen & Seminarvorträge:

[BAU02]:
Sebastian Bauer: Seminar Quanteninformationsverarbeitung; Vortrag Nr. 4, Quantengatter & -Algorithmen; Walther-Meissner-Institut für Tieftemperaturforschung, Bayerische Akademie der Wissenschaften 2002;
http://www.wmi.badw.de/E23/seminars/WS0203-QIV/2002-12-17.manuskript.pdf

[TOR03]:
Jacobo Torán: Vorlesung Quantencomputer; Mitschrift von Jan Paul Buchwald; Universität Ulm 2003;
http://theorie.informatik.uni-ulm.de/Lehre/SS3/Quantum/skript.ps

[SAU03]:
Martin Sauerhoff: Spezialvorlesung Quantenrechner Algorithmen und Komplexität; Universität Dortmund 2003;
http://ls2-www.cs.uni-dortmund.de/lehre/sommer2003/QR/skript.pdf

[RAT03]
Martin Ratajczak: Hauptseminar Quantencomputer; Universität München 2003;
http://www.tcs.informatik.uni-muenchen.de/lehre/SS03/Quanten/vortraege/ratajczak.ps