CS290A, Spring 2005:

Quantum Information & Quantum Computation

Wim van Dam

Engineering 1, Room 5109
vandam@cs

http://www.cs.ucsb.edu/~vandam/teaching/CS290/

Hadamard Transfrom

- Define the Hadamard transform: $$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$
- We have for this H:

$$\begin{aligned} |0\rangle &\mapsto \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle &\mapsto \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\ \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) &\mapsto |0\rangle \\ \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle) &\mapsto |1\rangle \end{aligned}$$

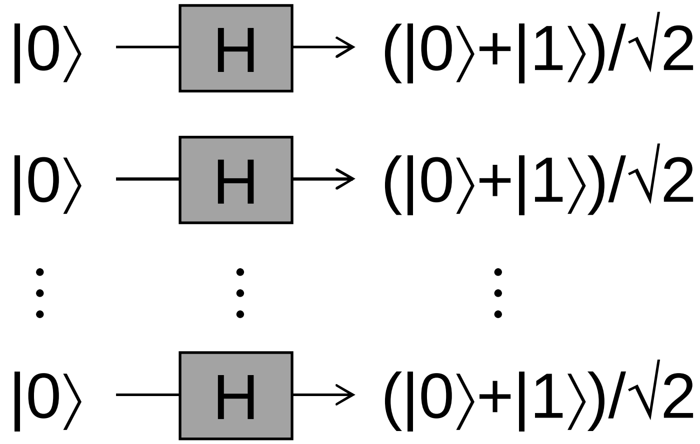
- Note: H^2 = Id.
  It changes classical bits
  into superpositions
  and vice versa.

- It sees the difference between the uniform superpositions $(|0\rangle+|1\rangle)/\sqrt{2}$ and $(|0\rangle-|1\rangle)/\sqrt{2}$.

Hadamard as a Quantum Gate

- Often we will apply the H gate to several qubits.
- Take the n-zeros state $|0,\ldots,0\rangle$ and perform in parallel n Hadamard gates to the zeros, as a circuit:

Starting with the all-zero state and with only n elementary qubit gates we can create a uniform superposition of 2^n states.



Typically, a quantum algorithm will start with this state, then it will work in "quantum parallel" on all states at the same time.

As an equation:

$$|0,\ldots,0\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$$

Combining Qubits

If we have a qubit $|x\rangle = (|0\rangle+|1\rangle)\sqrt{2}$, then 2 qubits $|x\rangle$ give the state $\frac{1}{2}(|00\rangle+|01\rangle+|10\rangle+|11\rangle)$.

Tensor product notation for combining states $|x\rangle\in\mathbb{C}^N$ and $|y\rangle\in\mathbb{C}^M$: $|x\rangle\otimes|y\rangle = |x\rangle|y\rangle = |x,y\rangle \in \mathbb{C}^{NM}$.

Example for two qubits: $(\alpha_0|0\rangle+\alpha_1|1\rangle)\otimes(\beta_0|0\rangle+\beta_1|1\rangle)$

$$= \alpha_0\beta_0|00\rangle+\alpha_0\beta_1|01\rangle+\alpha_1\beta_0|10\rangle+\alpha_1\beta_1|11\rangle$$

Note that we *multiply* the amplitudes of the states.

Also note the exponential growth of the dimensions.

Braket Calculus

- See handout “Mathematics of Quantum Computation”

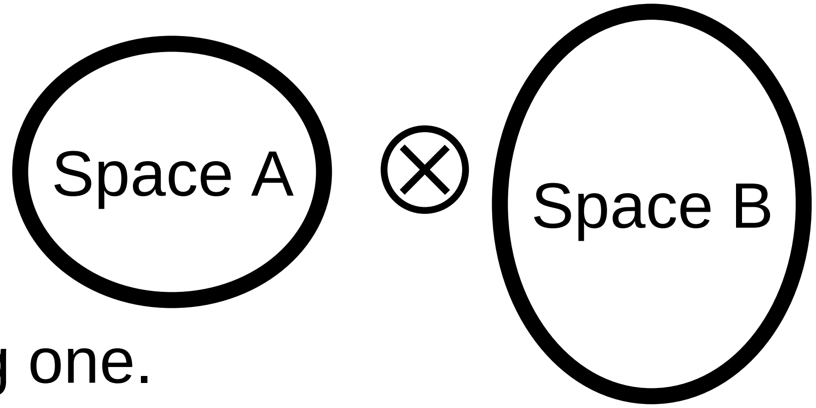
- To get familiar with the braket notation:
  Find patterns like $(A\otimes B)(C\otimes D) = AC\otimes BD$,
  Calculate ‘small’ examples in matrix notation;
  Prove the general case using braket notation.

- See exercises in Chapter 2-2.1.7 in Nielsen&Chuang.

- Specific exercises will be announced this Friday.

The Tensor Product

- Keep in mind the picture

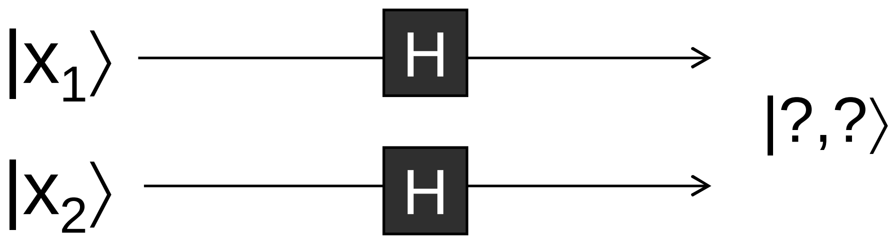


- The tensor product glues two subspaces to one big one.

- Often states and operations in this big space can not be represented as a tensor product.
  Example for a 2 qubit state space:
  Entangled qubits: $(|0,0\rangle+|1,1\rangle)/\sqrt{2} \neq |\psi\rangle\otimes|\varphi\rangle$
  Joint Operations: $\text{CNOT} \neq U\otimes W$

Two Hadamard Gates

What does this circuit do on {00,01,10,11}?



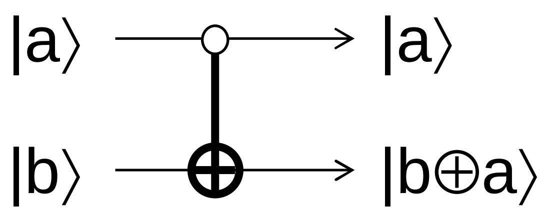
$$|x_1, x_2\rangle \mapsto \frac{1}{2} \sum_{(y_1,y_2)\in\{0,1\}^2} (-1)^{x_1 y_1 + x_2 y_2} |y_1, y_2\rangle$$

Controlled NOT Gate

- Define the 2 qubit gate CNOT by
- Depending on the first **control** bit, the gate applies a NOT to the second, **target** qubit.

$$
\begin{aligned}
|0,0\rangle &\mapsto |0,0\rangle \\
|0,1\rangle &\mapsto |0,1\rangle \\
|1,0\rangle &\mapsto |1,1\rangle \\
|1,1\rangle &\mapsto |1,0\rangle
\end{aligned}
$$

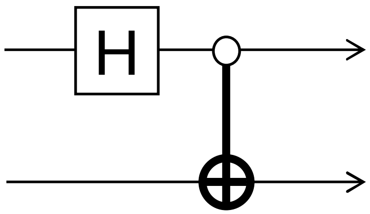
- Circuit notation:

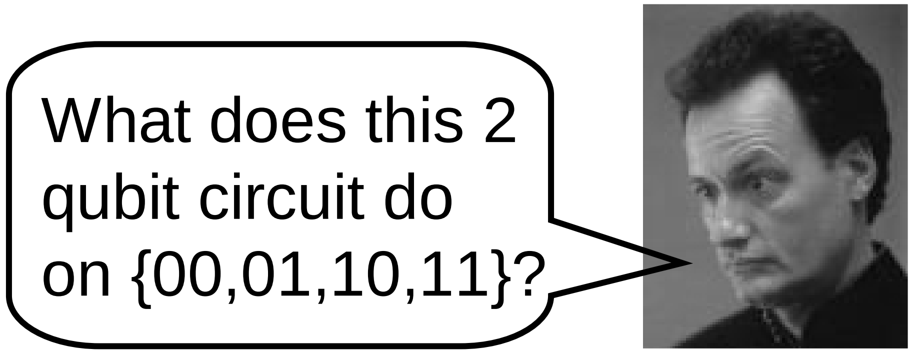


- Note that $b \oplus 1 = \mathrm{NOT}(b)$
- As a matrix

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

Hadamard + CNOT Gate





Answer for the four basis states:

$$|0,0\rangle \mapsto \tfrac{1}{\sqrt{2}}(|0,0\rangle + |1,1\rangle)$$
$$|0,1\rangle \mapsto \tfrac{1}{\sqrt{2}}(|0,1\rangle + |1,0\rangle)$$
$$|1,0\rangle \mapsto \tfrac{1}{\sqrt{2}}(|0,0\rangle - |1,1\rangle)$$
$$|1,1\rangle \mapsto \tfrac{1}{\sqrt{2}}(|0,1\rangle - |1,1\rangle)$$

Note that the output states are not tensor products of 2 qubits. Instead the qubits are *entangled*.

The Pauli Matrices

Four elementary single qubit gates, including the NOT gate and the Identity.

$$\sigma_0 = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\sigma_1 = \sigma_x = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\sigma_2 = \sigma_y = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

$$\sigma_3 = \sigma_z = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Exercises:

- What other gates can you make with these gates?
- Play around with them and see how these gates "anti-commute".

Some more Gates

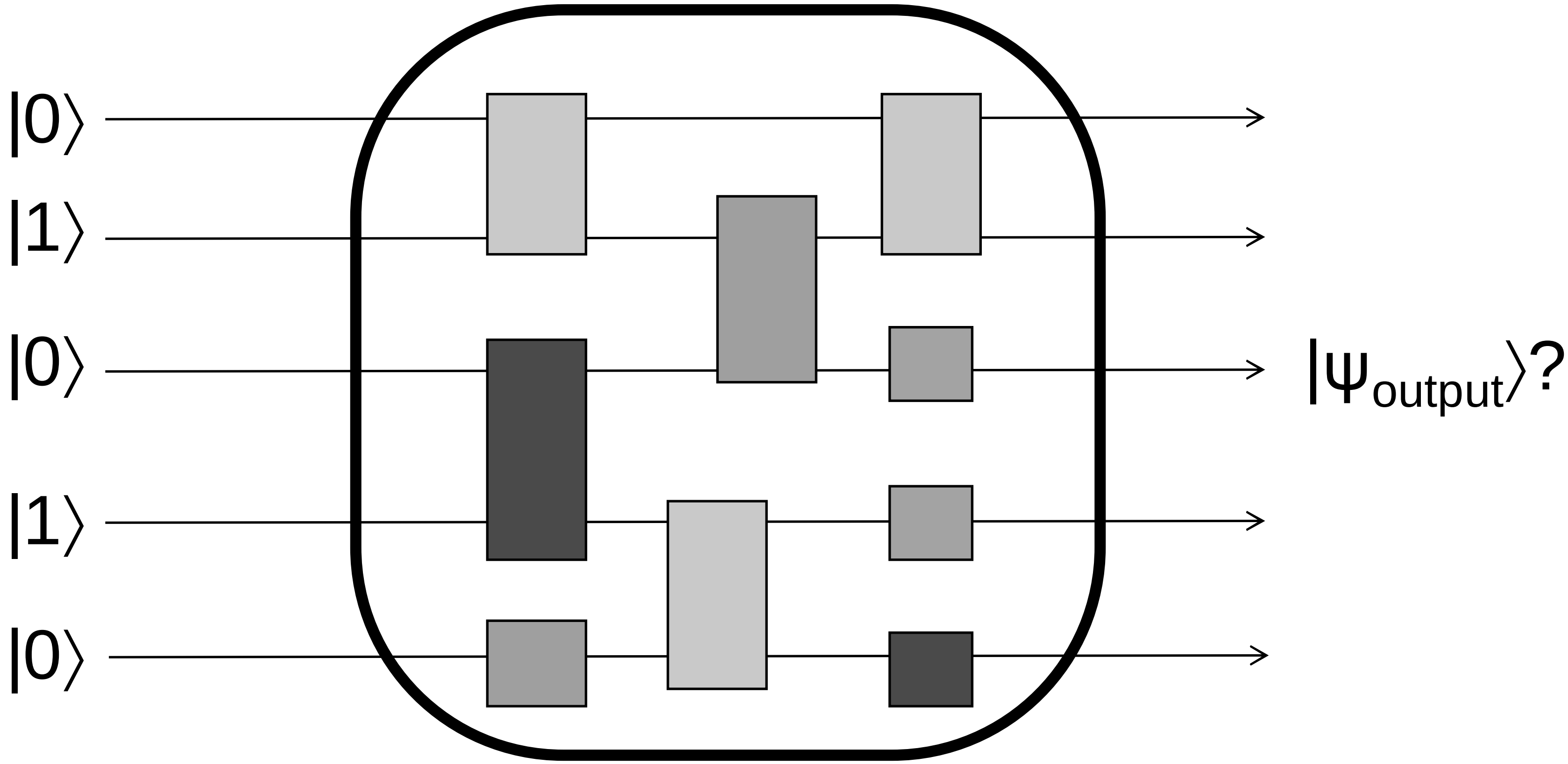
- Controlled-Controlled-NOT gate CCNOT:
  CCNOT:$|a,b,c\rangle \mapsto |a,b,c\oplus(ab)\rangle$ for all $(a,b,c)\in\{0,1\}^3$

- Single qubit (–1)-Phase Flip: $\alpha|0\rangle+\beta|1\rangle \mapsto \alpha|0\rangle-\beta|1\rangle$
- Single qubit φ-Phase Flip: $\alpha|0\rangle+\beta|1\rangle \mapsto \alpha|0\rangle+e^{i\varphi}\beta|1\rangle$

- Controlled-φ-Phase Flip: $|a,b\rangle \mapsto e^{i\varphi ab}|a,b\rangle$
  for all $(a,b)\in\{0,1\}^2$.

- And so on…

Quantum Circuits



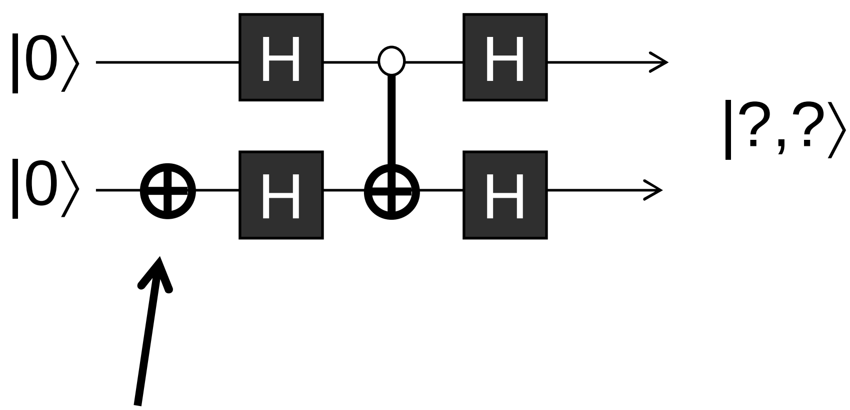
- Start with n classical bits as input.
- Apply a sequence of elementary gates
- Measure the outcome ψ_{output}.

Quantum Circuit Complexity

- Given an input size of $|x|=n$ (classical) bits,
  we apply a quantum circuit C_n to the input $x \in \{0,1\}^n$.
- Afterwards, we measure the output state ψ in the classical, computational basis $\{0,1\}^n$.
- The **outcome** of the quantum circuit algorithm is the probability distribution of ψ over $\{0,1\}^n$.
  (Typically this favors a specific string $\in \{0,1\}^n$.)
- The quantum circuit algorithm is efficient if the size of the circuits grows polynomially in n: $\text{size}(C_n) = \text{poly}(n)$.

Hadamard + CNOT Gate

What does *this* 2 qubit circuit do?



"single qubit NOT gate"

Quantum Computing

*The **superposition principle** in combination with the **interference phenomenon** of 'complex probabilities' makes it hard to compute the behavior of say 1000 qubits.*

We have no proof of this (yet), but we suspect that this task is inherently hard.

A 1000 qubit quantum computer would perform this computation efficiently.

classically

"quantumly"