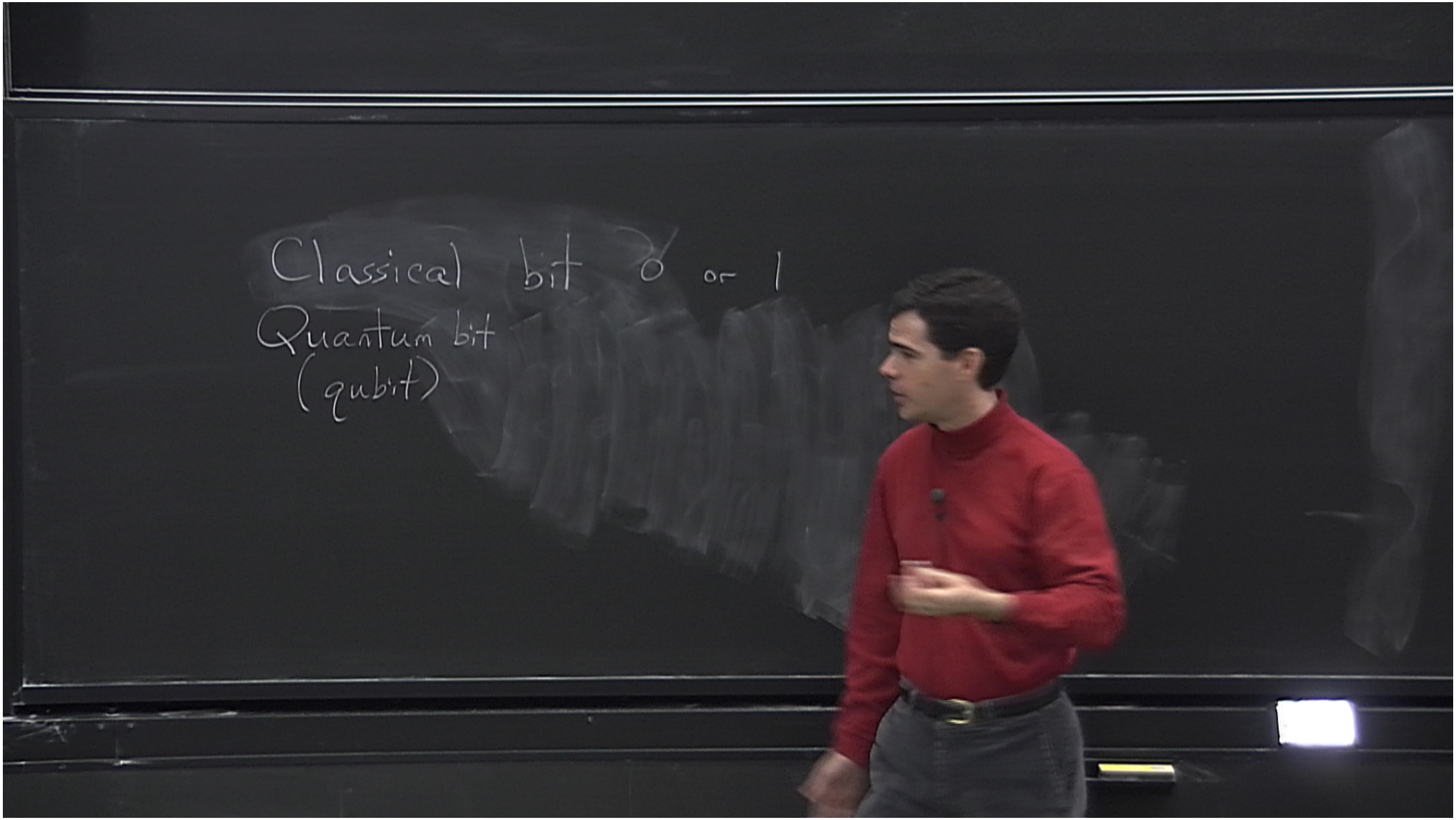
Title: Quantum Information (Review) - Lecture 1

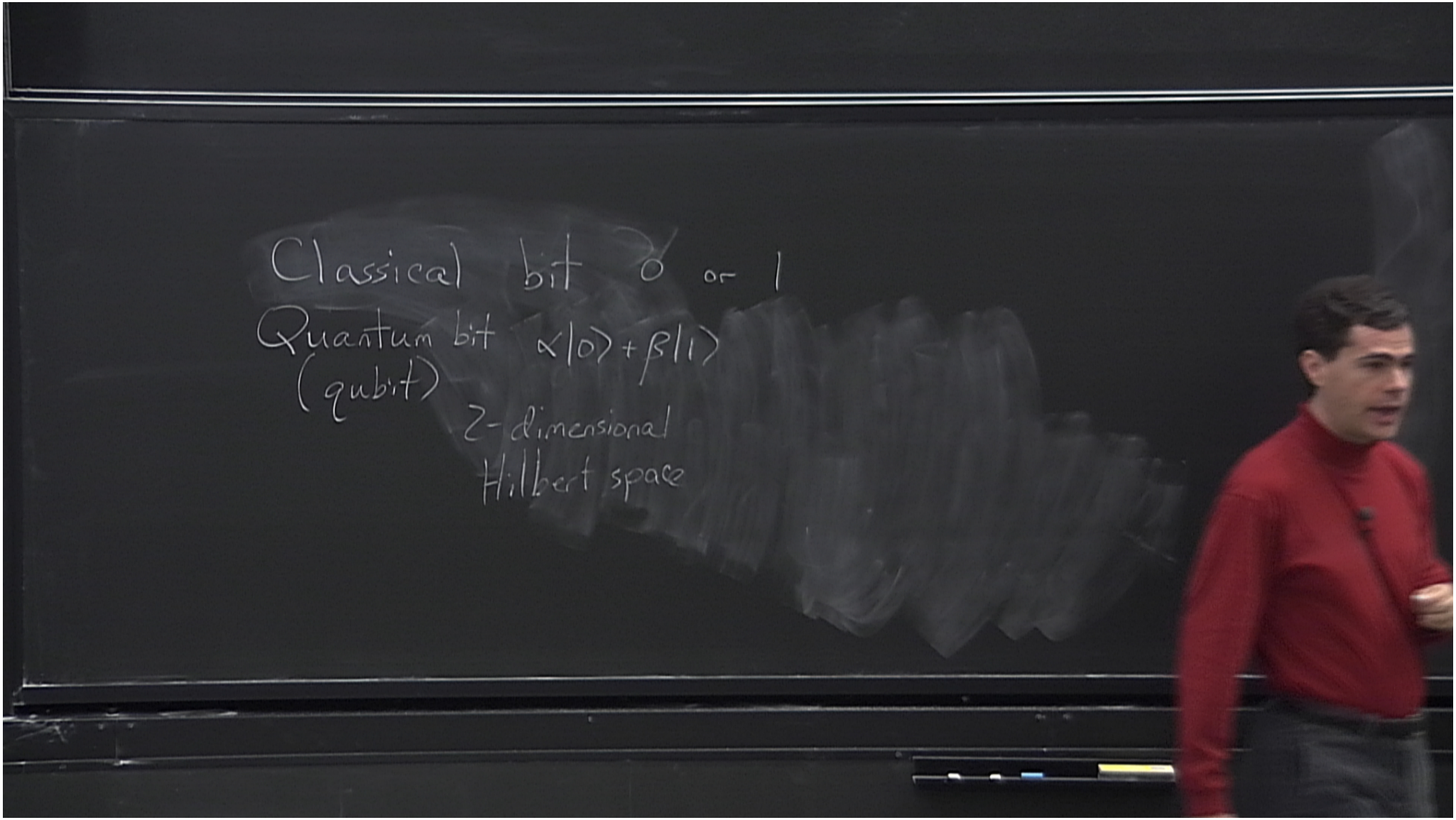
Date: Feb 13, 2012 10:15 AM

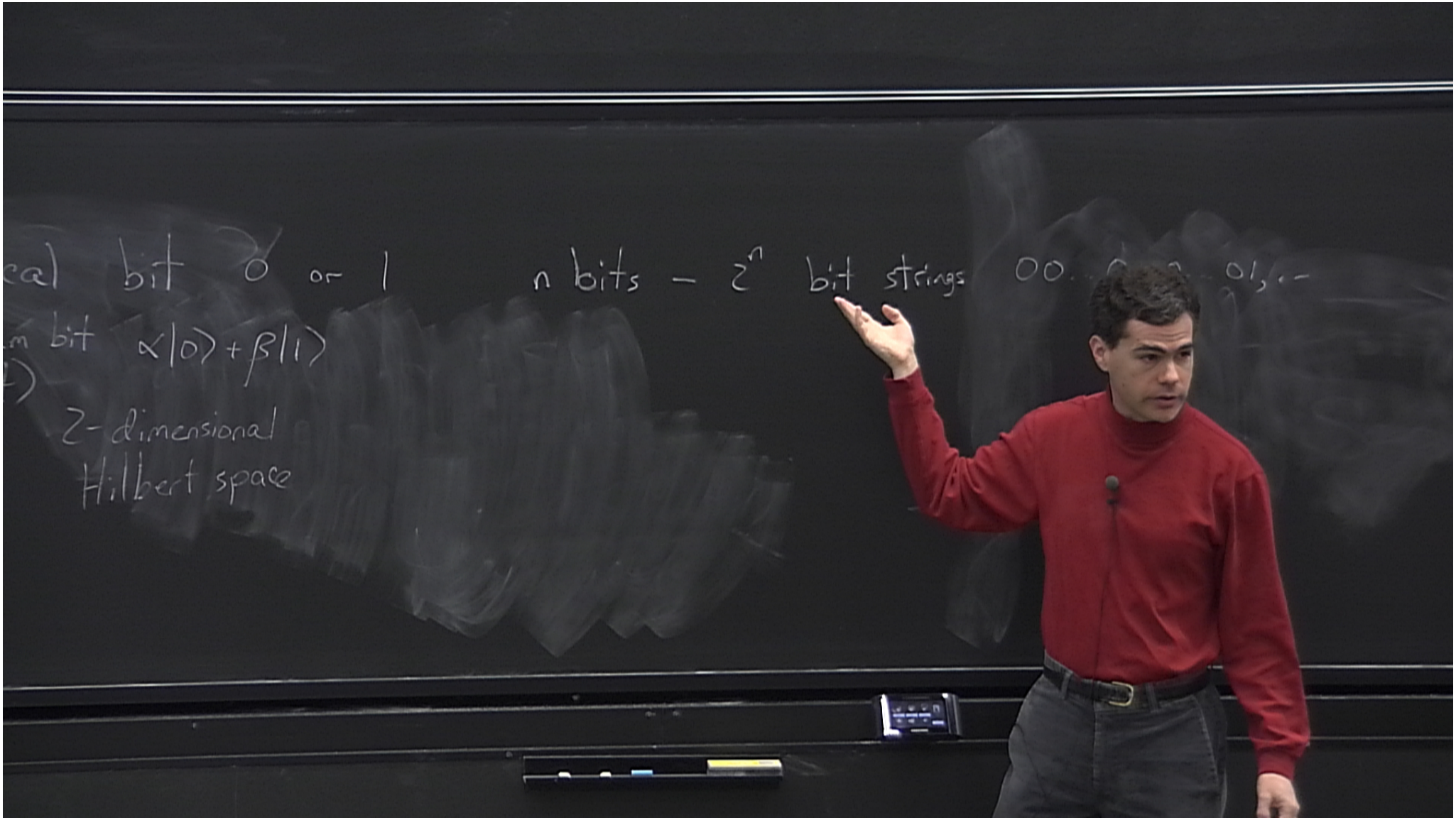
URL: http://pirsa.org/12020030

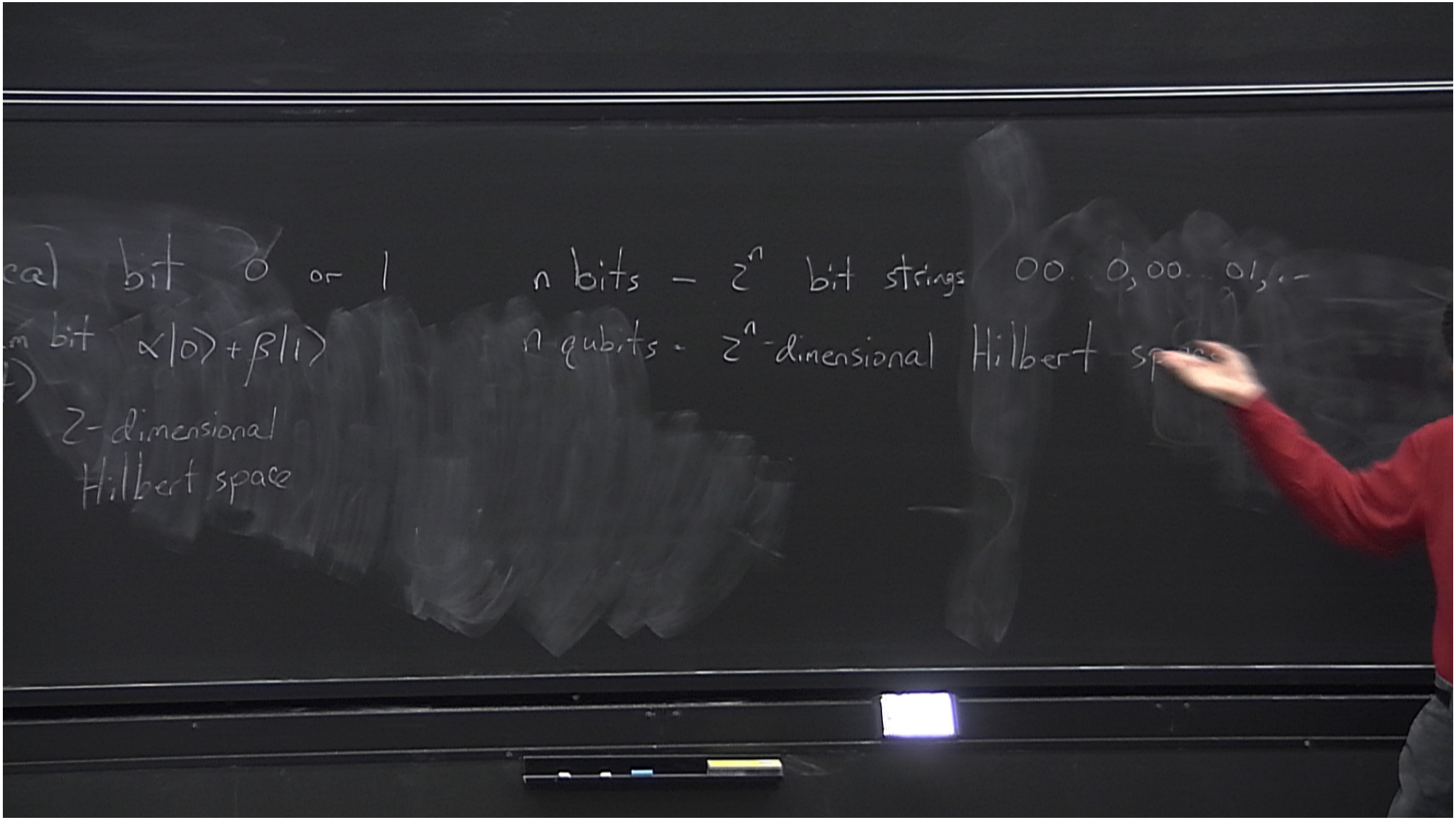
Abstract:

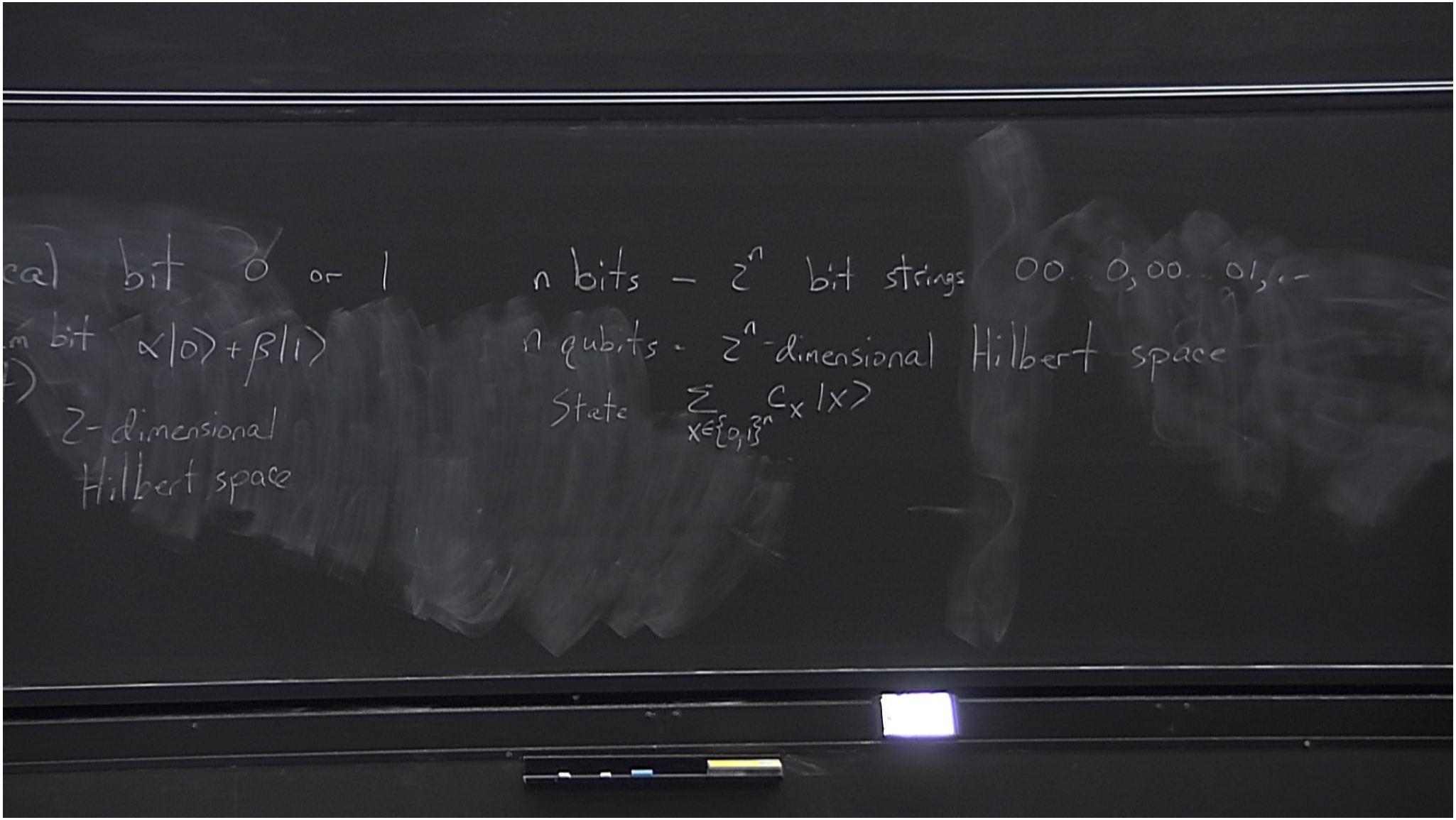


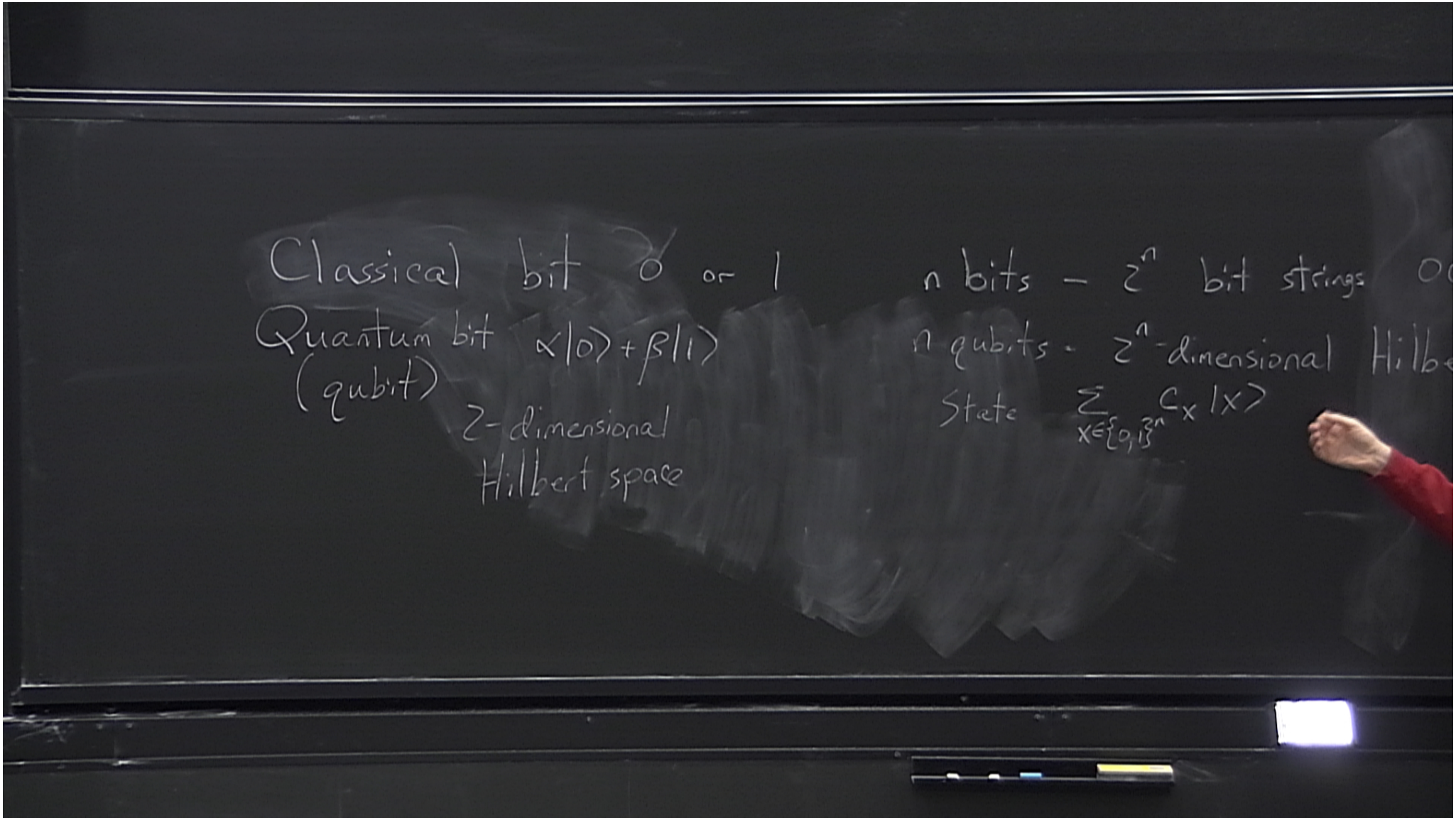

Classical bit 0 or 1
Quantum bit
(qubit)

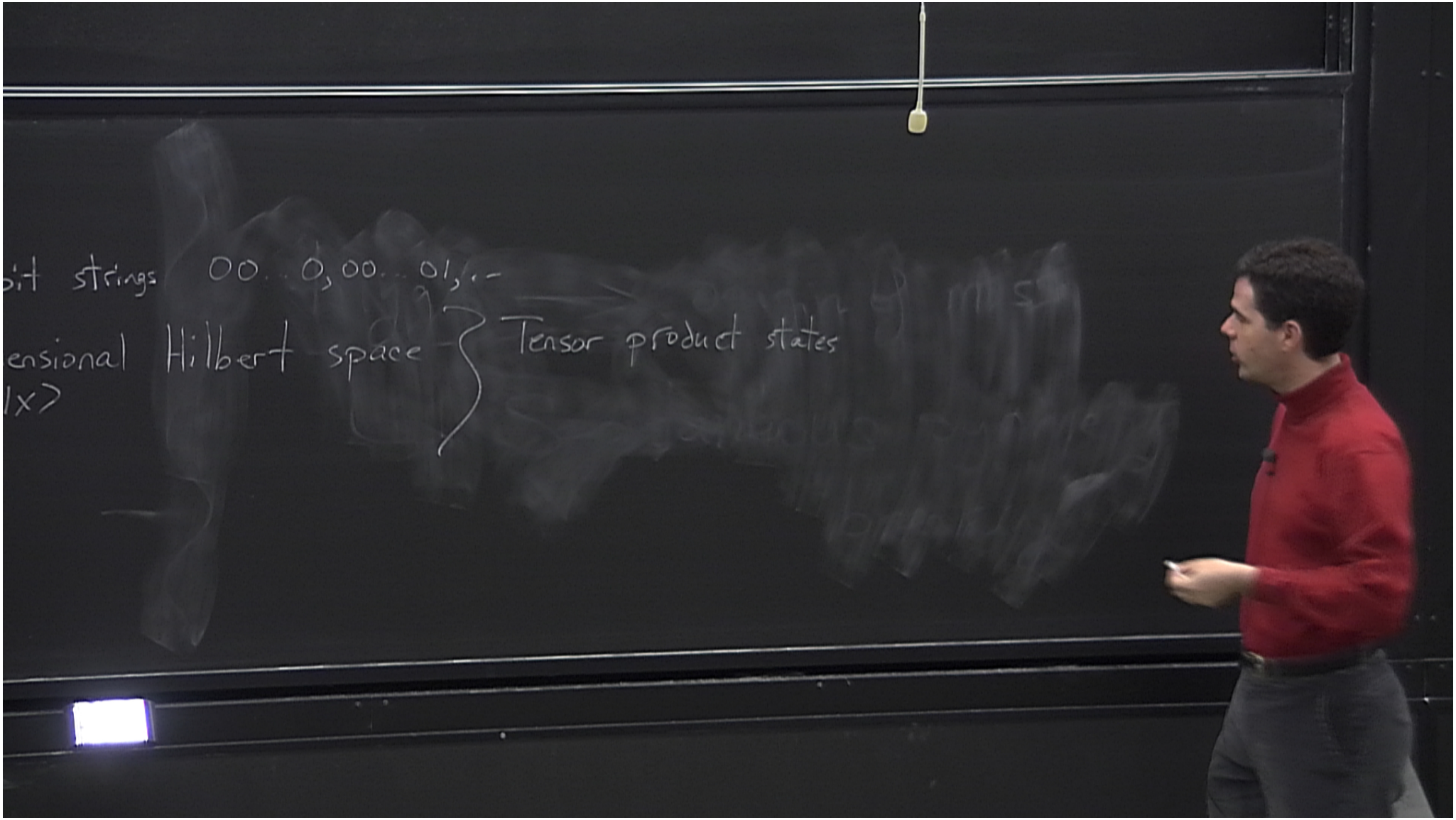

Classical bit 0 or 1
Quantum bit $\alpha|0\rangle + \beta|1\rangle$
(qubit)
2-dimensional
Hilbert space

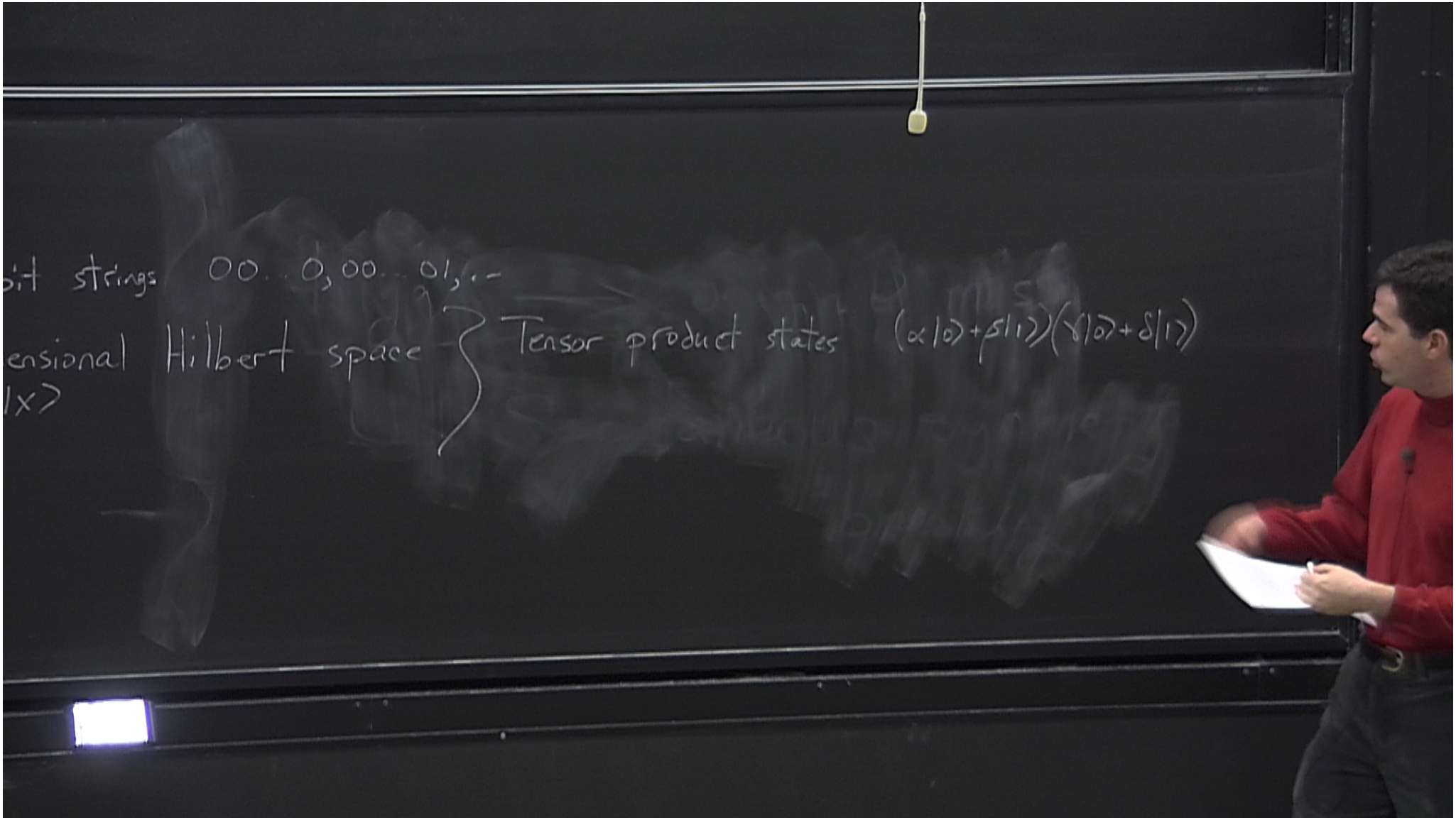

bit 0 or 1
n bits — 2^n bit strings
bit α|0⟩+β|1⟩
2-dimensional
Hilbert space

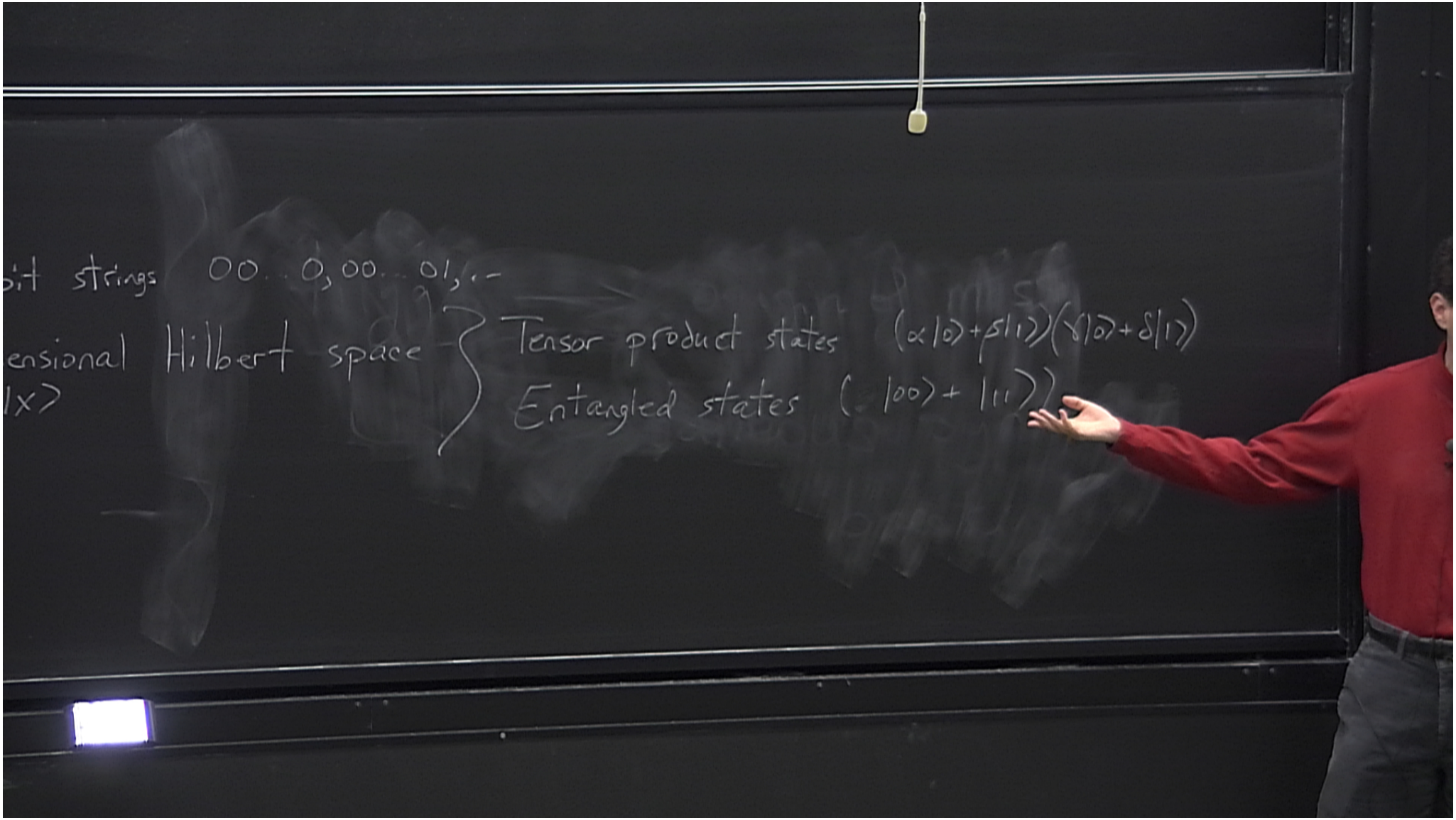

cal bit 0 or 1
n bits – 2^n bit strings 00...0, 00...01, ...
m bit α|0⟩+β|1⟩
n qubits - 2^n-dimensional Hilbert sp
2-dimensional
Hilbert space

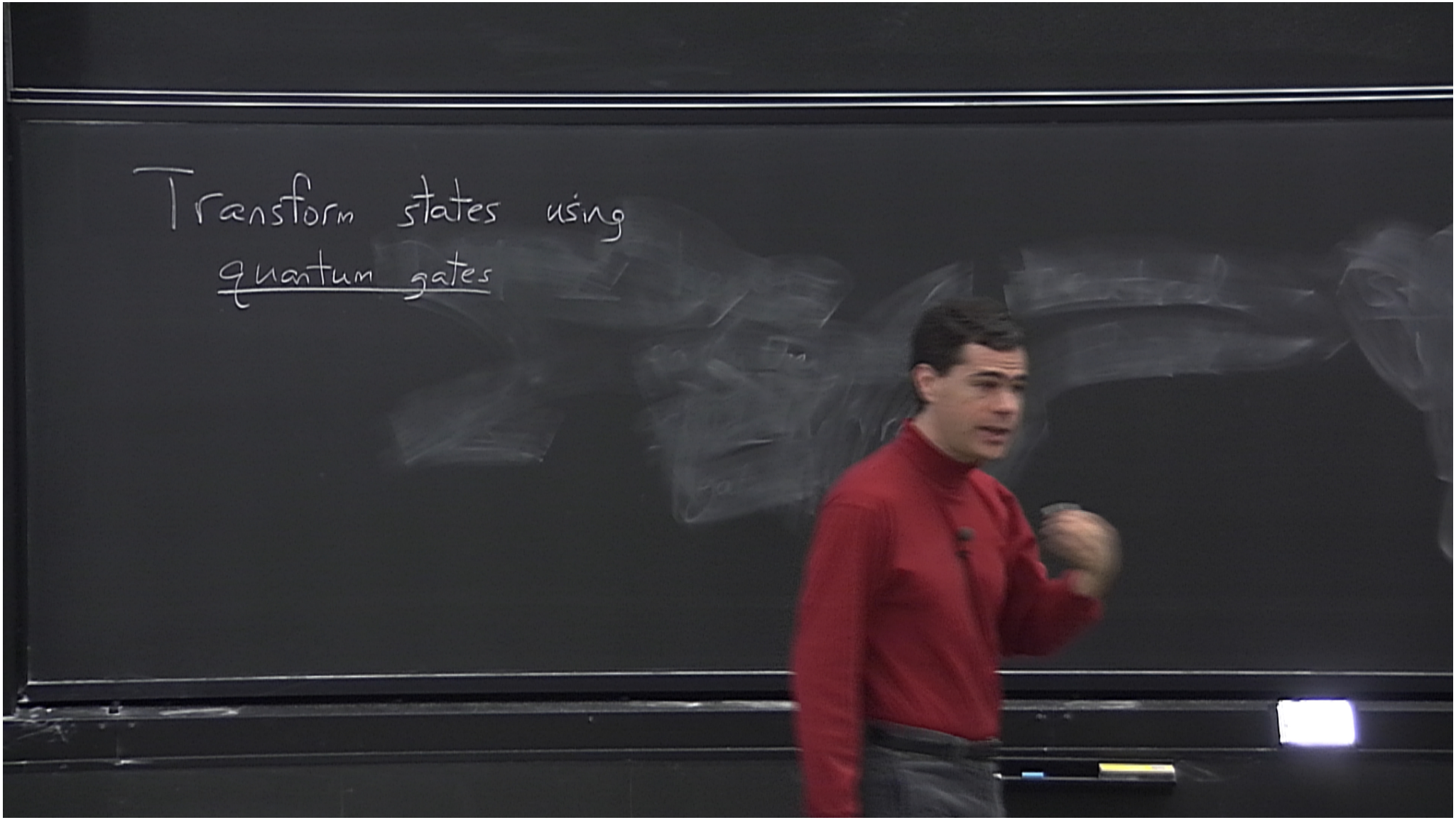

cal bit 0 or 1
n bits – 2^n bit strings 00...0, 00...01, ...
m bit $\alpha|0\rangle + \beta|1\rangle$
n qubits - 2^n-dimensional Hilbert space
State $\sum_{x \in \{0,1\}^n} c_x |x\rangle$
2-dimensional
Hilbert space

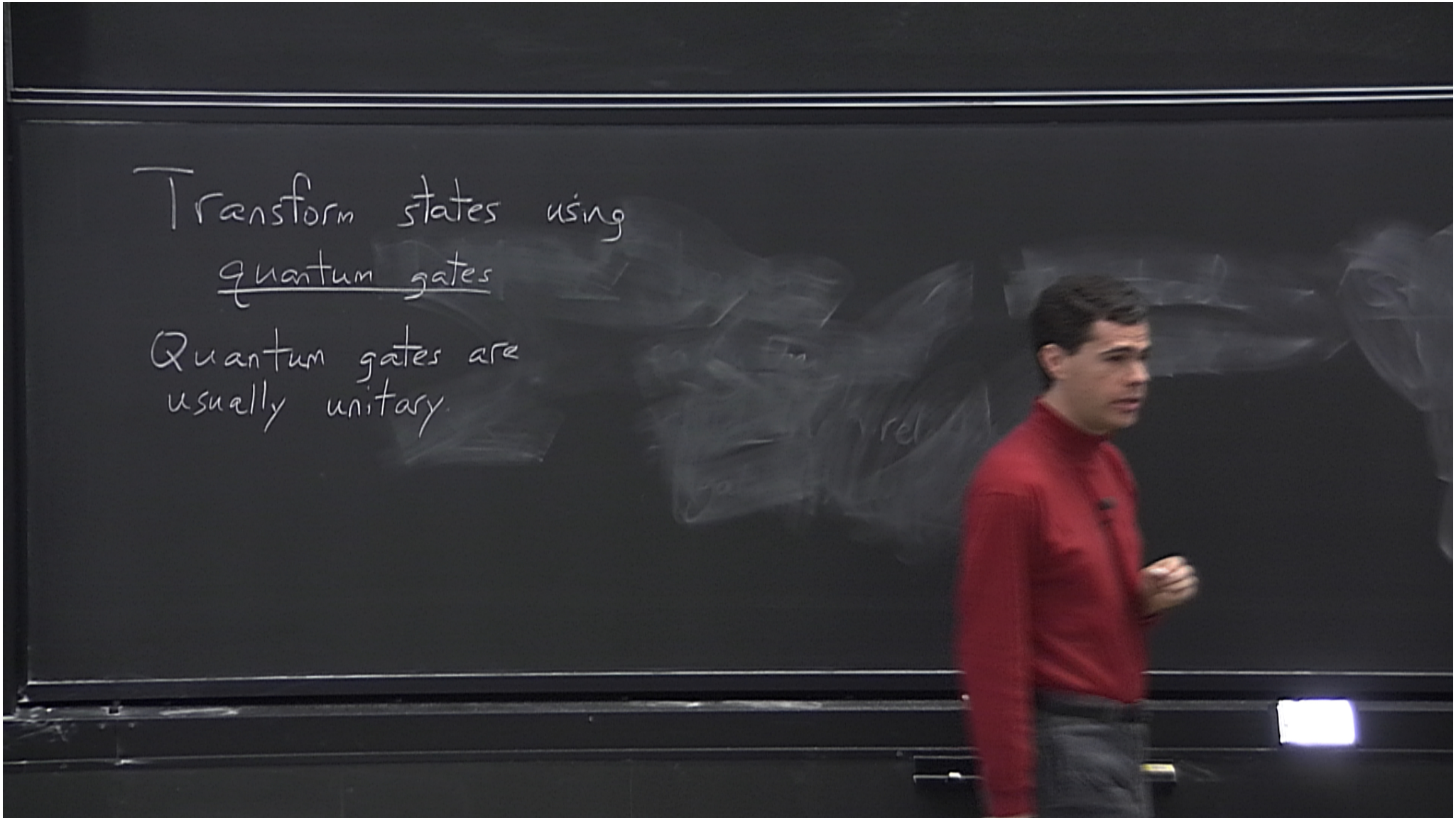

Classical bit 0 or 1
Quantum bit (qubit) $\alpha|0\rangle+\beta|1\rangle$
2-dimensional Hilbert space
n bits – 2^n bit strings
n qubits – 2^n-dimensional Hilb
State $\sum_{x\in\{0,1\}^n} c_x|x\rangle$

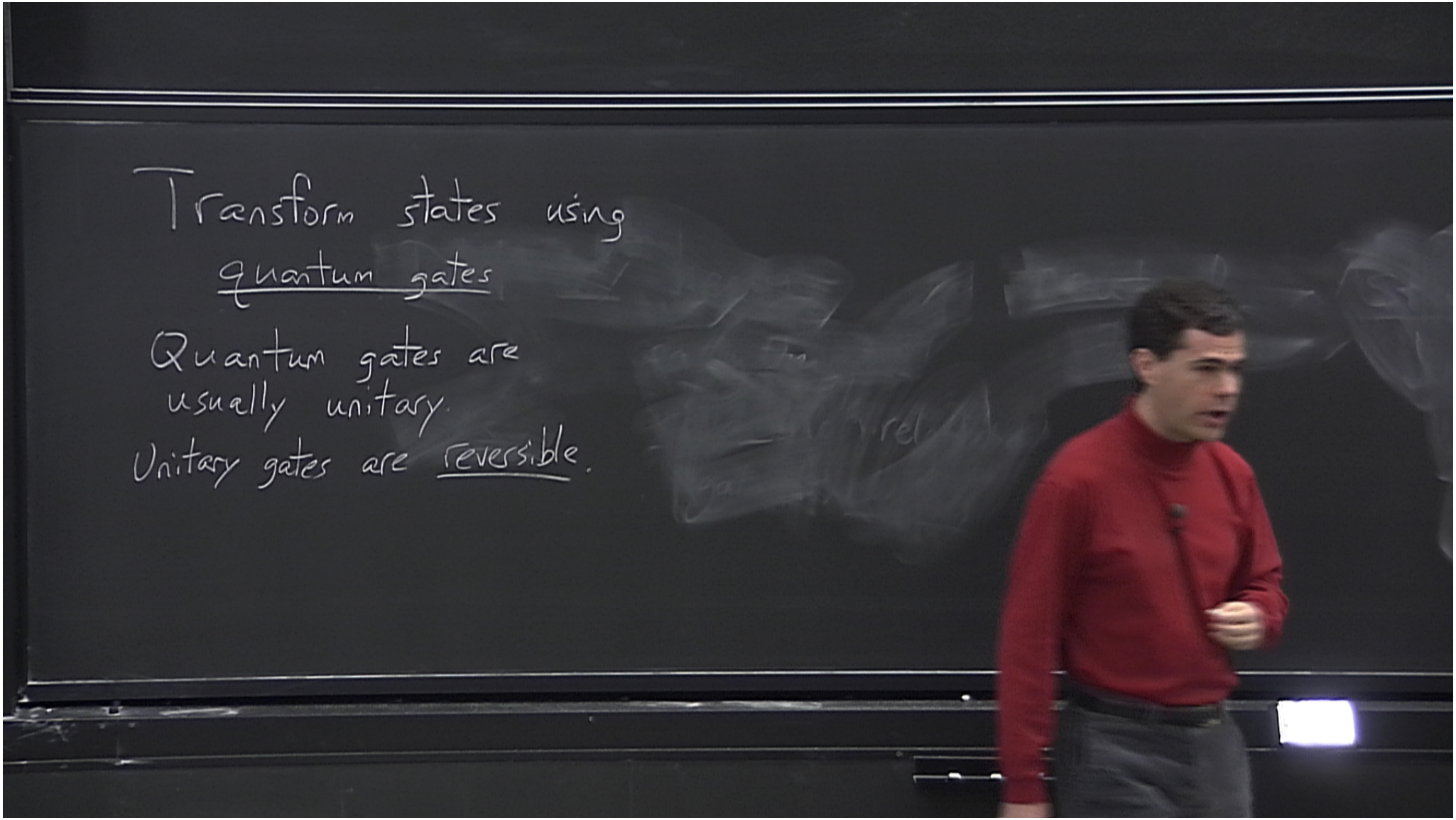

bit strings 00...0, 00...01, ...
ensional Hilbert space
Tensor product states
|x>

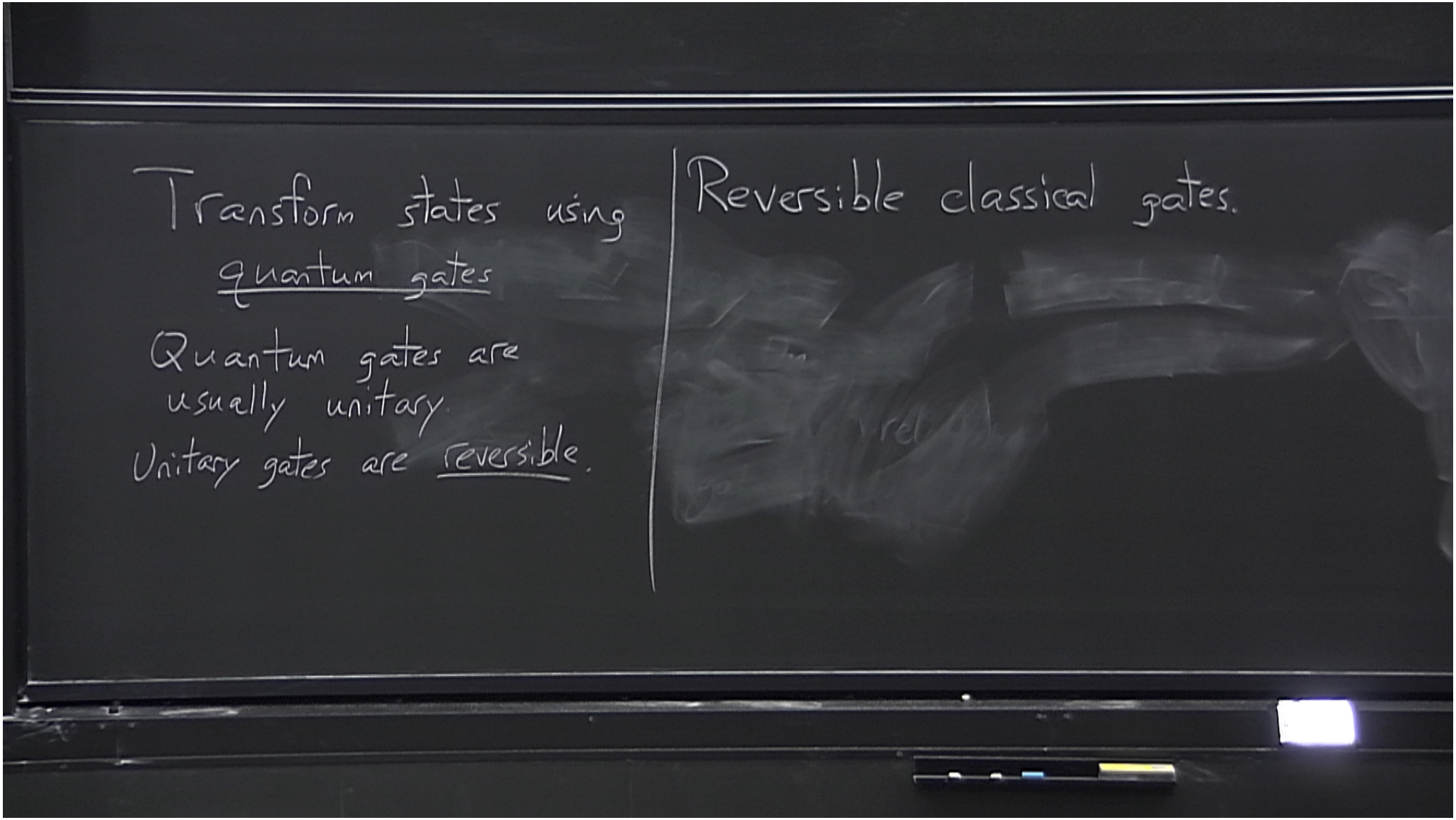

bit strings 00...0, 00...01, ...
ensional Hilbert space
|x>
Tensor product states
(α|0>+β|1>)(γ|0>+δ|1>)

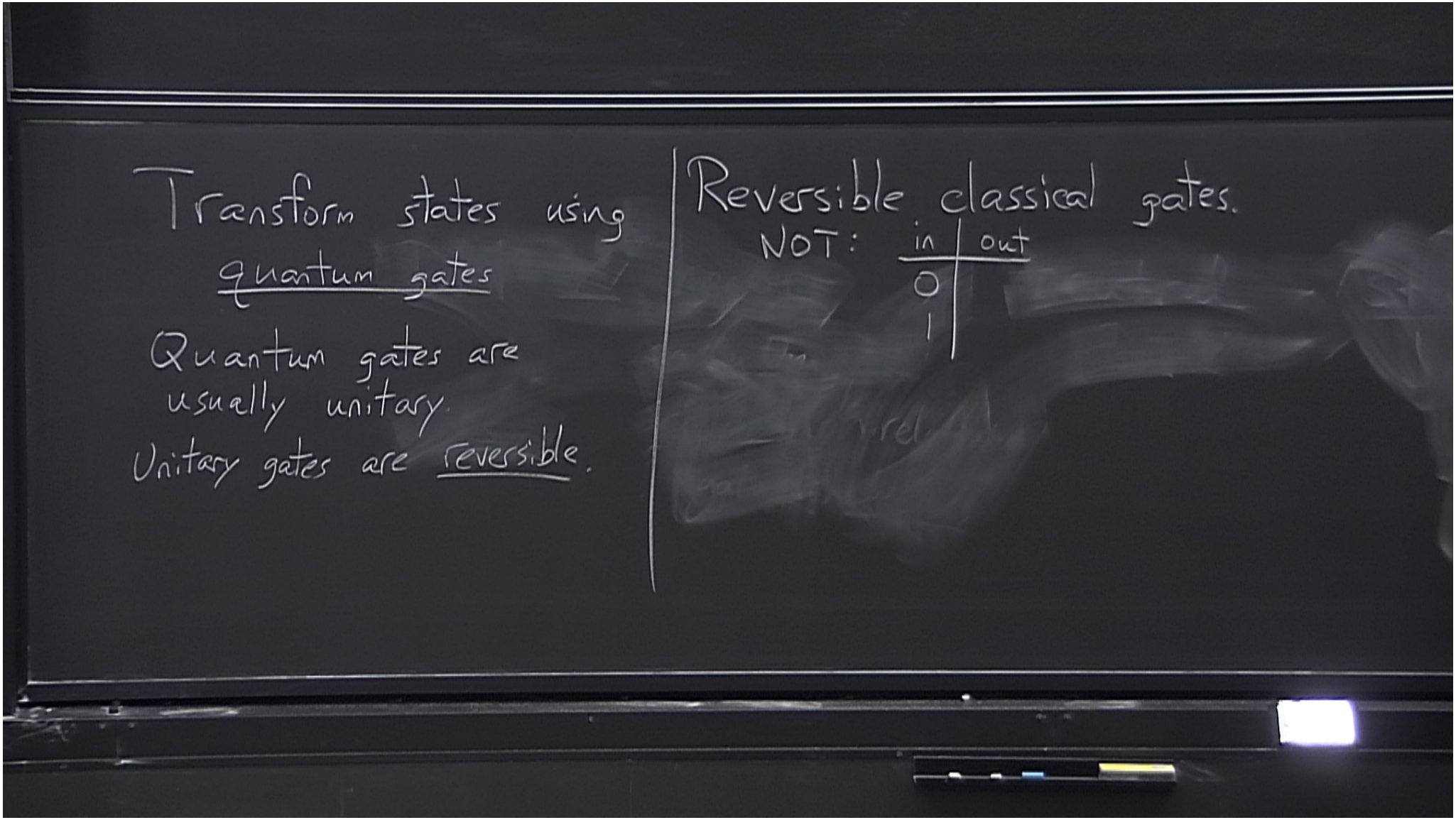

bit strings 00...0, 00...01, ...
ensional Hilbert space
|x>
Tensor product states (α|0>+β|1>)(γ|0>+δ|1>)
Entangled states (|00>+|11>)


Transform states using
quantum gates


Transform states using
quantum gates
Quantum gates are
usually unitary.


Transform states using
quantum gates
Quantum gates are
usually unitary.
Unitary gates are reversible.


Transform states using
quantum gates
Quantum gates are
usually unitary.
Unitary gates are reversible.
Reversible classical gates.


Transform states using
quantum gates
Quantum gates are usually unitary.
Unitary gates are reversible.
Reversible classical gates.
NOT: in | out
0
1

Transform states using quantum gates

Quantum gates are usually unitary.

Unitary gates are reversible.

Reversible classical gates.

NOT:

in	out
0	1
1	0

Transform states using quantum gates

Quantum gates are usually unitary.

Unitary gates are reversible.

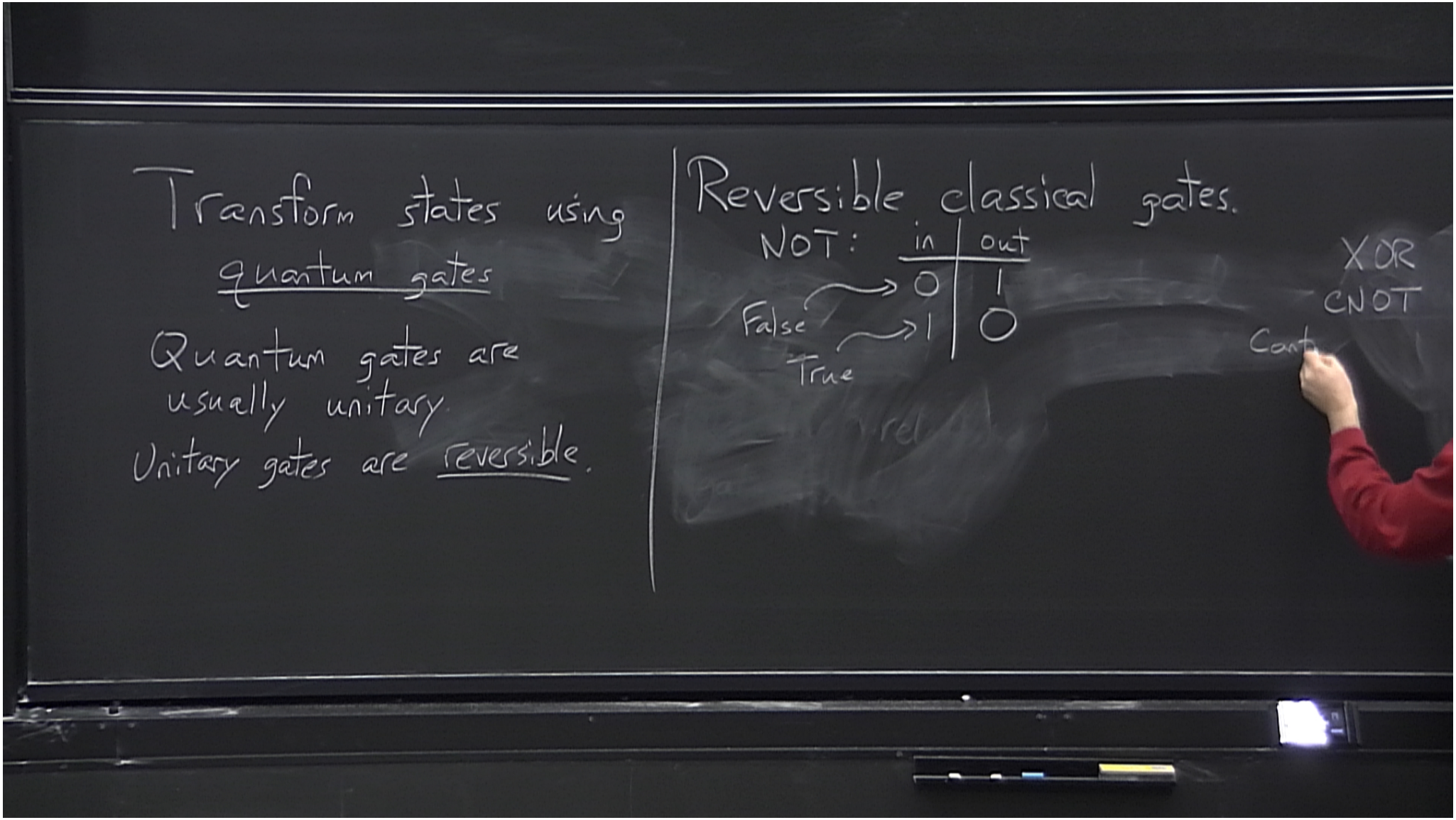
Reversible classical gates.

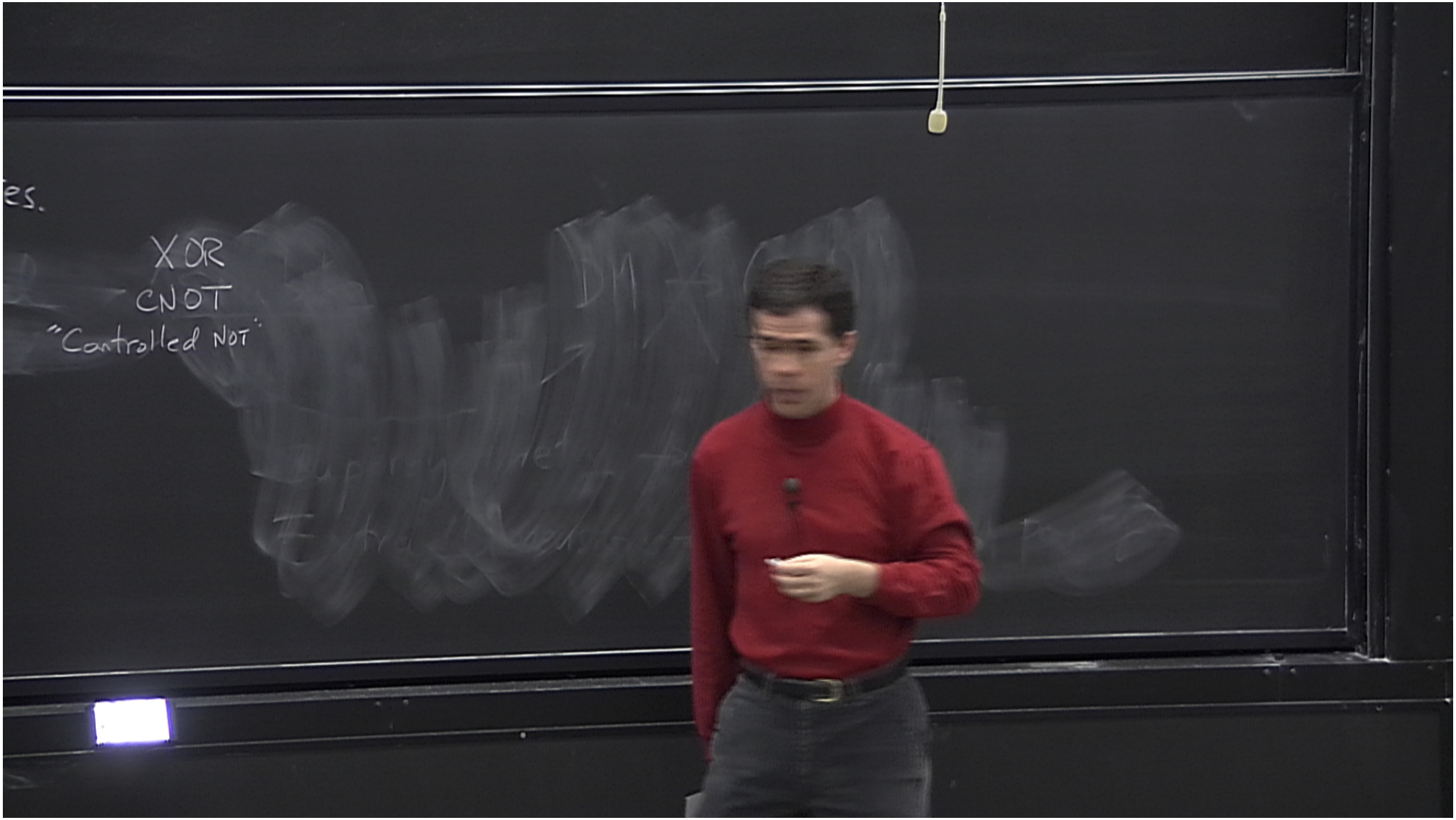
NOT:

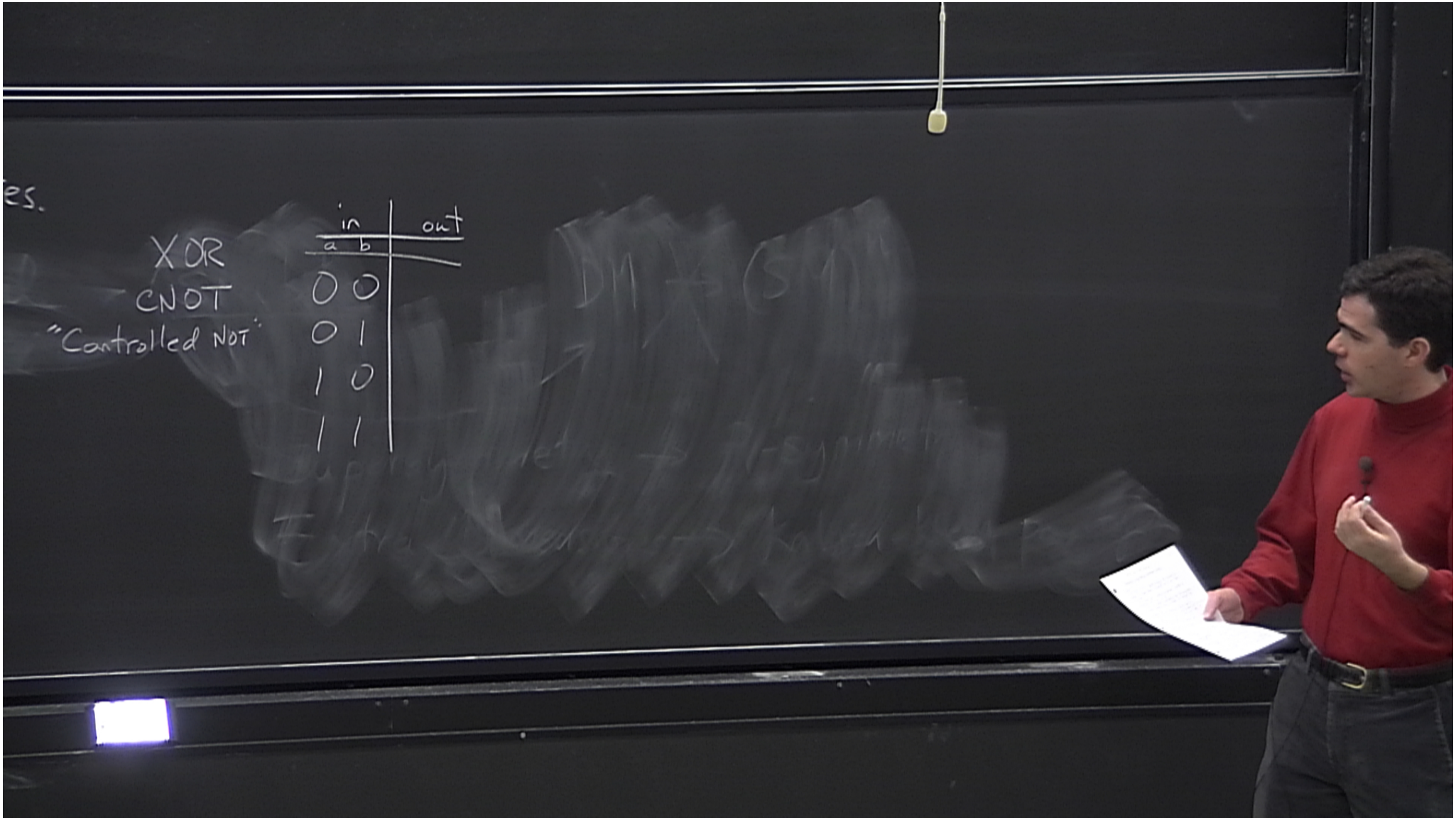
in	out
0	1
1	0

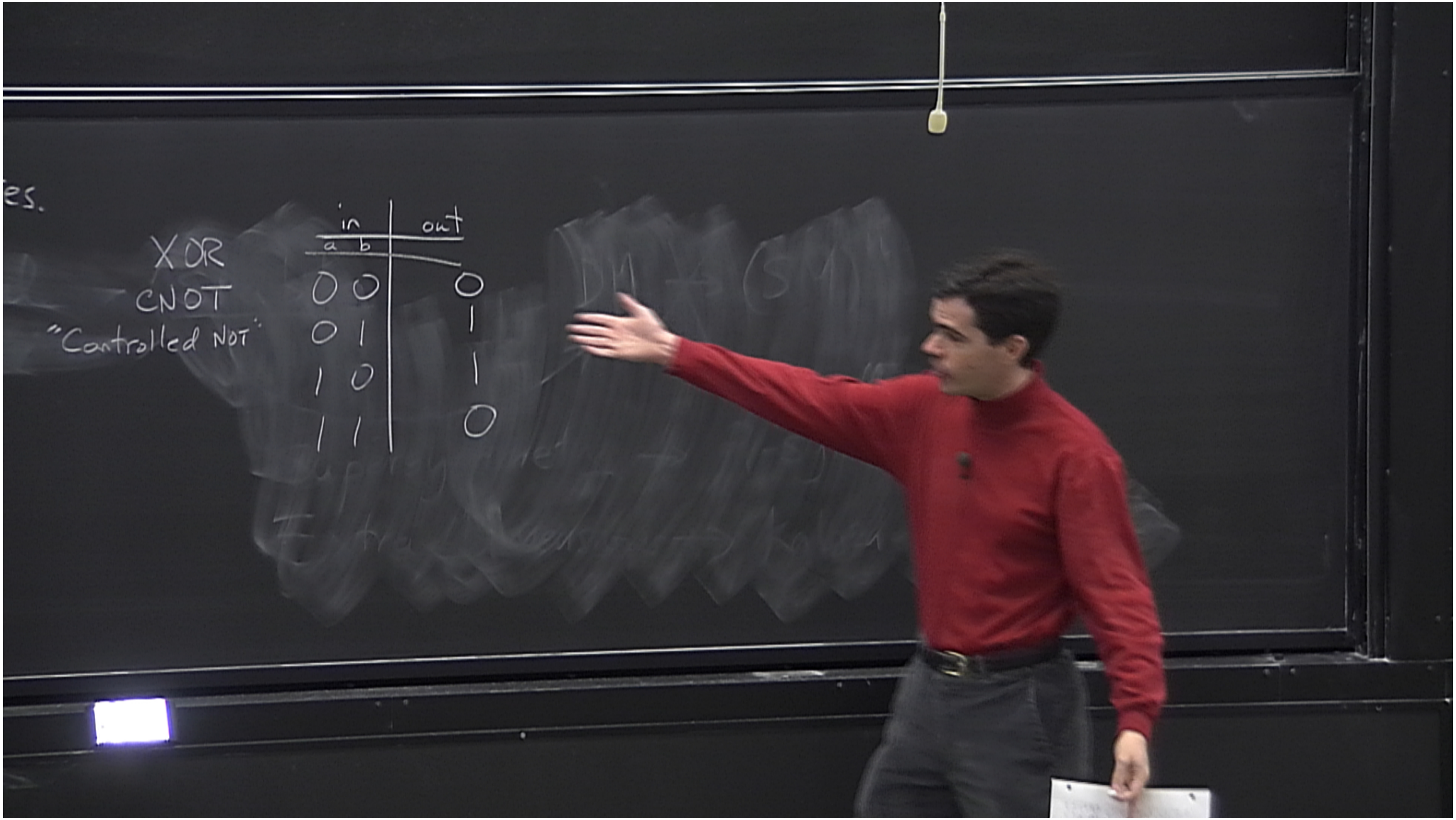
False

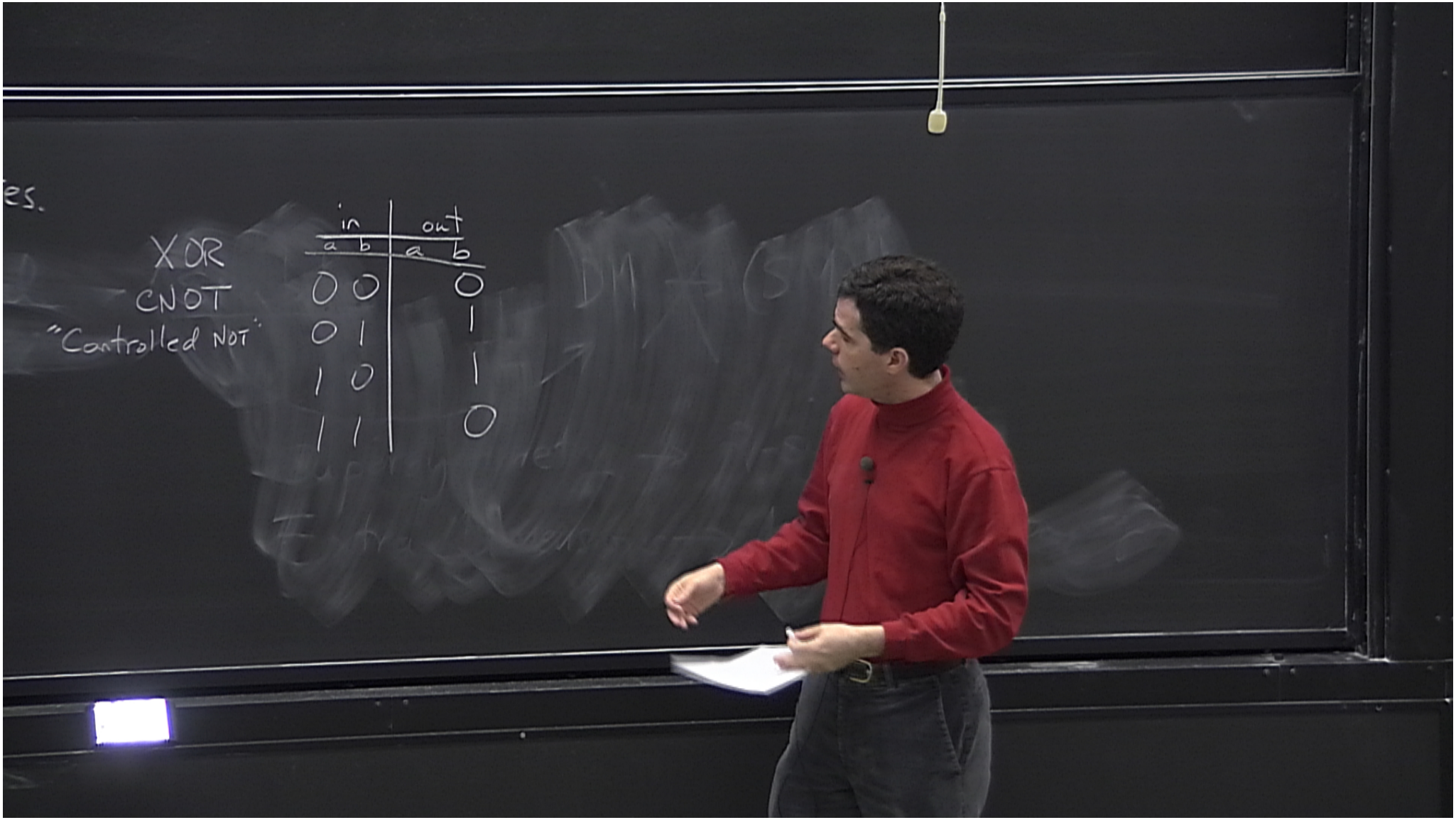
True

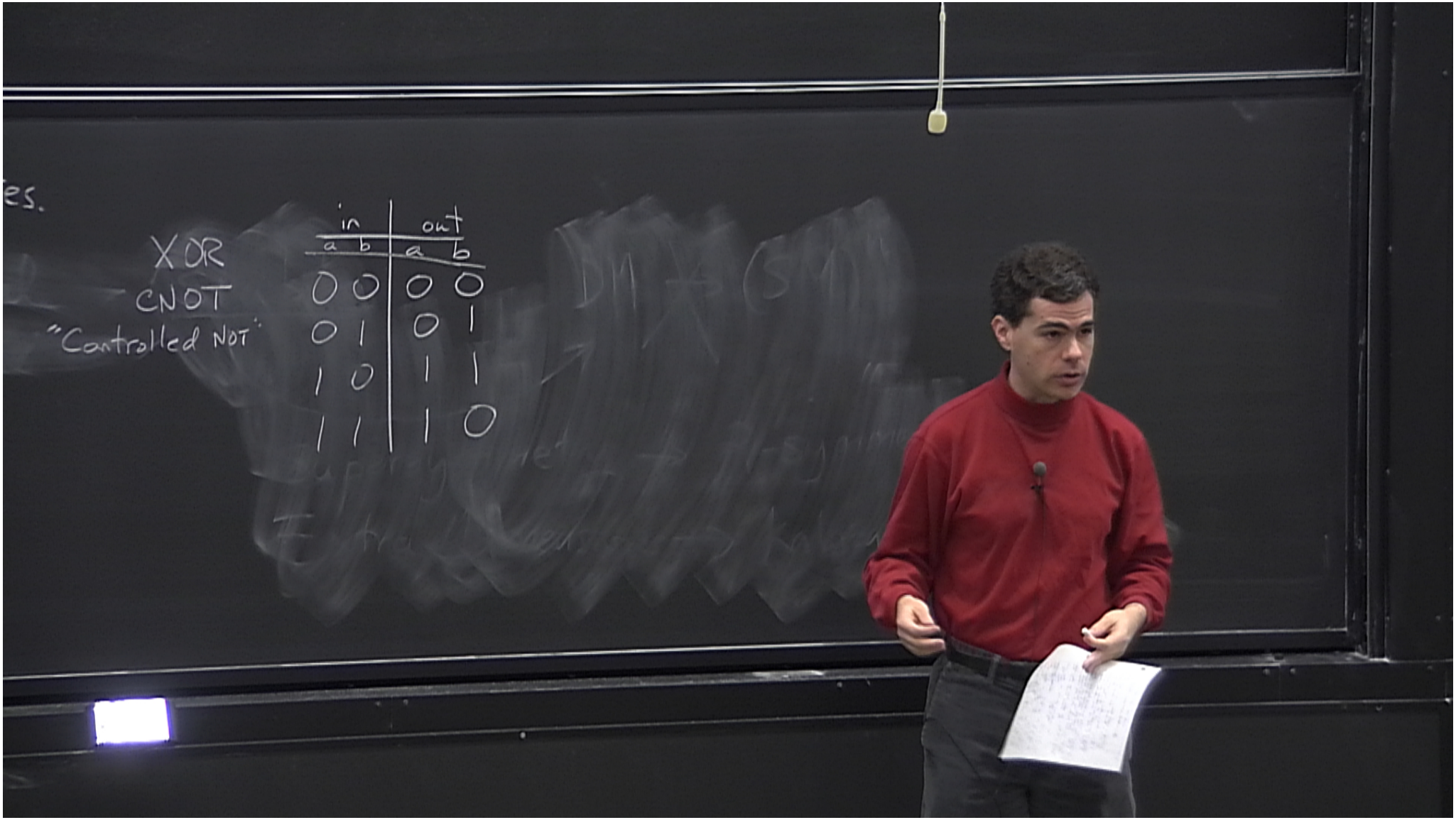

Transform states using
quantum gates
Quantum gates are usually unitary.
Unitary gates are reversible.
Reversible classical gates.
NOT:
in
out
0
1
False
1
0
True
XOR
CNOT

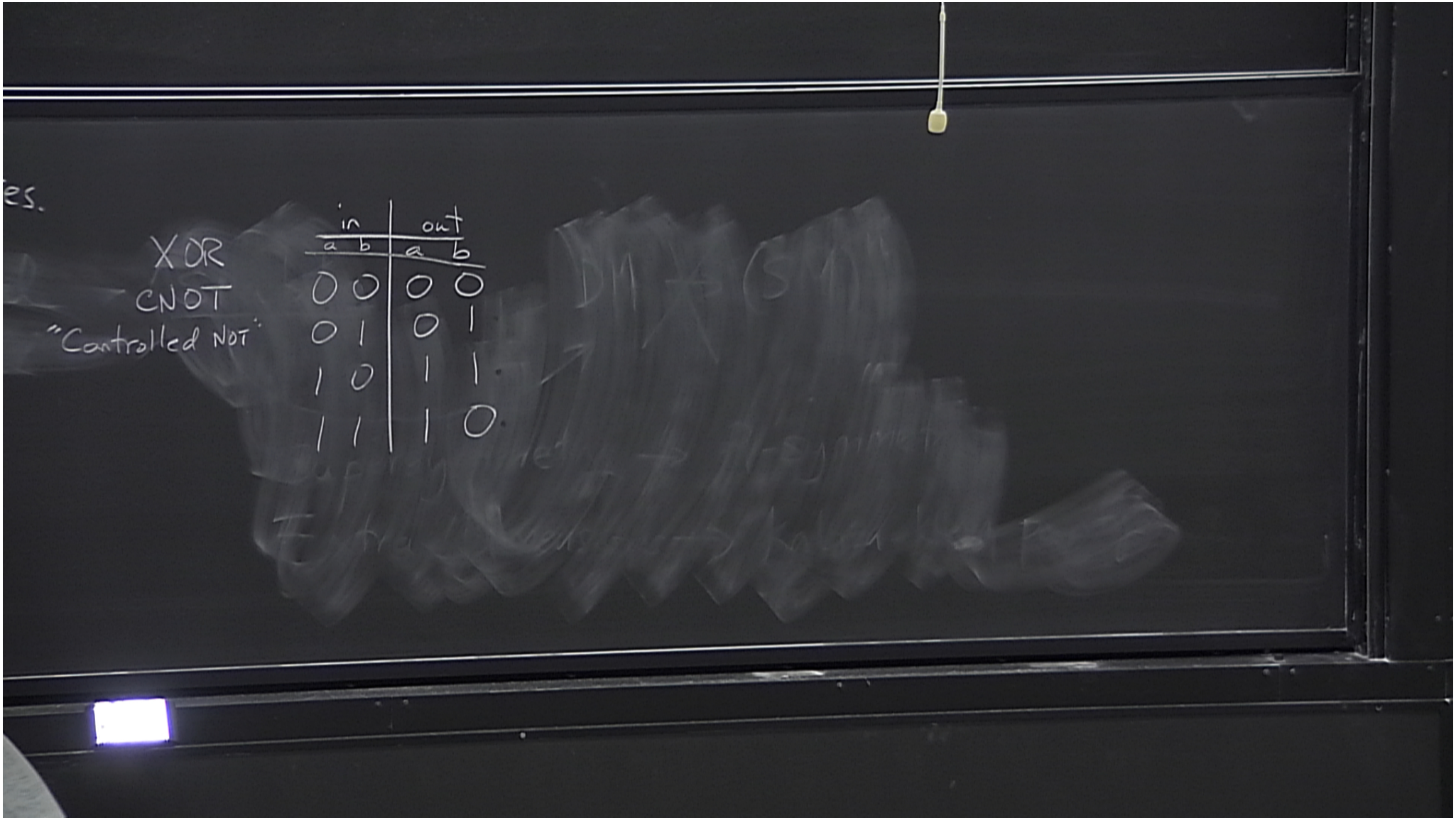

es.
XOR
CNOT
"Controlled NOT"


es.
XOR
CNOT
"Controlled NOT"
in
out
a b
0 0
0 1
1 0
1 1


es.
XOR
CNOT
"Controlled NOT"
in
out
a b
0 0 0
0 1 1
1 0 1
1 1 0


es.
XOR
CNOT
"Controlled NOT"
in
out
a b
a b
0 0
0
0 1
1
1 0
1
1 1
0


es.
XOR
CNOT
"Controlled NOT"
in
out
a b
a b
0 0 0 0
0 1 0 1
1 0 1 1
1 1 1 0


es.
XOR
CNOT
"Controlled NOT"
in
out
a b
a b
0 0
0 0
0 1
0 1
1 0
1 1
1 1
1 0

Transform states using quantum gates

Quantum gates are usually unitary.

Unitary gates are reversible.

Reversible classical gates.

NOT:

in	out
0	1
1	0

False → 0

True → 1

XOR

CNOT

"Controlled NOT"

Circuit diagrams

Transform states using quantum gates

Quantum gates are usually unitary.

Unitary gates are reversible.

Reversible classical gates.

NOT:

in	out
0	1
1	0

False

True



XOR

CNOT

"Controlled NOT"

Circuit diagrams

Transform states using quantum gates

Quantum gates are usually unitary.

Unitary gates are reversible.

Reversible classical gates.

NOT:

in	out
0	1
1	0

False

True



XOR

CNOT

"Controlled NOT"

Circuit diagrams:

Each line is a bit (or qubit)

Transform states using quantum gates

Quantum gates are usually unitary.

Unitary gates are reversible.

Reversible classical gates.

NOT:

in	out
0	1
1	0

False

True

XOR

CNOT

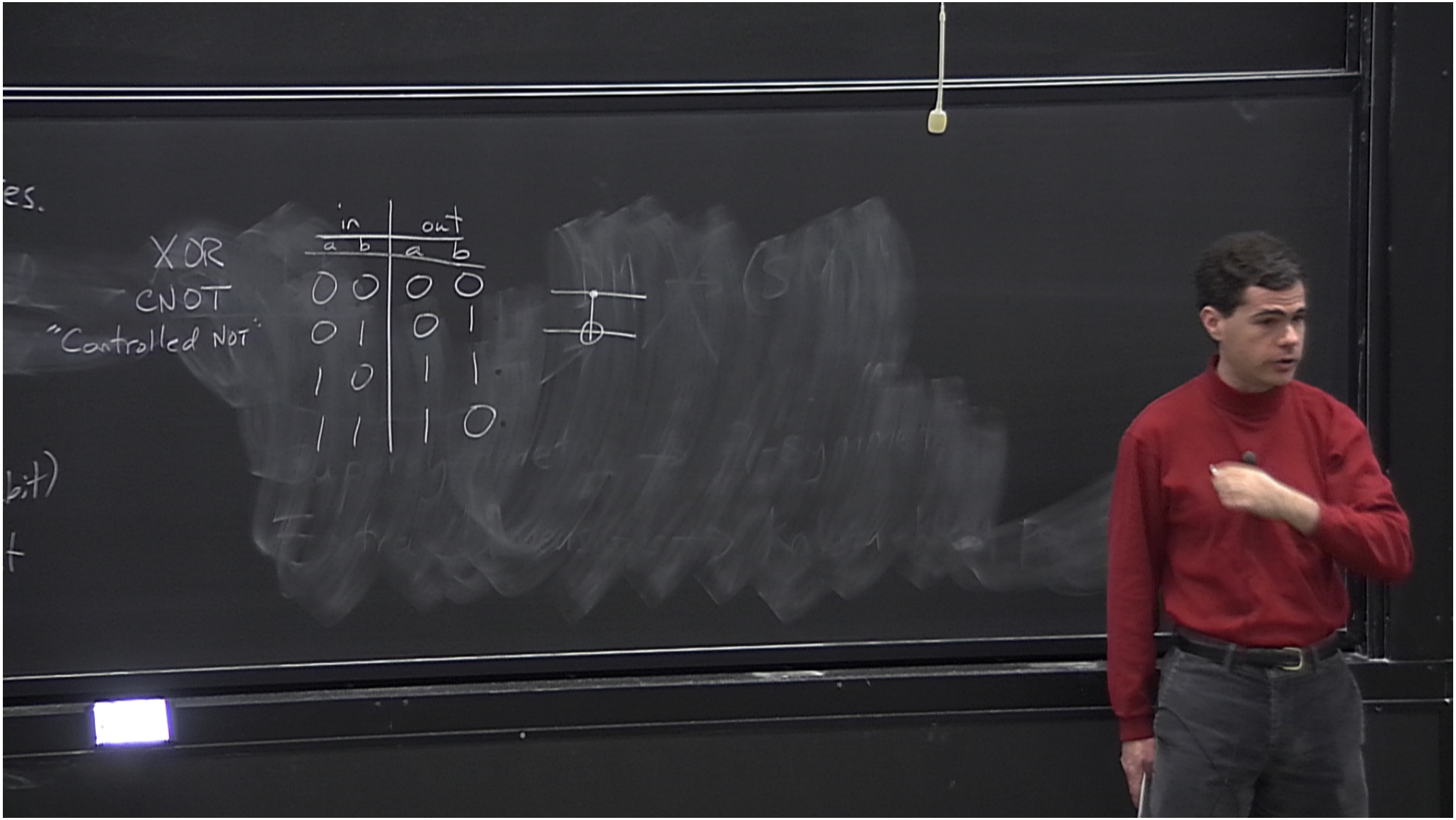
"Controlled NOT"

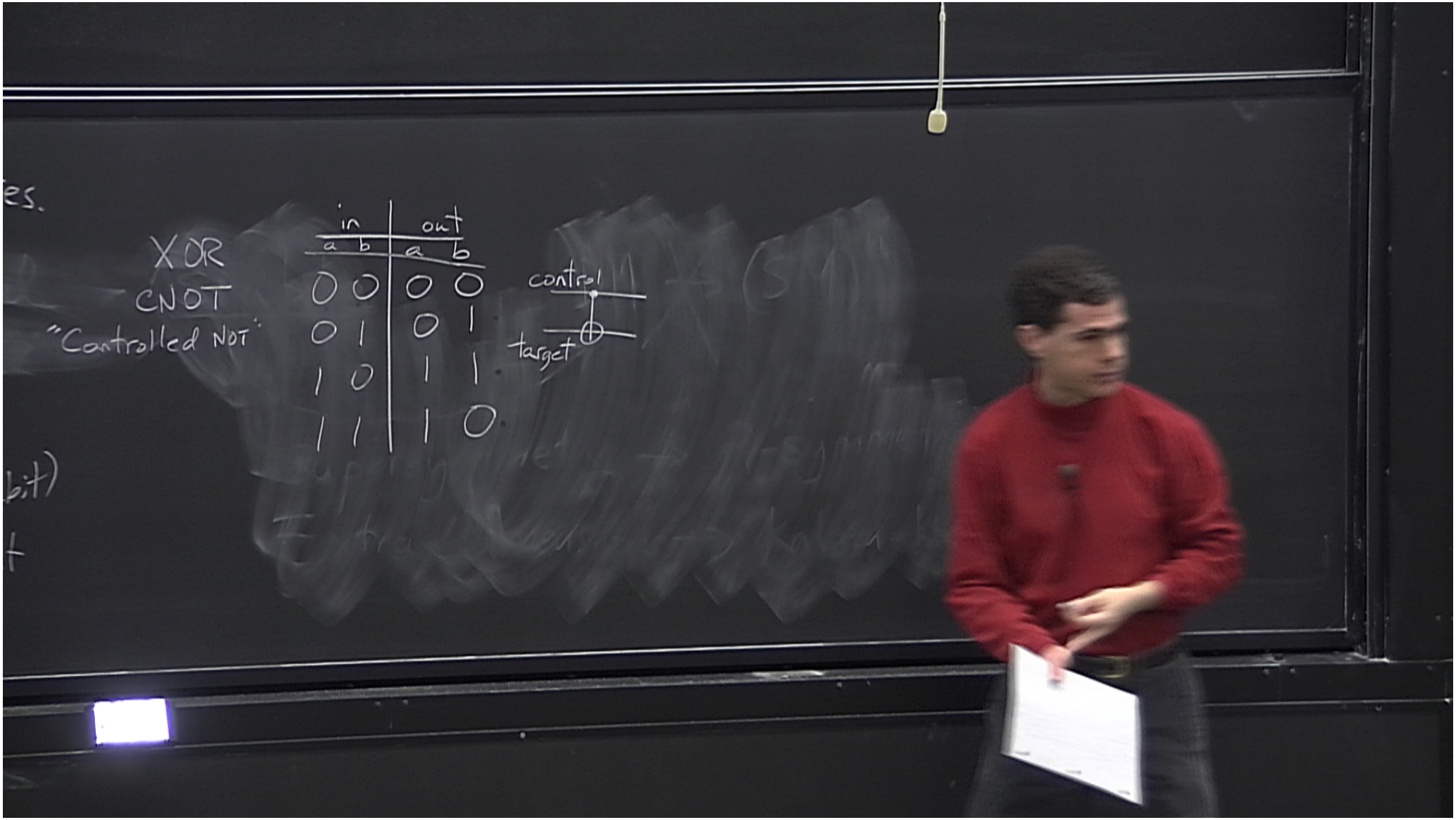
Circuit diagrams:

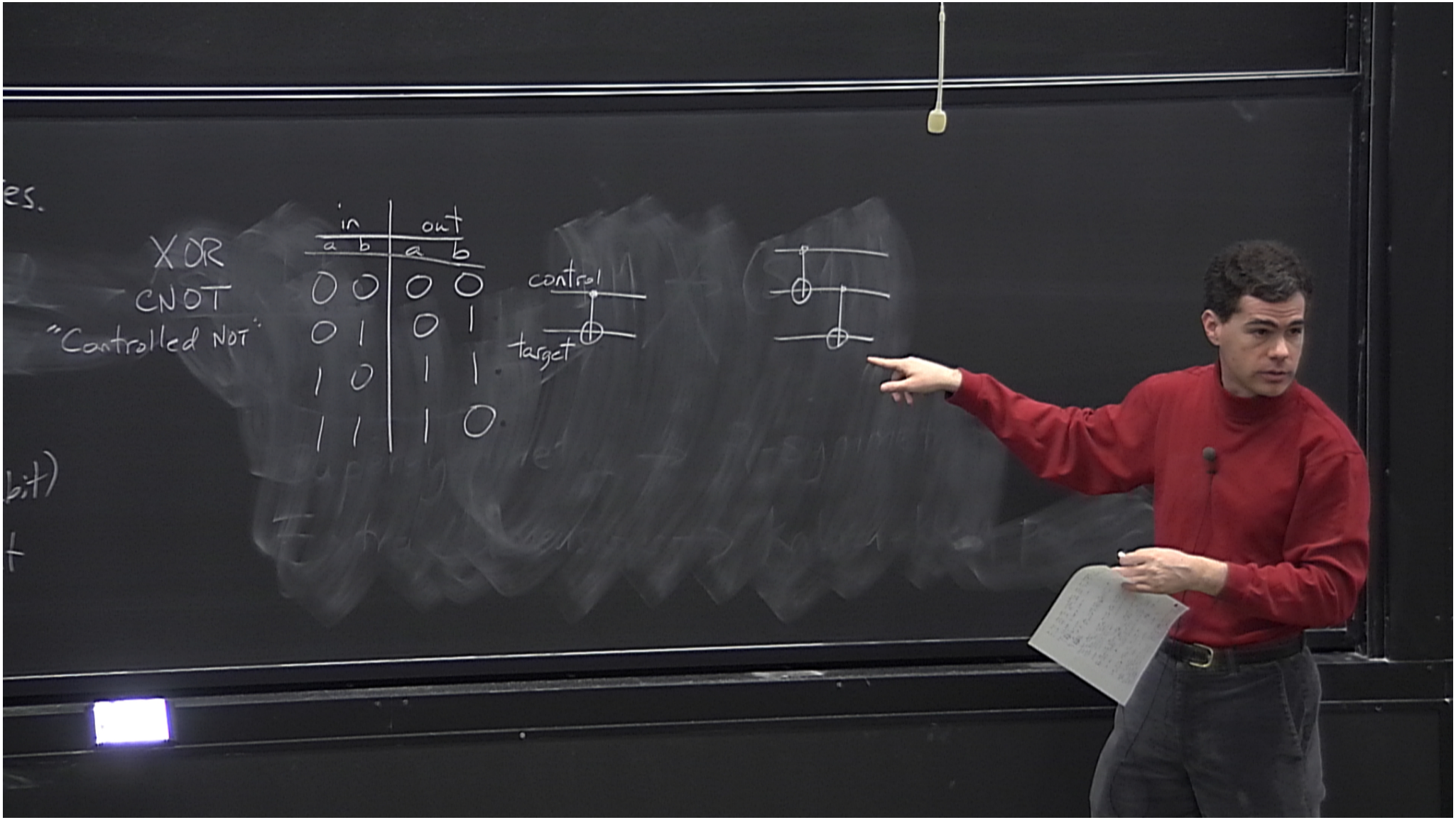
Each line is a bit (or qubit)

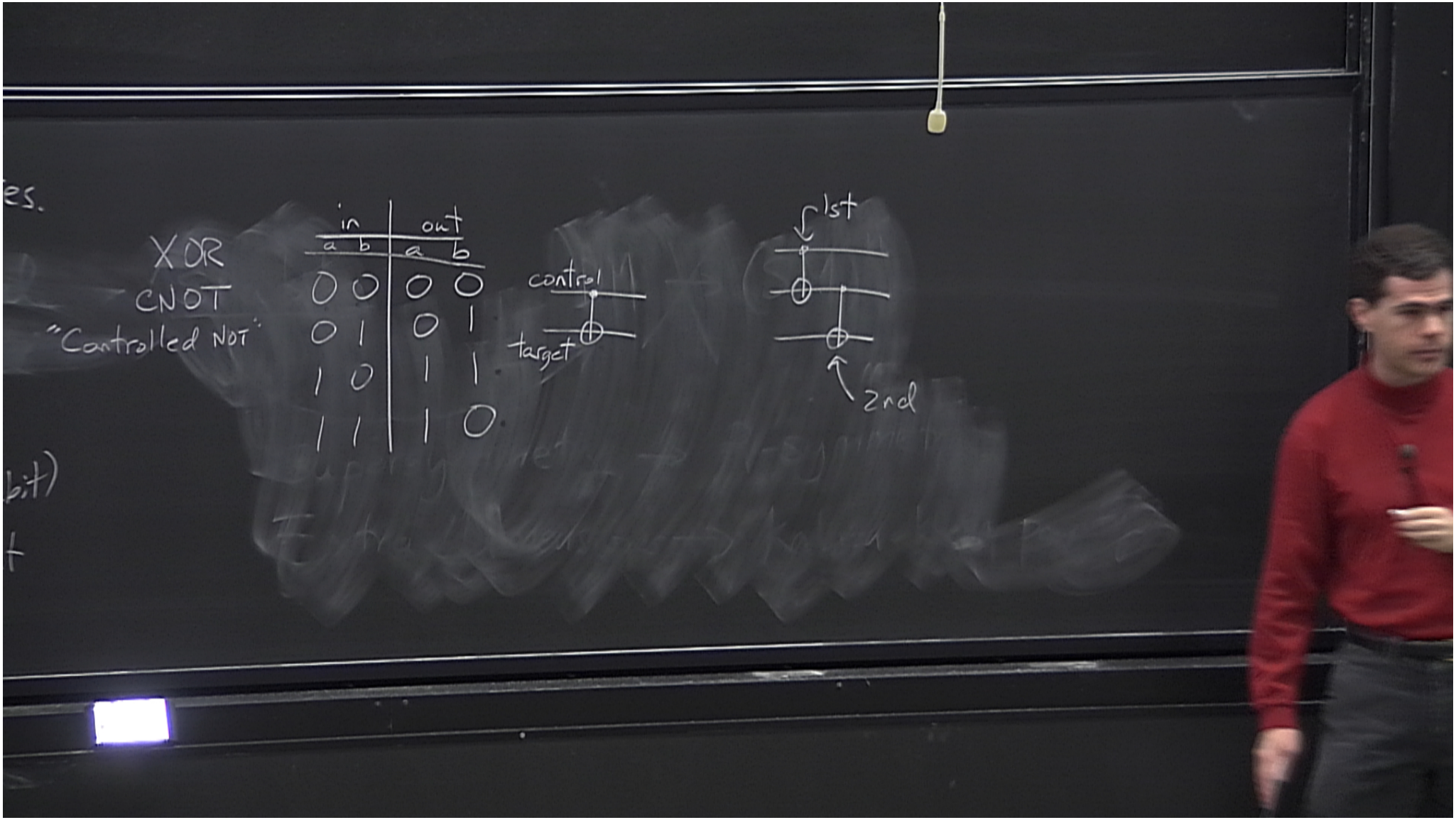
Time goes from left to right

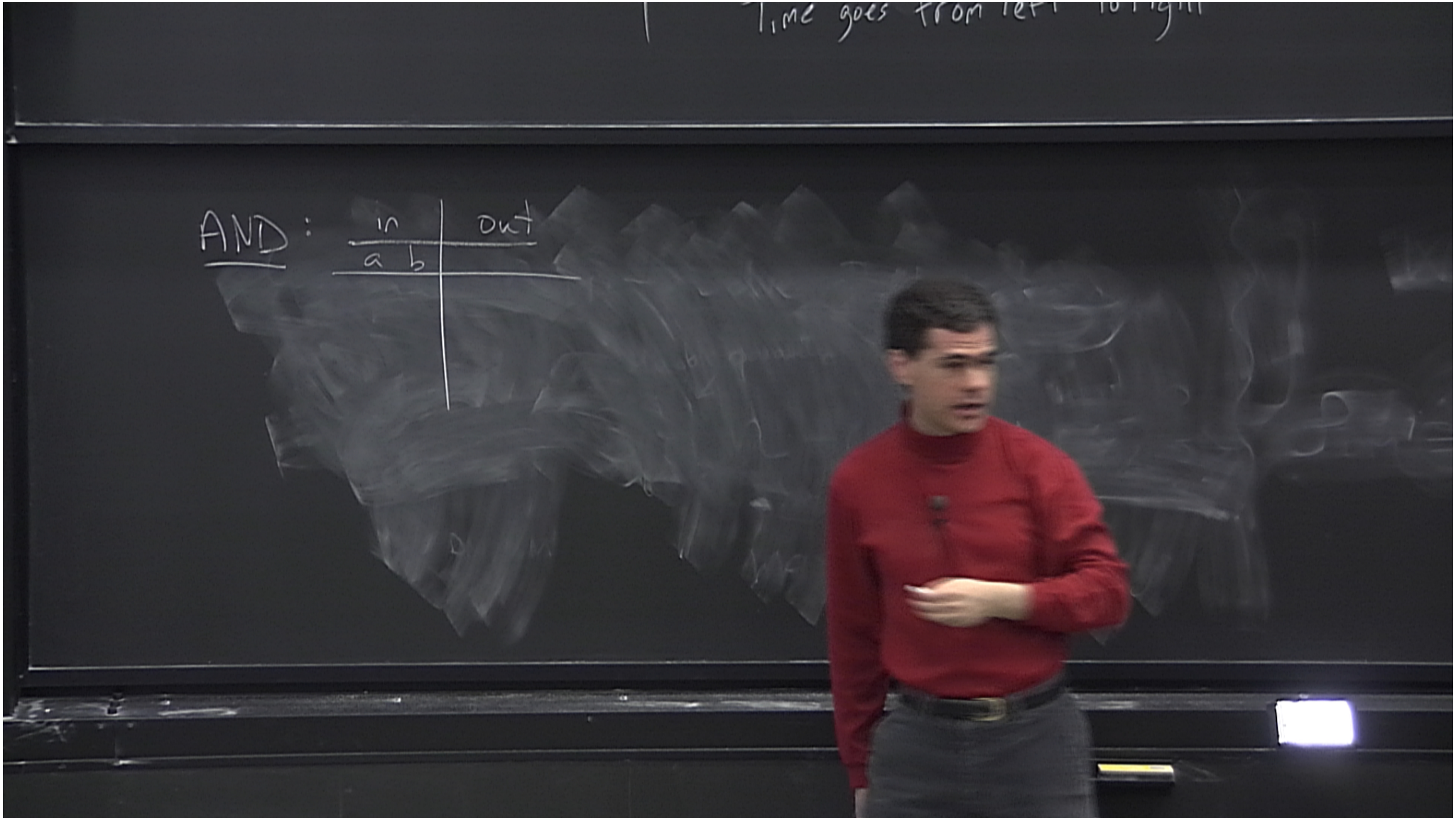

es.
XOR
CNOT
"Controlled NOT"
in
out
a b
a b
0 0
0 0
0 1
0 1
1 0
1 1
1 1
1 0
bit)
t

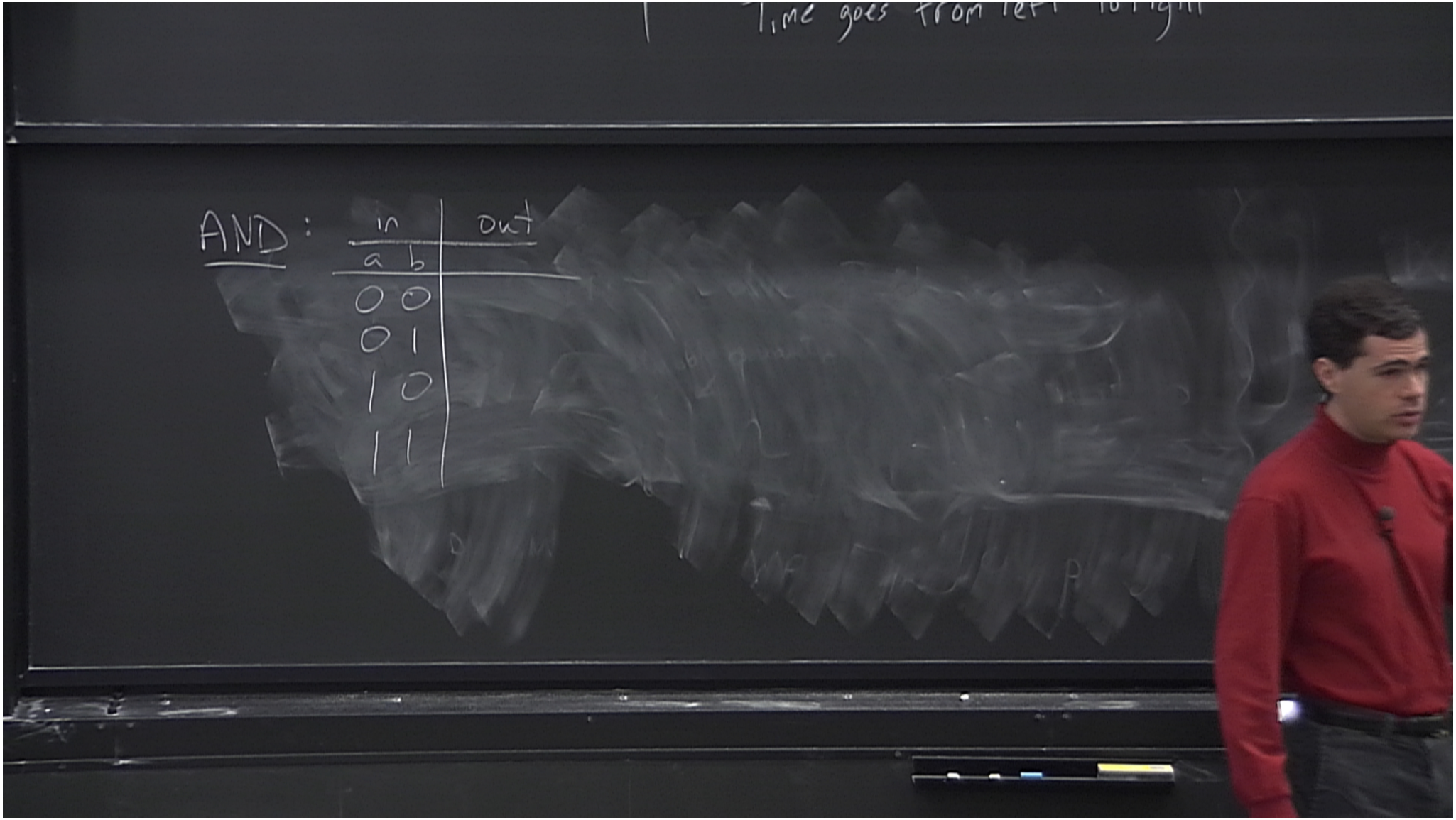

es.
XOR
CNOT
"Controlled NOT"
in
out
a b
a b
0 0
0 0
0 1
0 1
1 0
1 1
1 1
1 0
bit)
t

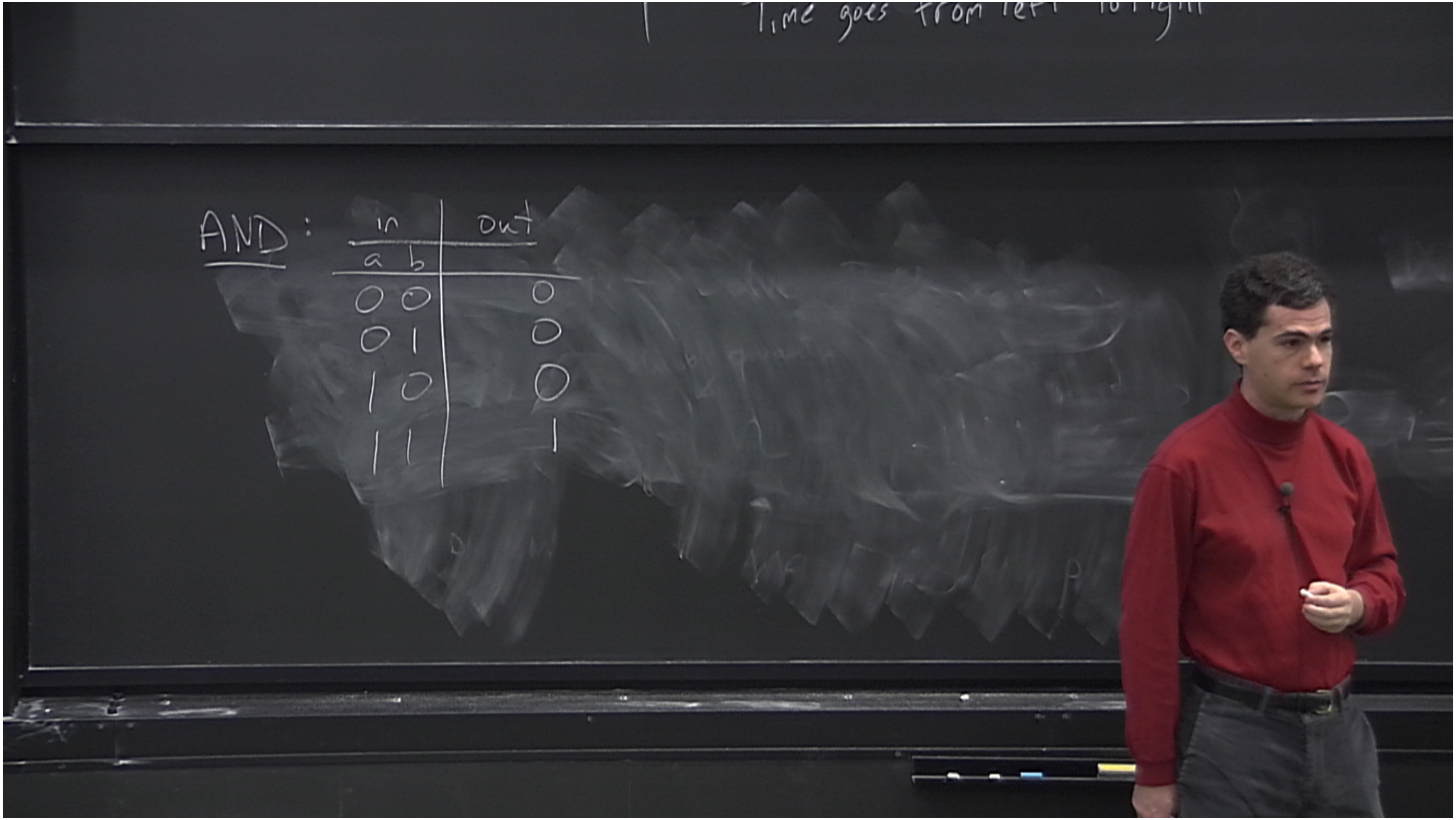

es.
XOR
CNOT
"Controlled NOT"
in
out
a b
a b
0 0 0 0
0 1 0 1
1 0 1 1
1 1 1 0
control
target
bit)
t

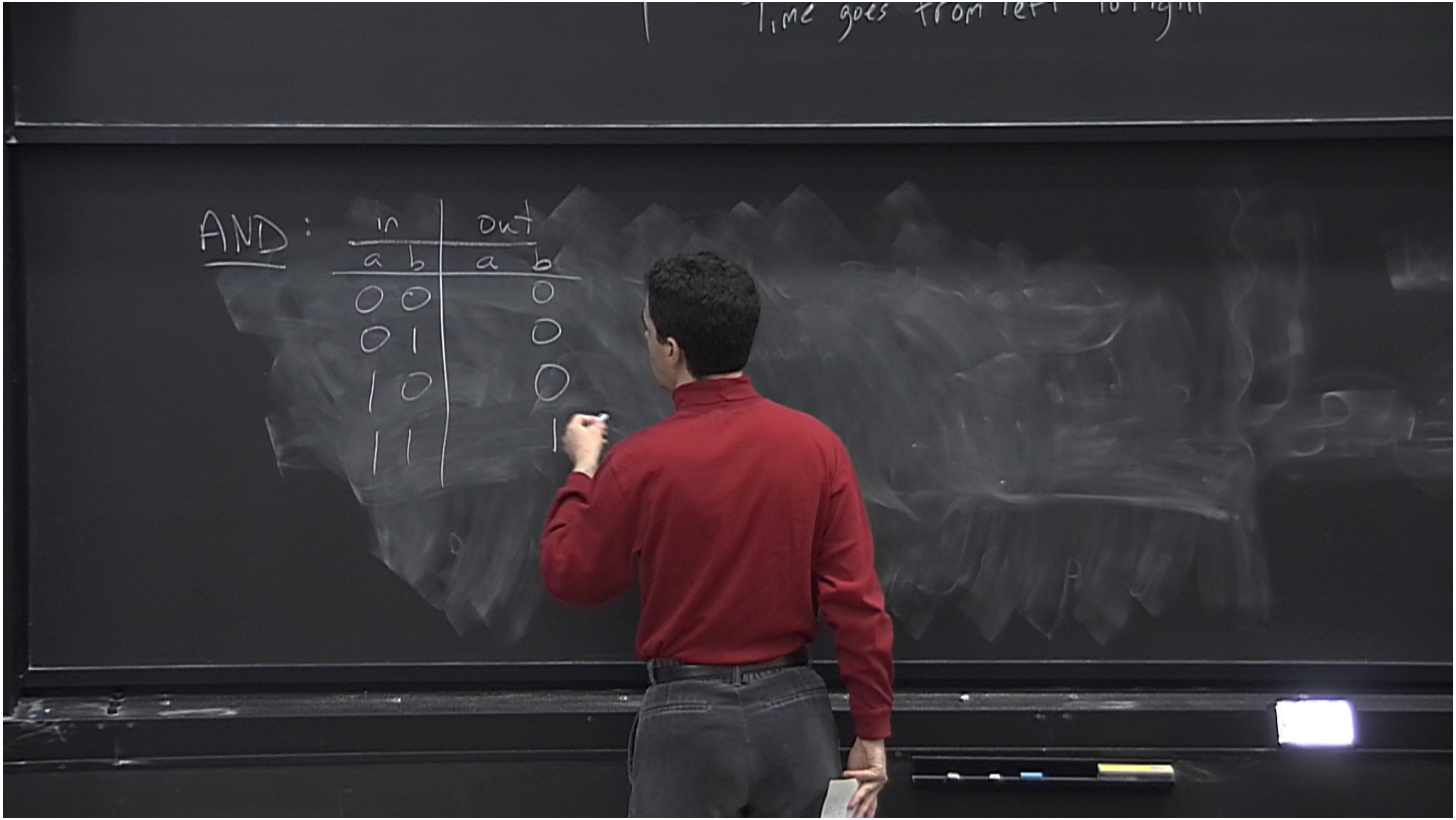

es.
XOR
CNOT
"Controlled NOT"
in
out
a b
a b
0 0 0 0
0 1 0 1
1 0 1 1
1 1 1 0
control
target
bit)
t

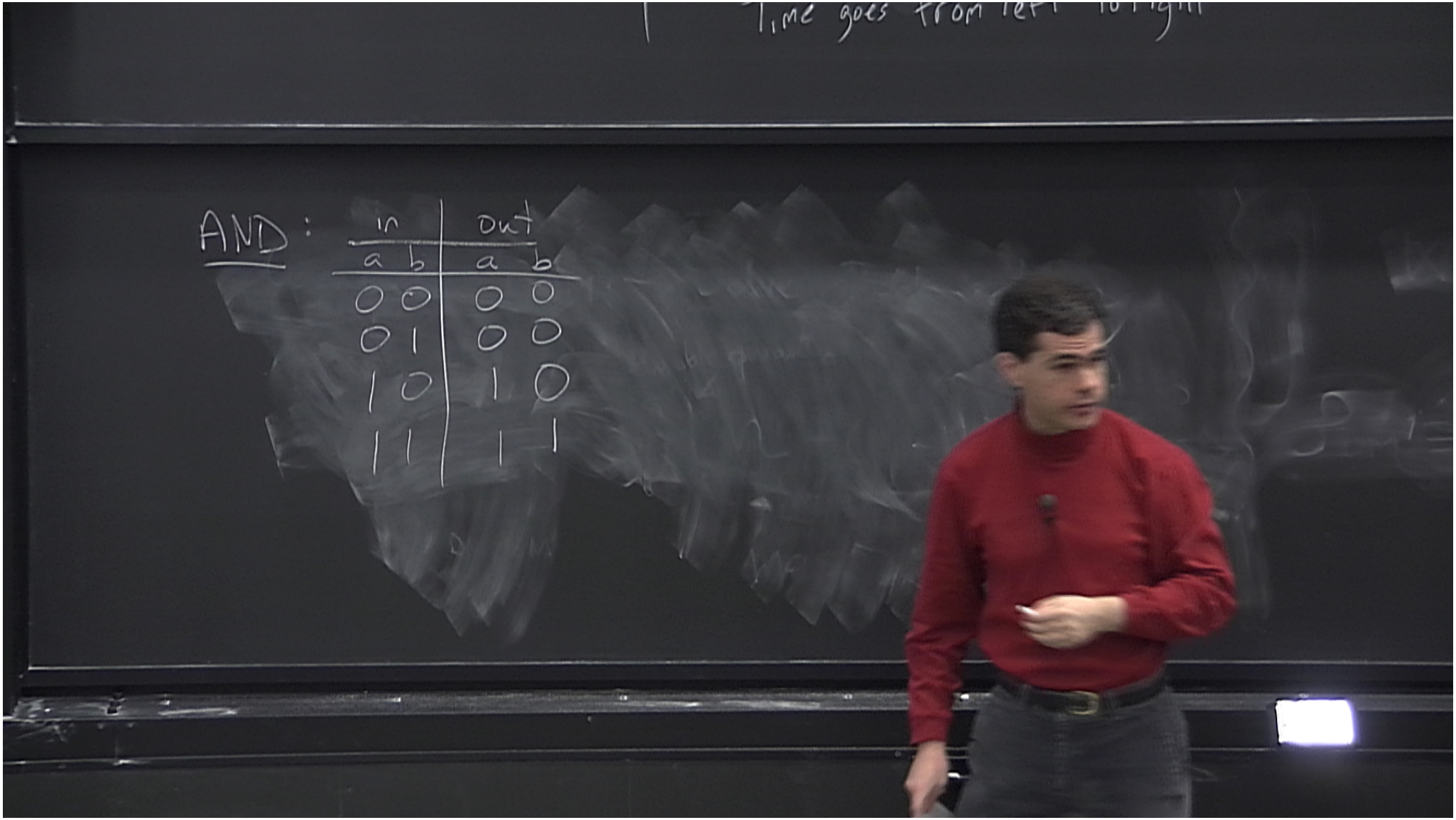

es.
XOR
CNOT
"Controlled NOT"
in
out
a b
a b
0 0 0 0
0 1 0 1
1 0 1 1
1 1 1 0
control
target
1st
2nd
bit)
t

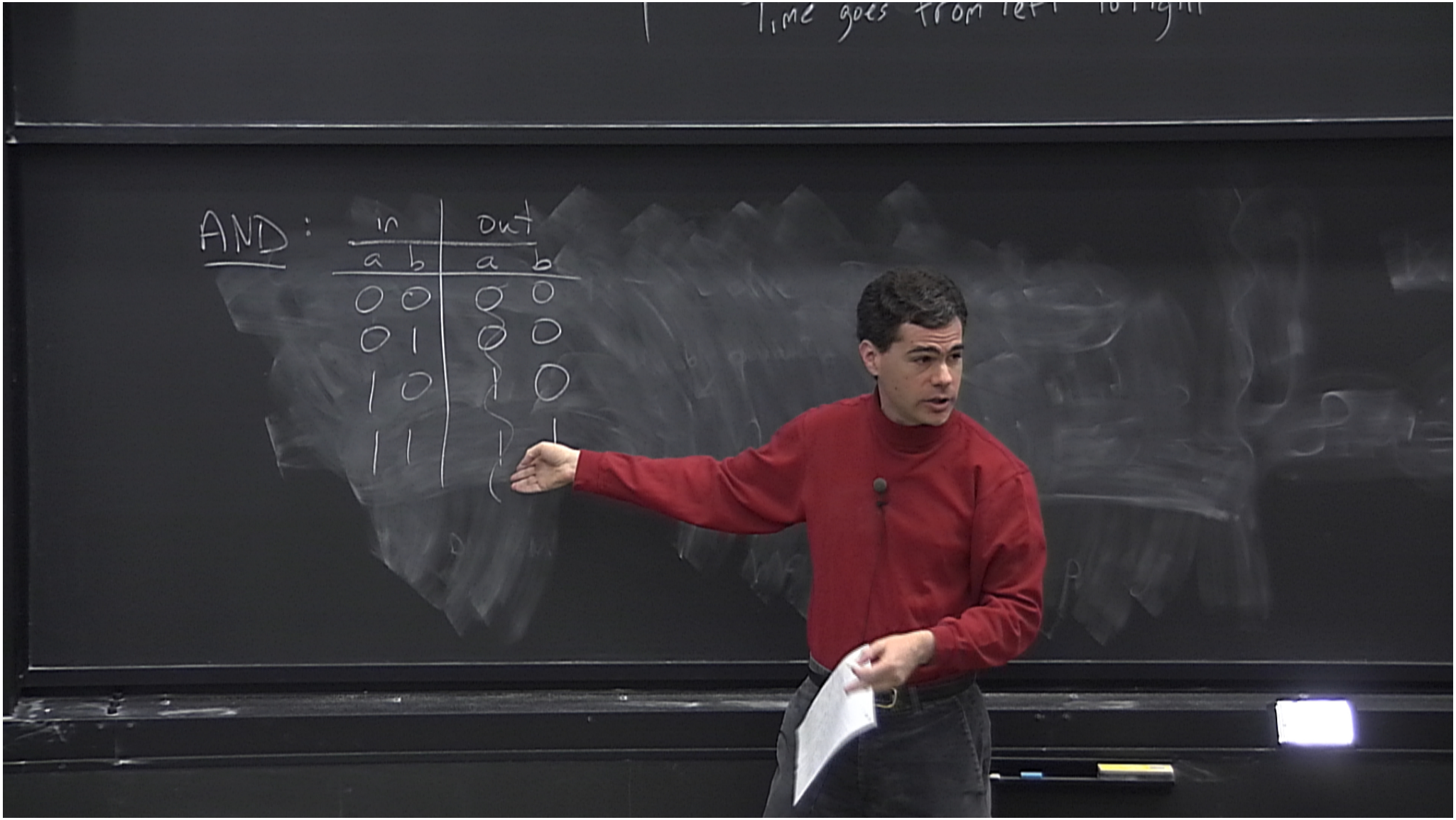

Time goes from left to right
AND :
in
out
a b

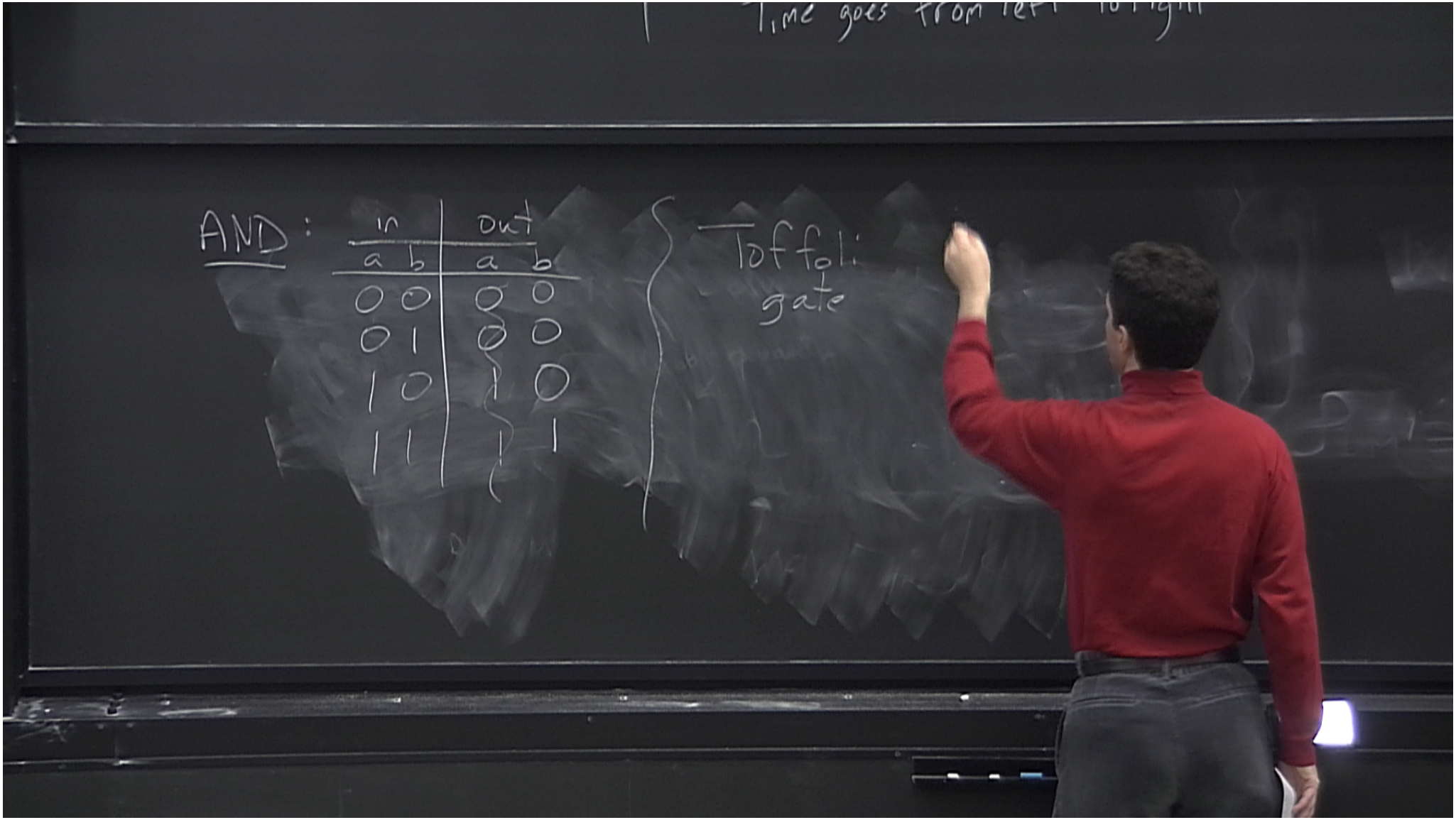

Time goes from left to right
AND :
in
a b
out
0 0
0 1
1 0
1 1

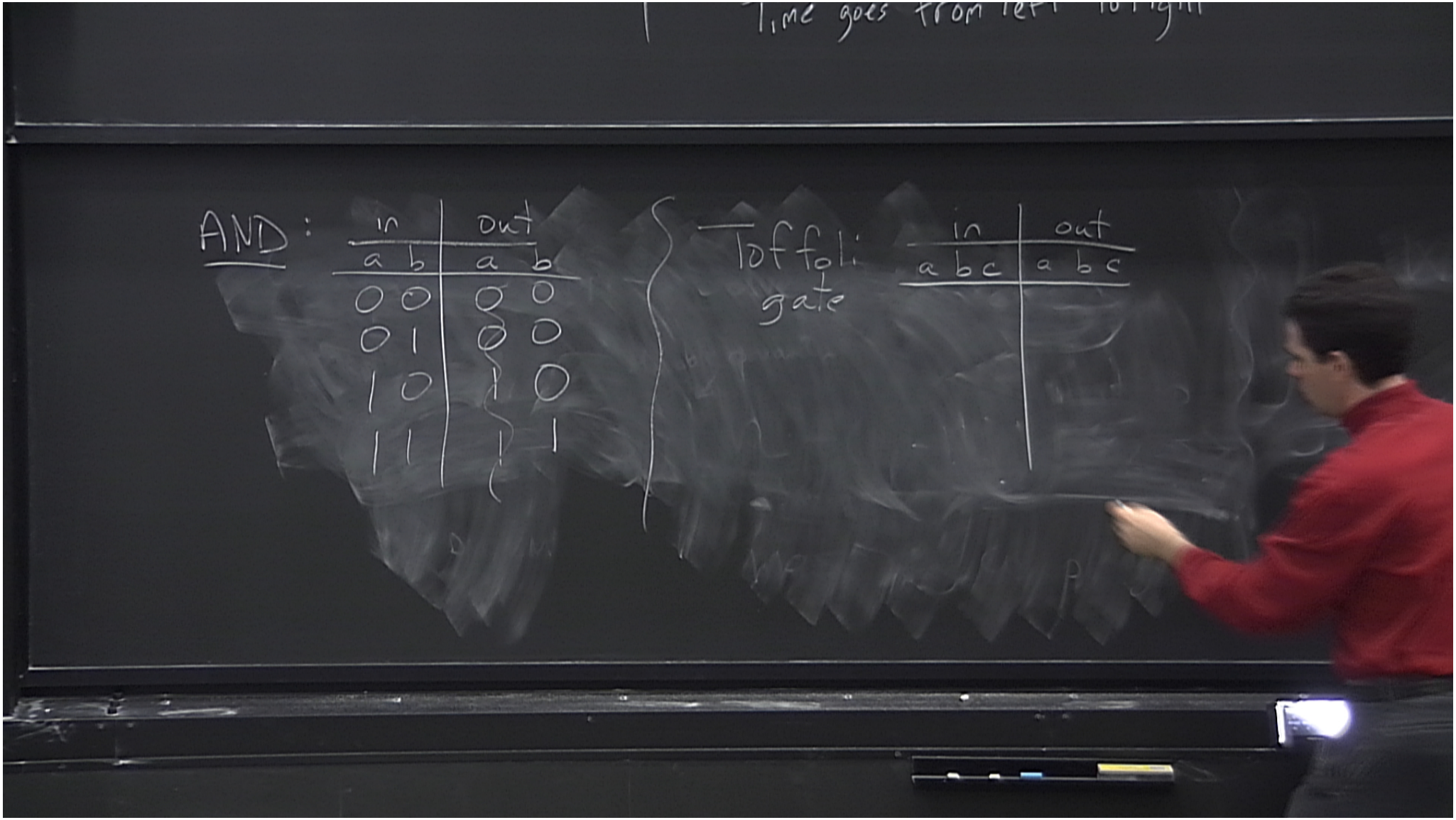

Time goes from left to right
AND:
in
a b
out
0 0 0
0 1 0
1 0 0
1 1 1

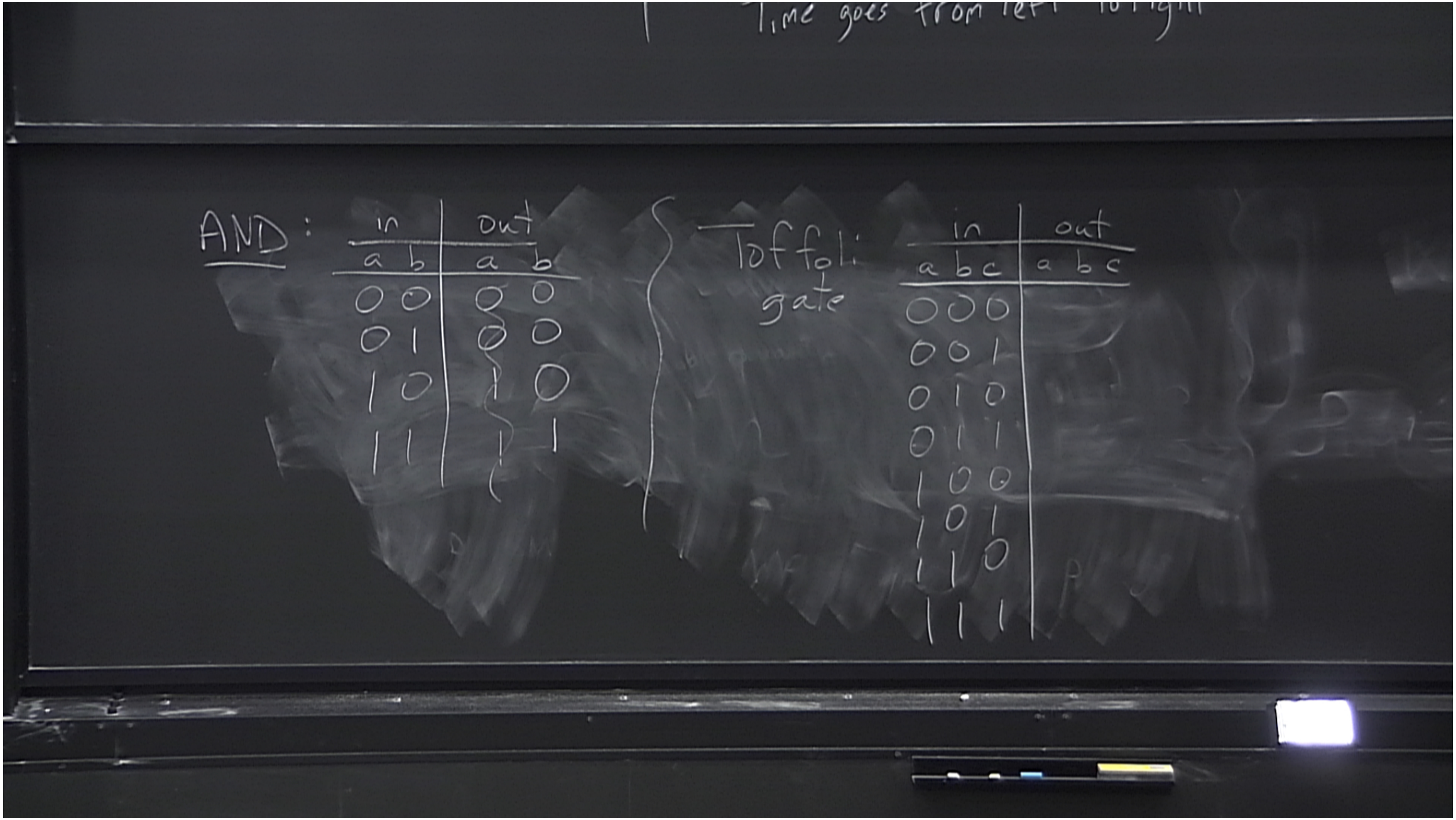

Time goes from left to right
AND:
in
out
a b
a b
0 0
0
0 1
0
1 0
0
1 1
1

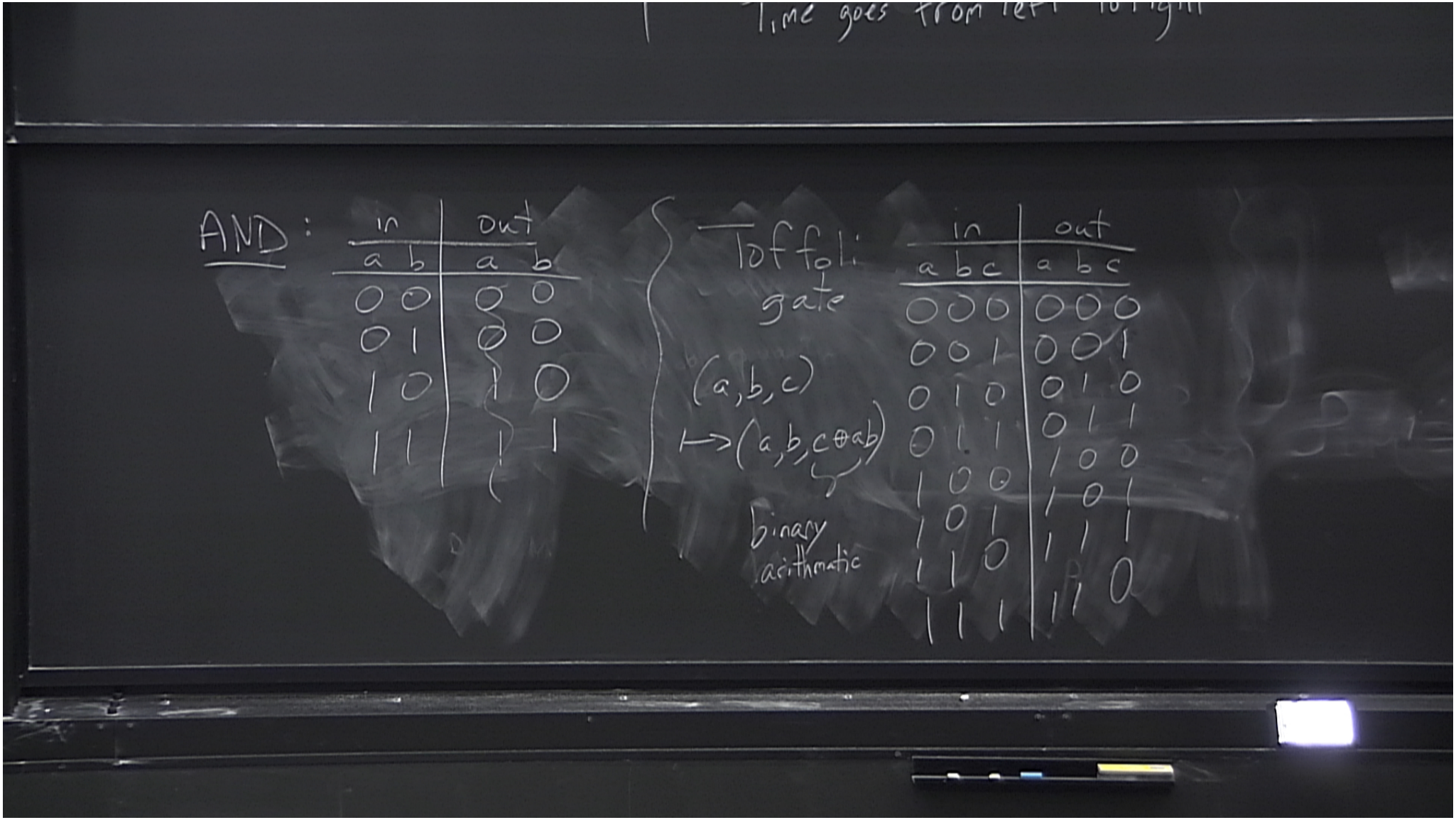

Time goes from left to right
AND:
in
out
a b
a b
0 0
0 0
0 1
0 0
1 0
1 0
1 1
1 1

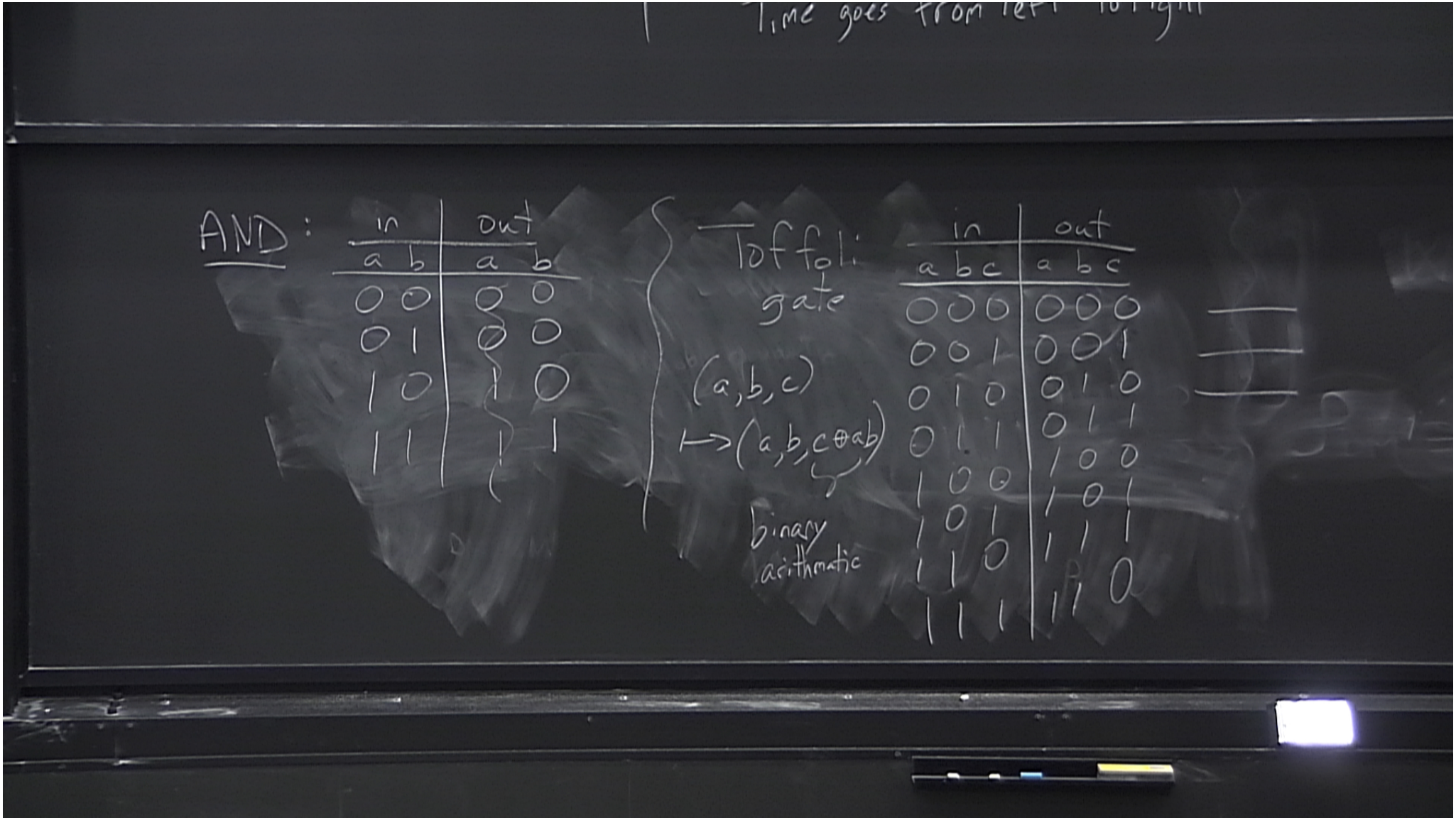

Time goes from left to right
AND:
in
out
a b
a b
0 0
0 0
0 1
0 0
1 0
1 0
1 1
1

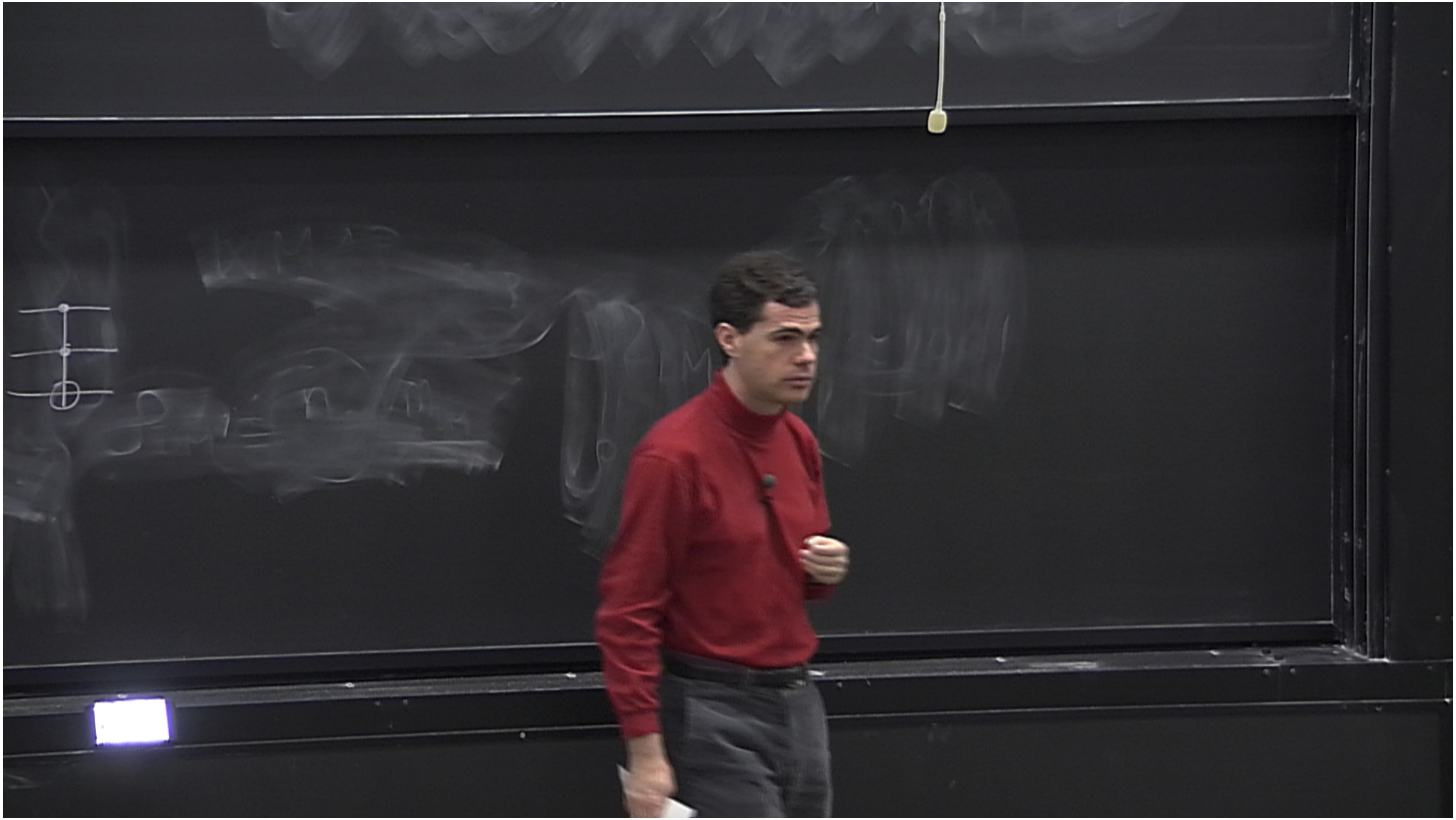

Time goes from left to right
AND:
in
out
a b
a b
0 0
0 0
0 1
0 0
1 0
1 0
1 1
1 1
Toffoli gate


Time goes from left to right
AND:
in | out
a b | a b
0 0 | 0 0
0 1 | 0 0
1 0 | 1 0
1 1 | 1 1
Toffoli gate
in | out
a b c | a b c


Time goes from left to right
AND:
in
out
a b
a b
0 0
0 0
0 1
0 0
1 0
1 0
1 1
1 1
Toffoli gate
in
out
a b c
a b c
0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1


Time goes from left to right
AND:
in | out
a b | a b
0 0 | 0 0
0 1 | 0 0
1 0 | 1 0
1 1 | 1 1
Toffoli gate
(a,b,c)
↦(a,b,c⊕ab)
binary arithmatic
in | out
a b c | a b c
0 0 0 | 0 0 0
0 0 1 | 0 0 1
0 1 0 | 0 1 0
0 1 1 | 0 1 1
1 0 0 | 1 0 0
1 0 1 | 1 0 1
1 1 0 | 1 1 1
1 1 1 | 1 1 0


Time goes from left to right
AND:
in | out
a b | a b
0 0 | 0 0
0 1 | 0 0
1 0 | 1 0
1 1 | 1 1
Toffoli gate
(a,b,c)
↦(a,b,c⊕ab)
binary arithmatic
in | out
a b c | a b c
0 0 0 | 0 0 0
0 0 1 | 0 0 1
0 1 0 | 0 1 0
0 1 1 | 0 1 1
1 0 0 | 1 0 0
1 0 1 | 1 0 1
1 1 0 | 1 1 1
1 1 1 | 1 1 0



Time goes from left to right

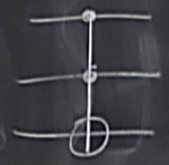
Toffoli gate

(a,b,c)

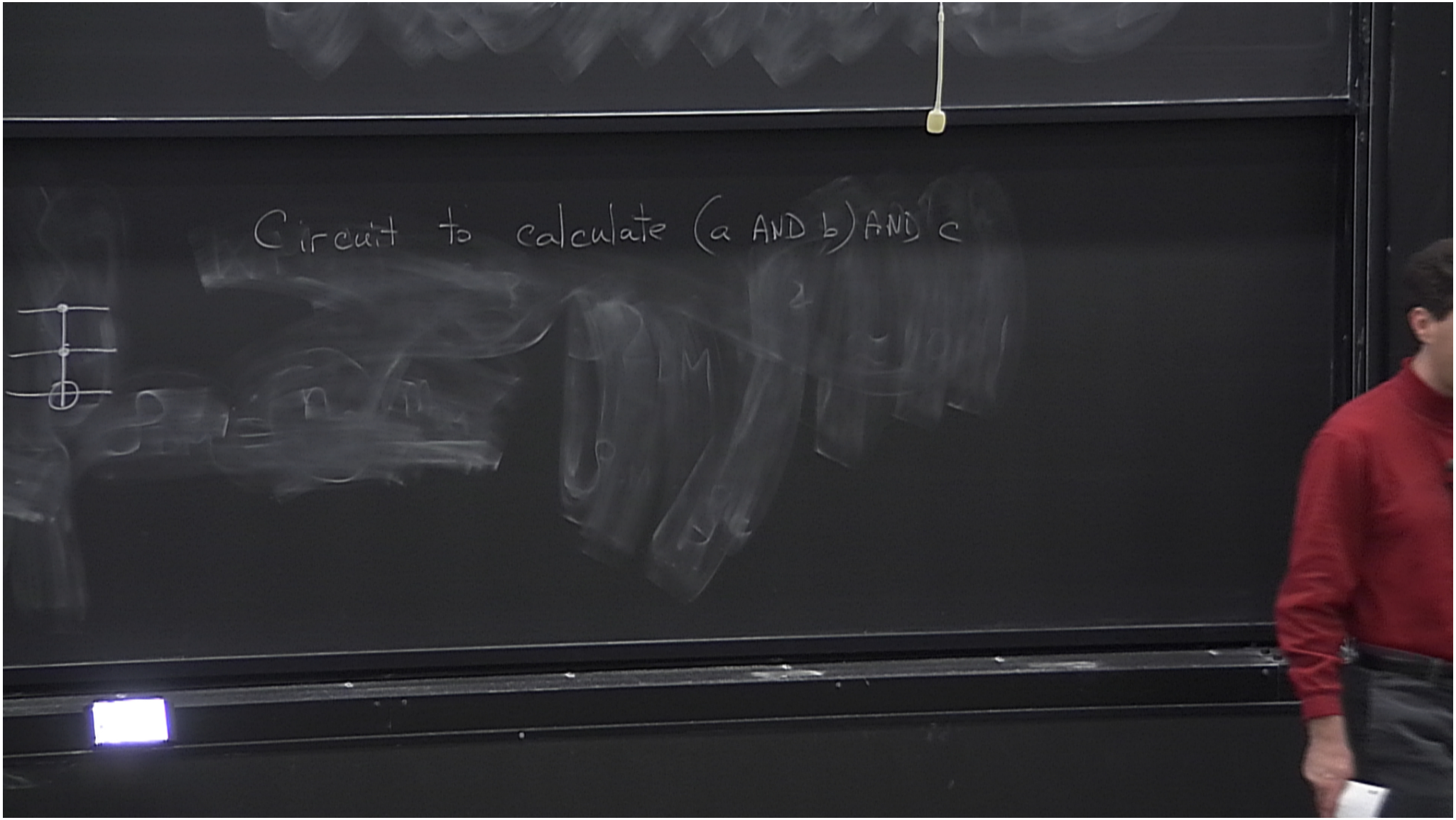
(a,b,c⊕ab)

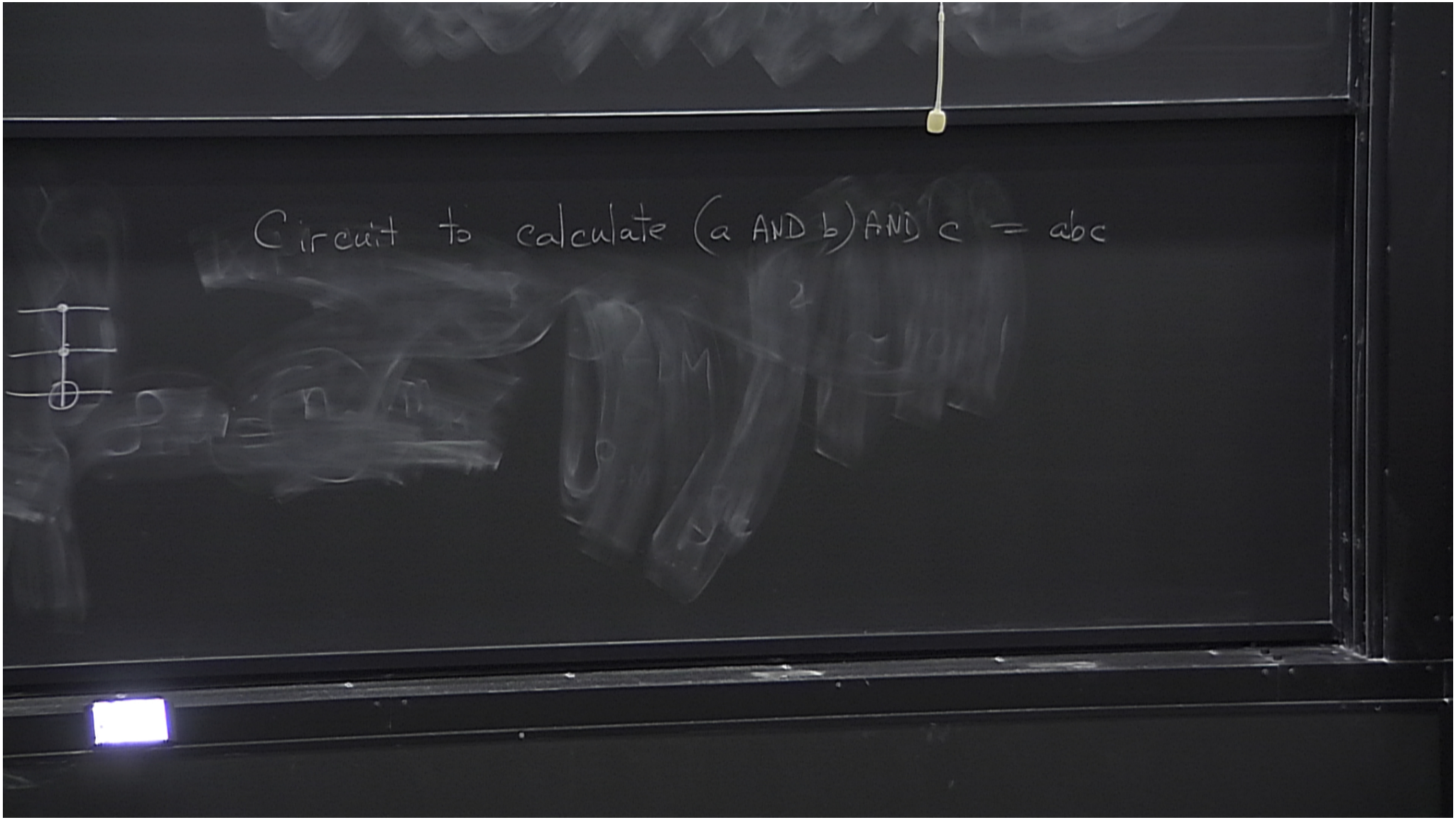
binary arithmatic

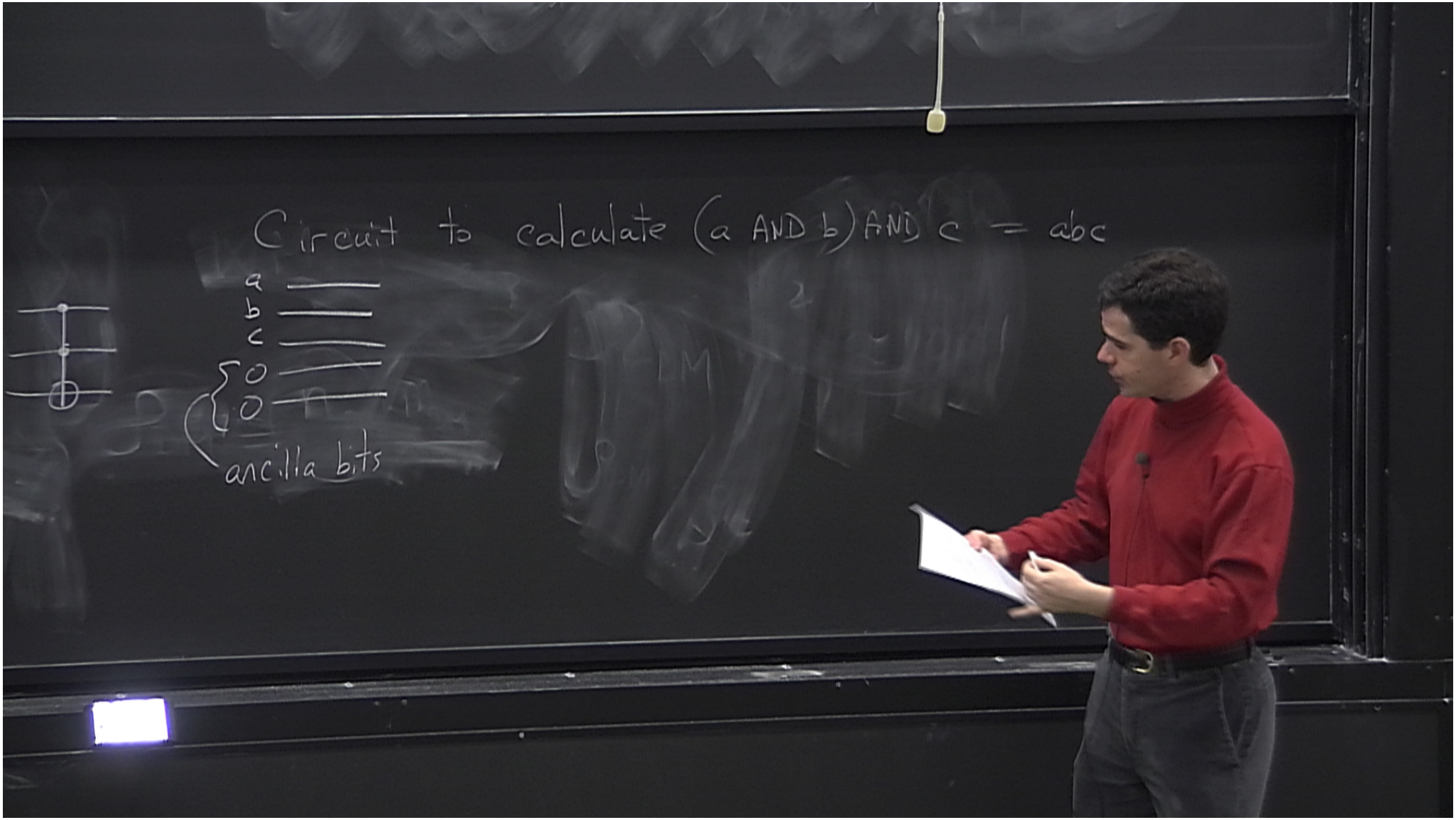
in			out		
a	b	c	a	b	c
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

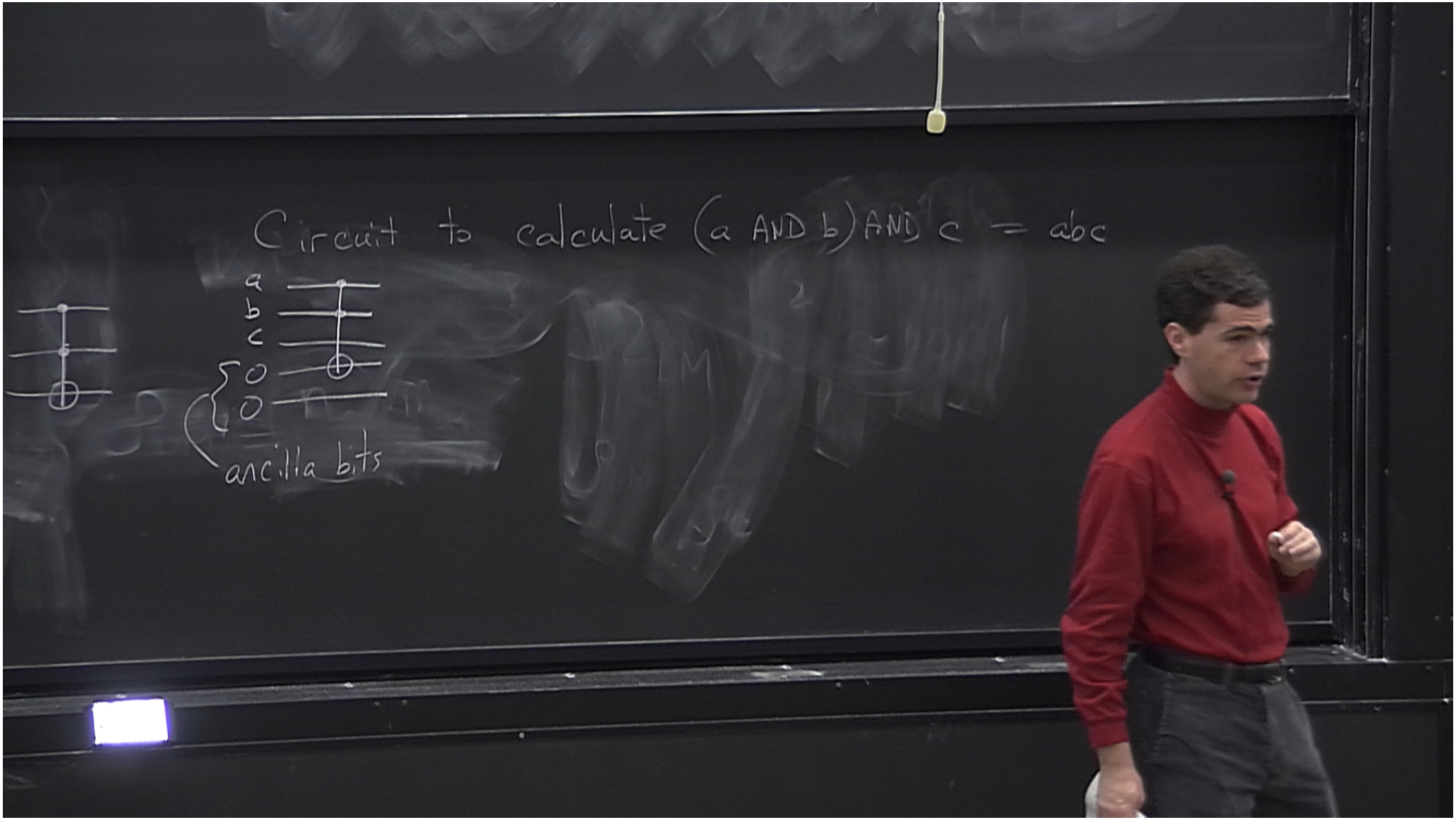


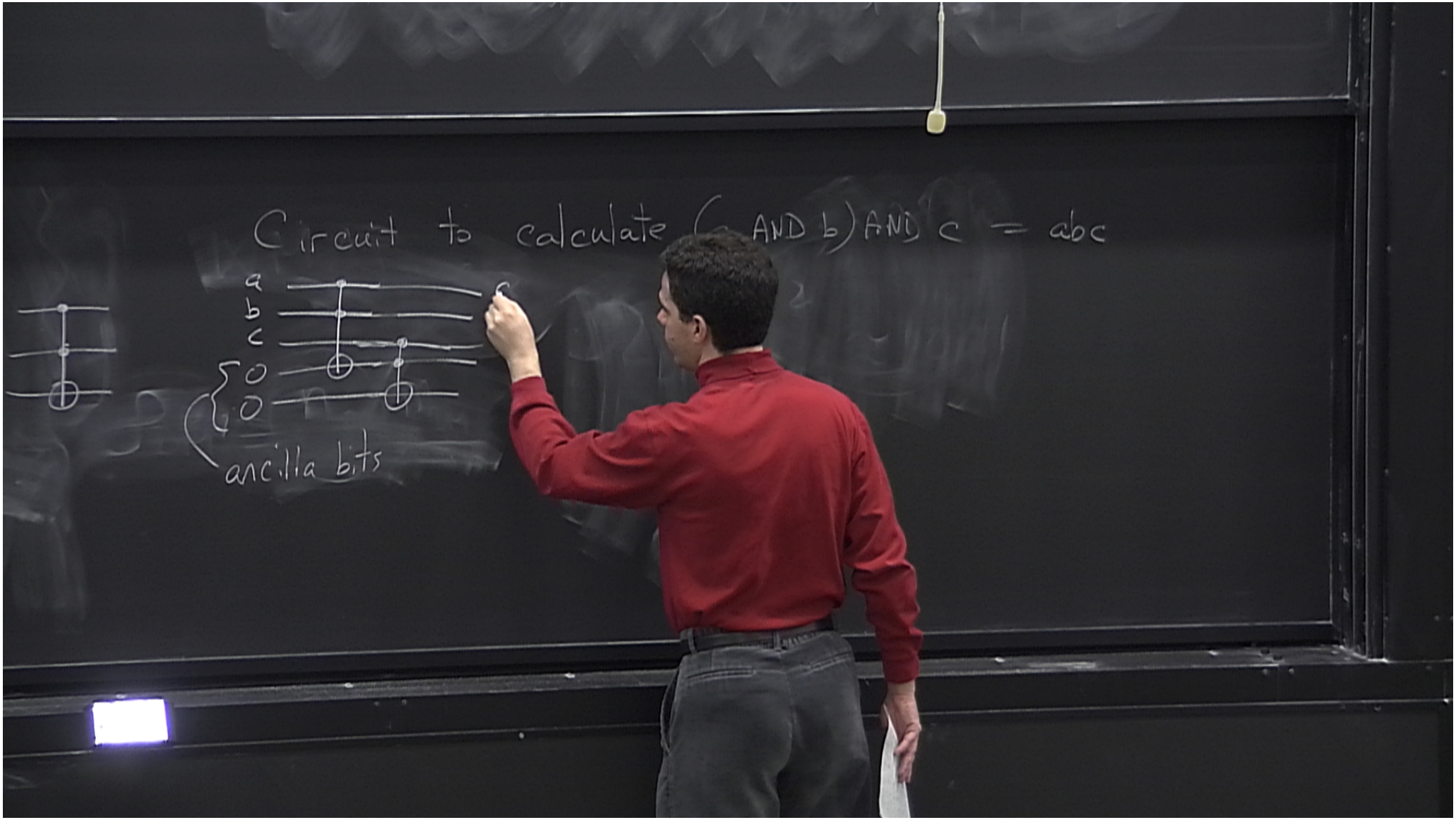
Circuit to calculate

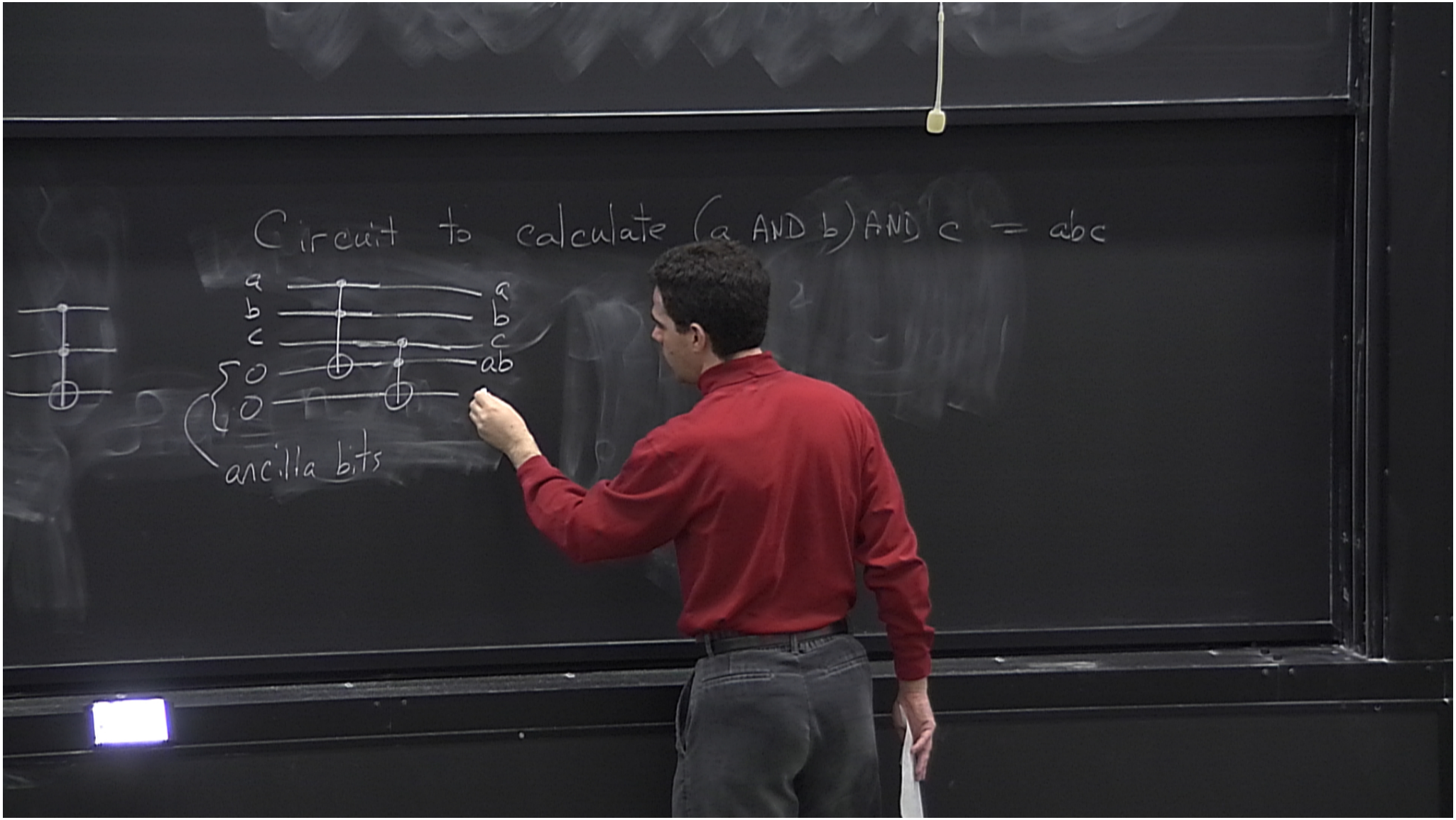

Circuit to calculate (a AND b) AND c

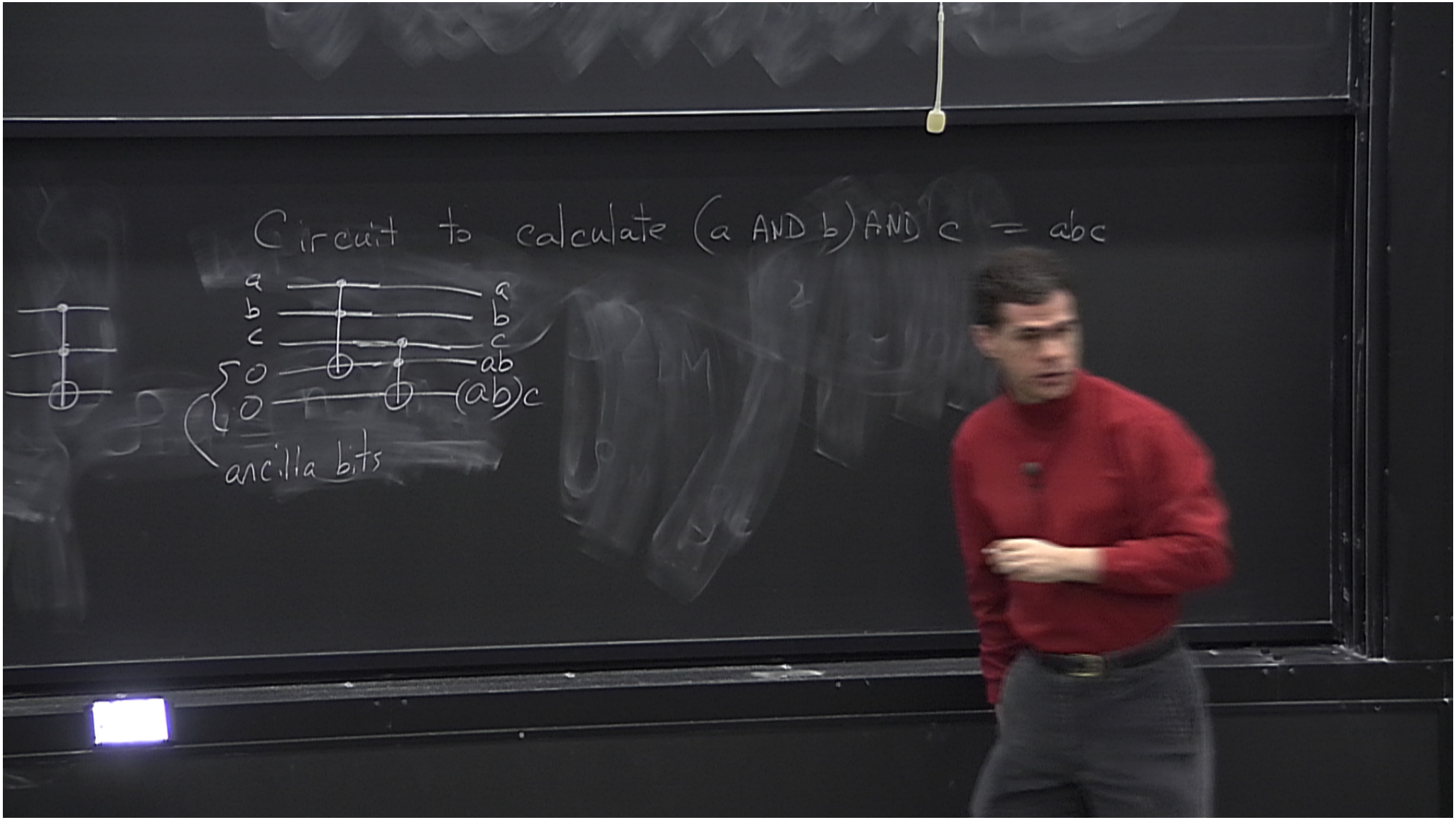

Circuit to calculate (a AND b) AND c = abc


Circuit to calculate (a AND b) AND c = abc
a
b
c
0
0
ancilla bits

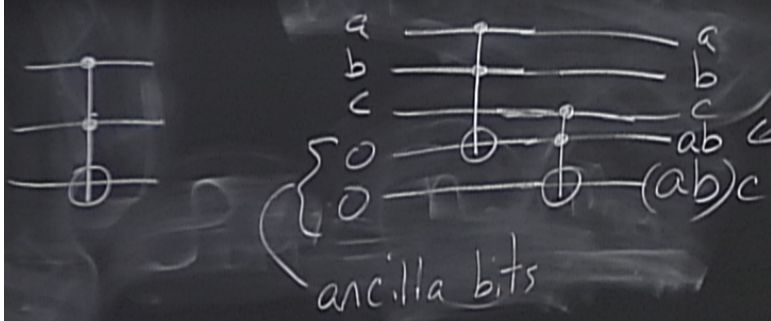

Circuit to calculate (a AND b) AND c = abc
a
b
c
0
0
ancilla bits


Circuit to calculate (a AND b) AND c = abc
a
b
c
0
0
ancilla bits

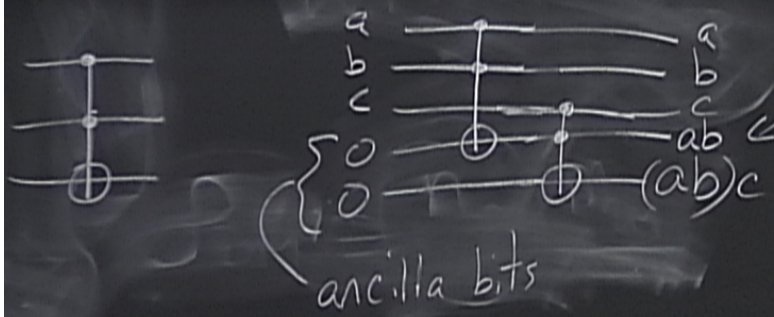

Circuit to calculate (a AND b) AND c = abc
a
b
c
0
0
a
b
c
ab
ancilla bits

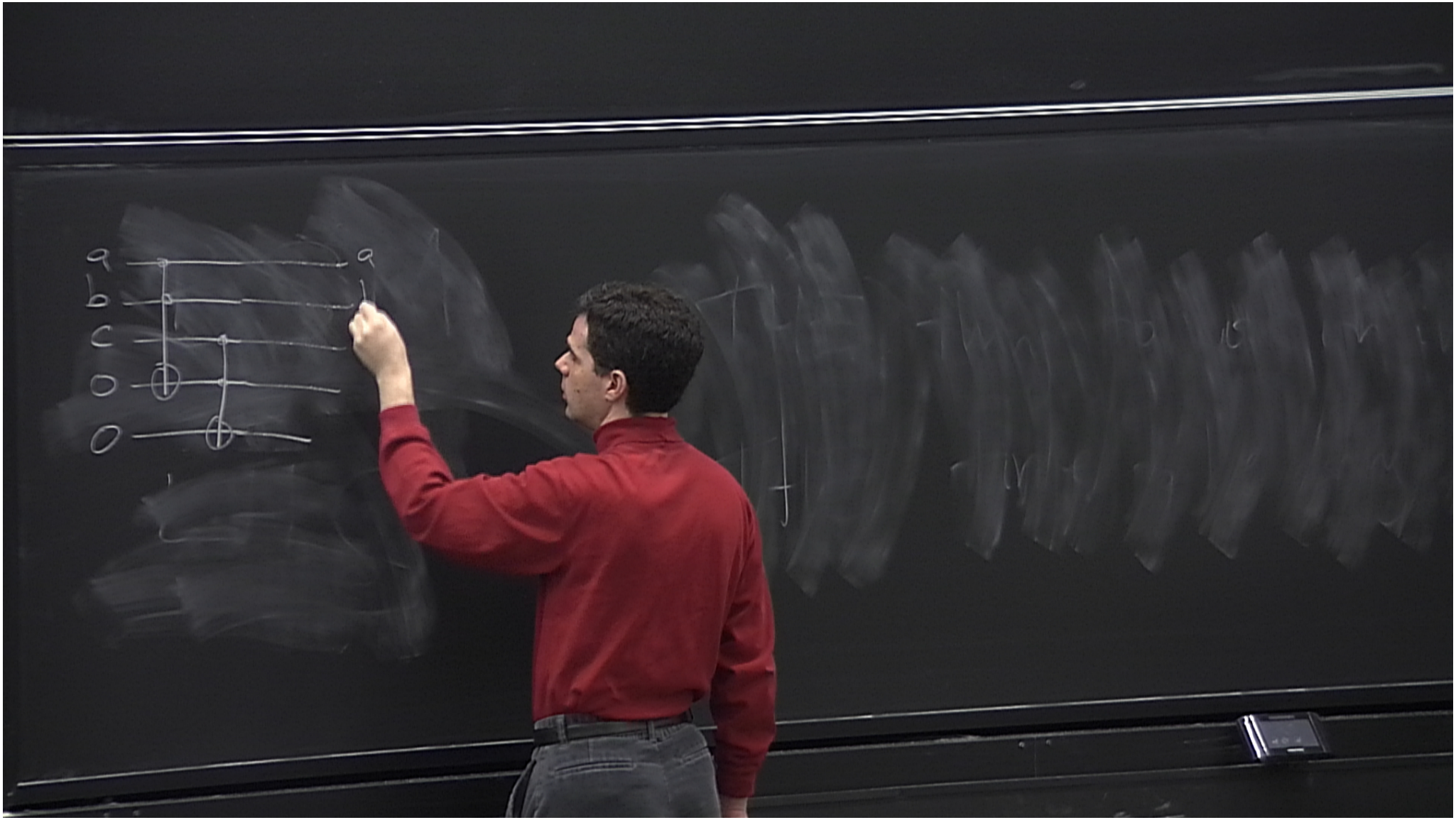

Circuit to calculate (a AND b) AND c = abc
a
b
c
0
0
a
b
c
ab
(ab)c
ancilla bits

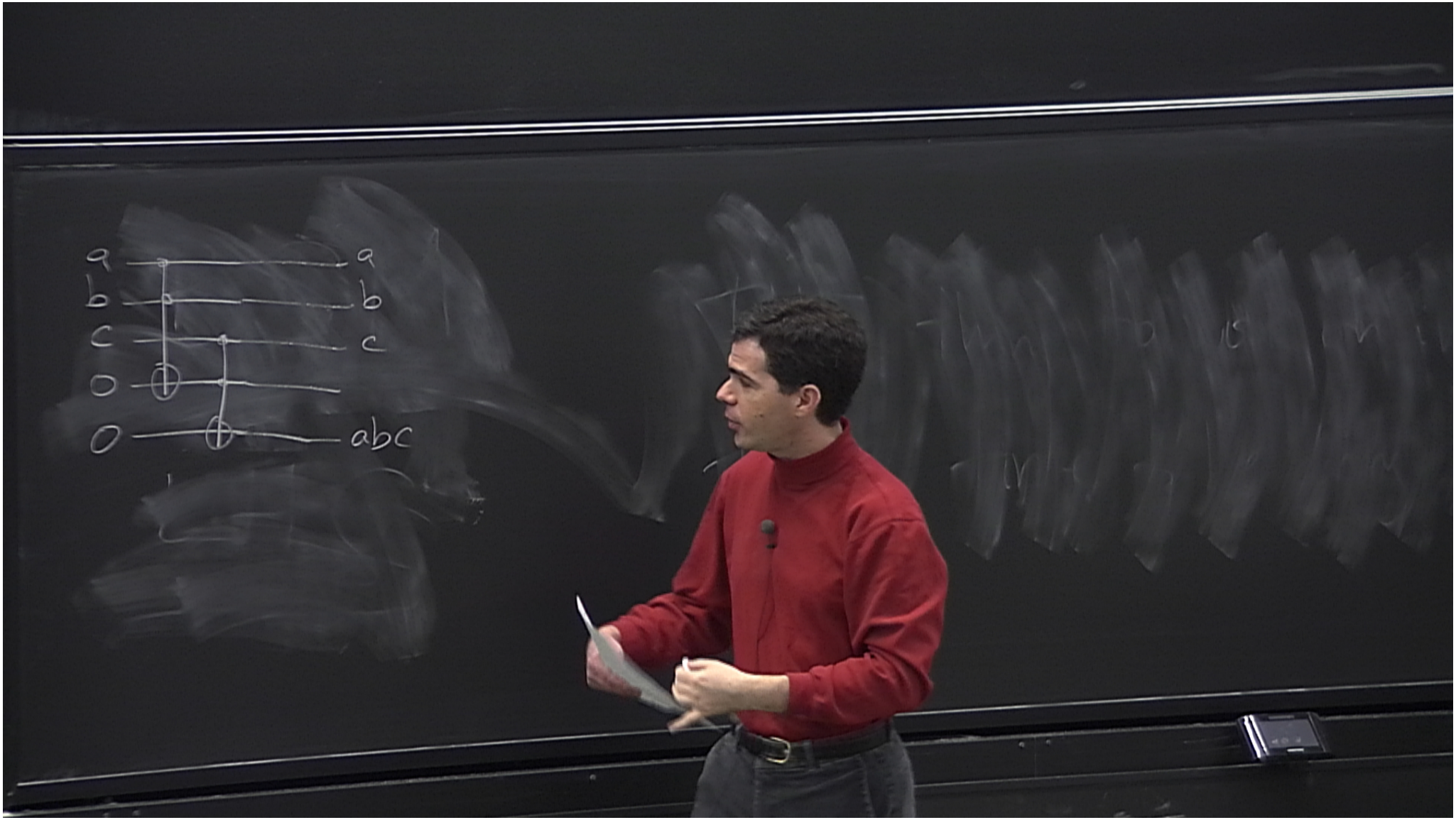
Circuit to calculate (a AND b) AND c = abc

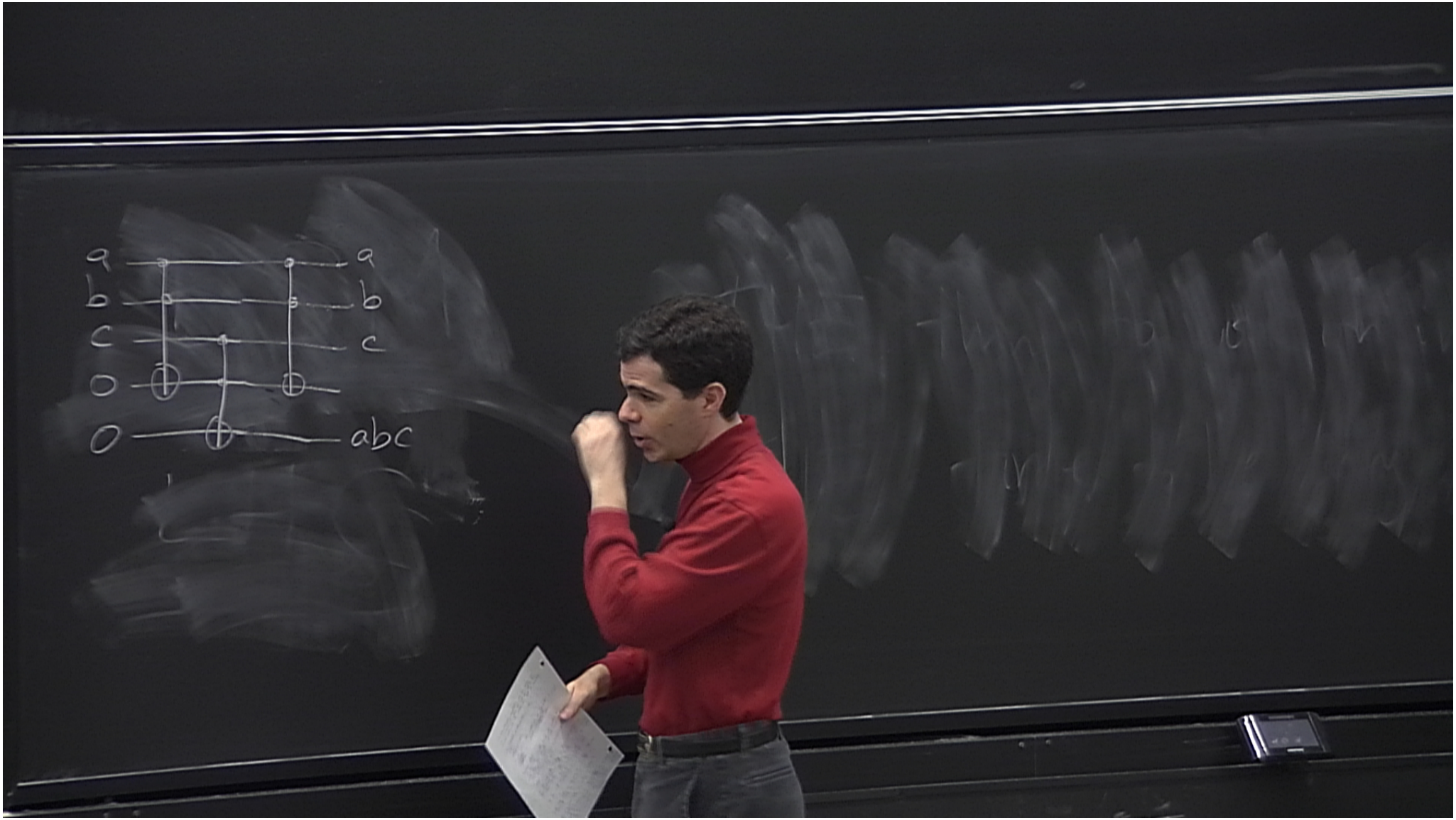


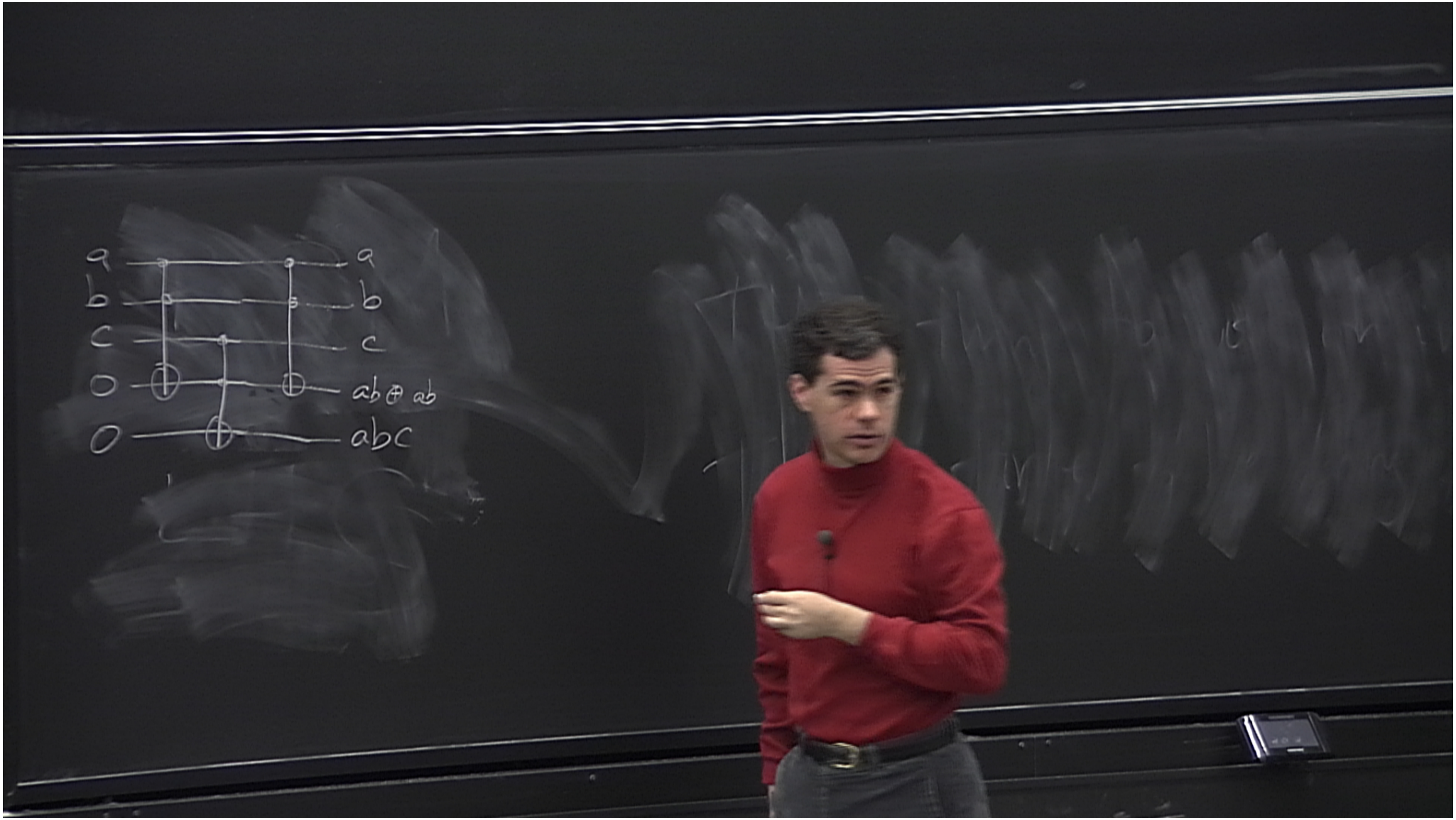
Circuit to calculate (a AND b) AND c = abc

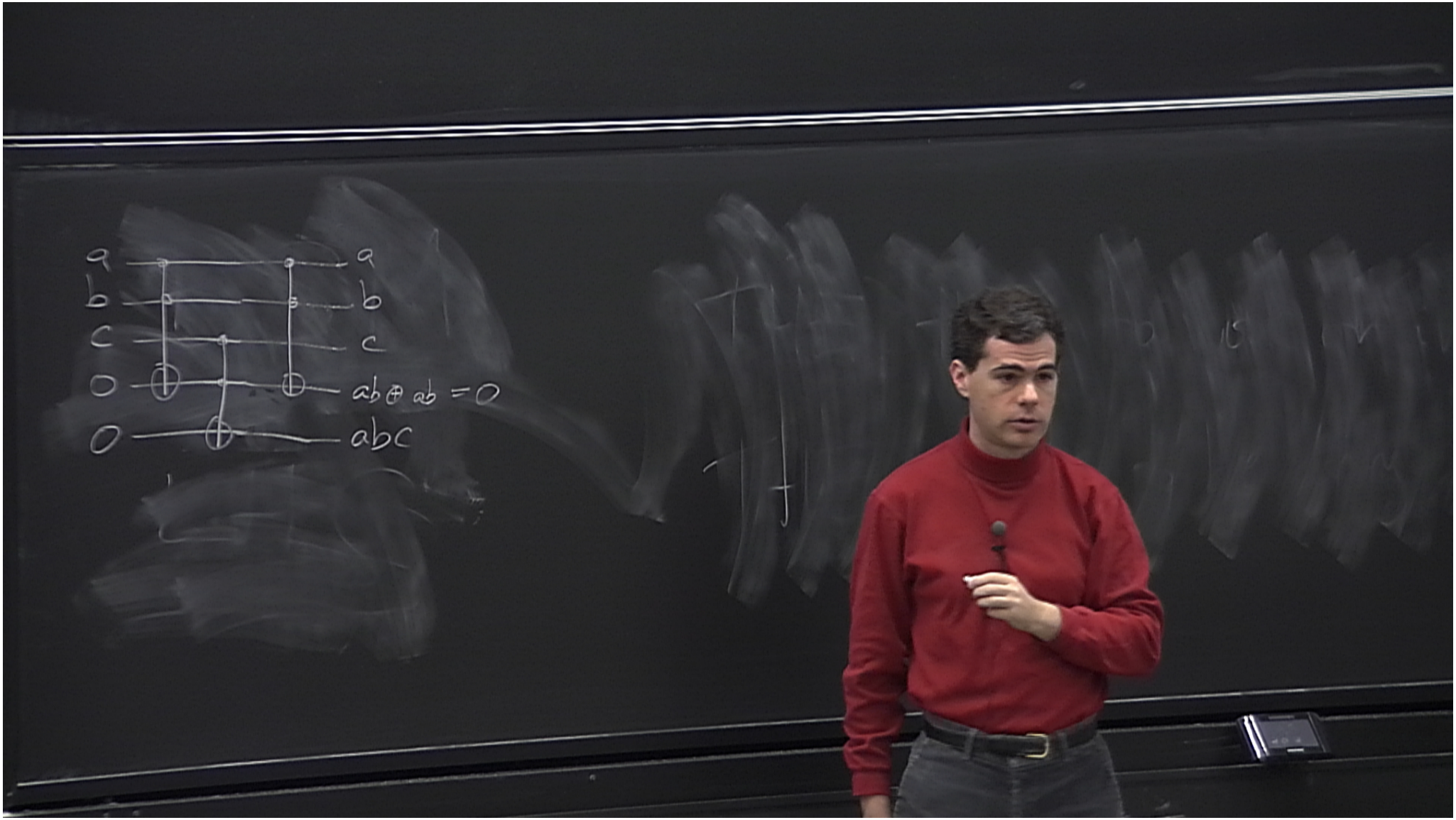


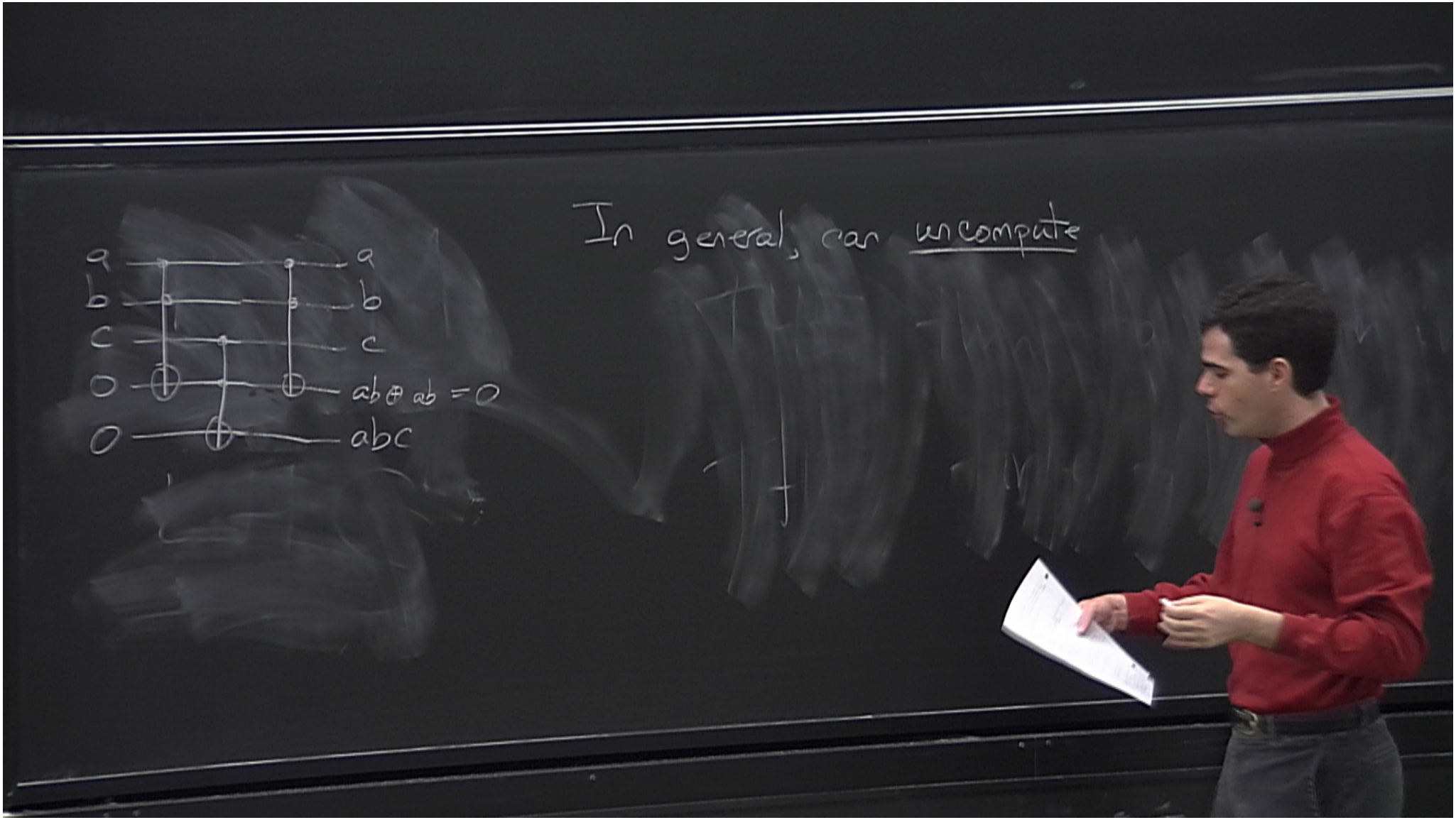

a
b
c
0
0
a

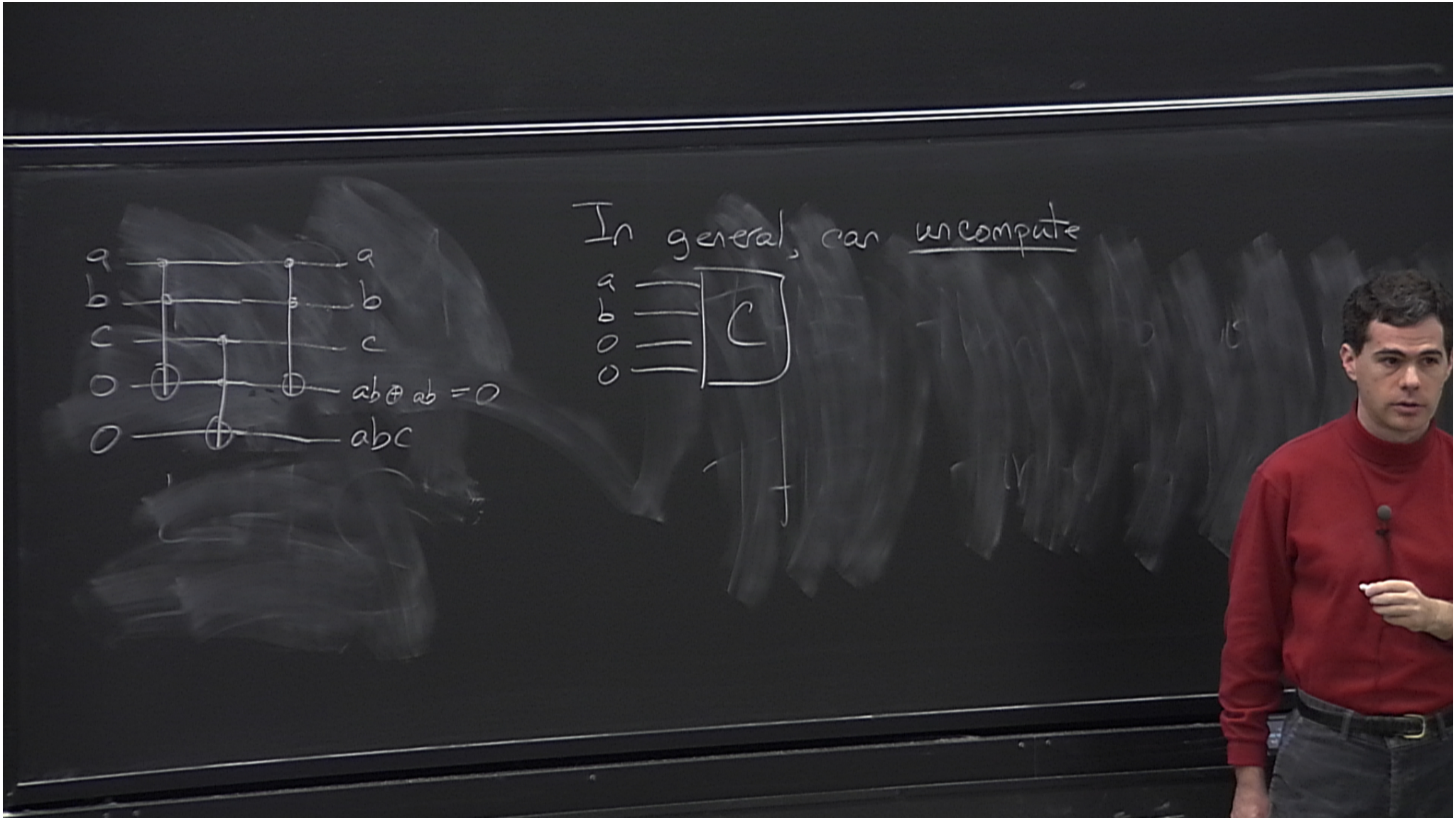

a
b
c
0
0
a
b
c
abc

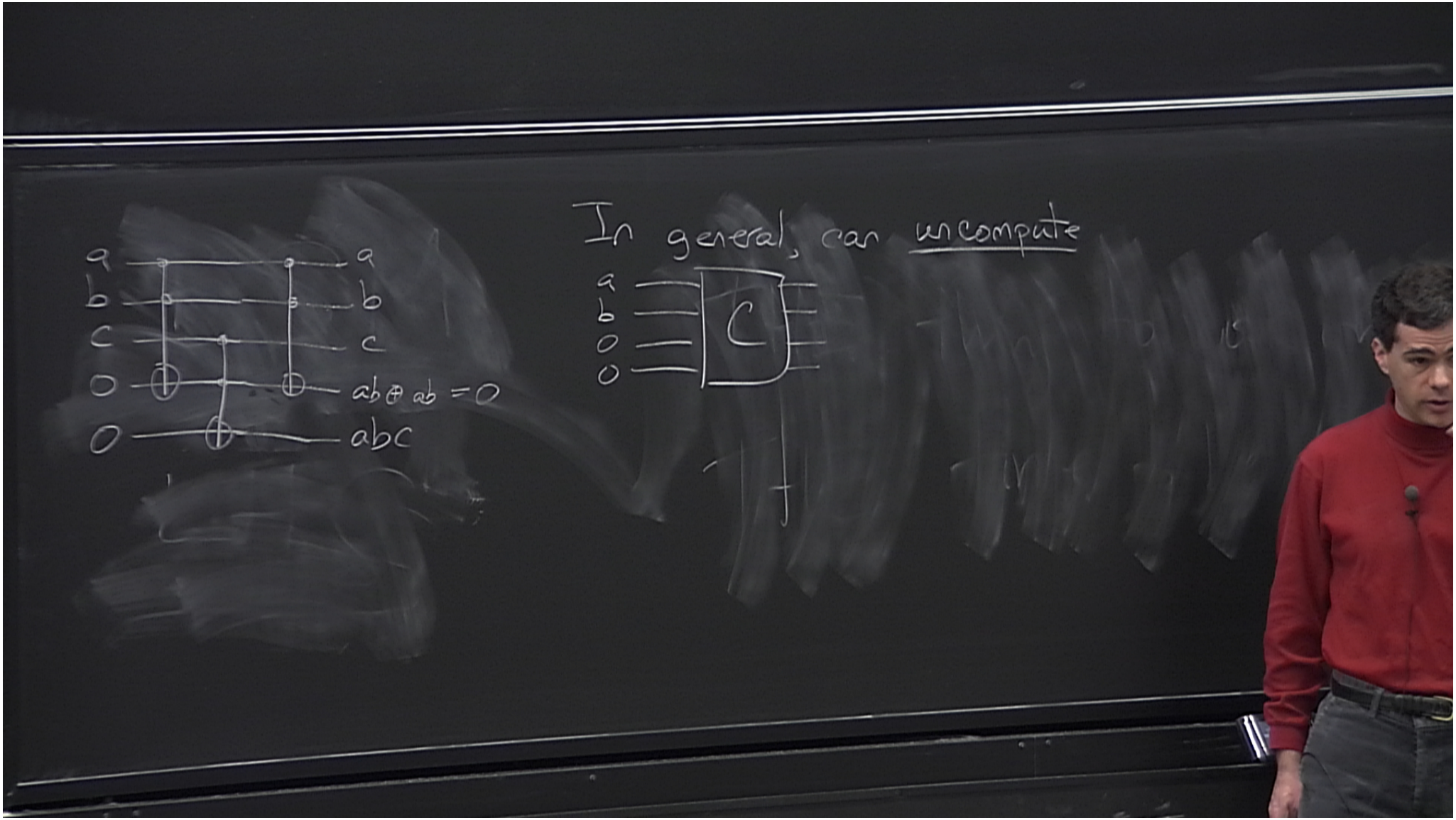

a
b
c
0
0
a
b
c
abc

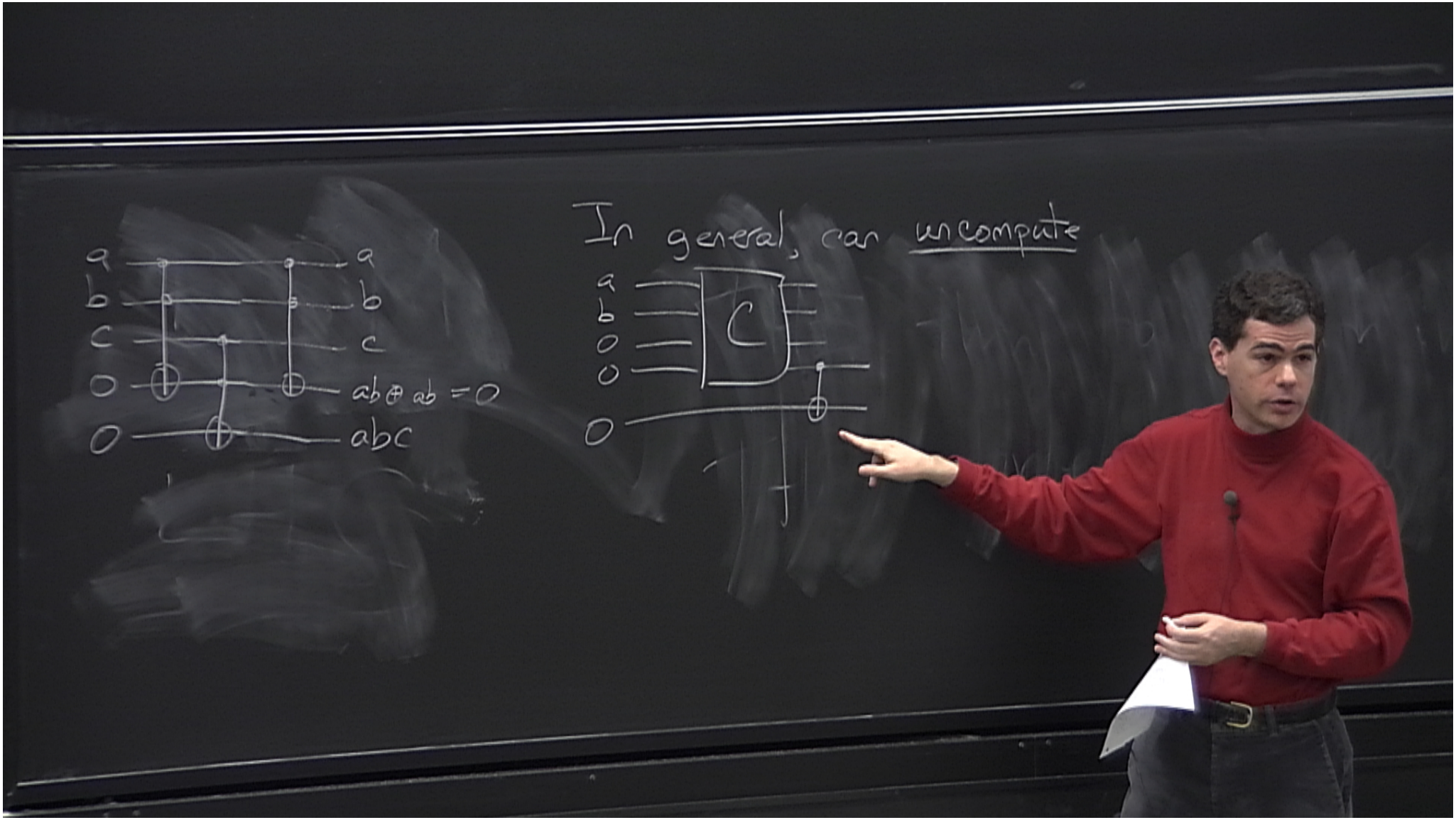

a
b
c
0
0
a
b
c
ab ⊕ ab
abc

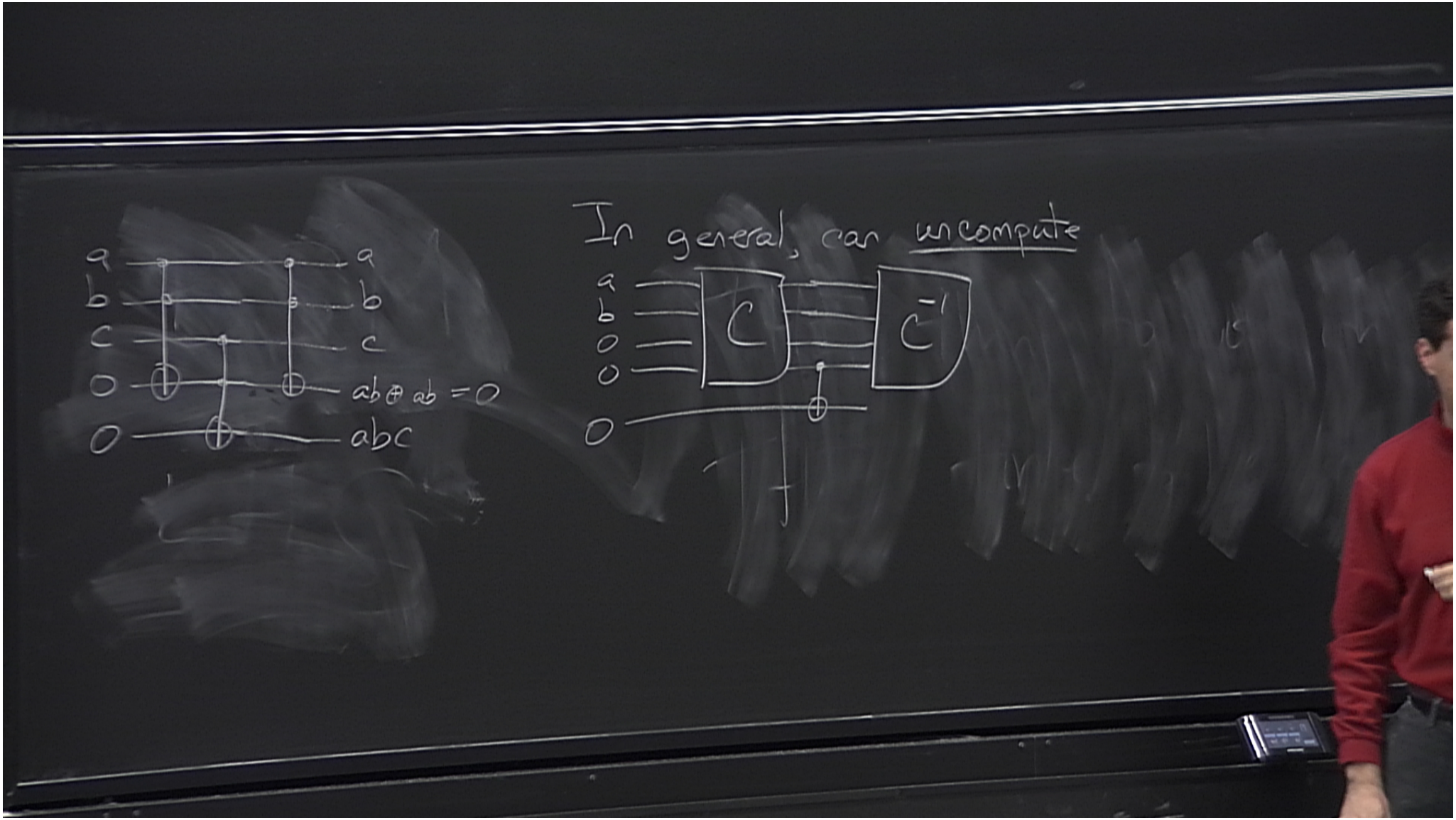

a
b
c
0
0
a
b
c
ab ⊕ ab = 0
abc

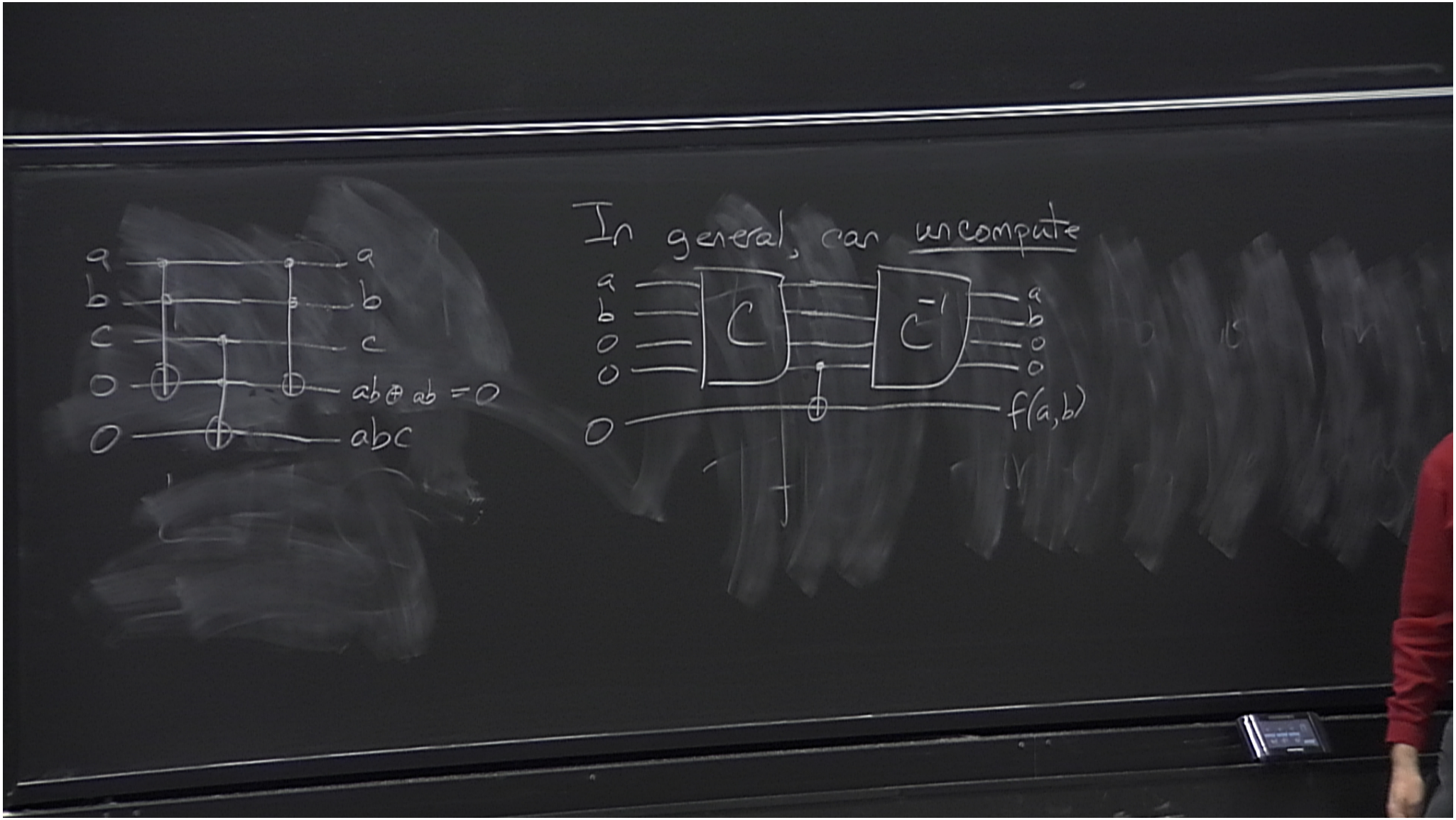

In general, can uncompute
a
b
c
0
0
a
b
c
ab ⊕ ab = 0
abc

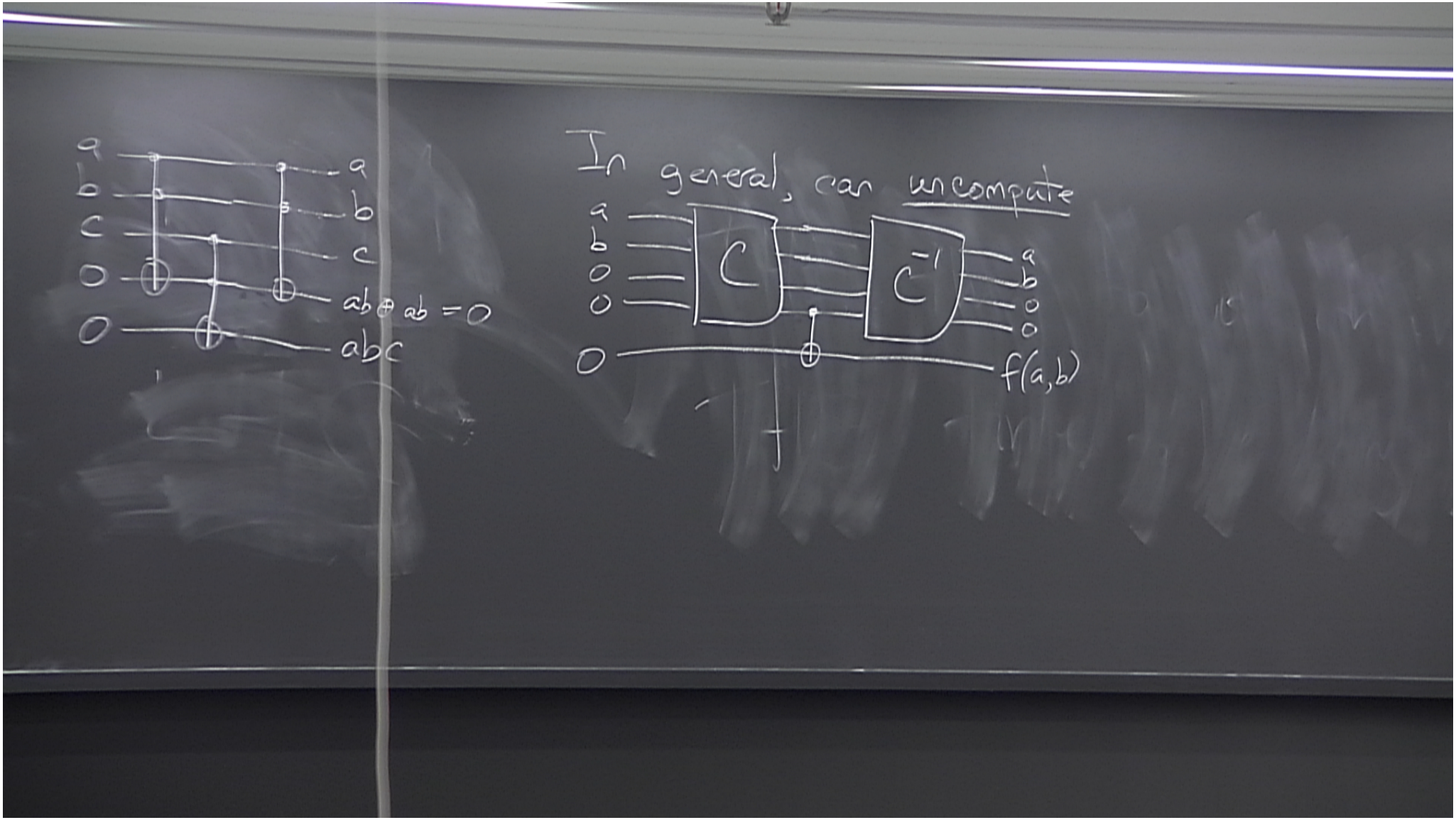

In general, can uncompute
a
b
c
0
0
C
a
b
c
ab ⊕ ab = 0
abc

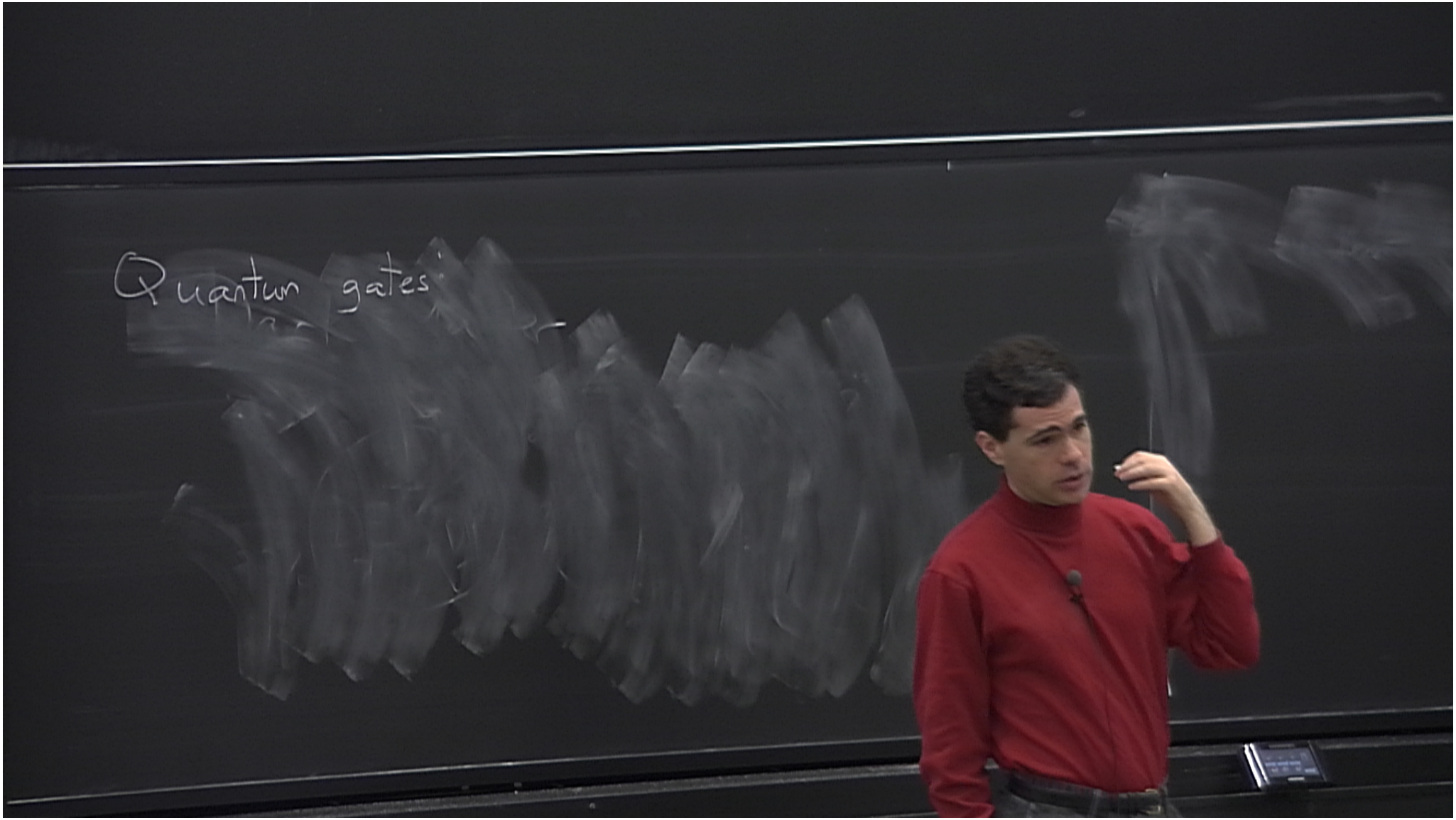

In general, can uncompute
a
b
c
0
0
C
a
b
c
ab ⊕ ab = 0
abc

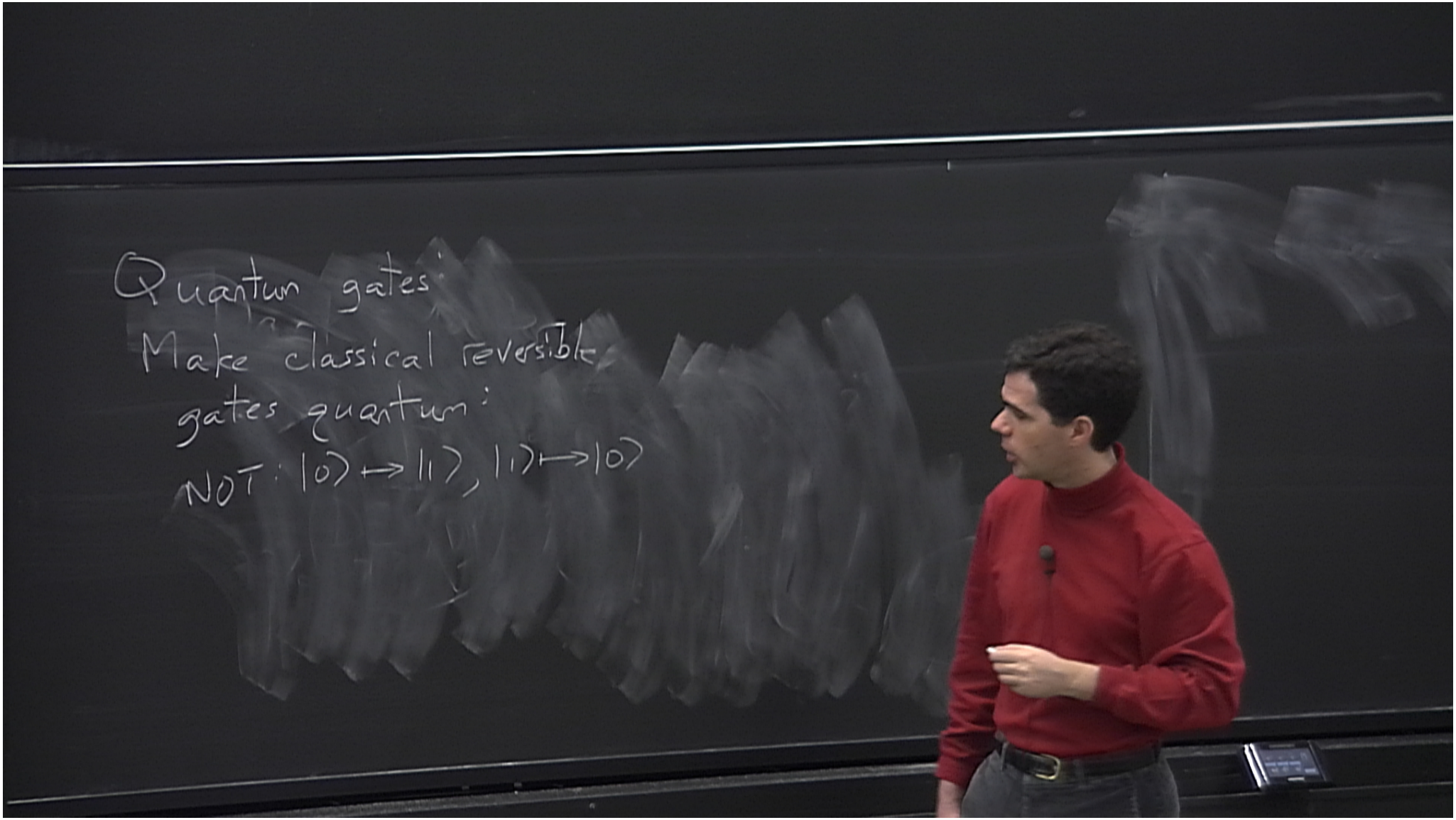

a
b
c
0
0
a
b
c
ab ⊕ ab = 0
abc
In general, can uncompute
a
b
0
0
C
0

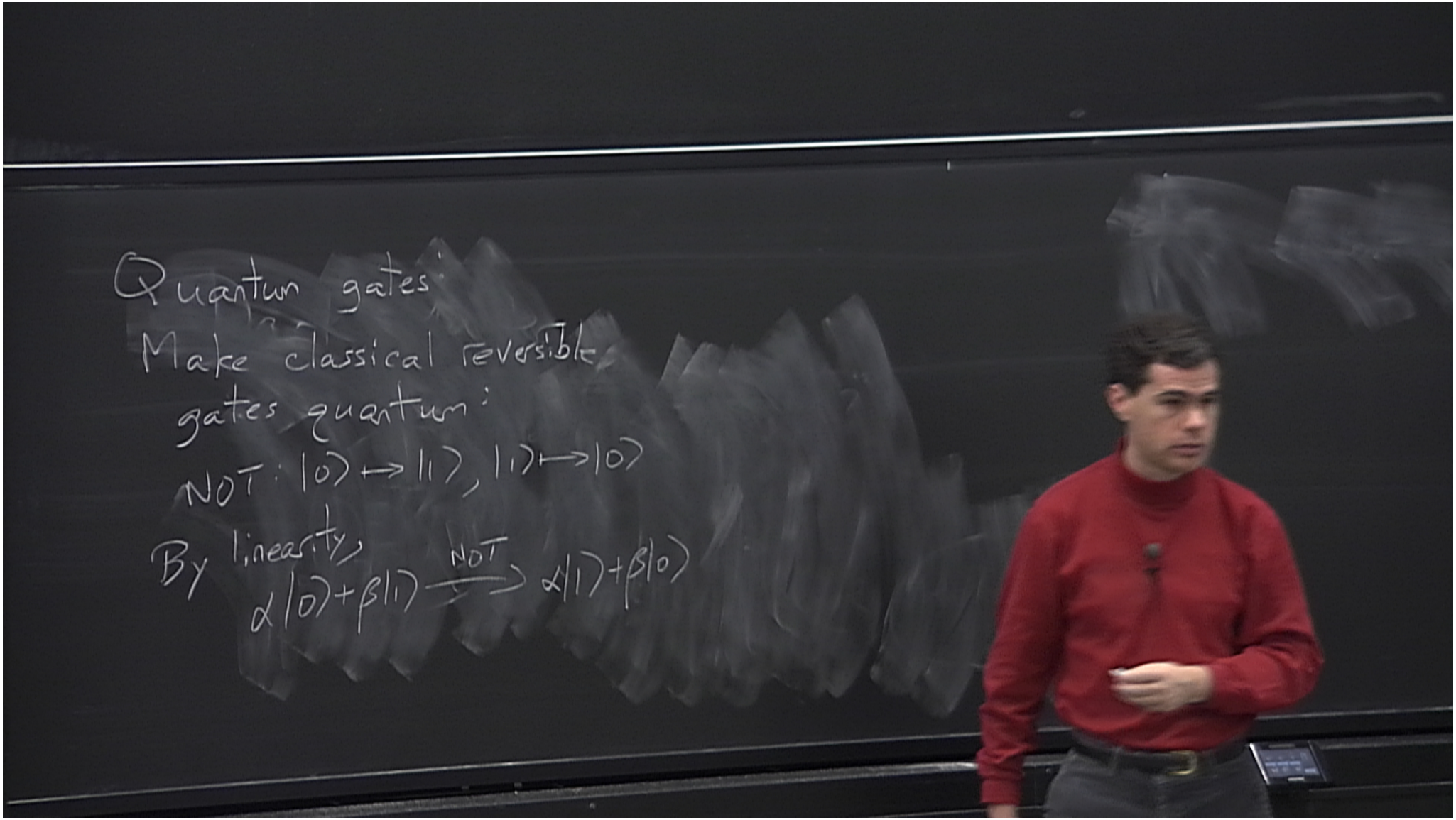

a
b
c
0
0
a
b
c
$ab \oplus ab = 0$
abc
In general, can uncompute
a
b
0
0
0
C
C^{-1}

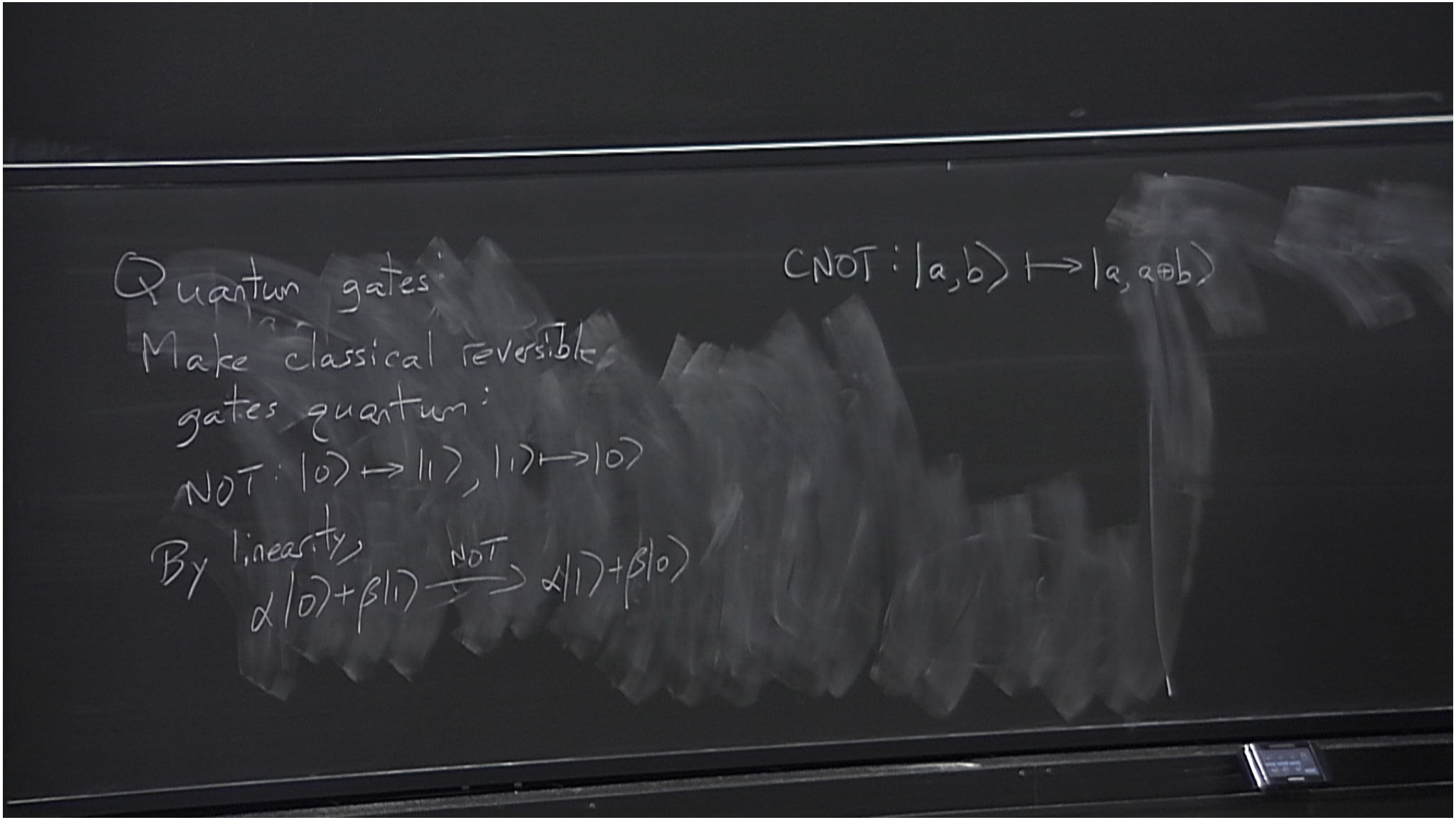

a
b
c
0
0
a
b
c
ab ⊕ ab = 0
abc
In general, can uncompute
a
b
0
0
C
C⁻¹
a
b
0
0
0
f(a,b)

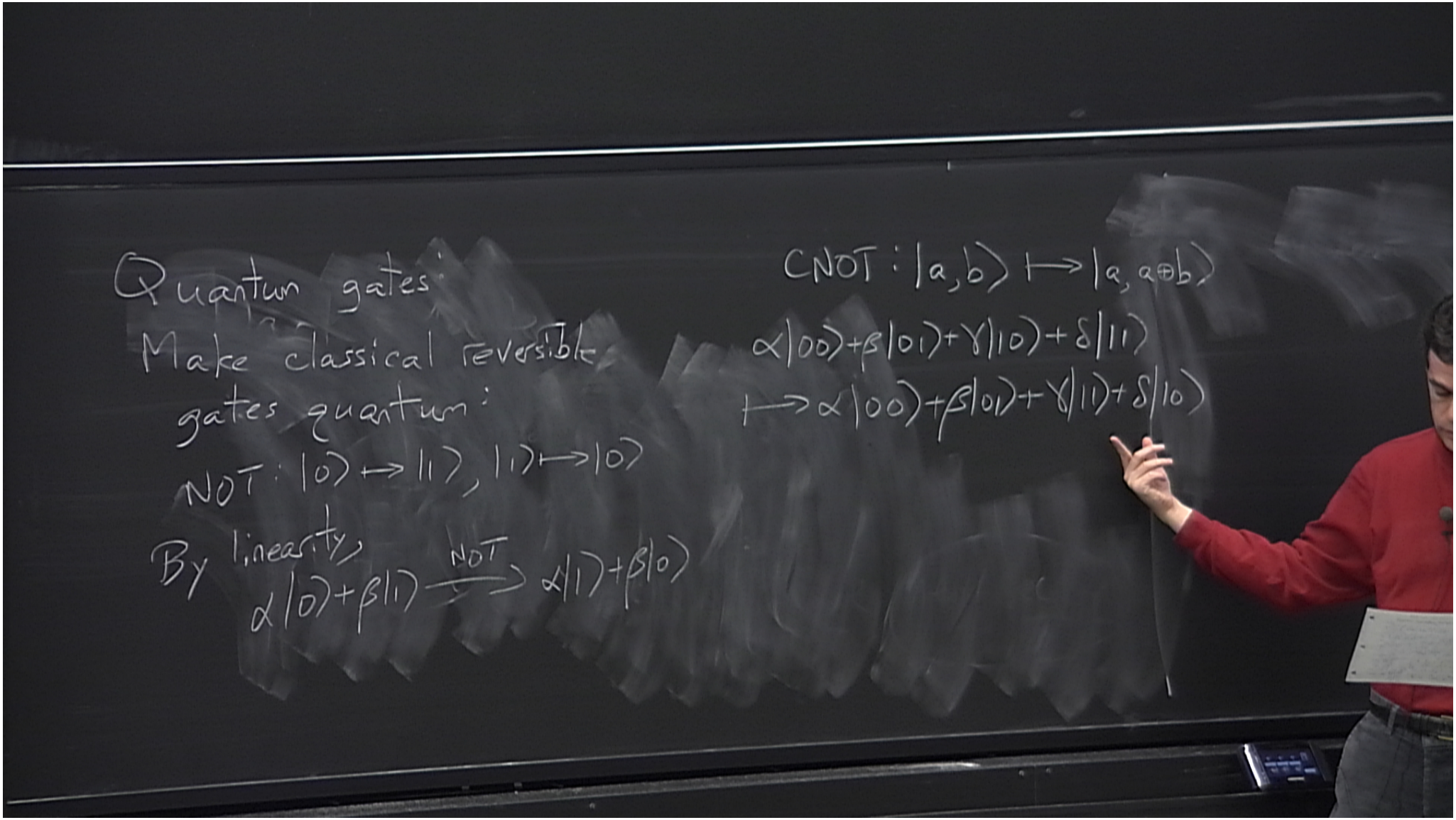

a
b
c
0
0
a
b
c
ab ⊕ ab = 0
abc
In general, can uncompute
a
b
0
0
0
C
C⁻¹
a
b
0
0
f(a,b)

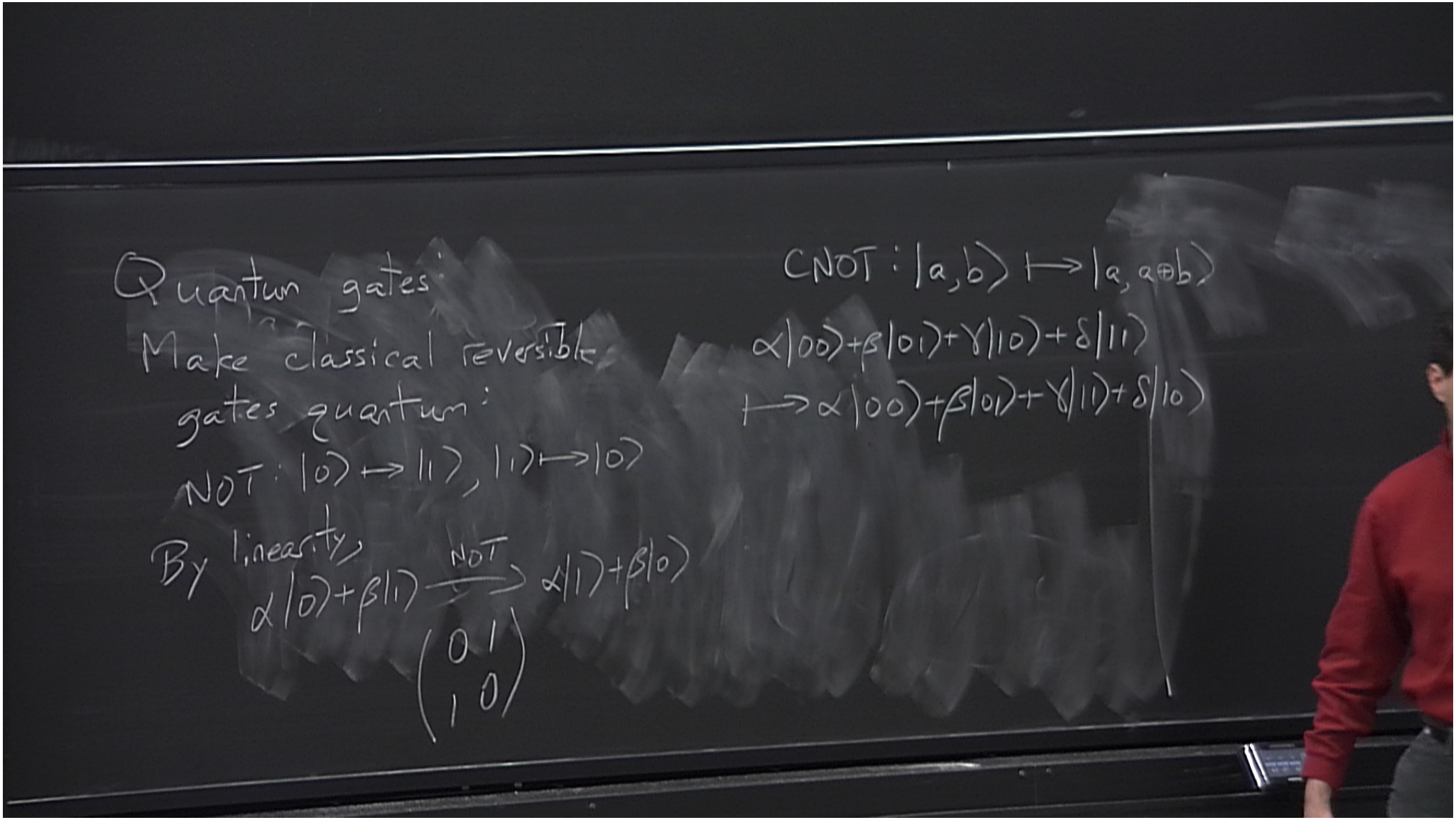

Quantum gates

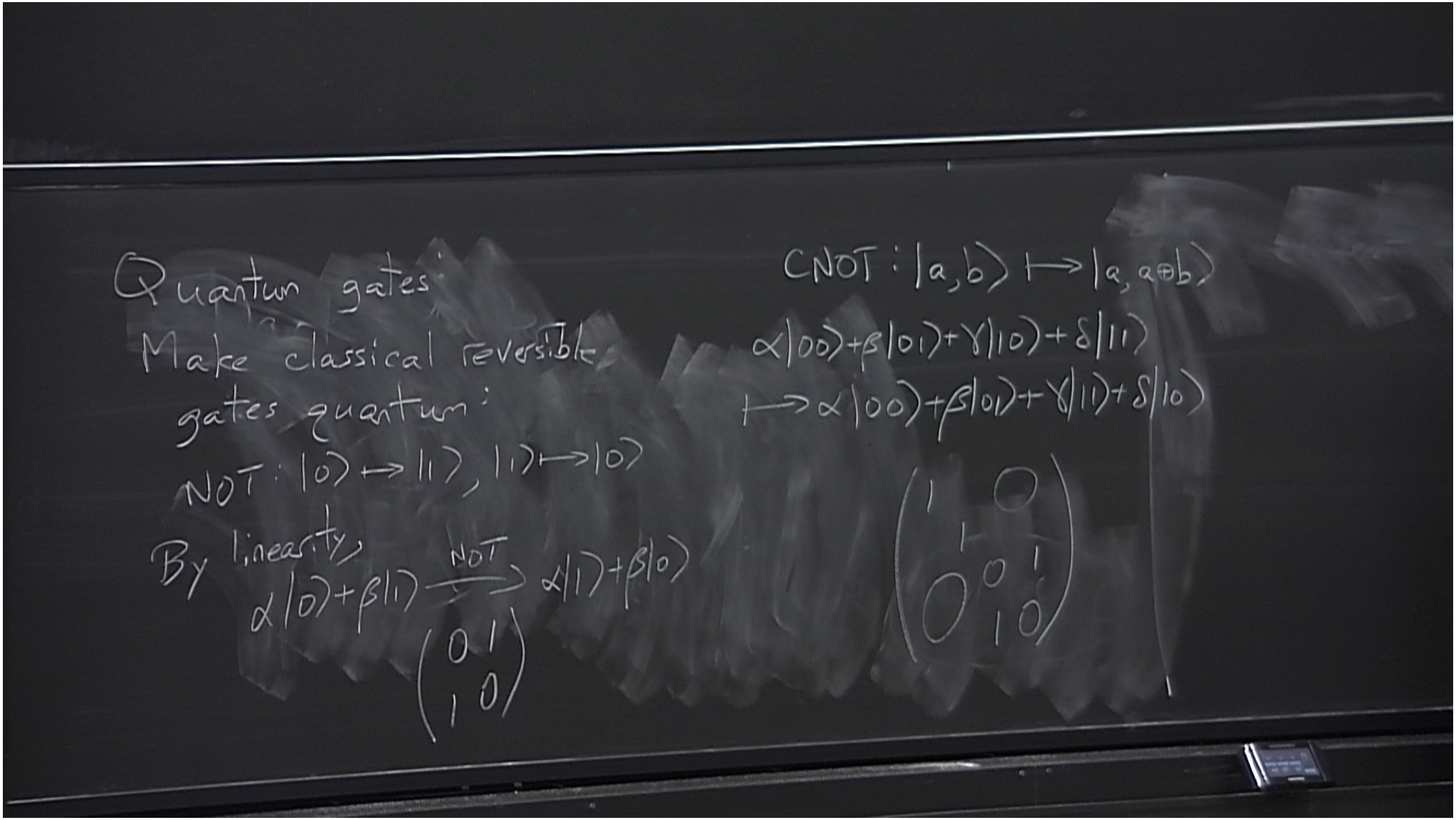

Quantum gates:
Make classical reversible gates quantum:
NOT: |0⟩ ↦ |1⟩, |1⟩ ↦ |0⟩

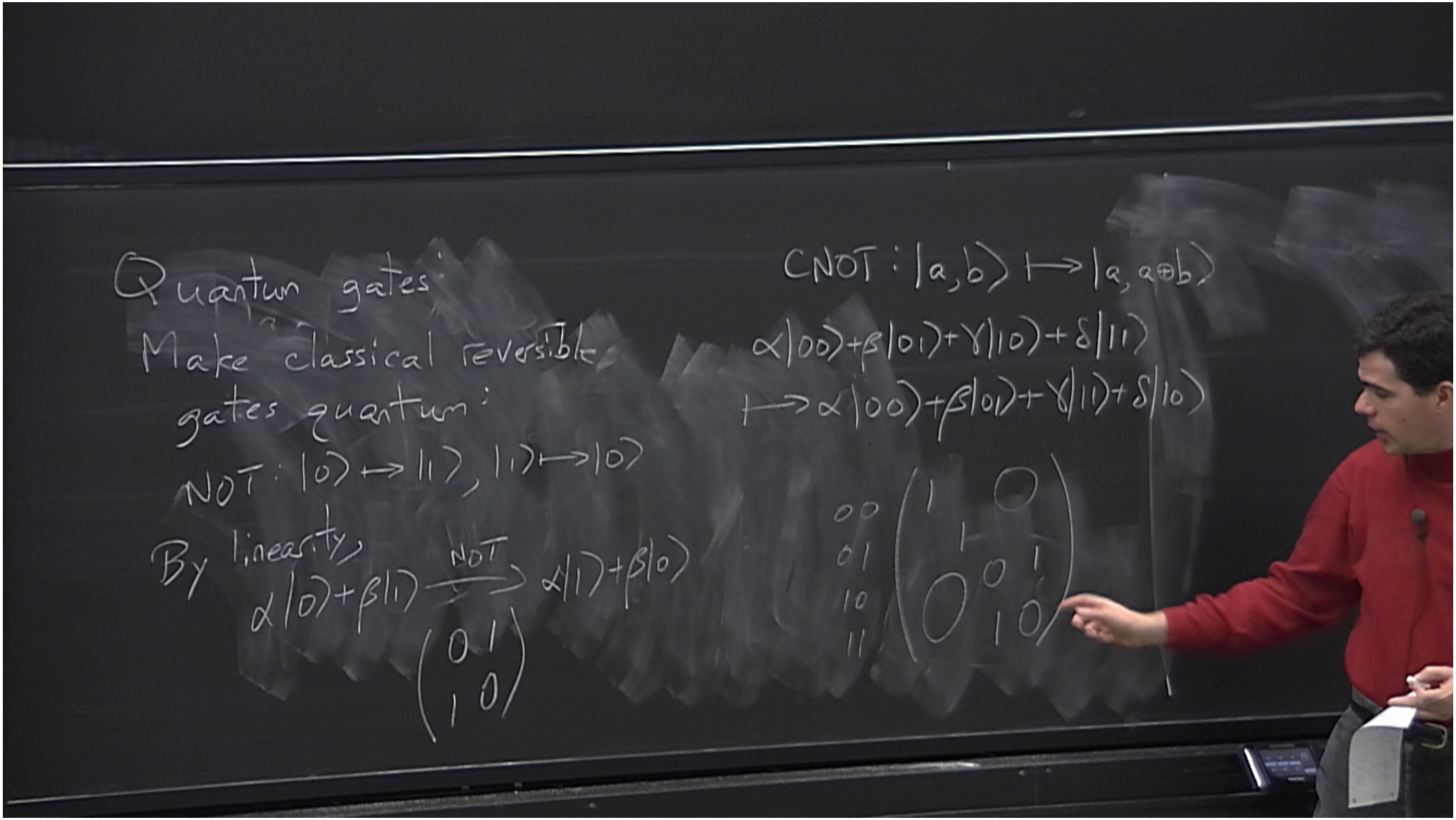

Quantum gates:
Make classical reversible gates quantum:
NOT: $|0\rangle \mapsto |1\rangle$, $|1\rangle \mapsto |0\rangle$
By linearity,
$\alpha|0\rangle + \beta|1\rangle \xrightarrow{\text{NOT}} \alpha|1\rangle + \beta|0\rangle$

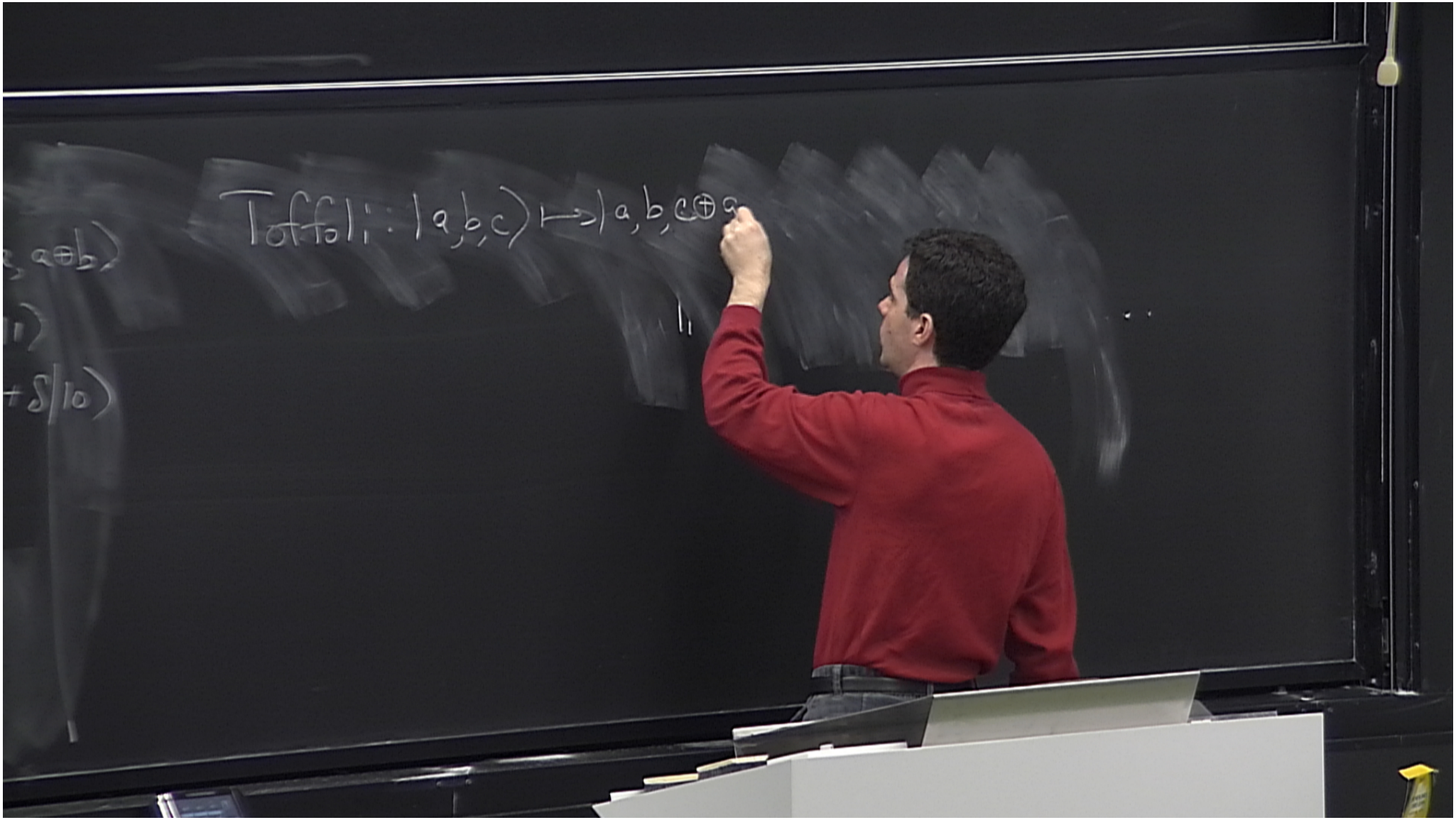

Quantum gates:
Make classical reversible gates quantum:
NOT: $|0\rangle \mapsto |1\rangle$, $|1\rangle \mapsto |0\rangle$
By linearity,
$\alpha|0\rangle+\beta|1\rangle \xrightarrow{NOT} \alpha|1\rangle+\beta|0\rangle$
CNOT: $|a,b\rangle \mapsto |a, a\oplus b\rangle$

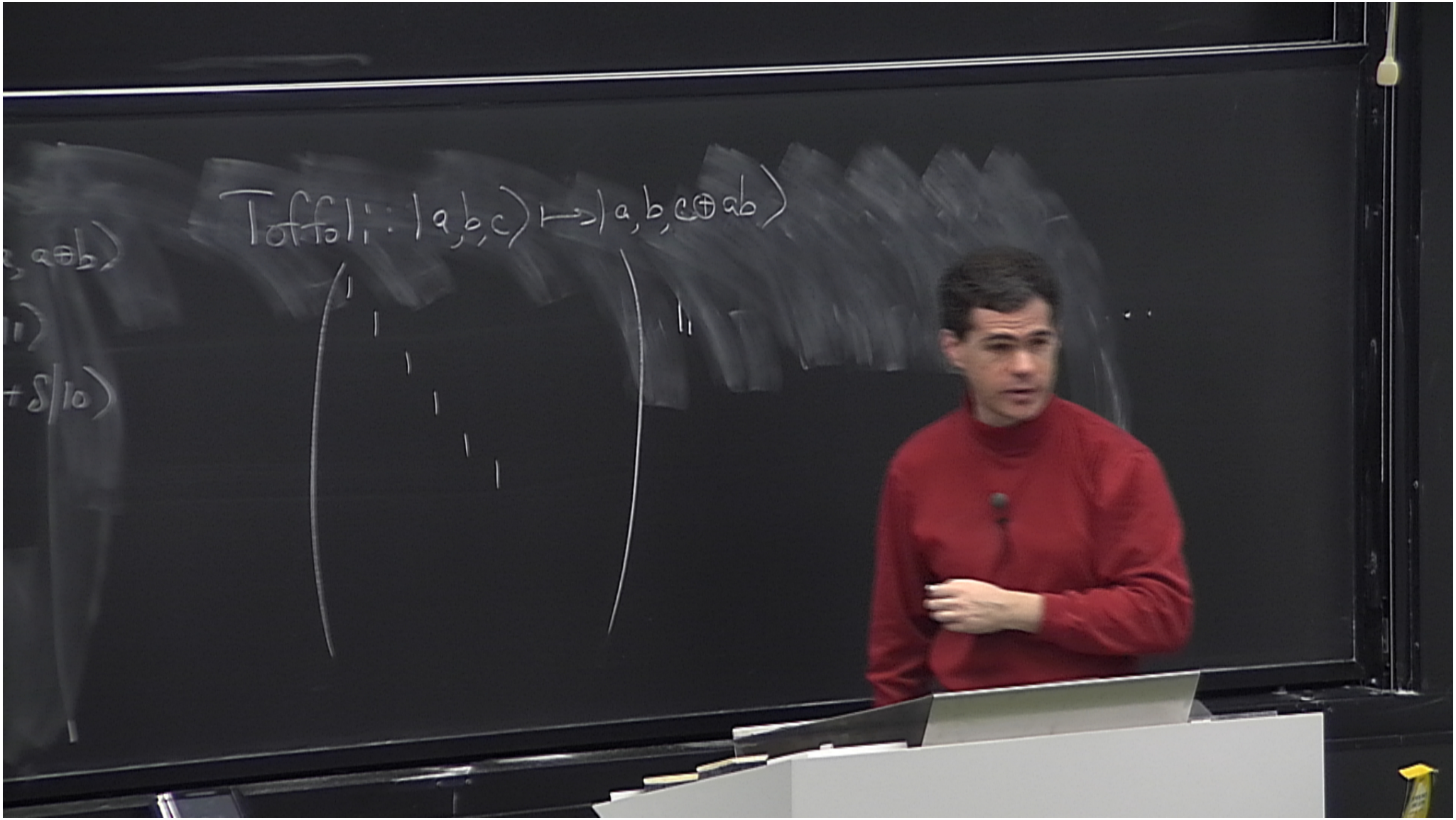

Quantum gates:
Make classical reversible gates quantum:
NOT: |0⟩ ↦ |1⟩, |1⟩ ↦ |0⟩
By linearity,
α|0⟩+β|1⟩ —NOT→ α|1⟩+β|0⟩
CNOT: |a,b⟩ ↦ |a, a⊕b⟩
α|00⟩+β|01⟩+γ|10⟩+δ|11⟩
↦ α|00⟩+β|01⟩+γ|11⟩+δ|10⟩

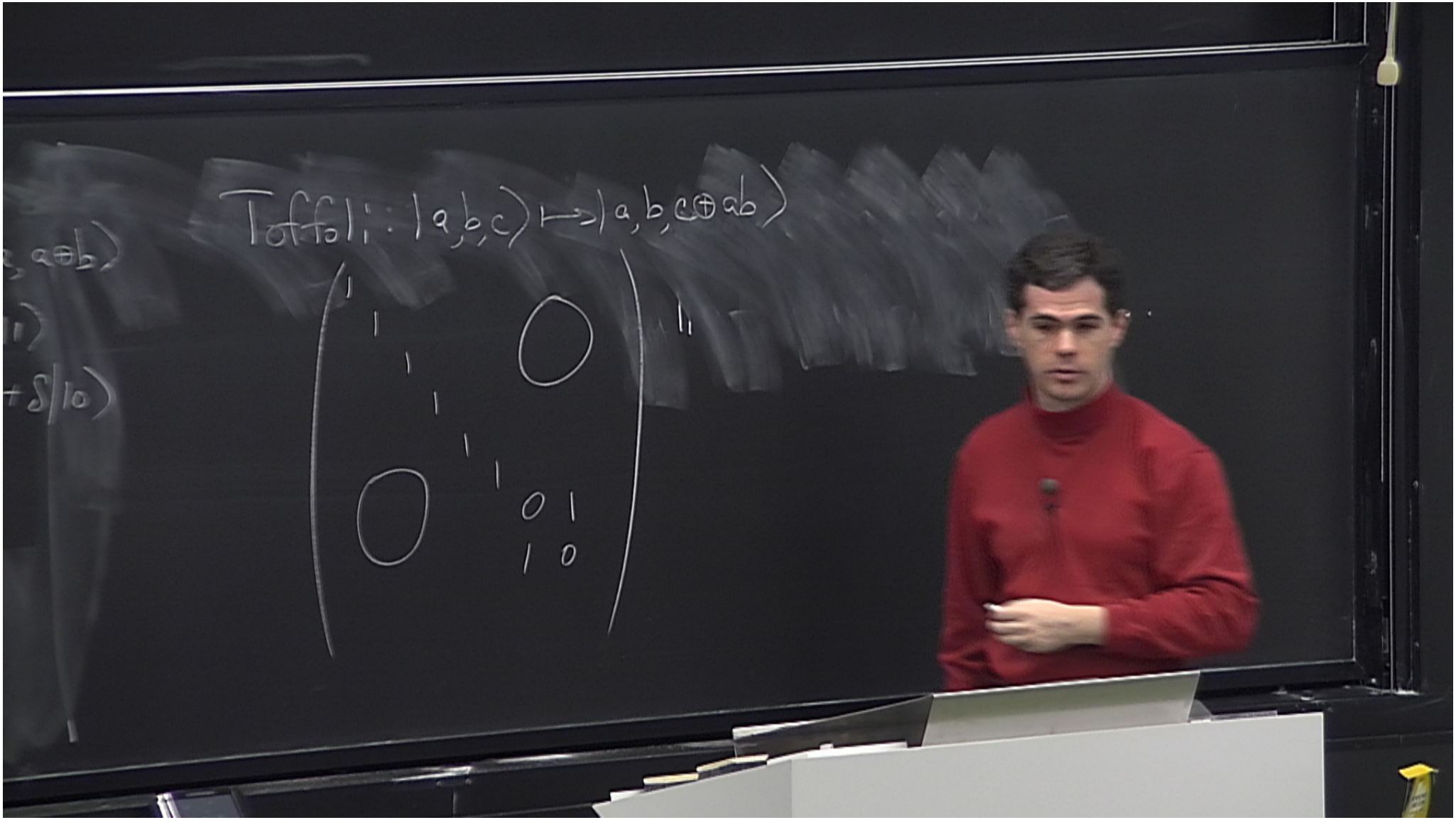

Quantum gates:
Make classical reversible gates quantum:
NOT: |0⟩ ↦ |1⟩, |1⟩ ↦ |0⟩
By linearity,
α|0⟩+β|1⟩ —NOT→ α|1⟩+β|0⟩
(0 1 / 1 0)
CNOT: |a,b⟩ ↦ |a, a⊕b⟩
α|00⟩+β|01⟩+γ|10⟩+δ|11⟩
↦ α|00⟩+β|01⟩+γ|11⟩+δ|10⟩

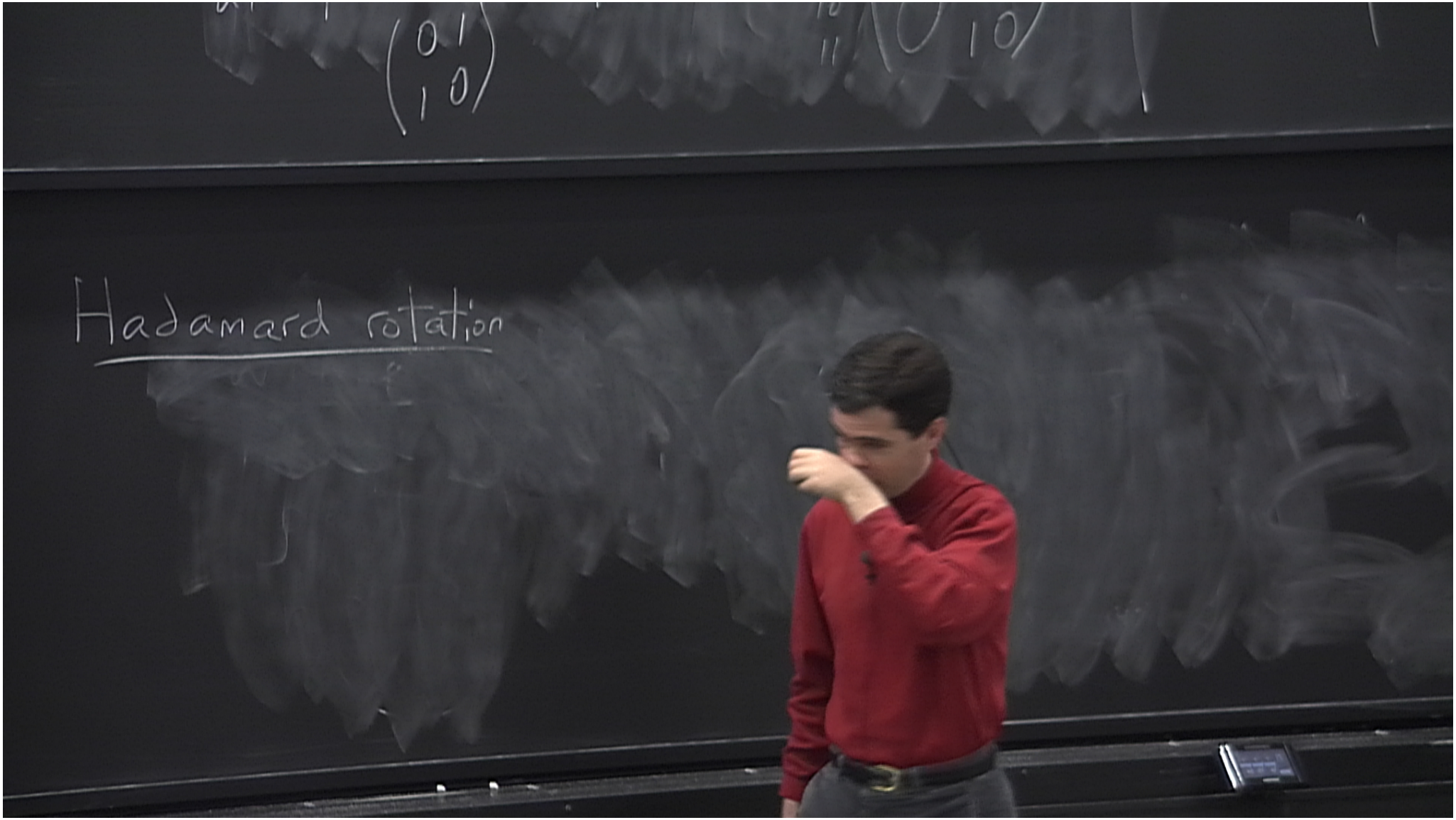

Quantum gates:
Make classical reversible gates quantum:
NOT: $|0\rangle \mapsto |1\rangle$, $|1\rangle \mapsto |0\rangle$
By linearity,
$\alpha|0\rangle+\beta|1\rangle \xrightarrow{\text{NOT}} \alpha|1\rangle+\beta|0\rangle$
$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
CNOT: $|a,b\rangle \mapsto |a, a\oplus b\rangle$
$\alpha|00\rangle+\beta|01\rangle+\gamma|10\rangle+\delta|11\rangle$
$\mapsto \alpha|00\rangle+\beta|01\rangle+\gamma|11\rangle+\delta|10\rangle$
$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

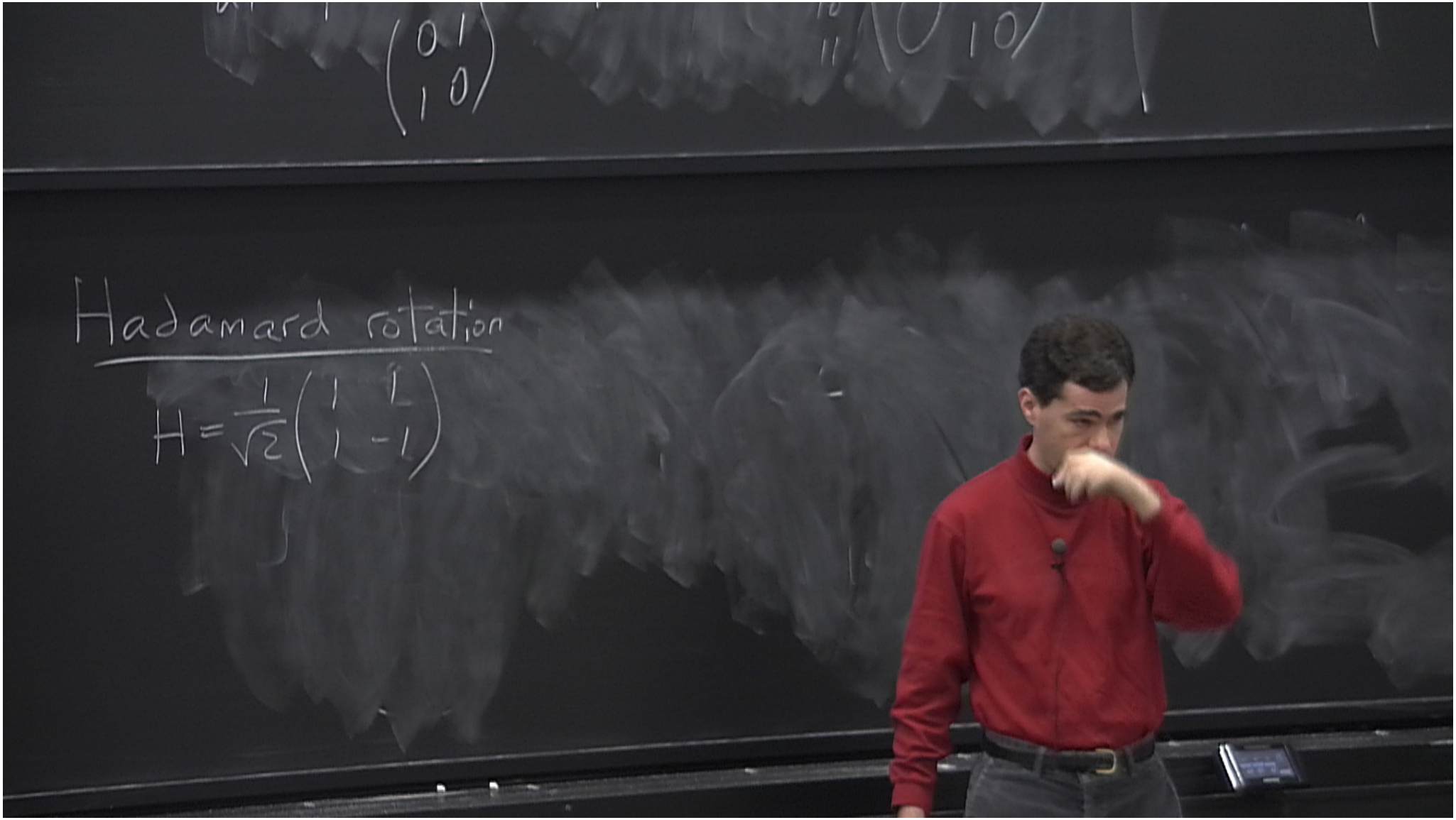

Quantum gates:
Make classical reversible gates quantum:
NOT: |0⟩ ↦ |1⟩, |1⟩ ↦ |0⟩
By linearity,
α|0⟩+β|1⟩ —NOT→ α|1⟩+β|0⟩
(0 1 / 1 0)
CNOT: |a,b⟩ ↦ |a, a⊕b⟩
α|00⟩+β|01⟩+γ|10⟩+δ|11⟩
↦ α|00⟩+β|01⟩+γ|11⟩+δ|10⟩
00 01 10 11
(1 0 0 0 / 0 1 0 0 / 0 0 0 1 / 0 0 1 0)

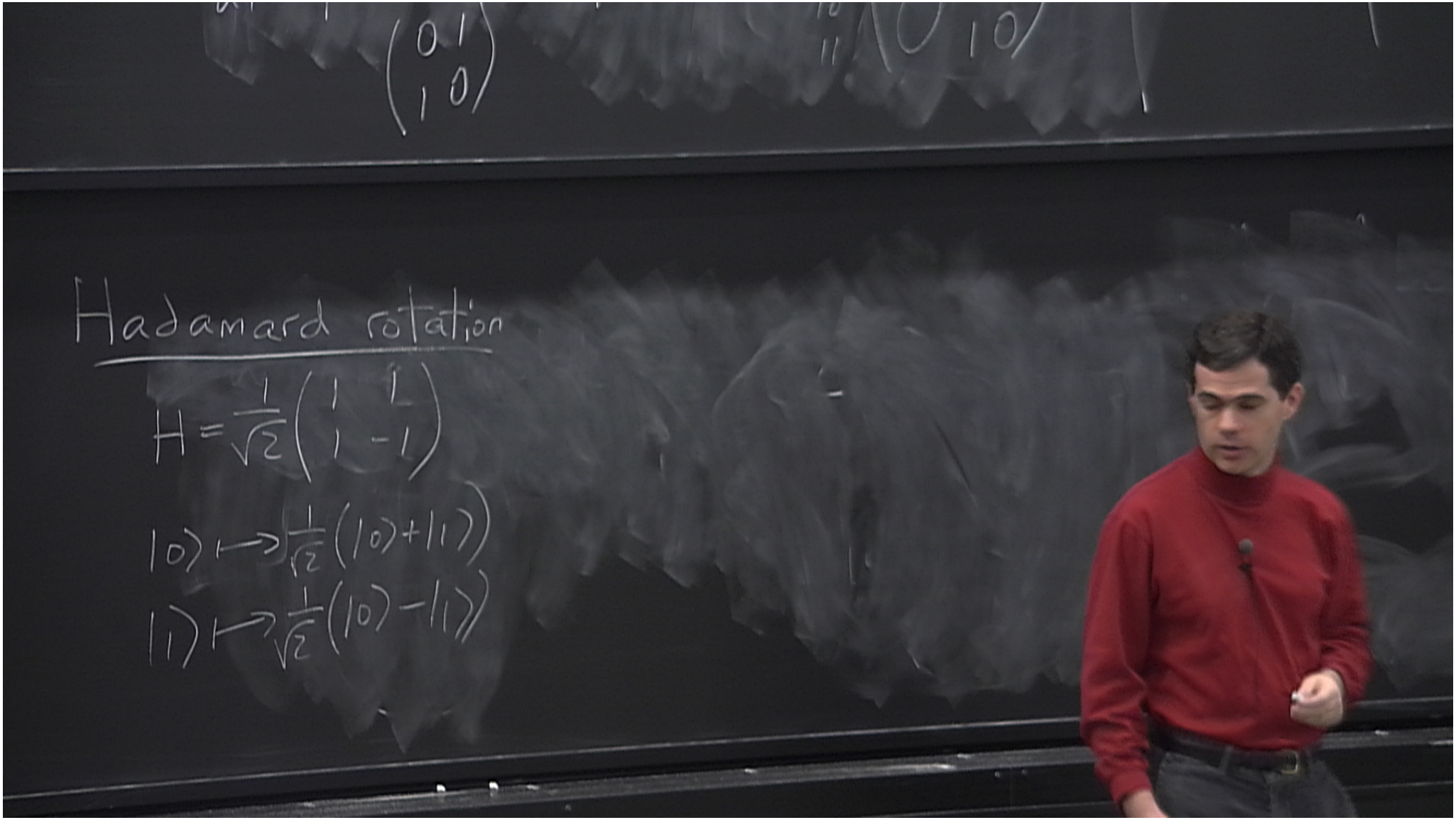

Toffoli : |a,b,c⟩ ↦ |a,b,c⊕

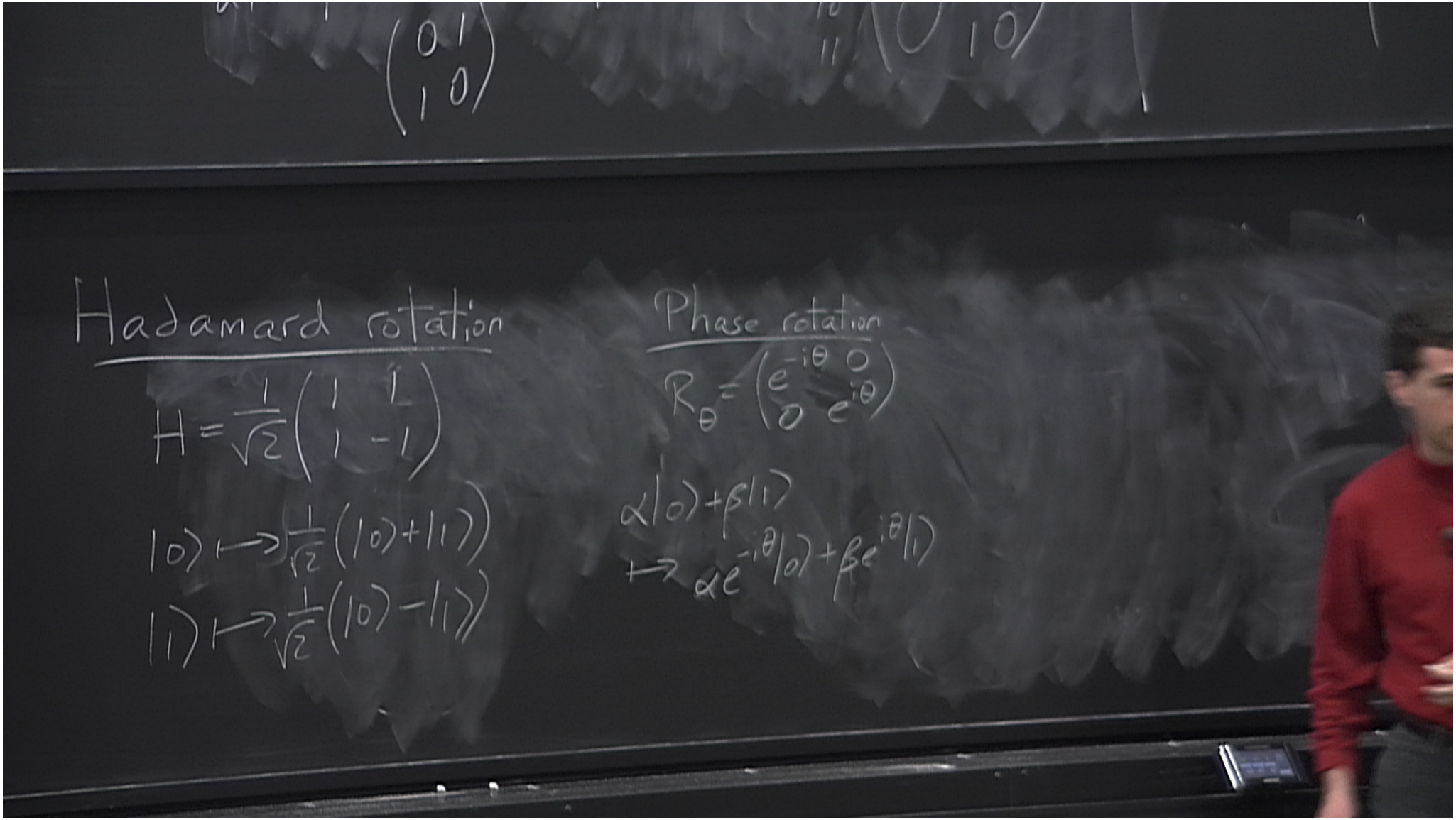

Toffoli : |a,b,c⟩ ↦ |a,b,c⊕ab⟩

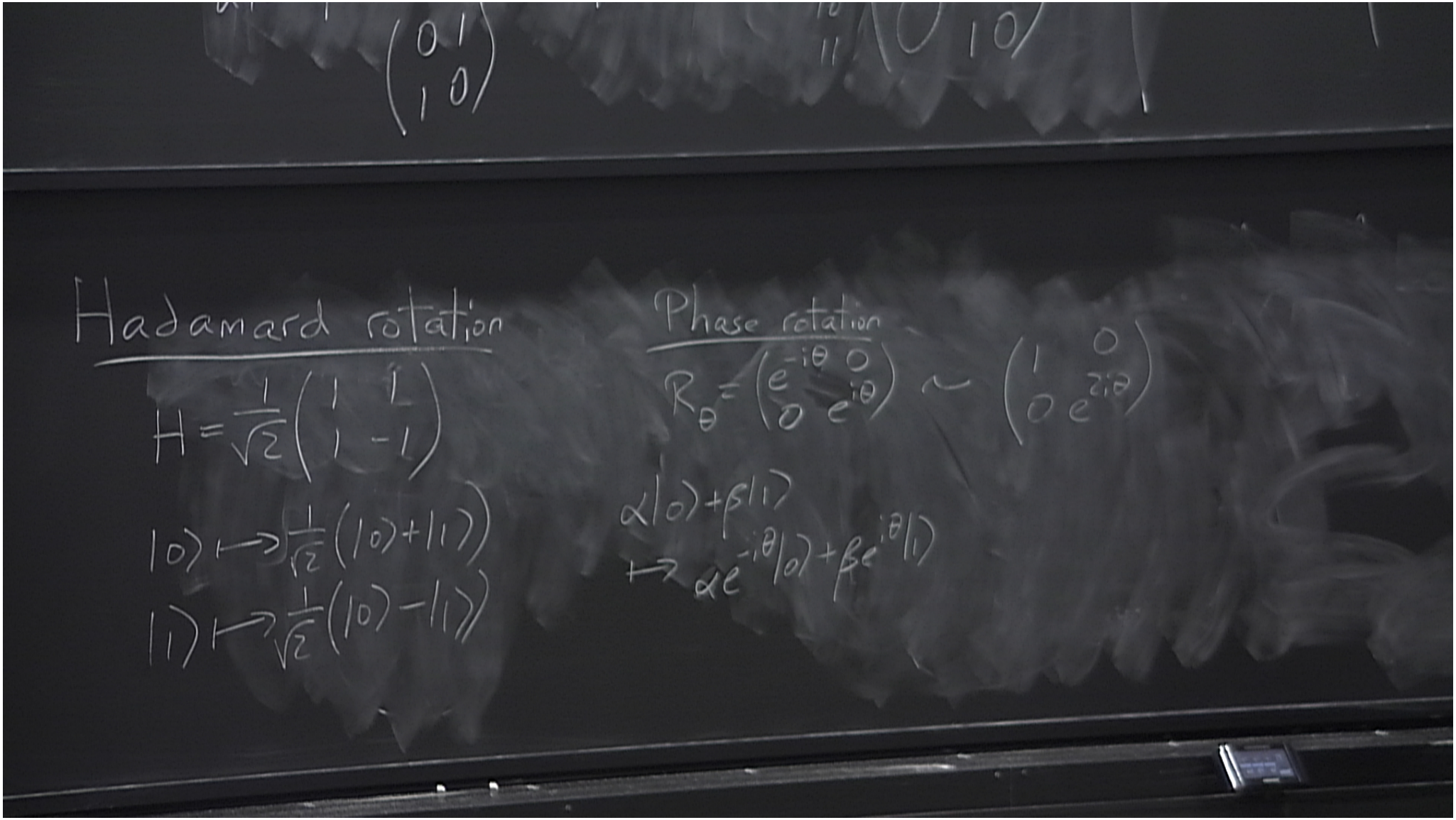

Toffoli: |a,b,c⟩ ↦ |a,b,c⊕ab⟩

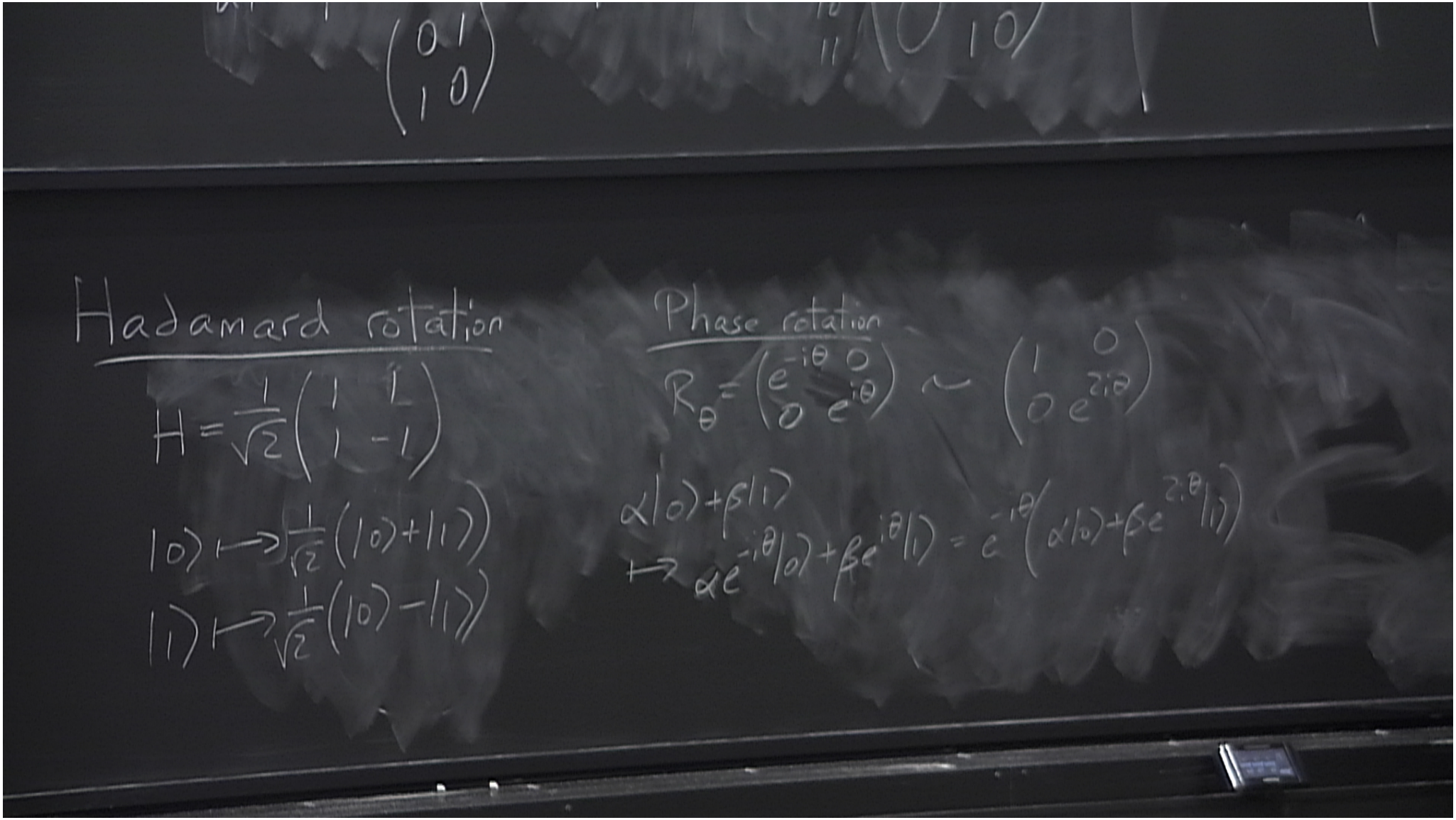

Hadamard rotation

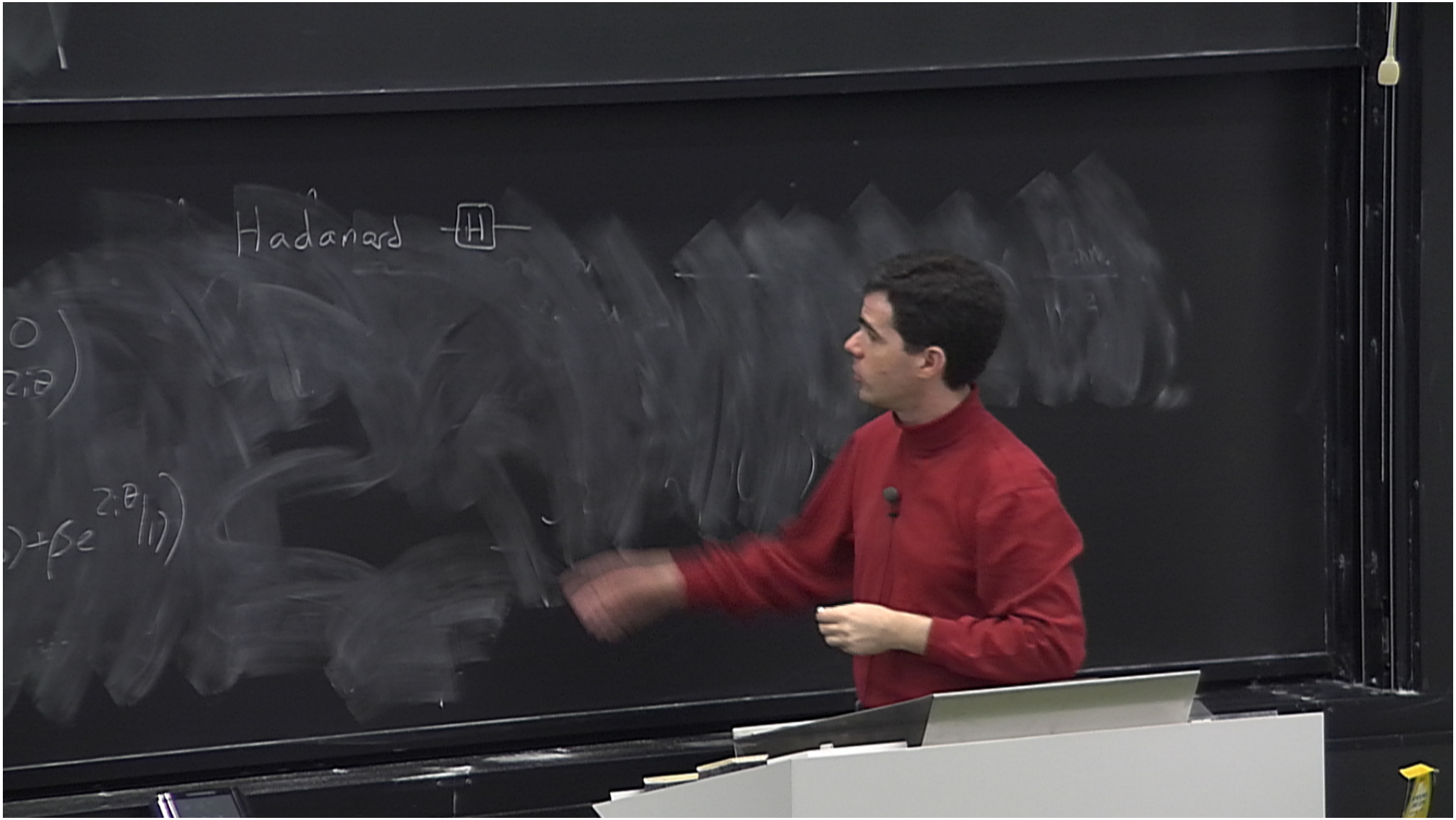

Hadamard rotation
$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

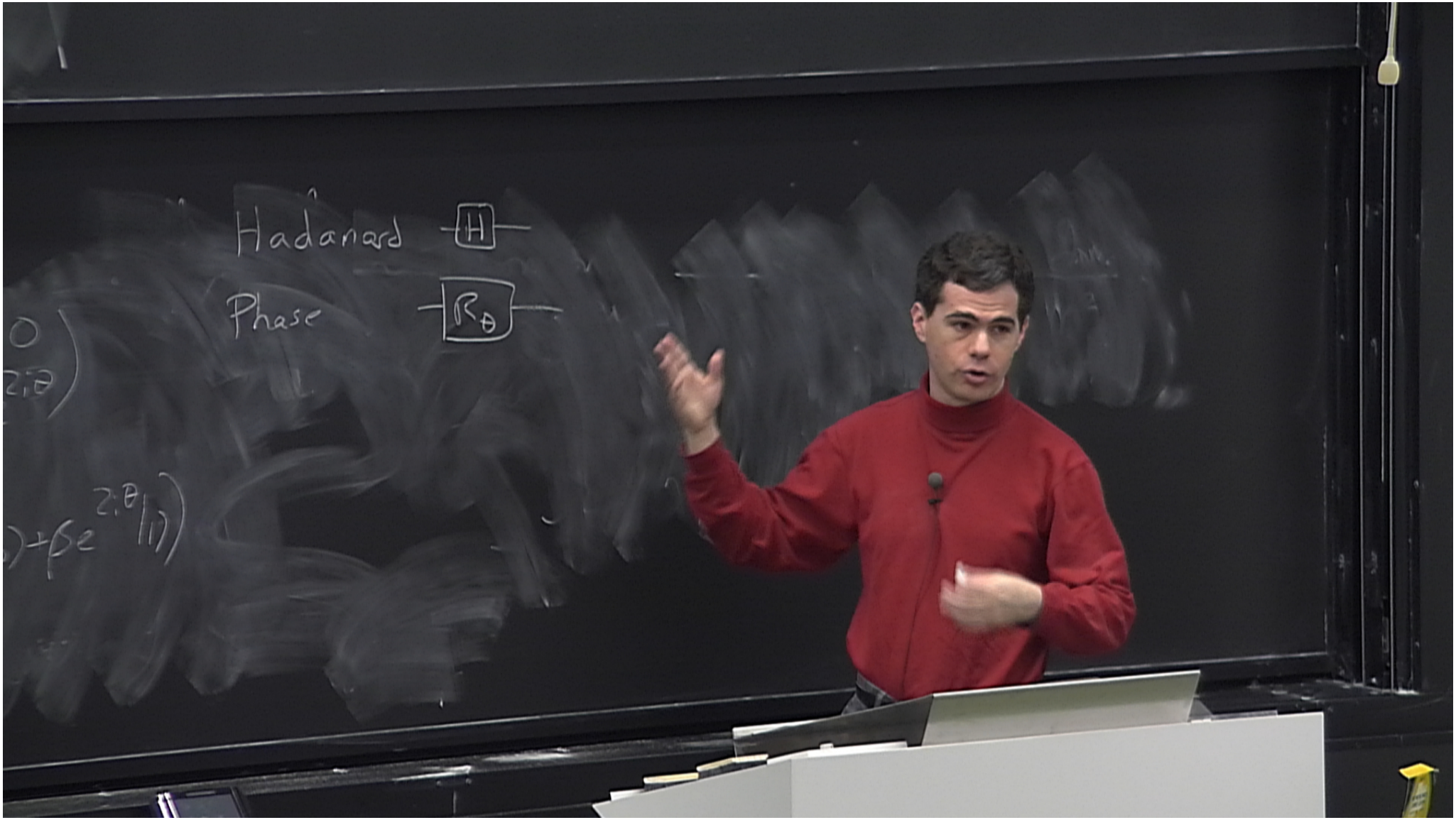

Hadamard rotation
$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
$|0\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
$|1\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

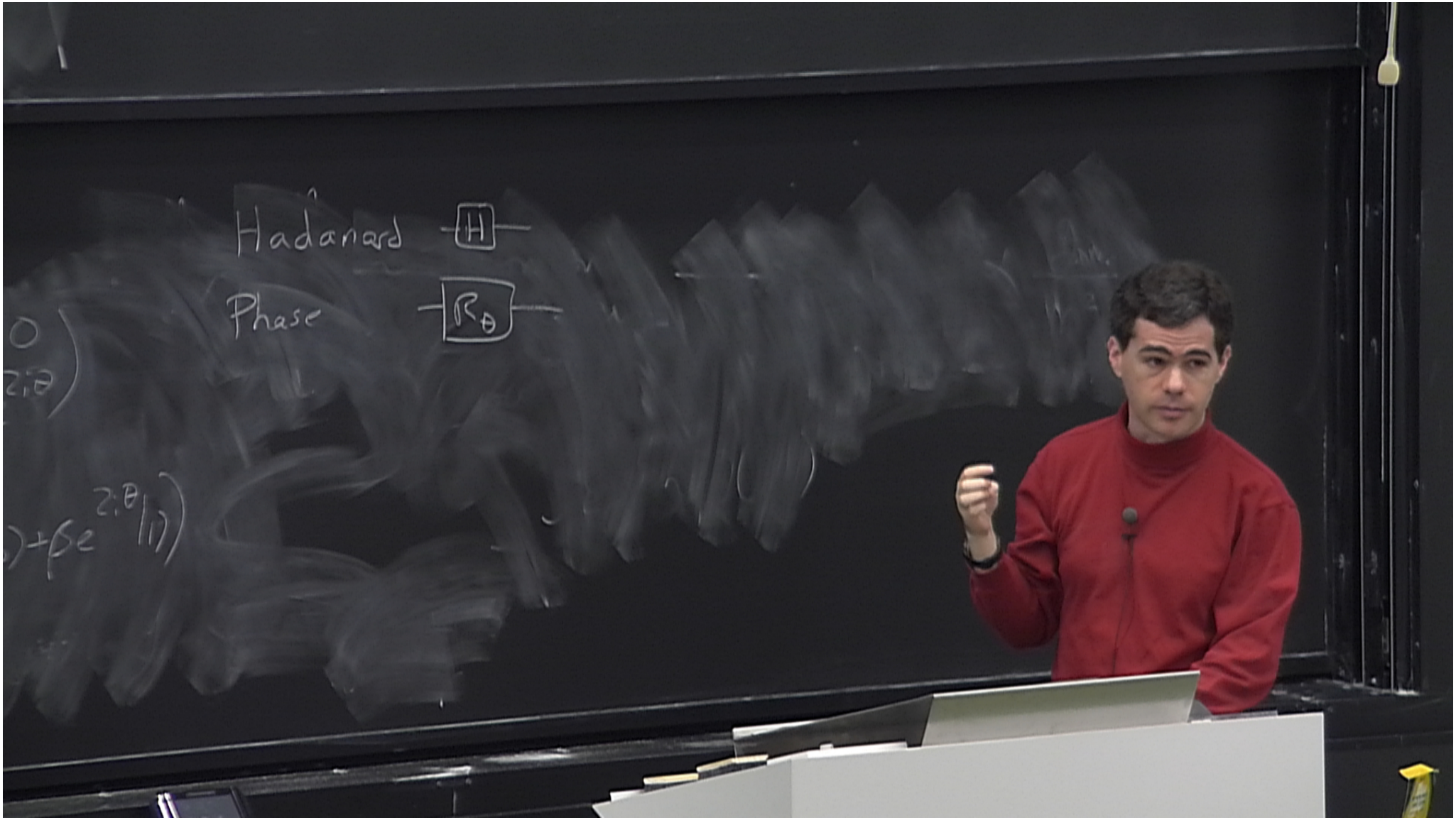

Hadamard rotation
$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
$|0\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
$|1\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
Phase rotation
$R_\theta = \begin{pmatrix} e^{-i\theta} & 0 \\ 0 & e^{i\theta} \end{pmatrix}$
$\alpha|0\rangle + \beta|1\rangle$
$\mapsto \alpha e^{-i\theta}|0\rangle + \beta e^{i\theta}|1\rangle$

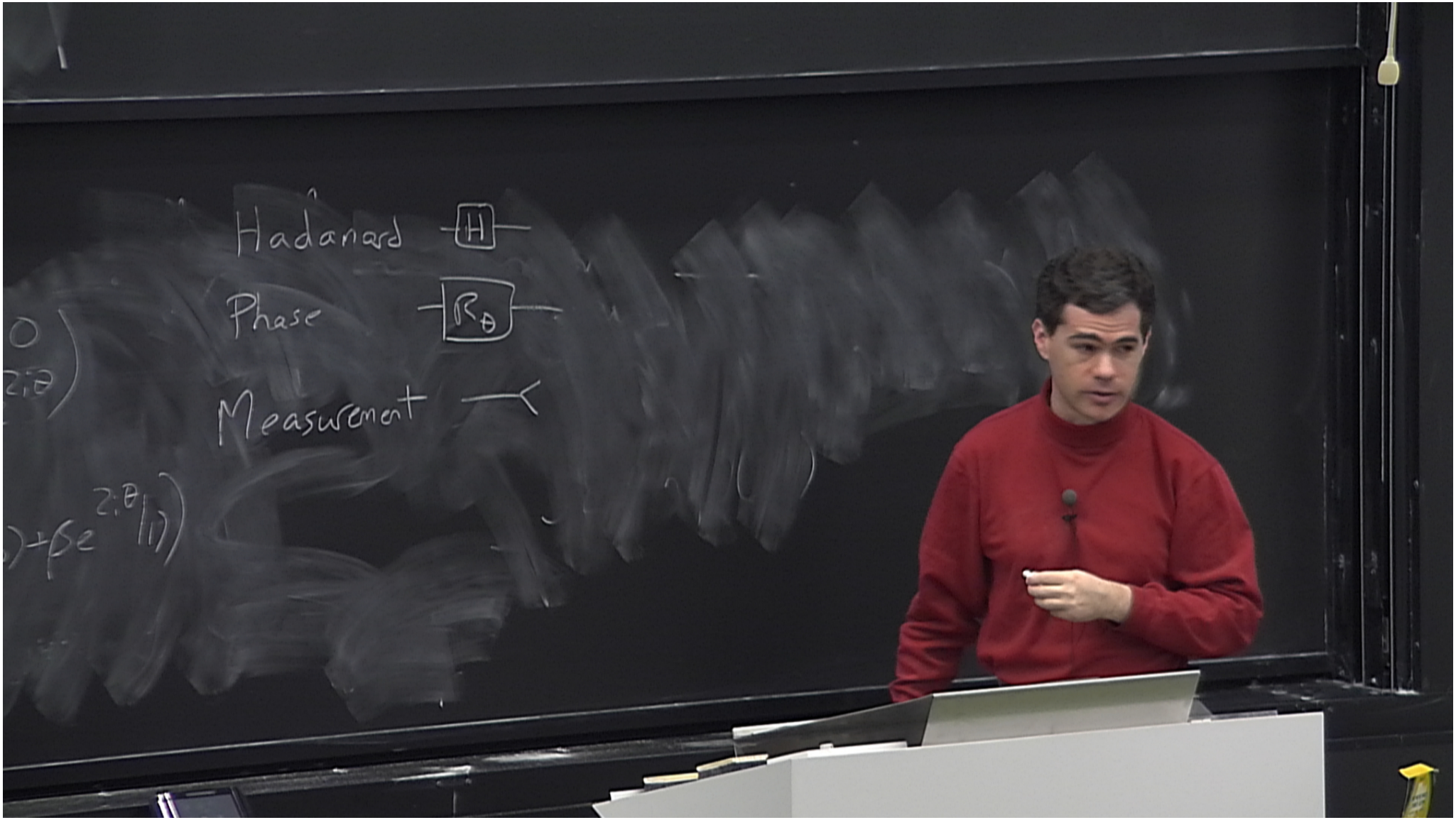

Hadamard rotation
$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
$|0\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
$|1\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
Phase rotation
$R_\theta = \begin{pmatrix} e^{-i\theta} & 0 \\ 0 & e^{i\theta} \end{pmatrix} \sim \begin{pmatrix} 1 & 0 \\ 0 & e^{2i\theta} \end{pmatrix}$
$\alpha|0\rangle + \beta|1\rangle$
$\mapsto \alpha e^{-i\theta}|0\rangle + \beta e^{i\theta}|1\rangle$

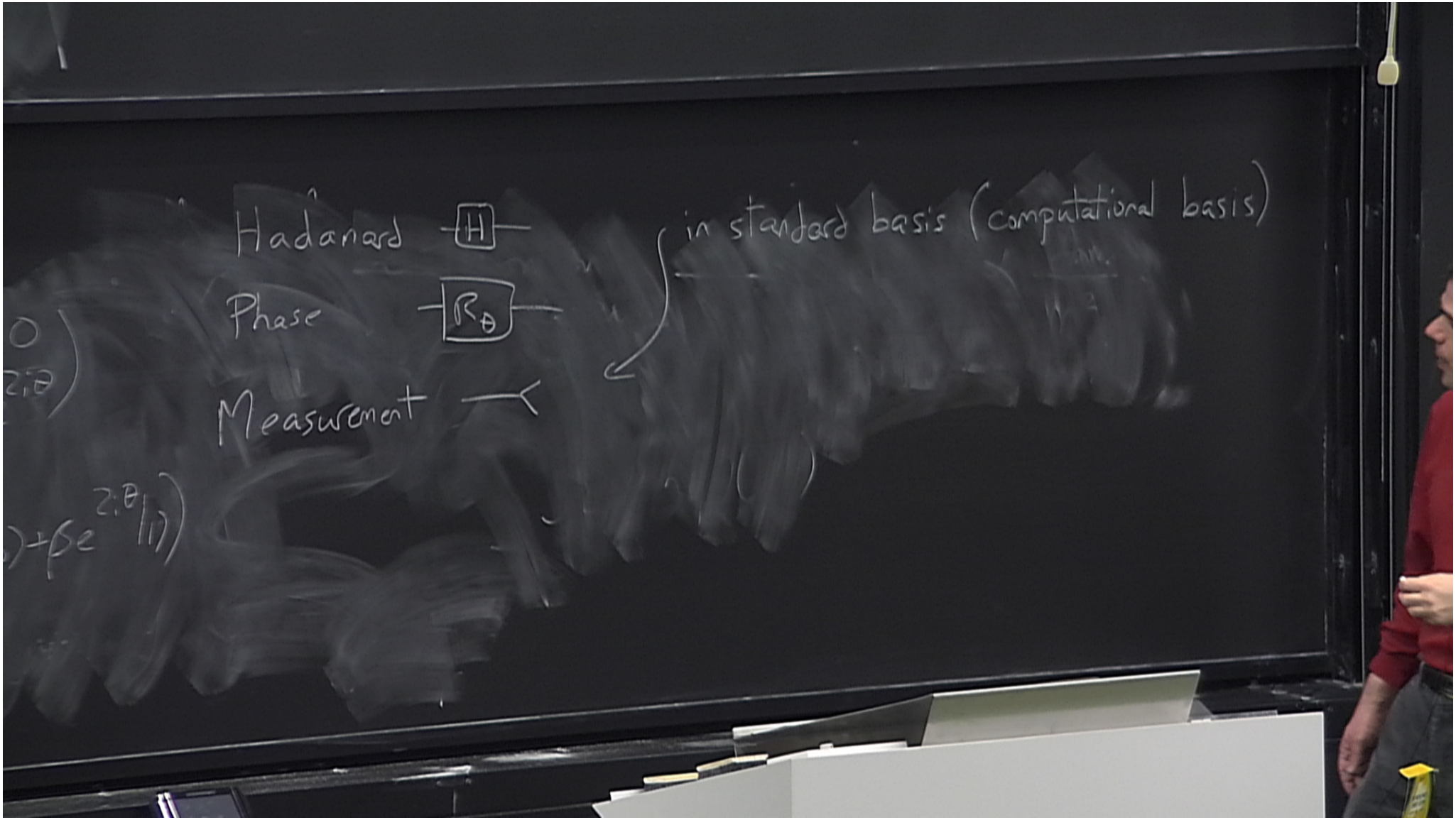

Hadamard rotation
$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
$|0\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
$|1\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
Phase rotation
$R_\theta = \begin{pmatrix} e^{-i\theta} & 0 \\ 0 & e^{i\theta} \end{pmatrix} \sim \begin{pmatrix} 1 & 0 \\ 0 & e^{2i\theta} \end{pmatrix}$
$\alpha|0\rangle + \beta|1\rangle$
$\mapsto \alpha e^{-i\theta}|0\rangle + \beta e^{i\theta}|1\rangle = e^{-i\theta}\left(\alpha|0\rangle + \beta e^{2i\theta}|1\rangle\right)$

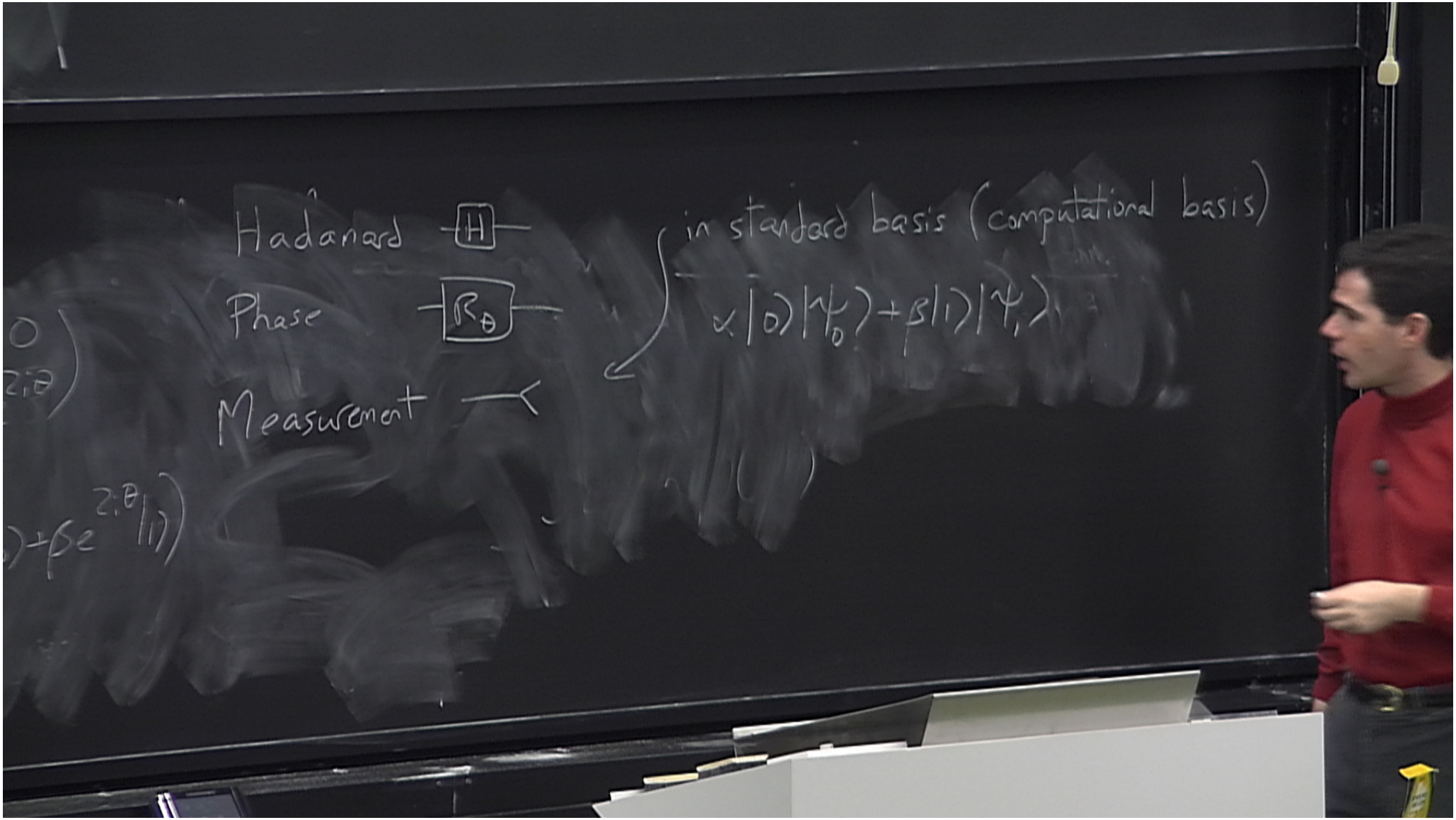

Hadanard
H

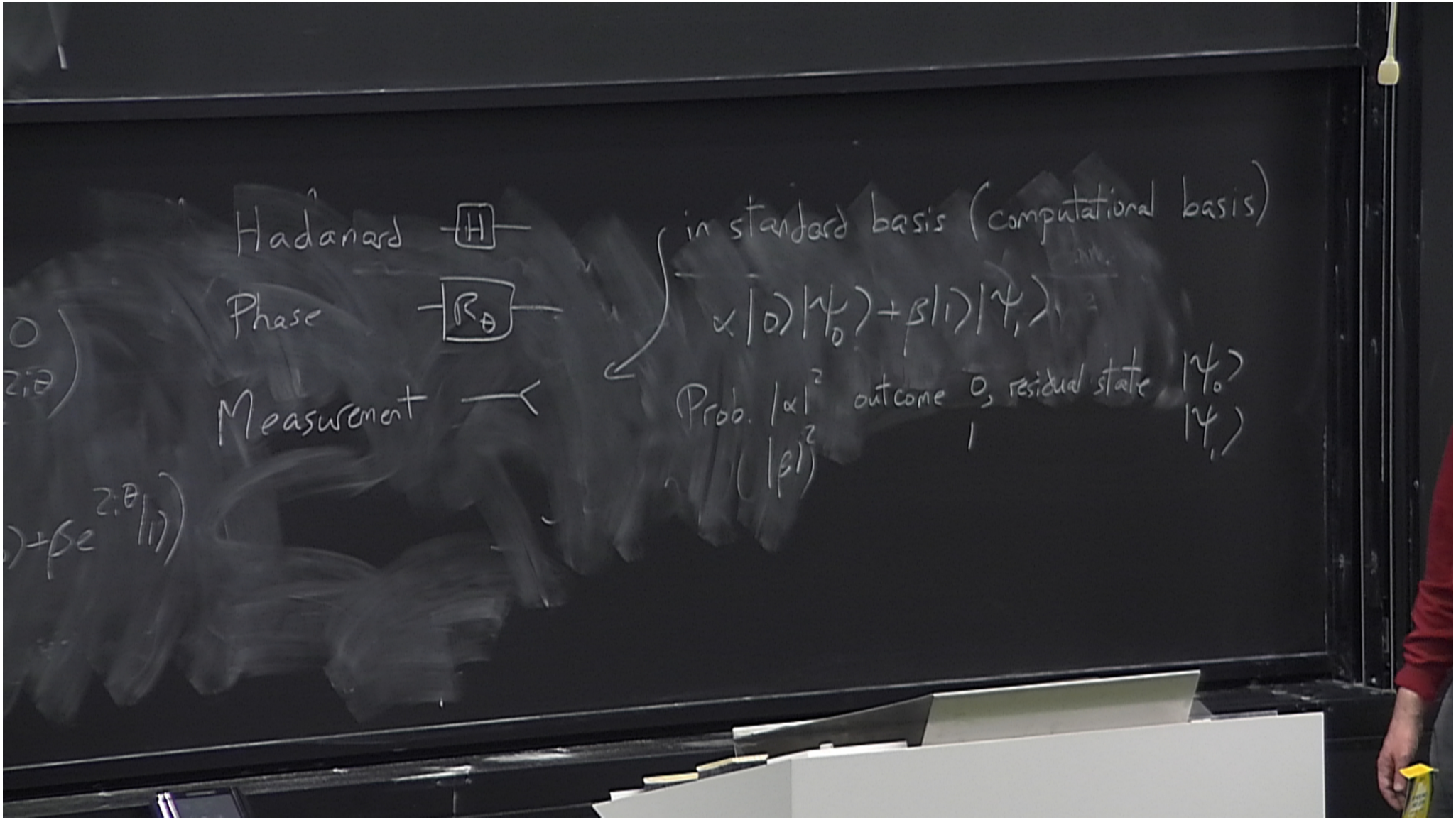

Hadamard
H
Phase
R_θ

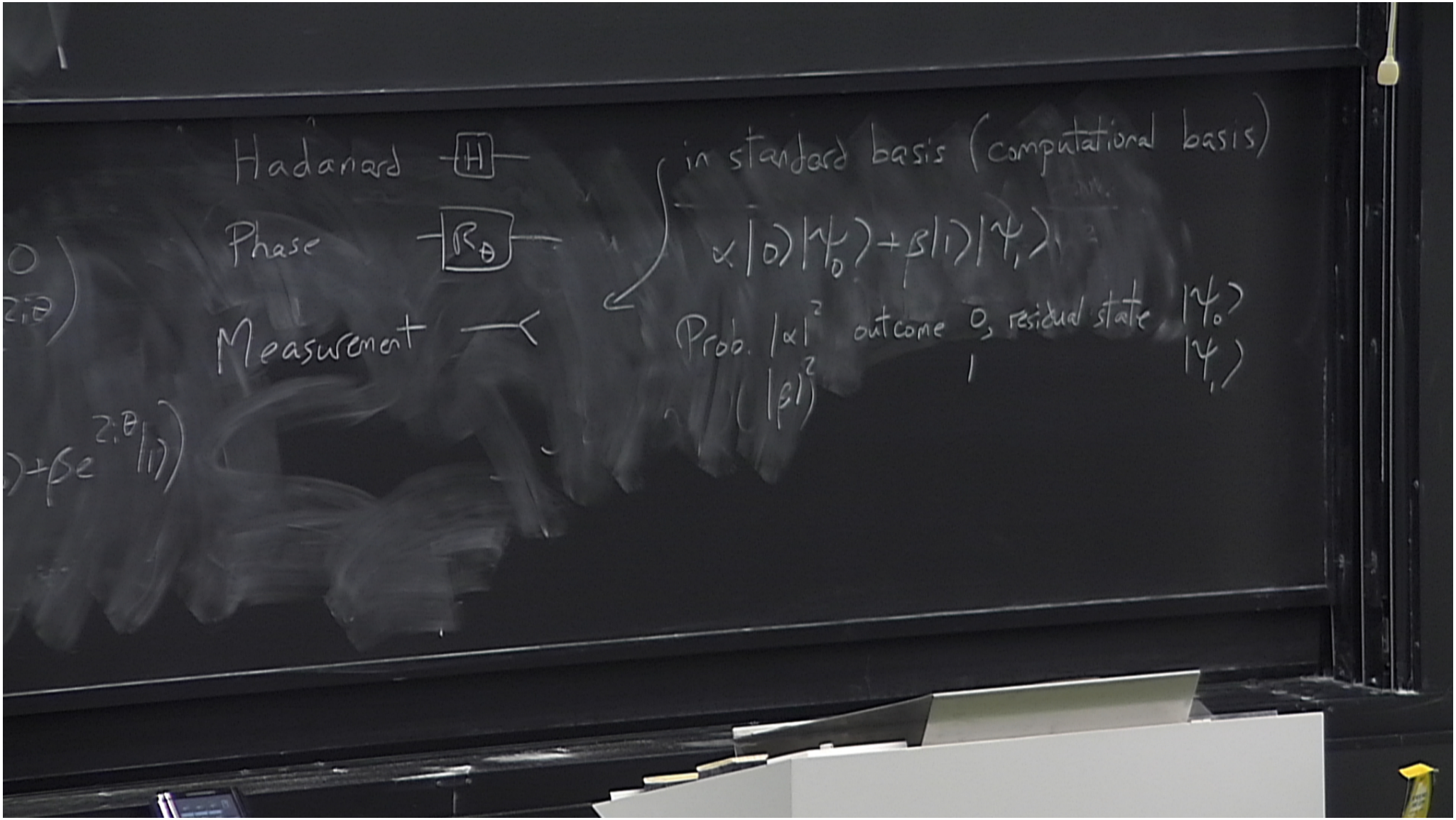

Hadamard
H
Phase
R_θ

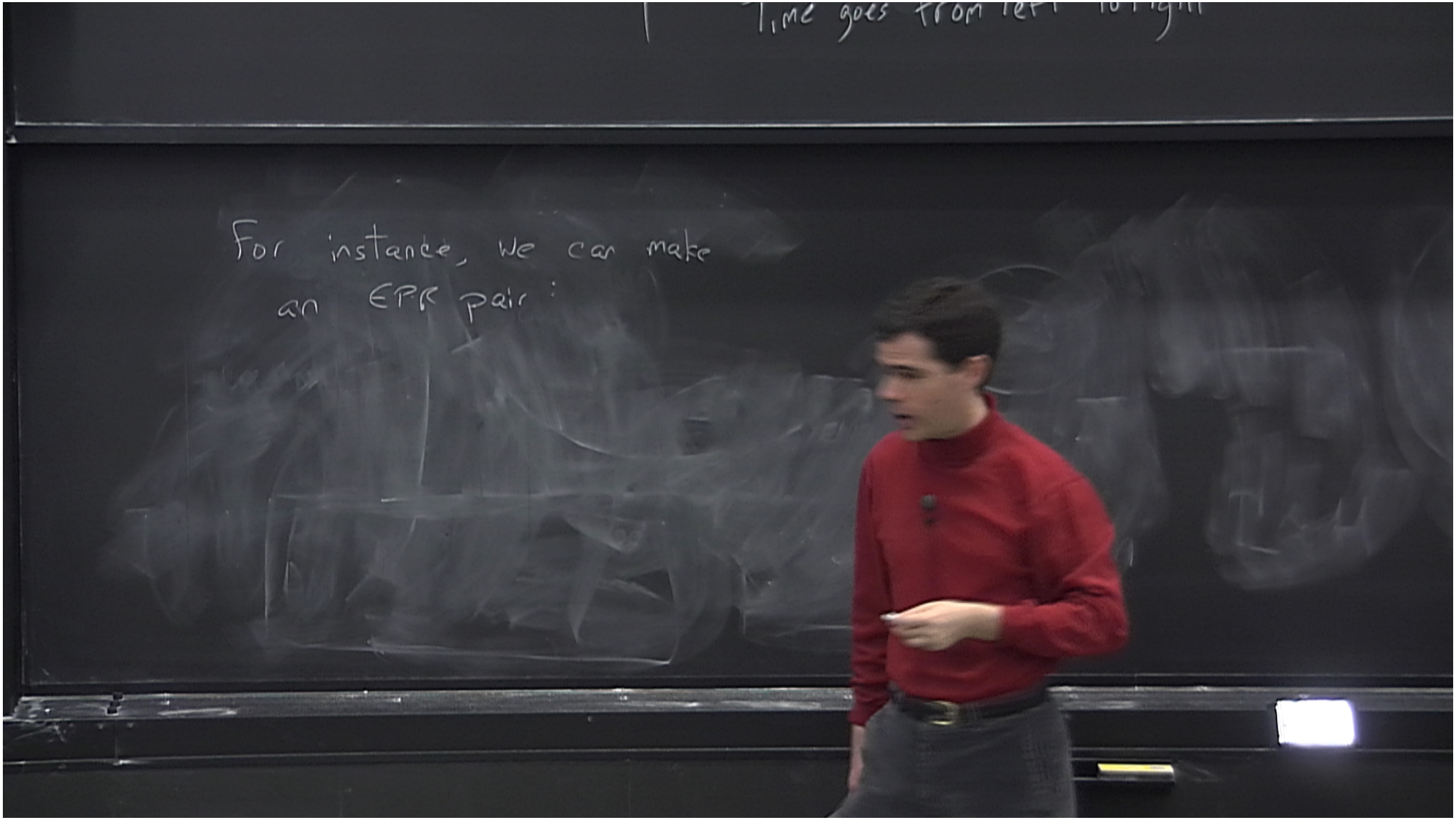

Hadamard
H
Phase
R_θ
Measurement

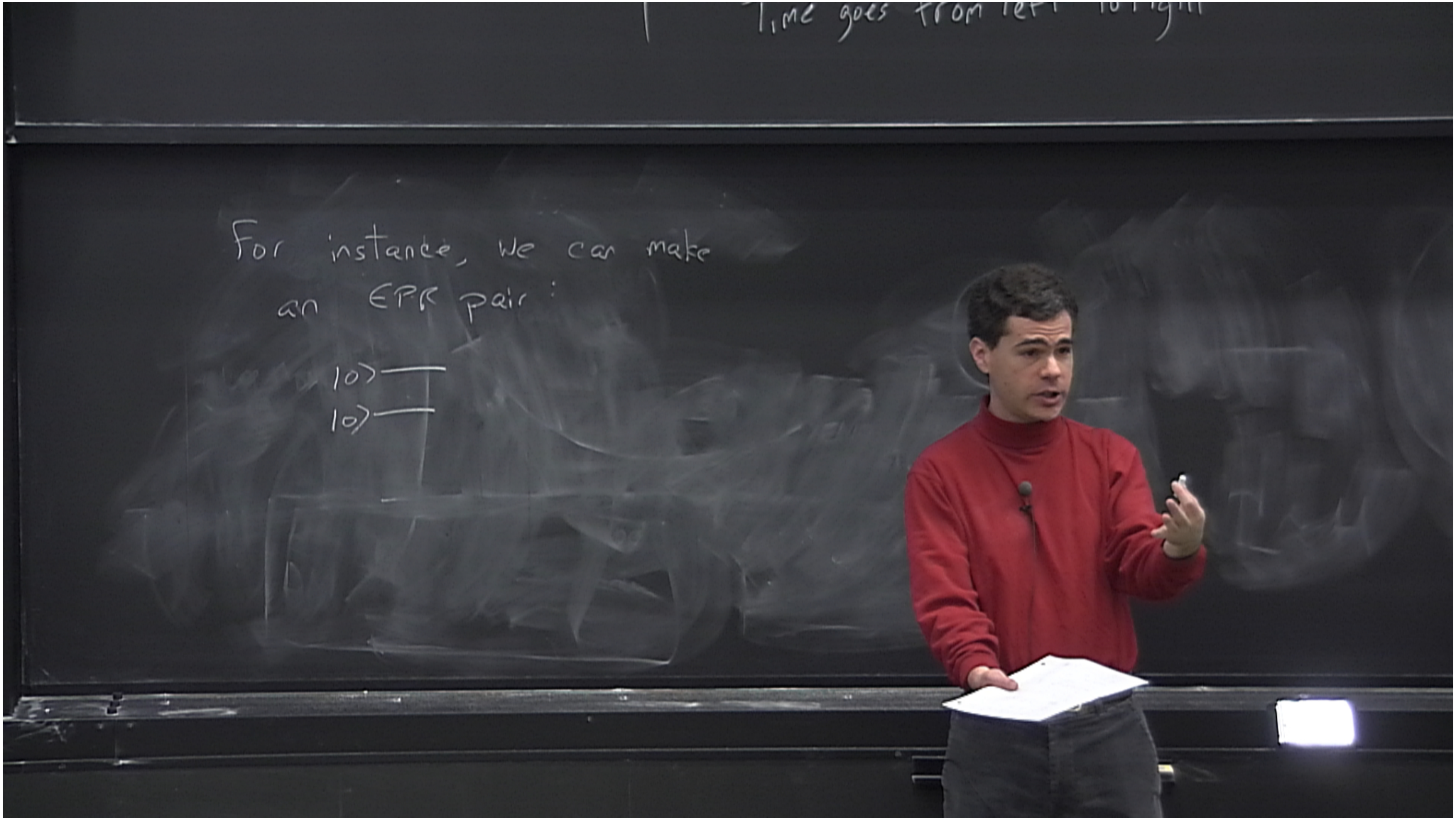

Hadamard
H
Phase
R_θ
Measurement
in standard basis (computational basis)

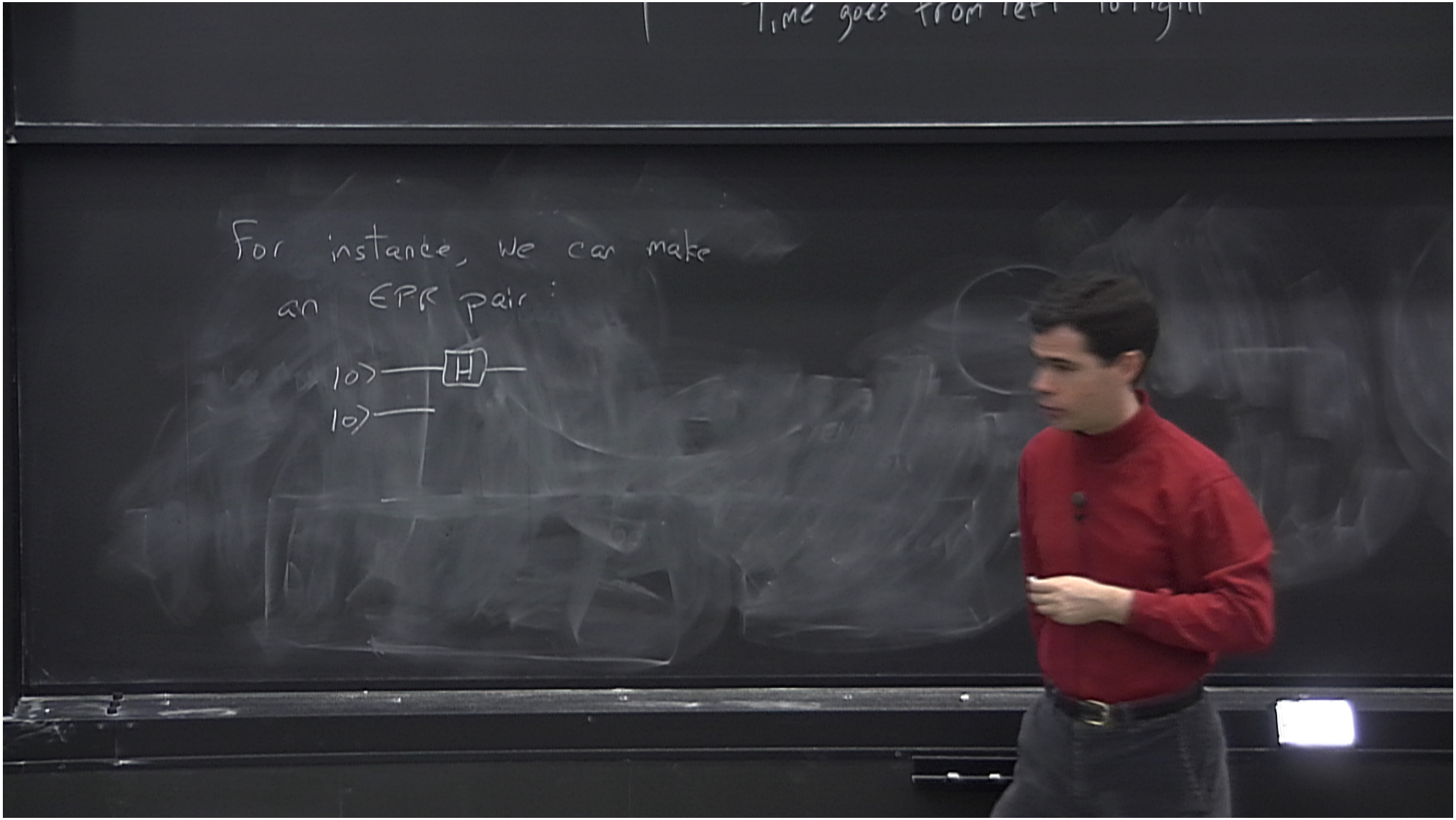

Hadamard
Phase
Measurement
in standard basis (computational basis)

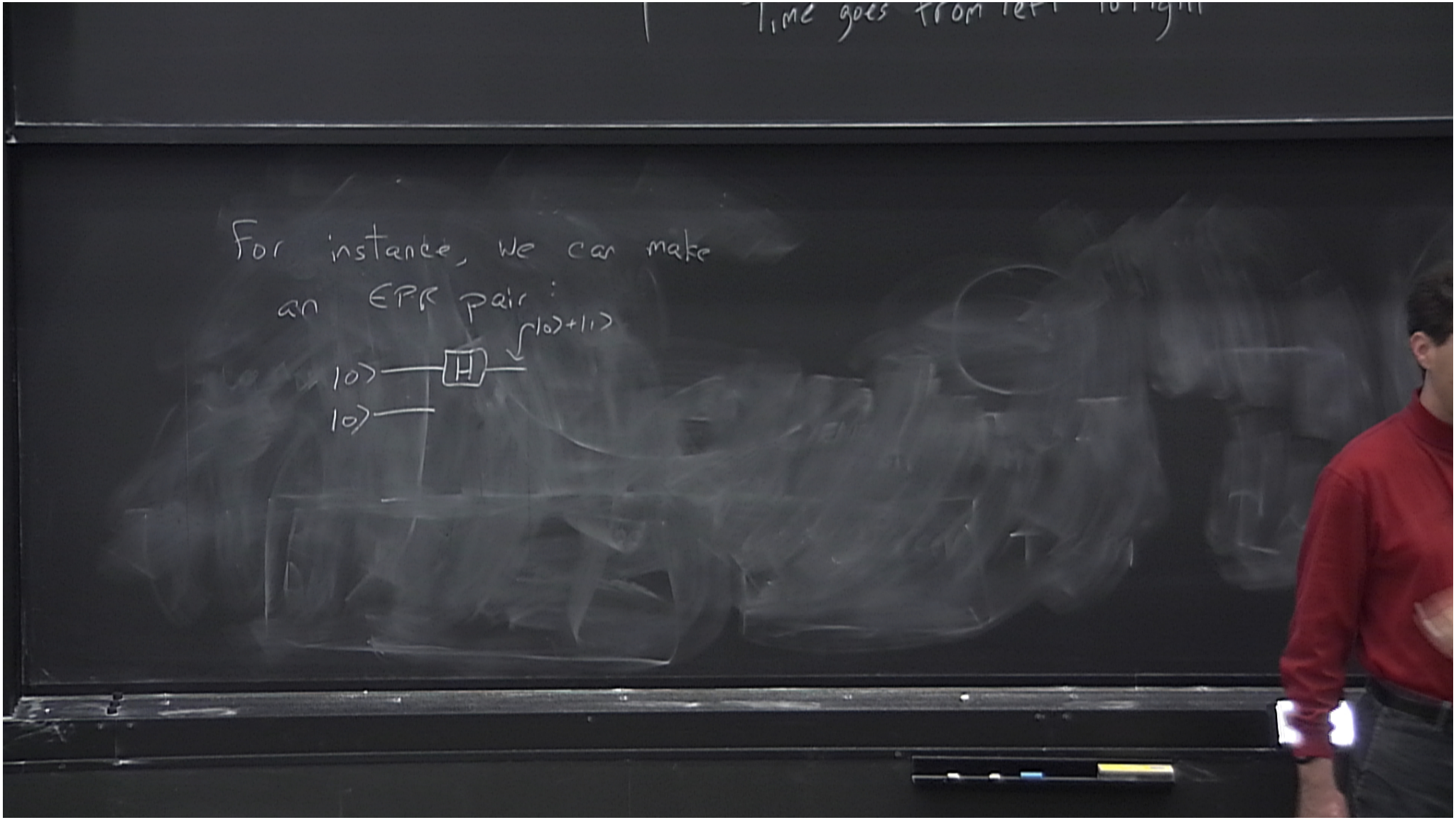

Hadamard H
Phase R_θ
Measurement
in standard basis (computational basis)
$\alpha|0\rangle|\psi_0\rangle + \beta|1\rangle|\psi_1\rangle$
Prob. $|\alpha|^2$ outcome 0, residual state $|\psi_0\rangle$
$(|\beta|^2)$ 1 $|\psi_1\rangle$


Hadamard
H
Phase
R_θ
Measurement
in standard basis (computational basis)
$\alpha|0\rangle|\psi_0\rangle + \beta|1\rangle|\psi_1\rangle$
Prob. $|\alpha|^2$ outcome 0, residual state $|\psi_0\rangle$
$|\beta|^2$ 1 $|\psi_1\rangle$

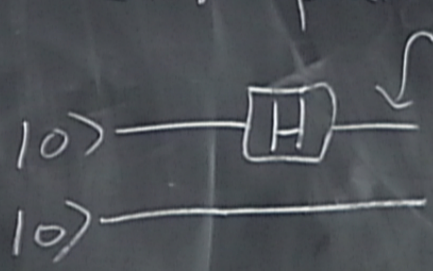

Time goes from
For instance, we can make
an EPR pair:


Time goes from
For instance, we can make
an EPR pair:
|0⟩
|0⟩

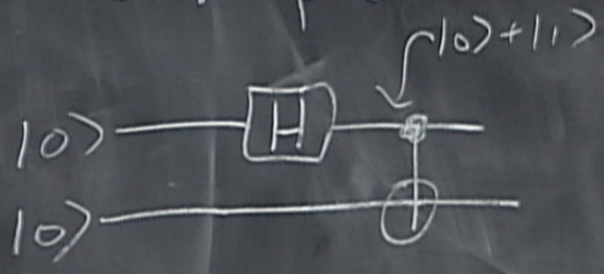

Time goes from
For instance, we can make
an EPR pair:
|0⟩
|0⟩
H

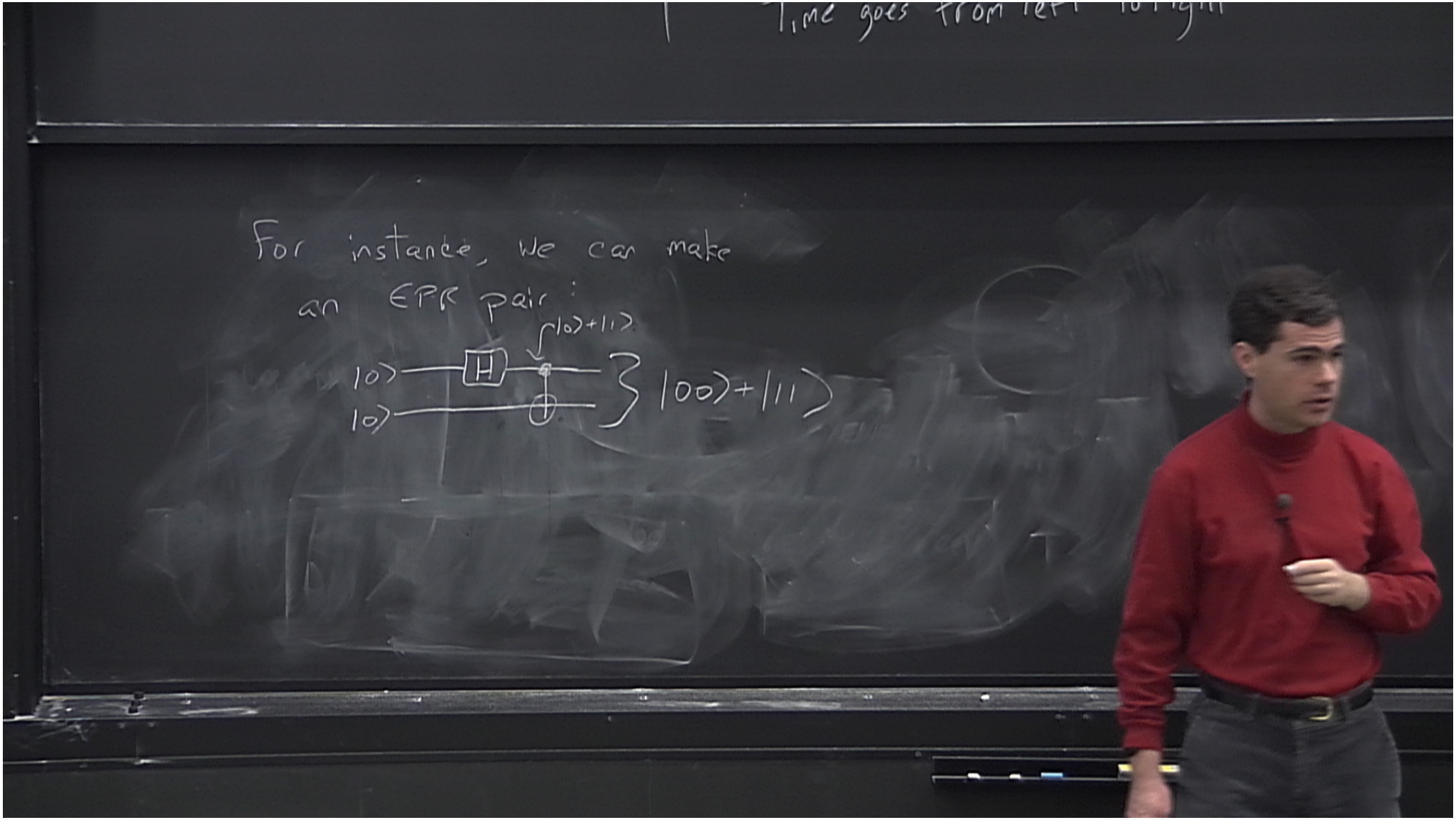

Time goes from left to right
For instance, we can make an EPR pair:
|0⟩+|1⟩
H
|0⟩
|0⟩

For instance, we can make an EPR pair:



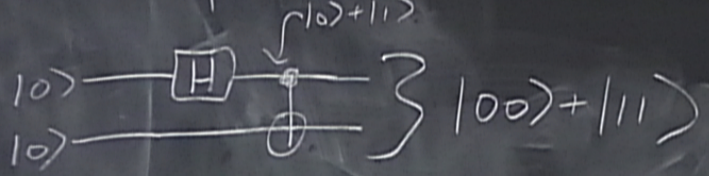
For instance, we can make an EPR pair:




Time goes from left to right
For instance, we can make an EPR pair:
|0⟩+|1⟩
H
|0⟩
|0⟩
|00⟩+|11⟩

Time goes from left to right

For instance, we can make an EPR pair:

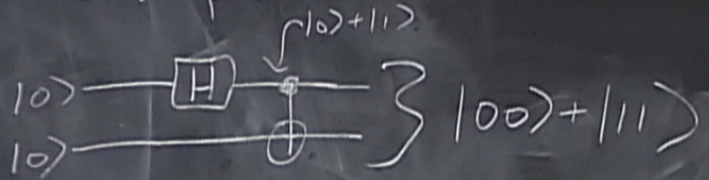


Calculate (a AND b) AND c on superposition

$(|000\rangle + |110\rangle)$

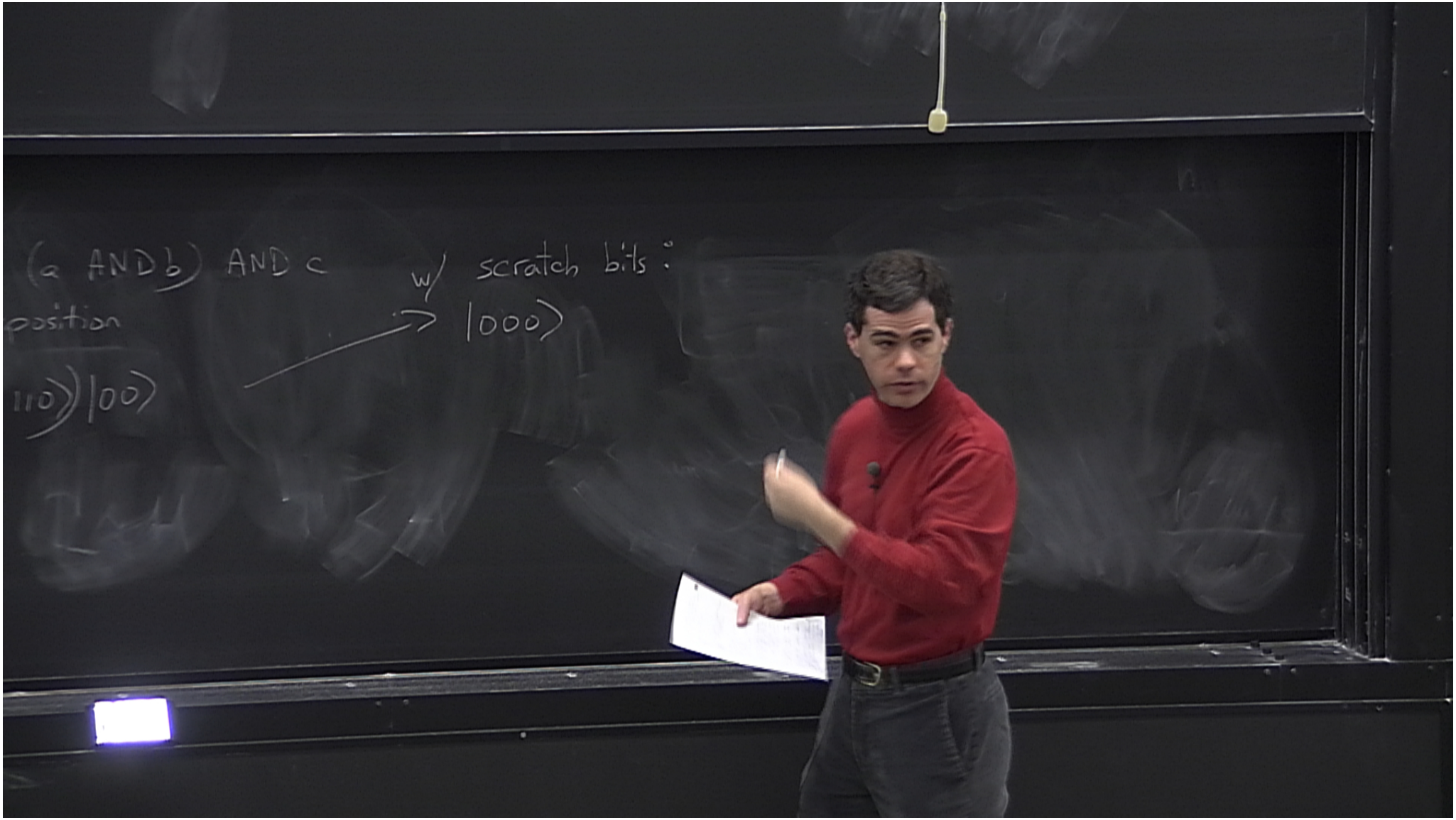
Time goes from left to right

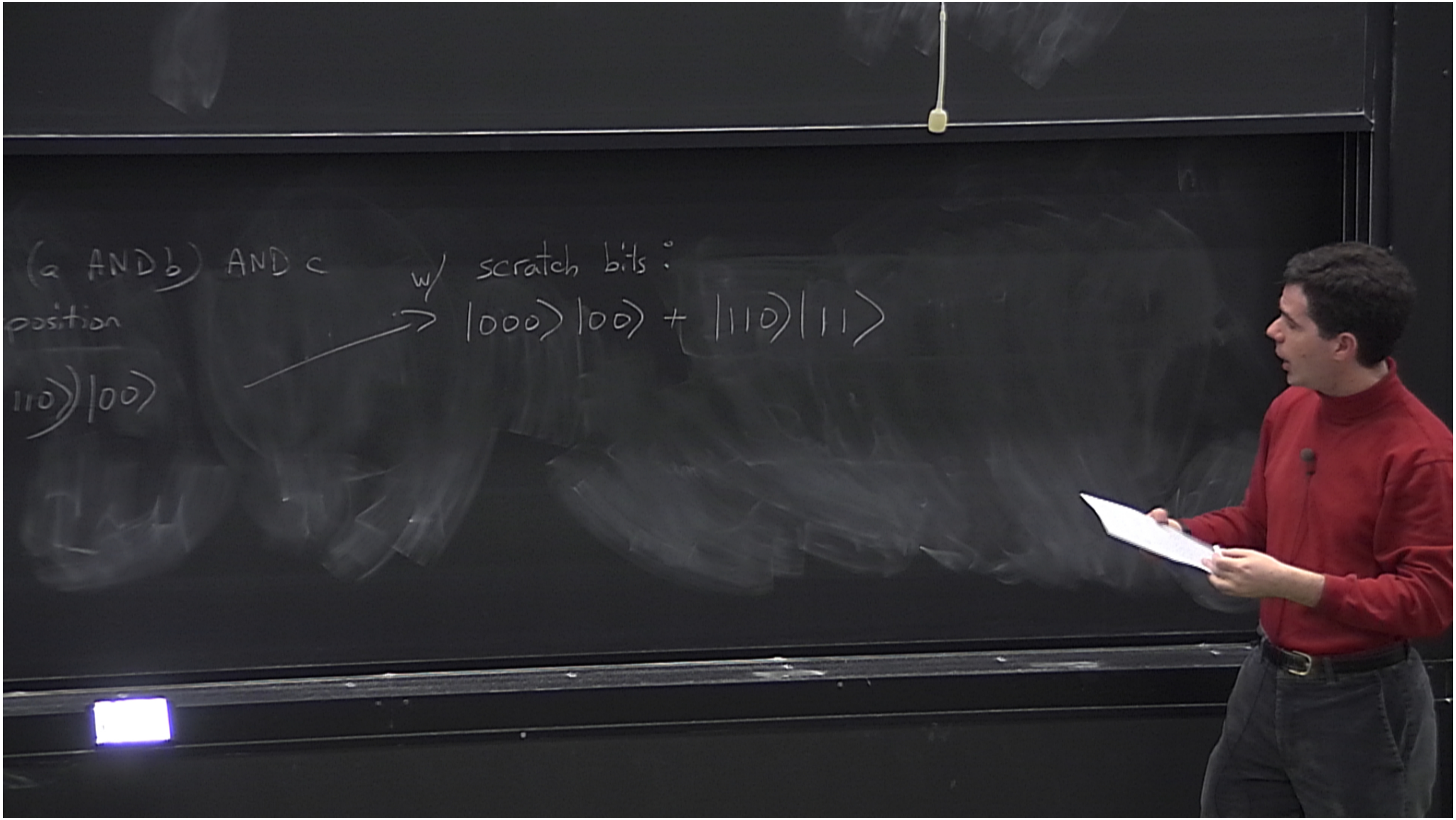
For instance, we can make an EPR pair:

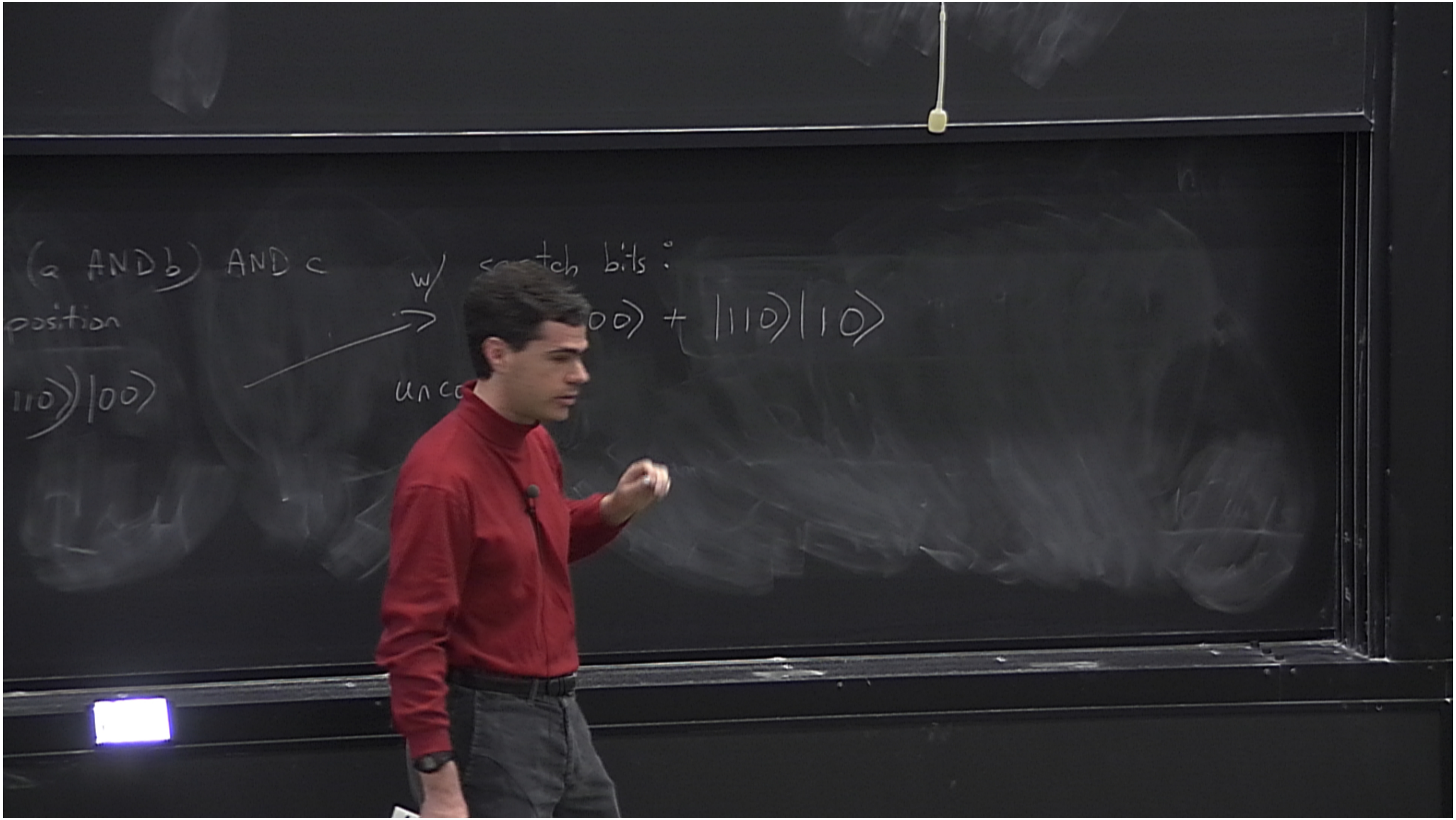


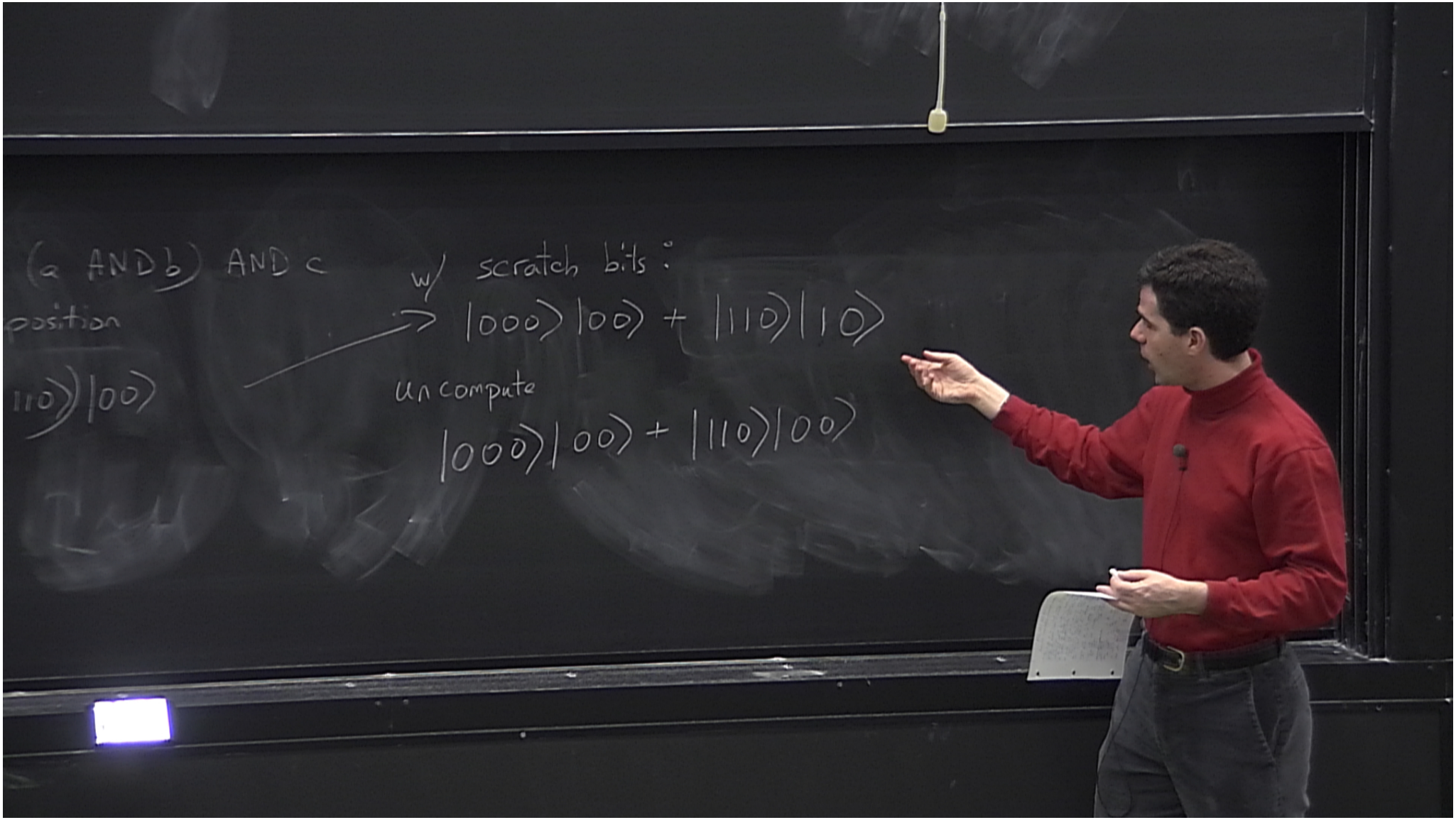
Calculate (a AND b) AND c on superposition

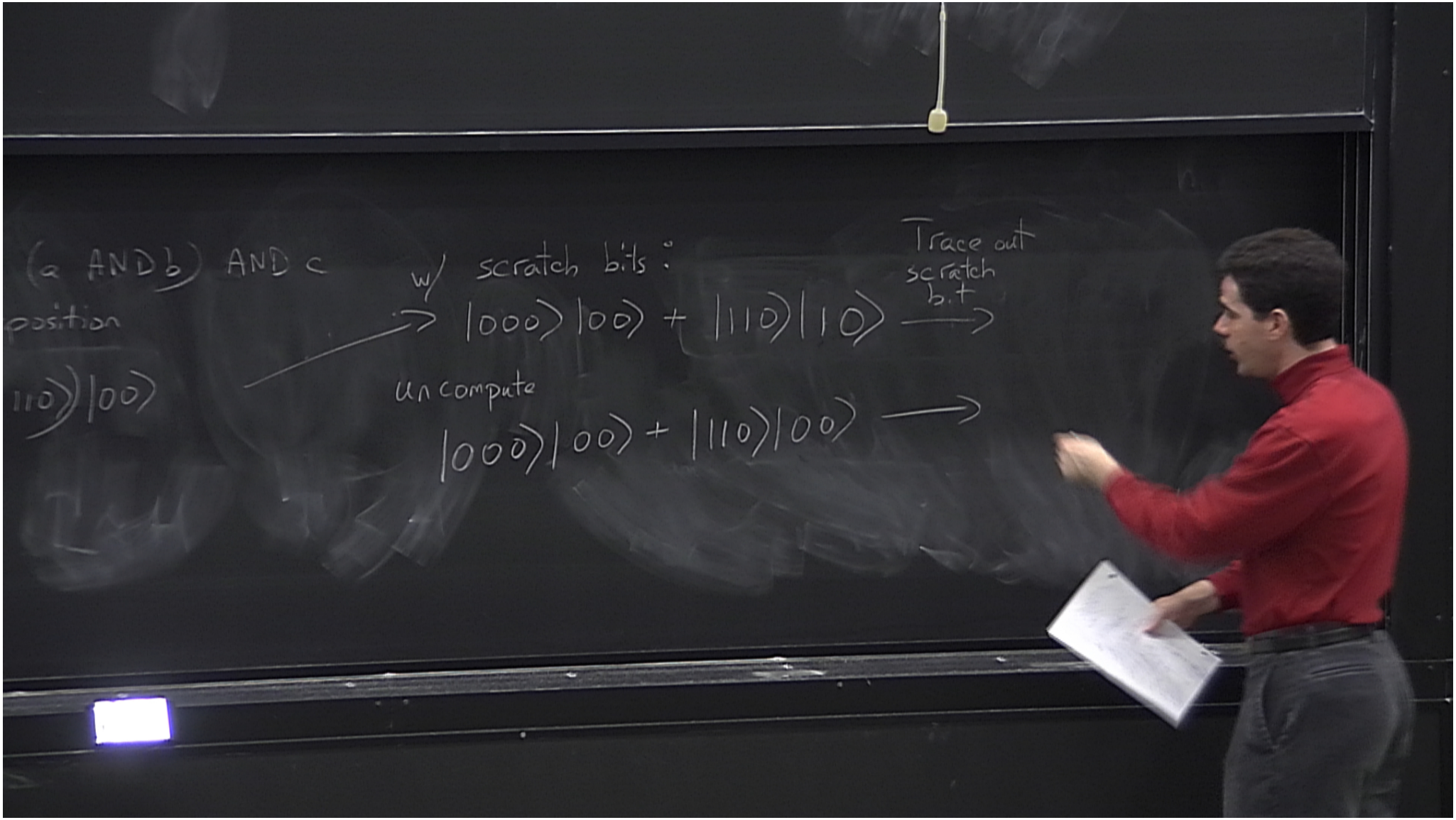
$(|000\rangle + |110\rangle)|00\rangle$

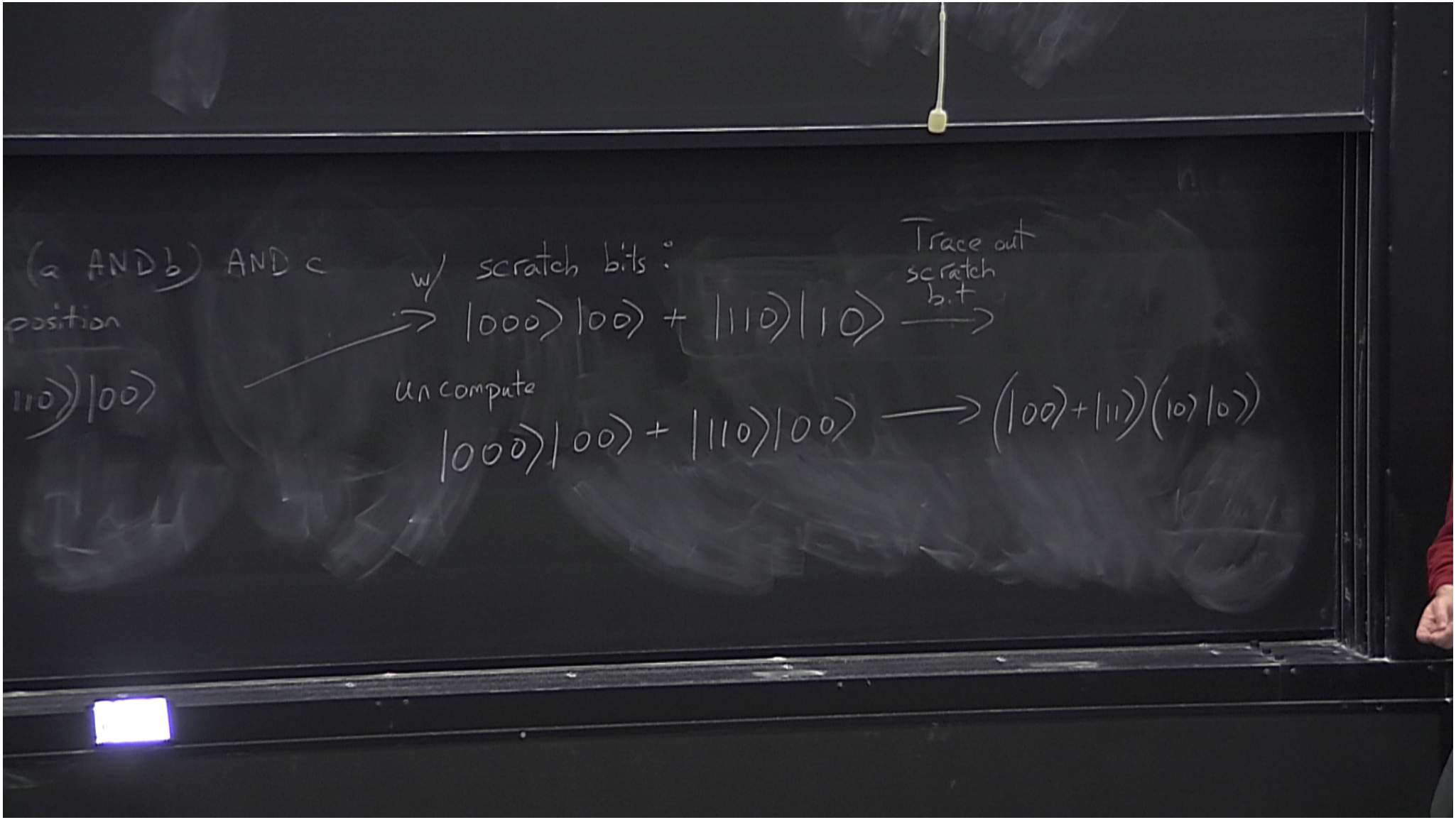

(a AND b) AND c
w/ scratch bits:
position
|000⟩
|10⟩|00⟩

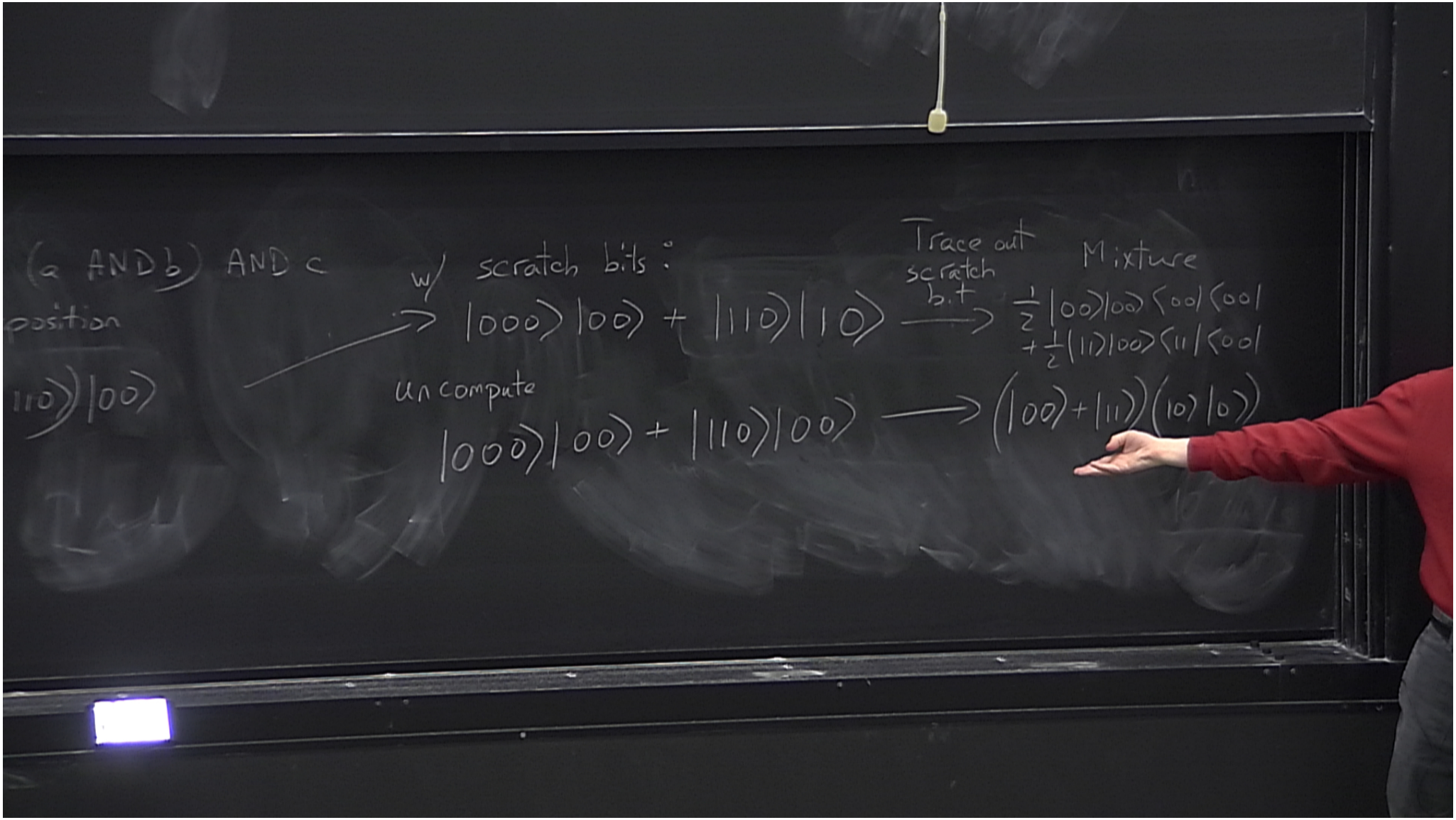

(a AND b) AND c
w/ scratch bits:
|000⟩|00⟩ + |110⟩|11⟩
position
110⟩|00⟩

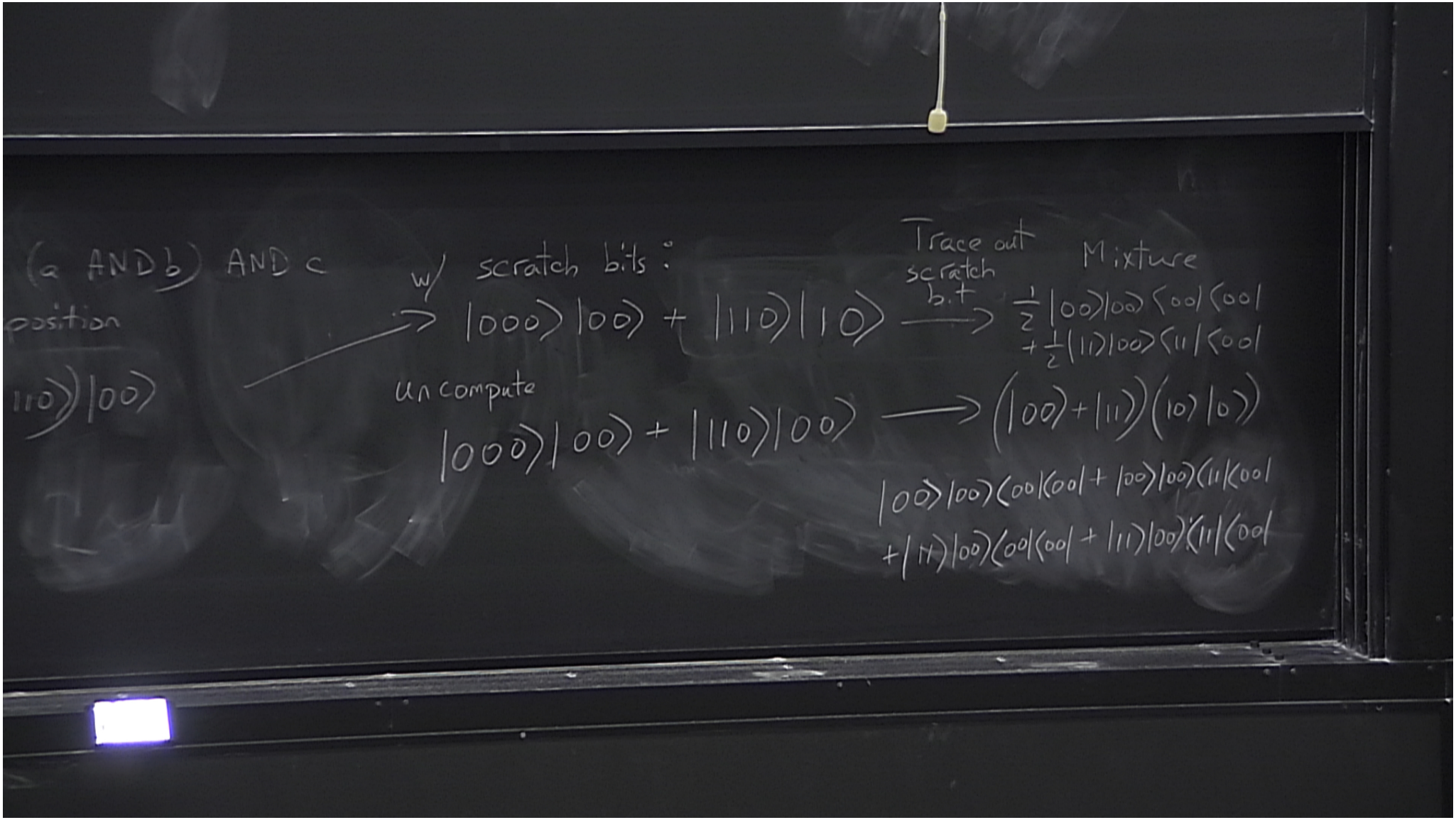

(a AND b) AND c
w/
bits:
|110⟩|00⟩ + |110⟩|10⟩

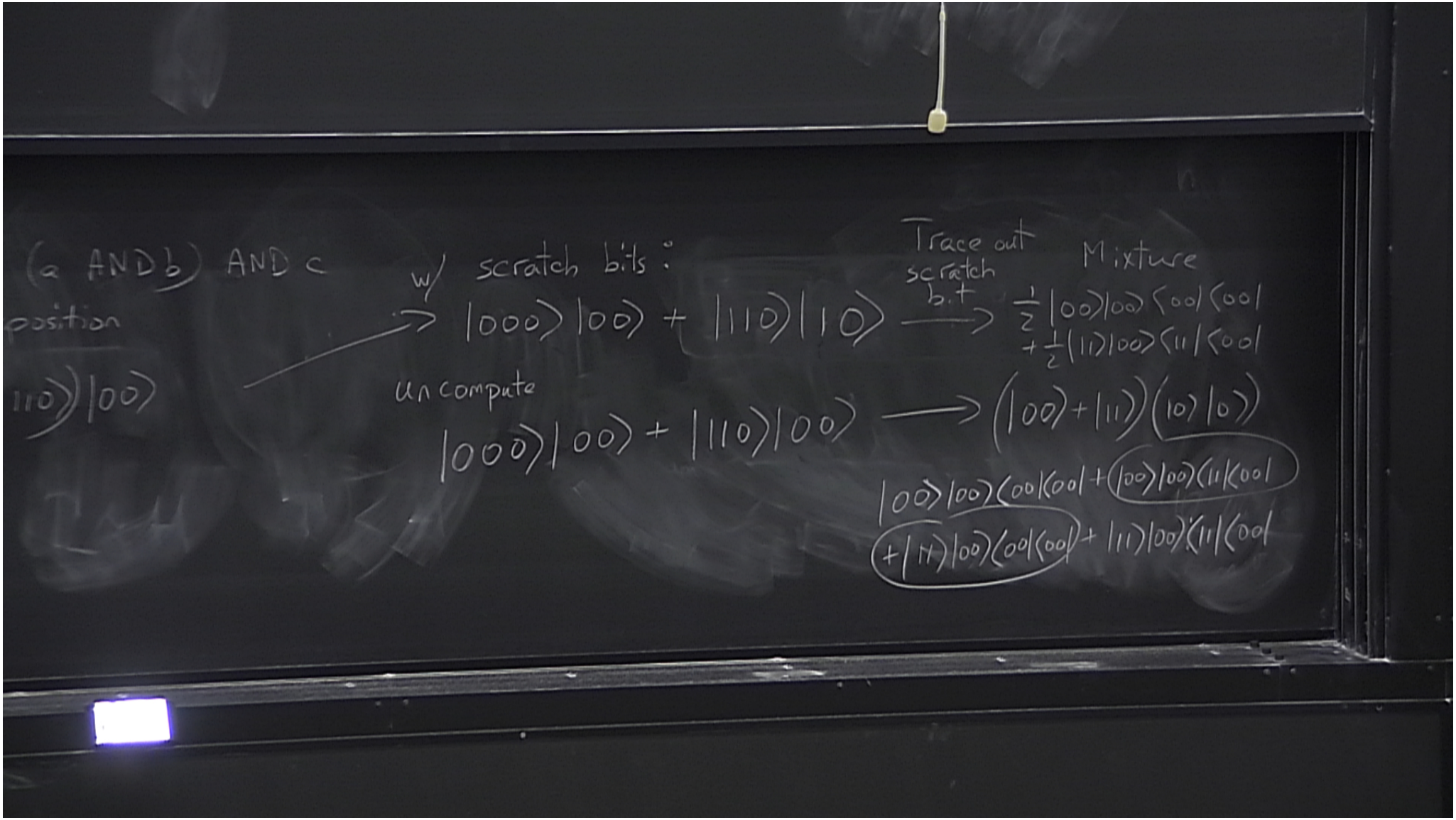

(a AND b) AND c
position
|110⟩|00⟩
w/ scratch bits:
|000⟩|00⟩ + |110⟩|10⟩
uncompute
|000⟩|00⟩ + |110⟩|00⟩


(a AND b) AND c
position
|110⟩|00⟩
w/ scratch bits:
|000⟩|00⟩ + |110⟩|10⟩
Trace out scratch b.t
uncompute
|000⟩|00⟩ + |110⟩|00⟩


(a AND b) AND c
position
|110⟩|00⟩
w/ scratch bits:
|000⟩|00⟩ + |110⟩|10⟩
Trace out scratch b.t
uncompute
|000⟩|00⟩ + |110⟩|00⟩
(|00⟩ + |11⟩)(|0⟩|0⟩)


(a AND b) AND c
position
|110⟩|00⟩
w/ scratch bits:
|000⟩|00⟩ + |110⟩|10⟩
Trace out scratch b.t
Mixture
½|00⟩|00⟩⟨00|⟨00|
+½|11⟩|00⟩⟨11|⟨00|
uncompute
|000⟩|00⟩ + |110⟩|00⟩
(|00⟩+|11⟩)(|0⟩|0⟩)


(a AND b) AND c
position
|110⟩|00⟩
w/ scratch bits:
|000⟩|00⟩ + |110⟩|10⟩
uncompute
|000⟩|00⟩ + |110⟩|00⟩
Trace out scratch b.t
Mixture
½|00⟩|00⟩⟨00|⟨00|
+½|11⟩|00⟩⟨11|⟨00|
(|00⟩ + |11⟩)(|0⟩|0⟩)
|00⟩|00⟩⟨00|⟨00| + |00⟩|00⟩⟨11|⟨00|
+|11⟩|00⟩⟨00|⟨00| + |11⟩|00⟩⟨11|⟨00|


(a AND b) AND c
position
|110⟩|00⟩
w/ scratch bits:
|000⟩|00⟩ + |110⟩|10⟩
uncompute
|000⟩|00⟩ + |110⟩|00⟩
Trace out scratch b.t
Mixture
½|00⟩|00⟩⟨00|⟨00|
+½|11⟩|00⟩⟨11|⟨00|
(|00⟩ + |11⟩)(|10⟩|0⟩)
|00⟩|00⟩⟨00|⟨00| + |00⟩|00⟩⟨11|⟨00|
+|11⟩|00⟩⟨00|⟨00| + |11⟩|00⟩⟨11|⟨00|