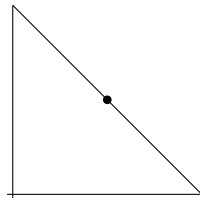


9. microsoft videocast

youtube Op de [microsoft site](#) dezelfde film, maar met extra info.

Computers werken met bits, dat zijn twee-toestand modellen, waar of onwaar, nul of één, kop of munt. In een computer wordt met spanningen gerekend. Transistoren zijn de schakelaars die door een programma aan of uitgezet worden. We kunnen daar ingewikkelde dingen mee doen, maar daar houdt de vergelijking met een quantum computer zo'n beetje op.



Figuur 9.1: Een klassiekbit, muntje opgooien

Als we een muntje opgooien geldt kans op munt + kans op kop = 1. Een eerlijk muntje zal met gelijke kans kop of munt geven.

We kunnen die toestand weergeven als een vector $\begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$ of $\begin{pmatrix} \frac{1}{6} \\ \frac{5}{6} \end{pmatrix}$ als we bijvoorbeeld de uitkomst=drie of geen drie als basis nemen.

Er is een groot verschil met quantum computers. Hiervoor moeten we even wat lineaire algebra ophalen.

9.0.1 lineaire algebra

We gebruiken de eenheidsvectoren $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ en $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ om de twee toestanden van een bit te definiëren.

Bit met waarde 0 $\equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix} \equiv |0\rangle$

Definitie:

$$0 \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix} \equiv |0\rangle$$

9. microsoft videocast

Kansen met Quantum

Definitie:

$$1 \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix} \equiv |1\rangle$$

Bit met waarde 1 $\equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix} \equiv |1\rangle$

(braket notatie nog niet ingevoerd.)

(matrix)(kolomvector)=(kolomvector):

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax + by \\ cx + dy \end{pmatrix}$$

- Bereken de uitkomst van de volgende vectoroperatie

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

matrixvermenigvuldiging

(matrix)(matrix)=(matrix) Een product van twee matrices levert een nieuwe matrix op. Je vindt de elementen van de nieuwe matrix door de elementen van rijen van de linker matrix paarsgewijs te vermenigvuldigen met elementen van kolommen uit de rechter matrix. Je vindt dan het element op het kruispunt van de rij en kolom. De matrix moet vierkant zijn, anders werkt het niet.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} p & q \\ r & s \end{pmatrix} = \begin{pmatrix} ap + br & aq + bs \\ cp + dr & cq + ds \end{pmatrix}$$

We geven een voorbeeld in twee dimensies. Als we op een toestand $\begin{pmatrix} a \\ b \end{pmatrix}$ eerst de X-operator loslaten en vervolgens de Z operator, dan

ziet dat er als volgt uit. $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} b \\ a \end{pmatrix} =$

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

$$\begin{pmatrix} b \\ -a \end{pmatrix}$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Hetzelfde resultaat bereik je als je eerst de matrixen vermenigvuldigt:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} b \\ a \end{pmatrix}$$

en vervolgens deze operatie op de vector toepast. $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} b \\ a \end{pmatrix} = \begin{pmatrix} b \\ -a \end{pmatrix}$

- ★ Laat zien dat als de twee matrixen worden verwisseld, dit een andere uitkomst levert.

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} b \\ a \end{pmatrix} = \begin{pmatrix} -b \\ a \end{pmatrix}$$

De volgorde is dus van belang.

Matrixen zijn operatoren, vectoren zijn toestanden. Een matrix toegepast op een toestand levert een nieuwe toestand. We zullen zien dat we slechts een handjevol operatoren nodig hebben, en ze hebben allemaal een naam. Omdat we ons in deze module beperken tot operaties met reële getallen, is het aantal nog eens beperkt. S0, S1, voorbeelden uit klassieke computers I, X, Z, H en CNOT voor reversibele, quantum computers.

pas op:

$ZX \neq XZ$ Let op, je mag de operaties niet van plaats verwisselen, matrixvermenigvuldiging is niet commutatief!

toestand in quantum is abstract, anders dan bij een klassieke computer is de toestand zelf niet meetbaar
Operator op een toestand geeft een nieuwe toestand

tensorproduct

$$\begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \otimes \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_0 \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \\ x_1 \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} x_0 y_0 \\ x_0 y_1 \\ x_1 y_0 \\ x_1 y_1 \end{pmatrix}$$

Weergeven van meerdere *klassieke* bits

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$4_{10} = 100_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{Het product van } n \text{ bits}$$

is een vector met afmetingen 2^n .

Links staat de notatie zoals een quantumcomputer die ziet: drie qubits. Rechts staat het uitgeschreven tensorproduct zoals je dat op een klassieke computer zou weergeven. Je ziet dat het aantal mogelijkheden bij elke bit verdubbelt. Als je een quantumprobleem op een klassieke manier weergeeft heb je daar dus enorm veel geheugen voor nodig. De berekeningen lopen enorm uit de hand. In de quantumwereld kunnen de tensoroperaties uitvoeren.

Computer poorten

En computer moet een aaneenschakeling van bewerkingen op bits kunnen verwerken. Met een opeenvolging van instructies zijn mooie programma's te schrijven. Een tekstverwerperpakket bestaat al gauw uit miljoenen regels code. We zien zullen zien dat zelfs een heel klein quantumprogramatje al dingen kan doen die een klassiek programma nooit zou kunnen.

NB het is niet zinnig klassieke poorten als EN en OF in deze wijze weer te geven. Deze poorten hebben twee ingangen en een uitgang.

Klassieke computer

Beschouw de volgende vier klassieke operatoren op 1 bit:

S0

De operatie beeldt een bit altijd op $|0\rangle$ af. De bewerking is onomkeerbaar. Deze operatie bestaat daarom alleen in een klassieke computer. Een quantumcomputer moeten de bewerkingen omkeerbaar zijn. Hier gaan we later verder op in. $f(x) = (0)$, $S0 =$

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

$$|0\rangle \longleftrightarrow |0\rangle$$

$$|1\rangle \longleftrightarrow |0\rangle$$

S1

Set to 1 Deze operatie bestaat alleen in een klassieke computer. Een quantumcomputer moeten de bewerkingen omkeerbaar zijn.

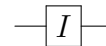
Hier gaan we later verder op in. $f(x) = (1)$, $S1 = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}$

$$|0\rangle \longleftrightarrow |1\rangle$$

$$|1\rangle \longleftrightarrow |1\rangle$$

Identiteit

Identiteit. Dit is de simpelste operatie. Hij verandert niets aan de toestandsvector. De bewerking is omkeerbaar en is zowel in een klassieke als in een quantum computer gedefinieerd. $f(x) = x, I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

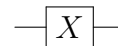


$$|0\rangle \longleftrightarrow |0\rangle$$

$$|1\rangle \longleftrightarrow |1\rangle$$

● X

Deze poort wordt ook wel de *bitflip* genoemd. Geef deze poort weer met de letter X. De bewerking is omkeerbaar en is zowel in een klassieke als in een quantum computer gedefinieerd. Het komt er op neer dat de twee coördinaten worden verwisseld. $f(x) = (\text{not } x), X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

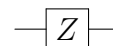


$$|0\rangle \longleftrightarrow |1\rangle$$

$$|1\rangle \longleftrightarrow |0\rangle$$

★ Z

De Z-gate komt later aan de orde.



$$f(x) = (\text{not } x), Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$|0\rangle \longleftrightarrow |0\rangle$$

$$|1\rangle \longleftrightarrow |-1\rangle$$

je ziet dat deze poort alleen in een quantumcomputer kan bestaan. De afbeelding op $\begin{pmatrix} 0 \\ -1 \end{pmatrix}$ heeft pas betekenis na et kwadrateren van de coëfficiënt ($\beta = -1$).

reversibele computing

Een operator is reversibel als je uit de uitkomst van een operatie de beginwaarde kunt terugrekenen. Er bestaat dan een inverse operatie. Je kent dit uit bijvoorbeeld de analyse van een rechte lijn: Uit de x-waarde kun je een y-waarde uitrekenen, en omgekeerd.

$$Ax = y, x = A^{-1}y$$

★ Bij twee lijnen gaat dat mis. Welke?

$$\text{De X-as } (y = 0) \text{ en de Y-as } (x = 0)$$

het overschrijven van de bits in de poorten S0 en S1 is niet reversibel. Zij passen niet in een quantumcomputer. I en X zijn reversibel. Quantum computers gebruiken alleen reversibele operatoren. Bovendien zijn alle quantumoperatoren hun *eigen* inverse. Wat gebeurt er als je ze twee keer toepast?

$$XX = X^{-1}X = I$$

Makkelijk in te zien bij de bitflip: Een bitflip twee keer toepassen levert je het origineel.

● Noem nog een makkelijk voorbeeld.

VanNeuman-Landauer Identiteit.

limit: minimale hoeveelheid energie voor een bit, de eenheid van informatie. Deze ligt mogelijk lager voor een reversibel systeem.

Hadamard

Deze poort wordt later behandeld. (kan hier ook)

CNOT

de CNOT is de enige operator (die wij behandelen) die op twee bits tegelijk werkt. Het MSB is het controlebit, het LSB is het doelbit

(target). Als het controlebit gelijk is aan $|1\rangle$ dan wordt doelbit geflipped. Als het controlebit is $|0\rangle$ dan blijft het doel onveranderd. Het controlebit is altijd onveranderd.

$$|MSBLSB\rangle = |CT\rangle$$

$$|00\rangle \longleftrightarrow |00\rangle$$

$$|01\rangle \longleftrightarrow |01\rangle$$

$$|10\rangle \longleftrightarrow |10\rangle$$

$$|11\rangle \longleftrightarrow |11\rangle$$

De matrixvorm van de CNOT is 4x4, we moeten inmers twee bits

bijhouden. $CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

We schrijven het nog twee keer uit:

$$CNOT|10\rangle = CNOT \left(\begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} =$$

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |11\rangle$$

$$CNOT|11\rangle = CNOT \left(\begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} =$$

$$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |10\rangle$$

In beide voorbeelden hierboven wordt een tensorproduct uitvermenigvuldigd, bewerkt met een operator en vervolgens wordt het het omgekeerde gedaan, een tensor wordt gefactoriseerd. We zullen

zien dat dit factoriseren niet altijd mogelijk is. De berekening kan alleen als tensor bestaan. Dit is een eigenschap die met quantum-pits eenvoudig te maken is, maar met klassieke bits enorm veel geheugen vraagt.

- Ga zelf na dat $CNOT |00\rangle = |00\rangle$ en dat $CNOT |01\rangle = |01\rangle$

Recap

klassieke bits stel je voor met 0 en 1. in vectornotatie $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ en $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$

- Bewerkingen op bits (operaties) kun je voorstellen door matrixvermenigvuldigingen van vectoren
- Quantum computers gebruiken alleen reversibele operatoren
- multibit toestanden kun je uitdrukken als tensorproduct van enkel bit vectoren
- CNOT poort is een fundamenteel element van reversibele computer

9.0.2 van klassiek bit naar qubit

Een klassiek bit is een speciale versie van een qubit. Een qubit is een veralgemenisering van een klassiek bit $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ waarbij $\alpha, \beta \in \mathbb{C}$ Er

dit is superpositie. geldt bovendien dat $|\alpha|^2 + |\beta|^2 = 1$ De volgende toestandsvectoren uitleggen zijn voorbeelden van qubits:

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \begin{pmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Is $\begin{pmatrix} -1 \\ 0 \end{pmatrix}$ een geldig qubit? en $\begin{pmatrix} -1 \\ 1 \end{pmatrix}$

Ja, levert met gelijke kans $|0\rangle$ of $|1\rangle$.

ja, levert met kans $\frac{1}{4}$ en $\frac{3}{4}$ een $|0\rangle$ of $|1\rangle$.

Ja levert altijd een $|1\rangle$.

Ja levert altijd een $|1\rangle$.

Nee, niet genormaliseerd

Een qubit is in superpositie tot we een meting doen.

een meting bestaat uit het opmeten en kwadrateren van de coëfficiënten.

Als we een qubit meten vervalst de toestand van het qubit tot $|0\rangle$ of $|1\rangle$

Meten verdient een eigen paragraaf

op een qubit $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ kun je quantum bewerkingen uitvoeren. Daarbij worden de mogelijke uitkomsten in de lucht gehouden tot dat een meting wordt verricht. Tijdens deze berekeningen ben je in de quantum wereld en is het bit in superpositie van de mogelijke uitkomsten. Denk aan de kat van Schrödinger, zie fig. 9.2. Zodra je meet is het quantumstelsel verstoord en niet meer bruikbaar.

Aan het eind van een berekening wordt een meting uitgevoerd. Daarbij valt een qubit met zekerheid terug tot een klassieke $|0\rangle$ of $|1\rangle$. De uitkomst wordt met een kans bepaald. Tijdens de berekening in de quantumwereld, de berekening in een quantumcomputer gaan met complexe getallen. In deze module gebruiken we alleen bewerkingen die met reële getallen kunnen worden opgelost. (Misschien in de steilste pistes?) Een qubit $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ Een berekening valt het bit zeker na stort in een $|0\rangle$ met kans $|\alpha|^2$ en tot $|1\rangle$ met kans $|\beta|^2$ Bijvoorbeeld $\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$ heeft gelijke kans op $|0\rangle$ en $|1\rangle$ (muntje opgooien)

$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ levert met 100 % kans $|0\rangle$ op. (wat is α , wat is β ?)

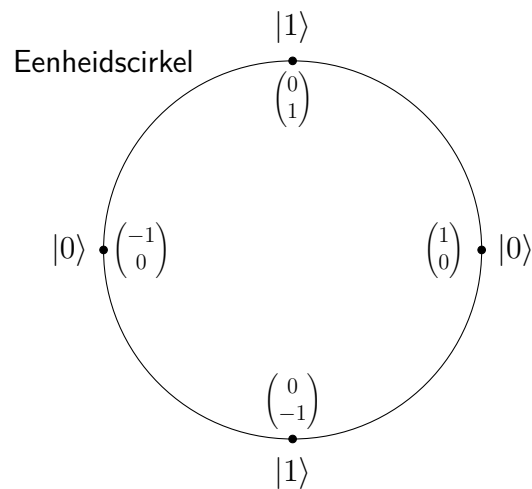
$$\frac{1}{\sqrt{2}}|\text{cat}\rangle + \frac{1}{\sqrt{2}}|\text{cat}\rangle$$

Figuur 9.2:
toestand van
Schrödinger's kat

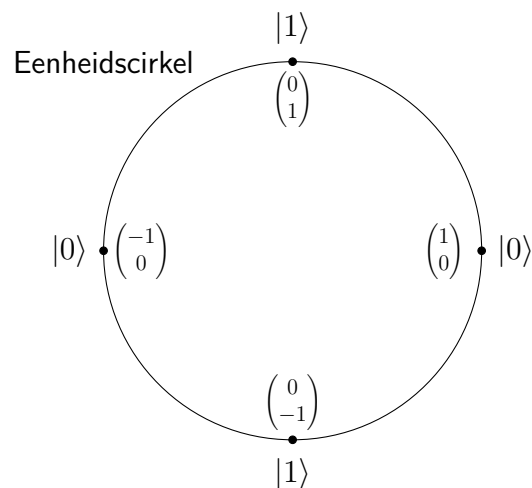
9.0.3 normalisatie

Een toestandsvector $\begin{pmatrix} a \\ b \end{pmatrix}$ voldoet aan de eis $|a|^2 + |b|^2 = 1$

Alle paren (a, b) die aan deze voorwaarde voldoen liggen op de eenheidscirkel. Wij beschouwen geen complexe waarden van a , en b , maar deze regel geldt evengoed met complexe getallen. Wij gebruiken alleen reële getallen. Als we complexe getallen toestaan, zoals bij een echte quantumcomputing, wordt deze cirkel een bol (Bloch-sphere) en is er nog veel meer mogelijk.



Figuur 9.3: Eenheidscirkel met projecties op klassieke bits.



Figuur 9.4: Eenheidscirkel met projecties op klassieke bits.

In fig. 9.4 staan aan de binnenkant de toestanden weergegeven. Aan de buitenkant staat hoe deze toestanden gemeten worden uit het kwadraat van de coëfficiënten α en β . Vanwege deze kwadraten komen $|0\rangle$ en $|1\rangle$ twee keer voor.

twee qubits

Hoe reken je met coëfficiënten als je twee bits beschouwt? Het tensorproduct moet aan de normalisatie van kansen voldoen. een uitgewerkt voorbeeld:

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix} \text{ met } |ac|^2 + |ad|^2 + |bc|^2 + |bd|^2 = 1$$

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \otimes \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$$

Ga na dat het resultaat genormaliseerd is. Er is gelijke kans van $\frac{1}{4}$ op $|00\rangle, |01\rangle, |10\rangle, |11\rangle$.

Later zullen we zien dat de combinatie van quantumbits bijzondere toestanden op kan leveren.

9.0.4 Operatoren voor qubits

Kijk nog eens naar de operaties die we bij klassieke computer hebben gedefinieerd.

$f(x) = \text{not}(x) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ Dit levert voor klassieke computers altijd coëfficiënten α 's en β 's op die nul of een zijn. Meer smaken zijn er niet.

Met quantumbits, bijvoorbeeld: $f(x) = \text{not}(x) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{3}}{2} \\ -\frac{1}{2} \end{pmatrix}$

de coëfficiënten α en β moeten weliswaar aan de voorwaarde $|a|^2 + |b|^2 = 1$, maar zij kunnen alle waarden tussen nul en een aannemen. Dit is een eigenschap die quantumcomputers zo krachtig maakt. Bovendien werken quantumcomputers met coëfficiënten in het domein complexe getallen, dat maakt het aantal mogelijkheden natuurlijk nog groter. Het lijkt een enorme beperking voor een poort om zijn eigen inverse te zijn, maar dit volgt uit de eis van reversibiliteit. In quantum zijn er meer poorten mogelijk.

9.0.5 Hadamard

Aan de hand van een Hadamard gate laten we drie methoden zien waarop we beweringen op een quantumcomputer kunnen weergeven. De Hadamard gate is de poort waarmee we in en uit de quantumwereld kunnen stappen, en is daarom belangrijk.

$$H|0\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$H|1\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Als je de H-gate op het resultaat loslaat krijg je het origineel terug. Dat moet geen verrassing zijn, want quantum gates zijn hun eigen inverse.

$$H(H|0\rangle) = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$H(H|1\rangle) = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

We zijn uitgegaan van klassieke bits. Het in superpositie brengen, daarmee zijn we de quantumcomputer binnengekomen. We kunnen ook weer uit superpositie treden zonder te meten! Hierdoor kan quantumcomputing deterministisch worden i.p.v. probabilistisch.

NB Een quantumcomputer kan alle klassieke operaties uitvoeren als je enen en nullen gebruikt voor de coëfficiënten.

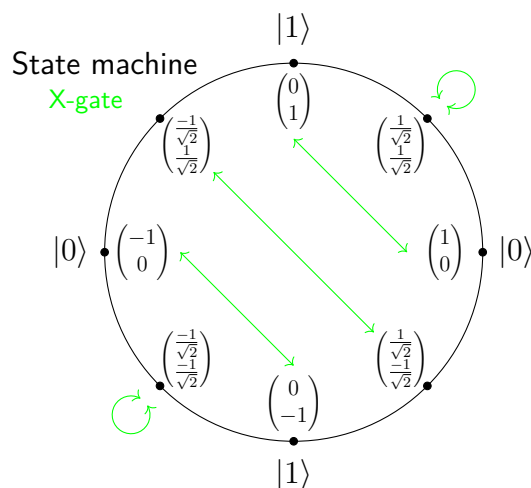
9.0.6 Finite state machine

Een finite state machine (eindige toestand machine) geeft een beschrijving van alle mogelijke toestanden en de overgangen daartussen. Dit is is een veelgebruikte techniek bij de beschrijving van apparaten of processen. We geven een voorbeeld uit het dagelijks leven.

- Een stoplicht kent drie toestanden, en enkele overgangen. Beschrijf een stoplicht met deze techniek. Maak een tekening met toestanden en gebruik pijlen om de overgang tussen de toestanden weer te geven.

vier notaties
poorten: ma
uitschrijven, L
en dirac, state r
nes, en blokjes
algoritme

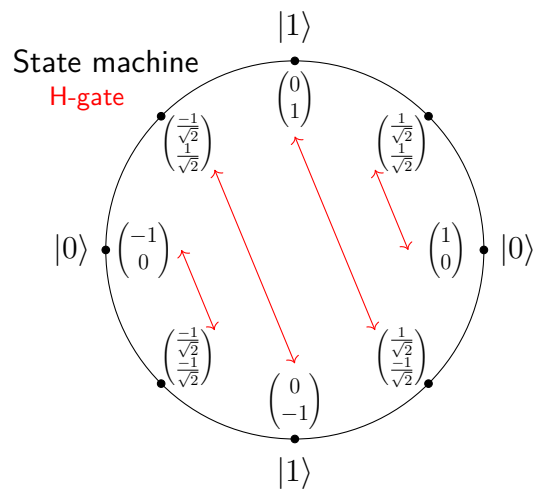
De toestand en de toestandsverandering van qubits als gevolg van de actie van poorten kun je ook in een state machine weergeven zie fig. 9.5. Binnen in de cirkel staan de toestanden gegeven zoals die in de quantumcomputer bestaan. Buiten de cirkel staan de toestanden $|0\rangle$ en $|1\rangle$ zoals we die zouden meten.



Figuur 9.5: State machine voor X-gate.

- Kun je een symmetrieas aangeven? Welke toestanden worden op zichzelf afgebeeld? Wat gebeurt er als je een X-gate twee keer toepast? Aan welke eisen van quantumpoorten voldoet deze poort? In de figuur komen de toestanden twee keer voor. Kun je dat verklaren? [kwadraat van coëfficiënten]. De toestandsruimte in de computer is dus groter dan onze klassieke wereld. We zien zo dat deze volledige beschrijving nodig is.

Met twee keer spiegelen breng je de afbeelding weer terug tot het origineel. Met de hulp van deze statemachines hoef je niet meer al die matrixen uit te rekenen. Ook voor een Hadamard gate kun je een toestandsdiagram maken.



Figuur 9.6: State machine voor H-gate.

Ook de Hadamard-gate is zo overzichtelijk weer te geven. De symmetrie van een Hadamard-gate is iets ingewikkelder.

- Kun je een symmetrie aangeven? Worden er toestanden op zichzelf afgebeeld? Wat gebeurt er als je een H-gate twee keer toepast? Aan welke eis van quantumgates voldoet deze gate? Lees uit de figuur af $H|0\rangle$

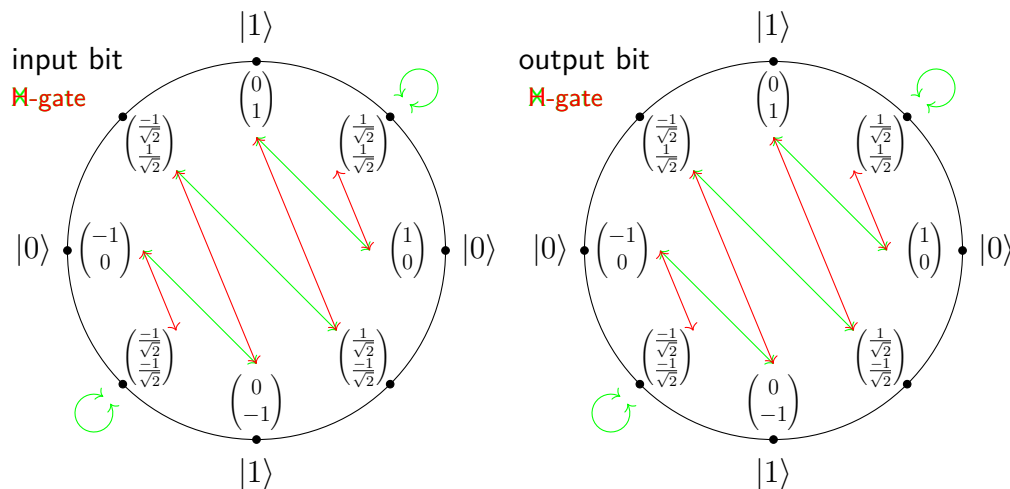
Hoogtijd om ons eerste quantumalgoritme te schrijven. Een algoritme is een opvolging van logische stappen, hier bewerkingen van een begintoestand met poorten.

- Welke twee poorten moet je toepassen om van toestand

$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$ naar $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ te komen? Er zijn twee manieren.

- Met welk algoritme kan je van toestand $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ naar toestand $\begin{pmatrix} -1 \\ 0 \end{pmatrix}$?

We zouden het helemaal kunnen uitrekenen, maar veel eenvoudiger is het om de toestandsdiagrammen te volgen. Met een aantal keer zigzaggen kom je op het antwoord.



Figuur 9.7: prepared state.

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \xrightarrow{X} \xrightarrow{H} \xrightarrow{X} \xrightarrow{H} \xrightarrow{X} \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

Wat als we een quantum computer bouwen die begint in toestand $|0\rangle$, vervolgens door een H-gate, een X-gate, en daarna weer door een H-gate en een

Deze cirkel lijkt erg op eenheidscirkel uit de wiskunde. Dat geldt voor de toestanden. Het kwadraat van de coëfficiënten levert de realisatie van de kans, daarom komen de waarden $|0\rangle$ en $|1\rangle$ ook twee keer voor. Voor onze bewerkingen hebben we voldoende aan een eenheidscirkel toestand machine. Als je complexe getallen gebruikt wordt de voorstelling een bol (Bloch-sphere). Dat staat natuurlijk meer bewerkingen toe, maar het principe blijft hetzelfde.

recap:

- qubits zijn 2-vectoren met complexe coëfficiënten. Cbits zijn speciale gevallen van qubits (reële coëfficiënten).
- multi-bits systemen zijn tensor producten van enkele qubits. Idem voor cbits (n bits spannen een vectorruimte 2^n op
- Operaties op bits kun je met een matrix weergeven
- Qubits kunnen net als cbits weergegeven worden door matrix operaties.
- de H-gate brengt $|0\rangle$ en $|1\rangle$ in gelijke superpositie, en terug.

- elke gate is zijn eigen inverse.
- Een gate twee keer (direct achter elkaar) uitgevoerd levert het origineel.
- Bits en operatoren kunnen op de eenheidscirkel (blochbol) worden weergegeven als state machine.

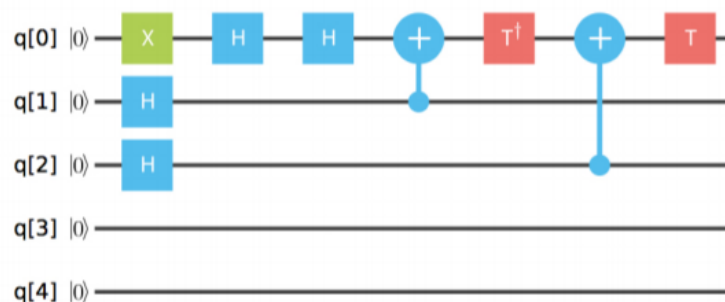
NB. Een MZI kan je nu weergeven als een quantumcircuit. Spiegel is een bitflipper

$$H(X(H|0\rangle)) = |1\rangle$$

9.1 algoritmen, notatie

We gebruiken lineaire algebra en state machines om de werking van poorten te beschrijven. Poorten zijn weliswaar de bouwstenen van computers, maar daarmee heb je nog geen programma, of algoritme. Een algoritme is een logisch geordende set rekenregels die, toegepast op een beginsituatie, tot een resultaat leidt. De rekenregels bestaan uit opeenvolging instructies waarmee uiteindelijk logische mee worden aangestuurd. Er zijn miljoenen regels code voor nodig om bijvoorbeeld een digitale televisie te laten werken.

Algoritmen voor quantumcomputers zijn niet groot. Met een handje vol instructies achter elkaar is het al mogelijk om complexe berekeningen uit te voeren zie fig. 9.2. Het aantal parallelle bewerkingen (het aantal qubits) bepaalt de kracht van het algoritme. In de schematische notatie houdt je de toestand van alle qubits bij. De individuele qubits zijn parallelle registers.



Figuur 9.8: Deel van een grafische voorstelling van een quantumalgoritme.

Het schema lees je van links naar rechts. Het LSB staat meestal bovenaan. Op het LSB worden de meeste bewerkingen gedaan. De meeste gates werken op een bit. De CNOT is de uitzondering deze poort werkt op twee bits en heeft dus twee in- en uitgangen.

We behandelen twee toepassingen.

Deutsch oracle

Je krijgt een cadeau, een grote zwarte doos die een van de functies (S0, S1, I en X) op één bit kan uitvoeren. Je weet niet welke functie in de doos zit, maar je kan inputs proberen en de outputs observeren. Hoeveel ondervragingen moet je doen om er achter te komen welke functie er in de doos zit op een klassieke computer? Hoeveel op een quantum computer?

klassiek: twee metingen nodig: stuur een $|0\rangle$ in kijk wat er uit komt, Stuur een $|1\rangle$ in en kijk wat er uit komt. Op een quantumcomputer zijn het er ook twee, want je hebt twee bits nodig om vier dingen te onderscheiden, maar ...

Als je wilt weten of de functie constant (S0 en S1) is of variabel (I en X). Hoeveel ondervragingen heb je nodig op een klassieke computer en hoeveel op een quantum computer?

Klassiek zijn hiervoor nog steeds twee vragen nodig, maar in quantum maar één. Als we dit hebben aangetoond, hebben we ons eerste succes van de quantumcomputer boven de klassieke computer aangetoond. David Deutsch toonde het in 1985 aan ¹

We gebruiken de kracht van superpositie.

Hoe zien de functies er uit op een quantumcomputer? We lopen gelijk tegen een probleem aan. De poorten S0 en S1 Deze zijn niet reversibel, en werken dus niet in een quantumcomputer.

We lossen het op door een simulatie van de black box op een quantumcomputer te bouwen. In de simulator gebruiken we een twee bits quantum computer. Het controle register blijft onveranderd in de simulator. Het doel register kan door de blackbox simulator gewijzigd worden.

¹D. Deutsch Proc. R. Soc. Lond. A 400, 97-117 (1985) Quantum theory, the Church-Turing principle and the universal quantum computer

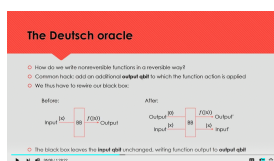
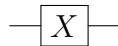
We simuleren de mogelijke uitkomsten vna de box met reversibele poorten. De vier toestanden die we willen ontdekken zijn Set-0, Set-1, identiteit en not. We bieden een $|0\rangle$ aan op het target kanaal, een testsignaal op het control, en we bekijken wat het target signaal levert, en negeren het control signaal, dat onveranderd de testmachine verlaat.

==--==--

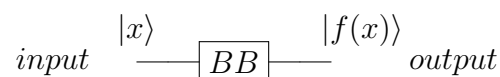
Hoe maak je niet-reversibele functies reversibel? Oplossing: voeg een qubit toe waarop de functie wordt toegepast.

[video t=36:08]

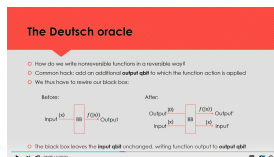
$$|x\rangle \rightarrow f(|x\rangle)$$



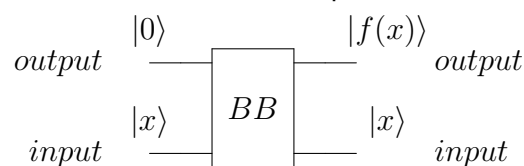
Figuur 9.9: een extra bit.



We bouwen een testmachine met qubits (fig.). De machine heeft twee qubits. Eén om de input bij te houden en één om de output bij te houden. Op het MSB (input) staat een testwaarde, op het LSB (output) bieden we $|0\rangle$ aan. We laten het input bit onveranderd, en lezen de functiewaarde van het output-bit.

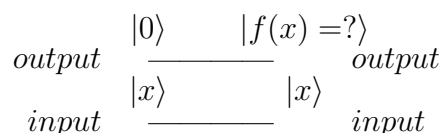


Figuur 9.10: een extra bit.



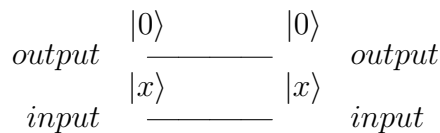
We testen vervolgens onze functies

==--==--



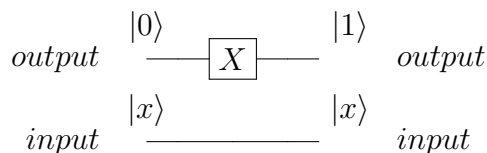
Set-0

Voor een set-0 hoeven we alleen de daadjes door te verbinden.
Controleer dat $|0\rangle \rightarrow |0\rangle$ en $|1\rangle \rightarrow |0\rangle$



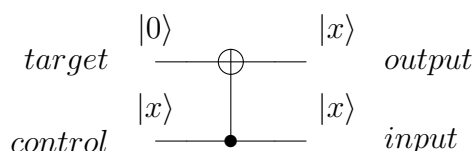
Set-1

Voor een set-1 hoeven we alleen een NOT in het doel-kanaal op te nemen. Controleer dat $|0\rangle \rightarrow |1\rangle$ en $|1\rangle \rightarrow |1\rangle$



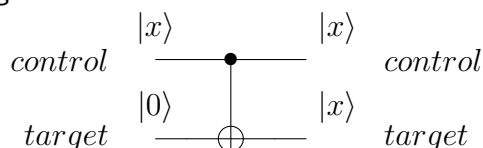
Identiteit

We zetten nu de CNOT in. De CNOT gate heeft een sleutelfunctie, hiermee kunnen we bits verstrngelen. Deze poort werkt op twee bits. Het LSB is het controlebit, dat het LSB, het target (doel) bestuurt. Als het controlbit $|0\rangle$ is, dan blijft het target $|0\rangle$ onveranderd. Als het controlbit $|1\rangle$ is dan flipt het targetbit $|1\rangle$. In een schema betekent het zwarte bolletje het controlbit, het plus



$ 00\rangle$	\xrightarrow{CNOT}	$ 00\rangle$
$ 01\rangle$	\xrightarrow{CNOT}	$ 11\rangle$
$ 10\rangle$	\xrightarrow{CNOT}	$ 10\rangle$
$ 11\rangle$	\xrightarrow{CNOT}	$ 01\rangle$

Andersom mag ook



Tabel 9.1:
waarheidstabel
van CNOT.

We bieden op het targetbit altijd een $|0\rangle$ aan. Als op het controlbit een $|0\rangle$ wordt aangeboden, verandert het outputbit niet. Was een $|0\rangle$, blijft een $|0\rangle$. We vergelijken de output met het controlbit. Ze

zijn hetzelfde. check. Als op het controlbit een $|1\rangle$ wordt aangeboden, verandert het outputbit. Was een $|0\rangle$, blijft een $|1\rangle$. We vergelijken de output met het controlbit. check.

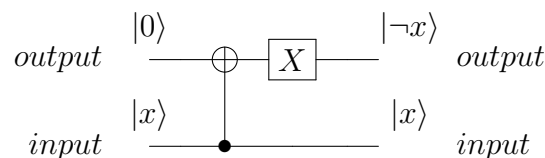
Bitflip

De laatste mogelijkheid die in de Black Box kan zitten is dat de input altijd 'geNOT' wordt, dus

$$|0\rangle \rightarrow |1\rangle$$

$$|1\rangle \rightarrow |0\rangle.$$

Deze functie is makkelijk uit de identiteit te maken. Gewoon een X-gate toevoegen op de contraollijn.

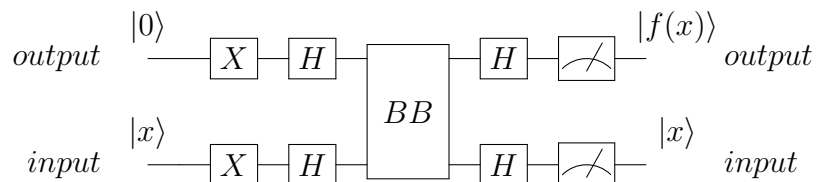


9.1.1 nu de quantum computer

Nu met de quantumcomputer: we bereiden het signaal voor voordat we het aan de input aanbieden.

- We bieden twee klassieke bits aan in $|0\rangle$ op zowel input als output.
- Beide kanalen gaan door een X-gate zodat ze beide $|1\rangle$ zijn
- daarna gaan ze ieder door een H-gate waardoor ze in dezelfde superpositie komen.
- Met deze twee qubits testen we de BB. In de BB komen ze een van de vier testcircuits tegen.
- De qubits gaan weer door een H-gate
- Waarna een meting volgt.

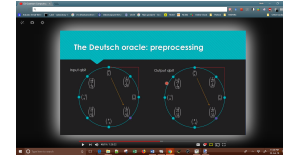
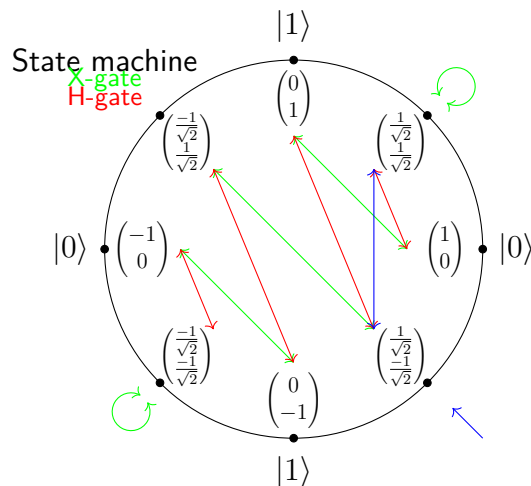
Het signaal op het bovenste kanaal is nu niet langer altijd $|0\rangle$.



Claim:

- Als de BB functie constant is, dan is de output $|11\rangle$ na meting
- Als de BB functie variabel is, dan is de output $|01\rangle$ na meting

NB input is het MSB



Figuur 9.11: preprocessing.

Figuur 9.12: State machine voor X-gate en H-gate $\alpha = 0, \beta = 1$.

Er is een spiegelsymmetrie voor elke poort. Net als bij een spiegelopertatie is een poort zijn eigen inverse. Alle 1-bit operaties kun je doen door over de eenheidscirkel te hoppen.

Probleem met de S0 en S1 gate: zij zijn niet reversibel. Truc: introduceer een extra bit. De nieuwe BB dlaat het input qubit onveranderd,, de functiewaarde wordt naar output geschreven.

Je kunt de BB met 4x4 matrices representeren, maar je kunt ook state machines gebruiken. wij doen het laatste. We bieden $|00\rangle$ aan op beide kanalen. Aan de ingang van de BB komt op beide kanalen $\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix}$ te staan. (zie pijl in fig.). We lopen de vier functies af:

S0

Laat de beide kanalen ongewijzigd. De H-gate na de BB beeldt de zowel de output als de input op 1 af.

Resultaat: $(0,0) \xrightarrow{S0} (1,1)$

S1

Het output kanaal wordt in de BB geflipt naar $\begin{pmatrix} \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$, de H-gate beeldt dit vervolgens op $\begin{pmatrix} 0 \\ -1 \end{pmatrix}$ af. Voor een meting moeten we het kwadraat van de abs. waarde nemen: 1. Met het inputkanaal gebeurt niets in de BB. Na de BB nog een H-gate toepassen: levert: $|1\rangle$

Resultaat: $(0,0) \xrightarrow{S^0} (1,1)$

Identiteit

Dit is wat meer werk. Hiervoor moeten we de CNOT operatie nader bekijken. CNOT werkt op twee bits. Voor klassieke bits weten we al wat de CNOT operatie op levert, maar voor qubits nog niet. Laten we het gewoon uitschrijven voor onze input. Het onderste,, input bit, het MSB, is het control bit. Dat blijft onveranderd, en verlaat de BB als $\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix}$ Het bovenste, output bit, het LSB, is het target bit.

De actie van CNOT is in het algemeen niet in de state machine weer te geven omdat de actie afhankelijk is van twee bits. Voor de onze situatie kunnen we het uitrekenen:

Het tensorproduct mag je niet omdraaien. Het is niet commutatief. De gebruikte vorm is $MSB \otimes LSB$. Nu ja, in dit geval zijn ze gelijk.

$$C \left(\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix} \otimes \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix} \right) = C \begin{pmatrix} \frac{1}{2} \\ \frac{-1}{2} \\ \frac{-1}{2} \\ \frac{1}{2} \end{pmatrix} =$$

$$\frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix} \otimes \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix}$$

Dit resultaat moet nog door een Hadamard poort geleid worden. In fig. 9.6 staat het state diagram van een H-poort.

Voor het MSB: $\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \xrightarrow{H} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$

Voor het LSB: $\begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \xrightarrow{H} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$

De identiteit levert dus het resultaat $|01\rangle$.

bitflip

. De voorbereiding tot de BB is weer als bij alle andere gevallen. In de BB wordt net als bij de identiteit eerst een CNOT toegepast. Met de input gebeurt verder niets en levert uiteindelijk bij de meting $|0\rangle$ op. Het LSB wordt na de CNOT nog extra door een bitflip gehaald.

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \xrightarrow{X} \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \xrightarrow{H} \begin{pmatrix} 0 \\ -1 \end{pmatrix} = |1\rangle.$$

Na de laatste H-gate wordt de output op $|1\rangle$ afgebeeld.

De bitflip levert als resultaat: $|01\rangle$

We hebben net een eigenschap ontsloten die we met een klassieke computer niet zo efficiënt zouden kunnen doen. Deutsch' algoritme is veralgemeniseerd. Deze generalisering was op zijn beurt weer de basis voor Shor's algoritme.

recap

- we modelleren klassieke computers met lineaire algebra
- qbits, superposition, quantum gates
- Deutsch' Oracle, een quantum algoritme dat klassiek achter zich allt.

Verstrengeling

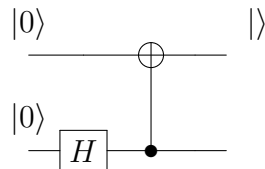
video [58 min] Stel we hebben twee qubits in de volgende toestand

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} \text{ Voor de coëfficiënten geldt: } ac = \frac{1}{\sqrt{2}} \text{ } ad = 0$$

$$bc = 0 \text{ } bd = \frac{1}{\sqrt{2}}$$

Dit stelsel is niet oplosbaar. Je kunt het niet ontbinden in twee bits. De toestanden zijn dan verstrengeld. De coëfficiënten leren je dat dit een even grote kans om ineen te storten tot $|00\rangle$ en $|11\rangle$.

Zo'n toestand is eenvoudig te maken. (gechecked, werkt niet!)

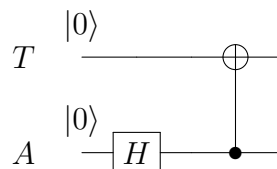


$$C \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes H \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) \right) = C \left(\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) =$$

verstrengeling

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

Dit systeem kan niet gefactoriseerd worden. De bits zijn verstrengeld. Als het gemeten wordt vervalst het met 50/50 $|00\rangle/|11\rangle$



- de bits zijn verbonden, zelfs op grote afstand
- als je een meet, vervalst de andere toestand ook. Je weet hoe de ander vervalst.
- Het verval van de andere bit is instantaan, sneller dan het licht dus.
- Hoe de deeltjes vervallen ligt per se niet vast.
- de hidden variable theorie is in 1964 weerlegd door Bell
- Breekt dit localiteit? informatie kan niet worden overgezonden. Daarvoor is een klassiek bit nodig, dat hooguit met de lichtsnelheid kan reizen. Informatie kan dus niet sneller reizen dan de lichtsnelheid.

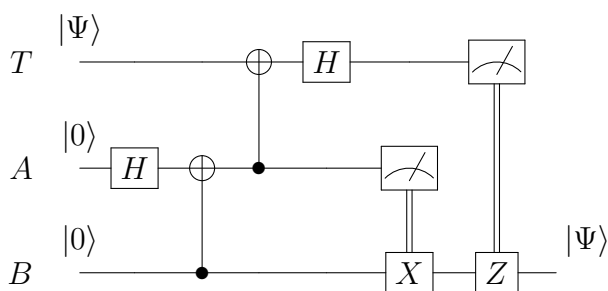
Nu gaat er een wereld open. Twee qubits zijn verstrengeld. Maakt niet uit op welke afstand. Als een van hen gemeten wordt valt de

toestand van de ander instantaan op zijn plek (experiment atoom-klokken 2013?).

Spooky action at a distance. EPR

Teleportatie [dit is niet goed uitgewerkt in de presentatie] Teleportatie is het proces waarbij een qubit van de ene plaats naar een ander wordt gebracht d.m.v. twee andere verstrengelde qubits.

Je kunt wel qubits verplaatsen (knippen en plakken) maar niet kopiëren. Teleportatie is niet sneller dan het licht omdat voor informatieoverdracht klassieke bits nodig zijn.



Het teleportatie schema bestaat uit drie kanalen. Alice heeft twee qubits, Ψ en A en Bob heeft ook een qubit B . Zij verstrengelen qubits A en B , en houden ze uit elkaar. Qubit A wordt met Ψ verstrengeld. Het is dus een drie qubit toestand die niet te factoriseren is. Zij past een H-gate toe op Ψ , en meet de beide bits die zij tot haar beschikking heeft. (A en Ψ). Dit levert haar twee klassieke bits. Dit zijn de bits die zij naar Bob moet sturen. De uitkomst van bit A bepaalt of hij zijn antwoord door een Z-gate moet halen, de uitkomst van bit Ψ bepaalt of hij zijn bit door een X-gate moet halen. (Z-gate is phase flip nog niet uitgewerkt).

NB bij entangled state: De reden waarom het exponentieel veel geheugen kost om quantum computers te simuleren is omdat entangled states niet geëvalueerd kunnen worden. Anders gezegd, in klassieke termen neemt een qubit veel ruimte in beslag, exponentieel veel.

[video=1h15]Further learning goals

- Deutsch-Jozsa algoritme (n-bit generalisation of Deutsch)
- Simon's periodiciteitsprobleem
- Shor's algoritme

- Grover's algoritme
- quantum cryptografische sleuteluitwisseling (niet zo moeilijk)
- Fysische realisatie van quantum bits en gates
- Error correction
- quantum languages

literature

- quantum computing for computer scientists
- Quantum Computing: a gentle introduction
- Quantum computer science: an introduction (heavy math)
- microsoft quantum development kit
- Some scpeticism about development of quantum computers: noise might increase eponentially with the number of qubits.

9.2 superposition

9.3 after thoughts

literature

noise

Qsharp demonstration of Deutch

Dwave

Delft website says that there is an optimum number of runs $\pi \times \sqrt{N}$ or so.

notes

15 jan 2020 Vedran: bloch sphere redundant, nieuw model (blokjes over draadjes (artikel)

15 jan 2020 [universiteit twente](#)



9.4 Quantum Inspire

Ik heb de circuits getest met quantum Inspire, een simulator voor quantum circuits van de TUDelft. Je kunt deze simulator gebruiken zonder in te loggen. Je hebt dan tot maximaal vijf qubits beschikbaar, meer dan genoeg voor ons.

Als bij een simulatie geen meting wordt aangegeven (het symbool met het metertje), dan wordt het schema doorgerekend volgens de gegeneraliseerde kans. Het is voldoende om het schema één keer uit te voeren. Als je wel een meetblok opneemt in het schema, dan komt het kansproces in het spel. Als je het schema nu uitvoert krijg je een realisatie van een monte carlo statistisch behandeld. Het aantal runs is dan van belang.

Dus als je geen meetblokje opgeeft, rekent het systeem een ideale deterministische simulatie door. $N=1$ werkt prima. Als je wel een meetblokje opgeeft, is de keuze van N van belang. Bij $N=1$ krijg je een realisatie.

Bij ingewikkelder problemen ligt de keuze van N niet altijd voor de hand.

Bijvoorbeeld het Grover algoritme op [Quantum inspire](#).



9.5 Mechanical view of reality

Video serie [1] van in totaal 8 filmpjes van Richard Feynman workshop over Mermin's apparaat. heel filosofisch, wat is de logica van de quantum wereld? Brownse beweging (tandpasta heeft glimende kristallen, is macroscopisch. Heisenberg staat centraal Welke vraag kan je stellen bij Schrodinger's kat? part 1.3 9: 40 citeerd John Wheeler: "A phenomenon isn't a phenomenon until it is an observed phenomenon".

Bell's theorem: als de resultaten van het exp 2/3 op zou leveren, zou het allemaal klassiek te verklaren zijn, maar ze leveren 3/4 oop. Het is een bewijs uit het ongerijmde. Er kan een klassiek schema bestaan waarmee dit resultaat verklaard kan worden.

9.6 extra plaatjes, stijlvoorbeelden

voorbeelden van stijl

Voetnoot² en hyperlink doe je zo. \url gebruik je voor een externe link www.quantumrules.nl.

²[http://refractiveindex.info/?shelf=glass&book=BK7&page=](http://refractiveindex.info/?shelf=glass&book=BK7&page=SCHOTT)
SCHOTT

in text $|\uparrow\downarrow\swarrow\rangle$ ziet het er zo uit

in een $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ regel tekst in een regel $\begin{pmatrix} a \\ c \end{pmatrix}$ tekst in een $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ regel
 tekst in een regel $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ tekst in een regel $\begin{pmatrix} a \\ c \end{pmatrix}$ tekst

Bibliografie

- [1] Richard Feynman. *Quantum Mechanical View of Reality - Richard Feynman*. URL: https://www.youtube.com/playlist?list=PLW_HsOU6YZRkdhFFznHNEfua9NK3deBQy.