Title: Introduction to Quantum Information and Computation from a Foundational Standpoint

Date: Dec 11, 2006 10:30 AM

URL: http://pirsa.org/06120048

Abstract:

Where does the speedup come from?

- Where does the speedup in quantum computation come from?
- Some answers:
  - superposition and entanglement
  - the fact that the state space of n bits is a space of 2^n states while the state space of n qubits is a space of 2^n dimensions
  - the possibility of computing all values of a function in a single computational step by 'quantum parallelism'
  - the possibility of an efficient implementation of the discrete quantum Fourier transform.
  - all of the above

Where does the speedup come from?

- Where does the speedup in quantum computation come from?
- Some answers:
  - superposition and entanglement
  - the fact that the state space of n bits is a space of 2^n states while the state space of n qubits is a space of 2^n dimensions
  - the possibility of computing all values of a function in a single computational step by 'quantum parallelism'
  - the possibility of an efficient implementation of the discrete quantum Fourier transform.
  - all of the above

Quantum logical perspective

- The salient feature of a quantum computation relative to a classical computation in period-finding and related algorithms is the possibility of processing the information in a disjunctive statement—this or that—without evaluating the truth or falsity of the disjuncts.
- This is redundant classical information for a quantum algorithm but essential information for a classical algorithm.

Quantum logical perspective

Rather than 'computing all values of a function at once,' the point of a quantum computation is precisely to avoid the evaluation of any values of the function at all, in the sense of producing a value in the range of the function for a value in its domain.

Deutsch's problem

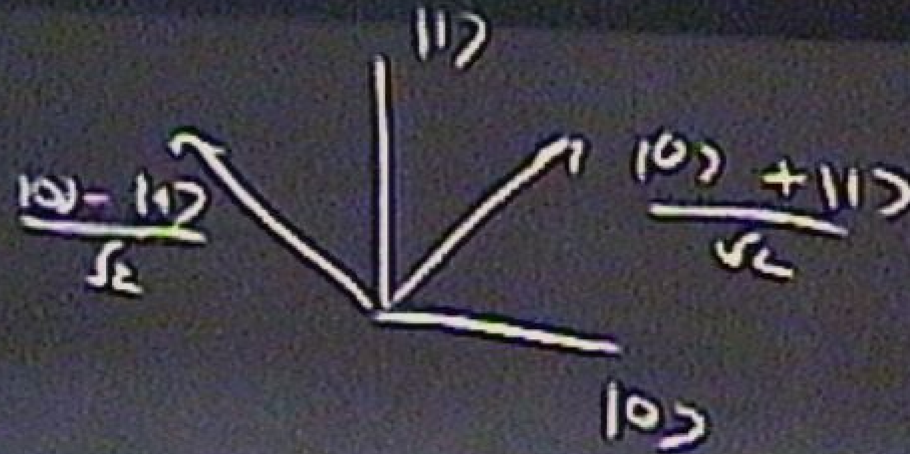
- $B = \{0, 1\}$, a Boolean algebra
- XOR problem: given a 'black box' or oracle that computes a function

  $$f : B \to B$$

  we are required to determine whether the function is 'constant' (takes the same value for both inputs) or 'balanced' (takes a different value for each input).

Deutsch's XOR Algorithm

$$
\begin{aligned}
|0\rangle|0\rangle \quad & \xrightarrow{\mathrm{H}} \quad \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle \\
& \xrightarrow{\mathrm{U_f}} \quad \frac{1}{\sqrt{2}}(|0\rangle|\mathrm{f}(0)\rangle + |1\rangle|\mathrm{f}(1)\rangle)
\end{aligned}
$$



|1⟩
(|0⟩ - |1⟩)/√2
(|0⟩ + |1⟩)/√2
|0⟩

Deutsch's XOR Algorithm

$$|0\rangle|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle$$
$$\xrightarrow{U_f} \frac{1}{\sqrt{2}}(|0\rangle|f(0)\rangle + |1\rangle|f(1)\rangle)$$

Deutsch's XOR Algorithm

f constant: the final state is :

$$|c_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|0\rangle)$$

or

$$|c_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle|1\rangle + |1\rangle|1\rangle)$$

f balanced: the final state is:

$$|b_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle)$$

or

$$|b_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle|1\rangle + |1\rangle|0\rangle)$$

Deutsch's XOR Algorithm

f constant: the final state is :

$$|c_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|0\rangle)$$

or

$$|c_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle|1\rangle + |1\rangle|1\rangle)$$

f balanced: the final state is:

$$|b_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle)$$

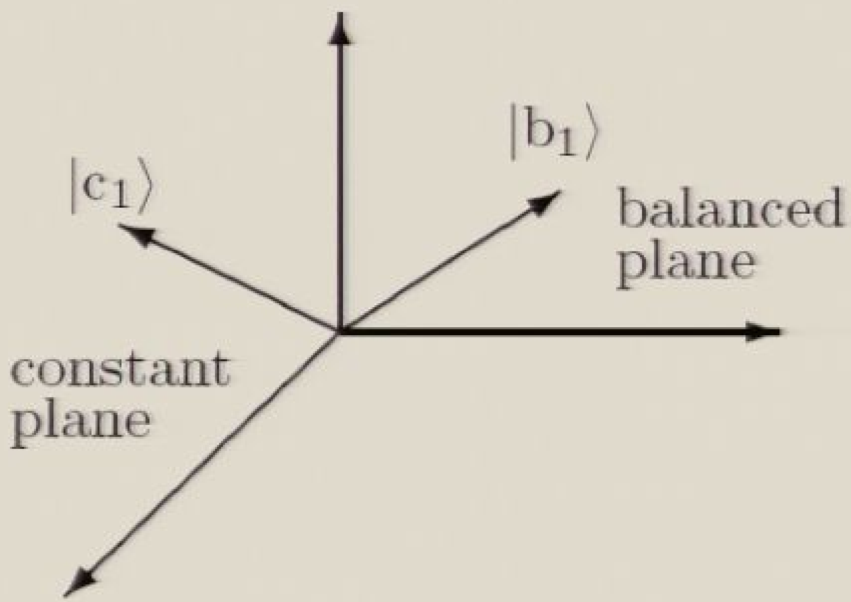
or

$$|b_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle|1\rangle + |1\rangle|0\rangle)$$

Deutsch's XOR Algorithm

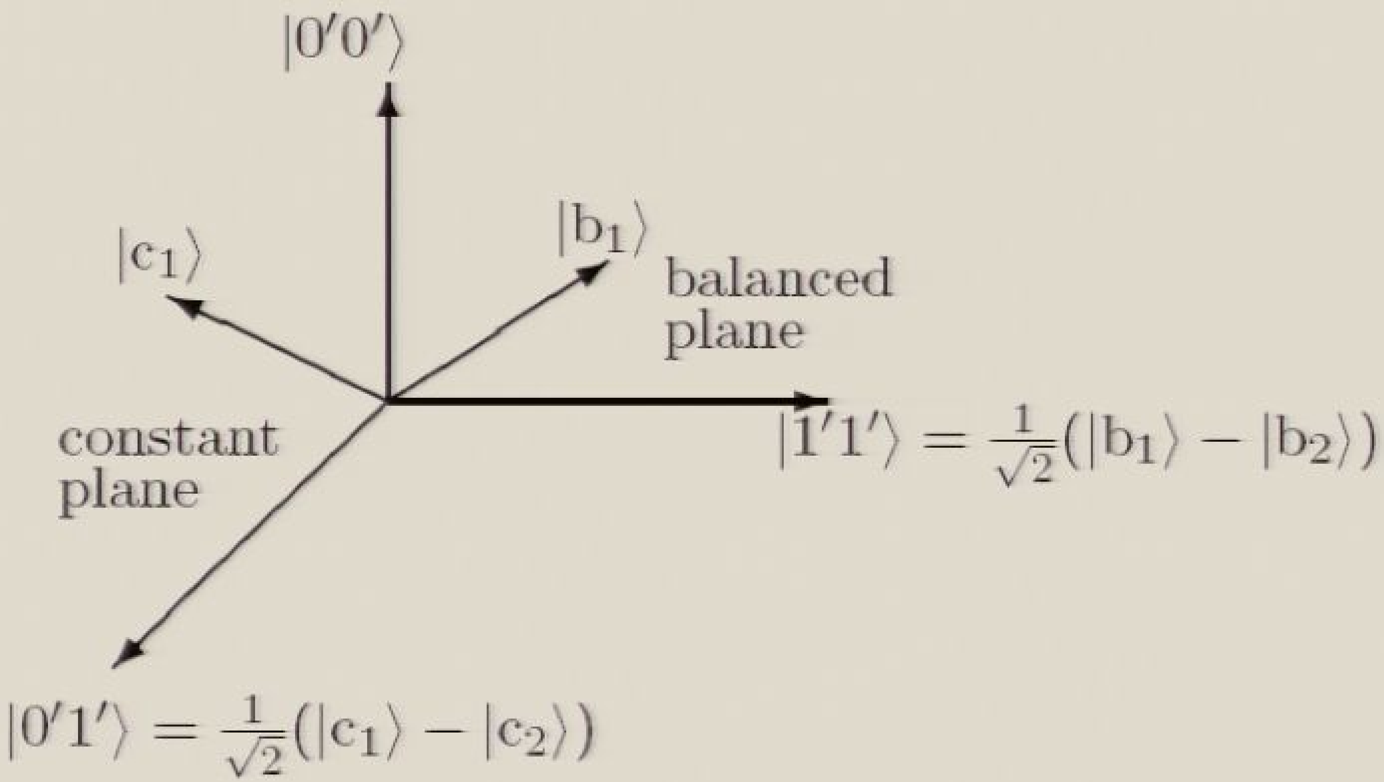
Orthogonal states $|c_1\rangle, |c_2\rangle$ and $|b_1\rangle, |b_2\rangle$ span two planes that intersect in the ray:

$$\frac{1}{\sqrt{2}}(|c_1\rangle + |c_2\rangle) = \frac{1}{\sqrt{2}}(|b_1\rangle + |b_2\rangle) = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

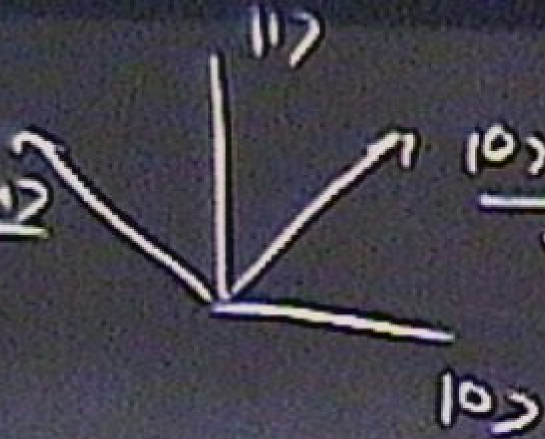


Deutsch's XOR Algorithm

In Hadamard basis $|0'\rangle = \mathrm{H}|0\rangle, |1'\rangle = \mathrm{H}|1\rangle$:



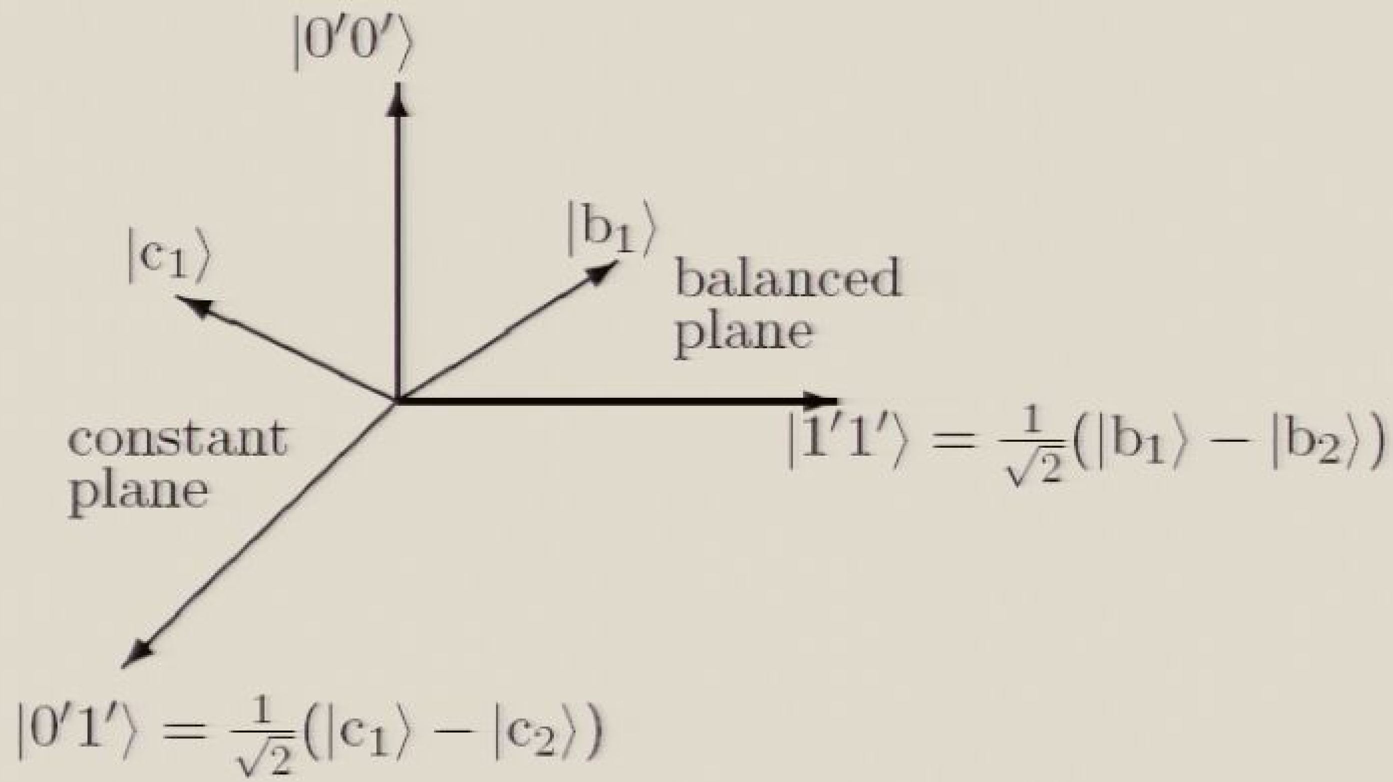
$$|1\rangle' = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$



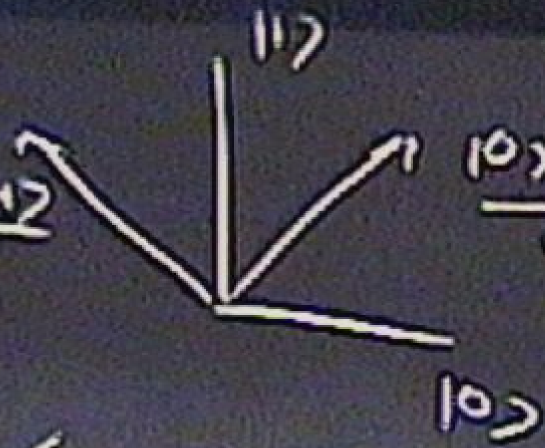
$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} = |0\rangle'$$

Deutsch's XOR Algorithm

In Hadamard basis $|0'\rangle = \mathrm{H}|0\rangle, |1'\rangle = \mathrm{H}|1\rangle$:



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

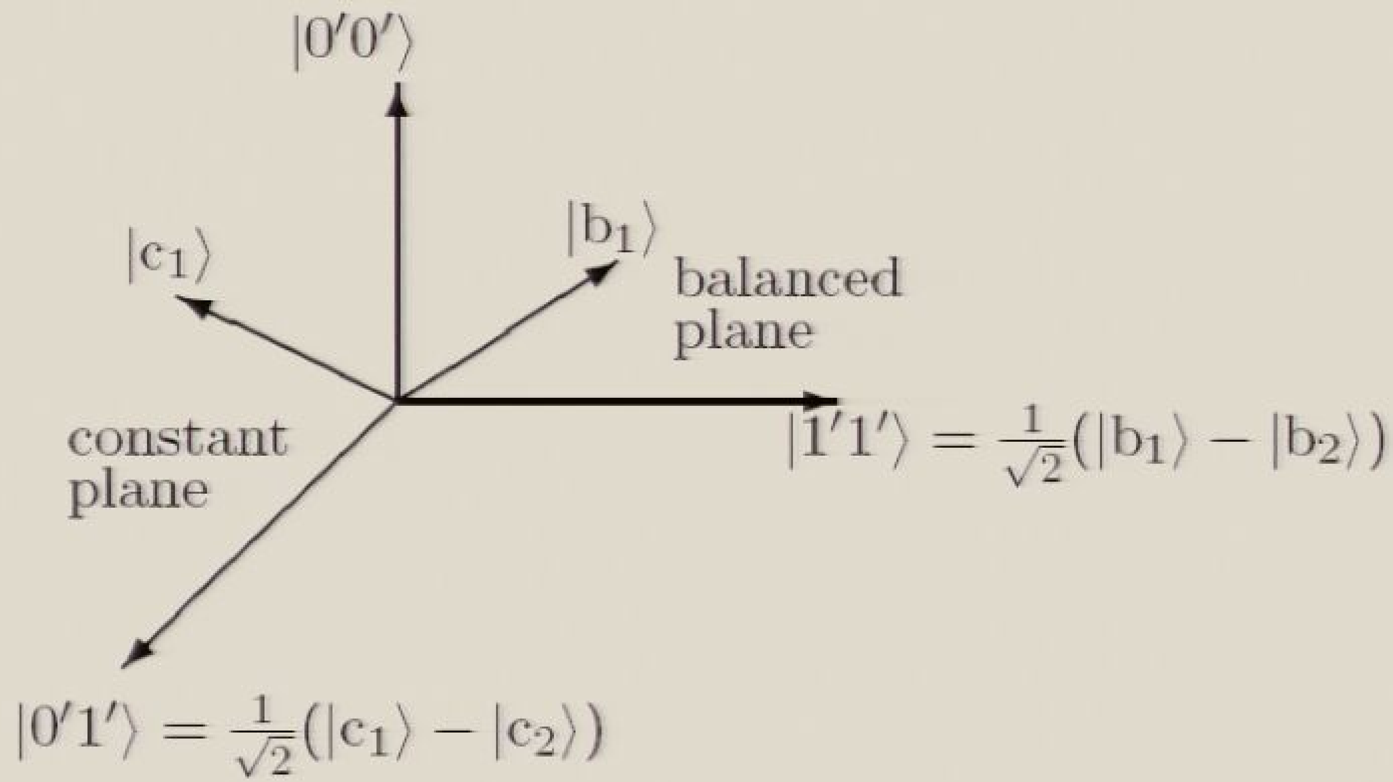


$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} = |0'\rangle$$

$$(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle)$$

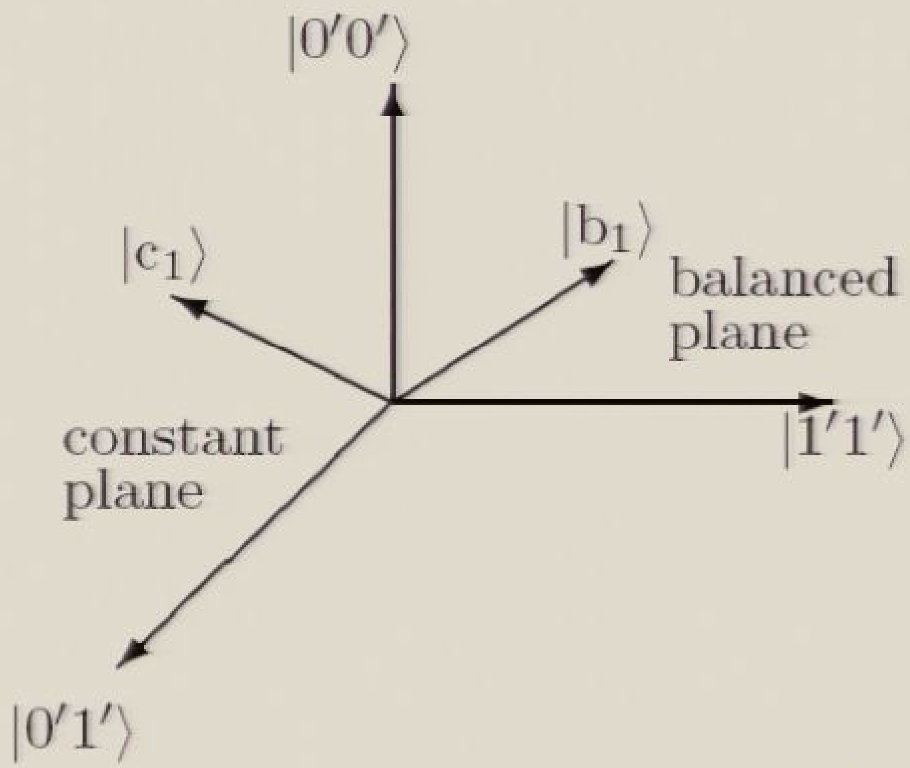
Deutsch's XOR Algorithm

In Hadamard basis $|0'\rangle = \mathrm{H}|0\rangle, |1'\rangle = \mathrm{H}|1\rangle$:



Deutsch's XOR Algorithm

To find whether f is constant or balanced, we could measure the observable with eigenstates $|0'0'\rangle$, $|0'1'\rangle$, $|1'0'\rangle$, $|1'1'\rangle$.

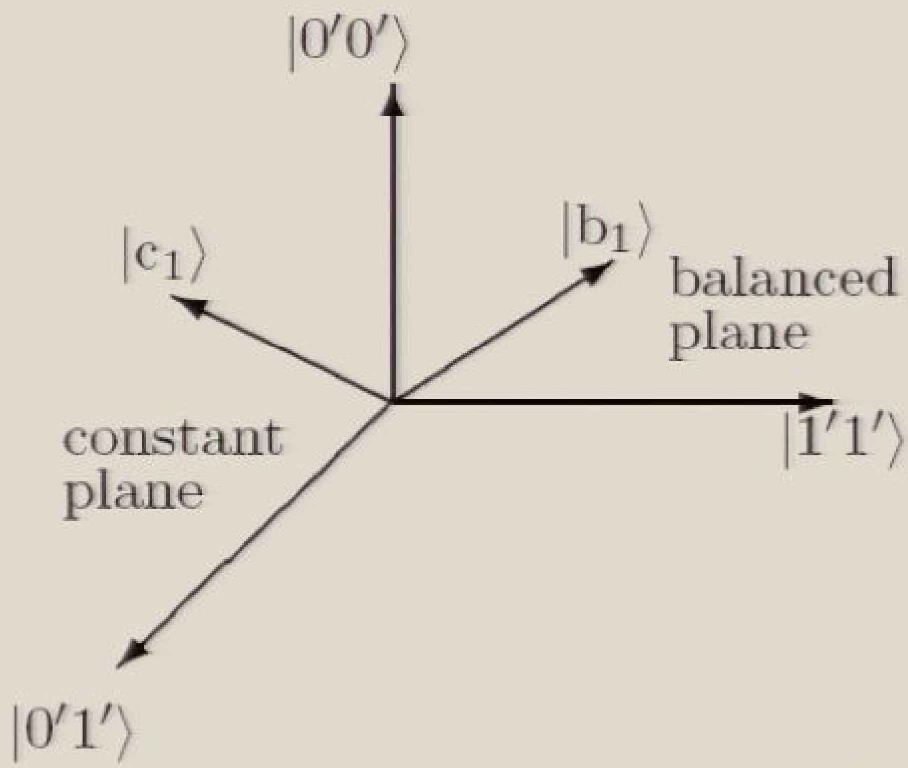


Deutsch's XOR Algorithm

- A Hadamard transformation to the final state amounts to dropping the primes in the representation for the constant and balanced planes (since $\mathrm{H}^2 = \mathrm{I}$, so $|0'0'\rangle \xrightarrow{\mathrm{H}} |00\rangle$, etc.).
- The relationship between the states $|c_1\rangle, |c_2\rangle, |b_1\rangle, |b_2\rangle$ and the constant and balanced planes $\mathrm{P}_{|0'\rangle|0'\rangle} + \mathrm{P}_{|0'\rangle|1'\rangle}$ and $\mathrm{P}_{|0'\rangle|0'\rangle} + \mathrm{P}_{|1'\rangle|1'\rangle}$ is the same, after the Hadamard transformation of the state, as the relationship between the states $\mathrm{H}|c_1\rangle, \mathrm{H}|c_2\rangle, \mathrm{H}|b_1\rangle, \mathrm{H}|b_2\rangle$ and the planes defined by $\mathrm{P}_{|0\rangle|0\rangle} + \mathrm{P}_{|0\rangle|1\rangle}$ and $\mathrm{P}_{|0\rangle|0\rangle} + \mathrm{P}_{|1\rangle|1\rangle}$.

Deutsch's XOR Algorithm

To find whether f is constant or balanced, we could measure the observable with eigenstates $|0'0'\rangle$, $|0'1'\rangle$, $|1'0'\rangle$, $|1'1'\rangle$.

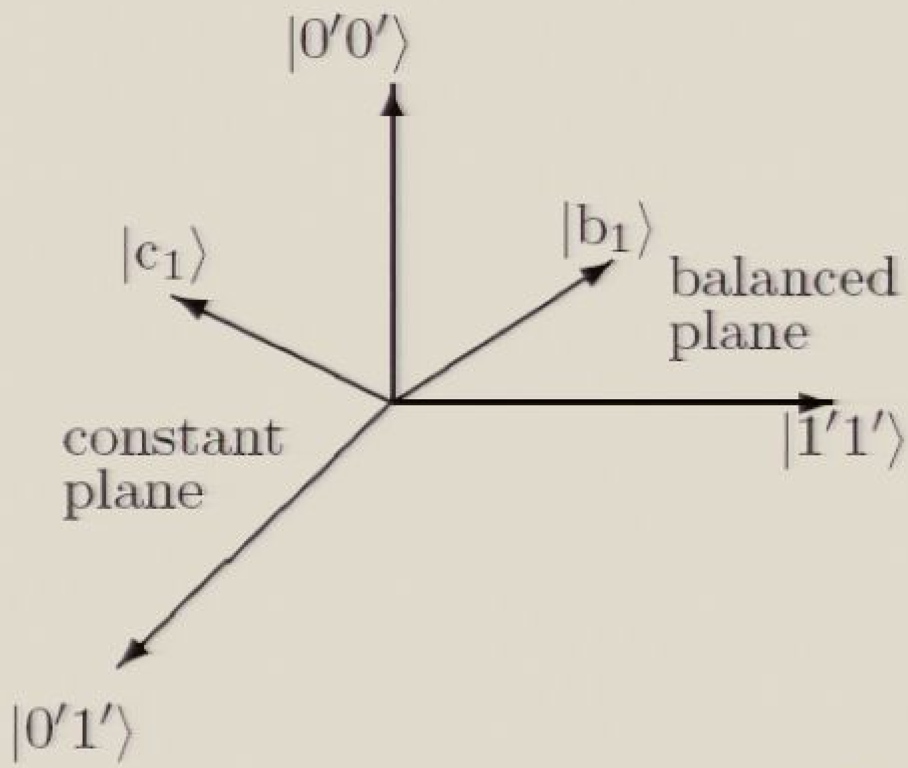


Deutsch's XOR Algorithm

- A Hadamard transformation to the final state amounts to dropping the primes in the representation for the constant and balanced planes (since $H^2 = I$, so $|0'0'\rangle \xrightarrow{H} |00\rangle$, etc.).
- The relationship between the states $|c_1\rangle, |c_2\rangle, |b_1\rangle, |b_2\rangle$ and the constant and balanced planes $P_{|0'\rangle|0'\rangle} + P_{|0'\rangle|1'\rangle}$ and $P_{|0'\rangle|0'\rangle} + P_{|1'\rangle|1'\rangle}$ is the same, after the Hadamard transformation of the state, as the relationship between the states $H|c_1\rangle, H|c_2\rangle, H|b_1\rangle, H|b_2\rangle$ and the planes defined by $P_{|0\rangle|0\rangle} + P_{|0\rangle|1\rangle}$ and $P_{|0\rangle|0\rangle} + P_{|1\rangle|1\rangle}$.

Deutsch's XOR Algorithm

To find whether f is constant or balanced, we could measure the observable with eigenstates $|0'0'\rangle$, $|0'1'\rangle$, $|1'0'\rangle$, $|1'1'\rangle$.



Deutsch's XOR Algorithm

- A Hadamard transformation to the final state amounts to dropping the primes in the representation for the constant and balanced planes (since $\mathrm{H}^2 = \mathrm{I}$, so $|0'0'\rangle \xrightarrow{\mathrm{H}} |00\rangle$, etc.).
- The relationship between the states $|c_1\rangle, |c_2\rangle, |b_1\rangle, |b_2\rangle$ and the constant and balanced planes $P_{|0'\rangle|0'\rangle} + P_{|0'\rangle|1'\rangle}$ and $P_{|0'\rangle|0'\rangle} + P_{|1'\rangle|1'\rangle}$ is the same, after the Hadamard transformation of the state, as the relationship between the states $\mathrm{H}|c_1\rangle, \mathrm{H}|c_2\rangle, \mathrm{H}|b_1\rangle, \mathrm{H}|b_2\rangle$ and the planes defined by $P_{|0\rangle|0\rangle} + P_{|0\rangle|1\rangle}$ and $P_{|0\rangle|0\rangle} + P_{|1\rangle|1\rangle}$.

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

$$\xrightarrow{H} 0$$

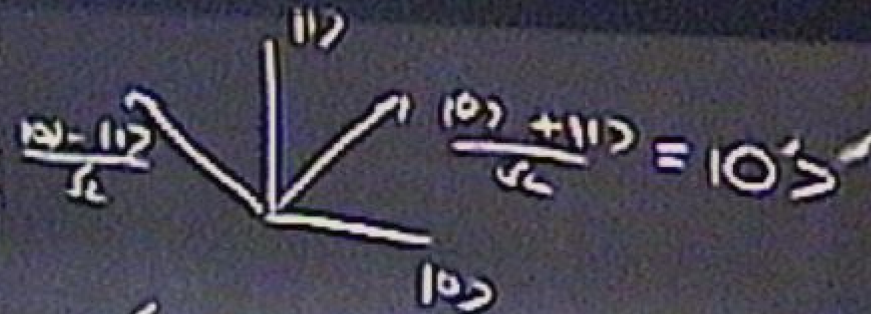
$$|1'\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \qquad \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |0'\rangle$$

$|1\rangle$, $|0\rangle$

$$\left(|0\rangle + |1\rangle\right) \otimes \left(|0\rangle + |1\rangle\right)$$

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

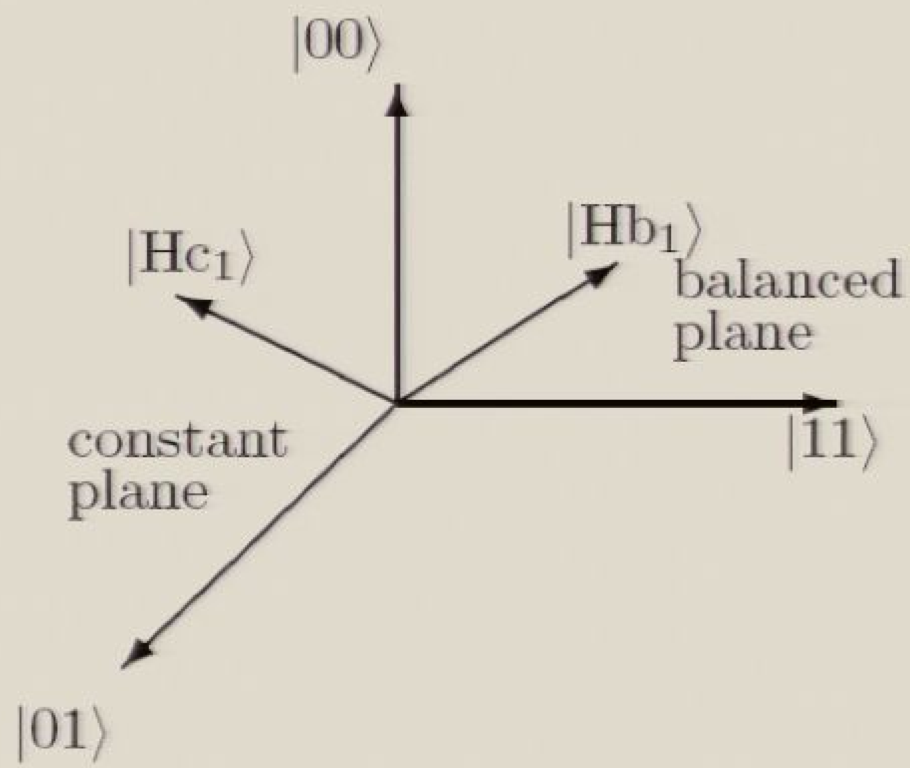
$$\xrightarrow{H} \frac{\frac{|0\rangle+|1\rangle}{\sqrt{2}} + \frac{|0\rangle-|1\rangle}{\sqrt{2}}}{\sqrt{2}}$$



$$\rightarrow \frac{\frac{2}{\sqrt{2}}|0\rangle}{\sqrt{2}} = \frac{\sqrt{2}}{\sqrt{2}}|0\rangle = |0\rangle$$

$$(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle)$$

Deutsch's XOR Algorithm



Cleve variation

- The algorithm has an even probability of failing.
- A variation by Cleve avoids this feature.

Cleve variation

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

$$|\mathrm{x}\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{\mathrm{U_f}} (-1)^{\mathrm{f(x)}}|\mathrm{x}\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

So:

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{\mathrm{U_f}} \frac{(-1)^{\mathrm{f(0)}}|0\rangle + (-1)^{\mathrm{f(1)}}|1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Cleve variation

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

$$|\mathrm{x}\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}} \xrightarrow{\mathrm{U_f}} (-1)^{\mathrm{f(x)}}|\mathrm{x}\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

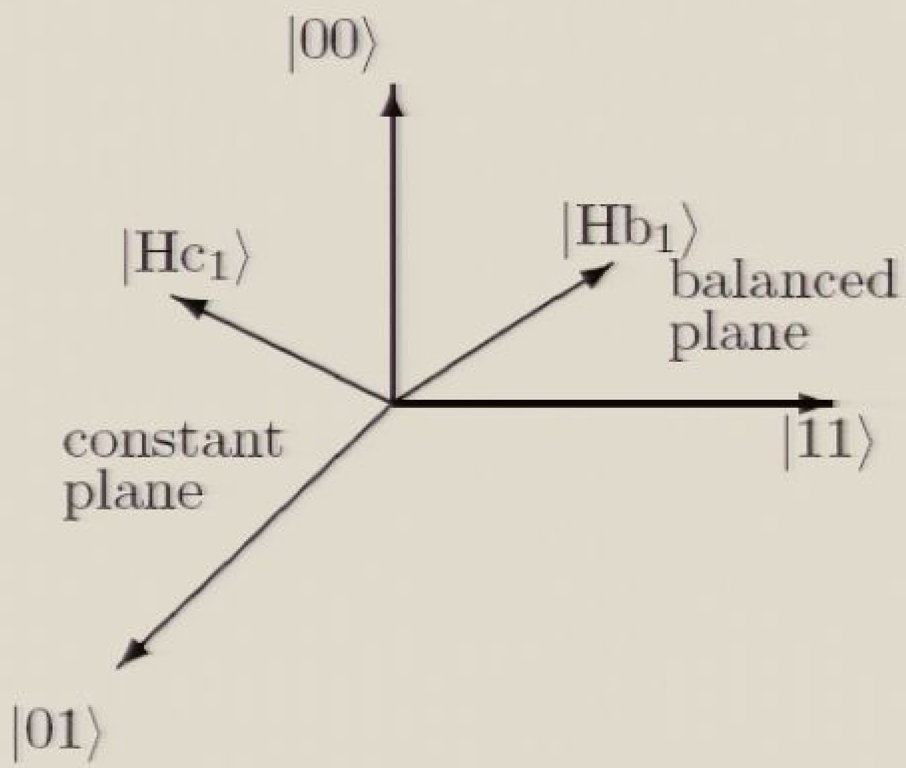
So:

$$\frac{|0\rangle+|1\rangle}{\sqrt{2}}\frac{|0\rangle-|1\rangle}{\sqrt{2}} \xrightarrow{\mathrm{U_f}} \frac{(-1)^{\mathrm{f(0)}}|0\rangle+(-1)^{\mathrm{f(1)}}|1\rangle}{\sqrt{2}}\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

Cleve variation

- The algorithm has an even probability of failing.
- A variation by Cleve avoids this feature.

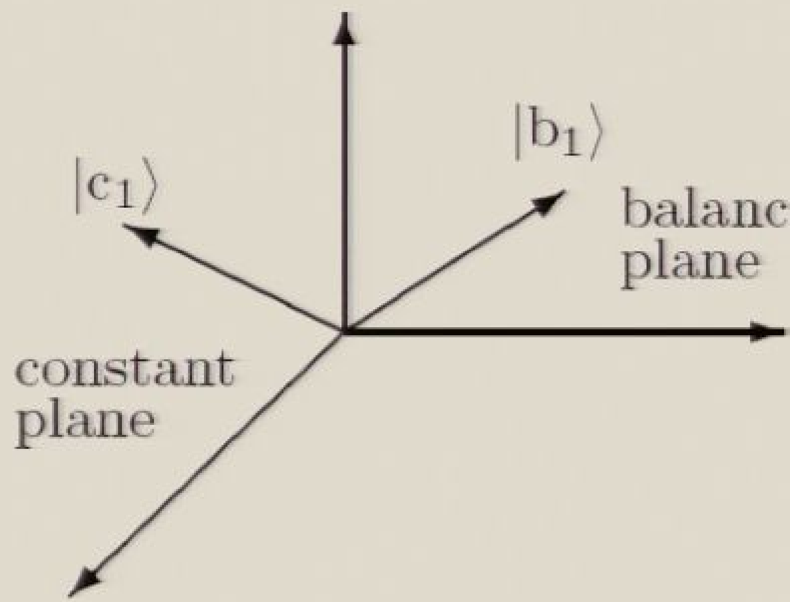
Deutsch's XOR Algorithm



Deutsch's XOR Algorithm

Orthogonal states $|c_1\rangle, |c_2\rangle$ and $|b_1\rangle, |b_2\rangle$ span two planes that intersect in the ray:

$$\frac{1}{\sqrt{2}}(|c_1\rangle + |c_2\rangle) = \frac{1}{\sqrt{2}}(|b_1\rangle + |b_2\rangle) = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$



Deutsch's XOR Algorithm

f constant: the final state is :

$$|c_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|0\rangle)$$

or

$$|c_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle|1\rangle + |1\rangle|1\rangle)$$

f balanced: the final state is:

Deutsch's XOR Algorithm

$$
\begin{aligned}
|0\rangle|0\rangle \quad &\xrightarrow{\mathrm{H}} \quad \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle \\
&\xrightarrow{\mathrm{U_f}} \quad \frac{1}{\sqrt{2}}(|0\rangle|f(0)\rangle + |1\rangle|f(1)\rangle)
\end{aligned}
$$

Deutsch's XOR Algorithm

f constant: the final state is :

$$|c_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|0\rangle)$$

or

$$|c_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle|1\rangle + |1\rangle|1\rangle)$$

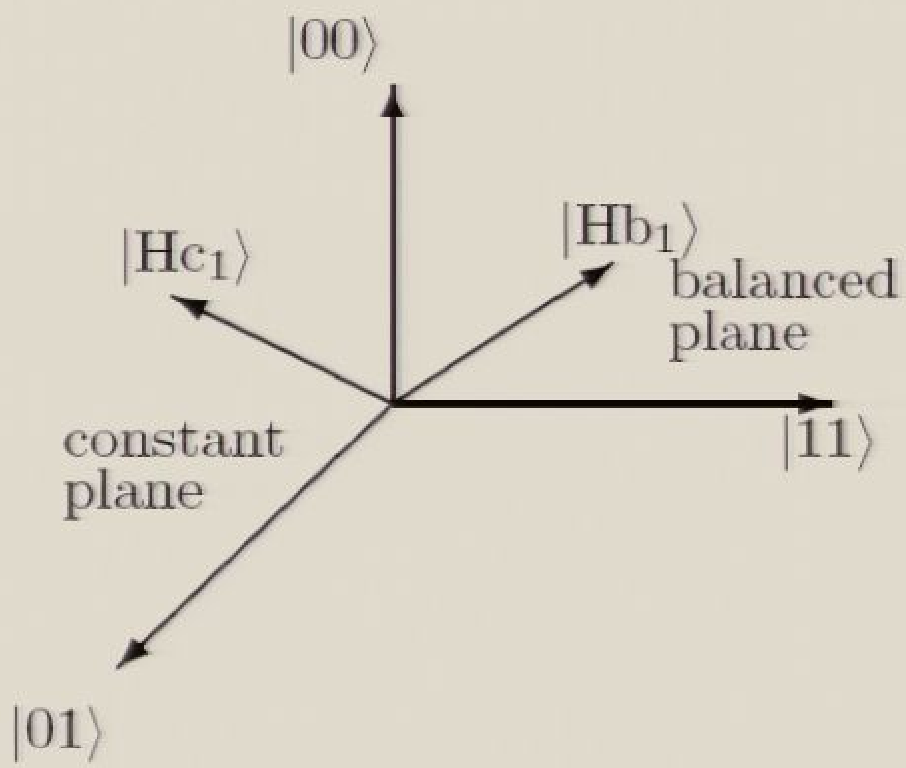
f balanced: the final state is:

$$|b_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle)$$

or

$$|b_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle|1\rangle + |1\rangle|0\rangle)$$

Deutsch's XOR Algorithm



Cleve variation

- The algorithm has an even probability of failing.
- A variation by Cleve avoids this feature.

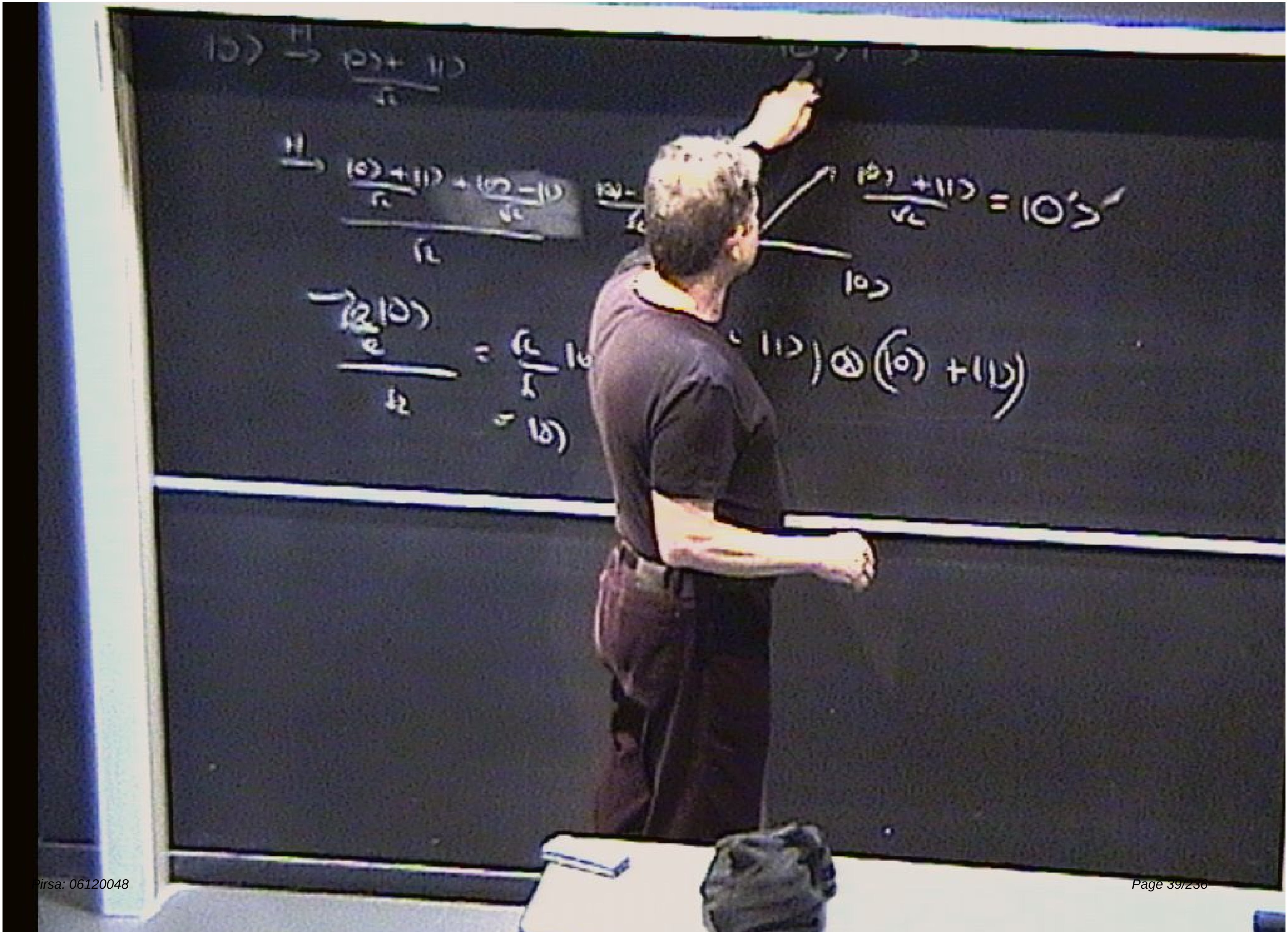
Cleve variation

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

$$|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} (-1)^{f(x)}|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

So:

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$



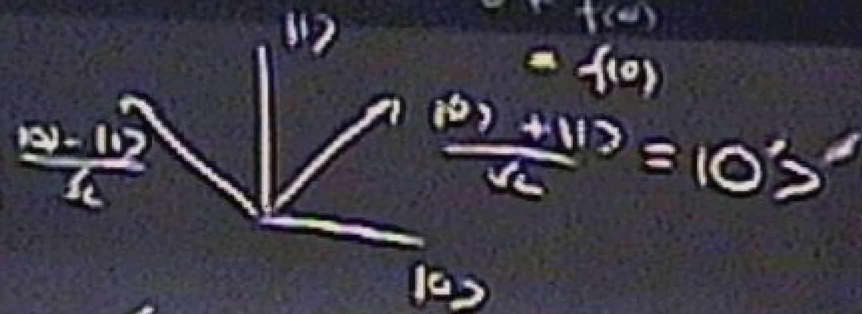
$|0\rangle \xrightarrow{H} \frac{|0\rangle + |1\rangle}{\sqrt{2}}$

$|0\rangle \oplus f(0) = \ldots$

$0 + f(0)$

$= f(0)$

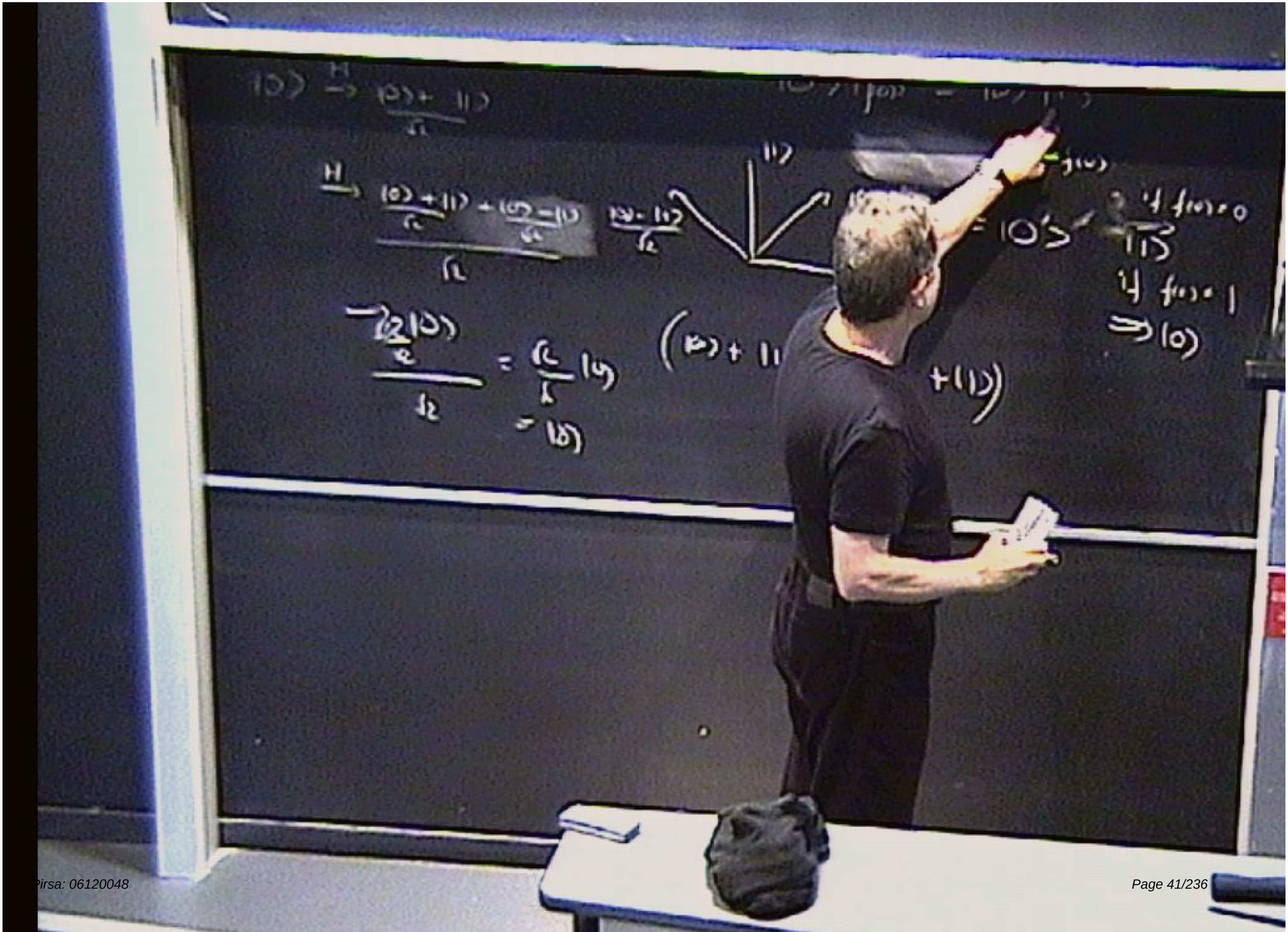
$\xrightarrow{H} \frac{\frac{|0\rangle + |1\rangle}{\sqrt{2}} + \frac{|0\rangle - |1\rangle}{\sqrt{2}}}{\sqrt{2}}$

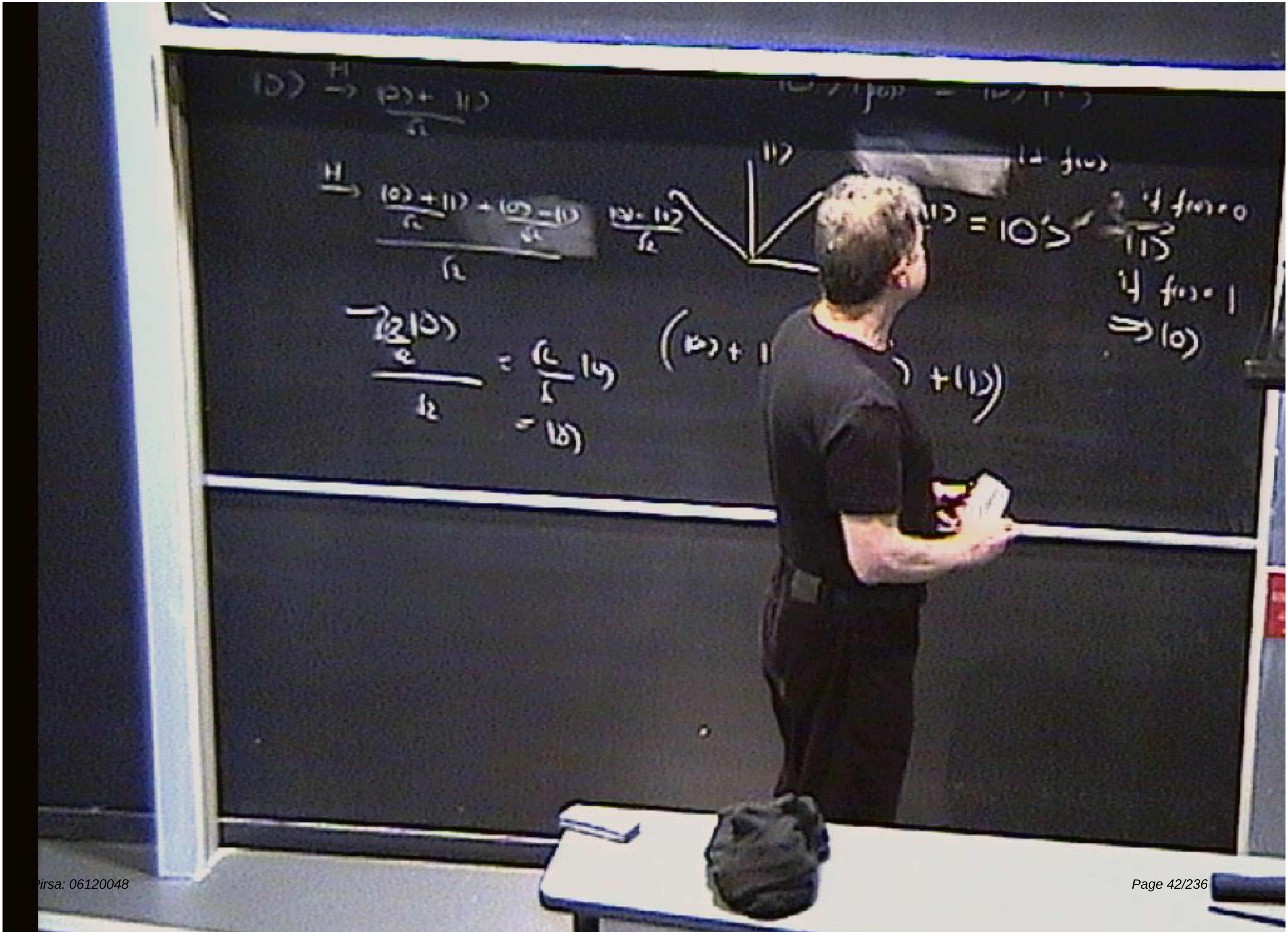


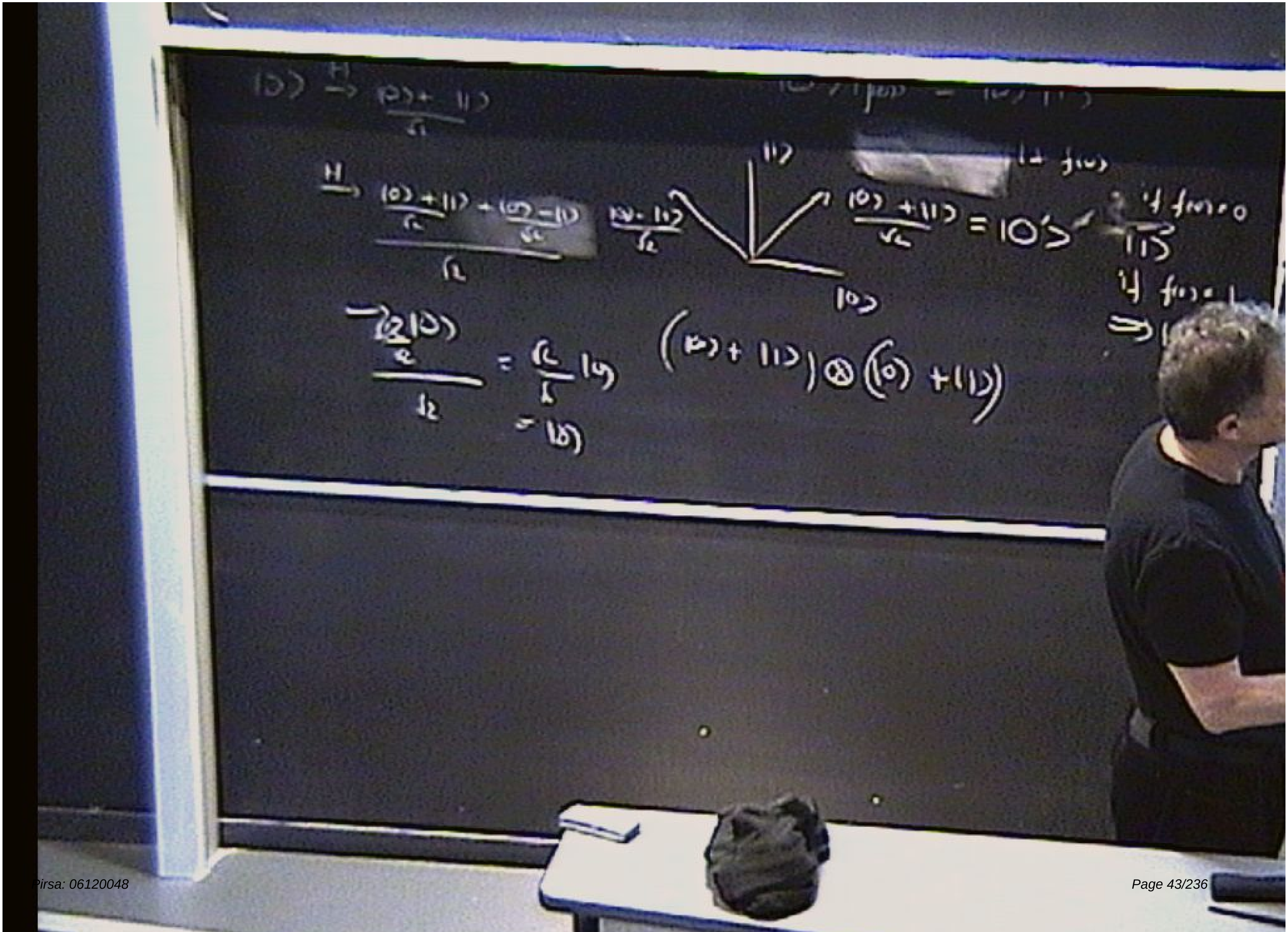
$\frac{\frac{2|0\rangle}{\sqrt{2}}}{\sqrt{2}} = \frac{\sqrt{2}}{\sqrt{2}}|0\rangle$

$= |0\rangle$

$(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle)$


|0⟩ →H (|0⟩ + |1⟩)/√2
|1⟩
(|0⟩ − |1⟩)/√2
if f(0) = 0
if f(0) = 1
⇒ |0⟩
= |0⟩




|1⟩
|0⟩
= |0'⟩
if f(0) = 0
|1'⟩

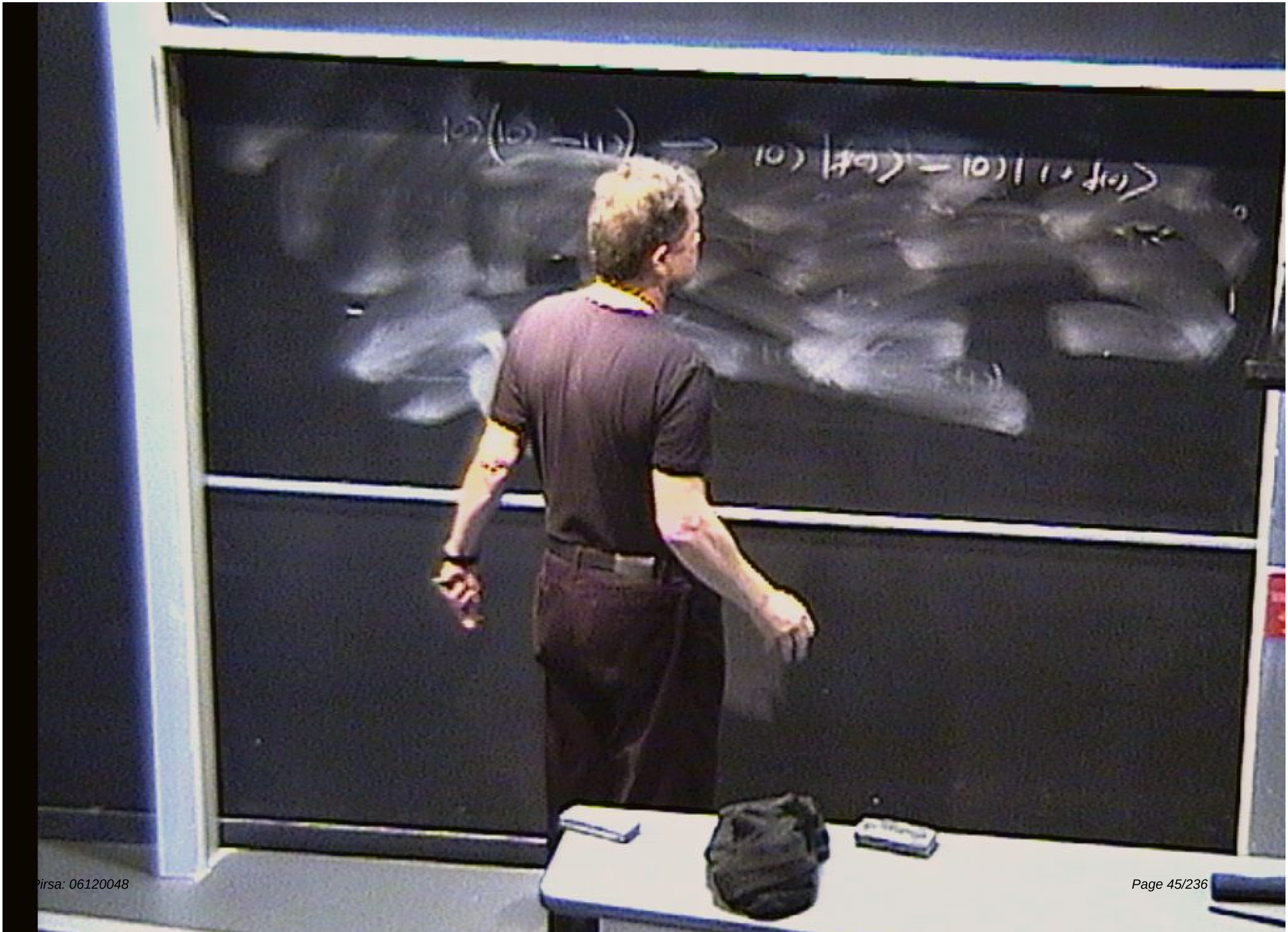
Cleve variation

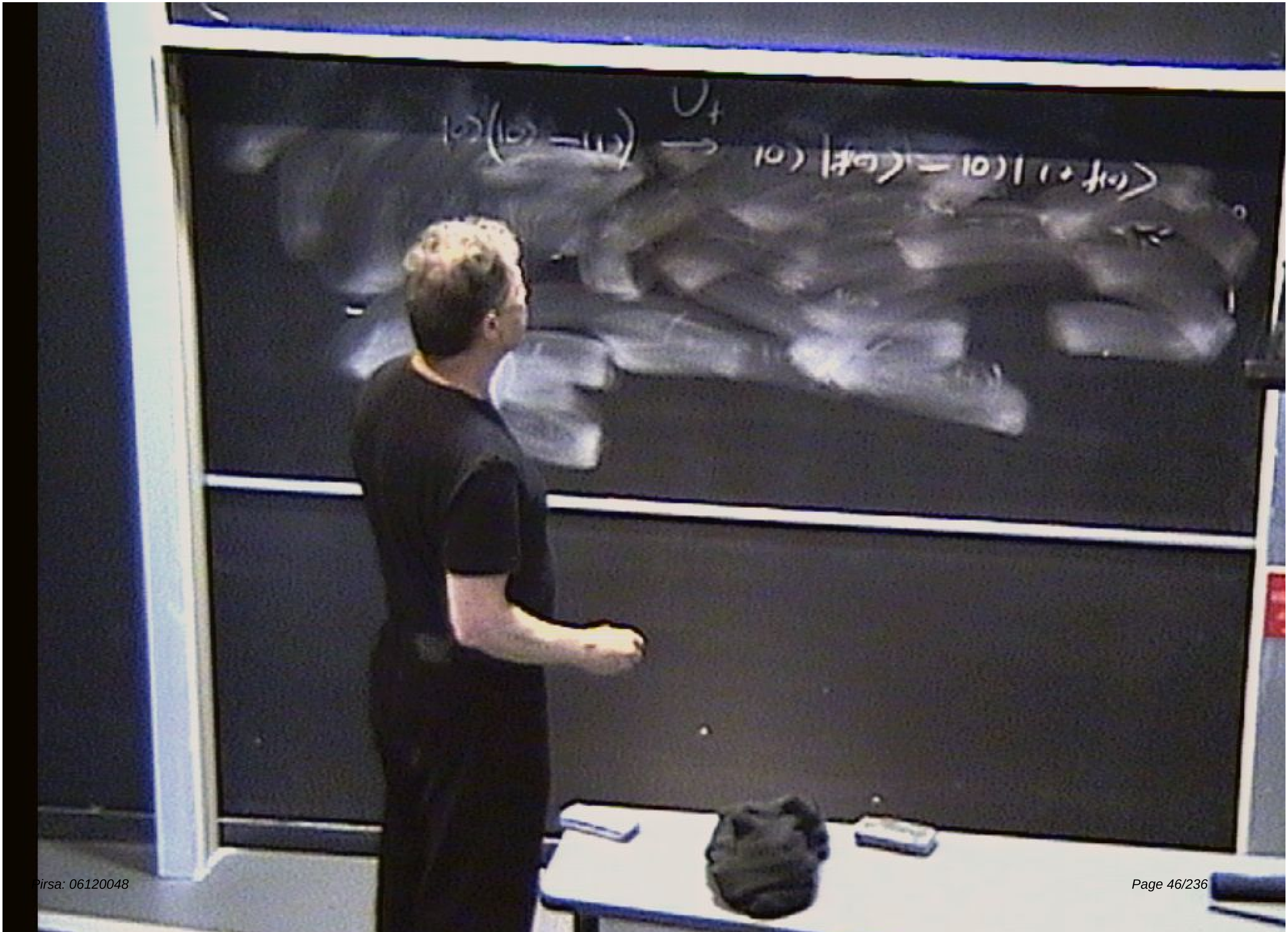
$$|0\rangle|1\rangle \xrightarrow{H} \frac{|0\rangle + |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

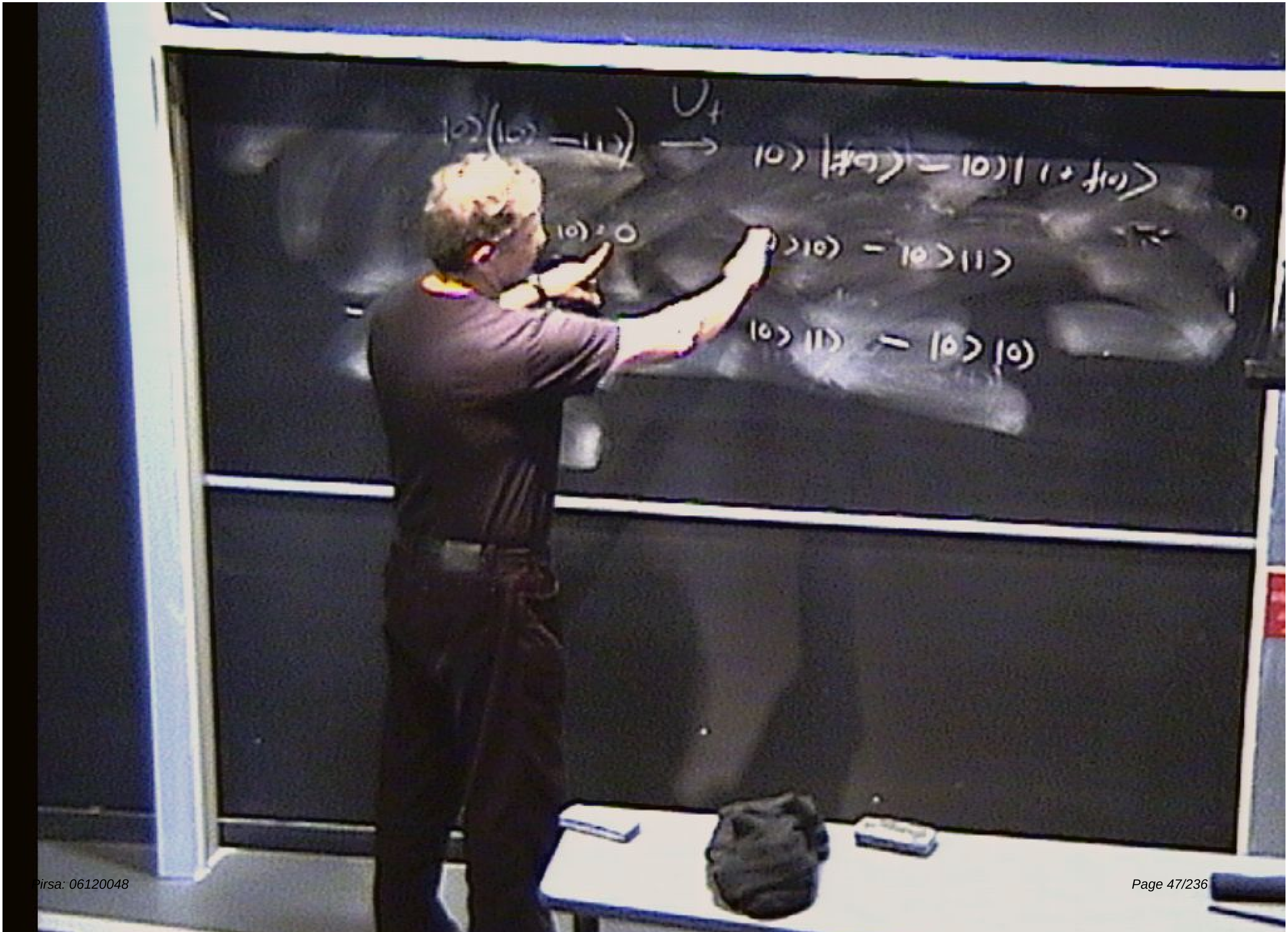
$$|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} (-1)^{f(x)}|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

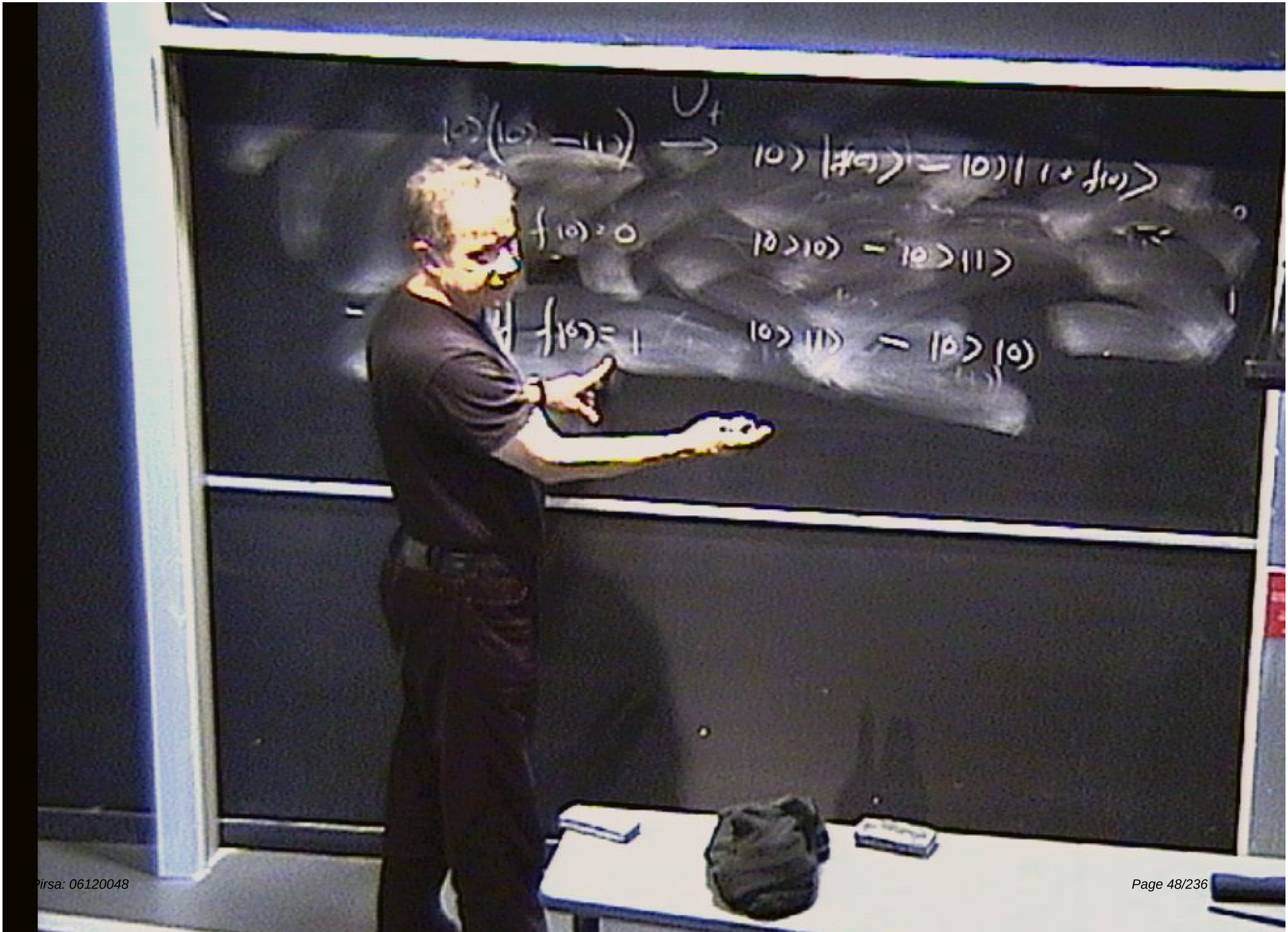
So:

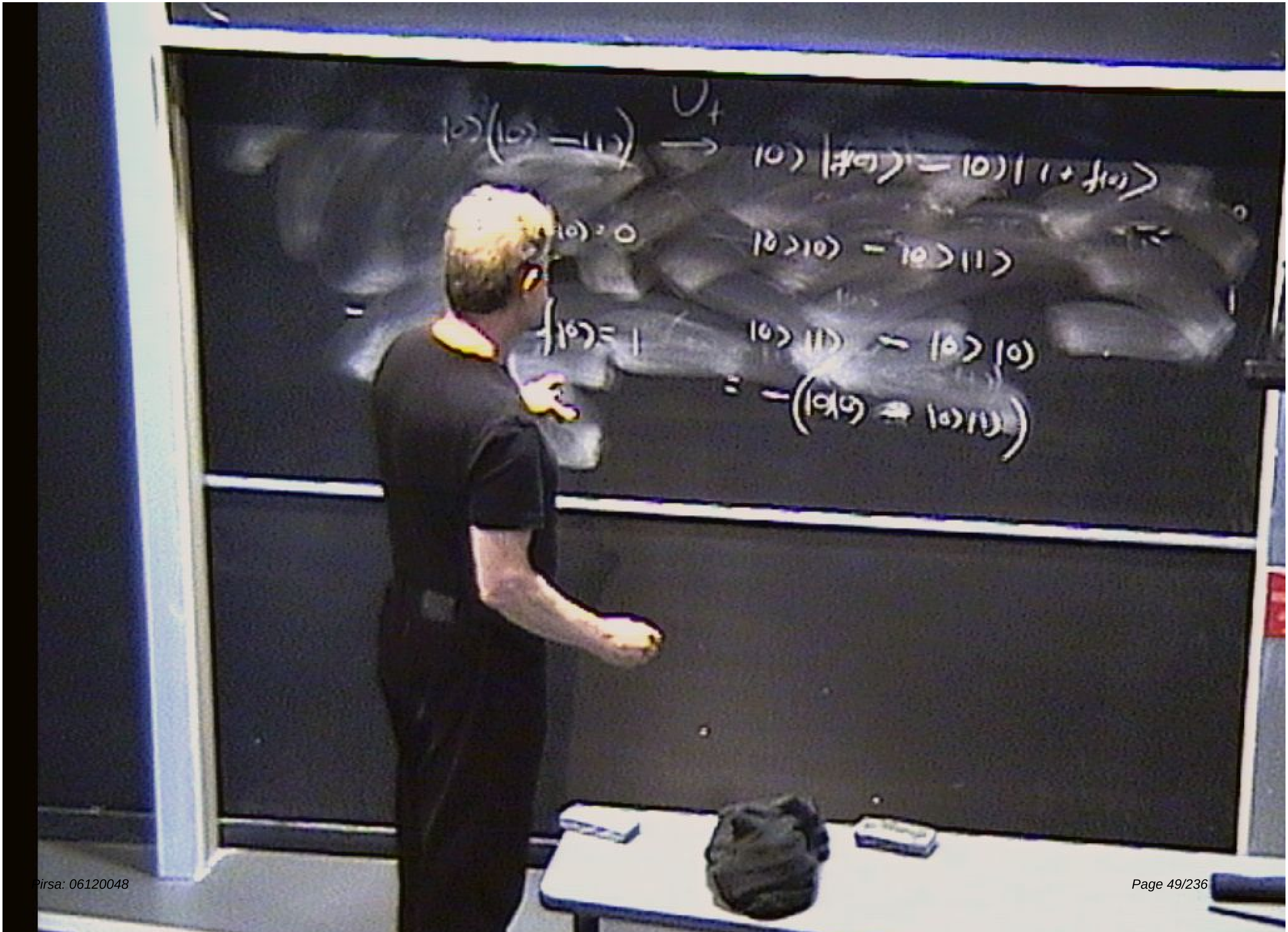
$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

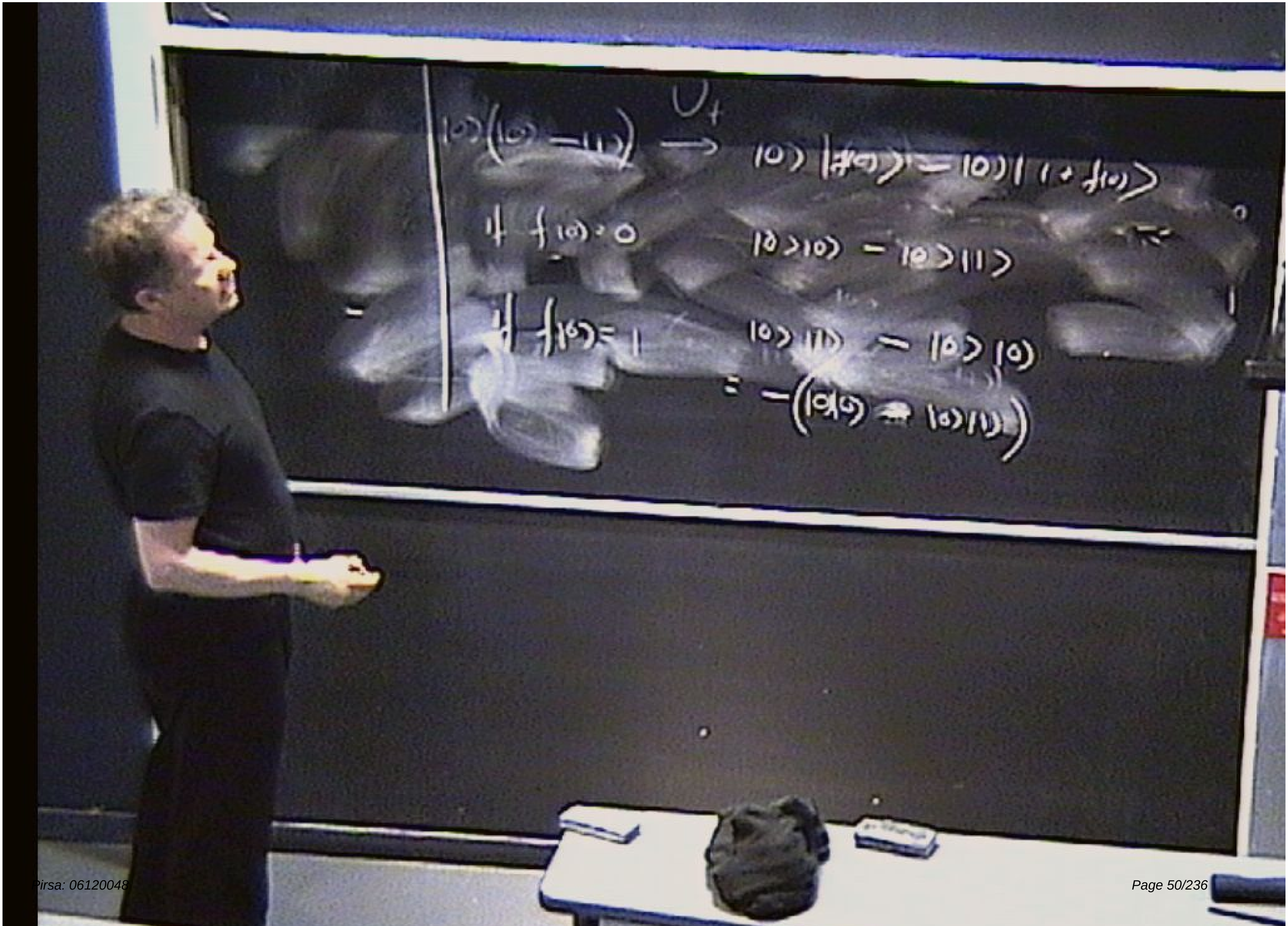


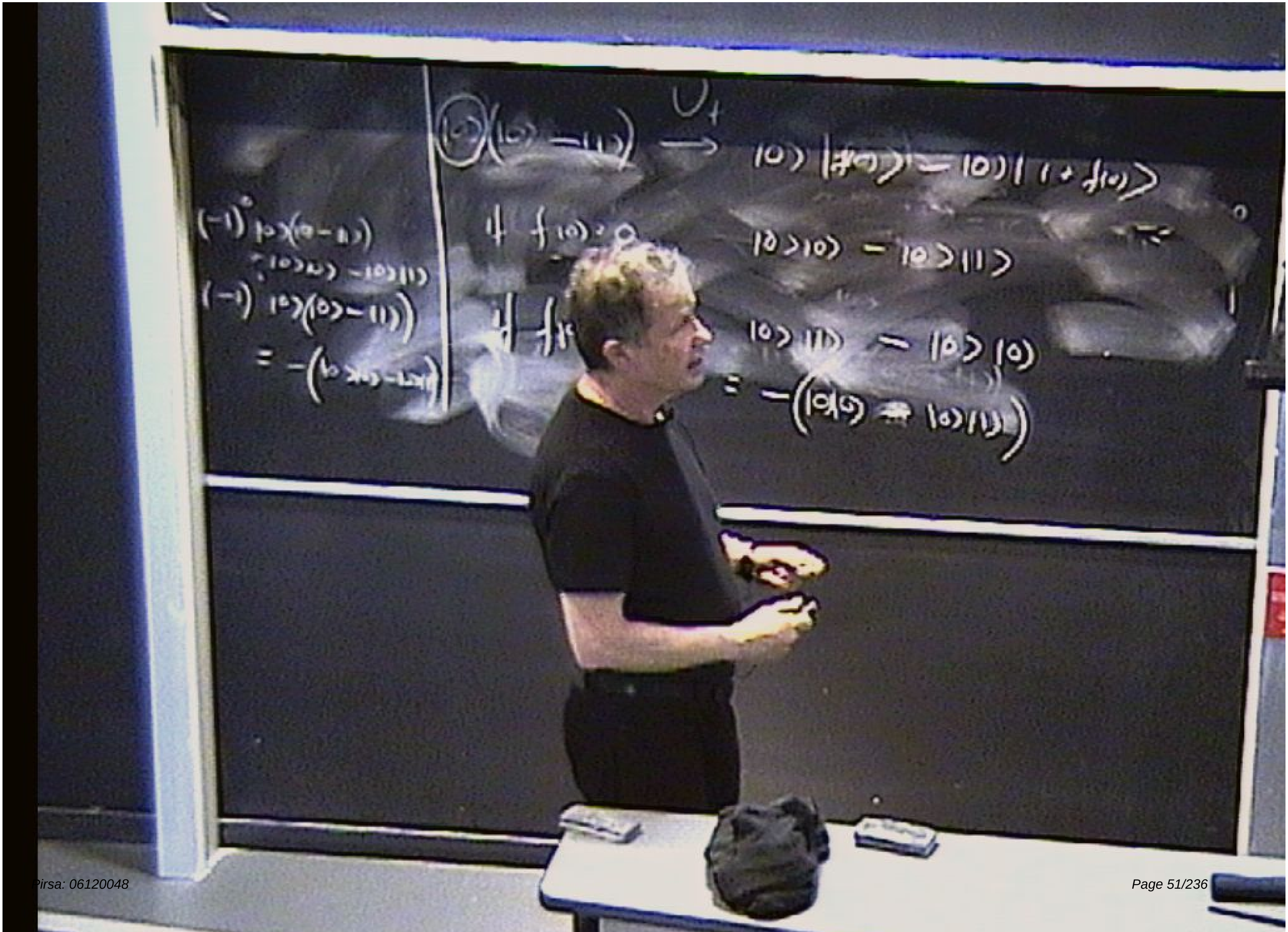

$|0\rangle(|0\rangle - |1\rangle) \xrightarrow{U_f} |0\rangle|f(0)\rangle - |0\rangle|1 \oplus f(0)\rangle$

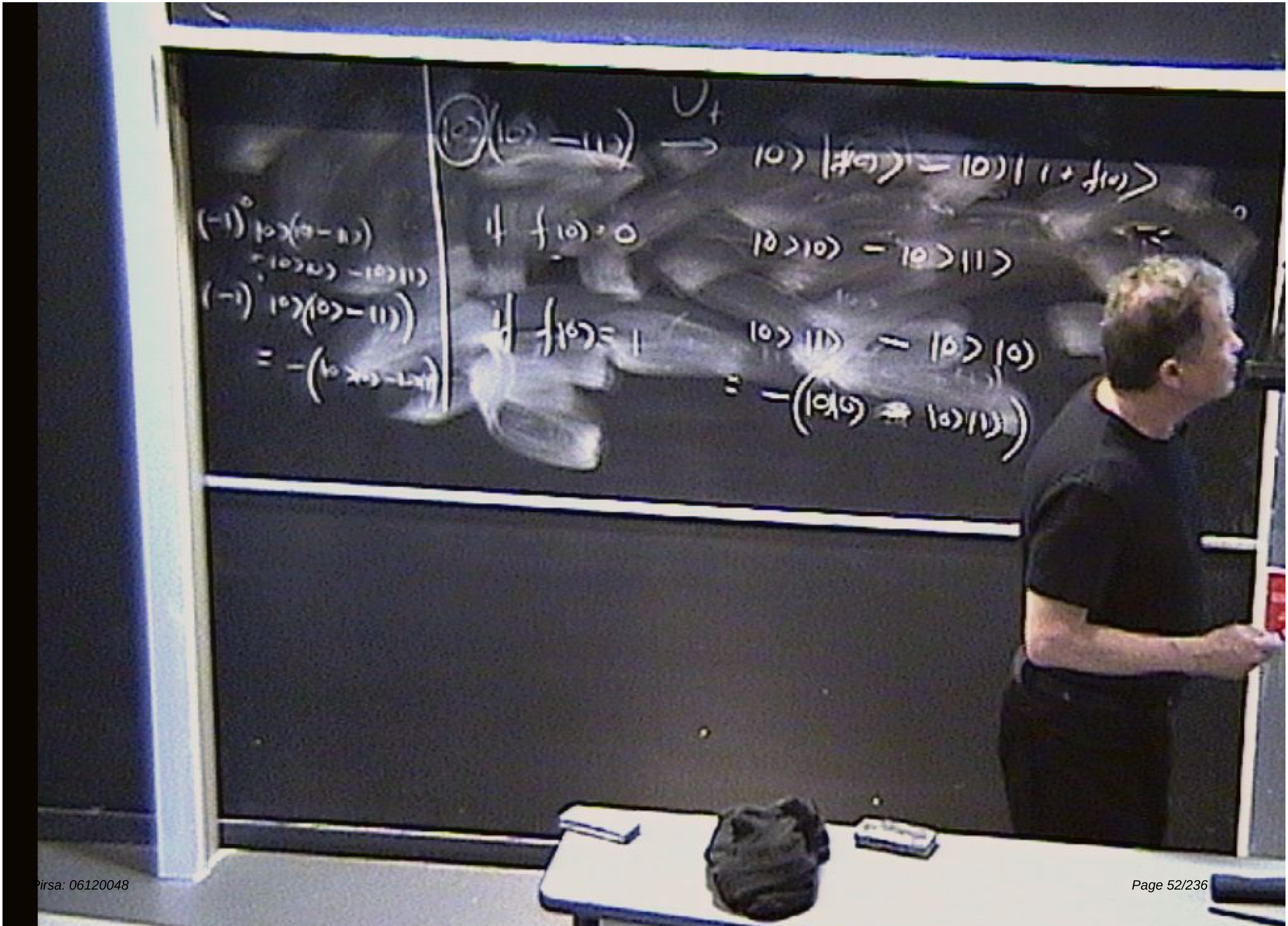

|0>|0> − |0>|1>
|0> |1> − |0> |0>


U_f
|0⟩|f(0)⟩ − |0⟩|1+f(0)⟩
f(0)=0
|0⟩|0⟩ − |0⟩|1⟩
f(0)=1
|0⟩|1⟩ − |0⟩|0⟩


|0⟩(|0⟩ − |1⟩) —U_f→ |0⟩|f(0)⟩ − |0⟩|1+f(0)⟩
f(0) = 0 |0⟩|0⟩ − |0⟩|1⟩
f(0) = 1 |0⟩|1⟩ − |0⟩|0⟩
= −(|0⟩|0⟩ − |0⟩|1⟩)


|0⟩(|0⟩ − |1⟩) →U_f |0⟩ |f(0)⟩ − |0⟩|1+f(0)⟩
if f(0)=0 |0⟩|0⟩ − |0⟩|1⟩
if f(0)=1 |0⟩|1⟩ − |0⟩|0⟩
= −(|0⟩|0⟩ − |0⟩|1⟩)




U_f
$|0\rangle(|0\rangle - |1\rangle) \longrightarrow |0\rangle|f(0)\rangle - |0\rangle|1+f(0)\rangle$
$(-1)^0 |0\rangle(|0\rangle - |1\rangle)$
$= |0\rangle|0\rangle - |0\rangle|1\rangle$
$(-1)^1 |0\rangle(|0\rangle - |1\rangle)$
$= -(|0\rangle|0\rangle - |0\rangle|1\rangle)$
if $f(0) = 0$
$|0\rangle|0\rangle - |0\rangle|1\rangle$
if $f(0) = 1$
$|0\rangle|1\rangle - |0\rangle|0\rangle$
$= -(|0\rangle|0\rangle - |0\rangle|1\rangle)$

Cleve variation

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

$$|x\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}} \xrightarrow{\mathrm{U_f}} (-1)^{f(x)}|x\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

So:

$$\frac{|0\rangle+|1\rangle}{\sqrt{2}}\frac{|0\rangle-|1\rangle}{\sqrt{2}} \xrightarrow{\mathrm{U_f}} \frac{(-1)^{f(0)}|0\rangle+(-1)^{f(1)}|1\rangle}{\sqrt{2}}\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

Cleve variation

So we end up with:

f constant (f(0) = f(1)): the final state is:

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

f balanced (f(0) $\neq$ f(1)): the final state is:

$$\pm\frac{|0\rangle - |1\rangle}{\sqrt{2}}\frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{H} \pm|1\rangle|1\rangle$$

Cleve variation

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

$$|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} (-1)^{f(x)}|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

So:

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Cleve variation

So we end up with:

f constant $(f(0) = f(1))$: the final state is:

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

f balanced $(f(0) \neq f(1))$: the final state is:

$$\pm\frac{|0\rangle - |1\rangle}{\sqrt{2}}\frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{H} \pm|1\rangle|1\rangle$$

Cleve variation

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

$$|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} (-1)^{f(x)}|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

So:

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Cleve variation

So we end up with:

f constant ($f(0) = f(1)$): the final state is:

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

f balanced ($f(0) \neq f(1)$): the final state is:

$$\pm\frac{|0\rangle - |1\rangle}{\sqrt{2}}\frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{H} \pm|1\rangle|1\rangle$$

Cleve variation

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

$$|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} (-1)^{f(x)}|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

So:

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Cleve variation

So we end up with:

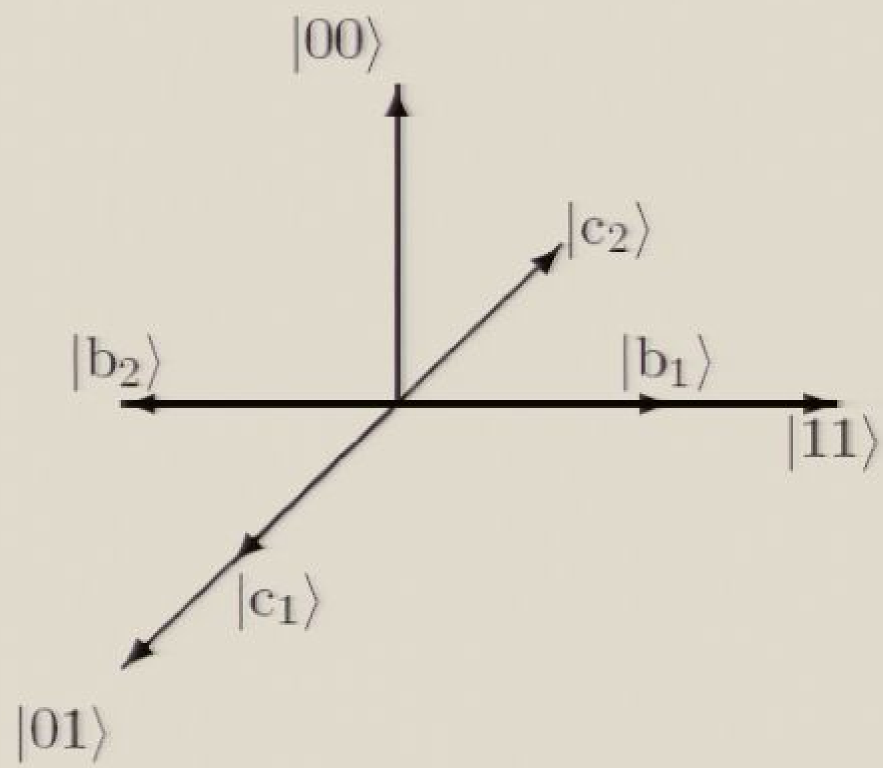
f constant $(f(0) = f(1))$: the final state is:

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

f balanced $(f(0) \neq f(1))$: the final state is:

$$\pm\frac{|0\rangle - |1\rangle}{\sqrt{2}}\frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{H} \pm|1\rangle|1\rangle$$

Cleve variation



Cleve variation

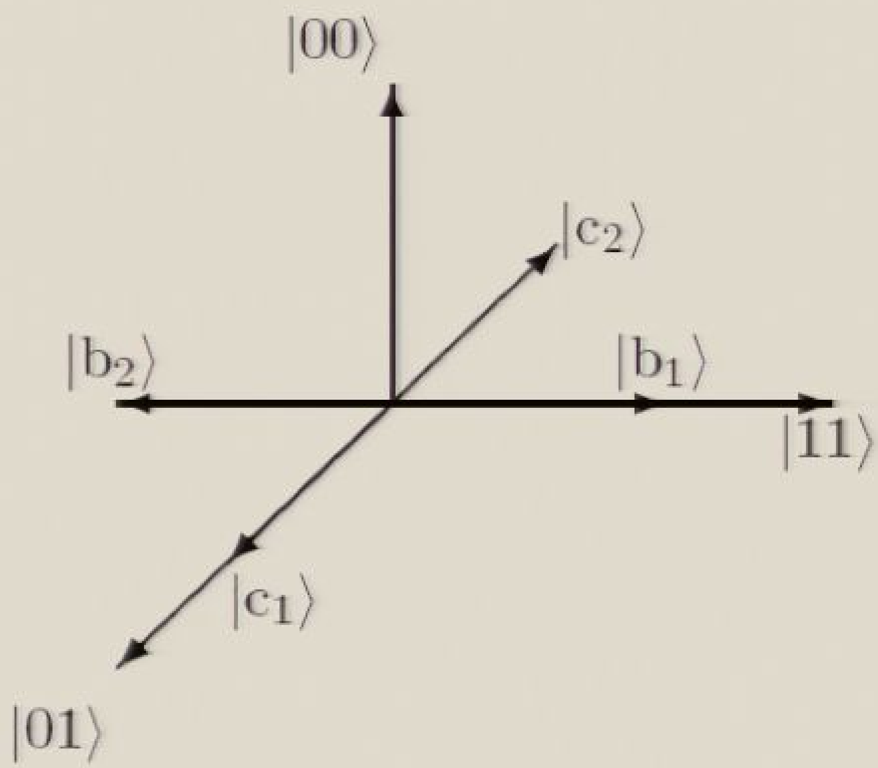
Note that U_f with $|1\rangle$ as the initial state in the output register either does

f constant ($f(0) = f(1)$): nothing

f balanced ($f(0) \neq f(1)$): reflects the state $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ of the input register in the line orthogonal to $|1\rangle$:

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Cleve variation



Cleve variation

So we end up with:

f constant ($f(0) = f(1)$): the final state is:

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

f balanced ($f(0) \neq f(1)$): the final state is:

$$\pm\frac{|0\rangle - |1\rangle}{\sqrt{2}}\frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{H} \pm|1\rangle|1\rangle$$

Cleve variation

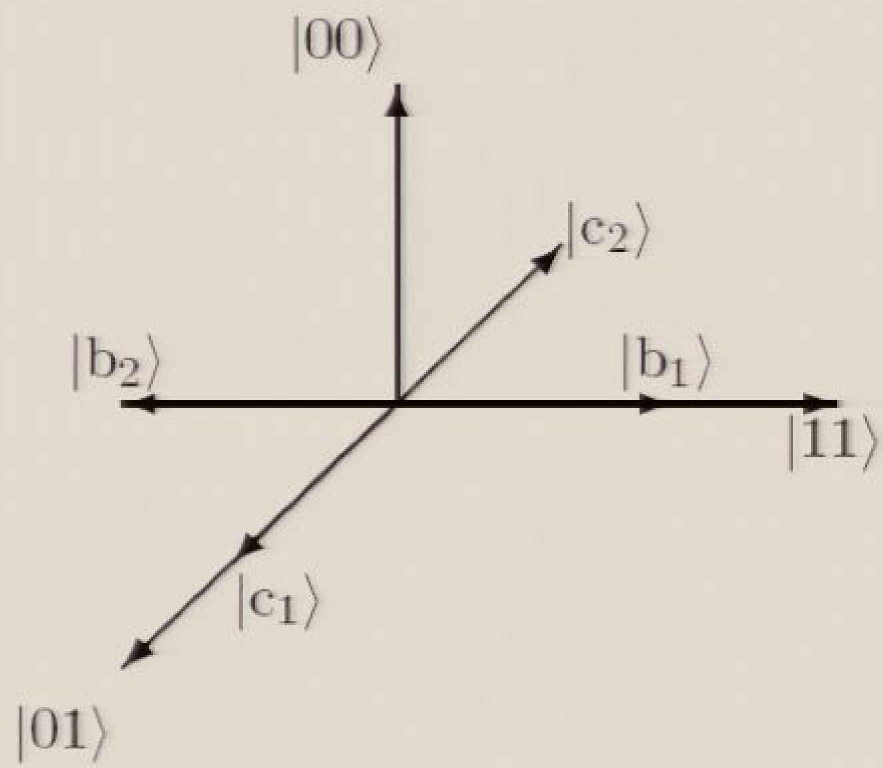
$$|0\rangle|1\rangle \xrightarrow{H} \frac{|0\rangle + |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} (-1)^{f(x)}|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

So:

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Cleve variation



Cleve variation

Note that U_f with $|1\rangle$ as the initial state in the output register either does

f constant $(f(0) = f(1))$: nothing

f balanced $(f(0) \neq f(1))$: reflects the state $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ of the input register in the line orthogonal to $|1\rangle$:

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Cleve variation

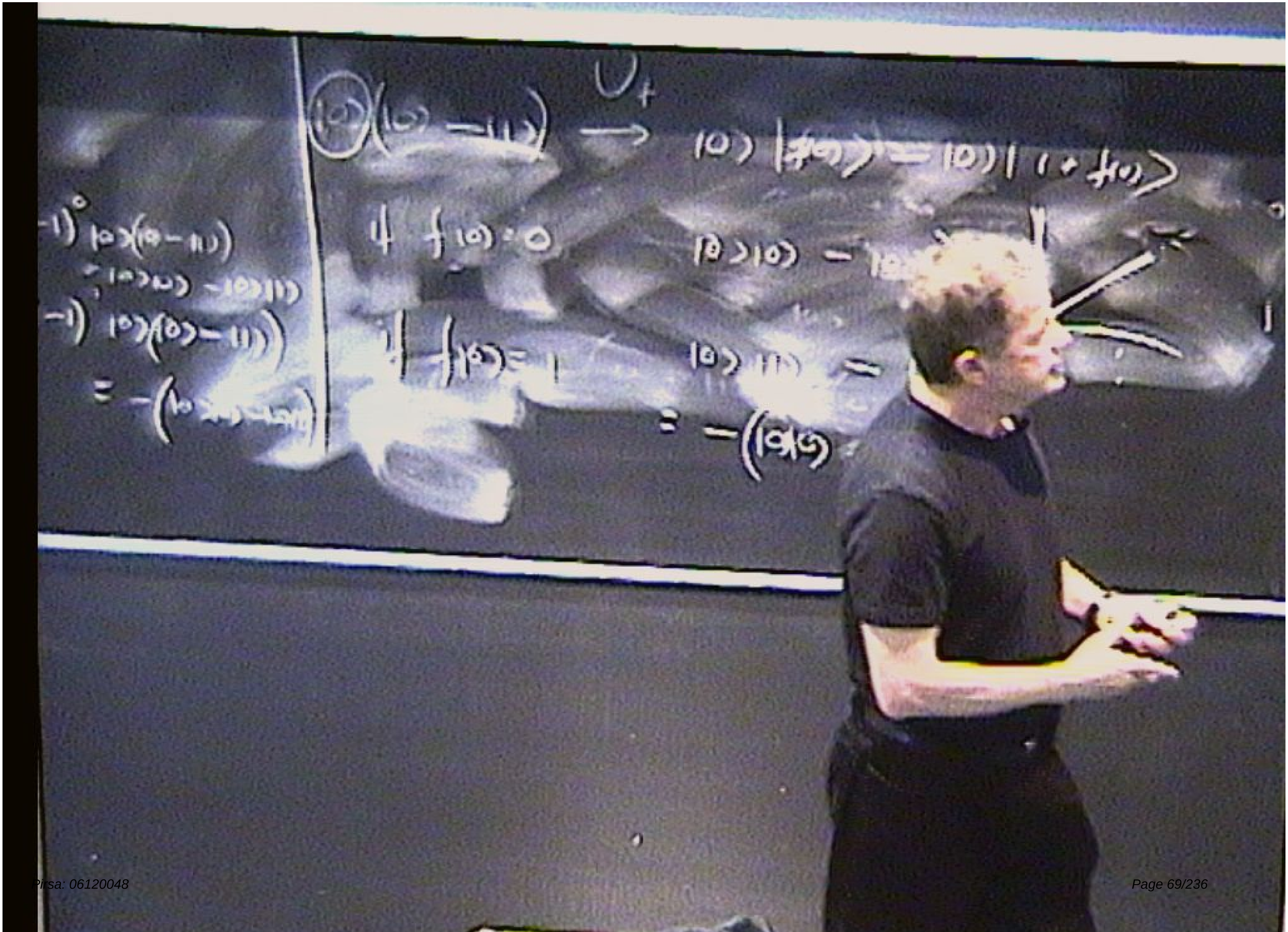
So we end up with:

f constant $(f(0) = f(1))$: the final state is:

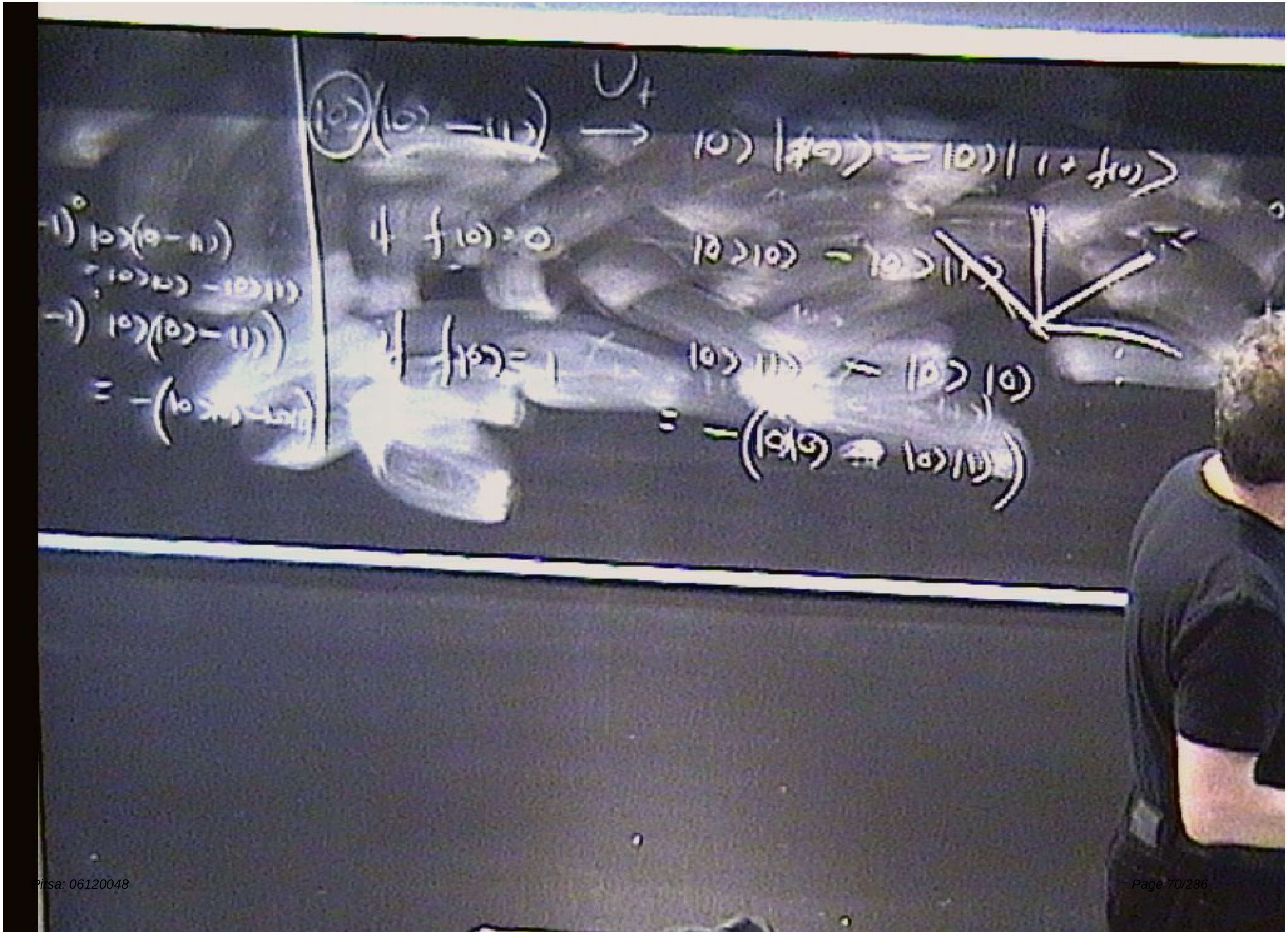
$$\pm\frac{|0\rangle + |1\rangle}{\sqrt{2}}\frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{H} \pm|0\rangle|1\rangle$$

f balanced $(f(0) \neq f(1))$: the final state is:

$$\pm\frac{|0\rangle - |1\rangle}{\sqrt{2}}\frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{H} \pm|1\rangle|1\rangle$$



U_f
|0⟩(|0⟩ − |1⟩) ⟶ |0⟩|f(0)⟩ = |0⟩|0 ⊕ f(0)⟩
If f(0) = 0
|0⟩|0⟩ −
If f(0) = 1
|0⟩|1⟩ −
= −(|0⟩|0⟩


U_f
if f(0) = 0
if f(0) = 1

Cleve variation

Note that U_f with $|1\rangle$ as the initial state in the output register either does

f constant ($f(0) = f(1)$): nothing

f balanced ($f(0) \neq f(1)$): reflects the state $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ of the input register in the line orthogonal to $|1\rangle$:

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Cleve variation

Note that U_f with $|1\rangle$ as the initial state in the output register either does

f constant ($f(0) = f(1)$): nothing

f balanced ($f(0) \neq f(1)$): reflects the state $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ of the input register in the line orthogonal to $|1\rangle$:

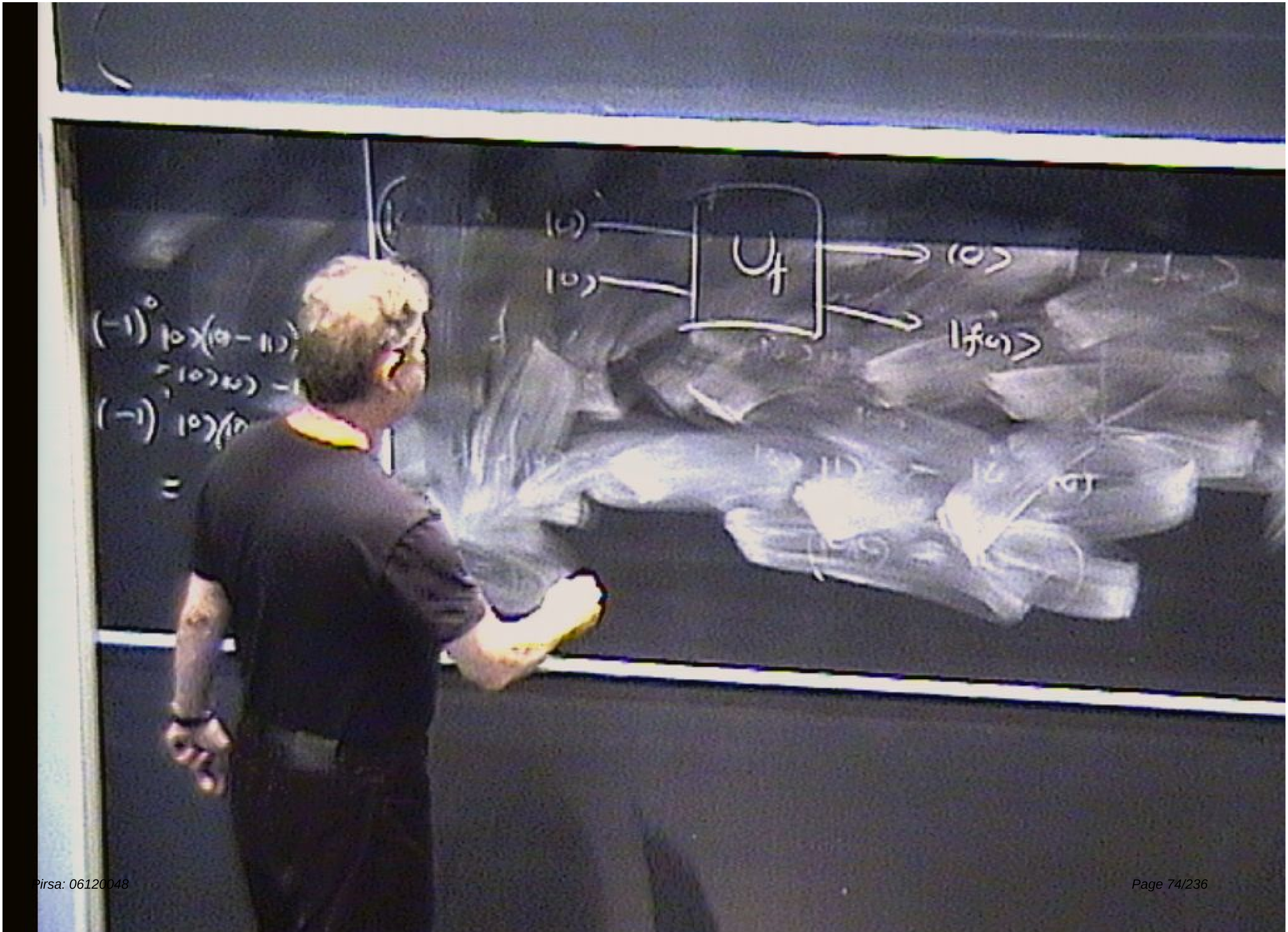
$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

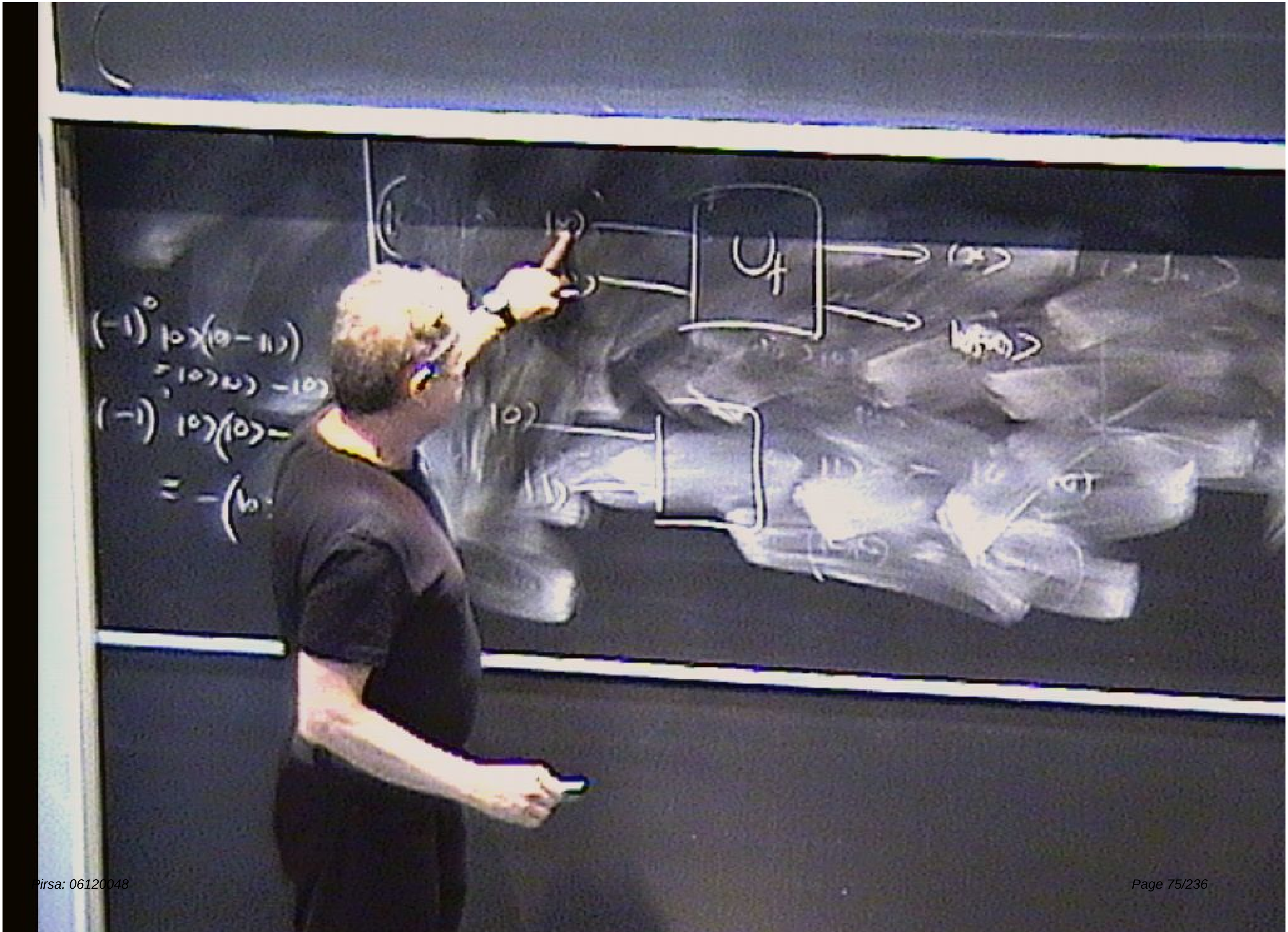
Cleve variation

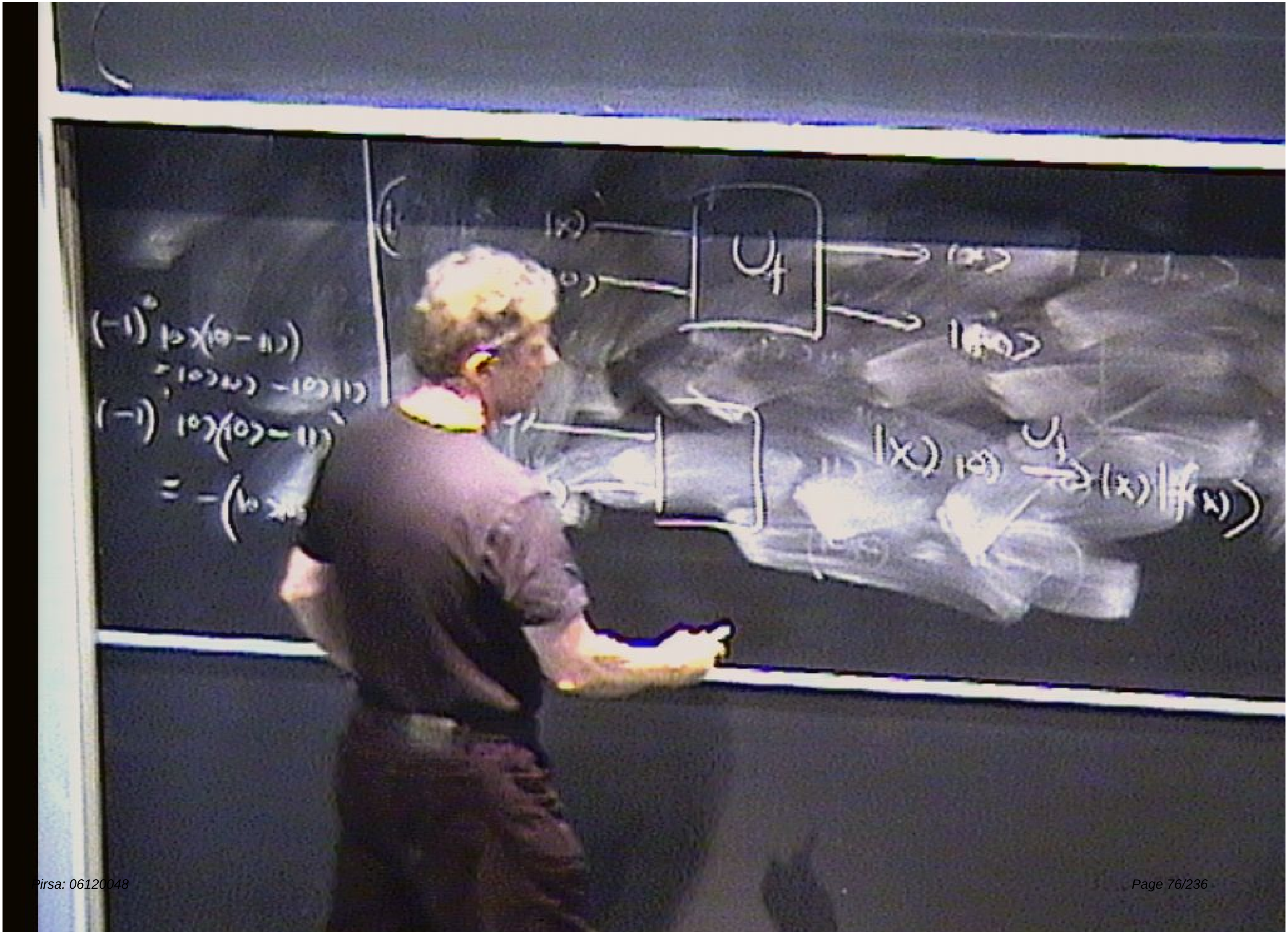
In general, if $f : B^n \rightarrow B$ maps all n-bit sequences except x_0 onto 0, then:

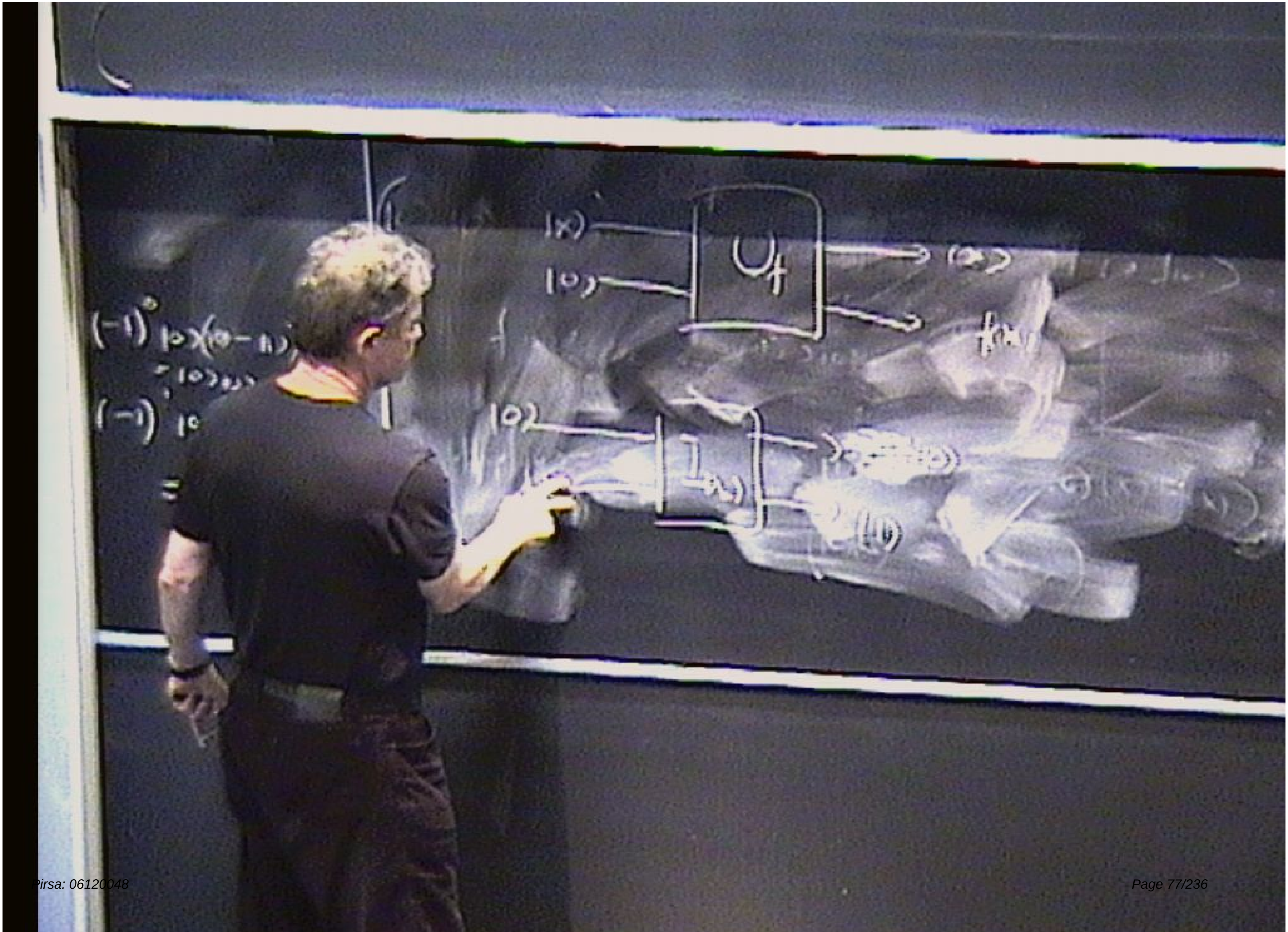
$$|\psi\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} I_{|x_0\rangle} |\psi\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

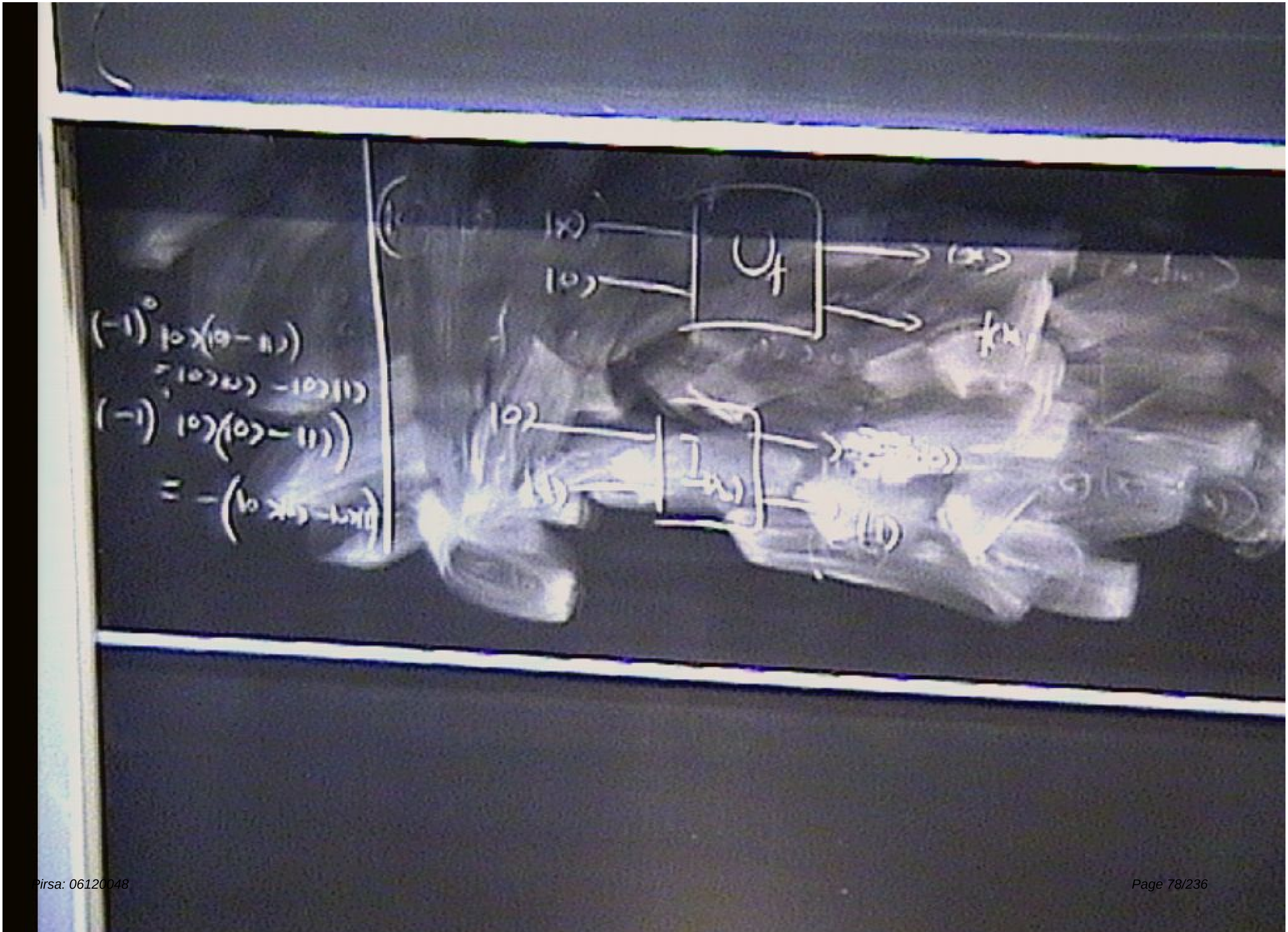
where $I_{|x_0\rangle}$ is a reflection in the hyperplane orthogonal to $|x_0\rangle$.


|0⟩
|0⟩
U_f
|0⟩
|f(0)⟩


U_f


U_f
$|x\rangle|0\rangle \xrightarrow{U_f} |x\rangle|f(x)\rangle$




|x⟩
|0⟩
U_f
|x⟩
f(x)
(−1)^0 |0⟩(|0⟩ − |1⟩)

Deutsch's problem

- Generalization to 'Deutsch's problem': determine whether a Boolean function $f : B^n \to B$ is constant or whether it is balanced, where it is promised that the function is either constant or balanced.
- 'Balanced' here means that the function takes the values 0 and 1 an equal number of times, i.e., 2^{n-1} times each.

Deutsch-Jozsa algorithm

$$|0\rangle^{\otimes n}|1\rangle \xrightarrow{H} \sum_{x \in B^n} \frac{|x\rangle}{\sqrt{2^n}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$\xrightarrow{U_f} \sum_{x} \frac{(-1)^{f(x)}}{\sqrt{2^n}} |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$\xrightarrow{H} \sum_{y} \sum_{x} \frac{(-1)^{x \cdot y + f(x)}}{\sqrt{2^n}} |y\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Deutsch-Jozsa algorithm

$$|0\rangle^{\otimes n}|1\rangle \xrightarrow{H} \sum_{x\in B^n} \frac{|x\rangle}{\sqrt{2^n}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$\xrightarrow{U_f} \sum_{x} \frac{(-1)^{f(x)}}{\sqrt{2^n}}|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

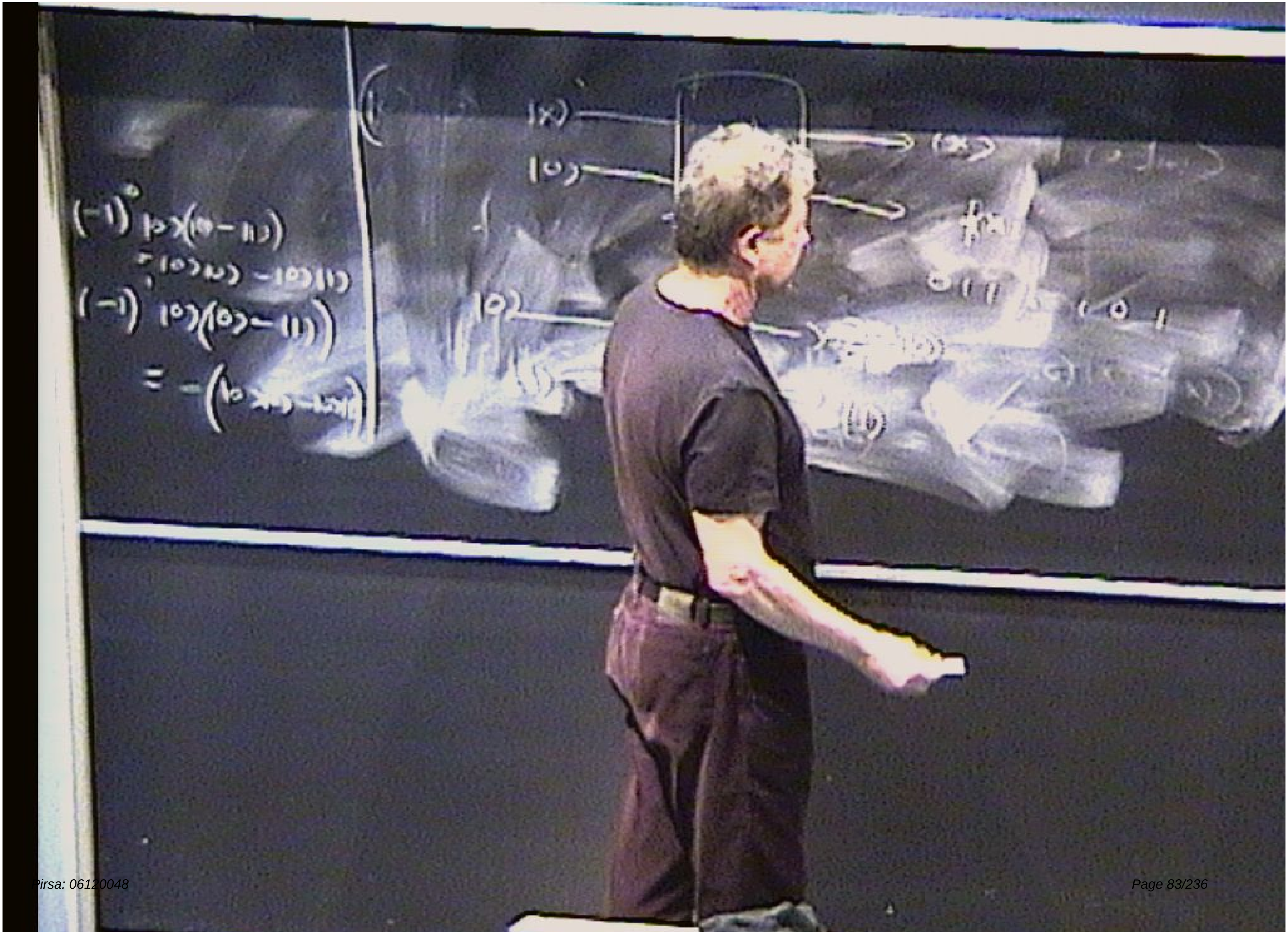
$$\xrightarrow{H} \sum_{y}\sum_{x} \frac{(-1)^{x\cdot y+f(x)}}{\sqrt{2^n}}|y\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

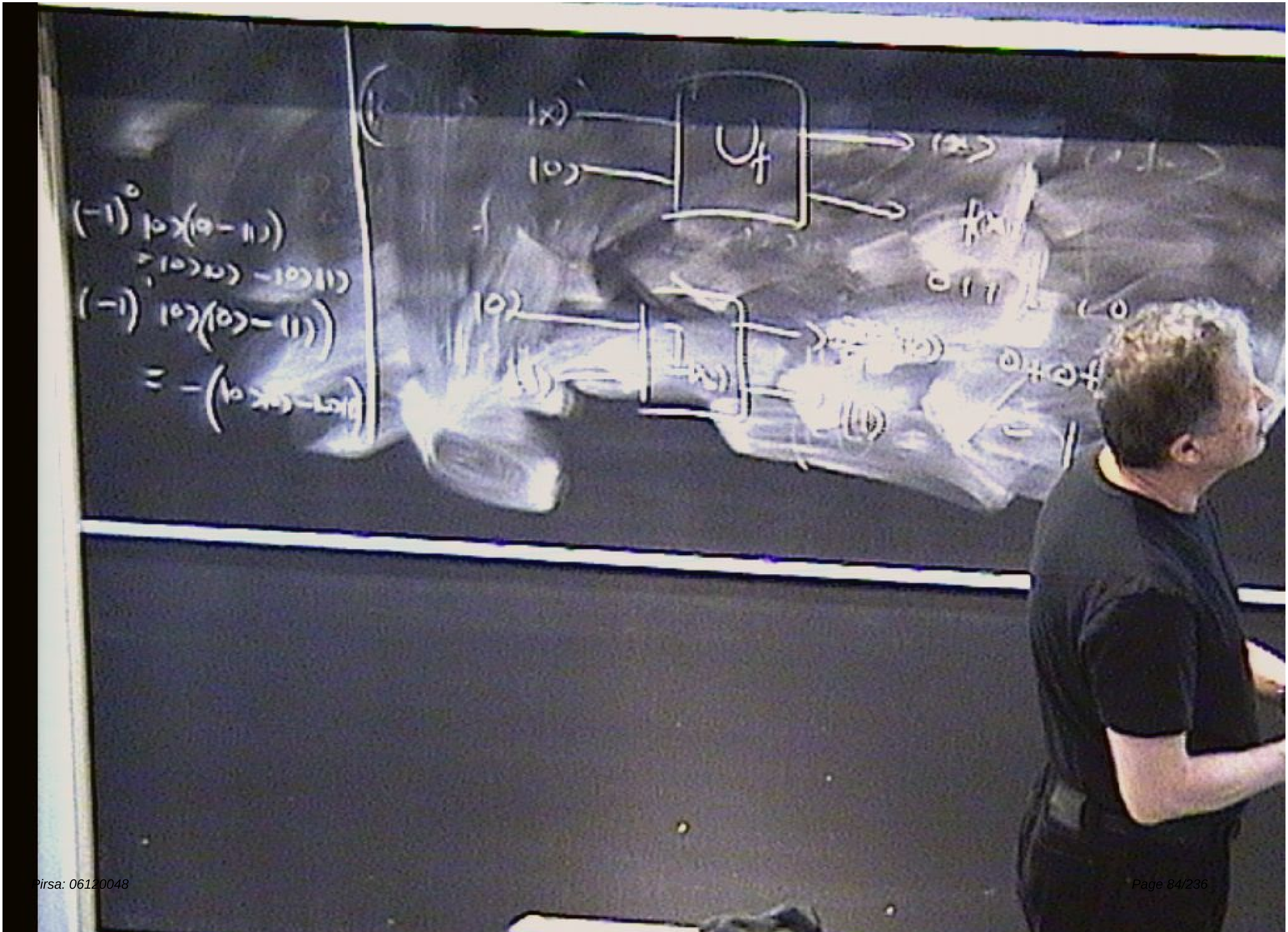
Deutsch-Jozsa algorithm

$$|0\rangle^{\otimes n}|1\rangle \xrightarrow{H} \sum_{x\in B^n} \frac{|x\rangle}{\sqrt{2^n}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$\xrightarrow{U_f} \sum_{x} \frac{(-1)^{f(x)}}{\sqrt{2^n}}|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$\xrightarrow{H} \sum_{y}\sum_{x} \frac{(-1)^{x\cdot y+f(x)}}{\sqrt{2^n}}|y\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$



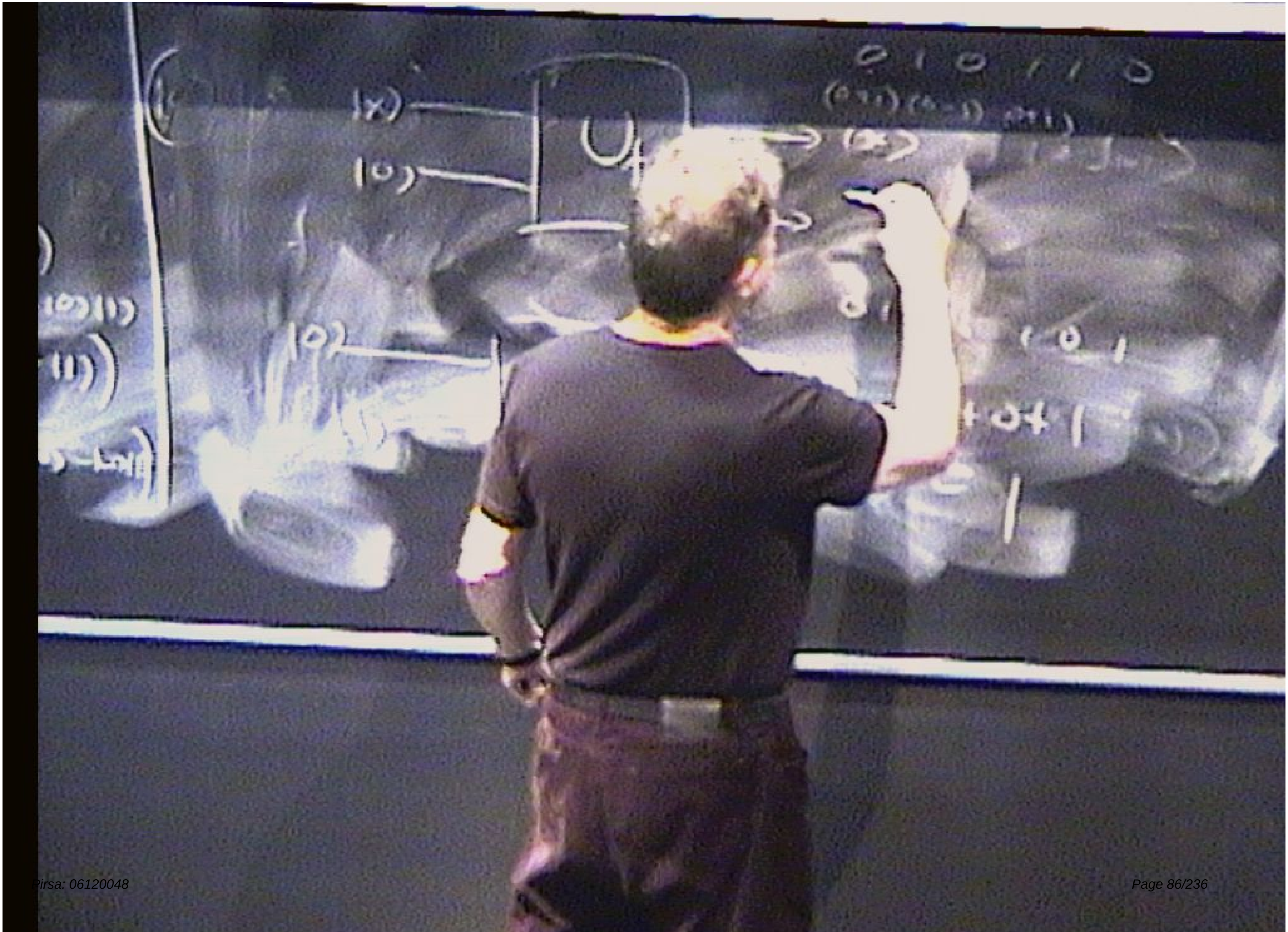

(-1)^0 |0⟩(|0⟩ - |1⟩)
= |0⟩|0⟩ - |0⟩|1⟩
(-1)^1 |0⟩(|0⟩ - |1⟩)
= -(|0⟩|0⟩ - |0⟩|1⟩)
|x⟩
|0⟩
U_f
|x⟩
f(x)

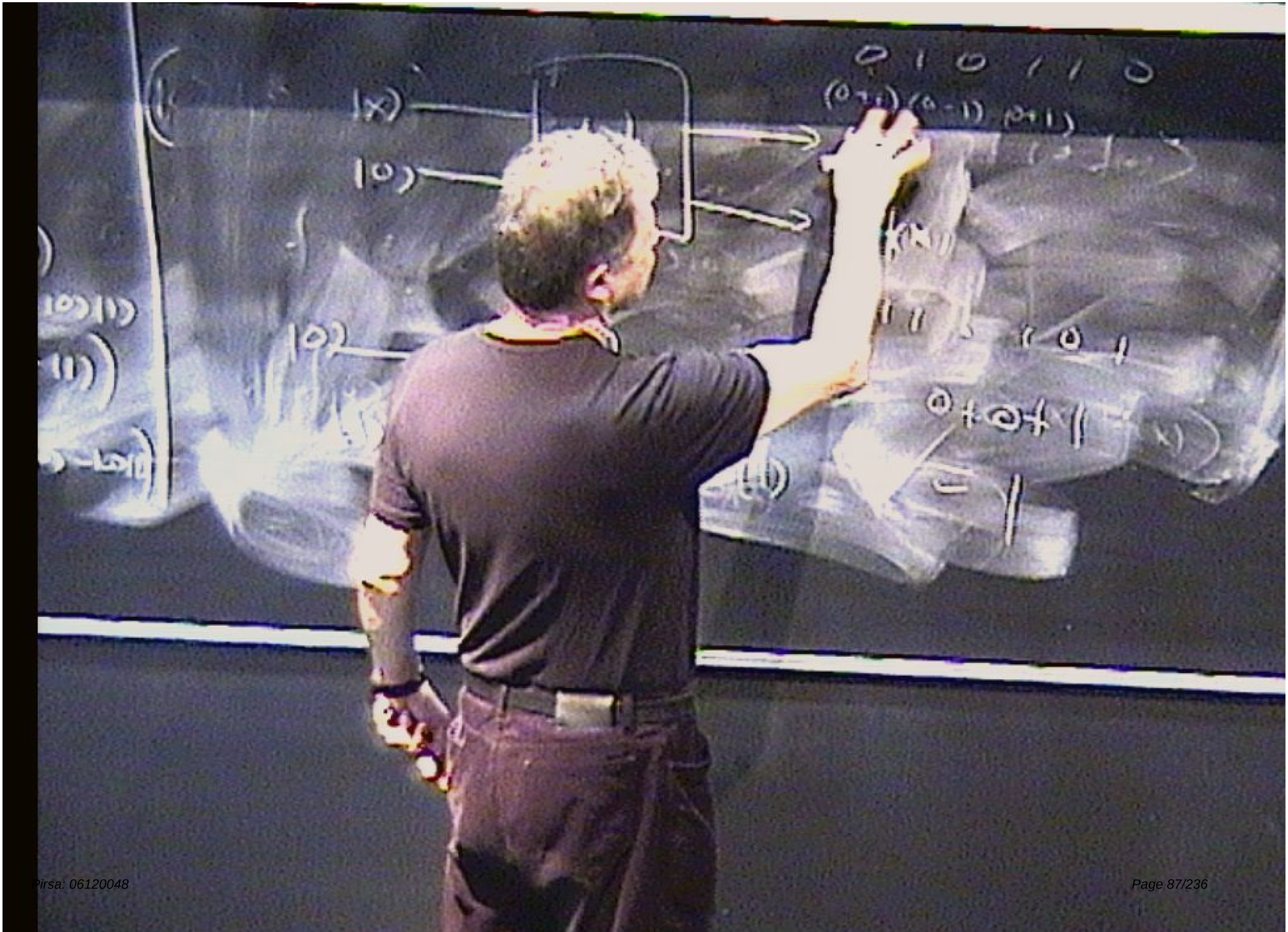
Deutsch-Jozsa algorithm

$$|0\rangle^{\otimes n}|1\rangle \xrightarrow{H} \sum_{x\in B^n} \frac{|x\rangle}{\sqrt{2^n}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$\xrightarrow{U_f} \sum_{x} \frac{(-1)^{f(x)}}{\sqrt{2^n}} |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$\xrightarrow{H} \sum_{y} \sum_{x} \frac{(-1)^{x\cdot y + f(x)}}{\sqrt{2^n}} |y\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$


0 1 0 1 1 0
|x⟩
|0⟩
U



Deutsch-Jozsa algorithm

$$|0\rangle^{\otimes n}|1\rangle \xrightarrow{H} \sum_{x \in B^n} \frac{|x\rangle}{\sqrt{2^n}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$\xrightarrow{U_f} \sum_{x} \frac{(-1)^{f(x)}}{\sqrt{2^n}} |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$\xrightarrow{H} \sum_{y} \sum_{x} \frac{(-1)^{x \cdot y + f(x)}}{\sqrt{2^n}} |y\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Deutsch-Jozsa algorithm

Case n = 2

$$|0\rangle^{\otimes 2}|1\rangle \xrightarrow{H} \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

After U_f, state is:

f constant:

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

f balanced:

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

with two $+$'s and two $-$'s.

Deutsch-Jozsa algorithm

$$|0\rangle^{\otimes n}|1\rangle \xrightarrow{H} \sum_{x\in B^n} \frac{|x\rangle}{\sqrt{2^n}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$\xrightarrow{U_f} \sum_{x} \frac{(-1)^{f(x)}}{\sqrt{2^n}} |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$\xrightarrow{H} \sum_{y} \sum_{x} \frac{(-1)^{x\cdot y + f(x)}}{\sqrt{2^n}} |y\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Deutsch-Jozsa algorithm

Case n = 2

$$|0\rangle^{\otimes 2}|1\rangle \xrightarrow{\text{H}} \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

After $\mathrm{U_f}$, state is:

f constant:

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

f balanced:

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

with two $+$'s and two $-$'s.

Deutsch-Jozsa algorithm

After final H, state of input register is:

f constant: $\pm|00\rangle$

f balanced: $\pm|01\rangle$ or $\pm|10\rangle$ or $\pm|11\rangle$

Deutsch-Jozsa algorithm

Case n = 2

$$|0\rangle^{\otimes 2}|1\rangle \xrightarrow{H} \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

After U_f, state is:

f constant:

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

f balanced:

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

with two +'s and two −'s.

Deutsch-Jozsa algorithm

After final H, state of input register is:

f constant: $\pm|00\rangle$

f balanced: $\pm|01\rangle$ or $\pm|10\rangle$ or $\pm|11\rangle$

Simon's problem

- Simon's problem: find the period r of a periodic function $\mathrm{f} : \mathrm{B}^{\mathrm{n}} \rightarrow \mathrm{B}^{\mathrm{n}}$, i.e., a Boolean function for which

$$\mathrm{f}(\mathrm{x_i}) = \mathrm{f}(\mathrm{x_j}) \text{ if and only if } \mathrm{x_j} = \mathrm{x_i} \oplus \mathrm{r} \text{ for all } \mathrm{x_i}, \mathrm{x_j} \in \mathrm{B}^{\mathrm{n}}$$

- Since $\mathrm{x} \oplus \mathrm{r} \oplus \mathrm{r} = \mathrm{x}$, the function is 2-to-1.

Simon's problem

- Since f is periodic, the possible outputs of f—the values of f for the different inputs—partition the set of input values into mutually exclusive and collectively exhaustive subsets, and these subsets depend on the period.
- Determining the period of f amounts to distinguishing the partition corresponding to the period from alternative partitions corresponding to alternative possible periods.

Simon's Algorithm—quantum logical picture

- Consider case $n = 2$. Apply H to input state $|00\rangle$, then U_f.
- There are $2^2 - 1 = 3$ possible values of the period r: 01, 10, 11, and the corresponding partitions are:

$$\begin{aligned} r = 01 &: \{00, 01\}, \{10, 11\} \\ r = 10 &: \{00, 10\}, \{01, 11\} \\ r = 11 &: \{00, 11\}, \{01, 10\} \end{aligned}$$

Simon's problem

- Since f is periodic, the possible outputs of f—the values of f for the different inputs—partition the set of input values into mutually exclusive and collectively exhaustive subsets, and these subsets depend on the period.
- Determining the period of f amounts to distinguishing the partition corresponding to the period from alternative partitions corresponding to alternative possible periods.

Simon's problem

- Simon's problem: find the period r of a periodic function $f : B^n \to B^n$, i.e., a Boolean function for which

$$f(x_i) = f(x_j) \text{ if and only if } x_j = x_i \oplus r \text{ for all } x_i, x_j \in B^n$$

- Since $x \oplus r \oplus r = x$, the function is 2-to-1.

Simon's Algorithm—quantum logical picture

- Consider case $n = 2$. Apply H to input state $|00\rangle$, then U_f.
- There are $2^2 - 1 = 3$ possible values of the period r: 01, 10, 11, and the corresponding partitions are:

$$\begin{aligned} r = 01 &: \{00, 01\}, \{10, 11\} \\ r = 10 &: \{00, 10\}, \{01, 11\} \\ r = 11 &: \{00, 11\}, \{01, 10\} \end{aligned}$$

Simon's Algorithm—quantum logical picture

- States of the input and output registers after U_f are:

$$r = 01: \ (|00\rangle + |01\rangle)|f(00)\rangle + (|10\rangle + |11\rangle)|f(10)\rangle$$
$$r = 10: \ (|00\rangle + |10\rangle)|f(00)\rangle + (|01\rangle + |11\rangle)|f(01)\rangle$$
$$r = 11: \ (|00\rangle + |11\rangle)|f(00)\rangle + (|01\rangle + |10\rangle)|f(01)\rangle$$

$n = 2$ case reduces to Deutsch's XOR algorithm

- The $n = 2$ case reduces to the same geometric construction as in Deutsch's XOR algorithm.

  $r = 10$: input register states are $|c_1\rangle = |00\rangle + |10\rangle$ or $|c_2\rangle = |01\rangle + |11\rangle$, depending on the outcome of the measurement of the output register.

  $r = 11$: input register states are $|b_1\rangle = |00\rangle + |11\rangle$ or $|b_2\rangle = |01\rangle + |10\rangle$, depending on the outcome of the measurement of the output register.

- So the three possible periods are associated with three orthogonal planes in $\mathcal{H}^2 \otimes \mathcal{H}^2$, which correspond to the constant and balanced planes in Deutsch's XOR algorithm, and a third orthogonal plane, all three planes intersecting in the line spanned by the vector $|00\rangle$.

Simon's Algorithm—quantum logical picture

- States of the input and output registers after U_f are:

$$r = 01: \ (|00\rangle + |01\rangle)|f(00)\rangle + (|10\rangle + |11\rangle)|f(10)\rangle$$
$$r = 10: \ (|00\rangle + |10\rangle)|f(00)\rangle + (|01\rangle + |11\rangle)|f(01)\rangle$$
$$r = 11: \ (|00\rangle + |11\rangle)|f(00)\rangle + (|01\rangle + |10\rangle)|f(01)\rangle$$

$n = 2$ case reduces to Deutsch's XOR algorithm

- The $n = 2$ case reduces to the same geometric construction as in Deutsch's XOR algorithm.

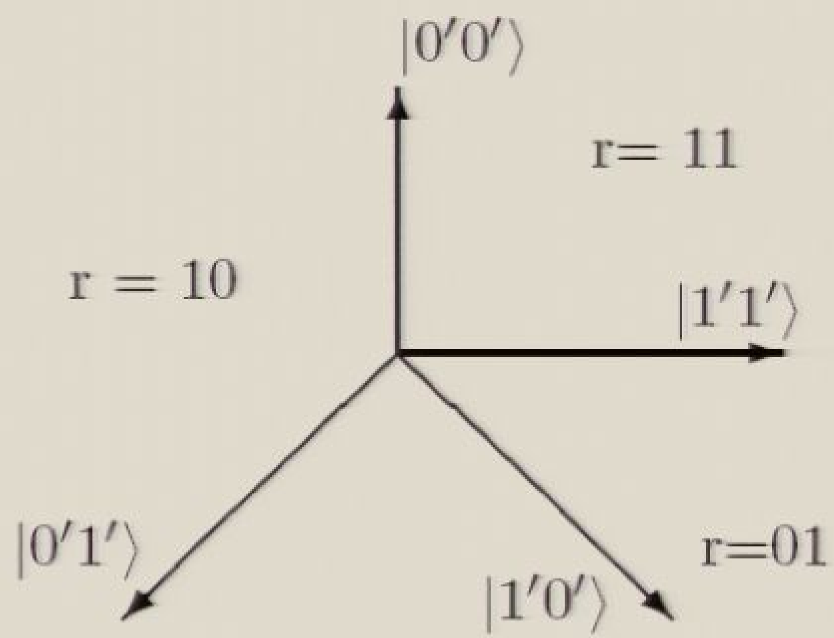
  $r = 10$: input register states are $|c_1\rangle = |00\rangle + |10\rangle$ or $|c_2\rangle = |01\rangle + |11\rangle$, depending on the outcome of the measurement of the output register.

  $r = 11$: input register states are $|b_1\rangle = |00\rangle + |11\rangle$ or $|b_2\rangle = |01\rangle + |10\rangle$, depending on the outcome of the measurement of the output register.

- So the three possible periods are associated with three orthogonal planes in $\mathcal{H}^2 \otimes \mathcal{H}^2$, which correspond to the constant and balanced planes in Deutsch's XOR algorithm, and a third orthogonal plane, all three planes intersecting in the line spanned by the vector $|00\rangle$.

Simon's algorithm: n = 2 case

In the Hadamard basis obtained by applying **H**:



$n = 2$ case reduces to Deutsch's XOR algorithm

- The $n = 2$ case reduces to the same geometric construction as in Deutsch's XOR algorithm.

  $r = 10$: input register states are $|c_1\rangle = |00\rangle + |10\rangle$ or $|c_2\rangle = |01\rangle + |11\rangle$, depending on the outcome of the measurement of the output register.

  $r = 11$: input register states are $|b_1\rangle = |00\rangle + |11\rangle$ or $|b_2\rangle = |01\rangle + |10\rangle$, depending on the outcome of the measurement of the output register.

- So the three possible periods are associated with three orthogonal planes in $\mathcal{H}^2 \otimes \mathcal{H}^2$, which correspond to the constant and balanced planes in Deutsch's XOR algorithm, and a third orthogonal plane, all three planes intersecting in the line spanned by the vector $|00\rangle$.

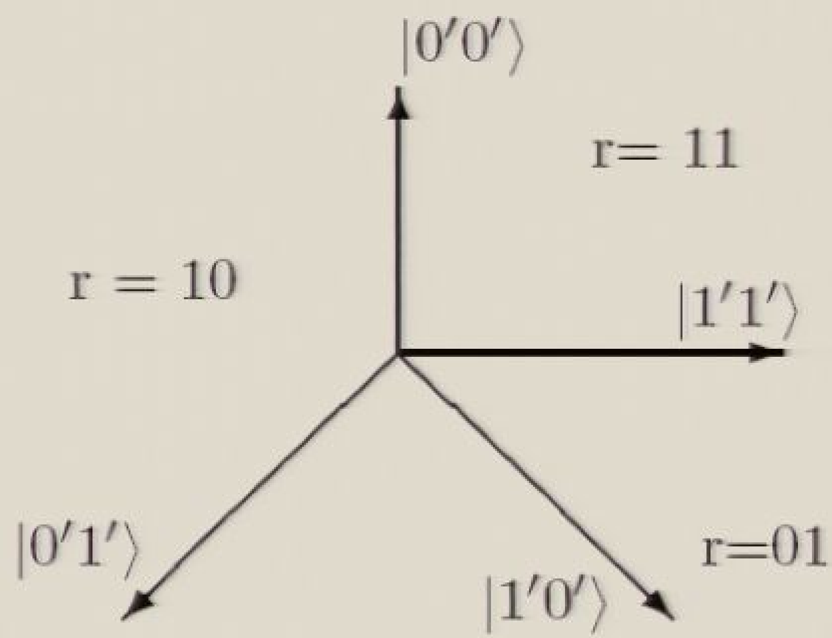
Simon's Algorithm—quantum logical picture

- States of the input and output registers after U_f are:

$$\mathrm{r} = 01: \ (|00\rangle + |01\rangle)|f(00)\rangle + (|10\rangle + |11\rangle)|f(10)\rangle$$
$$\mathrm{r} = 10: \ (|00\rangle + |10\rangle)|f(00)\rangle + (|01\rangle + |11\rangle)|f(01)\rangle$$
$$\mathrm{r} = 11: \ (|00\rangle + |11\rangle)|f(00)\rangle + (|01\rangle + |10\rangle)|f(01)\rangle$$

Simon's algorithm: n = 2 case

In the Hadamard basis obtained by applying **H**:



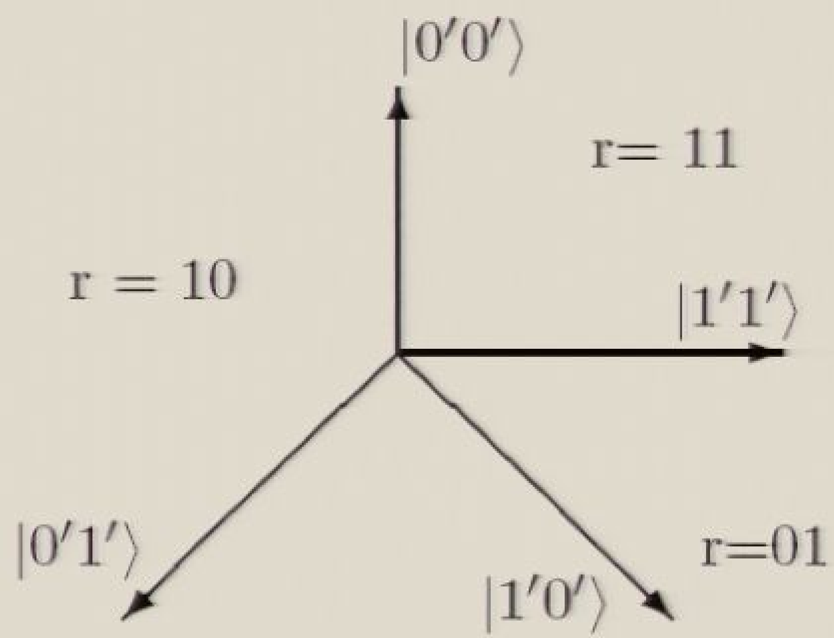
Simon's Algorithm—quantum logical picture

- States of the input and output registers after $\mathrm{U_f}$ are:

$$r = 01 : \; (|00\rangle + |01\rangle)|f(00)\rangle + (|10\rangle + |11\rangle)|f(10)\rangle$$
$$r = 10 : \; (|00\rangle + |10\rangle)|f(00)\rangle + (|01\rangle + |11\rangle)|f(01)\rangle$$
$$r = 11 : \; (|00\rangle + |11\rangle)|f(00)\rangle + (|01\rangle + |10\rangle)|f(01)\rangle$$

Simon's algorithm: n = 2 case

In the Hadamard basis obtained by applying **H**:

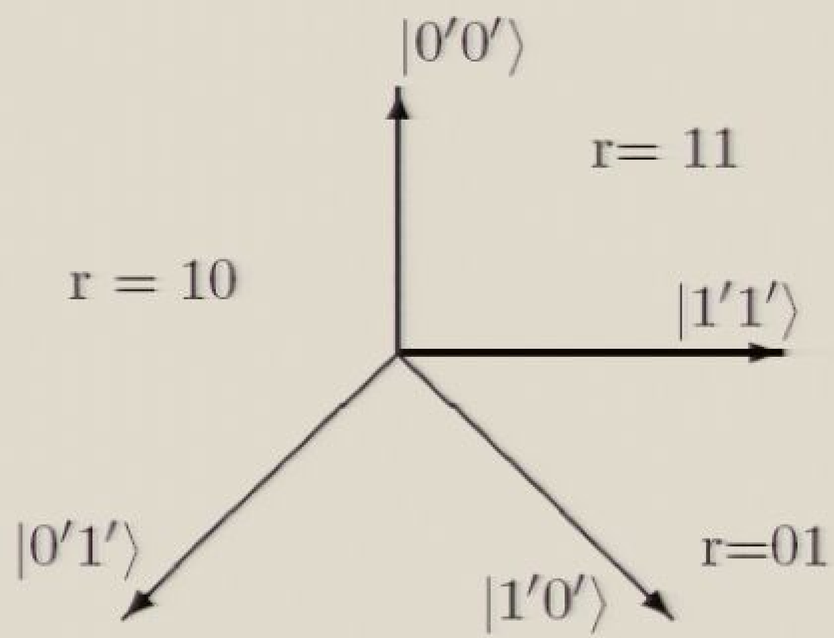


Simon's algorithm: n = 2 case

- As in Deutsch's algorithm, a final Hadamard transformation $|0'0'\rangle \xrightarrow{H} |00\rangle$, etc. amounts to dropping the primes. Allows the plane corresponding to the period to be identified by a measurement in the computational basis, except when the state of the register is projected by the measurement onto the state $|00\rangle$.
- So the algorithm will generally have to be repeated until we find an outcome that is not 00.

Simon's algorithm: n = 2 case

In the Hadamard basis obtained by applying **H**:



Simon's algorithm: n = 2 case

- As in Deutsch's algorithm, a final Hadamard transformation $|0'0'\rangle \xrightarrow{H} |00\rangle$, etc. amounts to dropping the primes. Allows the plane corresponding to the period to be identified by a measurement in the computational basis, except when the state of the register is projected by the measurement onto the state $|00\rangle$.
- So the algorithm will generally have to be repeated until we find an outcome that is not 00.

Simon's algorithm: $n = 3$ case

- We can see what happens in the general case if we consider the case $n = 3$.
- There are now seven possible periods: 001, 010, 011, 100, 101, 110, 111.

Simon's algorithm: n = 2 case

- As in Deutsch's algorithm, a final Hadamard transformation $|0'0'\rangle \xrightarrow{\mathrm{H}} |00\rangle$, etc. amounts to dropping the primes. Allows the plane corresponding to the period to be identified by a measurement in the computational basis, except when the state of the register is projected by the measurement onto the state $|00\rangle$.
- So the algorithm will generally have to be repeated until we find an outcome that is not 00.

Simon's algorithm: n = 3 case

- We can see what happens in the general case if we consider the case $n = 3$.
- There are now seven possible periods: 001, 010, 011, 100, 101, 110, 111.

Simon's algorithm: n = 3 case

- Period $r = 001$: the state of the two registers after the unitary transformation U_f is:

$$(|000\rangle + |001\rangle)|f(000)\rangle + (|010\rangle + |011\rangle)|f(010)\rangle$$
$$+(|100\rangle + |101\rangle)|f(100)\rangle + (|110\rangle + |111\rangle)|f(110)\rangle$$

- Measure the output register $\Longrightarrow$ the input register is left in one of four states, depending on the outcome of the measurement:

$$\begin{aligned}
|000\rangle + |001\rangle &= |0'0'0'\rangle + |0'1'0'\rangle + |1'0'0'\rangle + |1'1'0'\rangle \\
|010\rangle + |011\rangle &= |0'0'0'\rangle - |0'1'0'\rangle + |1'0'0'\rangle - |1'1'0'\rangle \\
|100\rangle + |101\rangle &= |0'0'0'\rangle + |0'1'0'\rangle - |1'0'0'\rangle - |1'1'0'\rangle \\
|110\rangle + |111\rangle &= |0'0'0'\rangle - |0'1'0'\rangle - |1'0'0'\rangle + |1'1'0'\rangle
\end{aligned}$$

Simon's algorithm: $n = 3$ case

- Applying a Hadamard transformation amounts to dropping the primes.
- So if the period is $r = 001$, the state of the input register ends up in the 4-dimensional subspace spanned by the vectors: $|000\rangle, |010\rangle, |100\rangle, |110\rangle$.

Simon's algorithm: n = 3 case

- Period $r = 001$: the state of the two registers after the unitary transformation U_f is:

$$(|000\rangle + |001\rangle)|f(000)\rangle + (|010\rangle + |011\rangle)|f(010)\rangle$$
$$+(|100\rangle + |101\rangle)|f(100)\rangle + (|110\rangle + |111\rangle)|f(110)\rangle$$

- Measure the output register $\Longrightarrow$ the input register is left in one of four states, depending on the outcome of the measurement:

$$\begin{aligned}
|000\rangle + |001\rangle &= |0'0'0'\rangle + |0'1'0'\rangle + |1'0'0'\rangle + |1'1'0'\rangle \\
|010\rangle + |011\rangle &= |0'0'0'\rangle - |0'1'0'\rangle + |1'0'0'\rangle - |1'1'0'\rangle \\
|100\rangle + |101\rangle &= |0'0'0'\rangle + |0'1'0'\rangle - |1'0'0'\rangle - |1'1'0'\rangle \\
|110\rangle + |111\rangle &= |0'0'0'\rangle - |0'1'0'\rangle - |1'0'0'\rangle + |1'1'0'\rangle
\end{aligned}$$

Simon's algorithm: n = 3 case

- Applying a Hadamard transformation amounts to dropping the primes.
- So if the period is $r = 001$, the state of the input register ends up in the 4-dimensional subspace spanned by the vectors: $|000\rangle, |010\rangle, |100\rangle, |110\rangle$.

Simon's algorithm: n = 3 case

A similar analysis applies to the other six possible periods. The corresponding subspaces are spanned by the following vectors:

$r = 001$: $|000\rangle, |010\rangle, |100\rangle, |110\rangle$

$r = 010$: $|000\rangle, |001\rangle, |100\rangle, |101\rangle$

$r = 011$: $|000\rangle, |011\rangle, |100\rangle, |111\rangle$

$r = 100$: $|000\rangle, |001\rangle, |010\rangle, |011\rangle$

$r = 101$: $|000\rangle, |010\rangle, |101\rangle, |111\rangle$

$r = 110$: $|000\rangle, |001\rangle, |110\rangle, |111\rangle$

$r = 111$: $|000\rangle, |011\rangle, |101\rangle, |110\rangle$

Simon's algorithm: $n = 3$ case

- These subspaces are orthogonal except for intersections in 2-dimensional planes. The period can be found by measuring in the computational basis.
- Repetitions of the measurement will eventually yield sufficiently many distinct values to determine the subspace containing the final state.
- In this case, it is clear by examining the above list that two values distinct from 000 suffice to determine the subspace. Note these are just the values y_i for which $y_i \cdot r = 0$.

Simon's algorithm: n = 3 case

A similar analysis applies to the other six possible periods. The corresponding subspaces are spanned by the following vectors:

$r = 001$: $|000\rangle, |010\rangle, |100\rangle, |110\rangle$

$r = 010$: $|000\rangle, |001\rangle, |100\rangle, |101\rangle$

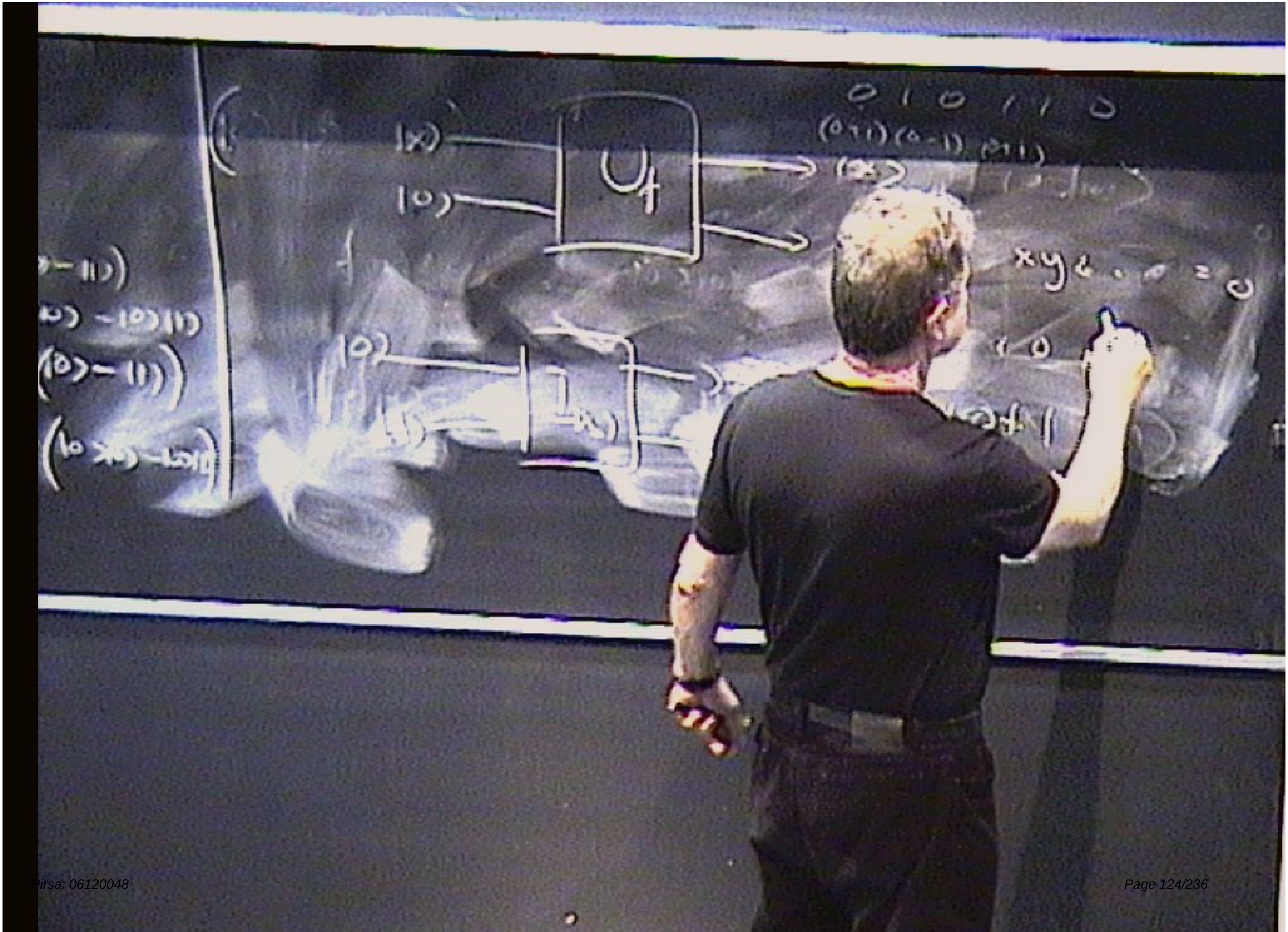
$r = 011$: $|000\rangle, |011\rangle, |100\rangle, |111\rangle$

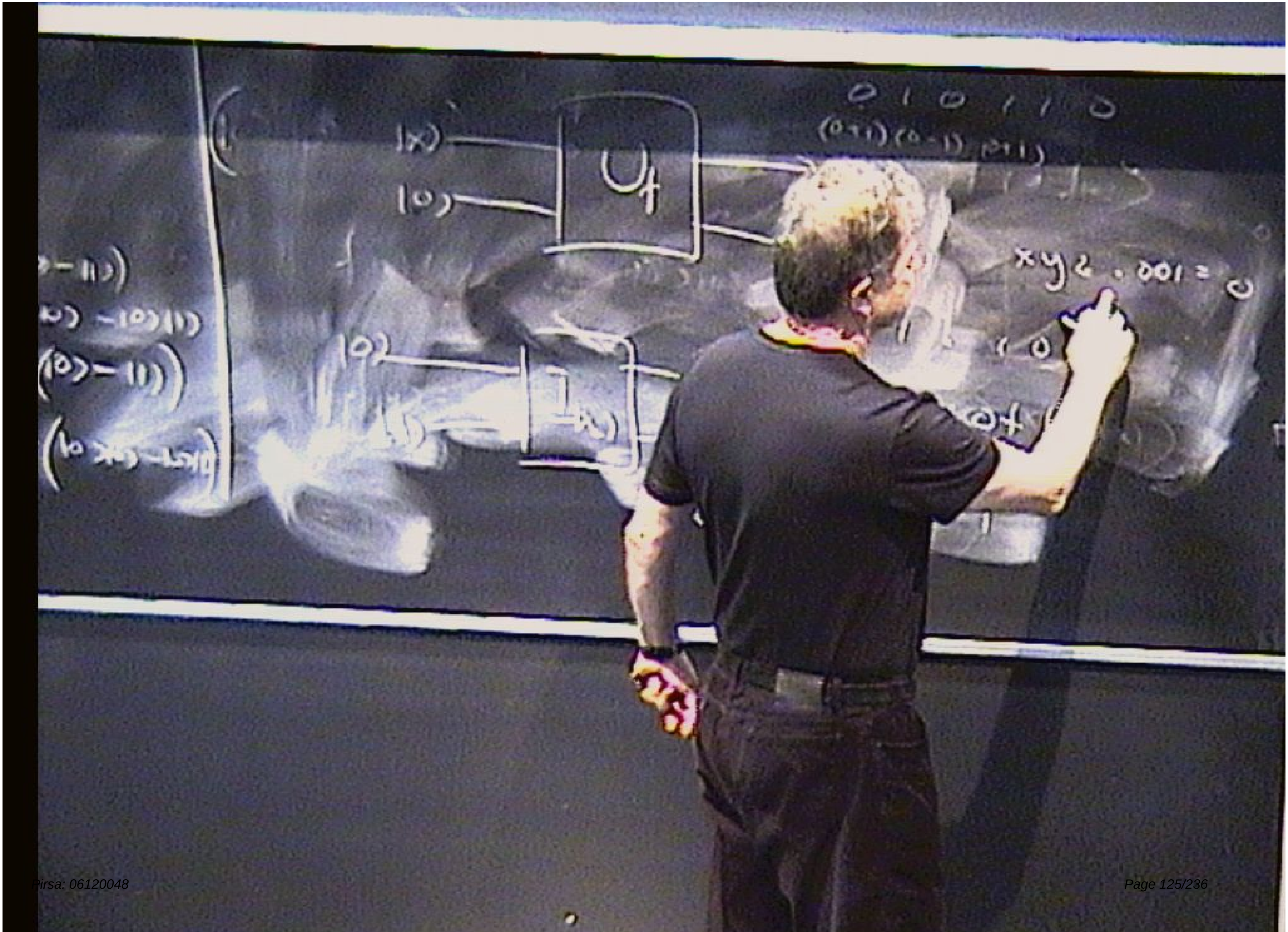
$r = 100$: $|000\rangle, |001\rangle, |010\rangle, |011\rangle$

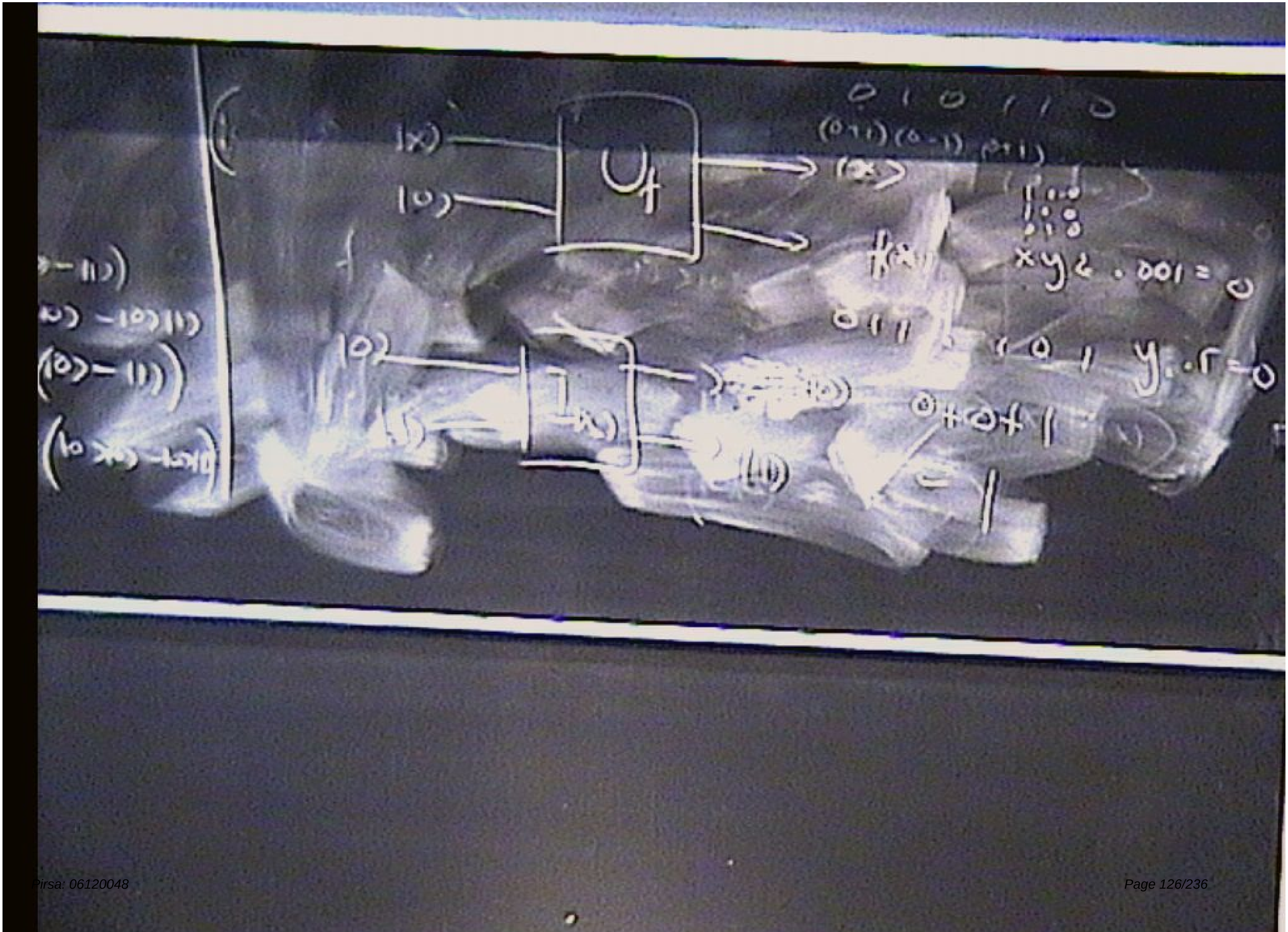
$r = 101$: $|000\rangle, |010\rangle, |101\rangle, |111\rangle$

$r = 110$: $|000\rangle, |001\rangle, |110\rangle, |111\rangle$

$r = 111$: $|000\rangle, |011\rangle, |101\rangle, |110\rangle$


|x⟩
|0⟩
U_f
0 1 0 1 1 0


|x⟩
|0⟩
U_f
0 1 0 1 1 0
xyz . 001 = 0


|x⟩
|0⟩
U_f
f(x)

Shor's factorization algorithm

- The problem of factorizing a composite integer N that is the product of two primes, $N = pq$, reduces to the problem of finding the period of a function $f(x) = a^x \bmod N$, for any $a < N$ which is coprime to N, i.e., has no common factors with N other than 1.

Shor's factorization algorithm

- The problem of factorizing a composite integer N that is the product of two primes, $N = pq$, reduces to the problem of finding the period of a function $f(x) = a^x \bmod N$, for any $a < N$ which is coprime to N, i.e., has no common factors with N other than 1.

Shor's algorithm: case N = 15

- Consider case $N = 15, a = 7$.
- The function $f(x) = a^x \bmod 15$, where $x \in [0, 63]$ $(f : Z_{64} \to Z_{15})$, is:

$$
\begin{aligned}
7^0 \bmod 15 &= 1 \\
7^1 \bmod 15 &= 7 \\
7^2 \bmod 15 &= 4 \\
7^3 \bmod 15 &= 13 \\
7^4 \bmod 15 &= 1 \\
&\vdots \\
7^{63} \bmod 15 &= 13
\end{aligned}
$$

and the period is evidently $r = 4$.

Shor's algorithm: case N = 15

- Consider case $N = 15, a = 7$.
- The function $f(x) = a^x \bmod 15$, where $x \in [0, 63]$ $(f : Z_{64} \to Z_{15})$, is:

$$
\begin{aligned}
7^0 \bmod 15 &= 1 \\
7^1 \bmod 15 &= 7 \\
7^2 \bmod 15 &= 4 \\
7^3 \bmod 15 &= 13 \\
7^4 \bmod 15 &= 1 \\
&\vdots \\
7^{63} \bmod 15 &= 13
\end{aligned}
$$

and the period is evidently $r = 4$.

Shor's algorithm: case $N = 15$

The factors 3 and 5 of 15 are derived as the greatest common factors of $a^{r/2} - 1 = 48$ and 15 and $a^{r/2} + 1 = 50$ and 15, respectively.

Shor's algorithm: case $N = 15$

After the application of the unitary transformation $U_f = a^x \bmod N$, the state of the two registers is:

$$\frac{1}{8}(|0\rangle|1\rangle + |1\rangle|7\rangle + |2\rangle|4\rangle + |3\rangle|13\rangle$$
$$+ |4\rangle|1\rangle + |5\rangle|7\rangle + |6\rangle|4\rangle + |7\rangle|13\rangle$$
$$\vdots$$
$$+ |60\rangle|1\rangle + |61\rangle|7\rangle + |62\rangle|4\rangle + |63\rangle|13\rangle)$$

Shor's algorithm: case N = 15

- Consider case $N = 15, a = 7$.
- The function $f(x) = a^x \bmod 15$, where $x \in [0, 63]$ $(f : Z_{64} \to Z_{15})$, is:

$$
\begin{aligned}
7^0 \bmod 15 &= 1 \\
7^1 \bmod 15 &= 7 \\
7^2 \bmod 15 &= 4 \\
7^3 \bmod 15 &= 13 \\
7^4 \bmod 15 &= 1 \\
&\vdots \\
7^{63} \bmod 15 &= 13
\end{aligned}
$$

and the period is evidently $r = 4$.

Shor's algorithm: case $N = 15$

After the application of the unitary transformation $U_f = a^x \bmod N$, the state of the two registers is:

$$\frac{1}{8}(|0\rangle|1\rangle + |1\rangle|7\rangle + |2\rangle|4\rangle + |3\rangle|13\rangle$$
$$+ |4\rangle|1\rangle + |5\rangle|7\rangle + |6\rangle|4\rangle + |7\rangle|13\rangle$$
$$\vdots$$
$$+ |60\rangle|1\rangle + |61\rangle|7\rangle + |62\rangle|4\rangle + |63\rangle|13\rangle)$$

Shor's algorithm: case N = 15

This state can be expressed as:

$$\begin{aligned}
&\tfrac{1}{8}(|0\rangle + |4\rangle + |8\rangle + \ldots + |60\rangle)|1\rangle \\
&+\tfrac{1}{8}(|1\rangle + |5\rangle + |9\rangle + \ldots + |61\rangle)|7\rangle \\
&+\tfrac{1}{8}(|2\rangle + |6\rangle + |10\rangle + \ldots + |62\rangle)|4\rangle \\
&+\tfrac{1}{8}(|3\rangle + |7\rangle + |11\rangle + \ldots + |63\rangle)|13\rangle)
\end{aligned}$$

Shor's algorithm: case N = 15

If we measure the output register, we obtain (equiprobably) one of four states for the input register, depending on the outcome of the measurement: 1, 7, 4, or 13:

$$\frac{1}{4}(|0\rangle + |4\rangle + |8\rangle + \ldots + |60\rangle)$$
$$\frac{1}{4}(|1\rangle + |5\rangle + |9\rangle + \ldots + |61\rangle)$$
$$\frac{1}{4}(|2\rangle + |6\rangle + |10\rangle + \ldots + |62\rangle)$$
$$\frac{1}{4}(|3\rangle + |7\rangle + |11\rangle + \ldots + |63\rangle)$$

Shor's algorithm: case N = 15

This state can be expressed as:

$$\frac{1}{8}(|0\rangle + |4\rangle + |8\rangle + \ldots + |60\rangle)|1\rangle$$
$$+\frac{1}{8}(|1\rangle + |5\rangle + |9\rangle + \ldots + |61\rangle)|7\rangle$$
$$+\frac{1}{8}(|2\rangle + |6\rangle + |10\rangle + \ldots + |62\rangle)|4\rangle$$
$$+\frac{1}{8}(|3\rangle + |7\rangle + |11\rangle + \ldots + |63\rangle)|13\rangle)$$

Shor's algorithm: case N = 15

If we measure the output register, we obtain (equiprobably) one of four states for the input register, depending on the outcome of the measurement: 1, 7, 4, or 13:

$$\frac{1}{4}(|0\rangle + |4\rangle + |8\rangle + \ldots + |60\rangle)$$
$$\frac{1}{4}(|1\rangle + |5\rangle + |9\rangle + \ldots + |61\rangle)$$
$$\frac{1}{4}(|2\rangle + |6\rangle + |10\rangle + \ldots + |62\rangle)$$
$$\frac{1}{4}(|3\rangle + |7\rangle + |11\rangle + \ldots + |63\rangle)$$

Shor's algorithm: case N = 15

- Apply a discrete quantum Fourier transform for the integers mod 64 (analogous to the Hadamard transform, which is a Fourier transform for the integers mod 2):

$$\frac{1}{\sqrt{\frac{s}{r}}}\sum_{j=0}^{\frac{s}{r}-1}|x_i + jr\rangle \xrightarrow{U_{DFT_s}} \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{2\pi i\frac{x_i k}{r}}|ks/r\rangle$$

(here $s = 64$, $r = 4$) which yields:

$$\begin{aligned} x_1 = 0: &\ \tfrac{1}{2}(|0\rangle + |16\rangle + |32\rangle + |48\rangle) \\ x_7 = 1: &\ \tfrac{1}{2}(|0\rangle + i|16\rangle - |32\rangle - i|48\rangle) \\ x_4 = 2: &\ \tfrac{1}{2}(|0\rangle - |16\rangle + |32\rangle - |48\rangle) \\ x_{13} = 3: &\ \tfrac{1}{2}(|0\rangle - i|16\rangle - |32\rangle + i|48\rangle) \end{aligned}$$

- So for the period $r = 4$, the state of the input register ends up in the 4-dimensional subspace spanned by the orthogonal vectors $|0\rangle, |16\rangle, |32\rangle, |48\rangle$.

Shor's algorithm: case N = 15

If we measure the output register, we obtain (equiprobably) one of four states for the input register, depending on the outcome of the measurement: 1, 7, 4, or 13:

$$\frac{1}{4}(|0\rangle + |4\rangle + |8\rangle + \ldots + |60\rangle)$$
$$\frac{1}{4}(|1\rangle + |5\rangle + |9\rangle + \ldots + |61\rangle)$$
$$\frac{1}{4}(|2\rangle + |6\rangle + |10\rangle + \ldots + |62\rangle)$$
$$\frac{1}{4}(|3\rangle + |7\rangle + |11\rangle + \ldots + |63\rangle)$$

Shor's algorithm: case N = 15

- Apply a discrete quantum Fourier transform for the integers mod 64 (analogous to the Hadamard transform, which is a Fourier transform for the integers mod 2):

$$\frac{1}{\sqrt{\frac{s}{r}}}\sum_{j=0}^{\frac{s}{r}-1}|x_i + jr\rangle \xrightarrow{U_{DFT_s}} \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{2\pi i \frac{x_i k}{r}}|ks/r\rangle$$

(here $s = 64$, $r = 4$) which yields:

$$\begin{aligned} x_1 = 0 &: \ \tfrac{1}{2}(|0\rangle + |16\rangle + |32\rangle + |48\rangle) \\ x_7 = 1 &: \ \tfrac{1}{2}(|0\rangle + i|16\rangle - |32\rangle - i|48\rangle) \\ x_4 = 2 &: \ \tfrac{1}{2}(|0\rangle - |16\rangle + |32\rangle - |48\rangle) \\ x_{13} = 3 &: \ \tfrac{1}{2}(|0\rangle - i|16\rangle - |32\rangle + i|48\rangle) \end{aligned}$$

- So for the period $r = 4$, the state of the input register ends up in the 4-dimensional subspace spanned by the orthogonal vectors $|0\rangle, |16\rangle, |32\rangle, |48\rangle$.

Shor's algorithm: case N = 15

If we measure the output register, we obtain (equiprobably) one of four states for the input register, depending on the outcome of the measurement: 1, 7, 4, or 13:

$$\frac{1}{4}(|0\rangle + |4\rangle + |8\rangle + \ldots + |60\rangle)$$
$$\frac{1}{4}(|1\rangle + |5\rangle + |9\rangle + \ldots + |61\rangle)$$
$$\frac{1}{4}(|2\rangle + |6\rangle + |10\rangle + \ldots + |62\rangle)$$
$$\frac{1}{4}(|3\rangle + |7\rangle + |11\rangle + \ldots + |63\rangle)$$

Shor's algorithm: case N = 15

- Apply a discrete quantum Fourier transform for the integers mod 64 (analogous to the Hadamard transform, which is a Fourier transform for the integers mod 2):

$$\frac{1}{\sqrt{\frac{s}{r}}}\sum_{j=0}^{\frac{s}{r}-1}|x_i + jr\rangle \xrightarrow{U_{DFT_s}} \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{2\pi i \frac{x_i k}{r}}|ks/r\rangle$$

(here $s = 64$, $r = 4$) which yields:

$$\begin{aligned} x_1 = 0 &: \ \tfrac{1}{2}(|0\rangle + |16\rangle + |32\rangle + |48\rangle) \\ x_7 = 1 &: \ \tfrac{1}{2}(|0\rangle + i|16\rangle - |32\rangle - i|48\rangle) \\ x_4 = 2 &: \ \tfrac{1}{2}(|0\rangle - |16\rangle + |32\rangle - |48\rangle) \\ x_{13} = 3 &: \ \tfrac{1}{2}(|0\rangle - i|16\rangle - |32\rangle + i|48\rangle) \end{aligned}$$

- So for the period $r = 4$, the state of the input register ends up in the 4-dimensional subspace spanned by the orthogonal vectors $|0\rangle, |16\rangle, |32\rangle, |48\rangle$.

Shor's algorithm: case N = 15

If we measure the output register, we obtain (equiprobably) one of four states for the input register, depending on the outcome of the measurement: 1, 7, 4, or 13:

$$\frac{1}{4}(|0\rangle + |4\rangle + |8\rangle + \ldots + |60\rangle)$$
$$\frac{1}{4}(|1\rangle + |5\rangle + |9\rangle + \ldots + |61\rangle)$$
$$\frac{1}{4}(|2\rangle + |6\rangle + |10\rangle + \ldots + |62\rangle)$$
$$\frac{1}{4}(|3\rangle + |7\rangle + |11\rangle + \ldots + |63\rangle)$$

Shor's algorithm: case N = 15

- Apply a discrete quantum Fourier transform for the integers mod 64 (analogous to the Hadamard transform, which is a Fourier transform for the integers mod 2):

$$\frac{1}{\sqrt{\frac{s}{r}}}\sum_{j=0}^{\frac{s}{r}-1}|x_i+jr\rangle \xrightarrow{U_{DFT_s}} \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{2\pi i\frac{x_i k}{r}}|ks/r\rangle$$

(here $s = 64$, $r = 4$) which yields:

$$\begin{aligned}
x_1 = 0 &: \ \tfrac{1}{2}(|0\rangle + |16\rangle + |32\rangle + |48\rangle) \\
x_7 = 1 &: \ \tfrac{1}{2}(|0\rangle + i|16\rangle - |32\rangle - i|48\rangle) \\
x_4 = 2 &: \ \tfrac{1}{2}(|0\rangle - |16\rangle + |32\rangle - |48\rangle) \\
x_{13} = 3 &: \ \tfrac{1}{2}(|0\rangle - i|16\rangle - |32\rangle + i|48\rangle)
\end{aligned}$$

- So for the period $r = 4$, the state of the input register ends up in the 4-dimensional subspace spanned by the orthogonal vectors $|0\rangle, |16\rangle, |32\rangle, |48\rangle$.

Shor's algorithm: case N = 15

If we measure the output register, we obtain (equiprobably) one of four states for the input register, depending on the outcome of the measurement: 1, 7, 4, or 13:

$$\frac{1}{4}(|0\rangle + |4\rangle + |8\rangle + \ldots + |60\rangle)$$
$$\frac{1}{4}(|1\rangle + |5\rangle + |9\rangle + \ldots + |61\rangle)$$
$$\frac{1}{4}(|2\rangle + |6\rangle + |10\rangle + \ldots + |62\rangle)$$
$$\frac{1}{4}(|3\rangle + |7\rangle + |11\rangle + \ldots + |63\rangle)$$

Shor's algorithm: case N = 15

- Apply a discrete quantum Fourier transform for the integers mod 64 (analogous to the Hadamard transform, which is a Fourier transform for the integers mod 2):

$$\frac{1}{\sqrt{\frac{s}{r}}}\sum_{j=0}^{\frac{s}{r}-1}|x_i + jr\rangle \xrightarrow{U_{DFT_s}} \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{2\pi i \frac{x_i k}{r}}|ks/r\rangle$$

(here $s = 64$, $r = 4$) which yields:

$$\begin{aligned} x_1 = 0 &: \ \tfrac{1}{2}(|0\rangle + |16\rangle + |32\rangle + |48\rangle) \\ x_7 = 1 &: \ \tfrac{1}{2}(|0\rangle + i|16\rangle - |32\rangle - i|48\rangle) \\ x_4 = 2 &: \ \tfrac{1}{2}(|0\rangle - |16\rangle + |32\rangle - |48\rangle) \\ x_{13} = 3 &: \ \tfrac{1}{2}(|0\rangle - i|16\rangle - |32\rangle + i|48\rangle) \end{aligned}$$

- So for the period $r = 4$, the state of the input register ends up in the 4-dimensional subspace spanned by the orthogonal vectors $|0\rangle, |16\rangle, |32\rangle, |48\rangle$.

Shor's algorithm: case N = 15

If we measure the output register, we obtain (equiprobably) one of four states for the input register, depending on the outcome of the measurement: 1, 7, 4, or 13:

$$\frac{1}{4}(|0\rangle + |4\rangle + |8\rangle + \ldots + |60\rangle)$$
$$\frac{1}{4}(|1\rangle + |5\rangle + |9\rangle + \ldots + |61\rangle)$$
$$\frac{1}{4}(|2\rangle + |6\rangle + |10\rangle + \ldots + |62\rangle)$$
$$\frac{1}{4}(|3\rangle + |7\rangle + |11\rangle + \ldots + |63\rangle)$$

Shor's algorithm: case N = 15

- Apply a discrete quantum Fourier transform for the integers mod 64 (analogous to the Hadamard transform, which is a Fourier transform for the integers mod 2):

$$\frac{1}{\sqrt{\frac{s}{r}}}\sum_{j=0}^{\frac{s}{r}-1}|x_i + jr\rangle \xrightarrow{U_{DFT_s}} \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{2\pi i\frac{x_i k}{r}}|ks/r\rangle$$

(here $s = 64$, $r = 4$) which yields:

$$\begin{aligned} x_1 = 0:&\ \tfrac{1}{2}(|0\rangle + |16\rangle + |32\rangle + |48\rangle)\\ x_7 = 1:&\ \tfrac{1}{2}(|0\rangle + i|16\rangle - |32\rangle - i|48\rangle)\\ x_4 = 2:&\ \tfrac{1}{2}(|0\rangle - |16\rangle + |32\rangle - |48\rangle)\\ x_{13} = 3:&\ \tfrac{1}{2}(|0\rangle - i|16\rangle - |32\rangle + i|48\rangle)\end{aligned}$$

- So for the period $r = 4$, the state of the input register ends up in the 4-dimensional subspace spanned by the orthogonal vectors $|0\rangle, |16\rangle, |32\rangle, |48\rangle$.

Shor's algorithm: case $N = 15$

- Consider case $N = 15, a = 7$.
- The function $f(x) = a^x \bmod 15$, where $x \in [0, 63]$ $(f : Z_{64} \to Z_{15})$, is:

$$
\begin{aligned}
7^0 \bmod 15 &= 1 \\
7^1 \bmod 15 &= 7 \\
7^2 \bmod 15 &= 4 \\
7^3 \bmod 15 &= 13 \\
7^4 \bmod 15 &= 1 \\
&\vdots \\
7^{63} \bmod 15 &= 13
\end{aligned}
$$

and the period is evidently $r = 4$.

Shor's factorization algorithm

- The problem of factorizing a composite integer N that is the product of two primes, $N = pq$, reduces to the problem of finding the period of a function $f(x) = a^x \bmod N$, for any $a < N$ which is coprime to N, i.e., has no common factors with N other than 1.

Simon's algorithm: n = 3 case

- Applying a Hadamard transformation amounts to dropping the primes.
- So if the period is $r = 001$, the state of the input register ends up in the 4-dimensional subspace spanned by the vectors: $|000\rangle, |010\rangle, |100\rangle, |110\rangle$.

Simon's algorithm: n = 3 case

- Period $r = 001$: the state of the two registers after the unitary transformation U_f is:

$$(|000\rangle + |001\rangle)|f(000)\rangle + (|010\rangle + |011\rangle)|f(010)\rangle$$
$$+(|100\rangle + |101\rangle)|f(100)\rangle + (|110\rangle + |111\rangle)|f(110)\rangle$$

- Measure the output register $\Longrightarrow$ the input register is left in one of four states, depending on the outcome of the measurement:

$$\begin{aligned}
|000\rangle + |001\rangle &= |0'0'0'\rangle + |0'1'0'\rangle + |1'0'0'\rangle + |1'1'0'\rangle \\
|010\rangle + |011\rangle &= |0'0'0'\rangle - |0'1'0'\rangle + |1'0'0'\rangle - |1'1'0'\rangle \\
|100\rangle + |101\rangle &= |0'0'0'\rangle + |0'1'0'\rangle - |1'0'0'\rangle - |1'1'0'\rangle \\
|110\rangle + |111\rangle &= |0'0'0'\rangle - |0'1'0'\rangle - |1'0'0'\rangle + |1'1'0'\rangle
\end{aligned}$$

Simon's algorithm: n = 3 case

- Applying a Hadamard transformation amounts to dropping the primes.
- So if the period is $r = 001$, the state of the input register ends up in the 4-dimensional subspace spanned by the vectors: $|000\rangle, |010\rangle, |100\rangle, |110\rangle$.

Shor's factorization algorithm

- The problem of factorizing a composite integer N that is the product of two primes, $N = pq$, reduces to the problem of finding the period of a function $f(x) = a^x \bmod N$, for any $a < N$ which is coprime to N, i.e., has no common factors with N other than 1.

Shor's algorithm: case N = 15

If we measure the output register, we obtain (equiprobably) one of four states for the input register, depending on the outcome of the measurement: 1, 7, 4, or 13:

$$\frac{1}{4}(|0\rangle + |4\rangle + |8\rangle + \ldots + |60\rangle)$$
$$\frac{1}{4}(|1\rangle + |5\rangle + |9\rangle + \ldots + |61\rangle)$$
$$\frac{1}{4}(|2\rangle + |6\rangle + |10\rangle + \ldots + |62\rangle)$$
$$\frac{1}{4}(|3\rangle + |7\rangle + |11\rangle + \ldots + |63\rangle)$$

Shor's algorithm: case N = 15

- Apply a discrete quantum Fourier transform for the integers mod 64 (analogous to the Hadamard transform, which is a Fourier transform for the integers mod 2):

$$\frac{1}{\sqrt{\frac{s}{r}}}\sum_{j=0}^{\frac{s}{r}-1}|x_i + jr\rangle \xrightarrow{U_{DFT_s}} \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{2\pi i \frac{x_i k}{r}}|ks/r\rangle$$

(here $s = 64$, $r = 4$) which yields:

$$\begin{aligned} x_1 = 0 &: \ \tfrac{1}{2}(|0\rangle + |16\rangle + |32\rangle + |48\rangle) \\ x_7 = 1 &: \ \tfrac{1}{2}(|0\rangle + i|16\rangle - |32\rangle - i|48\rangle) \\ x_4 = 2 &: \ \tfrac{1}{2}(|0\rangle - |16\rangle + |32\rangle - |48\rangle) \\ x_{13} = 3 &: \ \tfrac{1}{2}(|0\rangle - i|16\rangle - |32\rangle + i|48\rangle) \end{aligned}$$

- So for the period $r = 4$, the state of the input register ends up in the 4-dimensional subspace spanned by the orthogonal vectors $|0\rangle, |16\rangle, |32\rangle, |48\rangle$.

Shor's algorithm: case $N = 15$

- Consider all possible even periods r for which $f(x) = a^x \bmod 15$, where a is coprime to 15.
- The other possible values of a are 2, 4, 8, 11, 13, 14 and the corresponding periods turn out to be 4, 2, 4, 2, 4, 2. So we need only consider $r = 2$.
- Different values of a with the same period affect only the labels of the output register (e.g., for $a = 2$, the labels are $|1\rangle, |2\rangle, |4\rangle, |8\rangle$ instead of $|1\rangle, |7\rangle, |4\rangle, |13\rangle$). So different a values for the same period are irrelevant to the quantum algorithm.

Shor's algorithm: case $N = 15$

After the application of the unitary transformation $U_f = a^x \bmod N$, the state of the two registers is:

$$\frac{1}{8}(|0\rangle|1\rangle + |1\rangle|7\rangle + |2\rangle|4\rangle + |3\rangle|13\rangle$$
$$+ |4\rangle|1\rangle + |5\rangle|7\rangle + |6\rangle|4\rangle + |7\rangle|13\rangle$$
$$\vdots$$
$$+ |60\rangle|1\rangle + |61\rangle|7\rangle + |62\rangle|4\rangle + |63\rangle|13\rangle)$$

Shor's algorithm: case N = 15

This state can be expressed as:

$$\begin{aligned}
&\tfrac{1}{8}(|0\rangle + |4\rangle + |8\rangle + \ldots + |60\rangle)|1\rangle \\
&+\tfrac{1}{8}(|1\rangle + |5\rangle + |9\rangle + \ldots + |61\rangle)|7\rangle \\
&+\tfrac{1}{8}(|2\rangle + |6\rangle + |10\rangle + \ldots + |62\rangle)|4\rangle \\
&+\tfrac{1}{8}(|3\rangle + |7\rangle + |11\rangle + \ldots + |63\rangle)|13\rangle)
\end{aligned}$$

Shor's algorithm: case $N = 15$

After the application of the unitary transformation $U_f = a^x \bmod N$, the state of the two registers is:

$$\frac{1}{8}(|0\rangle|1\rangle + |1\rangle|7\rangle + |2\rangle|4\rangle + |3\rangle|13\rangle$$
$$+ |4\rangle|1\rangle + |5\rangle|7\rangle + |6\rangle|4\rangle + |7\rangle|13\rangle$$
$$\vdots$$
$$+ |60\rangle|1\rangle + |61\rangle|7\rangle + |62\rangle|4\rangle + |63\rangle|13\rangle)$$

Shor's algorithm: case $N = 15$

The factors 3 and 5 of 15 are derived as the greatest common factors of $a^{r/2} - 1 = 48$ and 15 and $a^{r/2} + 1 = 50$ and 15, respectively.

Shor's algorithm: case N = 15

- Consider case $N = 15, a = 7$.
- The function $f(x) = a^x \bmod 15$, where $x \in [0, 63]$ $(f : Z_{64} \to Z_{15})$, is:

$$\begin{aligned}
7^0 \bmod 15 &= 1 \\
7^1 \bmod 15 &= 7 \\
7^2 \bmod 15 &= 4 \\
7^3 \bmod 15 &= 13 \\
7^4 \bmod 15 &= 1 \\
&\vdots \\
7^{63} \bmod 15 &= 13
\end{aligned}$$

and the period is evidently $r = 4$.

Shor's algorithm: case $\mathrm{N} = 15$

After the application of the unitary transformation $\mathrm{U_f} = \mathrm{a^x} \bmod \mathrm{N}$, the state of the two registers is:

$$\frac{1}{8}(|0\rangle|1\rangle + |1\rangle|7\rangle + |2\rangle|4\rangle + |3\rangle|13\rangle$$
$$+ |4\rangle|1\rangle + |5\rangle|7\rangle + |6\rangle|4\rangle + |7\rangle|13\rangle$$
$$\vdots$$
$$+ |60\rangle|1\rangle + |61\rangle|7\rangle + |62\rangle|4\rangle + |63\rangle|13\rangle)$$

Shor's algorithm: case N = 15

This state can be expressed as:

$$\frac{1}{8}(|0\rangle + |4\rangle + |8\rangle + \ldots + |60\rangle)|1\rangle$$
$$+\frac{1}{8}(|1\rangle + |5\rangle + |9\rangle + \ldots + |61\rangle)|7\rangle$$
$$+\frac{1}{8}(|2\rangle + |6\rangle + |10\rangle + \ldots + |62\rangle)|4\rangle$$
$$+\frac{1}{8}(|3\rangle + |7\rangle + |11\rangle + \ldots + |63\rangle)|13\rangle)$$

Shor's algorithm: case N = 15

- Apply a discrete quantum Fourier transform for the integers mod 64 (analogous to the Hadamard transform, which is a Fourier transform for the integers mod 2):

$$\frac{1}{\sqrt{\frac{s}{r}}}\sum_{j=0}^{\frac{s}{r}-1}|x_i+jr\rangle \xrightarrow{U_{DFT_s}} \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}e^{2\pi i\frac{x_i k}{r}}|ks/r\rangle$$

(here $s=64$, $r=4$) which yields:

$$\begin{aligned} x_1 = 0: &\ \tfrac{1}{2}(|0\rangle+|16\rangle+|32\rangle+|48\rangle) \\ x_7 = 1: &\ \tfrac{1}{2}(|0\rangle+i|16\rangle-|32\rangle-i|48\rangle) \\ x_4 = 2: &\ \tfrac{1}{2}(|0\rangle-|16\rangle+|32\rangle-|48\rangle) \\ x_{13} = 3: &\ \tfrac{1}{2}(|0\rangle-i|16\rangle-|32\rangle+i|48\rangle) \end{aligned}$$

- So for the period $r=4$, the state of the input register ends up in the 4-dimensional subspace spanned by the orthogonal vectors $|0\rangle, |16\rangle, |32\rangle, |48\rangle$.

Shor's algorithm: case $N = 15$

- Consider all possible even periods r for which $f(x) = a^x \bmod 15$, where a is coprime to 15.
- The other possible values of a are 2, 4, 8, 11, 13, 14 and the corresponding periods turn out to be 4, 2, 4, 2, 4, 2. So we need only consider $r = 2$.
- Different values of a with the same period affect only the labels of the output register (e.g., for $a = 2$, the labels are $|1\rangle, |2\rangle, |4\rangle, |8\rangle$ instead of $|1\rangle, |7\rangle, |4\rangle, |13\rangle$). So different a values for the same period are irrelevant to the quantum algorithm.

Shor's algorithm: case N = 15

- For r = 2, if we measure the output register, we will obtain (equiprobably) one of two states for the input register, depending on the outcome of the measurement (say, a or b):

$$|0\rangle + |2\rangle + |4\rangle + \ldots + |62\rangle$$
$$|1\rangle + |3\rangle + |5\rangle + \ldots + |63\rangle$$

- After the discrete Fourier transform, these states are transformed to:

$$x_a = 0 : \ |0\rangle + |32\rangle$$
$$x_b = 1 : \ |0\rangle - |32\rangle$$

Shor's algorithm: case N = 15

- Apply a discrete quantum Fourier transform for the integers mod 64 (analogous to the Hadamard transform, which is a Fourier transform for the integers mod 2):

$$\frac{1}{\sqrt{\frac{s}{r}}}\sum_{j=0}^{\frac{s}{r}-1}|x_i + jr\rangle \xrightarrow{U_{DFT_s}} \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{2\pi i \frac{x_i k}{r}}|ks/r\rangle$$

(here $s = 64$, $r = 4$) which yields:

$$\begin{aligned}
x_1 = 0 &: \tfrac{1}{2}(|0\rangle + |16\rangle + |32\rangle + |48\rangle) \\
x_7 = 1 &: \tfrac{1}{2}(|0\rangle + i|16\rangle - |32\rangle - i|48\rangle) \\
x_4 = 2 &: \tfrac{1}{2}(|0\rangle - |16\rangle + |32\rangle - |48\rangle) \\
x_{13} = 3 &: \tfrac{1}{2}(|0\rangle - i|16\rangle - |32\rangle + i|48\rangle)
\end{aligned}$$

- So for the period $r = 4$, the state of the input register ends up in the 4-dimensional subspace spanned by the orthogonal vectors $|0\rangle, |16\rangle, |32\rangle, |48\rangle$.

Shor's algorithm: case N = 15

- Consider all possible even periods r for which $f(x) = a^x \bmod 15$, where a is coprime to 15.
- The other possible values of a are 2, 4, 8, 11, 13, 14 and the corresponding periods turn out to be 4, 2, 4, 2, 4, 2. So we need only consider $r = 2$.
- Different values of a with the same period affect only the labels of the output register (e.g., for $a = 2$, the labels are $|1\rangle, |2\rangle, |4\rangle, |8\rangle$ instead of $|1\rangle, |7\rangle, |4\rangle, |13\rangle$). So different a values for the same period are irrelevant to the quantum algorithm.

Shor's algorithm: case N = 15

- In this case, the 2-dimensional subspace $\mathcal{V}_{\mathrm{r}=2}$ spanned by $|0\rangle, |32\rangle$ for $\mathrm{r} = 2$ is included in the 4-dimensional subspace $\mathcal{V}_{\mathrm{r}=4}$ for $\mathrm{r} = 4$ (spanned by $|0\rangle, |16\rangle, |32\rangle, \mathrm{ket}48)$.
- A measurement can distinguish $\mathrm{r} = 4$ from $\mathrm{r} = 2$ reliably, i.e., whether the final state of the input register is in $\mathcal{V}_{\mathrm{r}=4}$ or $\mathcal{V}_{\mathrm{r}=2}$, only if the final state is in $\mathcal{V}_{\mathrm{r}=4} - \mathcal{V}_{\mathrm{r}=2}$, the part of $\mathcal{V}_{\mathrm{r}=4}$ orthogonal to $\mathcal{V}_{\mathrm{r}=2}$.
- What happens if the final state ends up in $\mathcal{V}_{\mathrm{r}=2}$?

Shor's algorithm: case N = 15

- Shor's algorithm works as a randomized algorithm. It produces a candidate value for the period r and hence a candidate factor of N, which can be tested (in polynomial time) by division into N.
- A measurement of the input register in the computational basis yields an outcome $c = ks/r$. The value of k is chosen equiprobably by the measurement of the output register.
- The procedure is to repeat the algorithm until the outcome yields a value of k coprime to r, in which case canceling c/s to lowest terms yields k and r as k/r.

Shor's algorithm: case N = 15

- In this case, the 2-dimensional subspace $\mathcal{V}_{\mathrm{r}=2}$ spanned by $|0\rangle, |32\rangle$ for $\mathrm{r} = 2$ is included in the 4-dimensional subspace $\mathcal{V}_{\mathrm{r}=4}$ for $\mathrm{r} = 4$ (spanned by $|0\rangle, |16\rangle, |32\rangle, \mathrm{ket48}$).
- A measurement can distinguish $\mathrm{r} = 4$ from $\mathrm{r} = 2$ reliably, i.e., whether the final state of the input register is in $\mathcal{V}_{\mathrm{r}=4}$ or $\mathcal{V}_{\mathrm{r}=2}$, only if the final state is in $\mathcal{V}_{\mathrm{r}=4} - \mathcal{V}_{\mathrm{r}=2}$, the part of $\mathcal{V}_{\mathrm{r}=4}$ orthogonal to $\mathcal{V}_{\mathrm{r}=2}$.
- What happens if the final state ends up in $\mathcal{V}_{\mathrm{r}=2}$?

Shor's algorithm: case N = 15

- For r = 2, if we measure the output register, we will obtain (equiprobably) one of two states for the input register, depending on the outcome of the measurement (say, a or b):

$$|0\rangle + |2\rangle + |4\rangle + \ldots + |62\rangle$$
$$|1\rangle + |3\rangle + |5\rangle + \ldots + |63\rangle$$

- After the discrete Fourier transform, these states are transformed to:

$$\mathrm{x_a} = 0: \; |0\rangle + |32\rangle$$
$$\mathrm{x_b} = 1: \; |0\rangle - |32\rangle$$

Shor's algorithm: case N = 15

- In this case, the 2-dimensional subspace $\mathcal{V}_{\mathrm{r}=2}$ spanned by $|0\rangle, |32\rangle$ for $\mathrm{r} = 2$ is included in the 4-dimensional subspace $\mathcal{V}_{\mathrm{r}=4}$ for $\mathrm{r} = 4$ (spanned by $|0\rangle, |16\rangle, |32\rangle, \mathrm{ket}48$).
- A measurement can distinguish $\mathrm{r} = 4$ from $\mathrm{r} = 2$ reliably, i.e., whether the final state of the input register is in $\mathcal{V}_{\mathrm{r}=4}$ or $\mathcal{V}_{\mathrm{r}=2}$, only if the final state is in $\mathcal{V}_{\mathrm{r}=4} - \mathcal{V}_{\mathrm{r}=2}$, the part of $\mathcal{V}_{\mathrm{r}=4}$ orthogonal to $\mathcal{V}_{\mathrm{r}=2}$.
- What happens if the final state ends up in $\mathcal{V}_{\mathrm{r}=2}$?

Shor's algorithm: case N = 15

- Shor's algorithm works as a randomized algorithm. It produces a candidate value for the period r and hence a candidate factor of N, which can be tested (in polynomial time) by division into N.
- A measurement of the input register in the computational basis yields an outcome $c = ks/r$. The value of k is chosen equiprobably by the measurement of the output register.
- The procedure is to repeat the algorithm until the outcome yields a value of k coprime to r, in which case canceling c/s to lowest terms yields k and r as k/r.

Shor's algorithm: case N = 15

- For r = 2, if we measure the output register, we will obtain (equiprobably) one of two states for the input register, depending on the outcome of the measurement (say, a or b):

$$|0\rangle + |2\rangle + |4\rangle + \ldots + |62\rangle$$
$$|1\rangle + |3\rangle + |5\rangle + \ldots + |63\rangle$$

- After the discrete Fourier transform, these states are transformed to:

$$x_a = 0: \ |0\rangle + |32\rangle$$
$$x_b = 1: \ |0\rangle - |32\rangle$$

Shor's algorithm: case N = 15

- Apply a discrete quantum Fourier transform for the integers mod 64 (analogous to the Hadamard transform, which is a Fourier transform for the integers mod 2):

$$\frac{1}{\sqrt{\frac{s}{r}}}\sum_{j=0}^{\frac{s}{r}-1}|x_i + jr\rangle \xrightarrow{U_{DFT_s}} \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{2\pi i\frac{x_i k}{r}}|ks/r\rangle$$

(here $s = 64$, $r = 4$) which yields:

$$\begin{aligned} x_1 = 0 &: \tfrac{1}{2}(|0\rangle + |16\rangle + |32\rangle + |48\rangle) \\ x_7 = 1 &: \tfrac{1}{2}(|0\rangle + i|16\rangle - |32\rangle - i|48\rangle) \\ x_4 = 2 &: \tfrac{1}{2}(|0\rangle - |16\rangle + |32\rangle - |48\rangle) \\ x_{13} = 3 &: \tfrac{1}{2}(|0\rangle - i|16\rangle - |32\rangle + i|48\rangle) \end{aligned}$$

- So for the period $r = 4$, the state of the input register ends up in the 4-dimensional subspace spanned by the orthogonal vectors $|0\rangle, |16\rangle, |32\rangle, |48\rangle$.

Shor's algorithm: case N = 15

- Consider all possible even periods r for which $f(x) = a^x \bmod 15$, where a is coprime to 15.
- The other possible values of a are 2, 4, 8, 11, 13, 14 and the corresponding periods turn out to be 4, 2, 4, 2, 4, 2. So we need only consider $r = 2$.
- Different values of a with the same period affect only the labels of the output register (e.g., for $a = 2$, the labels are $|1\rangle, |2\rangle, |4\rangle, |8\rangle$ instead of $|1\rangle, |7\rangle, |4\rangle, |13\rangle$). So different a values for the same period are irrelevant to the quantum algorithm.

Shor's algorithm: case N = 15

- In this case, the 2-dimensional subspace $\mathcal{V}_{\mathrm{r}=2}$ spanned by $|0\rangle, |32\rangle$ for $\mathrm{r} = 2$ is included in the 4-dimensional subspace $\mathcal{V}_{\mathrm{r}=4}$ for $\mathrm{r} = 4$ (spanned by $|0\rangle, |16\rangle, |32\rangle, \mathrm{ket}48$).
- A measurement can distinguish $\mathrm{r} = 4$ from $\mathrm{r} = 2$ reliably, i.e., whether the final state of the input register is in $\mathcal{V}_{\mathrm{r}=4}$ or $\mathcal{V}_{\mathrm{r}=2}$, only if the final state is in $\mathcal{V}_{\mathrm{r}=4} - \mathcal{V}_{\mathrm{r}=2}$, the part of $\mathcal{V}_{\mathrm{r}=4}$ orthogonal to $\mathcal{V}_{\mathrm{r}=2}$.
- What happens if the final state ends up in $\mathcal{V}_{\mathrm{r}=2}$?

Shor's algorithm: case N = 15

- Putting it geometrically: the value $k = 1$ for $r = 2$ corresponds to the same state, $|32\rangle$, as the value $k = 2$ for $r = 4$.
- Once we obtain the candidate period $r = 2$ (by cancelling $c/s = 32/64$ to lowest terms), we calculate the factors of N as the greatest common factors of $a^{\frac{r}{2}} \pm 1$ and N and test these by division into N.
- If $a = 7$, these calculated factors will be incorrect. If $a = 2$, the factors calculated in this way will be correct.

Exploiting the non-Boolean logic

- We see how a global property of a function is computed without actually computing any function values.
- Does it always work this way?

Exploiting the non-Boolean logic

- We see how a global property of a function is computed without actually computing any function values.
- Does it always work this way?

Shor's algorithm: case $N = 15$

- Putting it geometrically: the value $k = 1$ for $r = 2$ corresponds to the same state, $|32\rangle$, as the value $k = 2$ for $r = 4$.
- Once we obtain the candidate period $r = 2$ (by cancelling $c/s = 32/64$ to lowest terms), we calculate the factors of N as the greatest common factors of $a^{\frac{r}{2}} \pm 1$ and N and test these by division into N.
- If $a = 7$, these calculated factors will be incorrect. If $a = 2$, the factors calculated in this way will be correct.

Shor's algorithm: case N = 15

- For r = 2, if we measure the output register, we will obtain (equiprobably) one of two states for the input register, depending on the outcome of the measurement (say, a or b):

$$|0\rangle + |2\rangle + |4\rangle + \ldots + |62\rangle$$
$$|1\rangle + |3\rangle + |5\rangle + \ldots + |63\rangle$$

- After the discrete Fourier transform, these states are transformed to:

$$\mathrm{x_a} = 0 : \;\; |0\rangle + |32\rangle$$
$$\mathrm{x_b} = 1 : \;\; |0\rangle - |32\rangle$$

Shor's algorithm: case N = 15

- Consider all possible even periods r for which $f(x) = a^x \bmod 15$, where a is coprime to 15.
- The other possible values of a are 2, 4, 8, 11, 13, 14 and the corresponding periods turn out to be 4, 2, 4, 2, 4, 2. So we need only consider $r = 2$.
- Different values of a with the same period affect only the labels of the output register (e.g., for $a = 2$, the labels are $|1\rangle, |2\rangle, |4\rangle, |8\rangle$ instead of $|1\rangle, |7\rangle, |4\rangle, |13\rangle$). So different a values for the same period are irrelevant to the quantum algorithm.

Shor's algorithm: case N = 15

- For r = 2, if we measure the output register, we will obtain (equiprobably) one of two states for the input register, depending on the outcome of the measurement (say, a or b):

$$|0\rangle + |2\rangle + |4\rangle + \ldots + |62\rangle$$
$$|1\rangle + |3\rangle + |5\rangle + \ldots + |63\rangle$$

- After the discrete Fourier transform, these states are transformed to:

$$x_a = 0: \ |0\rangle + |32\rangle$$
$$x_b = 1: \ |0\rangle - |32\rangle$$

Shor's algorithm: case N = 15

- In this case, the 2-dimensional subspace $\mathcal{V}_{\mathrm{r}=2}$ spanned by $|0\rangle, |32\rangle$ for $\mathrm{r} = 2$ is included in the 4-dimensional subspace $\mathcal{V}_{\mathrm{r}=4}$ for $\mathrm{r} = 4$ (spanned by $|0\rangle, |16\rangle, |32\rangle, \mathrm{ket}48)$.
- A measurement can distinguish $\mathrm{r} = 4$ from $\mathrm{r} = 2$ reliably, i.e., whether the final state of the input register is in $\mathcal{V}_{\mathrm{r}=4}$ or $\mathcal{V}_{\mathrm{r}=2}$, only if the final state is in $\mathcal{V}_{\mathrm{r}=4} - \mathcal{V}_{\mathrm{r}=2}$, the part of $\mathcal{V}_{\mathrm{r}=4}$ orthogonal to $\mathcal{V}_{\mathrm{r}=2}$.
- What happens if the final state ends up in $\mathcal{V}_{\mathrm{r}=2}$?

Shor's algorithm: case $N = 15$

- Shor's algorithm works as a randomized algorithm. It produces a candidate value for the period r and hence a candidate factor of N, which can be tested (in polynomial time) by division into N.
- A measurement of the input register in the computational basis yields an outcome $c = ks/r$. The value of k is chosen equiprobably by the measurement of the output register.
- The procedure is to repeat the algorithm until the outcome yields a value of k coprime to r, in which case canceling c/s to lowest terms yields k and r as k/r.

Shor's algorithm: case $N = 15$

- Putting it geometrically: the value $k = 1$ for $r = 2$ corresponds to the same state, $|32\rangle$, as the value $k = 2$ for $r = 4$.
- Once we obtain the candidate period $r = 2$ (by cancelling $c/s = 32/64$ to lowest terms), we calculate the factors of N as the greatest common factors of $a^{\frac{r}{2}} \pm 1$ and N and test these by division into N.
- If $a = 7$, these calculated factors will be incorrect. If $a = 2$, the factors calculated in this way will be correct.

Exploiting the non-Boolean logic

- These quantum algorithms work by exploiting the non-Boolean logic represented by the subspace structure of Hilbert space in a similar way. Essentially, a global property of a function (such as a period, or a disjunctive property) is encoded as a subspace in Hilbert space representing a quantum proposition, which can then be efficiently distinguished from alternative propositions, corresponding to alternative global properties, by a measurement (or sequence of measurements) that identifies the target proposition as the proposition represented by the subspace containing the final state produced by the algorithm.

Exploiting the non-Boolean logic

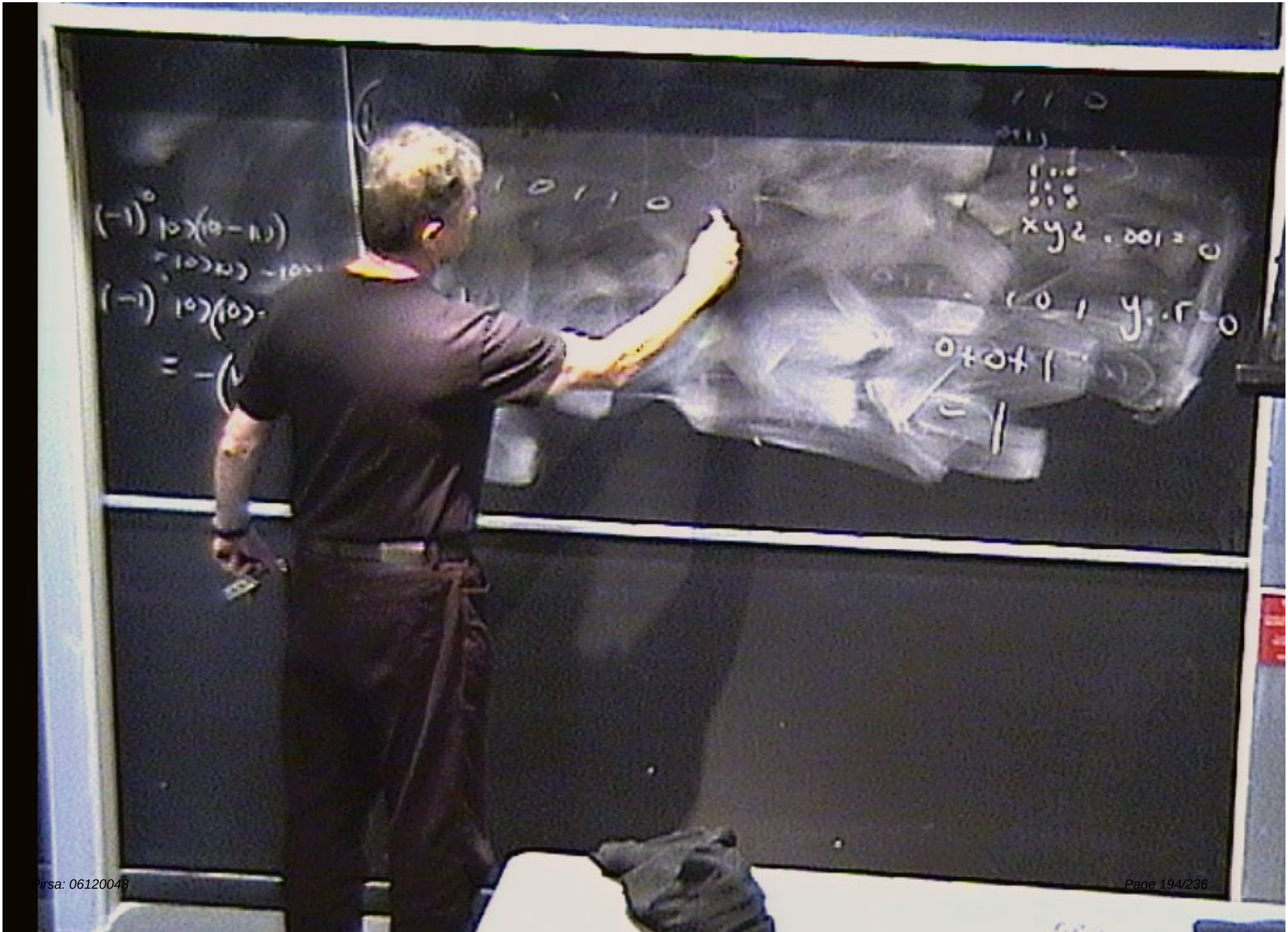
- We see how a global property of a function is computed without actually computing any function values.
- Does it always work this way?

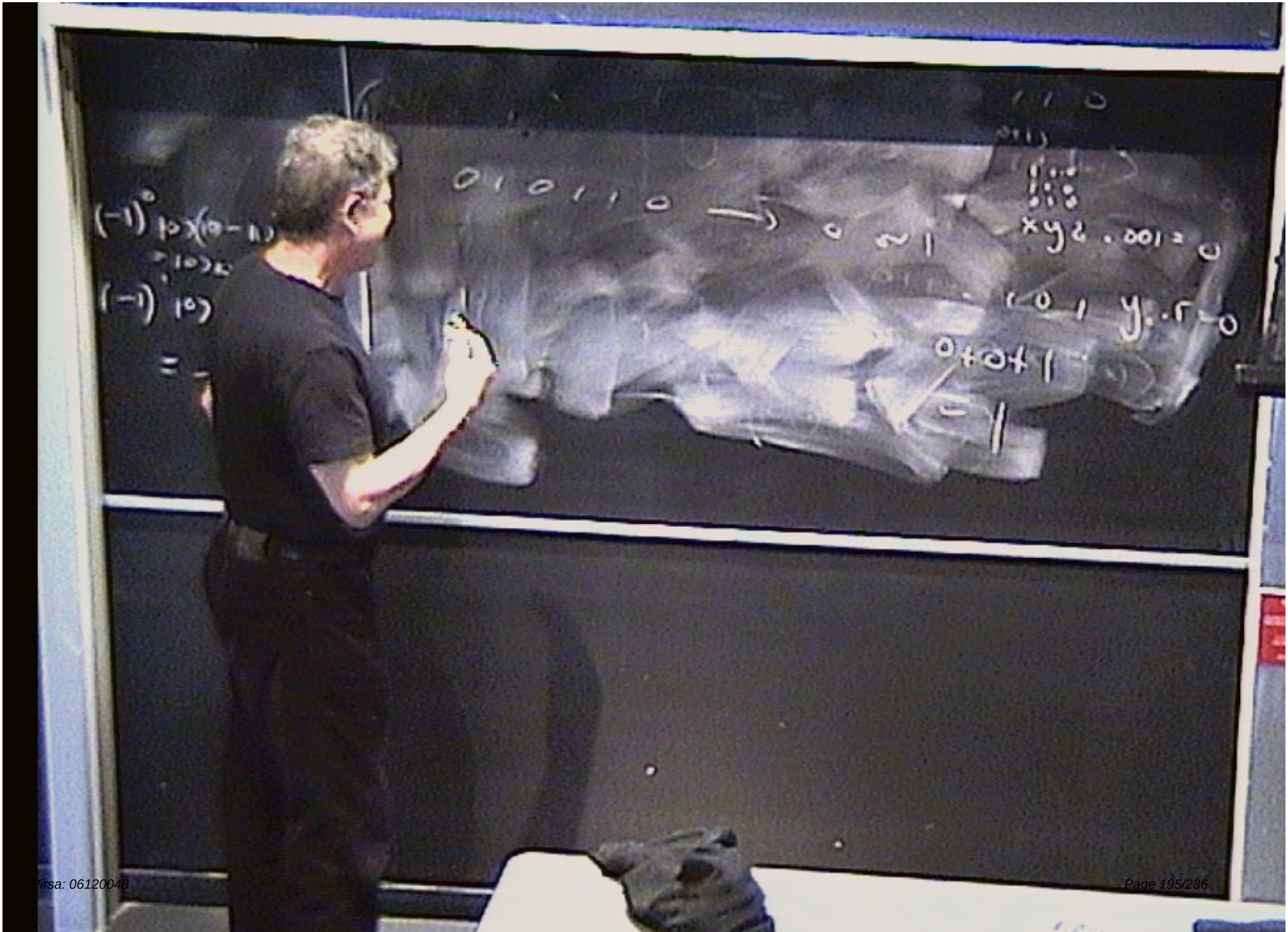
Parity problem

- The parity of a function $f : B^n \rightarrow B$ is defined by

$$\mathrm{par}(f) = \prod_{x \in B^n} (-1)^{f(x)}$$

- Classically, $N = 2^n$ function calls are required to determine the parity of f. Can a quantum algorithm do better?


xyz . 001 = 0
0+0+1
= 1


0 1 0 1 1 1 0
0 + 0 + 1
= 1

Parity problem

- $n = 2$ case: like Deutsch's problem, 1 constant state, 3 balanced states, 4 unbalanced states after U_f:

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

with three +'s and one −'s.

- The constant and balanced states have parity 1; the unbalanced states have parity -1.

Parity problem

- n = 2 case: like Deutsch's problem, 1 constant state, 3 balanced states, 4 unbalanced states after U_f:

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

  with three +'s and one −'s.

- The constant and balanced states have parity 1; the unbalanced states have parity -1.

Parity problem

After U_f, the state of the input register is:

$$\frac{1}{2}((-1)^{f(00)}|00\rangle + (-1)^{f(01)}|01\rangle + (-1)^{f(10)}|10\rangle + (-1)^{f(11)}|11\rangle)$$

Parity problem

To find the parity of f, apply VU_f twice, where V is the permutation:

$$
\begin{aligned}
V|00\rangle &\rightarrow |01\rangle \\
V|01\rangle &\rightarrow |00\rangle \\
V|10\rangle &\rightarrow |11\rangle \\
V|11\rangle &\rightarrow |10\rangle
\end{aligned}
$$

Parity problem

After $\mathrm{U_f}$, the state of the input register is:

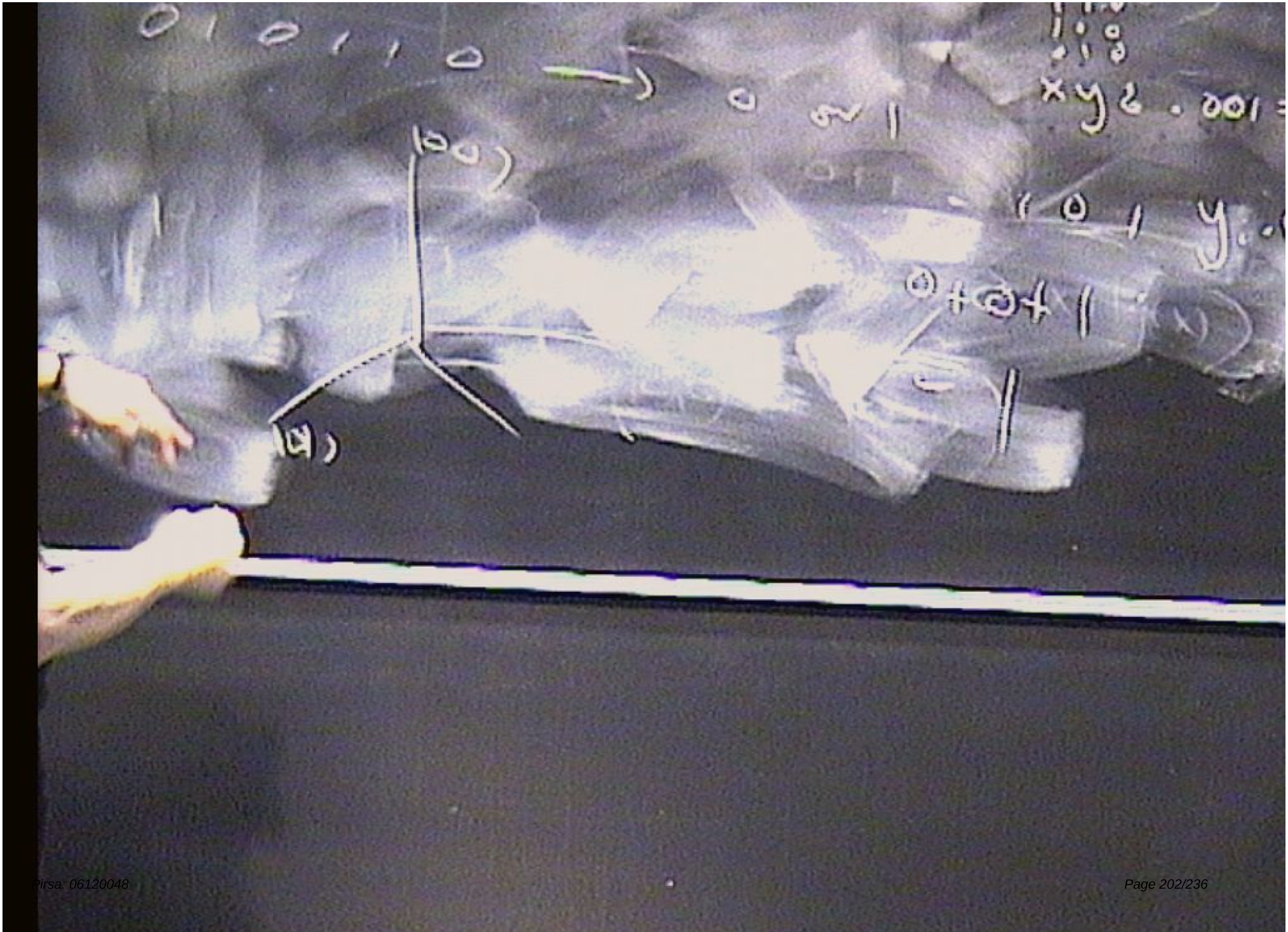
$$\frac{1}{2}((-1)^{\mathrm{f}(00)}|00\rangle + (-1)^{\mathrm{f}(01)}|01\rangle + (-1)^{\mathrm{f}(10)}|10\rangle + (-1)^{\mathrm{f}(11)}|11\rangle)$$

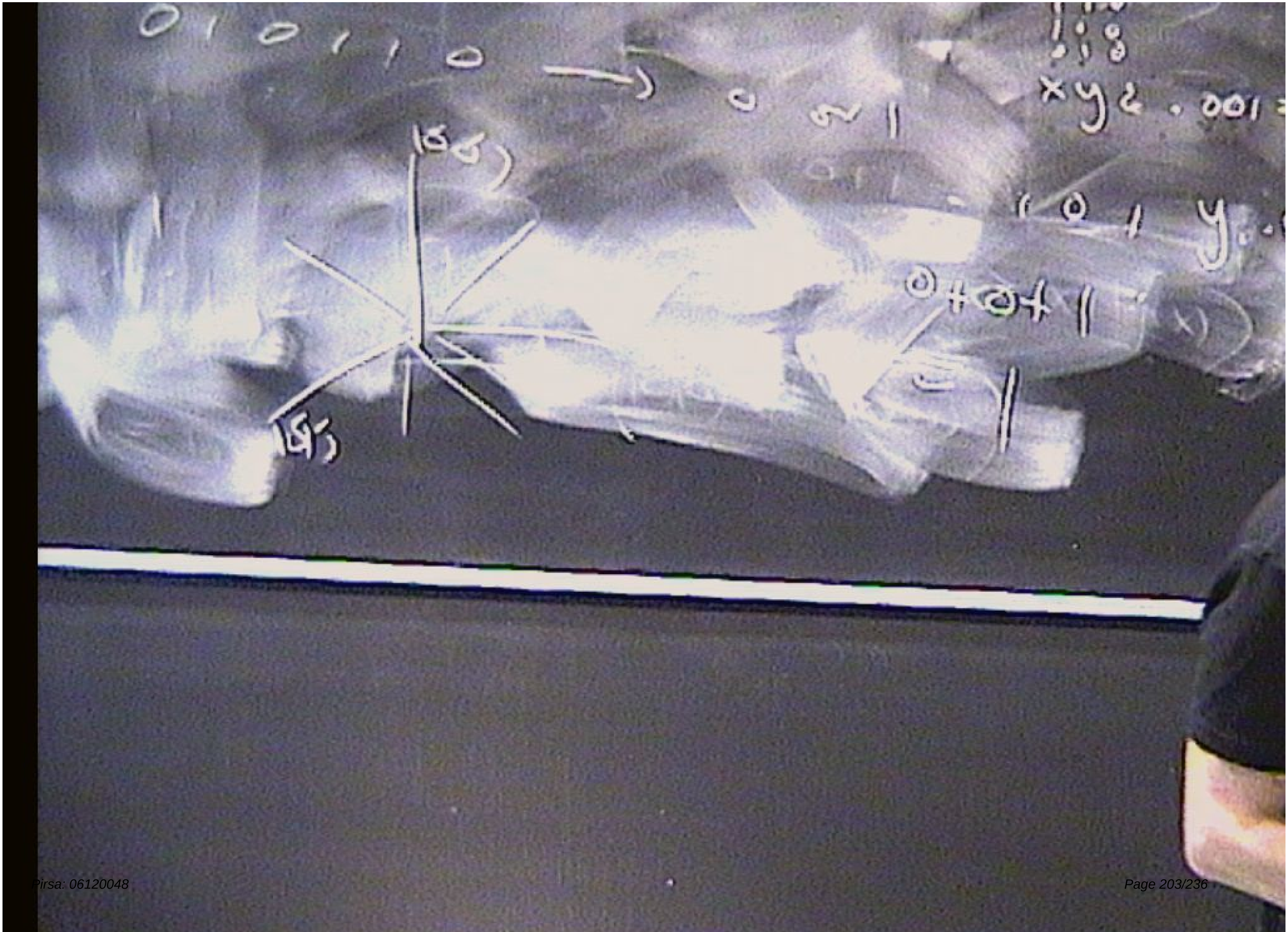
Parity problem

- $n = 2$ case: like Deutsch's problem, 1 constant state, 3 balanced states, 4 unbalanced states after U_f:

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

  with three $+$'s and one $-$'s.
- The constant and balanced states have parity 1; the unbalanced states have parity -1.




0 1 0 1 1 0
0+0+1

Parity problem

- $n = 2$ case: like Deutsch's problem, 1 constant state, 3 balanced states, 4 unbalanced states after U_f:

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

with three $+$'s and one $-$'s.

- The constant and balanced states have parity 1; the unbalanced states have parity -1.

Parity problem

To find the parity of f, apply VU_f twice, where V is the permutation:

$$
\begin{aligned}
V|00\rangle &\rightarrow |01\rangle \\
V|01\rangle &\rightarrow |00\rangle \\
V|10\rangle &\rightarrow |11\rangle \\
V|11\rangle &\rightarrow |10\rangle
\end{aligned}
$$

Parity problem

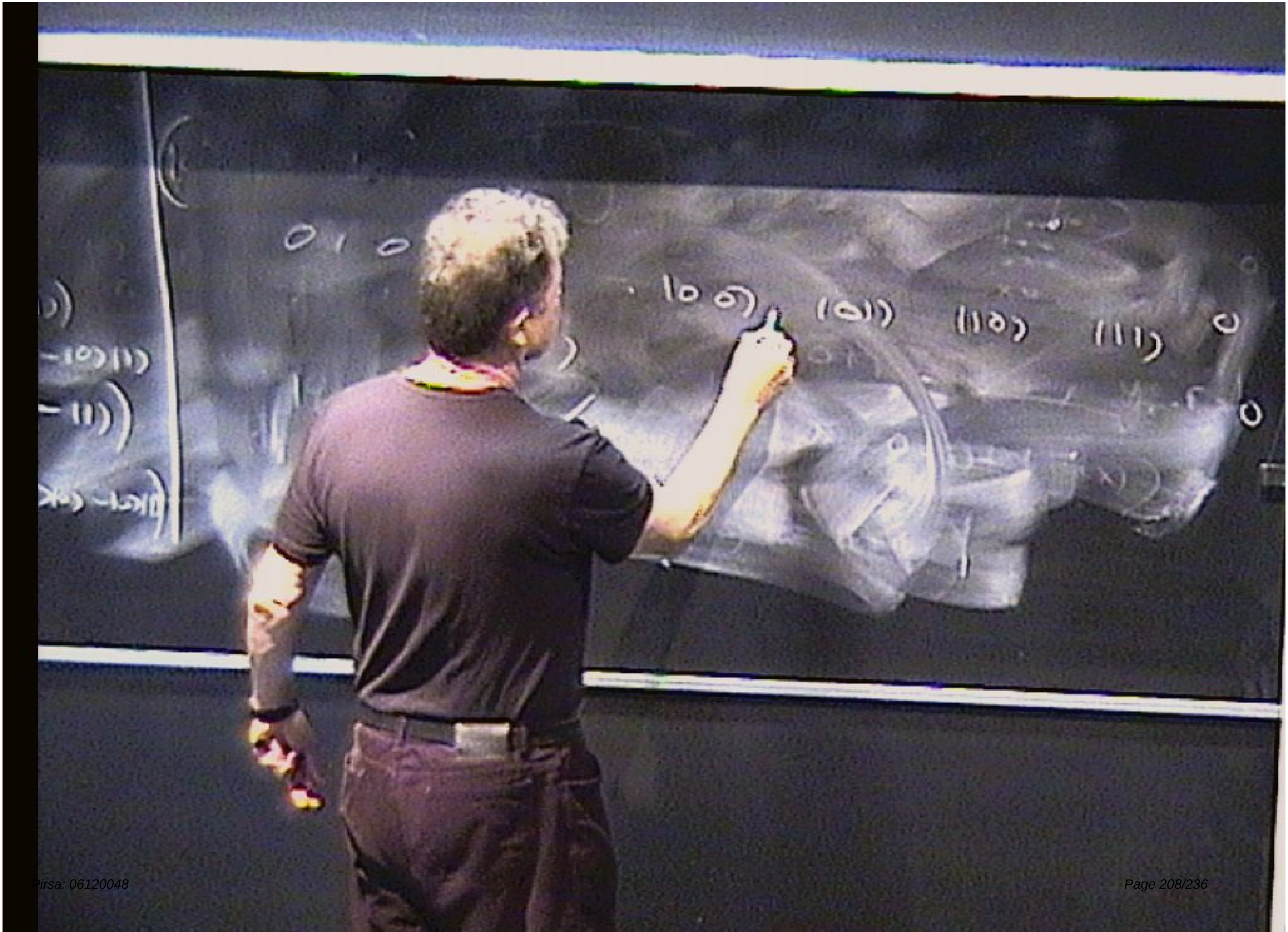
To find the parity of f, apply VU_f twice, where V is the permutation:

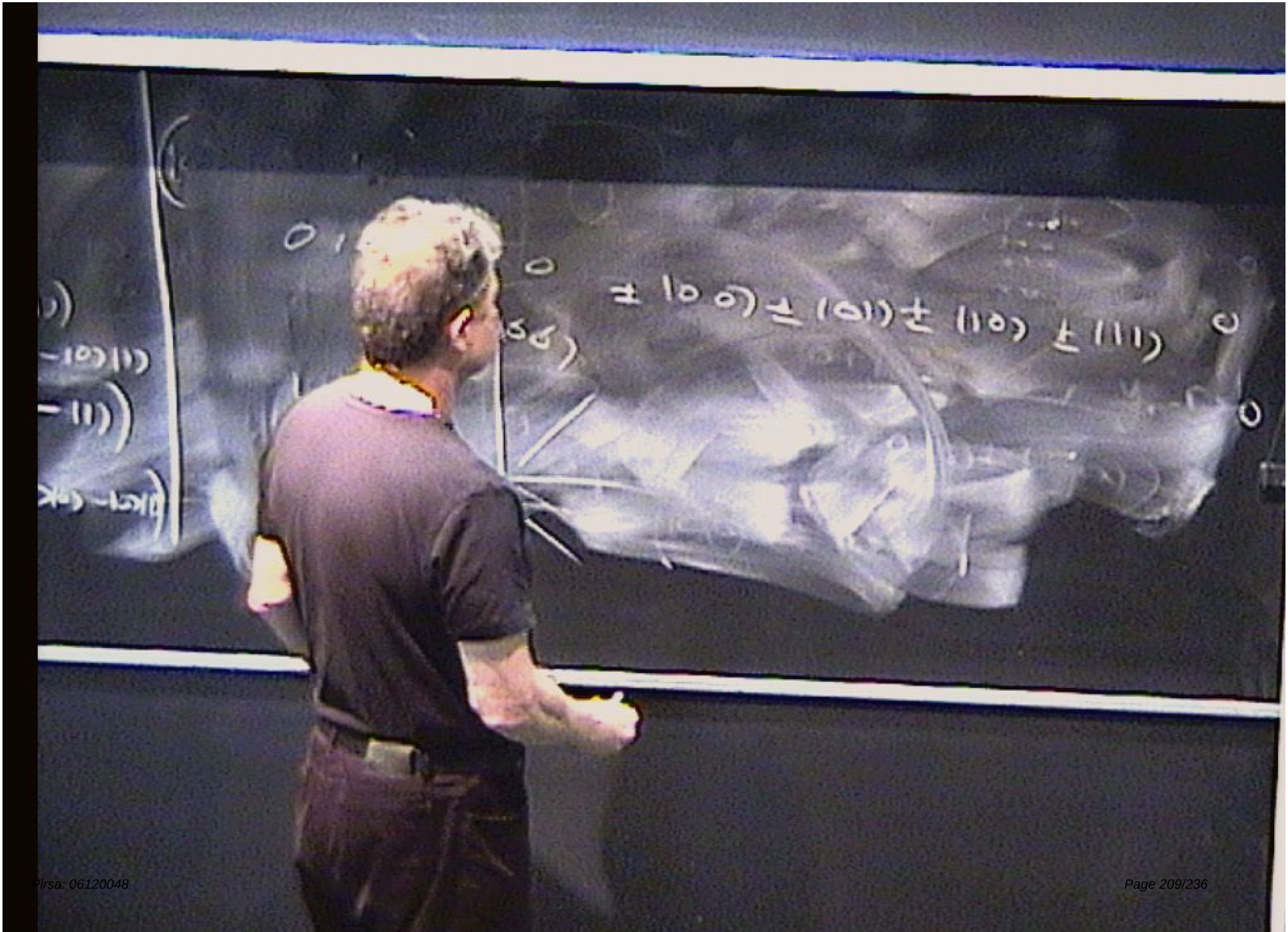
$$
\begin{aligned}
V|00\rangle &\rightarrow |01\rangle \\
V|01\rangle &\rightarrow |00\rangle \\
V|10\rangle &\rightarrow |11\rangle \\
V|11\rangle &\rightarrow |10\rangle
\end{aligned}
$$

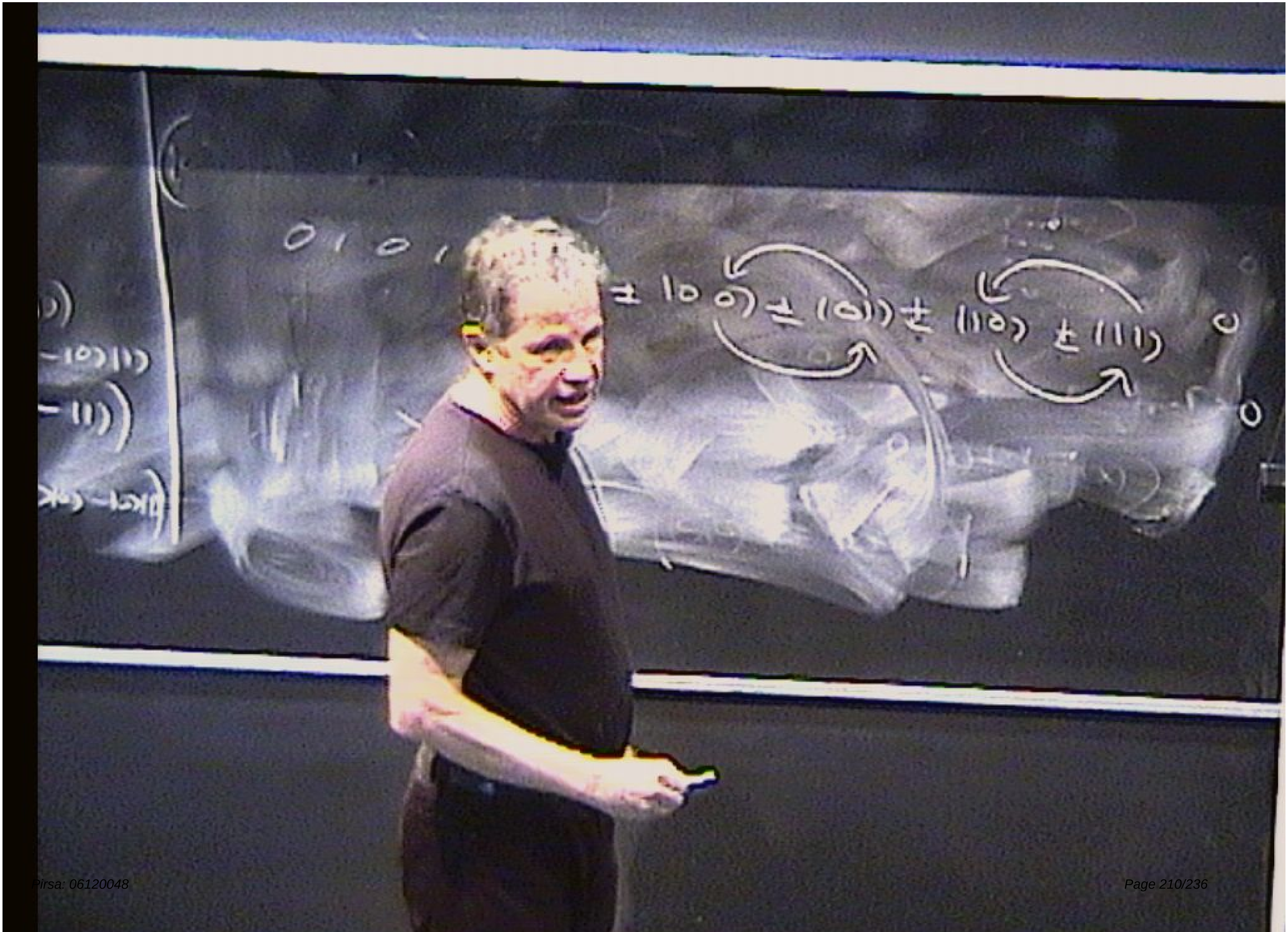
Parity problem

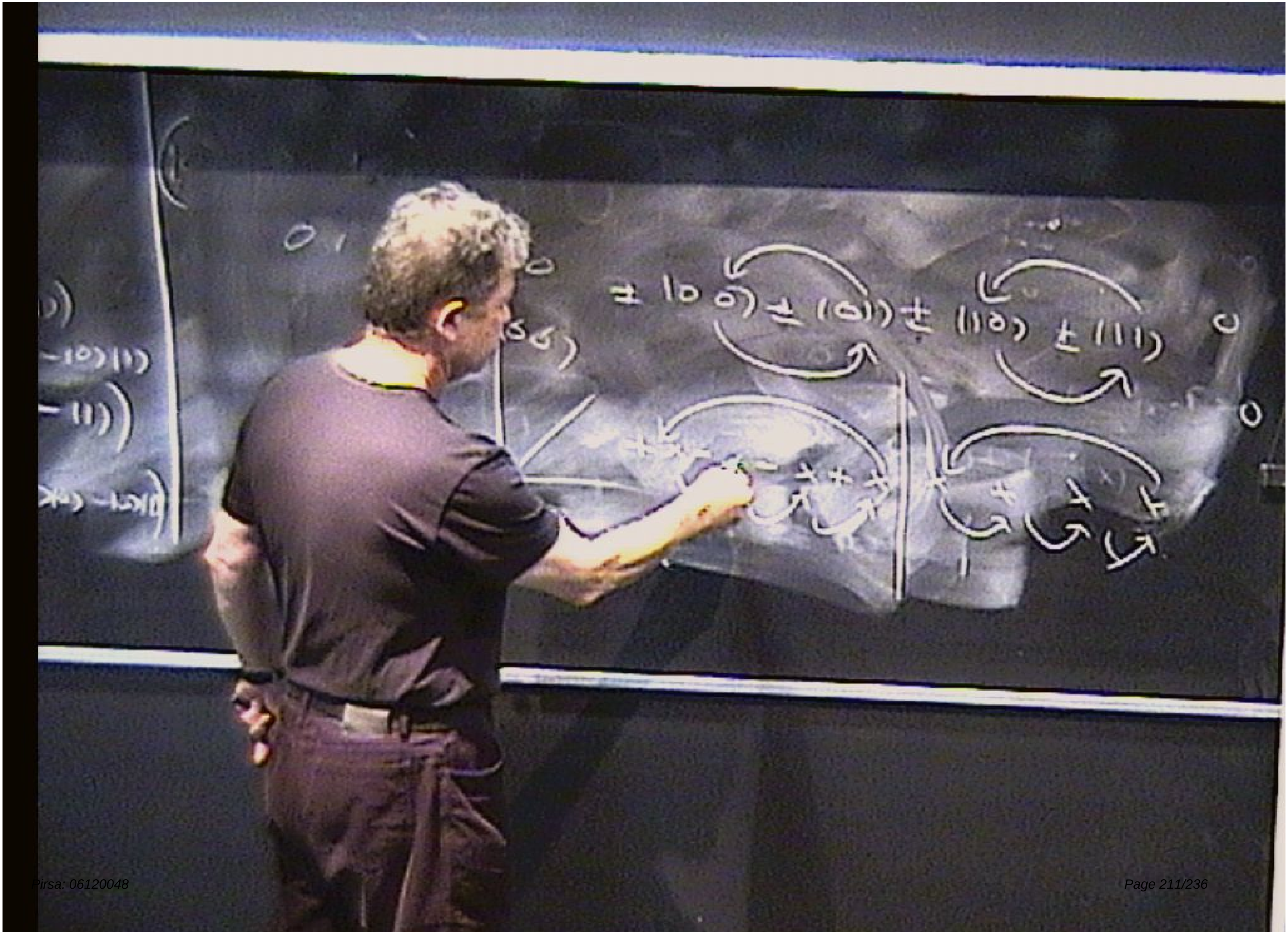
After $\mathrm{U_f}$, the state of the input register is:

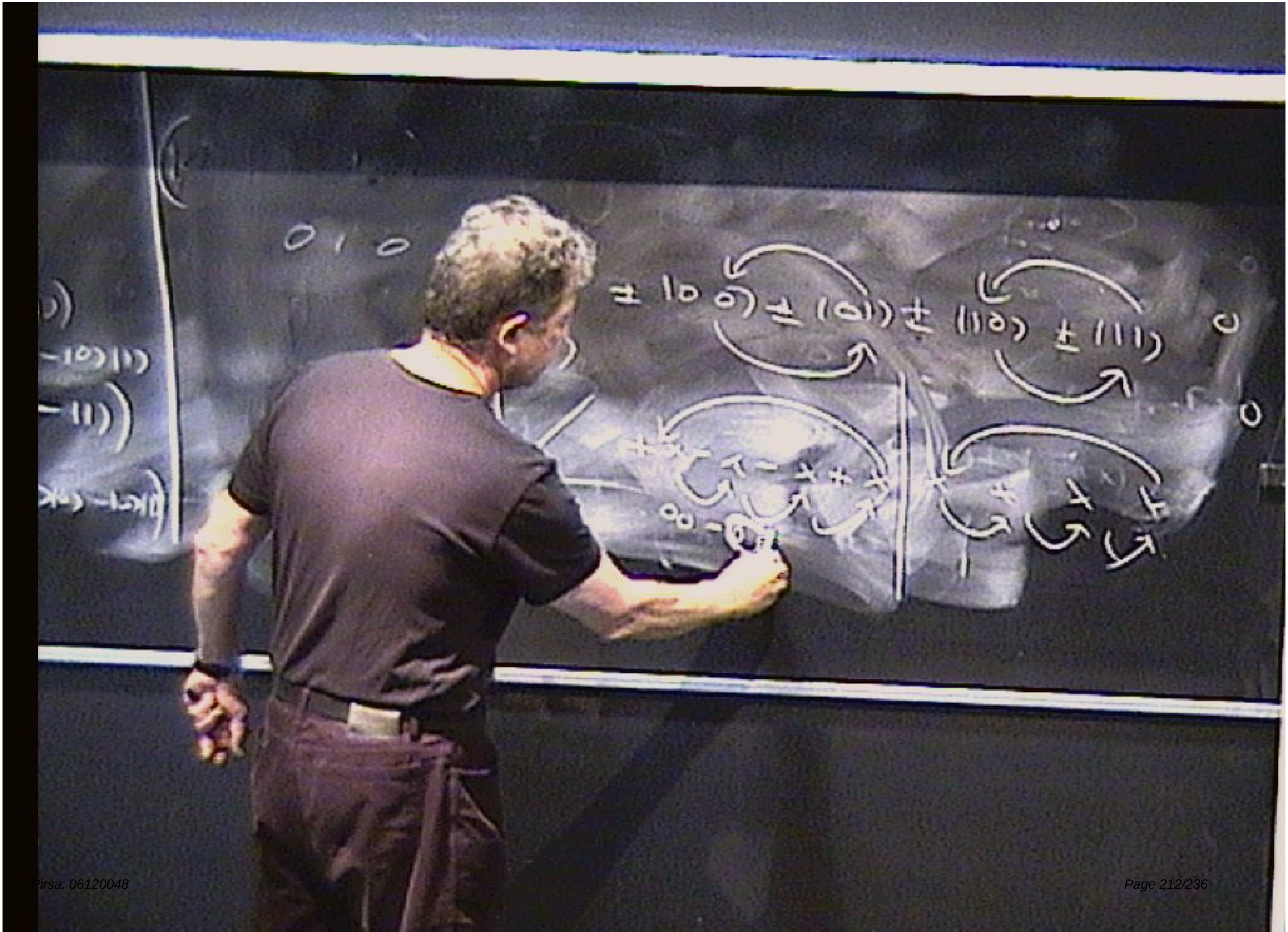
$$\frac{1}{2}((-1)^{\mathrm{f}(00)}|00\rangle + (-1)^{\mathrm{f}(01)}|01\rangle + (-1)^{\mathrm{f}(10)}|10\rangle + (-1)^{\mathrm{f}(11)}|11\rangle)$$

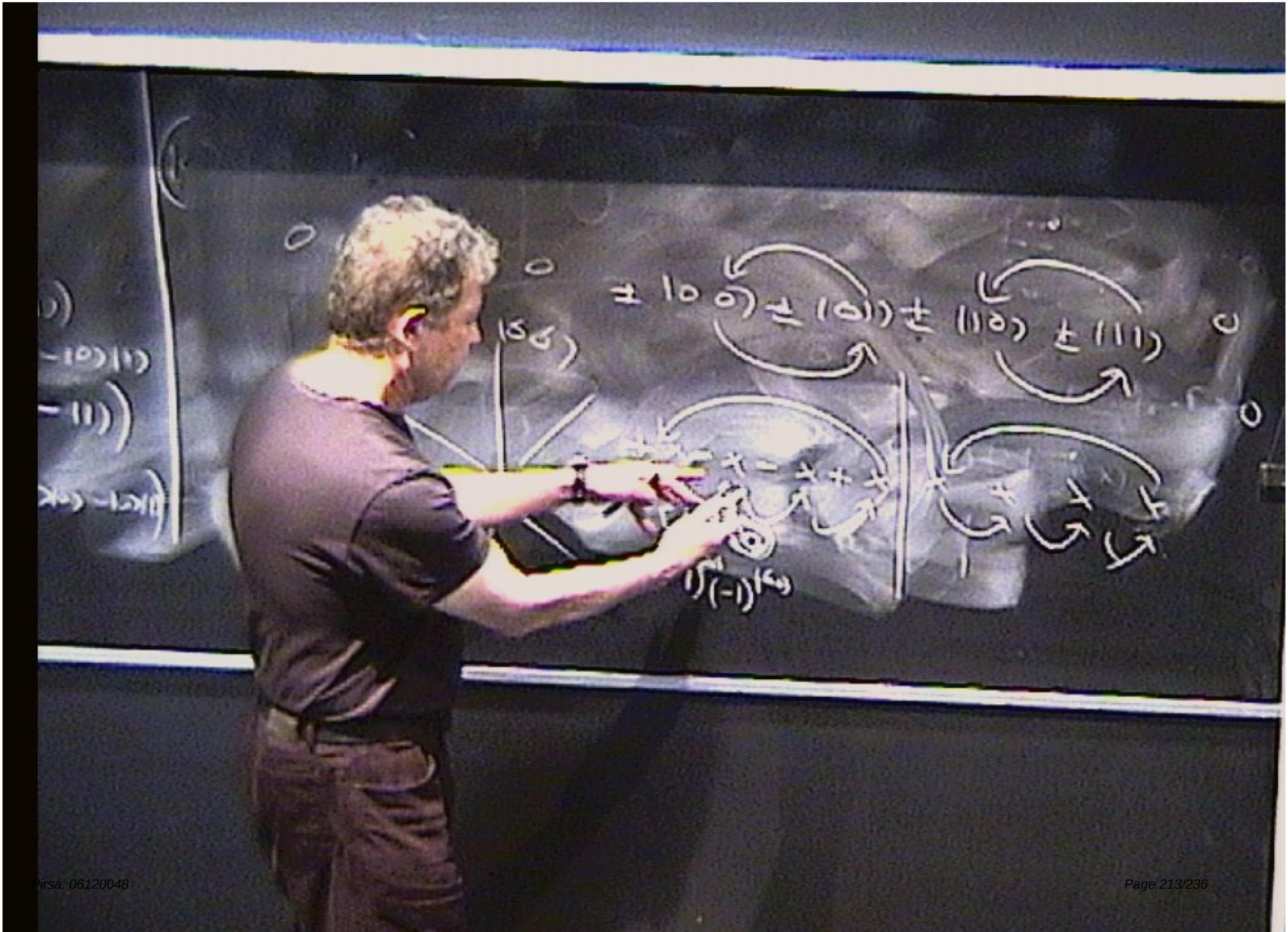

|00⟩ |01⟩ |10⟩ |11⟩

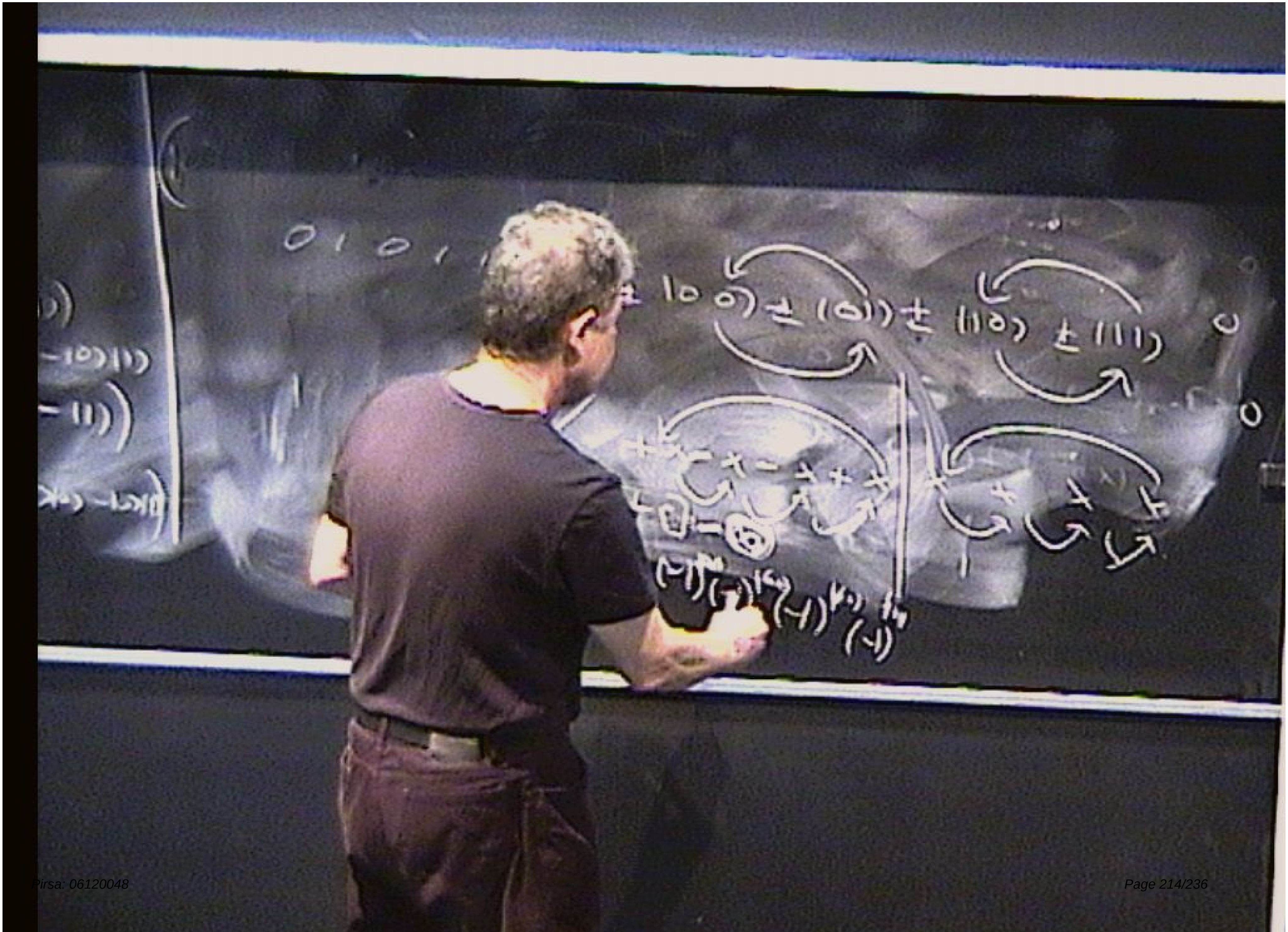

± |00⟩ ± |01⟩ ± |10⟩ ± |11⟩

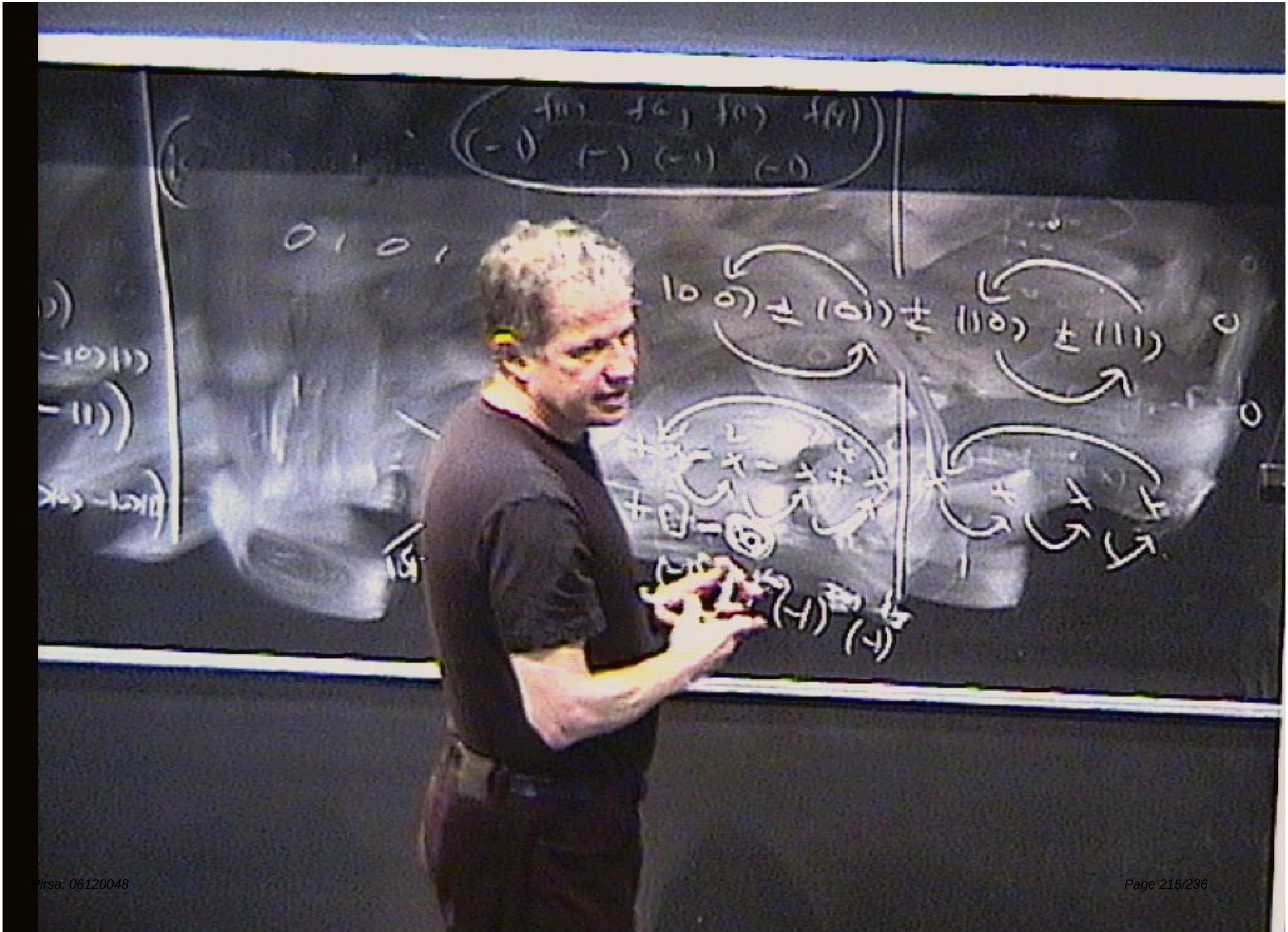



$\pm|00\rangle \pm |01\rangle \pm |10\rangle \pm |11\rangle$


0 1 0
± |100⟩ ± |101⟩ ± |110⟩ ± |111⟩


± |00⟩ ± |01⟩ ± |10⟩ ± |11⟩





Parity problem

- After applying VU_f twice, the state of the input register is:

$$\frac{1}{2}(-1)^{f(00)}(-1)^{f(01)}(|00\rangle+|01\rangle)+\frac{1}{2}(-1)^{f(10)}(-1)^{f(11)}(|10\rangle+|11\rangle)$$

- So the state is either:

  parity +1: $\pm\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$

  parity -1: $\pm\frac{1}{2}((|00\rangle + |01\rangle) - (|10\rangle + |11\rangle))$

- Parity determined in 2 runs of the algorithm instead of 4 classically.
- In general case, speedup N/2 is optimal for N values of the function.

Parity problem

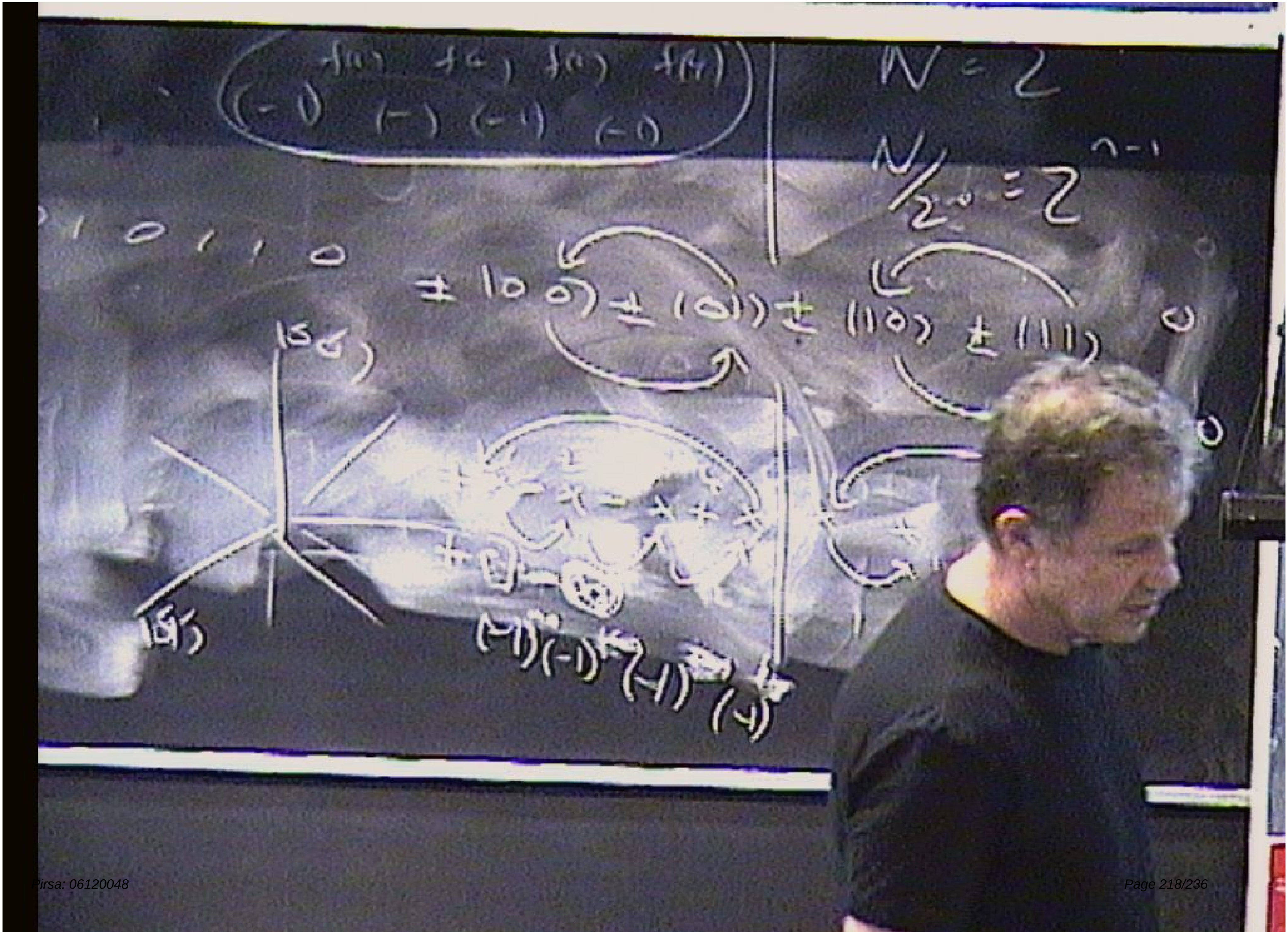
- After applying VU_f twice, the state of the input register is:

$$\frac{1}{2}(-1)^{f(00)}(-1)^{f(01)}(|00\rangle+|01\rangle)+\frac{1}{2}(-1)^{f(10)}(-1)^{f(11)}(|10\rangle+|11\rangle)$$

- So the state is either:

  parity +1: $\pm\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$
  parity -1: $\pm\frac{1}{2}((|00\rangle + |01\rangle) - (|10\rangle + |11\rangle))$

- Parity determined in 2 runs of the algorithm instead of 4 classically.
- In general case, speedup N/2 is optimal for N values of the function.


N = 2
± |00⟩ ± |01⟩ ± |10⟩ ± |11⟩

Grover's search algorithm

- $f : B^n \rightarrow B, B = \{0, 1\}$
- Promised that $f(x) = 0$ for all bit strings except one, x_0.
- Find x_0.

Grover's search algorithm

- $f : B^n \rightarrow B, B = \{0, 1\}$
- Promised that $f(x) = 0$ for all bit strings except one, x_0.
- Find x_0.

Grover's search algorithm

- $f : B^n \to B, B = \{0, 1\}$
- Promised that $f(x) = 0$ for all bit strings except one, x_0.
- Find x_0.

Grover's search algorithm

$$|0\rangle^{\otimes n}|1\rangle \xrightarrow{H} \frac{1}{\sqrt{2^n}}\sum_{x\in B^n}|x\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

$$\xrightarrow{U_f} \frac{1}{\sqrt{2^n}}\sum_{x}(-1)^{f(x)}|x\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

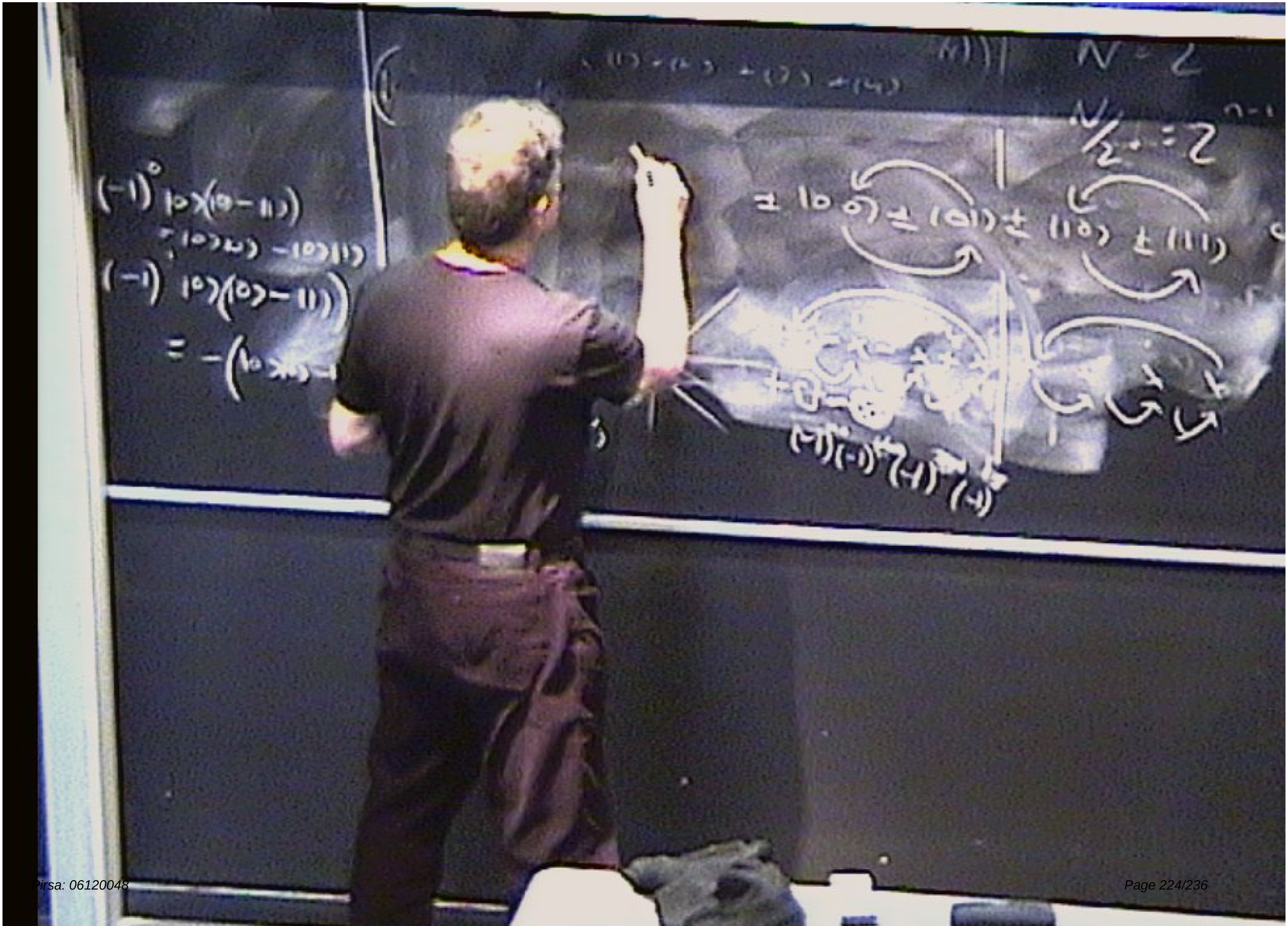
Reflects $\frac{1}{\sqrt{2^n}}\sum_x |x\rangle$ in hyperplane orthogonal to $|x_0\rangle$: equivalent to applying $I_{|x_0\rangle} = I - 2|x_0\rangle\langle x_0|$.

Grover's search algorithm

$$|0\rangle^{\otimes n}|1\rangle \quad \xrightarrow{H} \quad \frac{1}{\sqrt{2^n}}\sum_{x\in B^n}|x\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

$$\xrightarrow{U_f} \quad \frac{1}{\sqrt{2^n}}\sum_{x}(-1)^{f(x)}|x\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

Reflects $\frac{1}{\sqrt{2^n}}\sum_x|x\rangle$ in hyperplane orthogonal to $|x_0\rangle$: equivalent to applying $I_{|x_0\rangle} = I - 2|x_0\rangle\langle x_0|$.



Grover's search algorithm

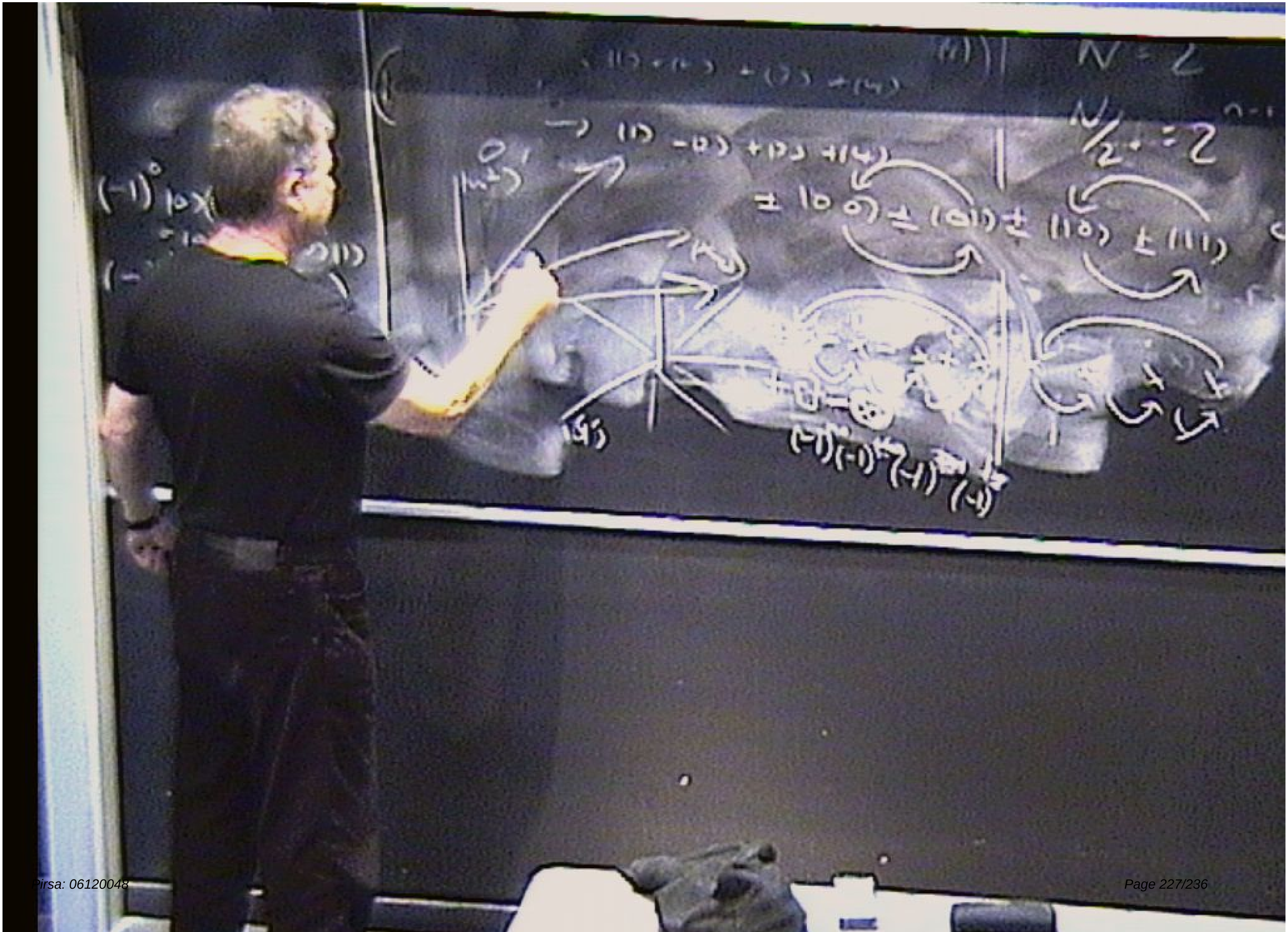
$$|0\rangle^{\otimes n}|1\rangle \quad \xrightarrow{H} \quad \frac{1}{\sqrt{2^n}} \sum_{x \in B^n} |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$\xrightarrow{U_f} \quad \frac{1}{\sqrt{2^n}} \sum_{x} (-1)^{f(x)} |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Reflects $\frac{1}{\sqrt{2^n}} \sum_x |x\rangle$ in hyperplane orthogonal to $|x_0\rangle$: equivalent to applying $I_{|x_0\rangle} = I - 2|x_0\rangle\langle x_0|$.

Grover's search algorithm

- Write: $|w\rangle = \frac{1}{2^n}\sum_{x\in B^n}|x\rangle = \frac{1}{\sqrt{N}}\sum_{x=1}^{N}|x\rangle$
- $I_{|w\rangle} = I - 2|w\rangle\langle w|$ is a reflection in the hyperplane orthogonal to $|w\rangle$.
- $-I_{|w\rangle}I_{|x_0\rangle} = I_{|w^\perp\rangle}I_{|x_0\rangle}$, where $|w^\perp\rangle$ is orthogonal to $|w\rangle$ in the plane spanned by $|w\rangle$ and $|x_0\rangle$.
- $I_{|w^\perp\rangle}I_{|x_0\rangle}$ is rotation through 2β, where β is angle between $|w^\perp\rangle$ and $|x_0\rangle$.



Grover's search algorithm

- Consider case $n = 2$ and suppose $|x_0\rangle = |11\rangle = |4\rangle$
- After H, state of input register is:

$$\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \frac{1}{2}(|1\rangle + |1\rangle + |3\rangle + |4\rangle)$$

Grover's search algorithm

- After $I_{|x_0\rangle} = I - 2|x_0\rangle\langle x_0|$ (U_f with output register in state $H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$), state of input register is:

$$\frac{1}{2}(|1\rangle + |1\rangle + |3\rangle - |4\rangle)$$

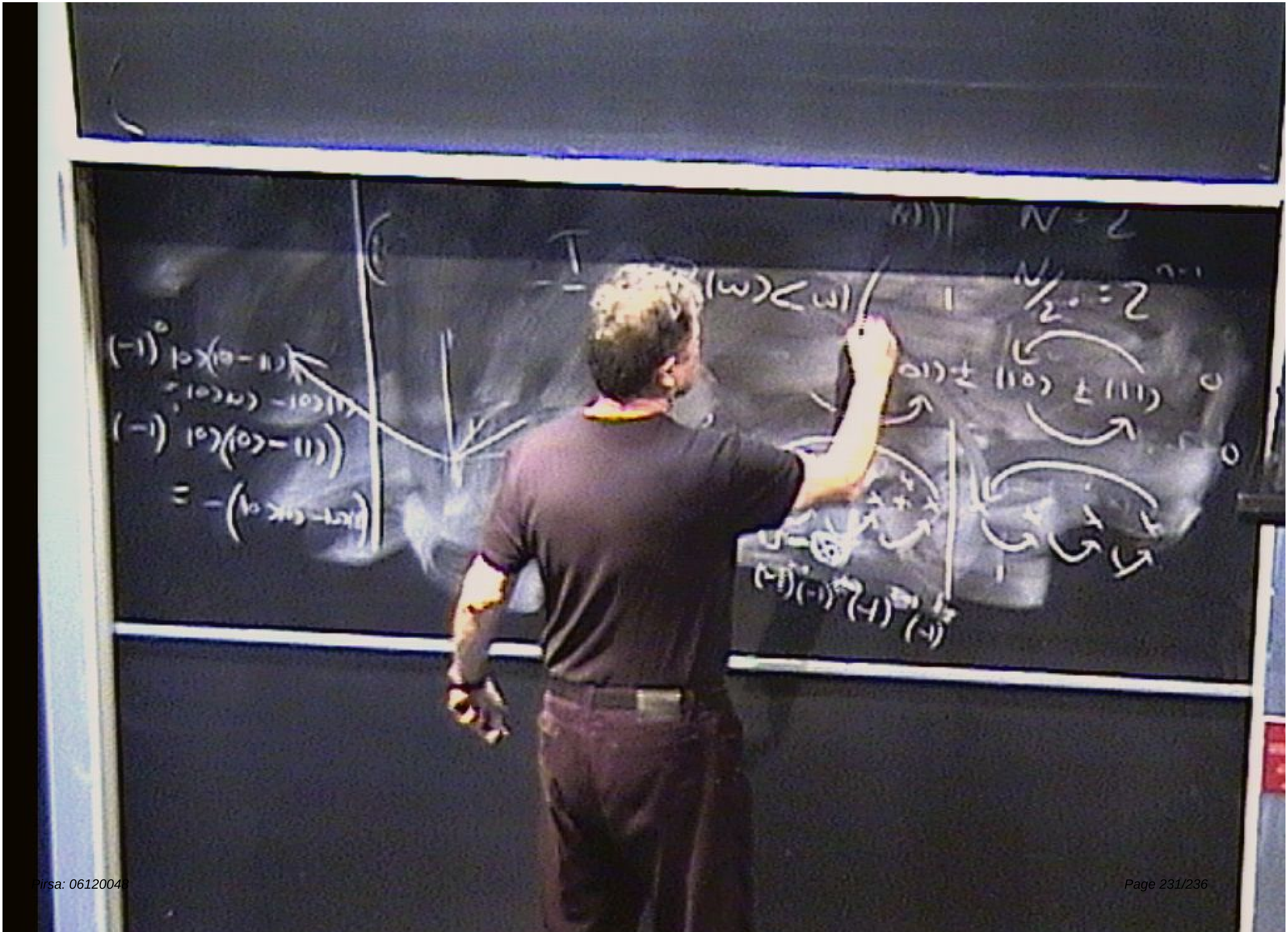
- After $-I_{|w\rangle} I_{|x_0\rangle} = I_{|w^{\perp}\rangle} I_{|x_0\rangle}$, state of input register is $|4\rangle$

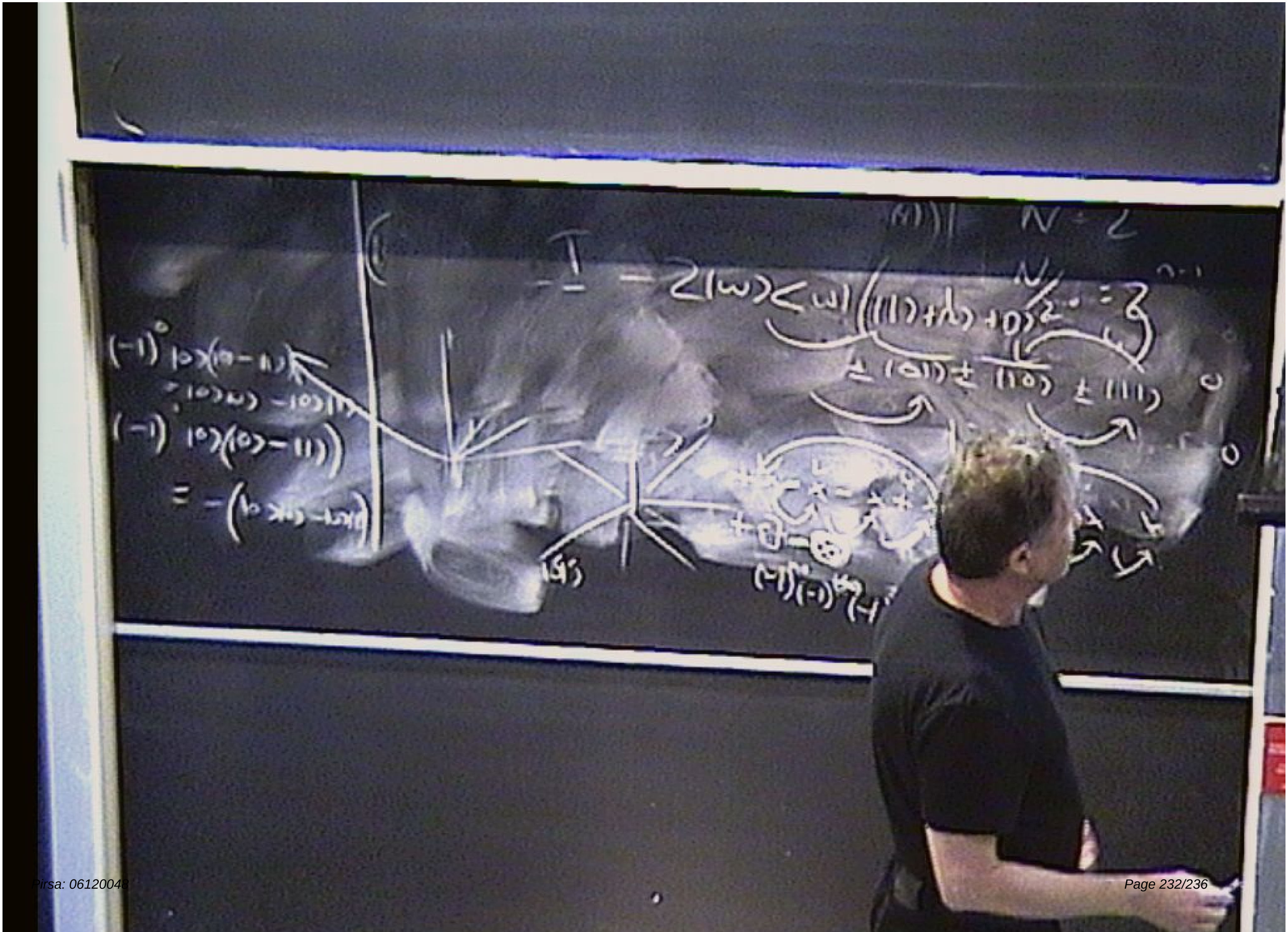
Grover's search algorithm

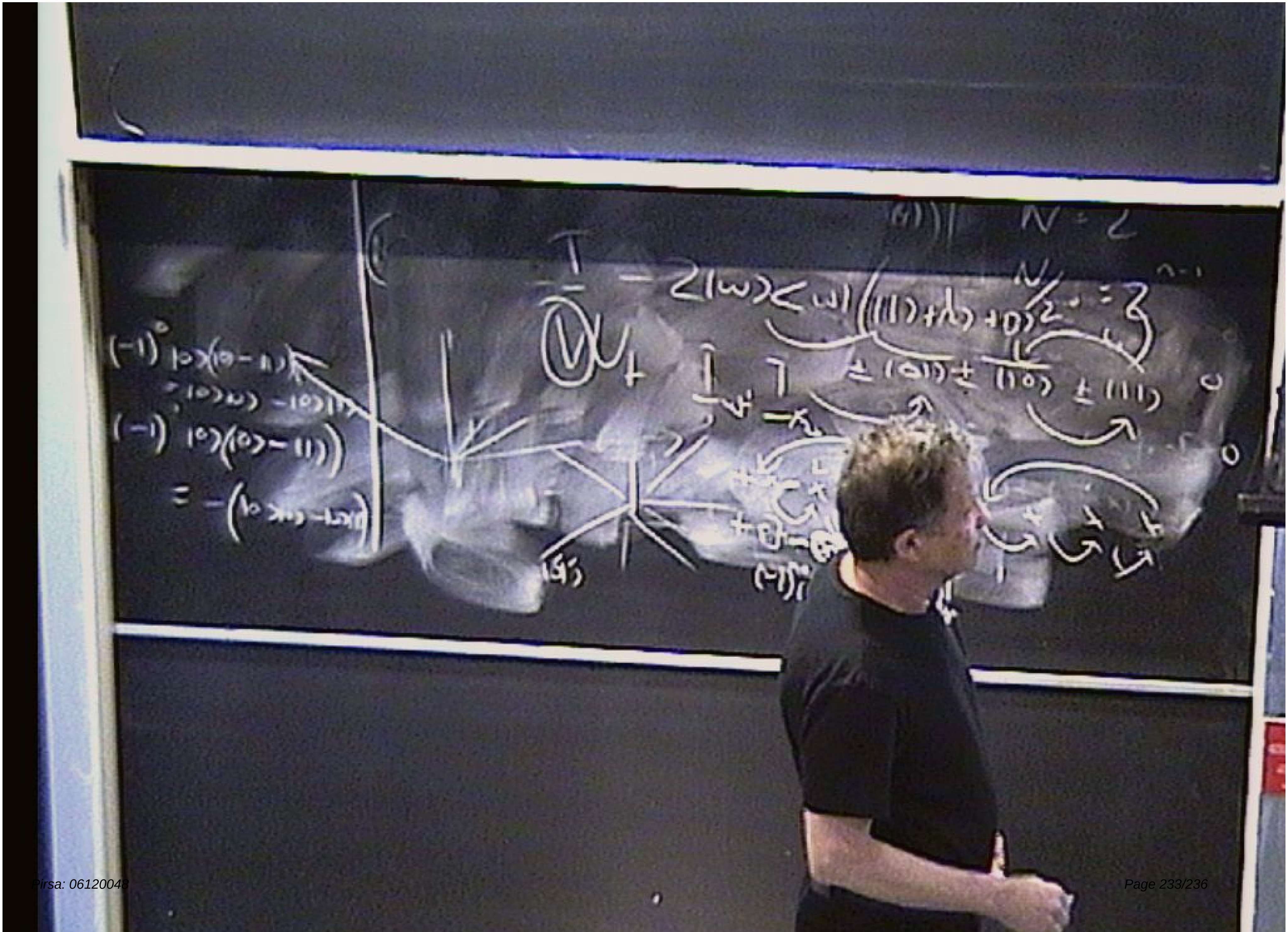
- After $I_{|x_0\rangle} = I - 2|x_0\rangle\langle x_0|$ (U_f with output register in state $H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$), state of input register is:

$$\frac{1}{2}(|1\rangle + |1\rangle + |3\rangle - |4\rangle)$$

- After $-I_{|w\rangle} I_{|x_0\rangle} = I_{|w^\perp\rangle} I_{|x_0\rangle}$, state of input register is $|4\rangle$







Grover's search algorithm

- Angle α between $|\mathrm{w}\rangle = \frac{1}{\sqrt{\mathrm{N}}}\sum_{\mathrm{x}=1}^{\mathrm{N}}|\mathrm{x}\rangle$ and $|\mathrm{x}_0\rangle$ is such that $\cos\alpha = \frac{1}{\sqrt{\mathrm{N}}}$ for any $|\mathrm{x}_0\rangle$.
- If N is large, $\alpha \approx \pi/2$.
- So angle β between $|\mathrm{w}^{\perp}\rangle$ and $|\mathrm{x}_0\rangle$ is given by $\sin\beta = \langle \mathrm{x}_0|\mathrm{w}\rangle = \frac{1}{\sqrt{\mathrm{N}}}$. So $\beta \approx \frac{1}{\sqrt{\mathrm{N}}}$.
- We need $\mathrm{O}(\sqrt{\mathrm{N}})$ iterations of rotation through angle $2\beta \approx 1/\sqrt{\mathrm{N}}$ to move $\frac{1}{\sqrt{\mathrm{N}}}\sum_{\mathrm{x}=1}^{\mathrm{N}}|\mathrm{x}\rangle$ near to $|\mathrm{x}_0\rangle$.
- In general case, apply $-\mathrm{I}_{|\mathrm{w}\rangle}\mathrm{I}_{|\mathrm{x}_0\rangle}$ $\mathrm{O}(\sqrt{\mathrm{N}})$ times to rotate $\frac{1}{\sqrt{\mathrm{N}}}\sum_{\mathrm{x}=1}^{\mathrm{N}}|\mathrm{x}\rangle$ to $|\mathrm{x}_0\rangle$ through angle $\theta \approx \frac{1}{\sqrt{\mathrm{N}}}$.

Grover's search algorithm

- After $I_{|x_0\rangle} = I - 2|x_0\rangle\langle x_0|$ (U_f with output register in state $H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$), state of input register is:

$$\frac{1}{2}(|1\rangle + |1\rangle + |3\rangle - |4\rangle)$$

- After $-I_{|w\rangle} I_{|x_0\rangle} = I_{|w^\perp\rangle} I_{|x_0\rangle}$, state of input register is $|4\rangle$