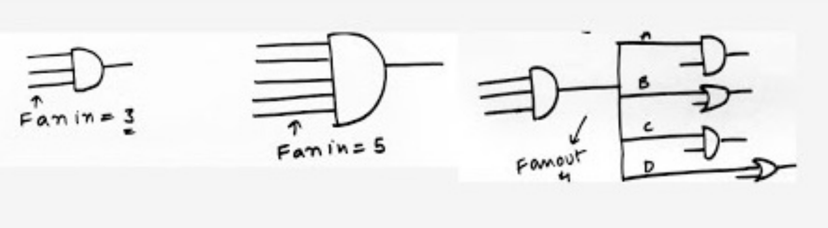
CS6846 – Quantum Algorithms and Cryptography

Going beyond Classical

Instructor: Shweta Agrawal, IIT Madras
Email: shweta@cse.iitm.ac.in

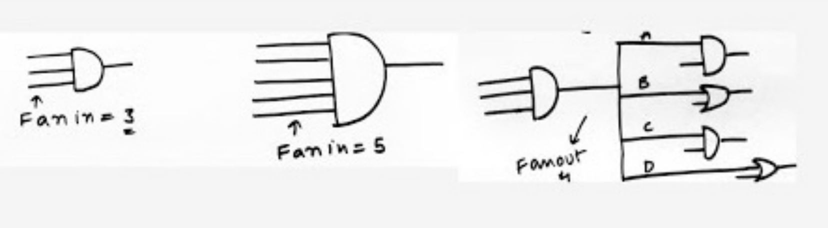
CCNOT or Toffoli Gate

Recall the definition of fanin and fanout



CCNOT or Toffoli Gate

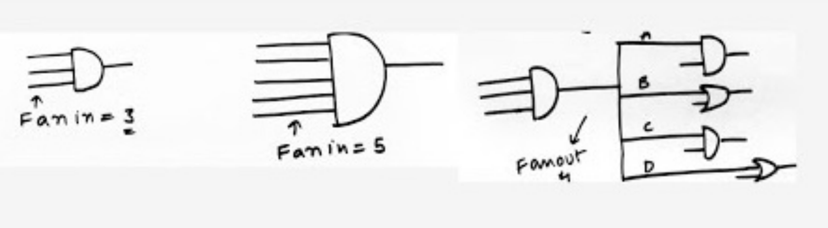
Recall the definition of fanin and fanout



Claim: Toffoli gate can be used to simulate fanout

CCNOT or Toffoli Gate

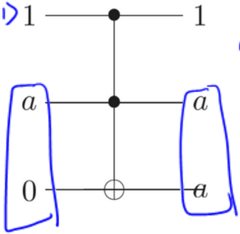
Recall the definition of fanin and fanout



Claim: Toffoli gate can be used to simulate fanout

$\alpha|0\rangle + \beta|1\rangle \Rightarrow \alpha|00\rangle + \beta|11\rangle$

$\neq (\alpha|0\rangle + \beta|1\rangle) \otimes (\alpha|0\rangle + \beta|1\rangle).$



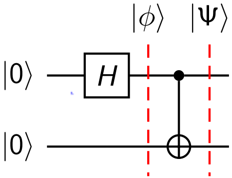
$a = |0\rangle$, then I get $|0\rangle|0\rangle$

$a = |1\rangle$ then I get $|1\rangle|1\rangle$

Contradiction to No Cloning? No.

Having fun with circuits

What state does the following circuit construct?



$$|\phi\rangle = H|0\rangle \otimes |0\rangle = |+\rangle|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)|0\rangle$$

$$|\psi\rangle = \mathrm{CNOT}\left(\frac{1}{\sqrt{2}}|00\rangle + |10\rangle\right)$$

$$= \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$$ EPR state

Having fun with EPR

$$\text{EPR} = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$$

We pass through $H \otimes H$, i.e. apply H to each qubit.

$$H \otimes H\left(\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)\right) = \frac{1}{\sqrt{2}}\left(|+\rangle|+\rangle\right) + \frac{1}{\sqrt{2}}\left(|-\rangle|-\rangle\right)$$

$$= \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) +$$

$$\frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)$$

$$= \frac{1}{2\sqrt{2}}|00\rangle + \cancel{\frac{1}{2\sqrt{2}}|01\rangle} + \cancel{\frac{1}{2\sqrt{2}}|10\rangle} + \frac{1}{2\sqrt{2}}|11\rangle$$

$$+ \frac{1}{2\sqrt{2}}|00\rangle \cancel{- \frac{1}{2\sqrt{2}}|01\rangle} \cancel{- \frac{1}{2\sqrt{2}}|10\rangle} + \frac{1}{2\sqrt{2}}|11\rangle$$

$$= \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle.$$

Capturing Classical Computation

- Want: Given classical $f : \{0,1\}^n \to \{0,1\}$, want to evaluate using quantum system.

Capturing Classical Computation

- Want: Given classical $f : \{0,1\}^n \rightarrow \{0,1\}$, want to evaluate using quantum system.
- Want: A unitary U such that $U|x\rangle = |f(x)\rangle$.

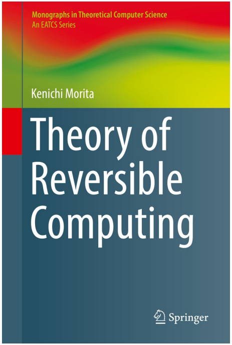
Capturing Classical Computation

- Want: Given classical $f : \{0,1\}^n \to \{0,1\}$, want to evaluate using quantum system.
- Want: A unitary U such that $U|x\rangle = |f(x)\rangle$.
- Note: f must be bijective for such a unitary to exist.

Capturing Classical Computation

- Want: Given classical $f : \{0,1\}^n \to \{0,1\}$, want to evaluate using quantum system.
- Want: A unitary U such that $U|x\rangle = |f(x)\rangle$.
- Note: f must be bijective for such a unitary to exist.

If f is bijective, the computation can be *reversed*.



Reversible Computation

Reversible Gates

A Boolean gate G is said to be reversible if it has the same number of inputs as outputs, and its mapping from input strings to output strings is a bijection.

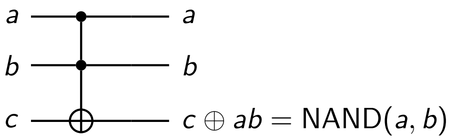
Reversible Computation

Reversible Gates

A Boolean gate G is said to be reversible if it has the same number of inputs as outputs, and its mapping from input strings to output strings is a bijection.

Reversible circuits can be implemented by unitary transformations – to reverse computation, run the circuit in inverse (or apply inverse of unitary).

Reversible Computation



Input	Output
$\|000\rangle$	$\|000\rangle$
$\|001\rangle$	$\|001\rangle$
$\|010\rangle$	$\|010\rangle$
$\|011\rangle$	$\|011\rangle$
$\|100\rangle$	$\|100\rangle$
$\|101\rangle$	$\|101\rangle$
$\|110\rangle$	$\|111\rangle$
$\|111\rangle$	$\|110\rangle$

Figure: The CCNOT (Toffoli) Gate is reversible implementation of NAND

Reversing Arbitrary Circuits

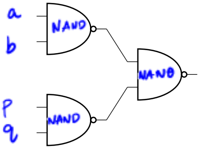
NAND gates are universal for classical computing.

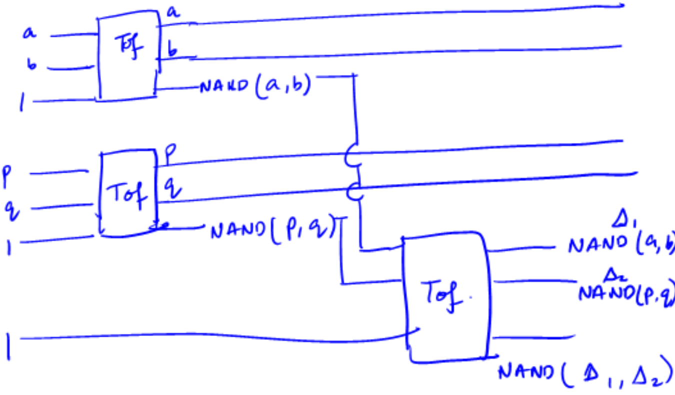
1) Eliminate OR.
DeMorgan's law:
OR (x_1, x_2) = NOT (AND (NOT(x_1), NOT(x_2)))

2). Eliminate AND
NOT (NAND)

3). Eliminate NOT.

x —[NAND]— NOT(x).
1 —





Uncomputing: Clean up ancillary bits

Consider circuit:

$$|x\rangle \otimes |0\rangle^{\otimes k} \otimes |0\rangle \longmapsto |x\rangle \otimes |0\rangle^{\otimes k} \otimes |\text{output}\rangle$$

Uncomputing: Clean up ancillary bits

Consider circuit:

$$|x\rangle \otimes |0\rangle^{\otimes k} \otimes |0\rangle \longmapsto |x\rangle \otimes |0\rangle^{\otimes k} \otimes |\text{output}\rangle$$

Uncompute $|0\rangle^{\otimes k}$ to "refresh" back to zeroes.

Uncomputing: Clean up ancillary bits

Consider circuit:

$f : \{0,1\}^n \to \{0,1\}.$

$$|x\rangle \otimes \cancel{|0\rangle^{\otimes k}} \otimes |0\rangle \longmapsto |x\rangle \otimes \cancel{|0\rangle^{\otimes k}} \otimes |\text{output}\rangle$$

Uncompute $|0\rangle^{\otimes k}$ to "refresh" back to zeroes.

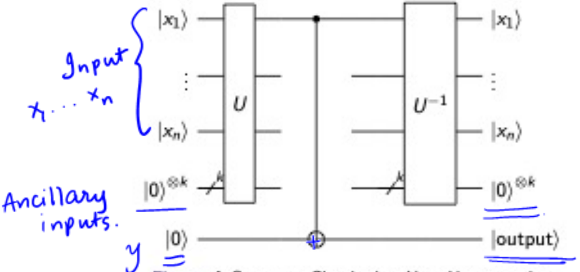


Figure: A Quantum Circuit that Uses Uncomputing

Uncomputing: Clean up ancillary bits

Consider circuit:

$$|x\rangle \otimes |0\rangle^{\otimes k} \otimes |0\rangle \longmapsto |x\rangle \otimes |0\rangle^{\otimes k} \otimes |\text{output}\rangle$$

Uncompute $|0\rangle^{\otimes k}$ to "refresh" back to zeroes.

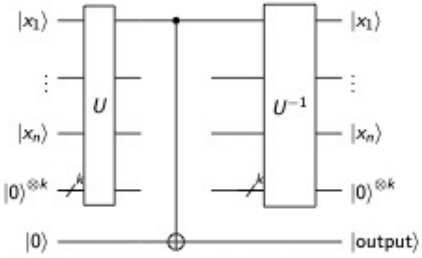


Figure: A Quantum Circuit that Uses Uncomputing

Can write circuit as $|x\rangle \otimes |0\rangle \longmapsto |x\rangle \otimes |\text{output}\rangle$

Now we can compute...

Implementing Classical Functions

A quantum circuit C_f implements a classical function $f : \{0,1\}^n \mapsto \{0,1\}^m$ if $\forall x \in \{0,1\}^n$, $\forall y \in \{0,1\}^m$,

$$C_f(|x\rangle |y\rangle \underline{|0\rangle^{\otimes k}}) \mapsto (|x\rangle |y \oplus \underline{f(x)}\rangle \underline{|0\rangle^{\otimes k}})$$

Now we can compute...

Implementing Classical Functions

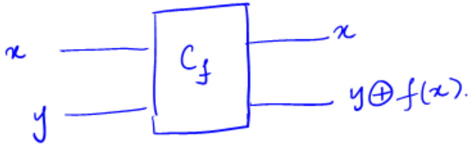
A quantum circuit C_f implements a classical function $f : \{0,1\}^n \mapsto \{0,1\}^m$ if $\forall x \in \{0,1\}^n$, $\forall y \in \{0,1\}^m$,

$$C_f(|x\rangle |y\rangle \cancel{|0\rangle^{\otimes k}}) \mapsto (|x\rangle |y \oplus f(x)\rangle \cancel{|0\rangle^{\otimes k}})$$

As mentioned, the ancillary qubits will be usually omitted and the transformation can be rewritten as $\underline{C_f(|x\rangle |y\rangle) \mapsto (|x\rangle |y \oplus f(x)\rangle}$

Classical Computation

- Claim: Can compute any deterministic classical circuit!



Classical Computation

- Claim: Can compute any randomized circuit!!

?

As we saw, can generat true randomness by using Hadamard basis states measured using classical basis.

Universal Gates for Quantum Computation

Can Toffoli be a universal gate for quantum computation?

Universal Gates for Quantum Computation

Can Toffoli be a universal gate for quantum computation?

Ans: No, just a permutation matrix. Cannot even implement Hadamard Transform.

Universal Gates for Quantum Computation

Can Toffoli be a universal gate for quantum computation?

Ans: No, just a permutation matrix. Cannot even implement Hadamard Transform.

Can {Toffoli, H} be universal gates for quantum computation?

Universal Gates for Quantum Computation

Can Toffoli be a universal gate for quantum computation?

Ans: No, just a permutation matrix. Cannot even implement Hadamard Transform.

Can {Toffoli, H} be universal gates for quantum computation?

Ans: No, these map real vectors to only real vectors so how could phase shifts and other complex transformations be supported?

Universal Gates for Quantum Computation

Can Toffoli be a universal gate for quantum computation?

Ans: No, just a permutation matrix. Cannot even implement Hadamard Transform.

Can {Toffoli, H} be universal gates for quantum computation?

Ans: No, these map real vectors to only real vectors so how could phase shifts and other complex transformations be supported?

Universal Quantum Gates

The set $\left\{\text{Toffoli(or CNOT)}, H, \begin{bmatrix} 1, 0 \\ 0, e^{i\pi/4} \end{bmatrix}\right.$ is quantum universal.

Ok, first let us see how to generate superposition of function inputs.
Apply $H^{\otimes 2}$ to $(|0\rangle, |0\rangle)$.

$$|0\rangle \text{—}\boxed{H}\text{—} |+\rangle$$
$$|0\rangle \text{—}\boxed{H}\text{—} |+\rangle$$

$$|+\rangle|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$= \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

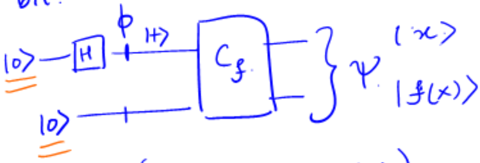
Generalize to n bits?

$$H^{\otimes n}(|0\rangle^n) = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle.$$

Quantum Parallelism

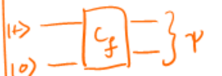
Now let us see how to generate superposition of function *outputs*.

Single bit:



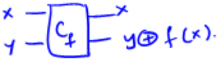
$$\phi = \frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|0\rangle\right)$$

showed



$$\psi = \frac{|0 f(0)\rangle + |1, f(1)\rangle}{\sqrt{2}}$$

If



If $x = |0\rangle$, we get (on 2^{nd} wire) $y \oplus |f(0)\rangle$. If $y = |0\rangle$, get $|f(0)\rangle$.

If $x = |1\rangle$, $y = |0\rangle$, get $|f(1)\rangle$

By linearity, if $x = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$

I get $\frac{|0\rangle|f(0)\rangle + |1\rangle|f(1)\rangle}{\sqrt{2}}$.

Remarkable state!

Quantum Parallelism

n bits.

- Prepare $n+1$ qubit state $|0\rangle^{\otimes n} \cdot |0\rangle$
- Apply $H^{\otimes n}$ on first n bits.
- Apply quantum circuit implementing f

  to get

$$\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle.$$