XIX. Simulation of the Deutsch-Jozsa Algorithm - 4 Qubit

The Deutsch-Jozsa algorithm determines whether a given function f(x): $\{0,1\}^n \rightarrow \{0,1\}$ is constant (same output for all inputs) or balanced (outputs 0 for half the inputs and 1 for the other half) **using only one query** to a quantum oracle. In contrast, **a classical algorithm might require $2^{(n-1)} + 1$ evaluations** in the worst case. This provides an exponential speedup, showcasing quantum computing's power.

$n := 4$ $\quad N := 2^n$ $\quad I := Ident(4)$

// 1. Initial state: ψ0 = |0000⟩

```
ψ0 :=  | f(i,j) ← 0
       | psi ← matrix(N,1,f)
       | psi_{0,0} ← 1
       | psi
```

// 2. Build H⊗n

```
Hn :=  | Hn ← H
       | for i ∈ 2..n
       |   Hn ← kr(Hn,H)
       | Hn
```

$$\psi_0^T = (1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0)$$

// 3. Apply Hadamard to all qubits

$$\psi_1 := Hn \cdot \psi_0$$

// 4. Define binary representation

```
Binarize(x) :=  | f(i,j) ← 0
                | B ← matrix(n,1,f)
                | for i ∈ 0..n − 1
                |   B_i ← mod(floor(x / 2^i), 2)
                | B
```

Choose Either Balanced or Constant Function Input

// Define balanced function: parity of x

$$Bf(x) := mod\left(\sum Binarize(x), 2\right)$$

Constant Function

$$Bf(x) := 0$$

4. **Construct oracle Uf**

```
Uf :=  | f(i,j) ← 0
       | Uf ← matrix(N,N,f)
       | for x ∈ 0..N − 1
       |   Uf_{x,x} ← (−1)^{Bf(x)}
       | Uf
```

// Apply Uf to state

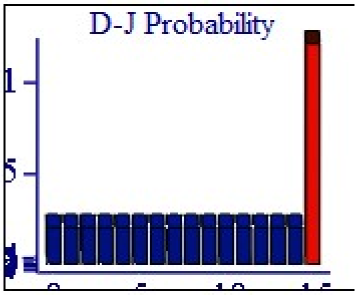
$$\psi_2 := Uf \cdot \psi_1$$

// Apply Hadamard again

$$\psi_3 := Hn \cdot \psi_2$$

// 5. Final state vector

$$\Psi_{result} = H^{\otimes 4} \cdot U_f \cdot H^{\otimes 4}\ |0000\rangle$$



6. Measurement output:

-> If all measured bits are 0: f(x) is constant
-> Otherwise: f(x) is balanced

The Deutsch–Jozsa algorithm guarantees:

P[0, 0] = 1 if the function is constant
P[0, 0] = 0 if the function is balanced (or near-zero, due to rounding)

$$\psi_{result} := \psi_3$$

// Compute probabilities

```
P :=  | f(i,j) ← 0
      | P ← matrix(N,1,f)
      | for j ∈ 0..N − 1
      |   P_j ← (ψresult_j)^2
      | P
```

ψ_result> =(1.000) |1111>

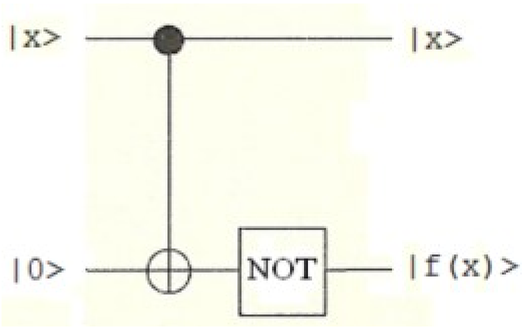
$$\psi_{result}^T = (0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1)$$

Deutsch-Jozsa Probability, P, Result

$$(P)^T = (0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1)$$

The D-J Algorithm demonstrates that there is a problem for which a QC runs faster than a Classic Computer.

Specifically, given a boolean function whose input is 1 bit f: : {0,1} --> {0,1}, is it constant?
The following circuit produces the table of results to its right. The top wires carry the value of x and the circuit places f(x) on the bottom wire. As is shown in the previous Section (XII), this circuit can also operate in parallel accepting as input all x-values and returning on the bottom wire a superposition of all values of f(x).



$$\begin{pmatrix} x & 0 & 1 & 2 & 3 \\ f(x) & 1 & 0 & 0 & 1 \end{pmatrix}$$ where

|0> = |0>|0>

|1> = |0>|1>

|2> = |1>|0>

|3> = |1>|1>

The function belongs to the balanced category because it produces 0 and 1 with equal frequency. A modification of this circuit (Ith algorithm, p.298 in *The Quest for the Quantum Computer*, by Julian Brown) answers the question of whether the function is constant or balanced. Naturally we already know the answer, so this is a simple demonstration that the circuit works.

The input is **|0>|0>|1>** followed by a Hadamard gate on each wire, as shown in the circuit shown below. As is well known the Hadamard operation creates the following **superposition states.**

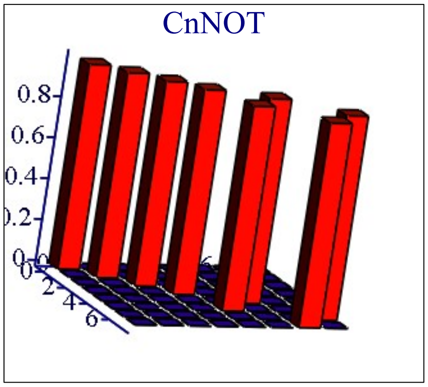
$$H\cdot\begin{pmatrix}1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\cdot\begin{pmatrix}1\\1\end{pmatrix} \qquad H\cdot\begin{pmatrix}0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\cdot\begin{pmatrix}1\\-1\end{pmatrix}$$

Therefore the Hadamard operation transforms the input state to the following three-qubit state which is fed to the quantum circuit.

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

The following matrices are required to execute the circuit. $I := \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$ $NOT := \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}$ $H := \frac{1}{\sqrt{2}}\cdot\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}$

$$CNOT := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad CnNOT := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

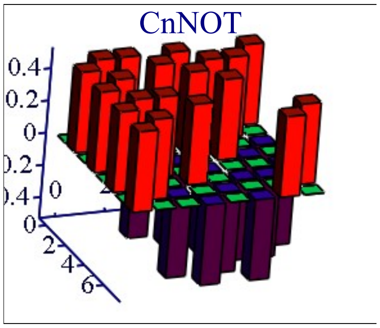


After the portion of the quantum circuit shown above, Hadamard gates are added to the top two wires, as shown in the circuit on 2nd page down. The matrix representing the circuit is assembled using tensor matrix multiplication and then allowed to operate on the wave function. The full circuit is shown below.

$$QCkt := kronecker(H, kronecker(H, I)) \cdot kronecker(I, kronecker(I, NOT)) \cdot kronecker(I, CNOT) \cdot CnNOT$$

$$QCkt = \frac{1}{2} \cdot \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 & 1 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & 0 & 1 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 & 0 & 0 & -1 \\ 1 & 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 & 0 & -1 & 1 & 0 \end{pmatrix} \qquad QCkt \cdot \frac{1}{2 \cdot \sqrt{2}} \cdot \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.707 \\ 0.707 \end{pmatrix}$$

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



Next the qubits on the top two wires are measured. If both are |0> the function is constant, but if at least one is **|1>** the function is balanced. The measurement on the top wires is implemented with projection operators **|0><0|** and **|1><1|**, and confirms that the function is not constant but belongs to the balanced category.

The first qubit is not |0>.

$$kronecker\left[\begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix}^T, kronecker(I, I)\right] \cdot QCkt \cdot \frac{1}{2 \cdot \sqrt{2}} \cdot \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

The second qubit is not |0>.

$$kronecker\left[I, kronecker\left[\begin{pmatrix}1\\0\end{pmatrix}\cdot\begin{pmatrix}1\\0\end{pmatrix}^{T}, I\right]\right]\cdot QCkt\cdot\frac{1}{2\cdot\sqrt{2}}\cdot\begin{pmatrix}1\\-1\\1\\-1\\1\\-1\\1\\-1\end{pmatrix}=\begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\0\end{pmatrix}$$

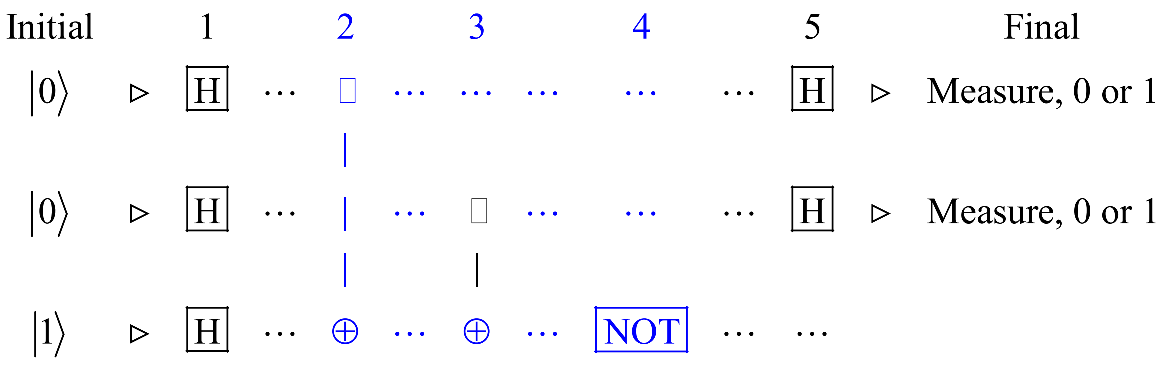
The first qubit is **|1>.**

$$kronecker\left[\begin{pmatrix}0\\1\end{pmatrix}\cdot\begin{pmatrix}0\\1\end{pmatrix}^{T}, kronecker(I, I)\right]\cdot QCkt\cdot\frac{1}{2\cdot\sqrt{2}}\cdot\begin{pmatrix}1\\-1\\1\\-1\\1\\-1\\1\\-1\end{pmatrix}=\begin{pmatrix}0\\0\\0\\0\\0\\0\\-0.707\\0.707\end{pmatrix}$$

The second qubit is **|1>**.

$$kronecker\left[I, kronecker\left[\begin{pmatrix}0\\1\end{pmatrix}\cdot\begin{pmatrix}0\\1\end{pmatrix}^{T}, I\right]\right]\cdot QCkt\cdot\frac{1}{2\cdot\sqrt{2}}\cdot\begin{pmatrix}1\\-1\\1\\-1\\1\\-1\\1\\-1\end{pmatrix}=\begin{pmatrix}0\\0\\0\\0\\0\\0\\-0.707\\0.707\end{pmatrix}$$

The following illustrates an algebraic analysis of the Deutsch-Jozsa algorithm.



$$\mathrm{H}|0\rangle\rightarrow\frac{1}{\sqrt{2}}\left[|0\rangle+|1\rangle\right]\qquad\mathrm{H}|1\rangle\rightarrow\frac{1}{\sqrt{2}}\left[|0\rangle-|1\rangle\right]$$

NOT

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

CNO

$$\begin{pmatrix} Decimal & Binary & ' & Binary & Decimal \\ 0 & 00 & ' & 00 & 0 \\ 1 & 01 & ' & 01 & 1 \\ 2 & 10 & ' & 11 & 3 \\ 3 & 11 & ' & 10 & 2 \end{pmatrix}$$

CnNO

$$\begin{pmatrix} Decimal & Binary & ' & Binary & Decimal \\ 0 & 000 & ' & 000 & 0 \\ 1 & 001 & ' & 001 & 1 \\ 2 & 010 & ' & 010 & 2 \\ 3 & 011 & ' & 011 & 3 \\ 4 & 100 & ' & 101 & 5 \\ 5 & 101 & ' & 100 & 4 \\ 6 & 110 & ' & 111 & 7 \\ 7 & 111 & ' & 110 & 6 \end{pmatrix}$$

$$|001\rangle$$

$$\mathrm{H}\otimes\mathrm{H}\otimes\mathrm{H}$$

$$\frac{1}{\sqrt{2}}\left[|0\rangle+|1\rangle\right]\frac{1}{\sqrt{2}}\left[|0\rangle+|1\rangle\right]\frac{1}{\sqrt{2}}\left[|0\rangle-|1\rangle\right]=\frac{1}{2\sqrt{2}}\left[|000\rangle-|001\rangle+|010\rangle-|011\rangle+|100\rangle-|101\rangle+|110\rangle-|111\rangle\right]$$

$$\mathrm{CnNOT}$$

$$\frac{1}{2\sqrt{2}}\left[|000\rangle-|001\rangle+|010\rangle-|011\rangle+|101\rangle-|100\rangle+|111\rangle-|110\rangle\right]$$

$$\mathrm{I}\otimes\mathrm{CNOT}$$

$$\frac{1}{2\sqrt{2}}\left[|000\rangle-|001\rangle+|011\rangle-|010\rangle+|101\rangle-|100\rangle+|110\rangle-|111\rangle\right]$$

$$\mathrm{I}\otimes\mathrm{I}\otimes\mathrm{NOT}$$

$$\frac{1}{\sqrt{2}}\left[|0\rangle-|1\rangle\right]\frac{1}{\sqrt{2}}\left[|0\rangle-|1\rangle\right]\frac{1}{\sqrt{2}}\left[|1\rangle-|0\rangle\right]$$

$$\mathrm{H}\otimes\mathrm{H}\otimes\mathrm{I}$$

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

Since the top wires contain **|1 >**, the function is balanced. **<u>This algorithm illustrates the roles of superposition</u>, <u>entanglement and interference in quantum computation</u>**. Regarding the latter, it is destructive interference in the last step that eliminates unwanted outcomes yielding the final result on the last line.

One pass through the quantum circuit answers the question (is the function balanced or constrant) that would take 4 calculations on a classical computer. <u>Thus given this problem, a QC is faster than a classical computer.</u>

The interference that occurs in the last step is illustrated by letting |a> = |0> and |b> = |1> and carrying out Hadamard transforms on the first two qubits.

$$\frac{1}{4\cdot\sqrt{2}}\cdot\begin{bmatrix}\left(a_1+b_1\right)\cdot\left(a_2+b_2\right)\cdot b_3-\left(a_1+b_1\right)\cdot\left(a_2+b_2\right)\cdot a_3\ \ldots \\ +\left(a_1+b_1\right)\cdot\left(a_2-b_2\right)\cdot b_3-\left(a_1+b_1\right)\cdot\left(a_2-b_2\right)\cdot a_3\ \ldots \\ +\left(a_1-b_1\right)\cdot\left(a_2+b_2\right)\cdot b_3-\left(a_1-b_1\right)\cdot\left(a_2+b_2\right)\cdot a_3\ \ldots \\ +\left(a_1-b_1\right)\cdot\left(a_2-b_2\right)\cdot b_3-\left(a_1-b_1\right)\cdot\left(a_2-b_2\right)\cdot a_3\end{bmatrix}\ \mathit{simplify}\ \rightarrow\ -\frac{\sqrt{2}\cdot a_1\cdot a_2\cdot\left(a_3-b_3\right)}{2}$$