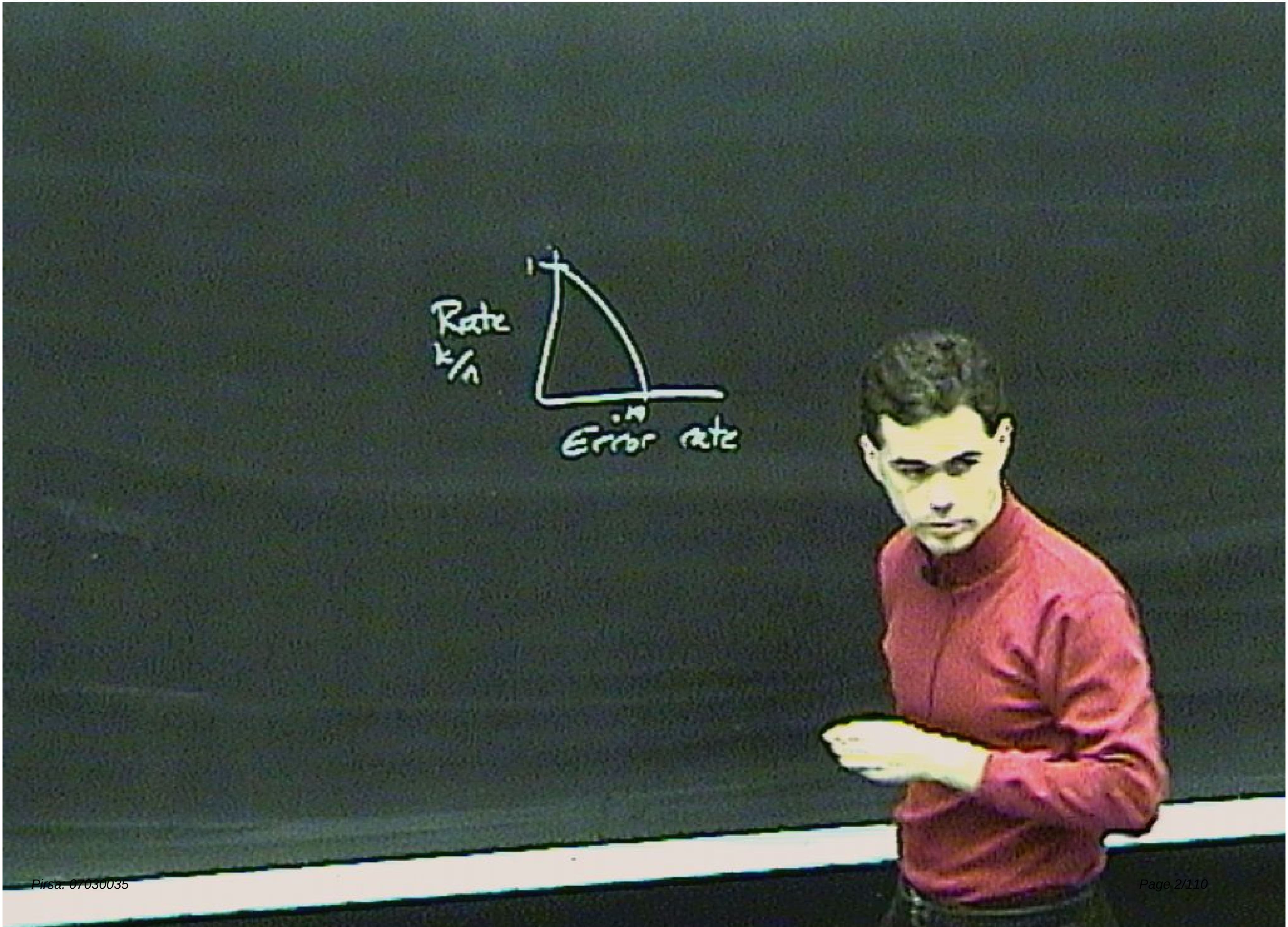
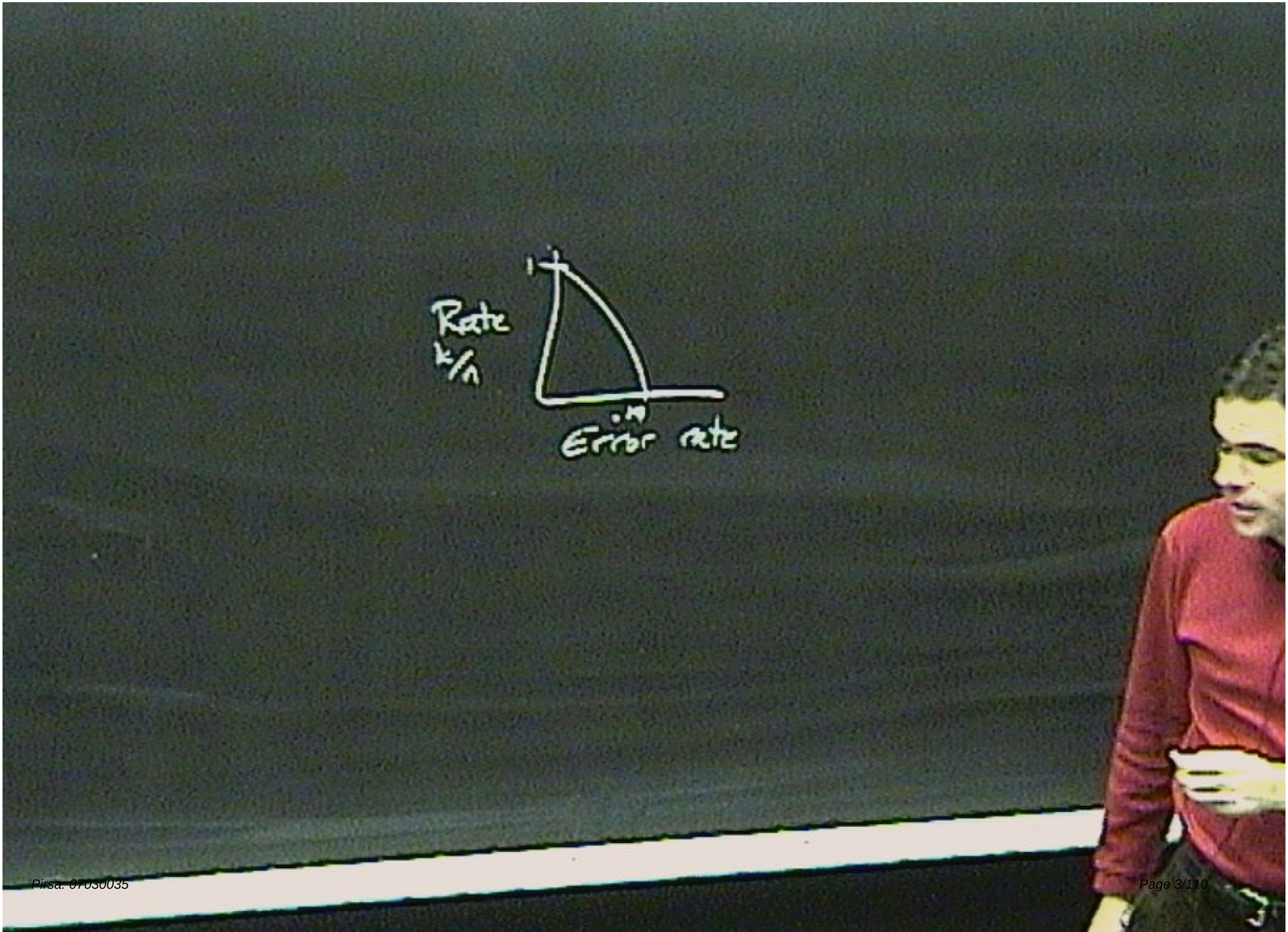
Title: Quantum Error Correction 11A

Date: Mar 20, 2007 03:30 PM

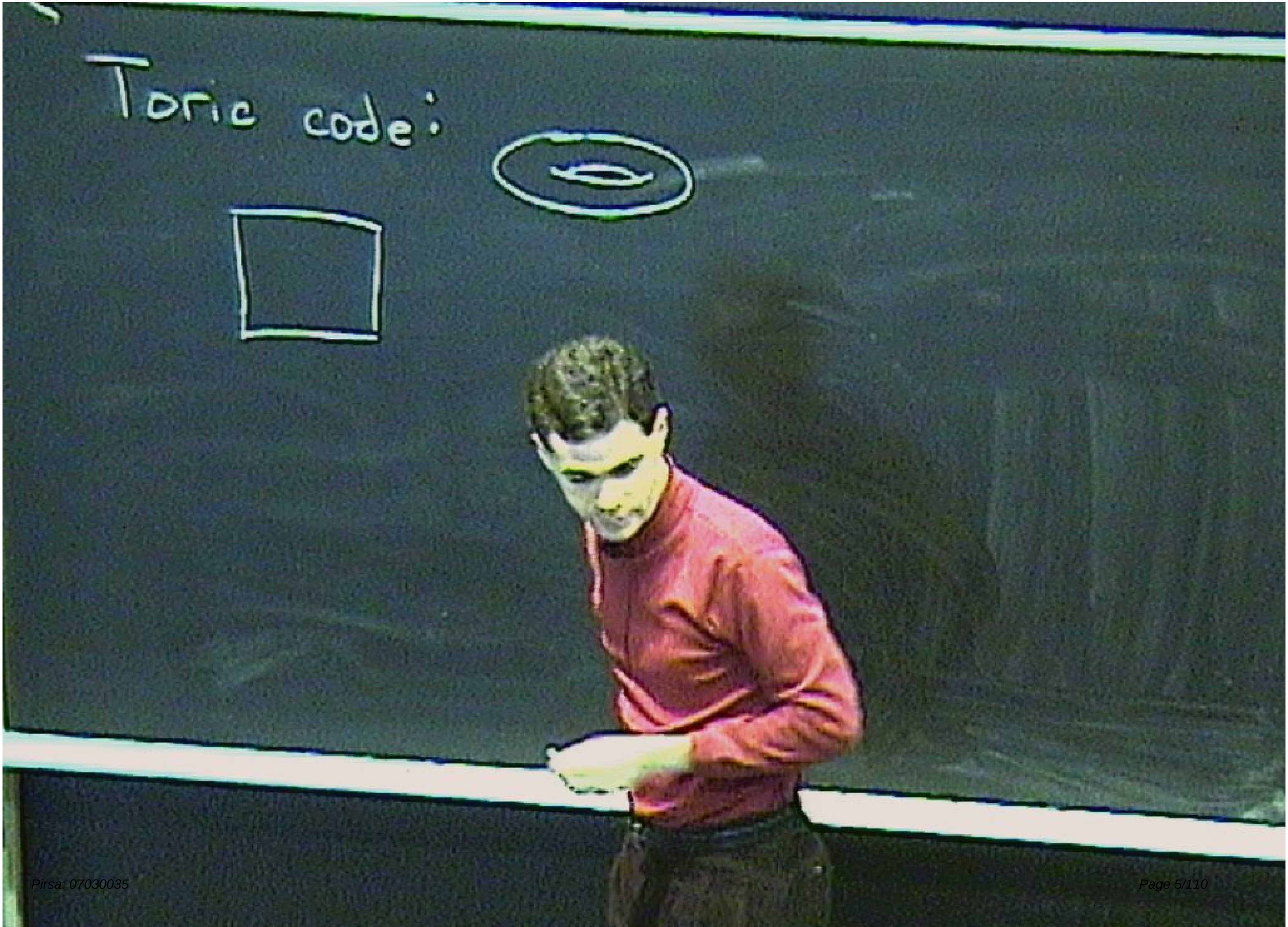
URL: http://pirsa.org/07030035

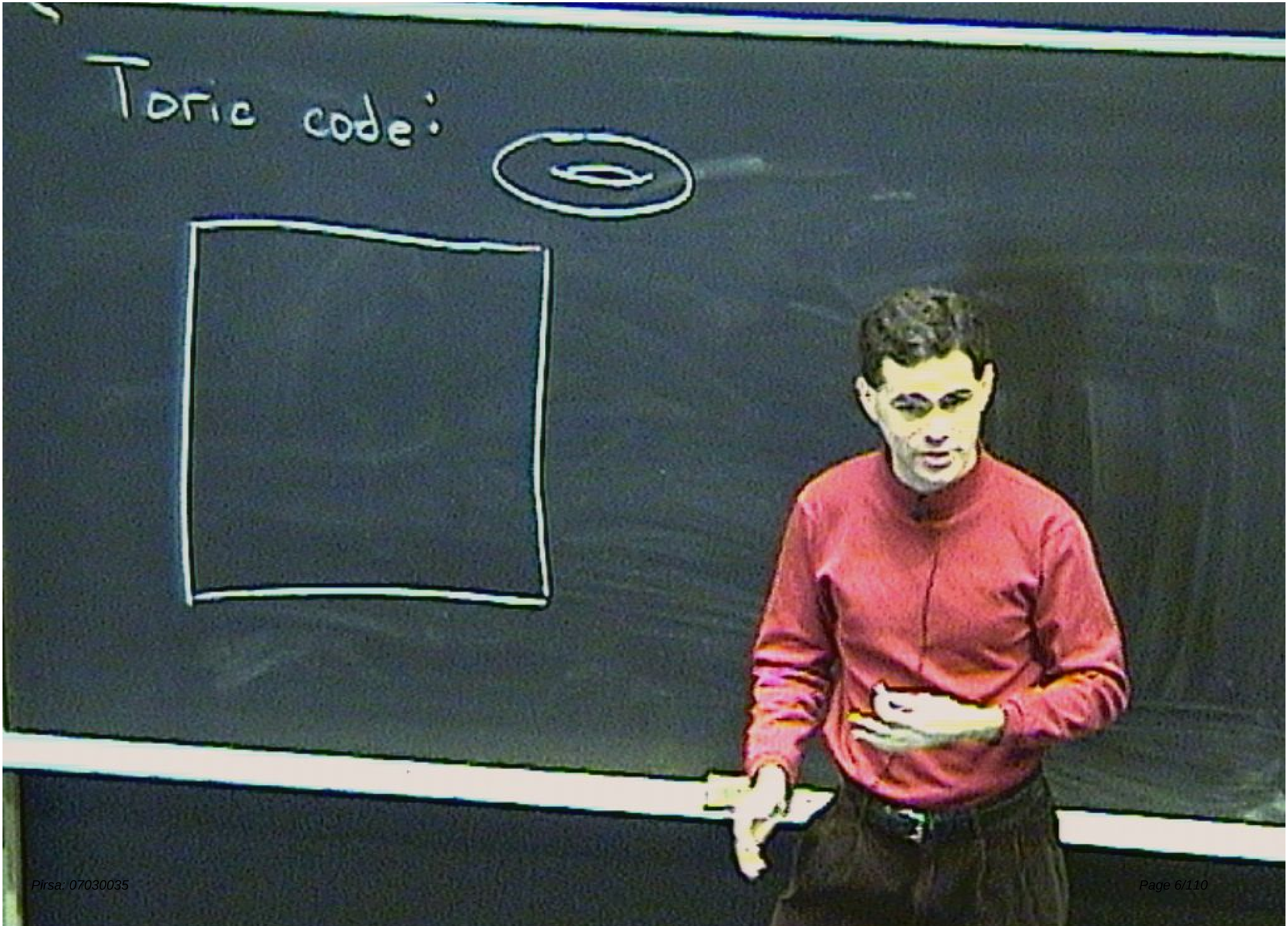
Abstract: Toric code (definition, fault-tolerance, particle model of errors), definition of qudit toric code

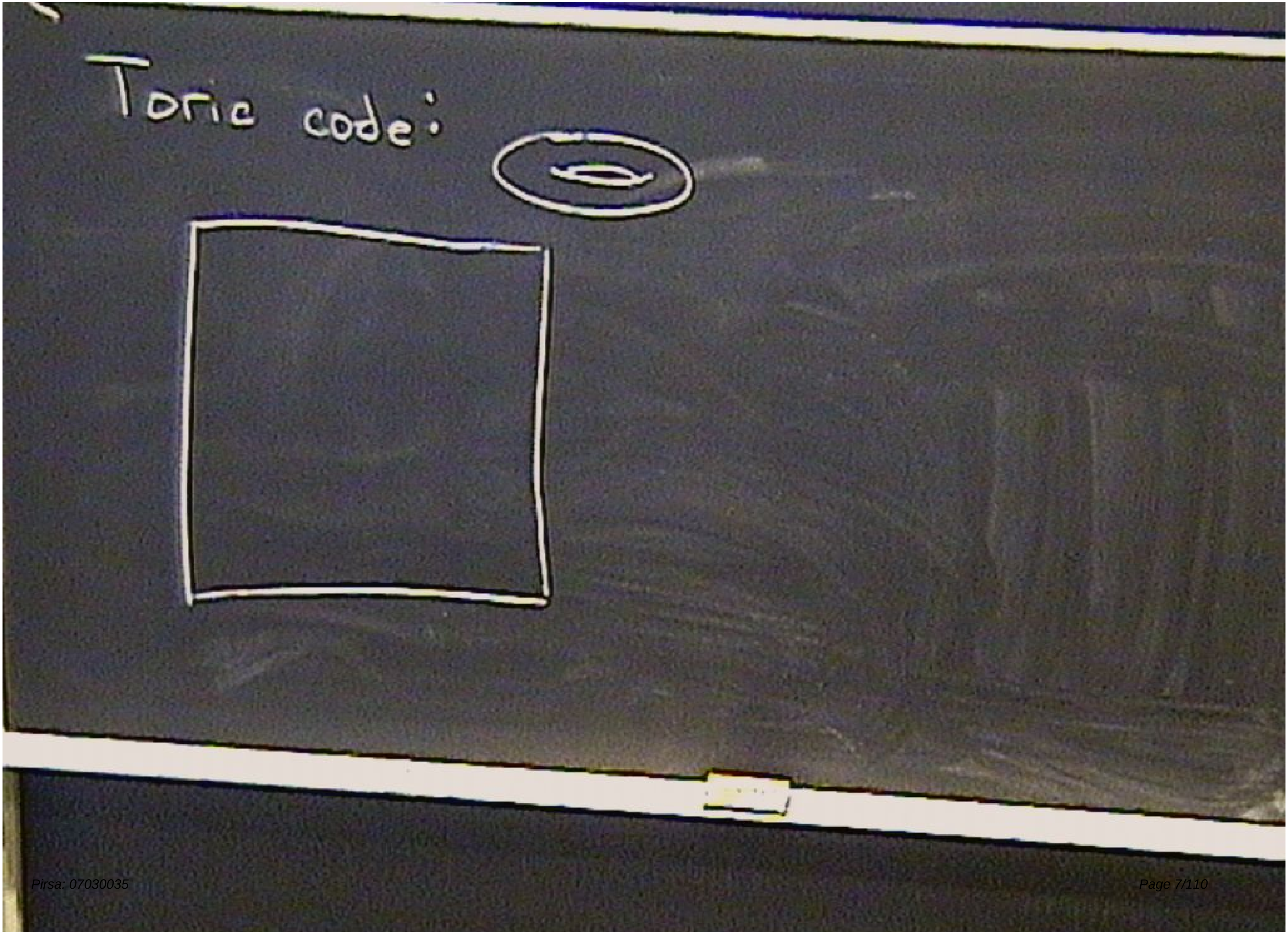

1
Rate
k/n
.11
Error rate

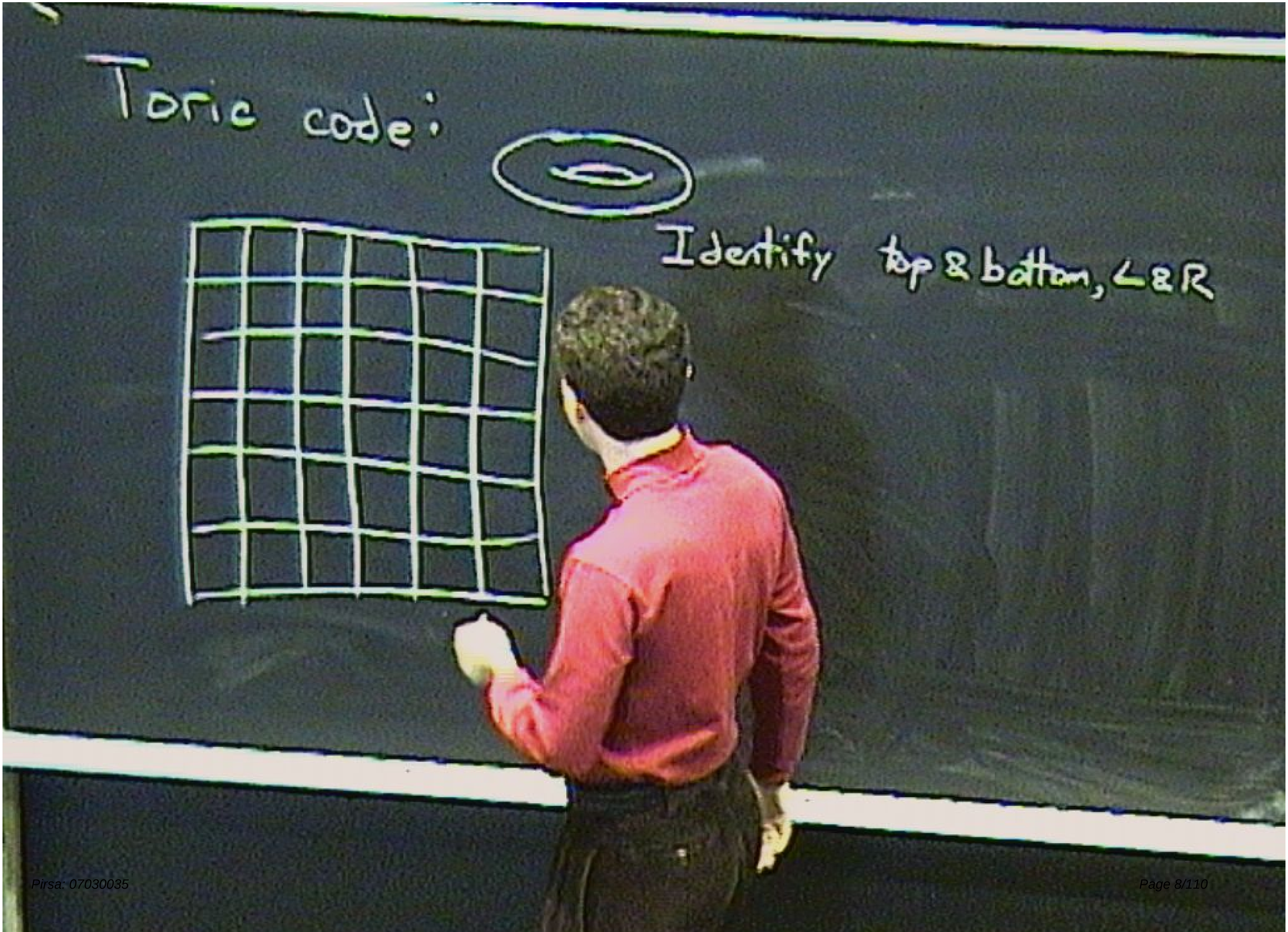

Rate
k/n
1
Error rate

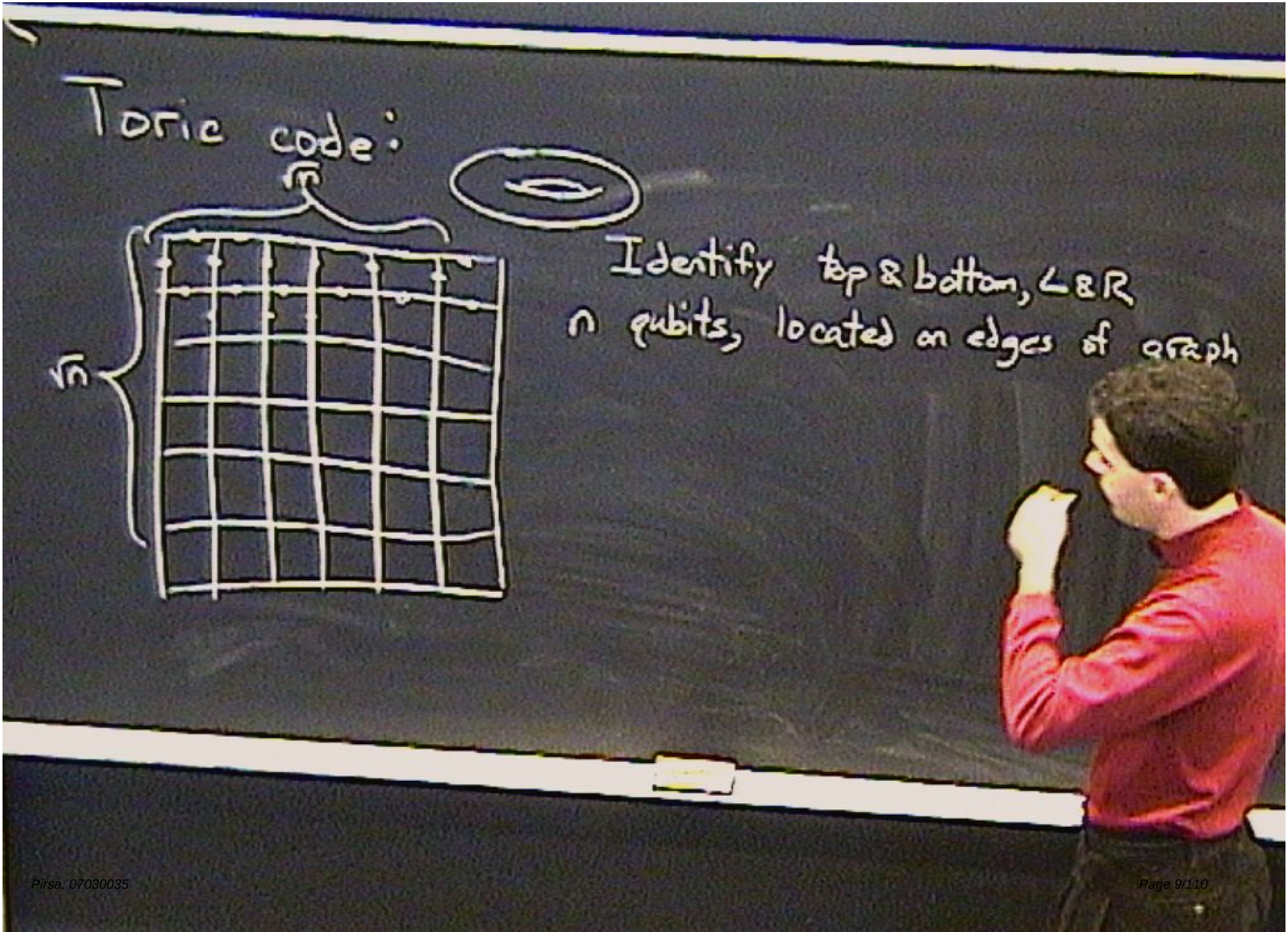
Toric code:


Toric code:

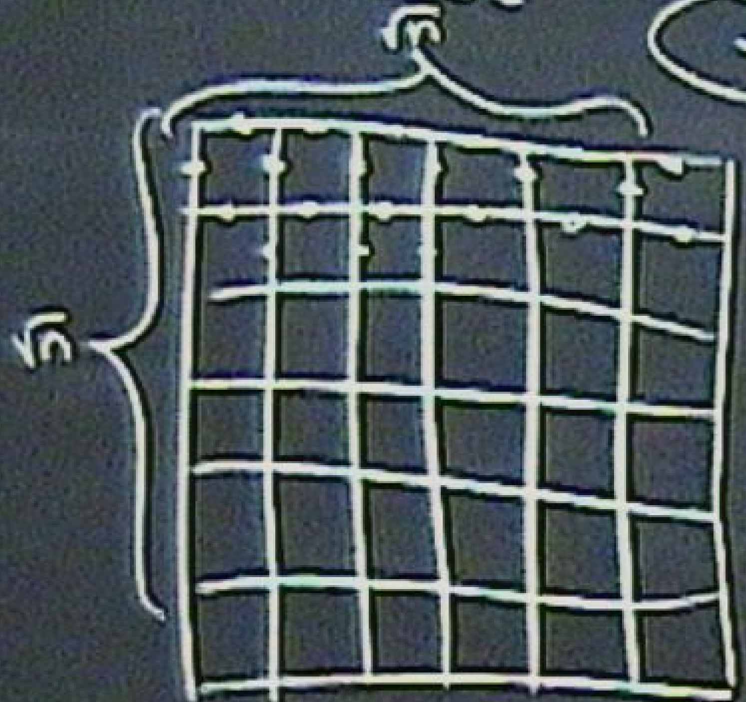

Toric code:


Toric code:


Toric code:
Identify top & bottom, L & R


Toric code:
√n
√n
Identify top & bottom, L & R
n qubits, located on edges of graph

Toric code:

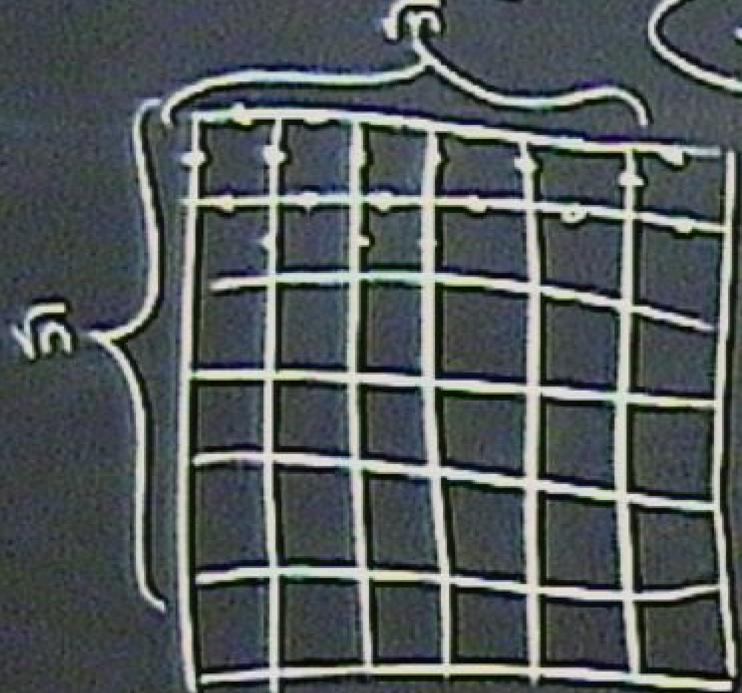


Identify top & bottom, L & R

2n qubits, located on edges of graph

Define stabilizer code:

Toric code:



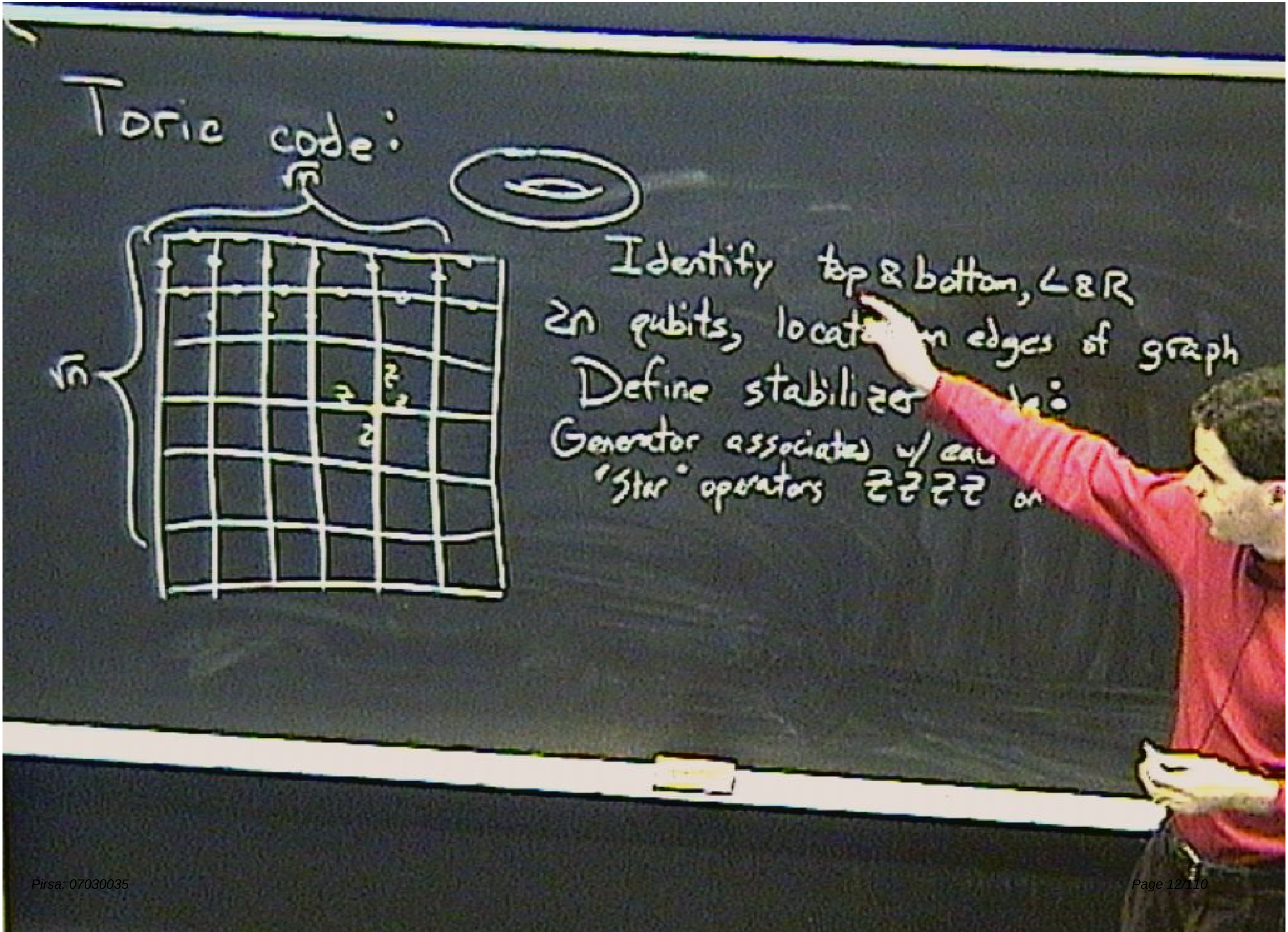
Identify top & bottom, L & R

$2n$ qubits, located on edges of graph

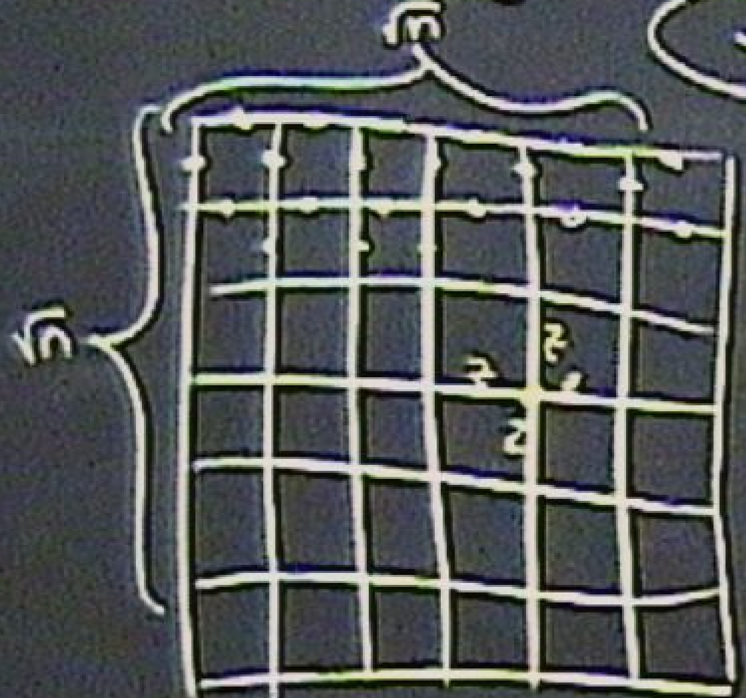
Define stabilizer code:

Generator associated w/ each vertex

"Star" operators


Toric code:
√n
√n
Identify top & bottom, L & R
2n qubits, locat n edges of graph
Define stabilize :
Generator associated w/ eac
"Star" operators ZZZZ on
Z
Z
Z
Z

Toric code:



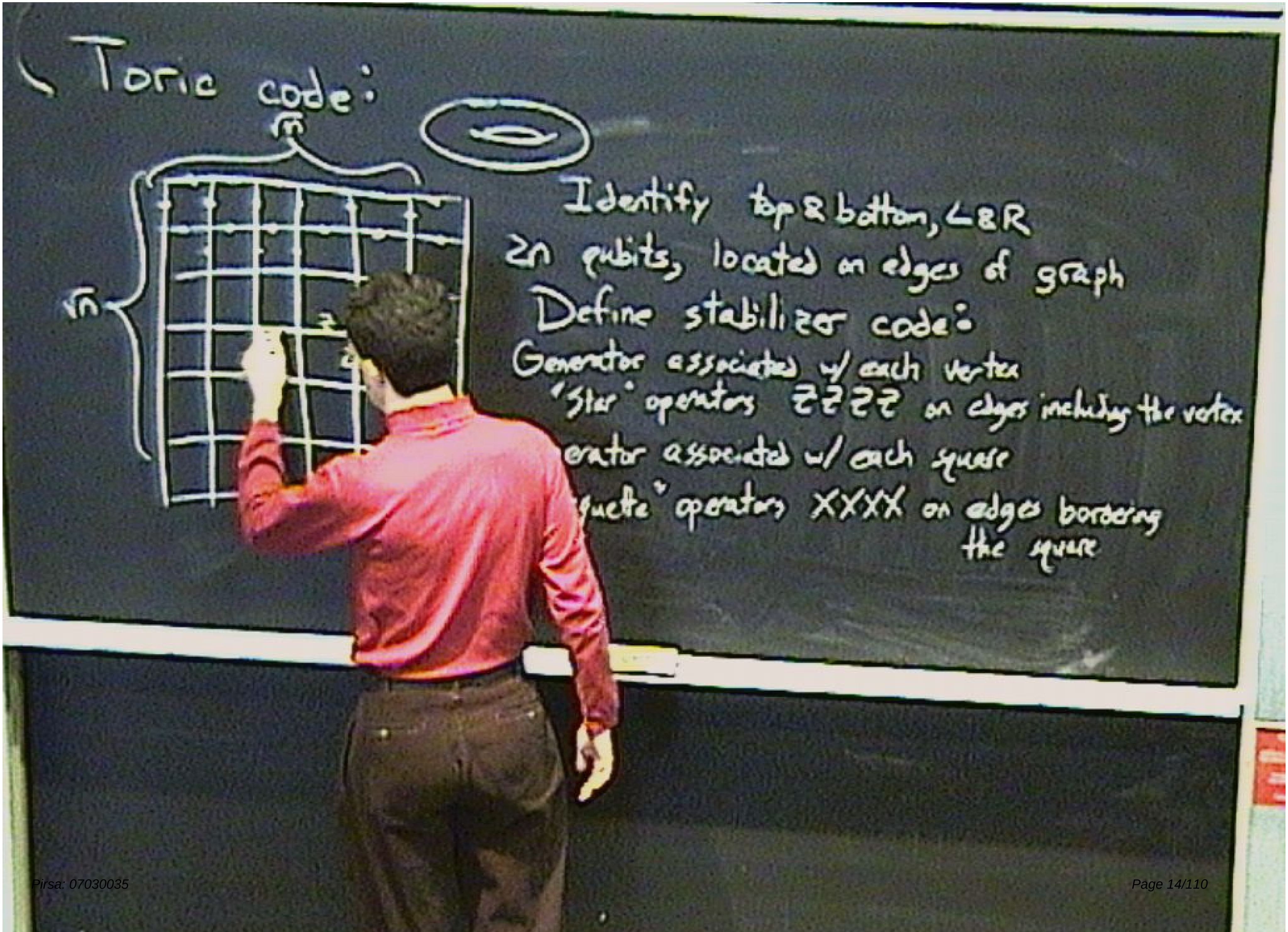
Identify top & bottom, L & R

$2n$ qubits, located on edges of graph

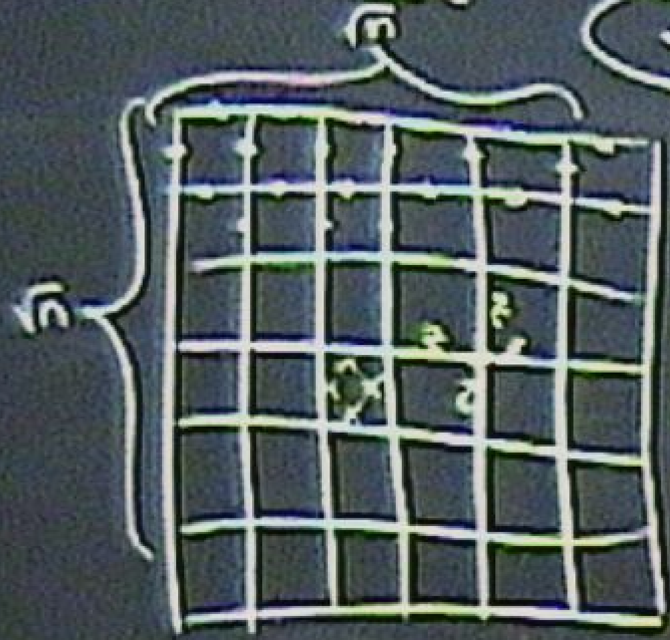
Define stabilizer code:

Generator associated w/ each vertex

"Star" operators $ZZZZ$ on edges including th


Toric code:
√n
√n
Identify top & bottom, L & R
2n qubits, located on edges of graph
Define stabilizer code:
Generator associated w/ each vertex
"Star" operators ZZZZ on edges including the vertex
erator associated w/ each square
quette" operators XXXX on edges bordering
the square

Toric code:



Identify top & bottom, L & R

$2n$ qubits, located on edges of graph

Define stabilizer code:

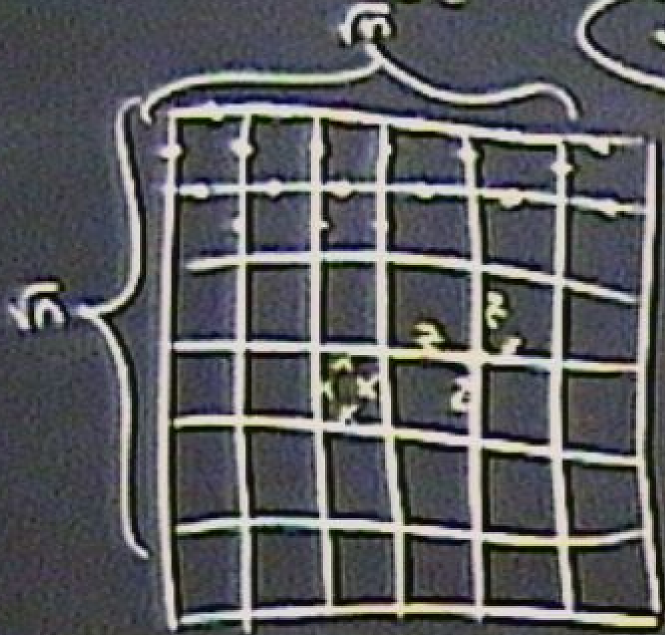
Generator associated w/ each vertex

"Star" operators $ZZZZ$ on edges including the vertex

Generator associated w/ each square

"Plaquette" operators $XXXX$ on edges bordering the square

Toric code:



Identify top & bottom, L & R

$2n$ qubits, located on edges of graph

Define stabilizer code:

Generator associated w/ each vertex

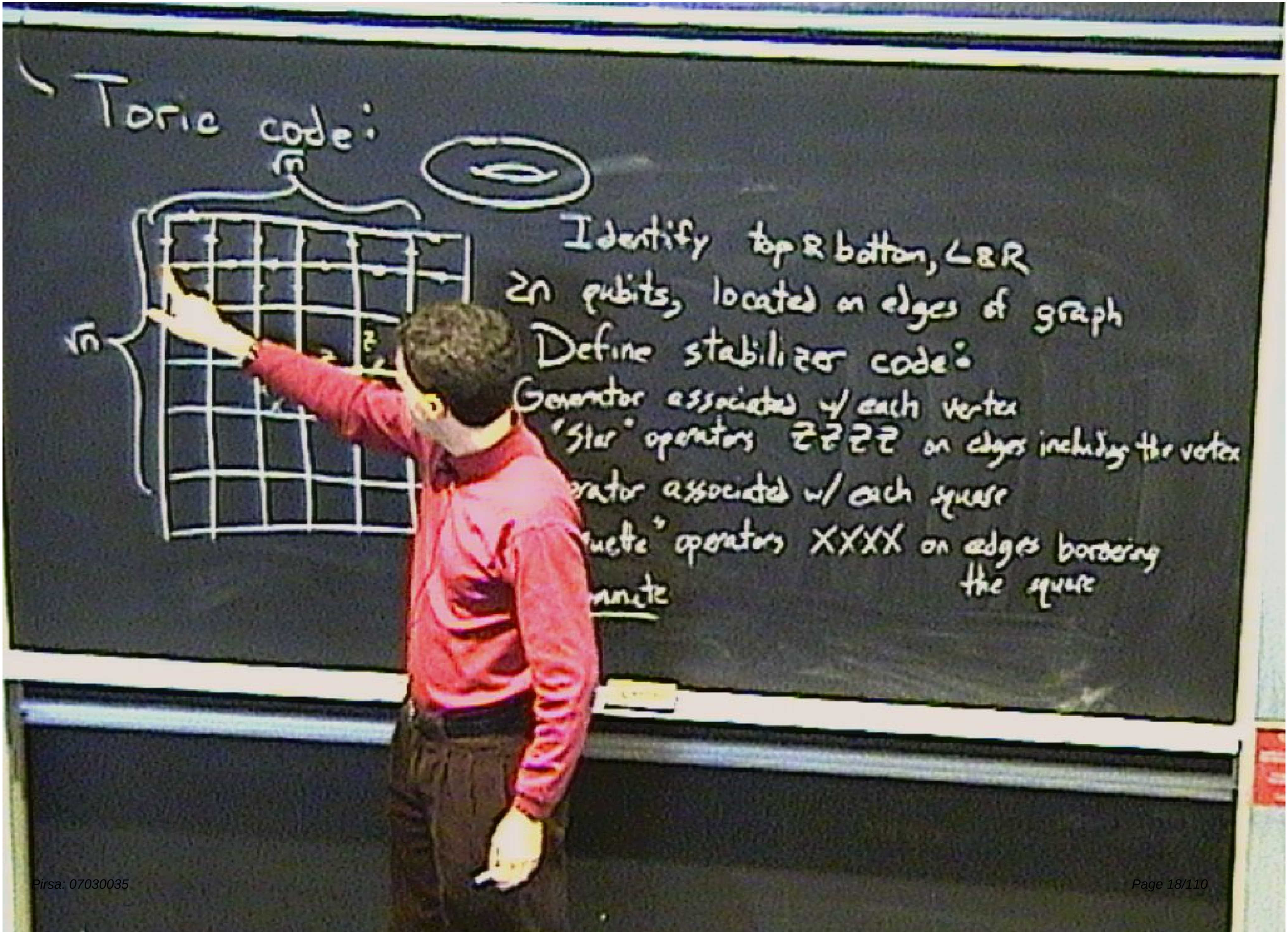
"Star" operators ZZZZ on edges including the vertex

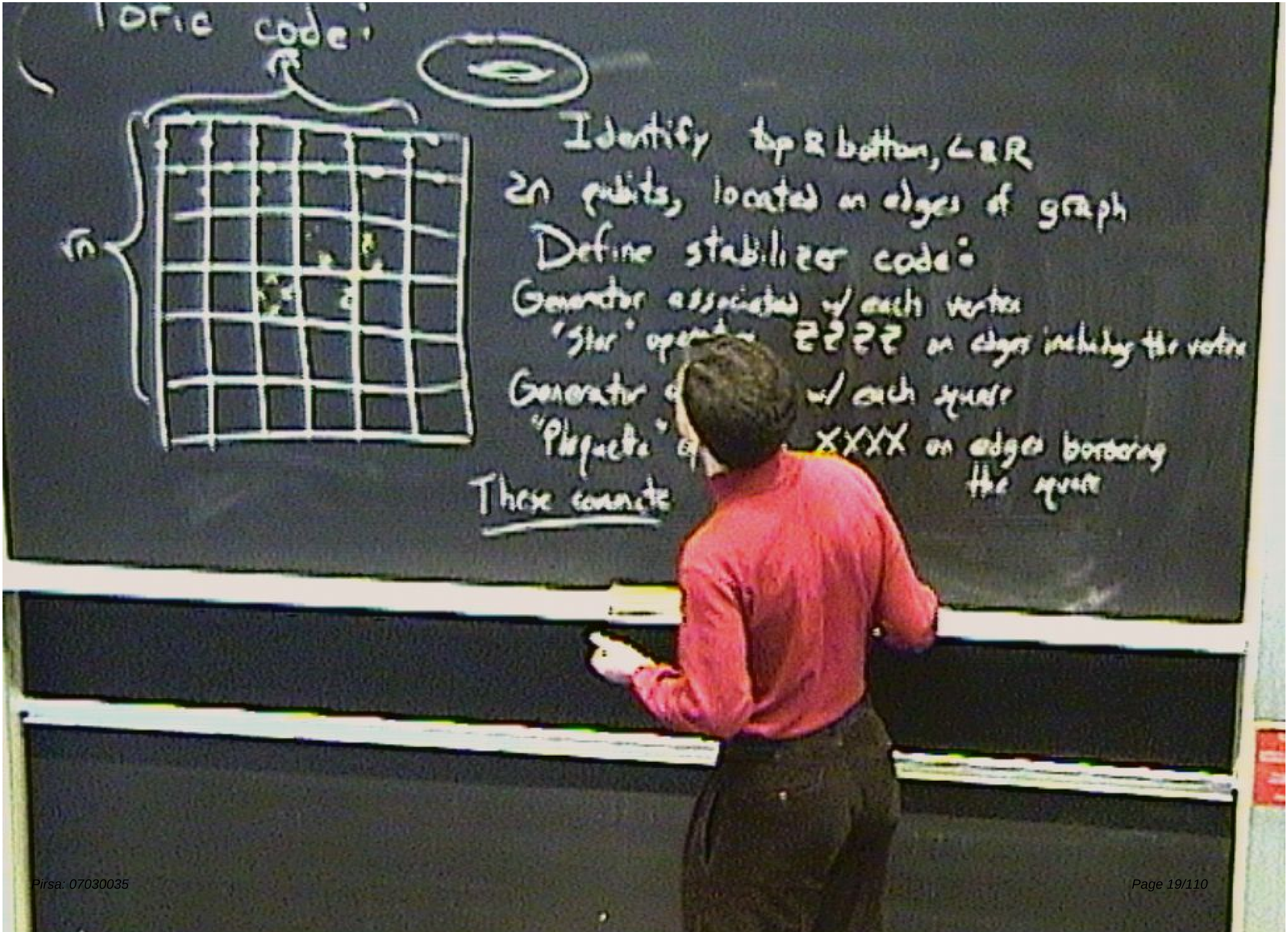
Generator associated w/ each square

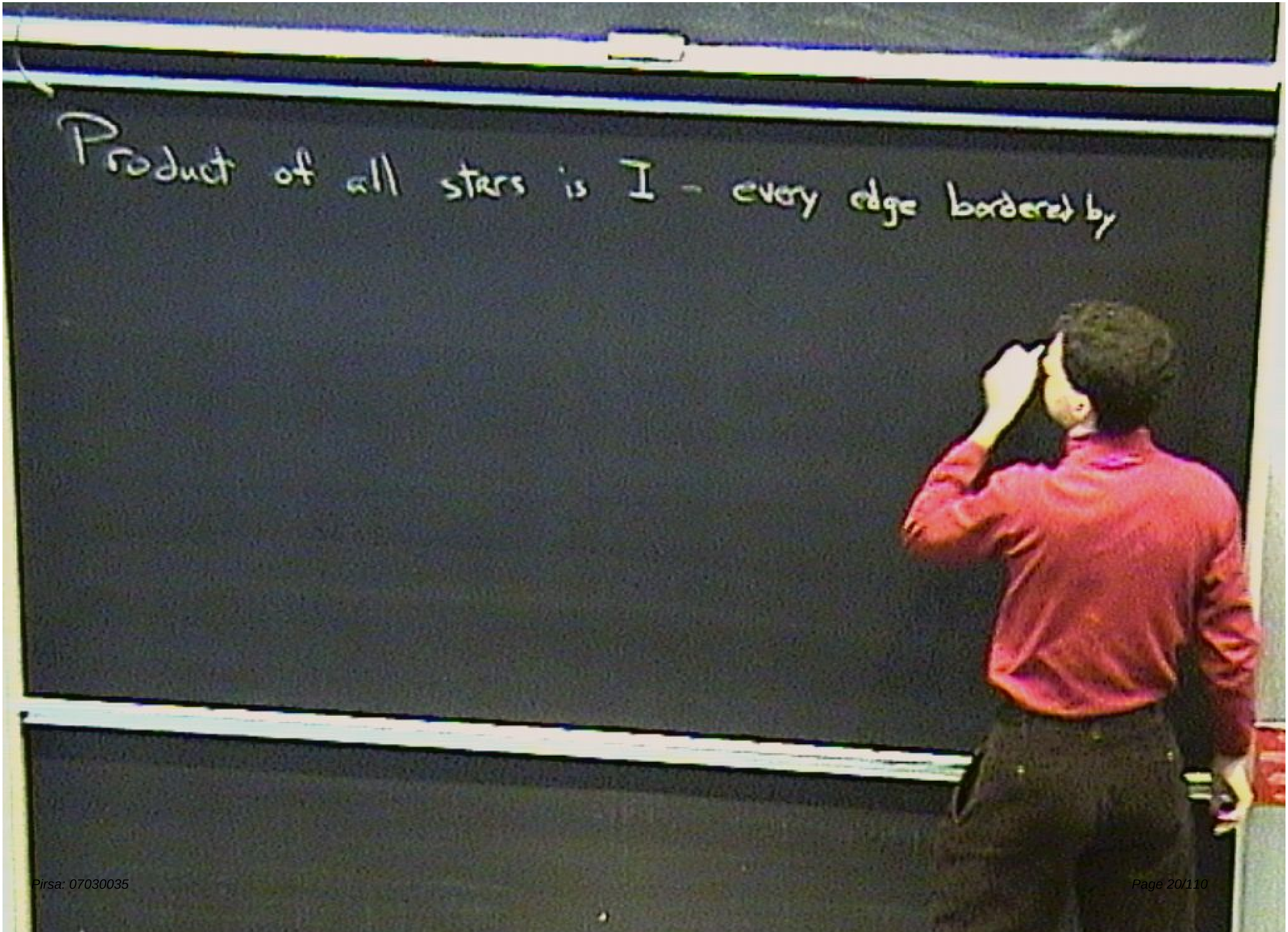
"Plaquette" operators XXXX on edges bordering the square

<u>These commute</u>

Product of all stars is I


Toric code:
√n
√n
Identify top & bottom, L&R
2n qubits, located on edges of graph
Define stabilizer code:
Generator associated w/ each vertex
"Star" operators ZZZZ on edges including the vertex
associated w/ each square
operators XXXX on edges bordering the square


Toric code:
Identify top & bottom, L & R
2n qubits, located on edges of graph
Define stabilizer code:
Generator associated w/ each vertex
"Star" operator ZZZZ on edges including the vertex
Generator w/ each square
"Plaquette" XXXX on edges bordering the square
These commute


Product of all stars is I - every edge bordered by

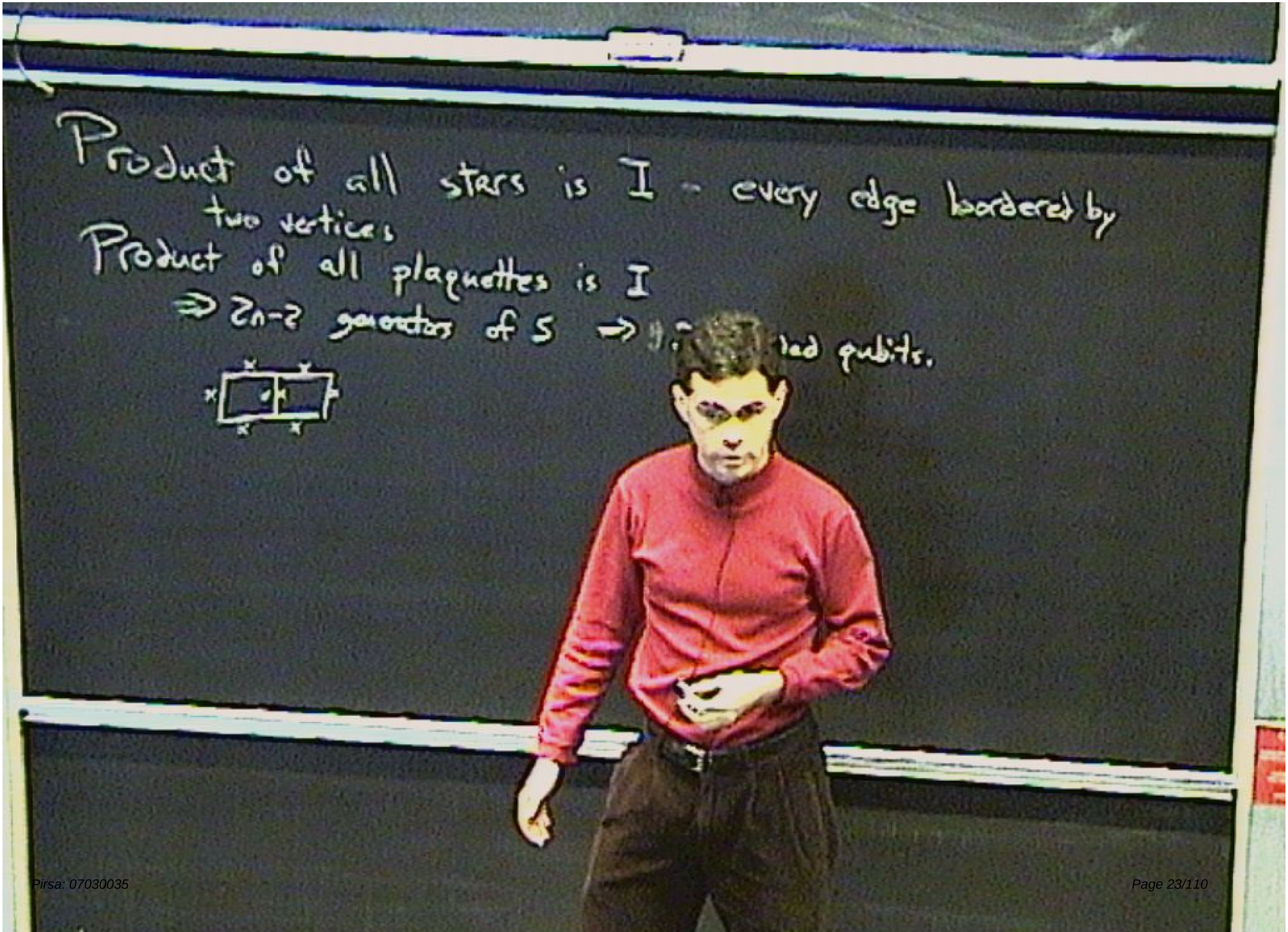
Product of all stars is I - every edge bordered by two vertices

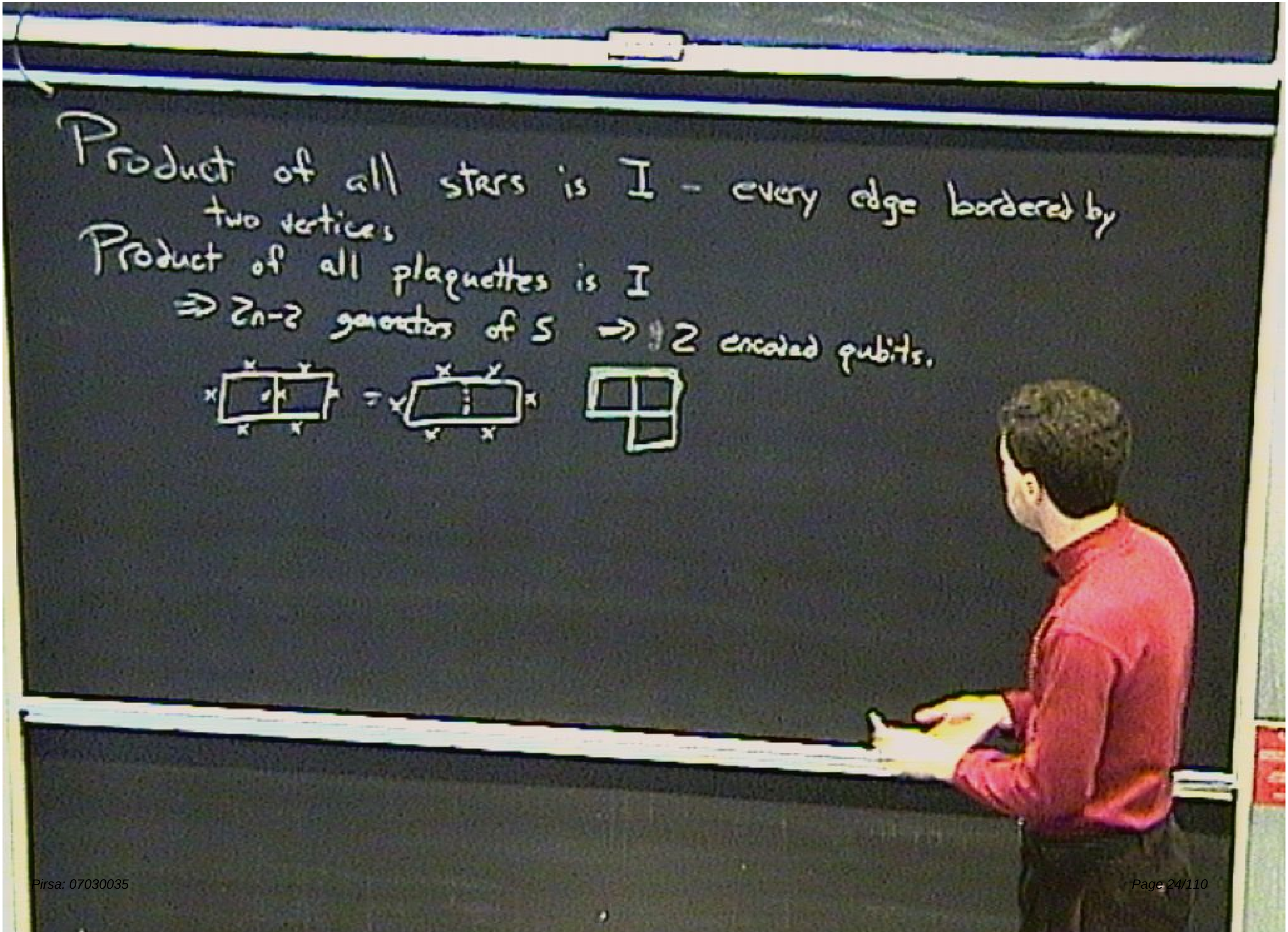
Product of all plaquettes is I

Product of all stars is I - every edge bordered by two vertices

Product of all plaquettes is I

$\Rightarrow 2n-2$ generators of S $\Rightarrow$ 2 encoded qubits.

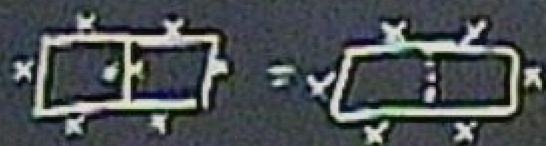

Product of all stars is I – every edge bordered by two vertices
Product of all plaquettes is I
⇒ 2n-2 generators of S ⇒
ded qubits.

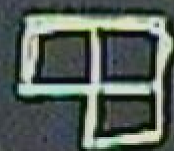

Product of all stars is I – every edge bordered by two vertices
Product of all plaquettes is I
⇒ 2n-2 generators of S ⇒ 2 encoded qubits.

Product of all stars is I — every edge bordered by two vertices

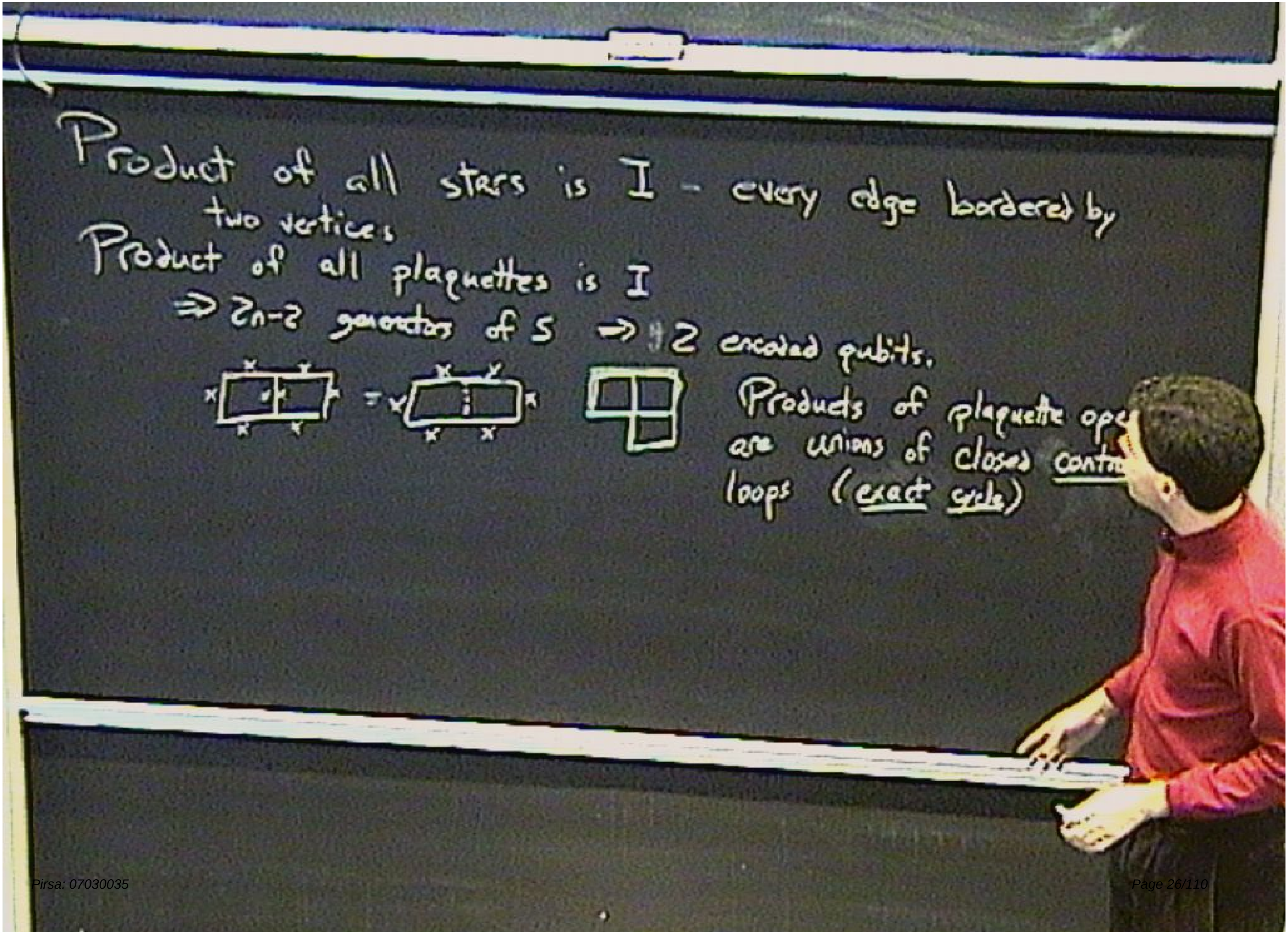
Product of all plaquettes is I

$\Rightarrow 2n-2$ generators of S $\Rightarrow$ 2 encoded qubits.





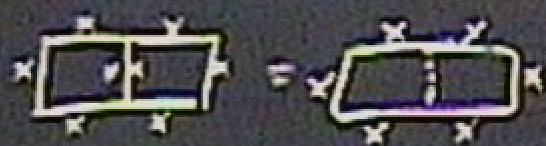
Products of plaquette operators are unions of closed loops

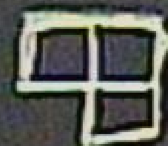

Product of all stars is I - every edge bordered by two vertices
Product of all plaquettes is I
⇒ 2n-2 generators of S ⇒ 2 encoded qubits.
Products of plaquette op
are unions of closed cont
loops (exact cycle)

Product of all stars is I – every edge bordered by two vertices.

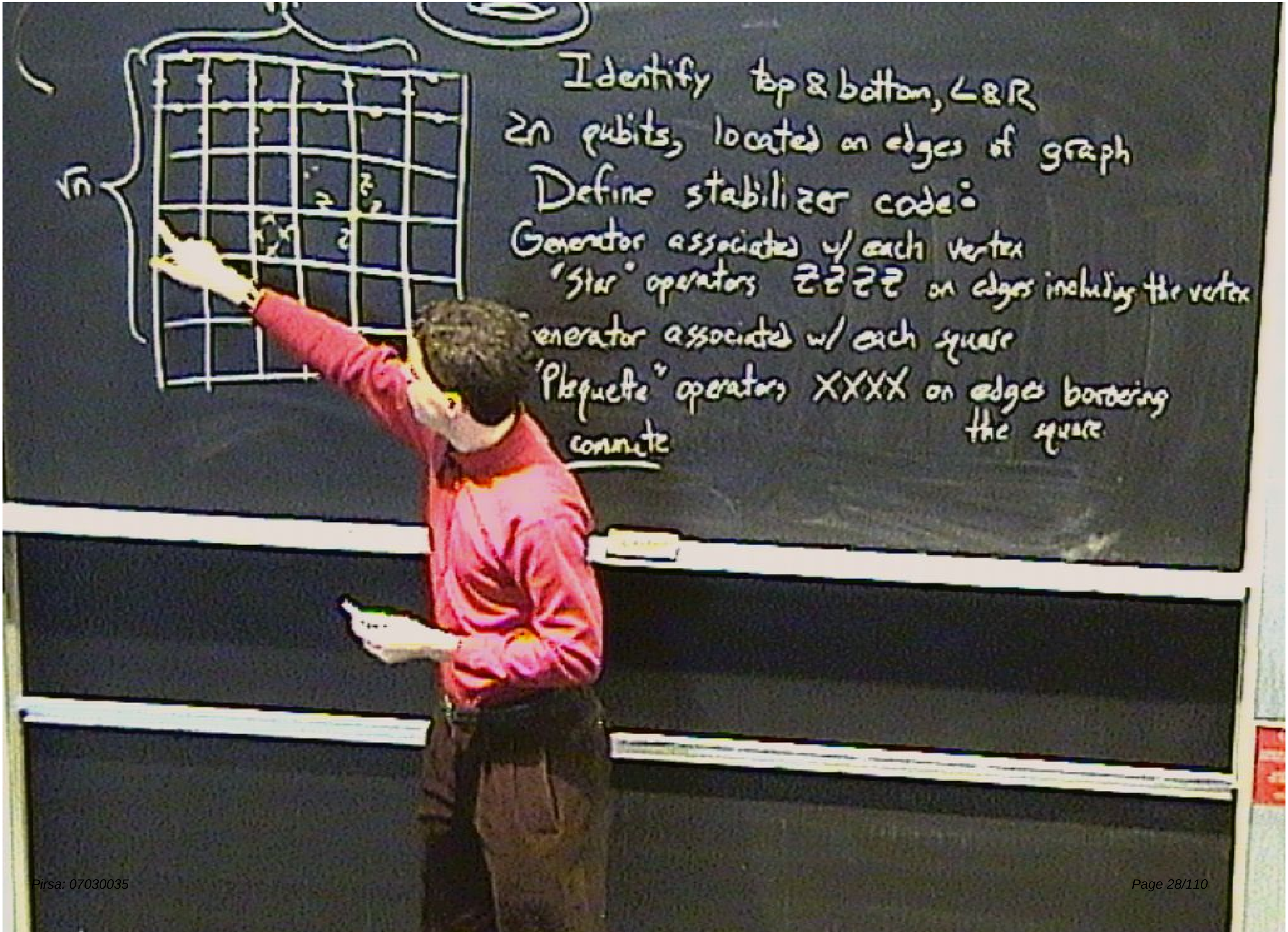
Product of all plaquettes is I

$\Rightarrow 2n-2$ generators of S $\rightarrow$ 2 encoded qubits.





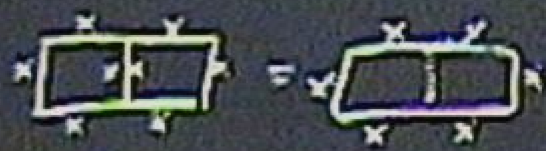
Products of plaquette operators are unions of closed contractible loops (exact cycle)

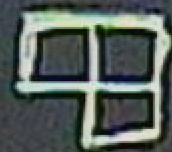

√n
Identify top & bottom, L & R
2n qubits, located on edges of graph
Define stabilizer code:
Generator associated w/ each vertex
"Star" operators ZZZZ on edges including the vertex
Generator associated w/ each square
"Plaquette" operators XXXX on edges bordering the square
commute

Product of all stars is I – every edge bordered by two vertices

Product of all plaquettes is I

$\Rightarrow$ $2n-2$ generators of S $\Rightarrow$ 2 encoded qubits.





Products of plaquette operators are unions of closed <u>contractible</u> loops (<u>exact</u> <u>cycle</u>)

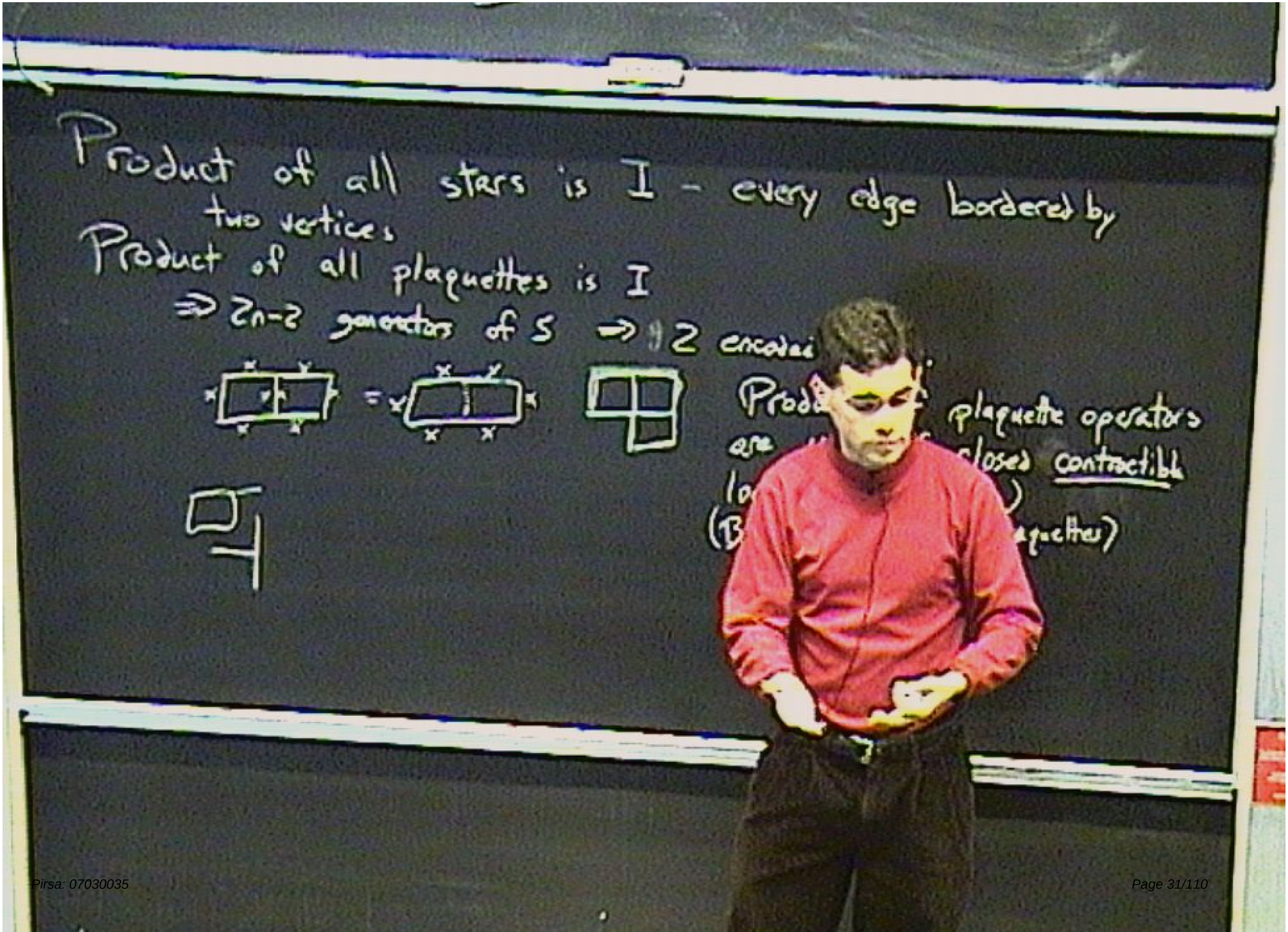
Product of all stars is I – every edge bordered by two vertices

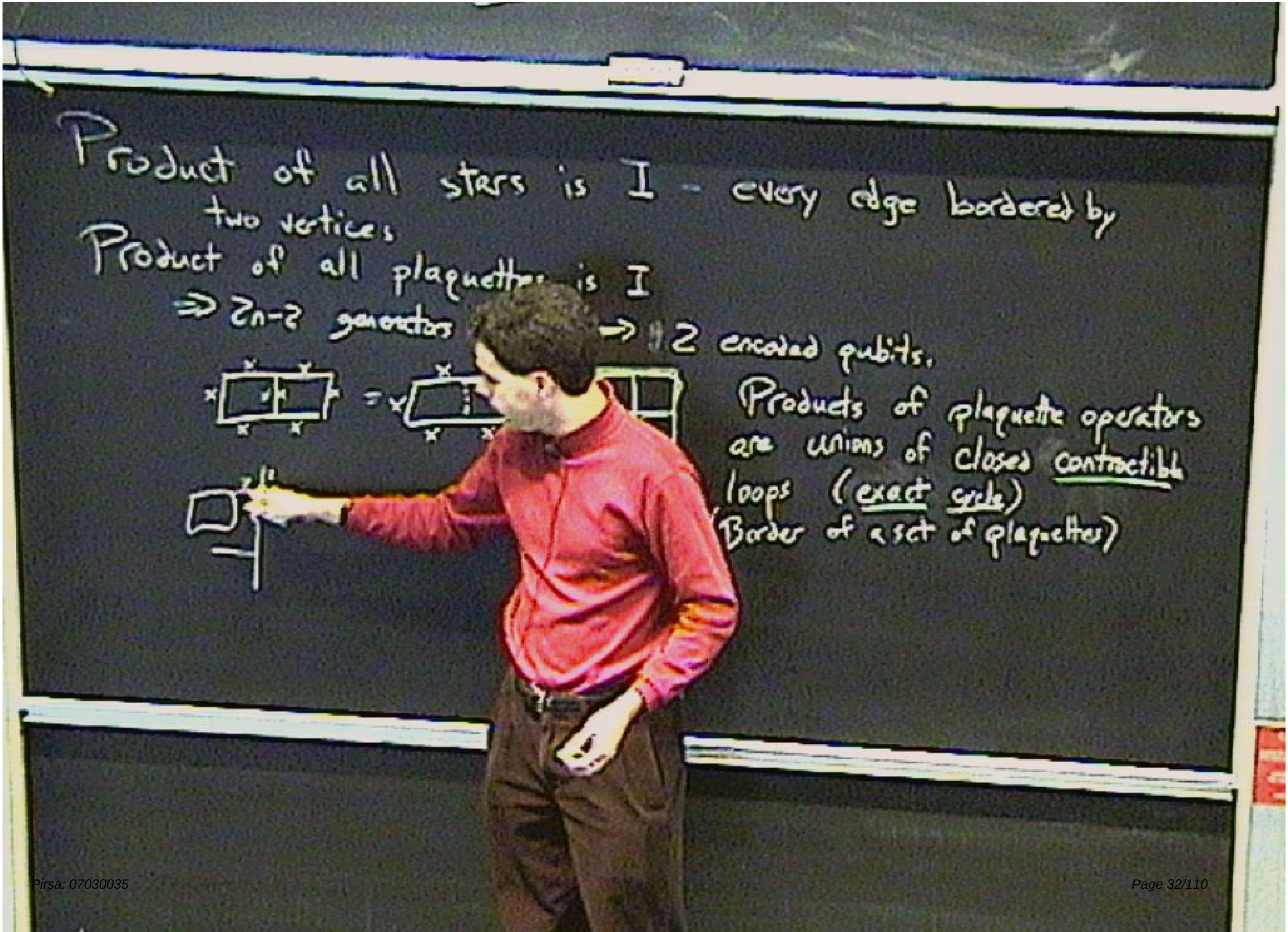
Product of all plaquettes is I

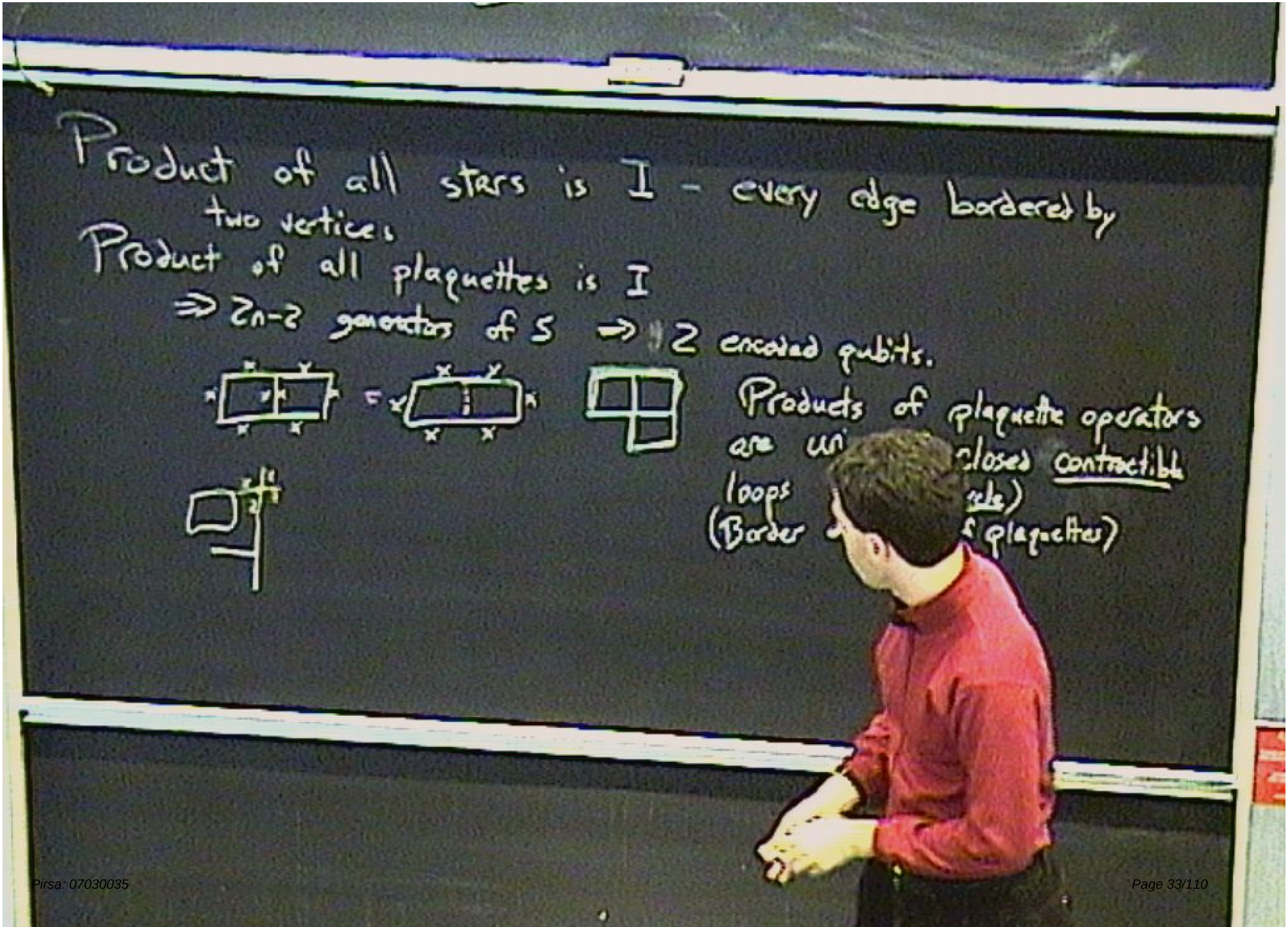
$\Rightarrow$ $2n-2$ generators of S $\Rightarrow$ 2 encoded qubits.

Products of plaquette operators are unions of closed contractible loops (exact cycle)

(Border of a set of plaquettes)


Product of all stars is I – every edge bordered by two vertices
Product of all plaquettes is I
⇒ 2n-2 generators of S → 2 encoded
plaquette operators
closed contractible

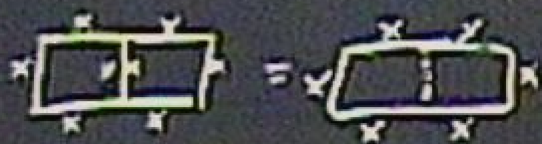

Product of all stars is I - every edge bordered by two vertices
Product of all plaquettes is I
⇒ 2n-2 generators
⇒ 2 encoded qubits.
Products of plaquette operators are unions of closed contractible loops (exact cycle)
(Border of a set of plaquettes)

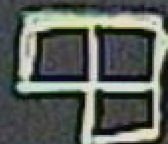

Product of all stars is I - every edge bordered by two vertices
Product of all plaquettes is I
⇒ 2n-2 generators of S ⇒ 2 encoded qubits.
Products of plaquette operators
are
closed contractible
loops
(Border
plaquettes)

Product of all stars is I – every edge bordered by two vertices

Product of all plaquettes is I

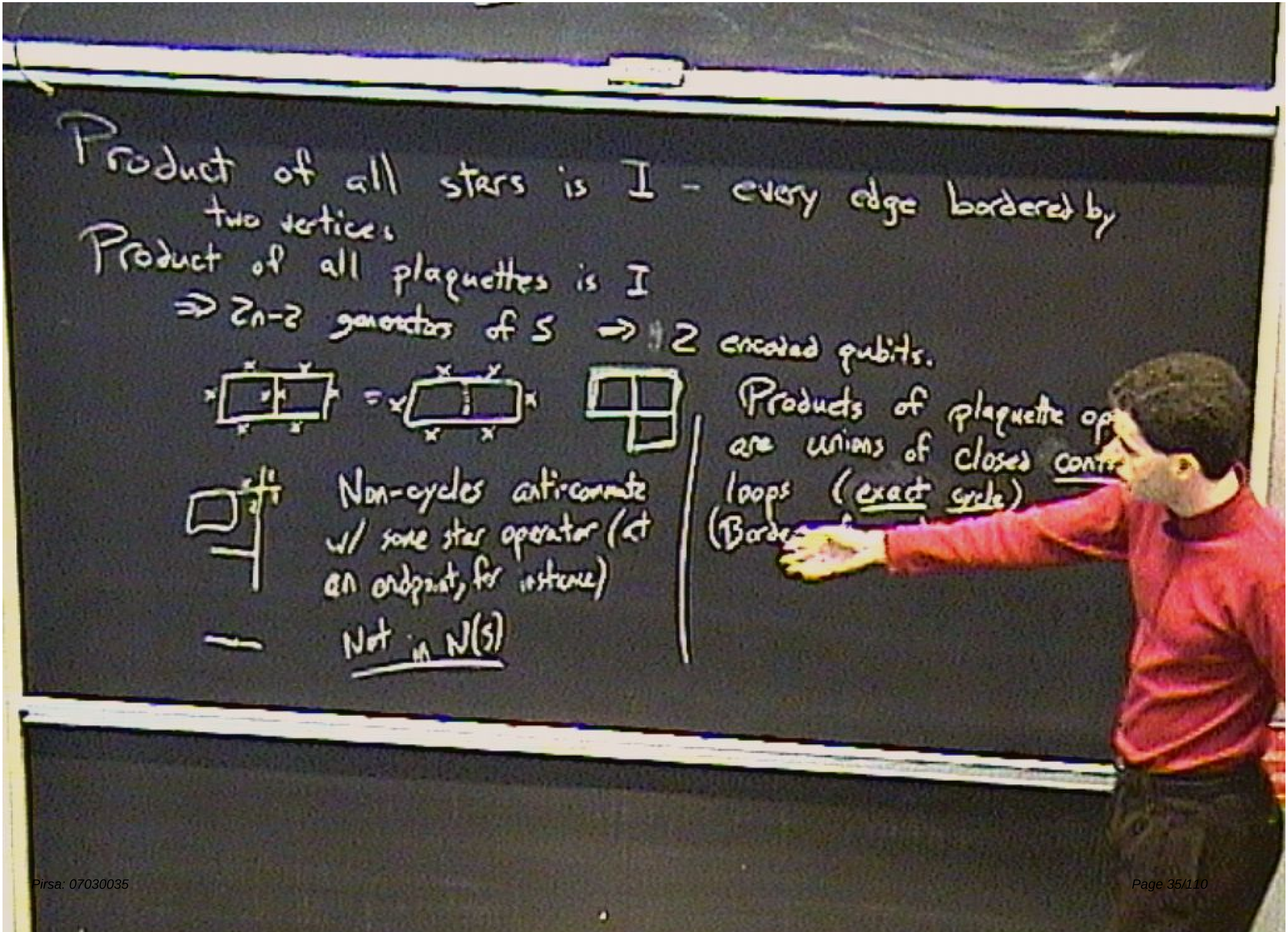
$\Rightarrow$ $2n-2$ generators of S $\Rightarrow$ 2 encoded qubits.





Non-cycles anticommute w/ some star operator (at an endpoint, for instance)

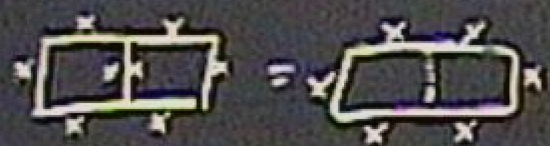
Products of plaquette operators are unions of closed contractible loops (exact cycle)
(Border of a set of plaquettes)

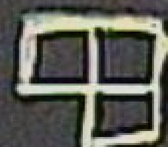

Product of all stars is I - every edge bordered by two vertices
Product of all plaquettes is I
⇒ 2n-2 generators of S → 2 encoded qubits.
Products of plaquette op
are unions of closed cont
loops (exact cycle)
Non-cycles anticommute w/ some star operator (at an endpoint, for instance)
Not in N(S)

Product of all stars is I – every edge bordered by two vertices.

Product of all plaquettes is I

$\Rightarrow$ $2n-2$ generators of S $\Rightarrow$ 2 encoded qubits.

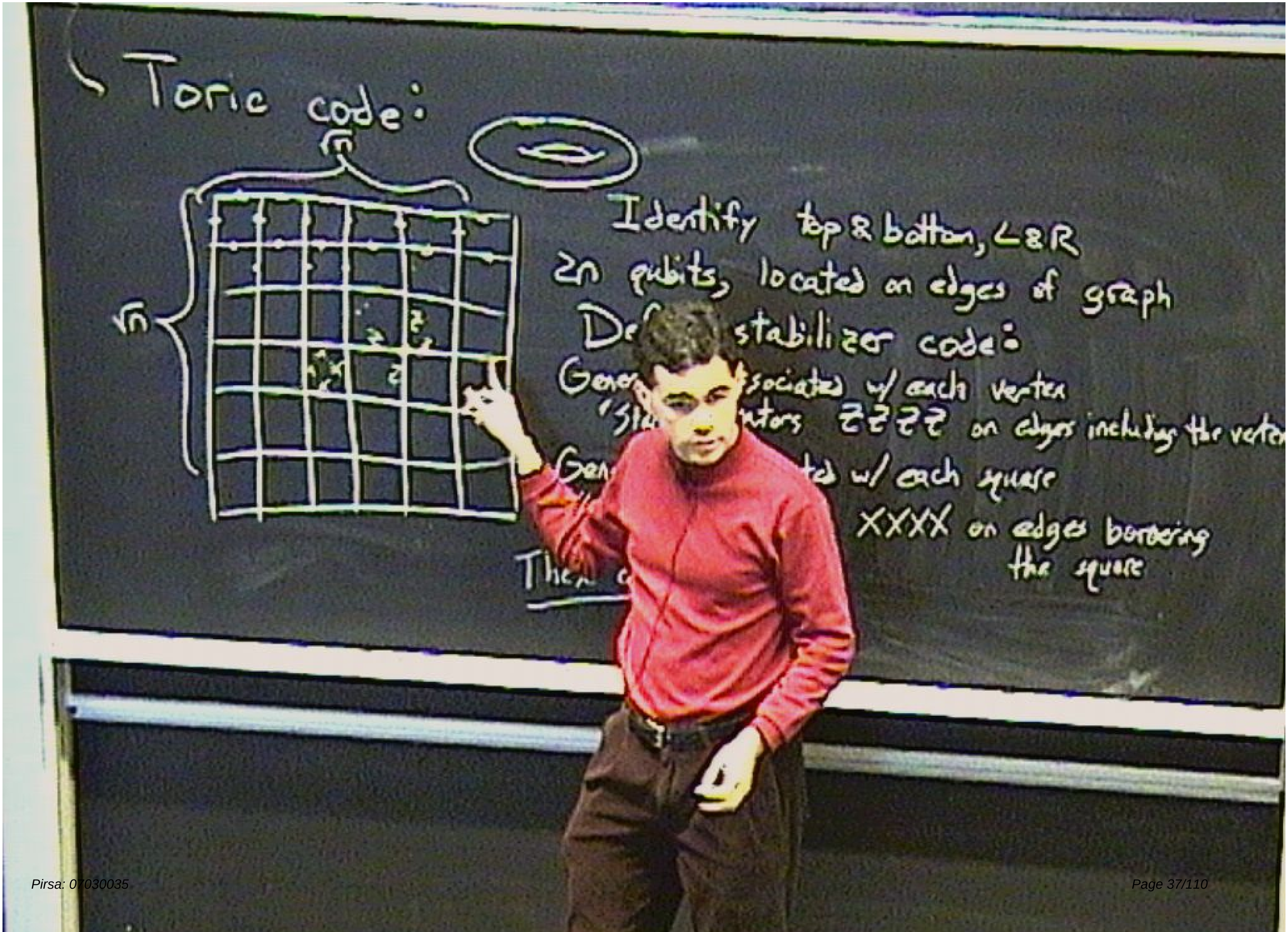


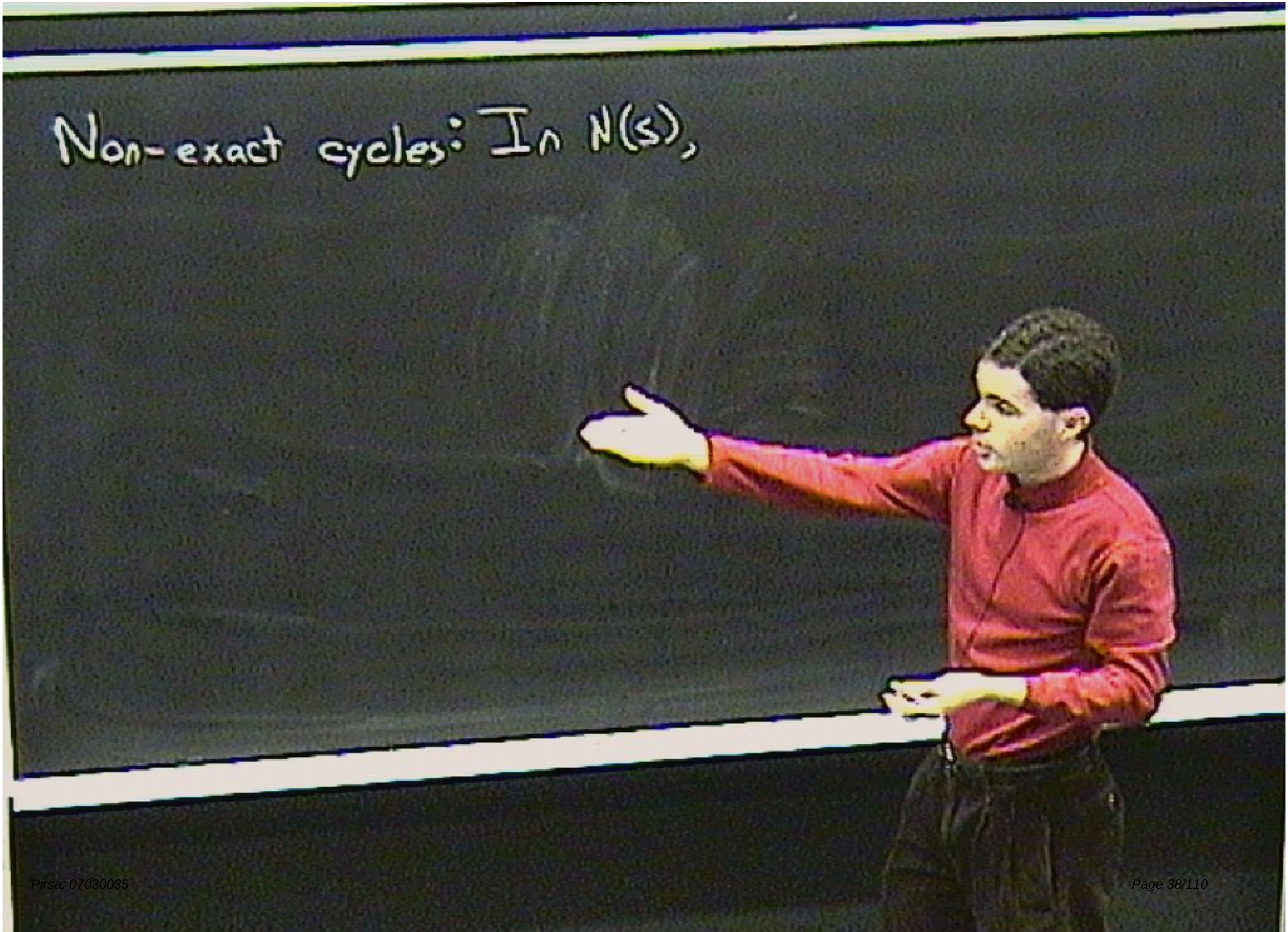


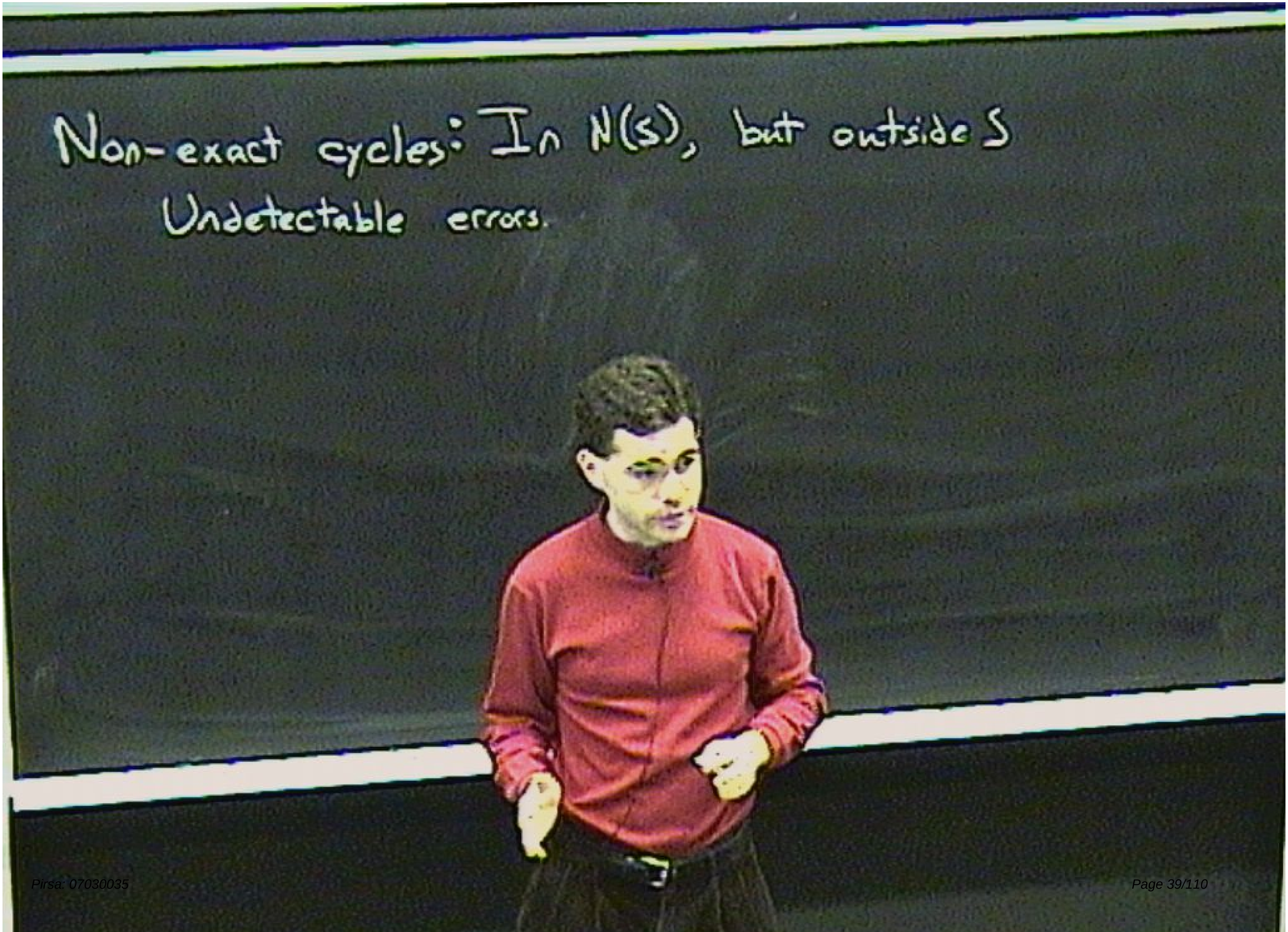
Products of plaquette operators are unions of closed <u>contractible</u> loops (<u>exact</u> <u>cycle</u>) (Border of a set of plaquettes)

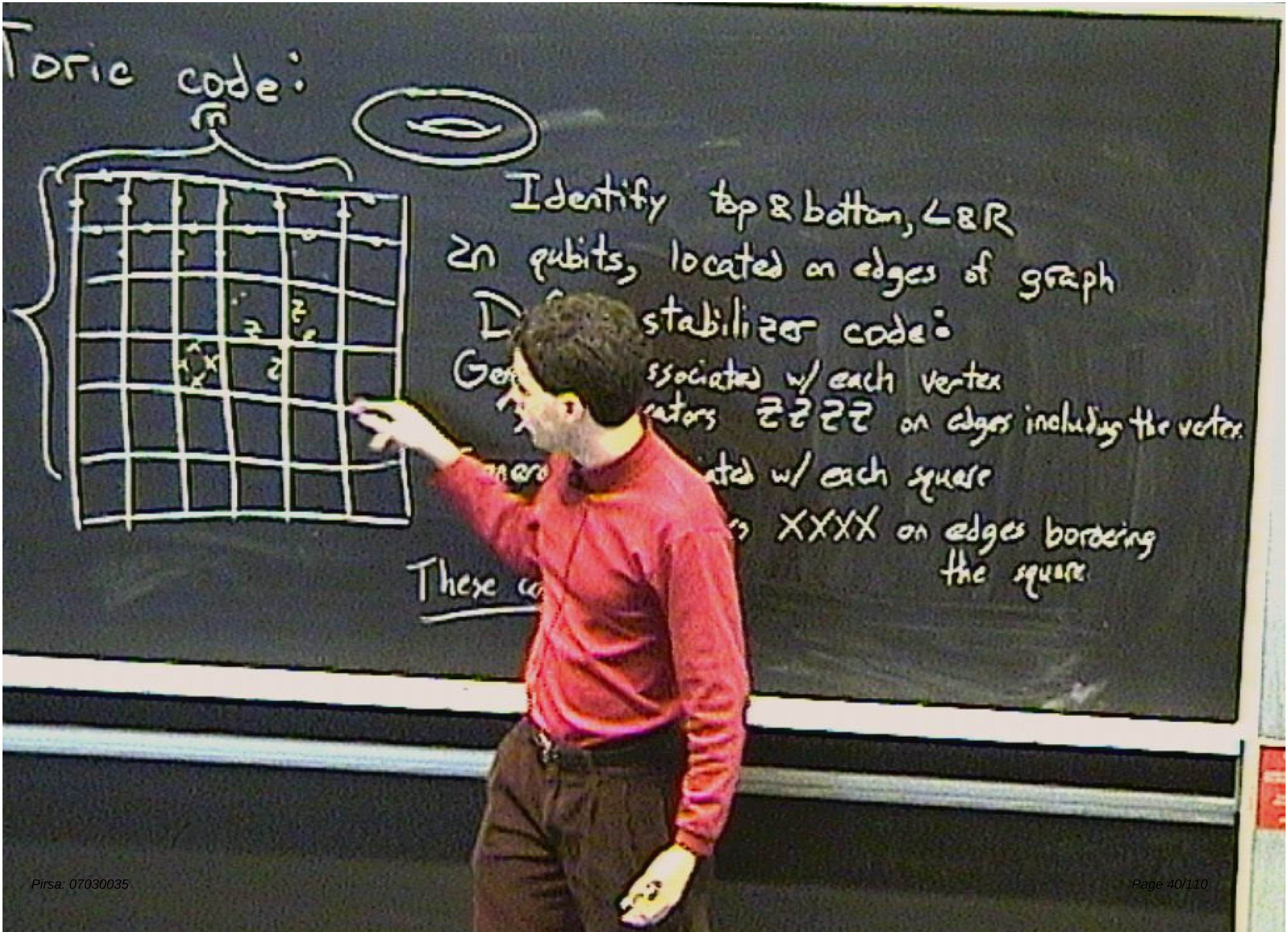
Non-cycles anticommute w/ some star operator (at an endpoint, for instance)

<u>Not in $N(S)$</u>


Toric code:
Identify top & botton, L & R
2n qubits, located on edges of graph
stabilizer code:
w/ each vertex
ZZZZ on edges including the vertex
w/ each square
XXXX on edges bordering
the square


Non-exact cycles: In N(s),


Non-exact cycles: In N(S), but outside S
Undetectable errors.



Toric code:
n
Identify top & bottom, L&R
2n qubits, located on edges of graph
stabilizer code:
w/ each vertex
ZZZZ on edges including the vertex
w/ each square
XXXX on edges bordering the square
These

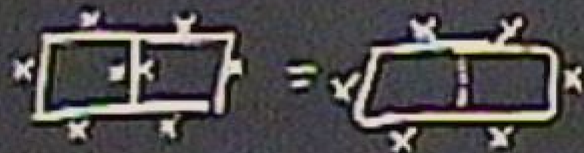
Non-exact cycles: In $N(S)$, but outside S

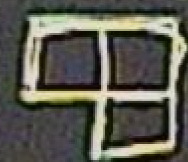
Undetectable errors. – shortest has weight $\sqrt{n}$.

Product of all stars is I – every edge bordered by two vertices

Product of all plaquettes is I

$\Rightarrow$ $2n-2$ generators of S $\Rightarrow$ # 2 encoded qubits.



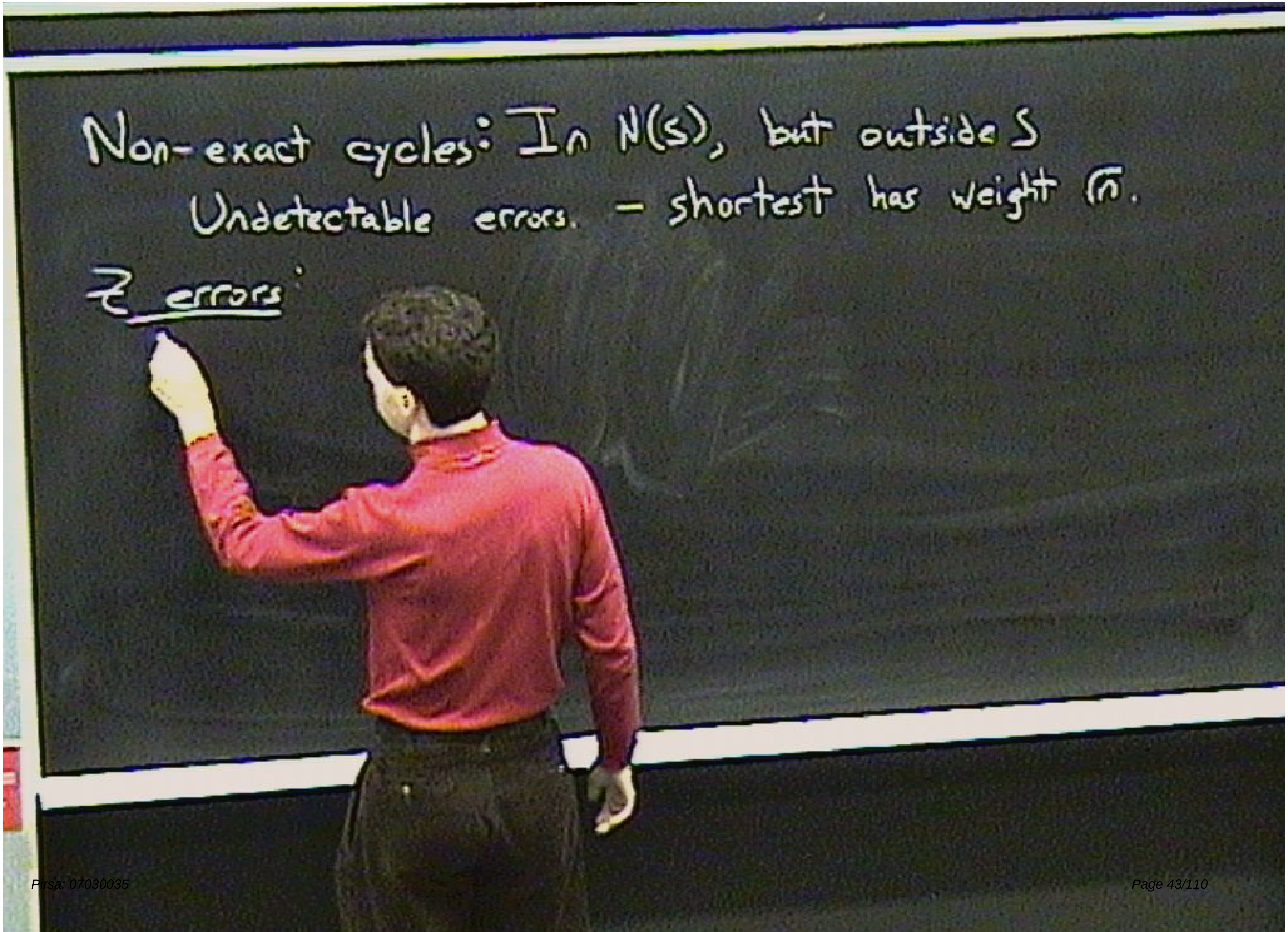


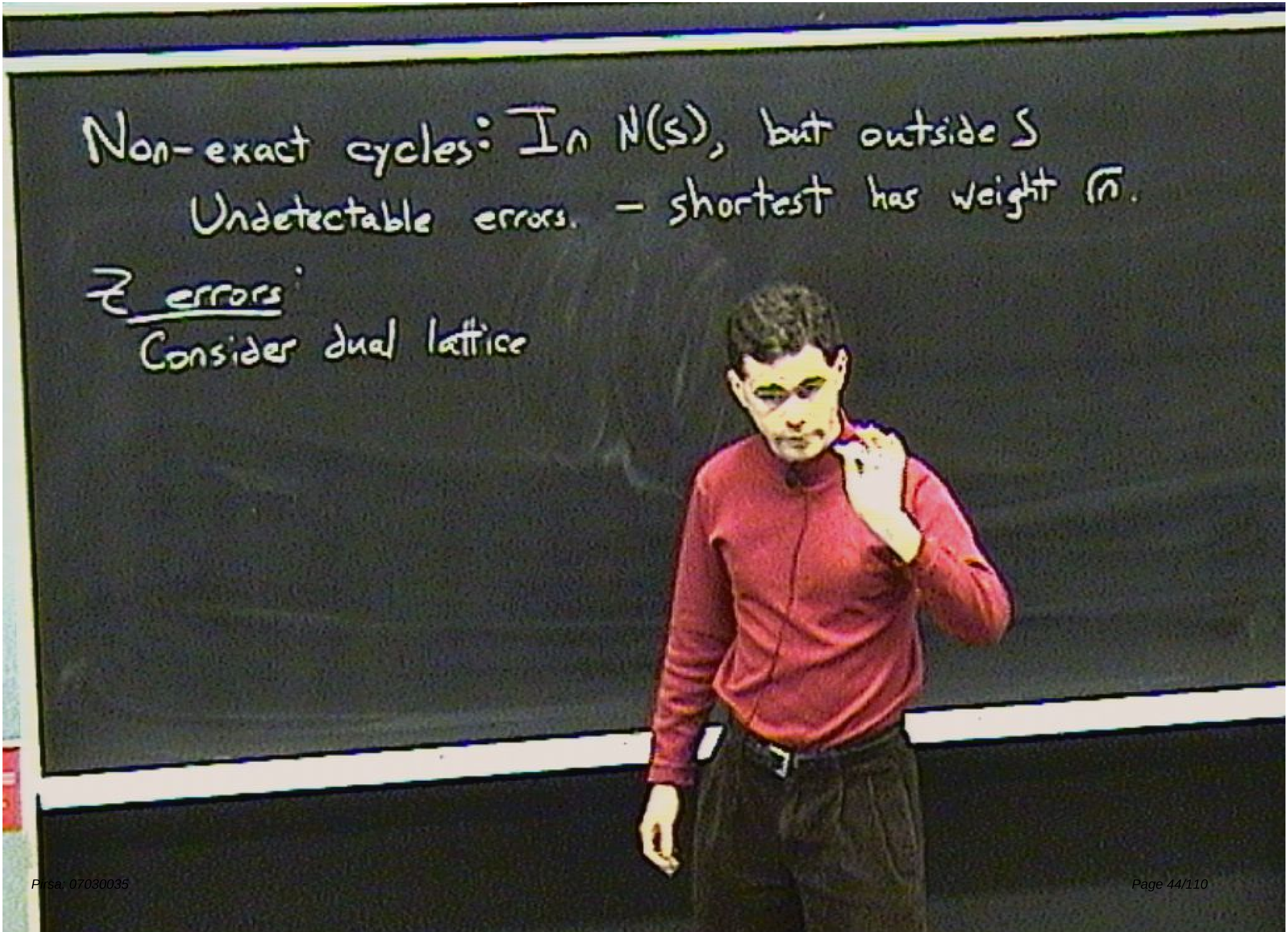
Products of plaquette operators are unions of closed <u>contractible</u> loops (<u>exact</u> <u>cycle</u>)

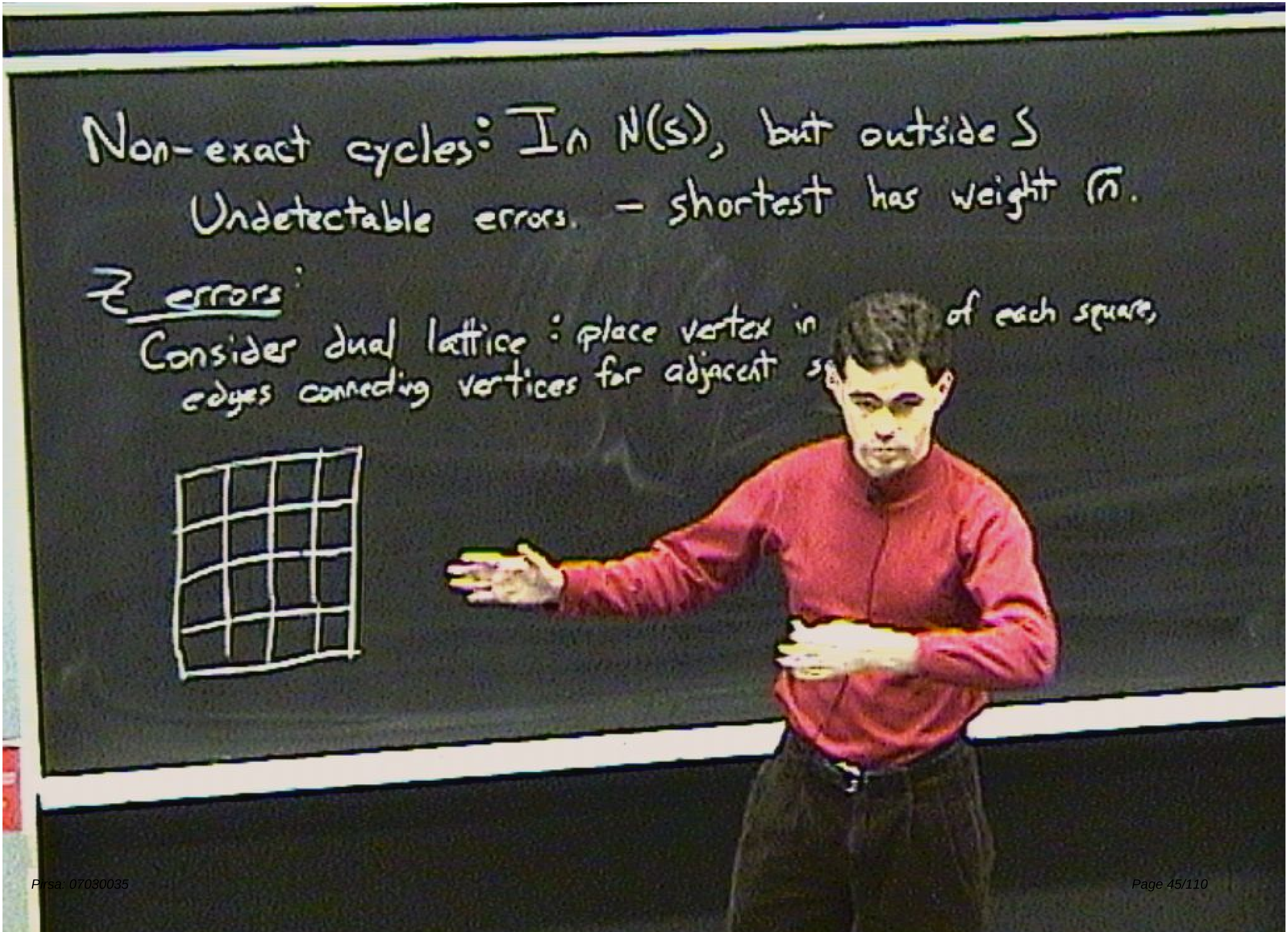
(Border of a set of plaquettes)

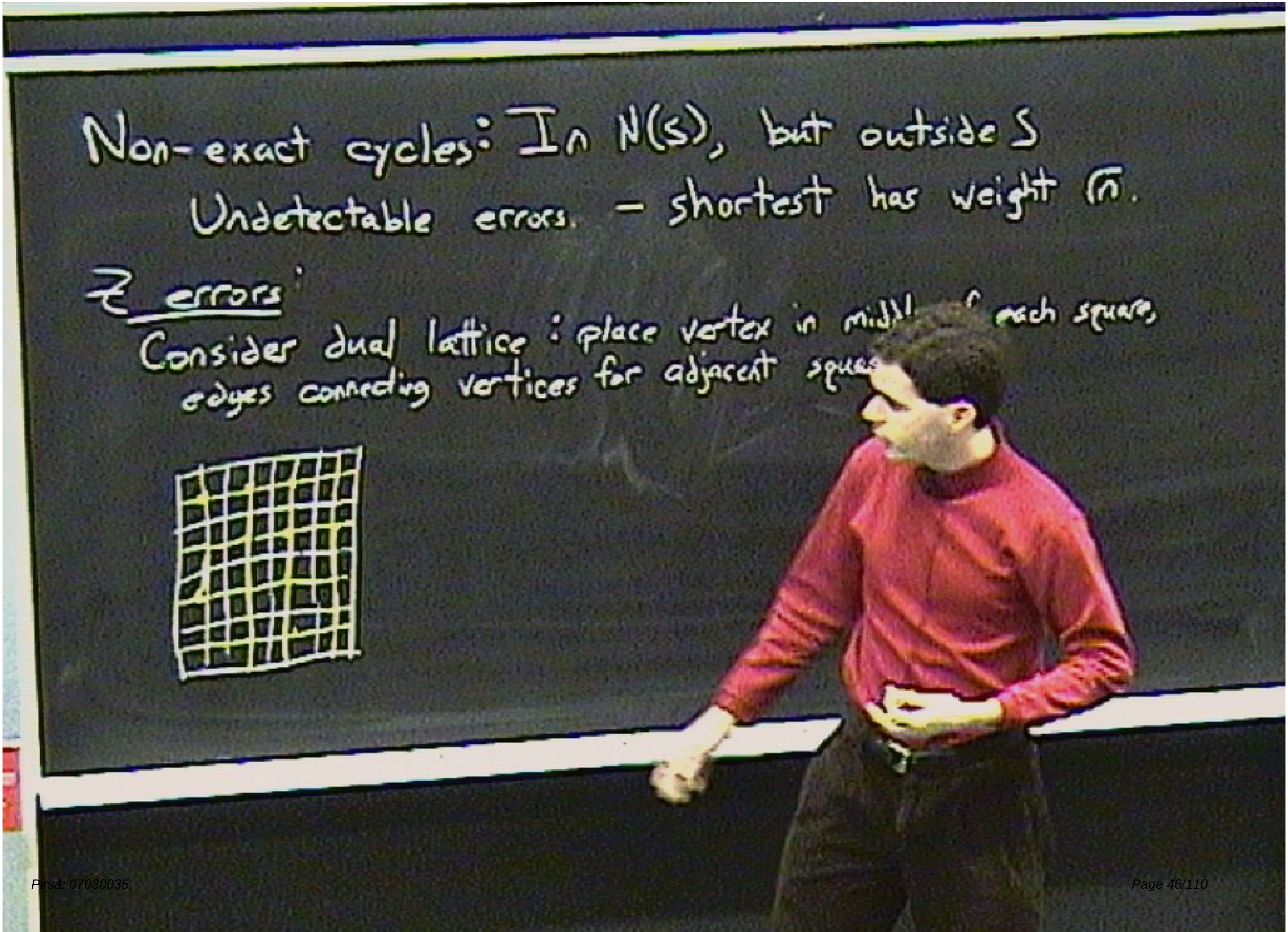
Non-cycles anti-commute w/ some star operator (at an endpoint, for instance)

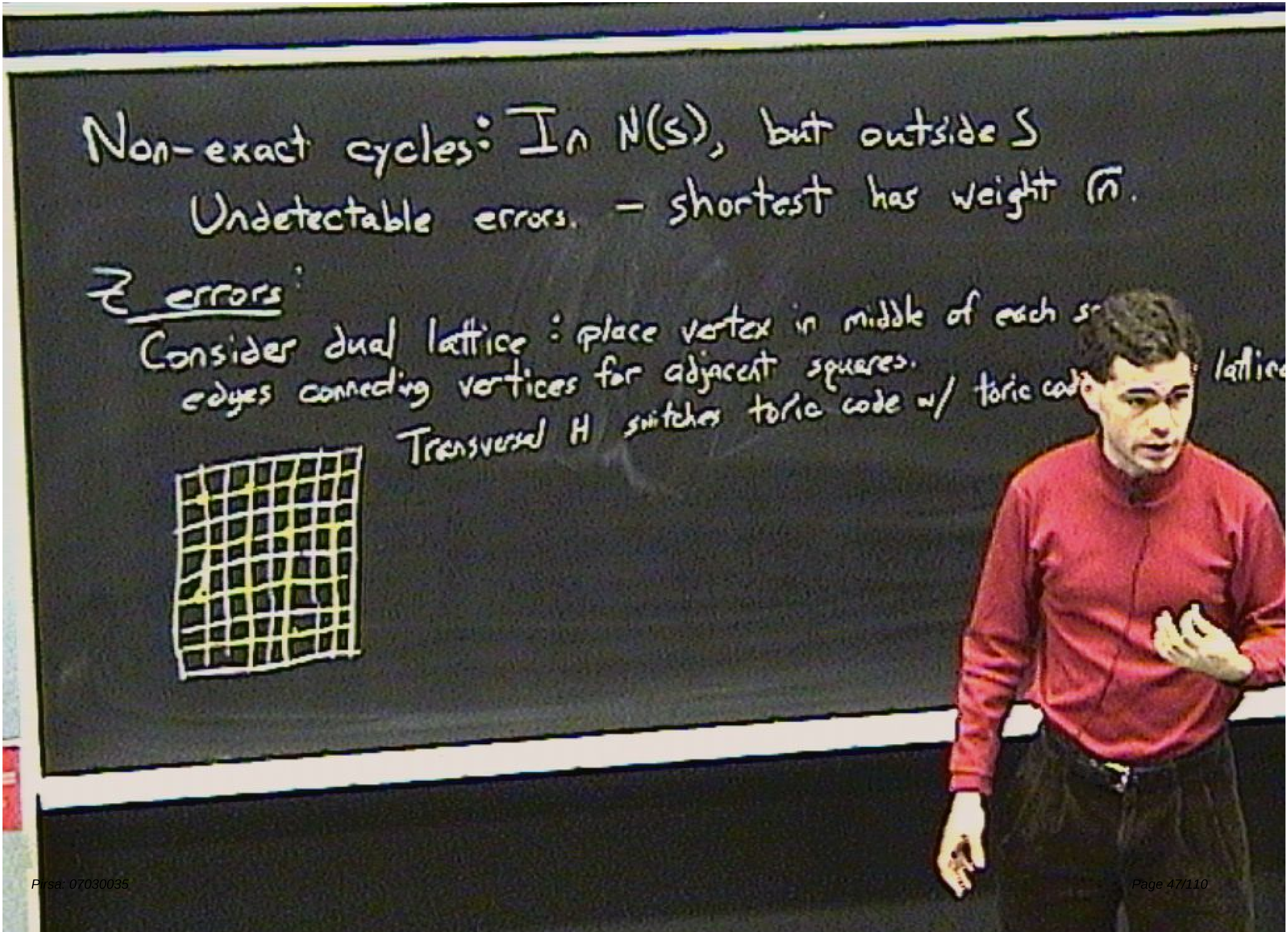
<u>Not in $N(S)$</u>


Non-exact cycles: In N(S), but outside S
Undetectable errors. – shortest has weight √n.
Z errors:


Non-exact cycles: In N(S), but outside S
Undetectable errors. – shortest has weight √n.
Z errors:
Consider dual lattice


Non-exact cycles: In N(S), but outside S
Undetectable errors. – shortest has weight √n.
Z errors:
Consider dual lattice: place vertex in
of each square,
edges connecting vertices for adjacent


Non-exact cycles: In N(S), but outside S
Undetectable errors. – shortest has weight √n.
Z errors:
Consider dual lattice: place vertex in
each square,
edges connecting vertices for adjacent

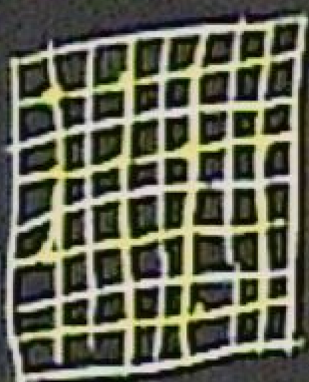

Non-exact cycles: In N(S), but outside S
Undetectable errors. – shortest has weight √n.
Z errors:
Consider dual lattice: place vertex in middle of each
edges connecting vertices for adjacent squares.
lattice
Transversal H switches toric code w/ toric

Non-exact cycles: In $N(S)$, but outside S

Undetectable errors. – shortest has weight $\sqrt{n}$.

Z errors:

Consider dual lattice: place vertex in middle of each square, edges connecting vertices for adjacent squares.



Transversal H switches toric code w/ toric code on dual lattice

Z stabilizer elements are exact cycles on dual lattice (cocycles)

Undetectable Z errors – non-exact cocycles.

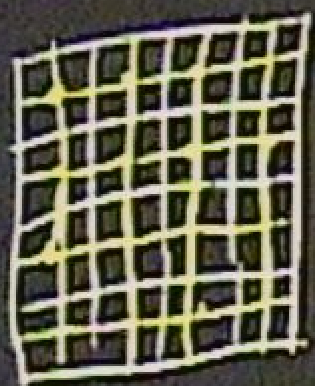
Non-exact cycles: In $N(S)$, but outside S

Undetectable errors. – shortest has weight $\sqrt{n}$.

<u>Z errors</u>:

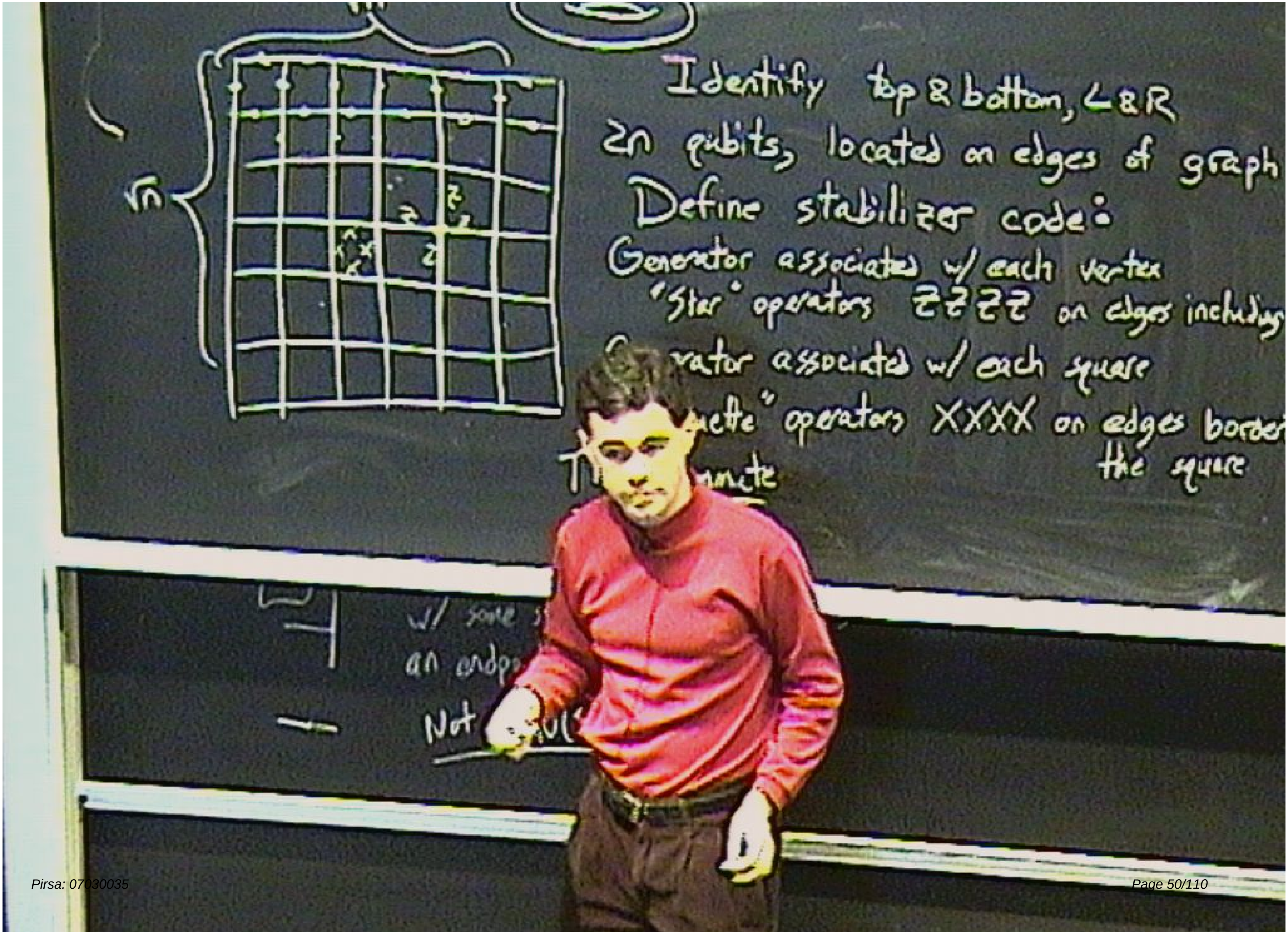
Consider dual lattice: place vertex in middle of each square, edges connecting vertices for adjacent squares.

Transversal H switches toric code w/ toric code on dual lattice



Z stabilizer elements are exact cycles on dual lattice
(<u>cocycles</u>)

Undetectable Z errors – non-exact cocycles.


√n
Identify top & bottom, L & R
2n qubits, located on edges of graph
Define stabilizer code:
Generator associated w/ each vertex
"Star" operators ZZZZ on edges including
rator associated w/ each square
uette" operators XXXX on edges border
the square
mute
w/ some
an endp
Not

Non-exact cycles: In $N(S)$, but outside S

Undetectable errors. – shortest has weight $\sqrt{n}$.

Z errors:

Consider dual lattice: place vertex in middle of each square, edges connecting vertices for adjacent squares.

Transversal H switches toric code w/ toric code on dual lattice

Z stabilizer elements are exact cycles on dual lattice (cocycles)

Undetectable Z errors – non-exact cocycles.

$\Rightarrow$ Code is $[[2n, 2, \sqrt{n}]]$.

Non-exact cycles: In $N(S)$, but outside S

Undetectable errors. – shortest has weight $\sqrt{n}$.

Z errors:

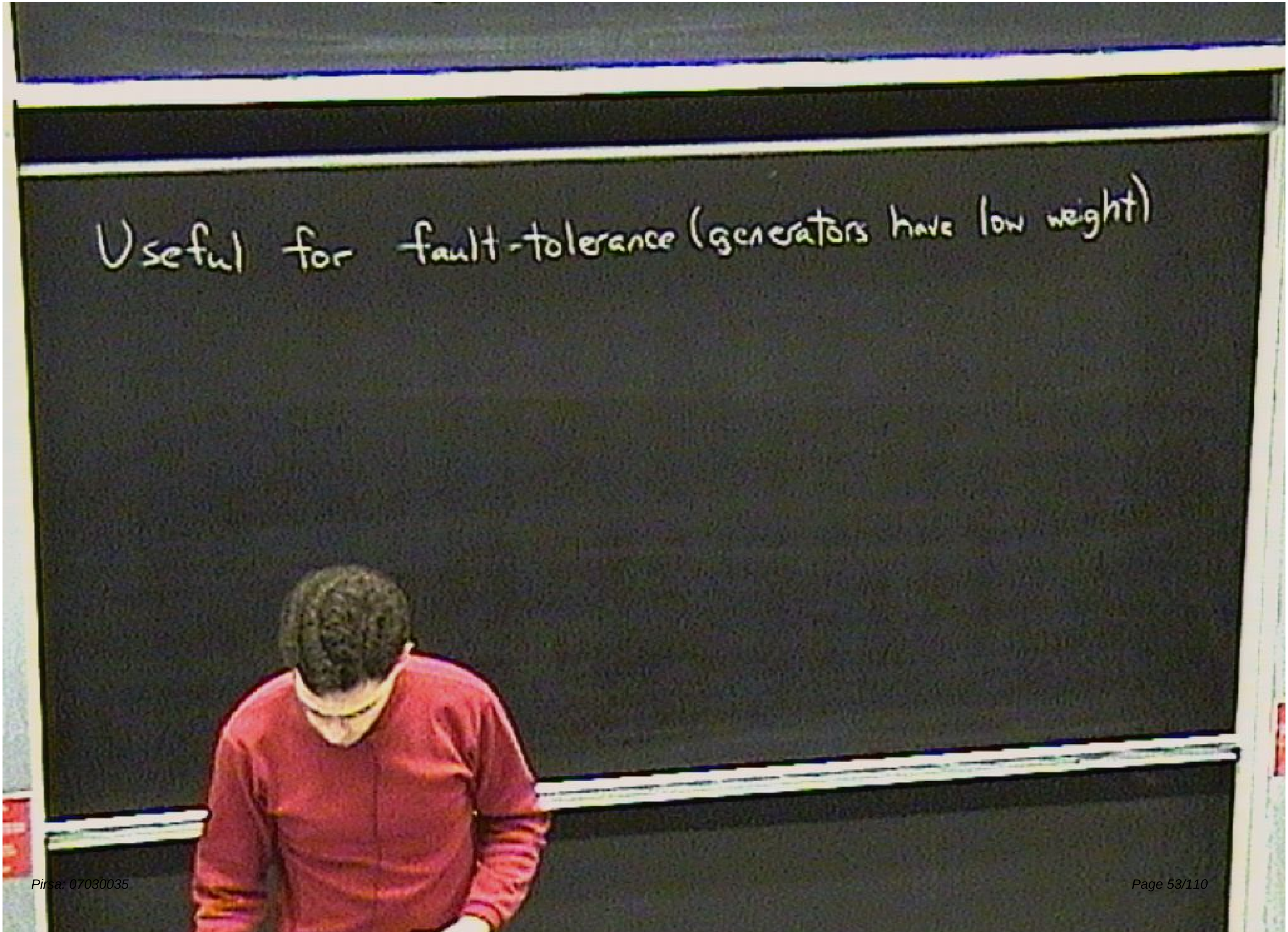
Consider dual lattice: place vertex in middle of each square, edges connecting vertices for adjacent squares.

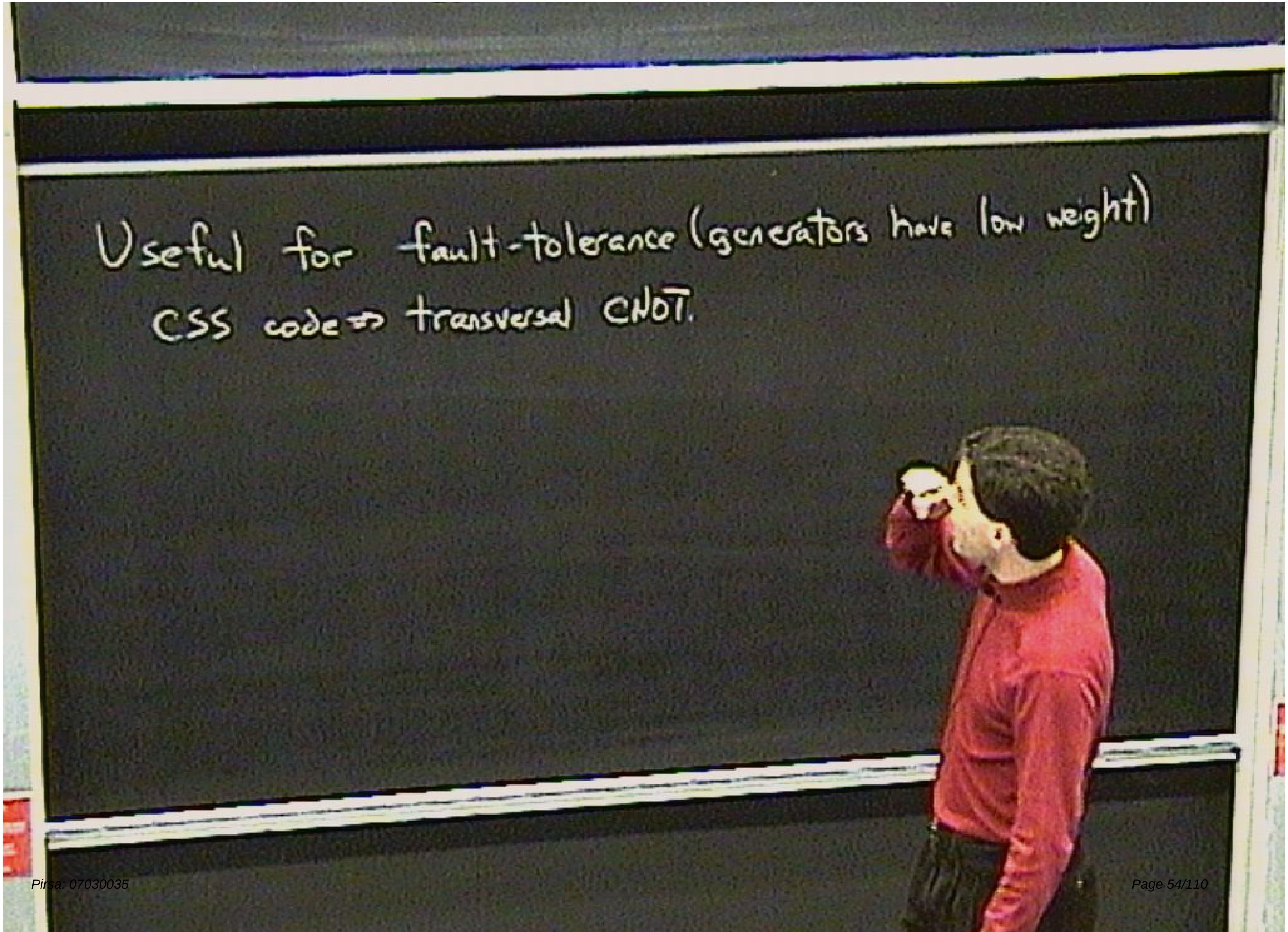
Transversal H switches toric code w/ toric code on dual lattice

Z stabilizer elements are exact cycles on dual lattice (cocycles)

Undetectable Z errors – non-exact cocycles.

$\Rightarrow$ Code is $[[2n, 2, \sqrt{n}]]$. (Codewords are associated w/ $\mathbb{Z}_2$ homology of torus.)

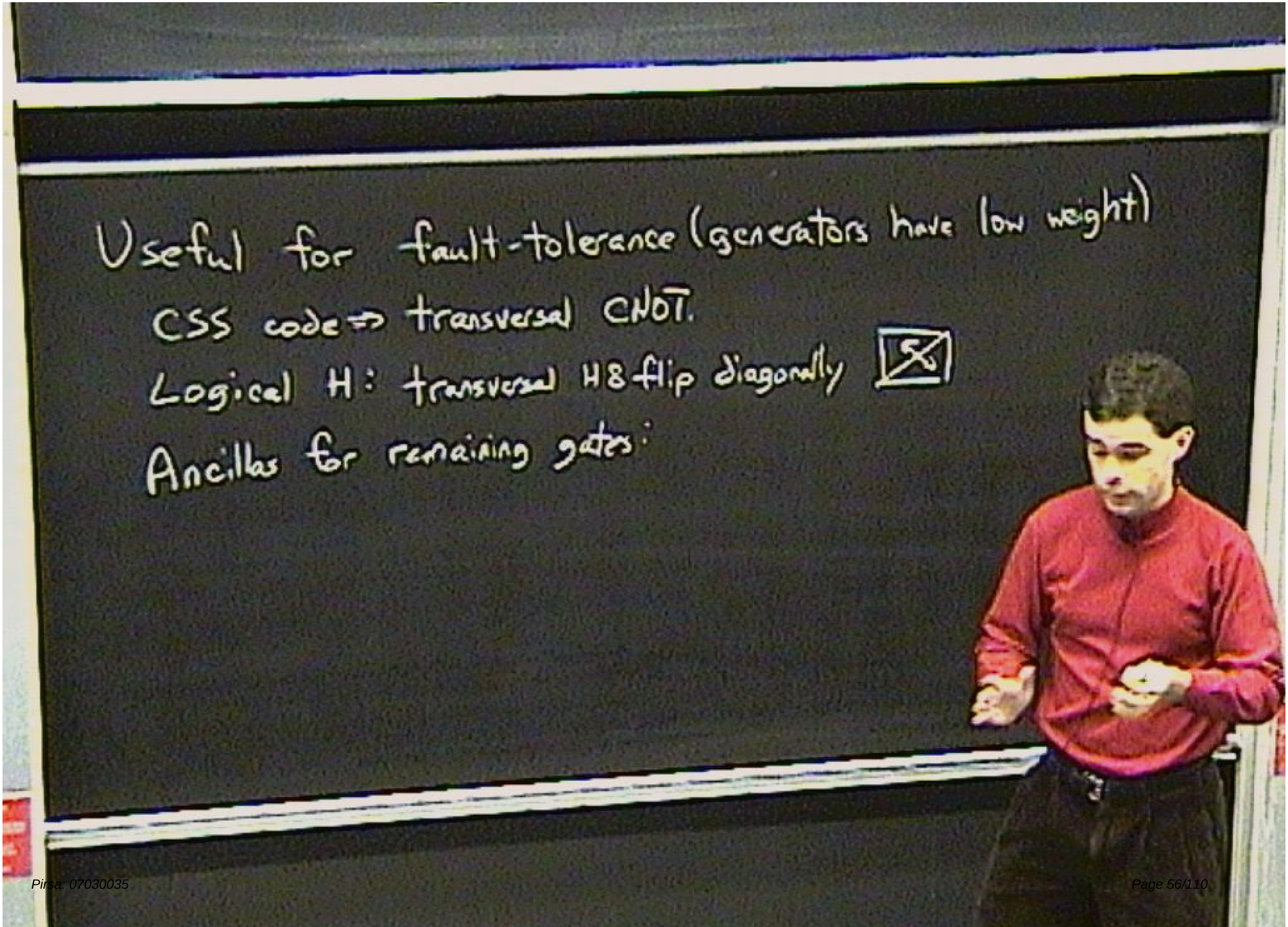

Useful for fault-tolerance (generators have low weight)

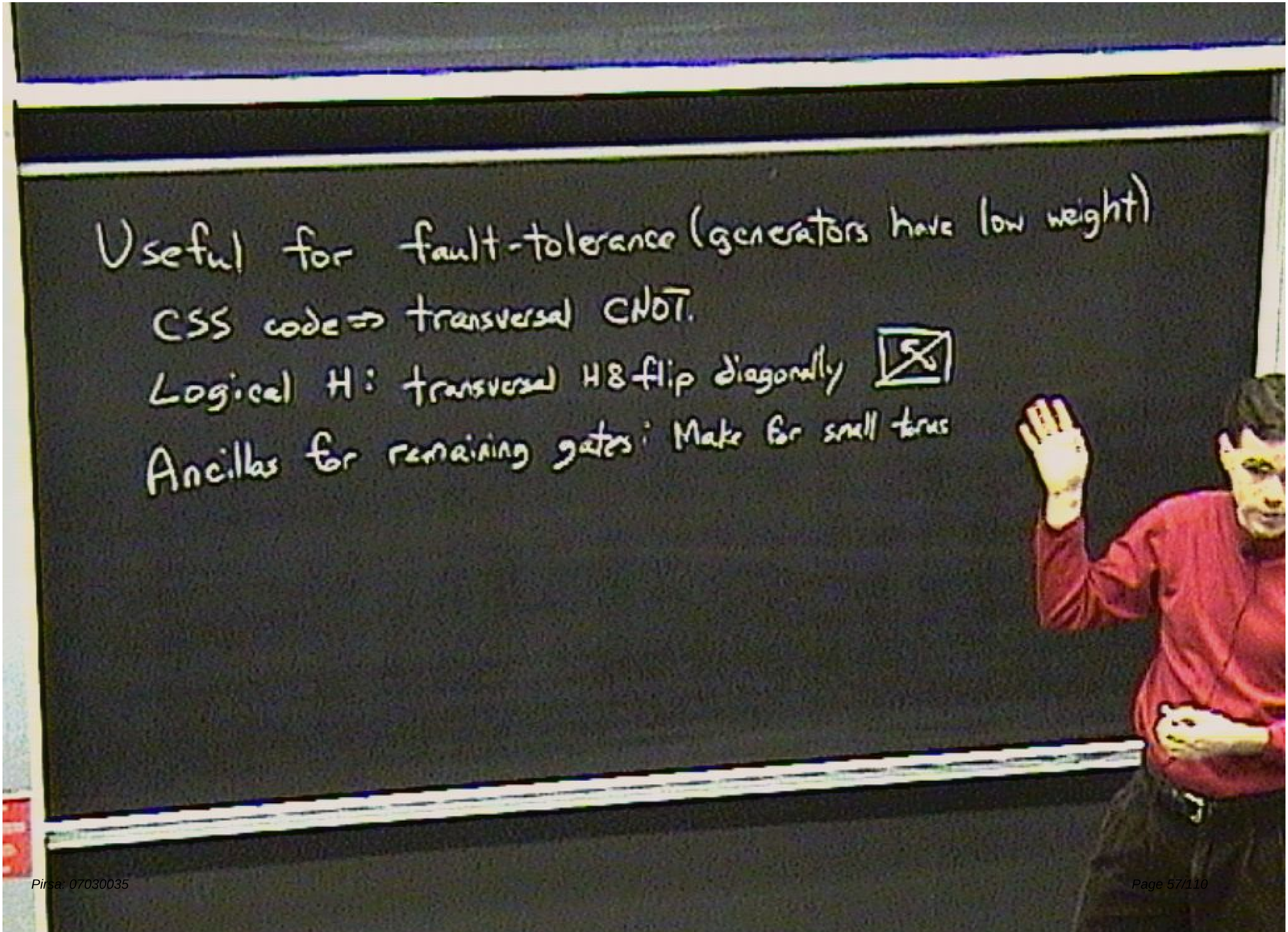

Useful for fault-tolerance (generators have low weight)
CSS code ⇒ transversal CNOT.

Useful for fault-tolerance (generators have low weight)

CSS code $\Rightarrow$ transversal CNOT.

Logical H: transversal H & flip diagonally 


Useful for fault-tolerance (generators have low weight)
CSS code ⇒ transversal CNOT.
Logical H: transversal H & flip diagonally
Ancillas for remaining gates:


Useful for fault-tolerance (generators have low weight)
CSS code ⇒ transversal CNOT.
Logical H: transversal H & flip diagonally
Ancillas for remaining gates: Make for small torus

Useful for fault-tolerance (generators have low weight)

CSS code $\Rightarrow$ transversal CNOT.

Logical H: transversal H & flip diagonally 

Ancillas for remaining gates: Make for small torus

Grow torus by adding rows & columns of qubits.

Useful for fault-tolerance (generators have low weight)

CSS code ⇒ transversal CNOT.

Logical H: transversal H & flip diagonally

Ancillas for remaining gates: Make for small torus

Grow torus by adding rows & columns of qubits.

Distill ancillas to improve reliability

Useful for fault-tolerance (generators have low weight)

CSS code ⇒ transversal CNOT.

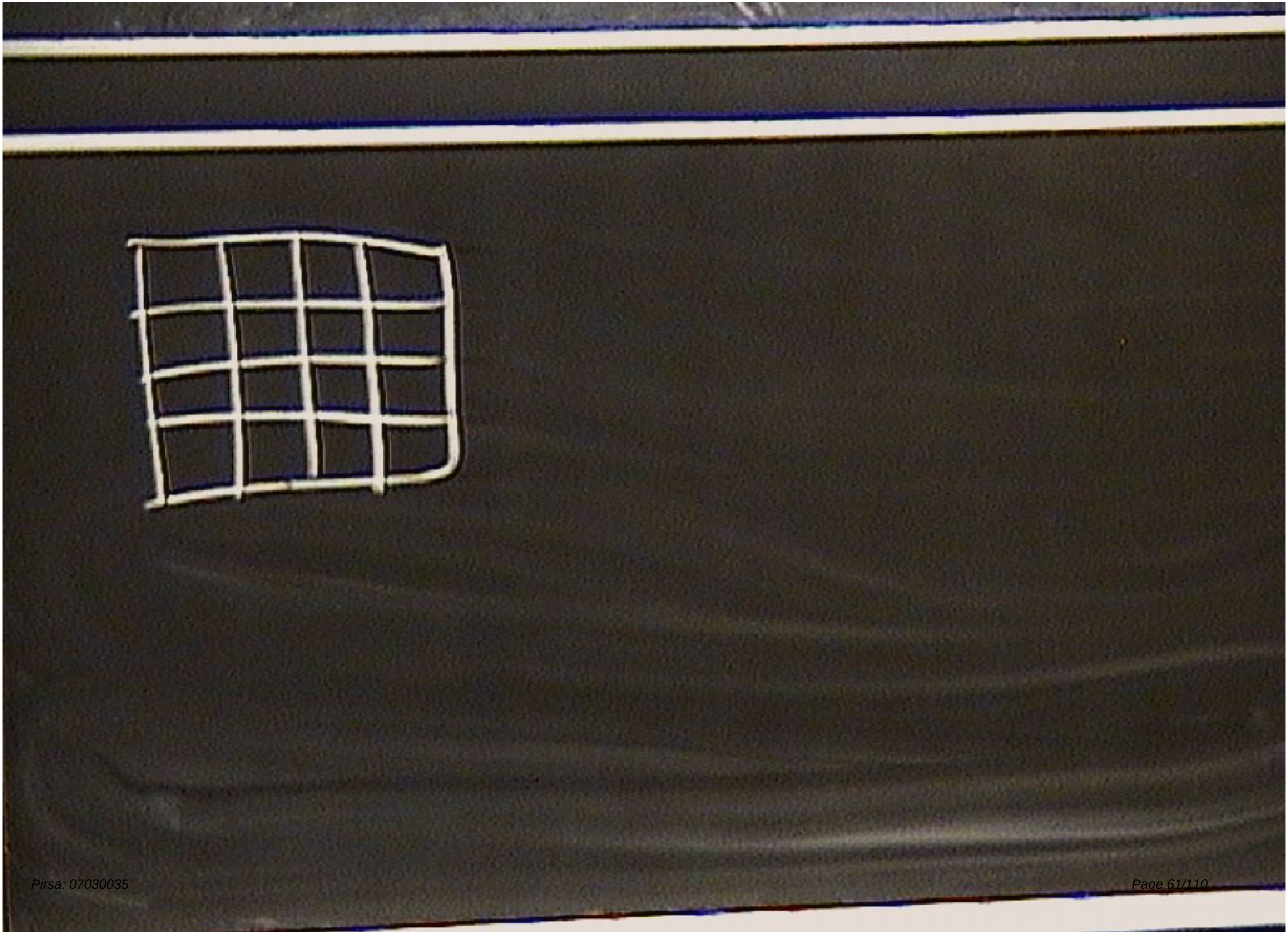
Logical H: transversal H & flip diagonally 

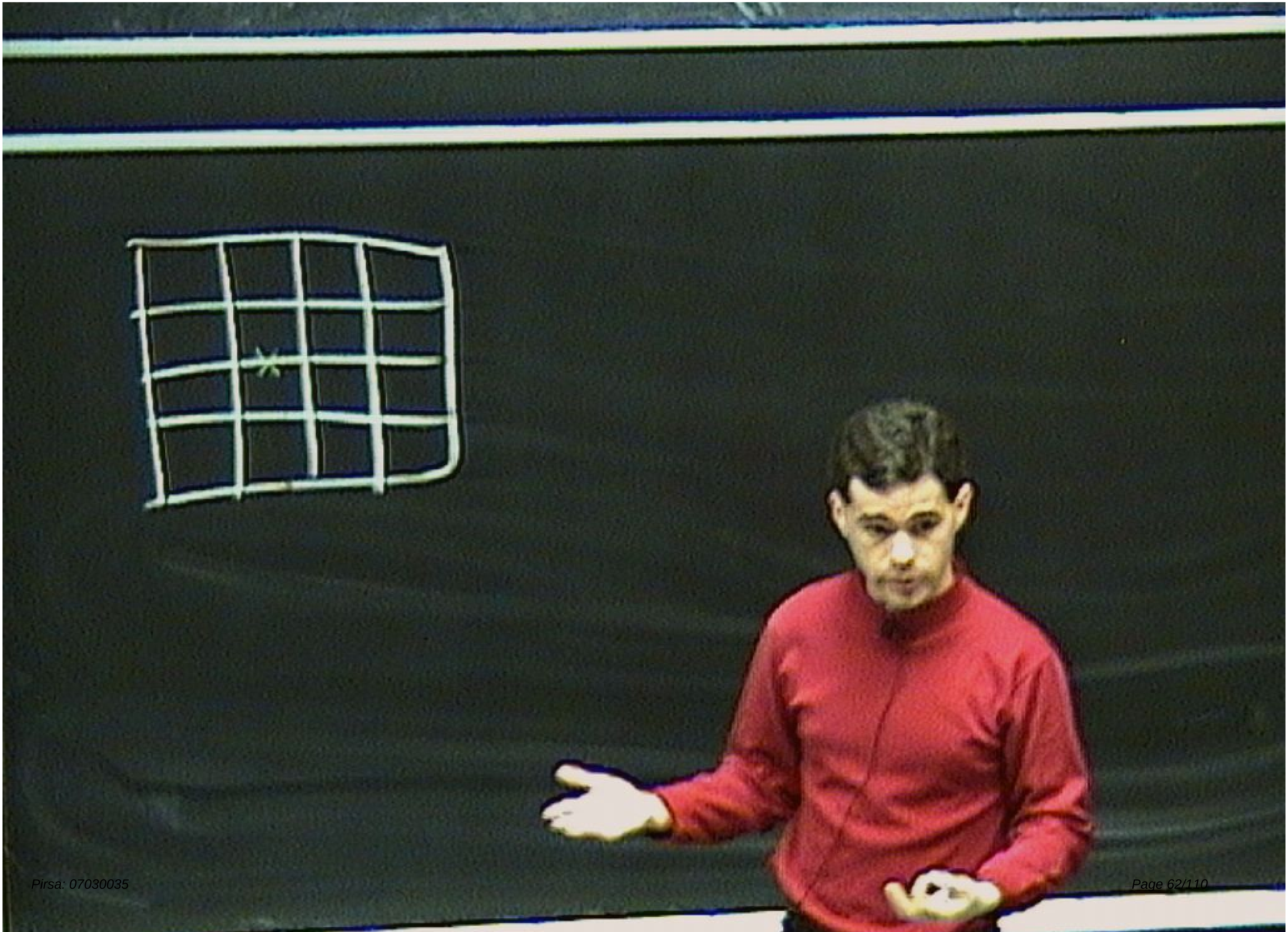
Ancillas for remaining gates: Make for small torus

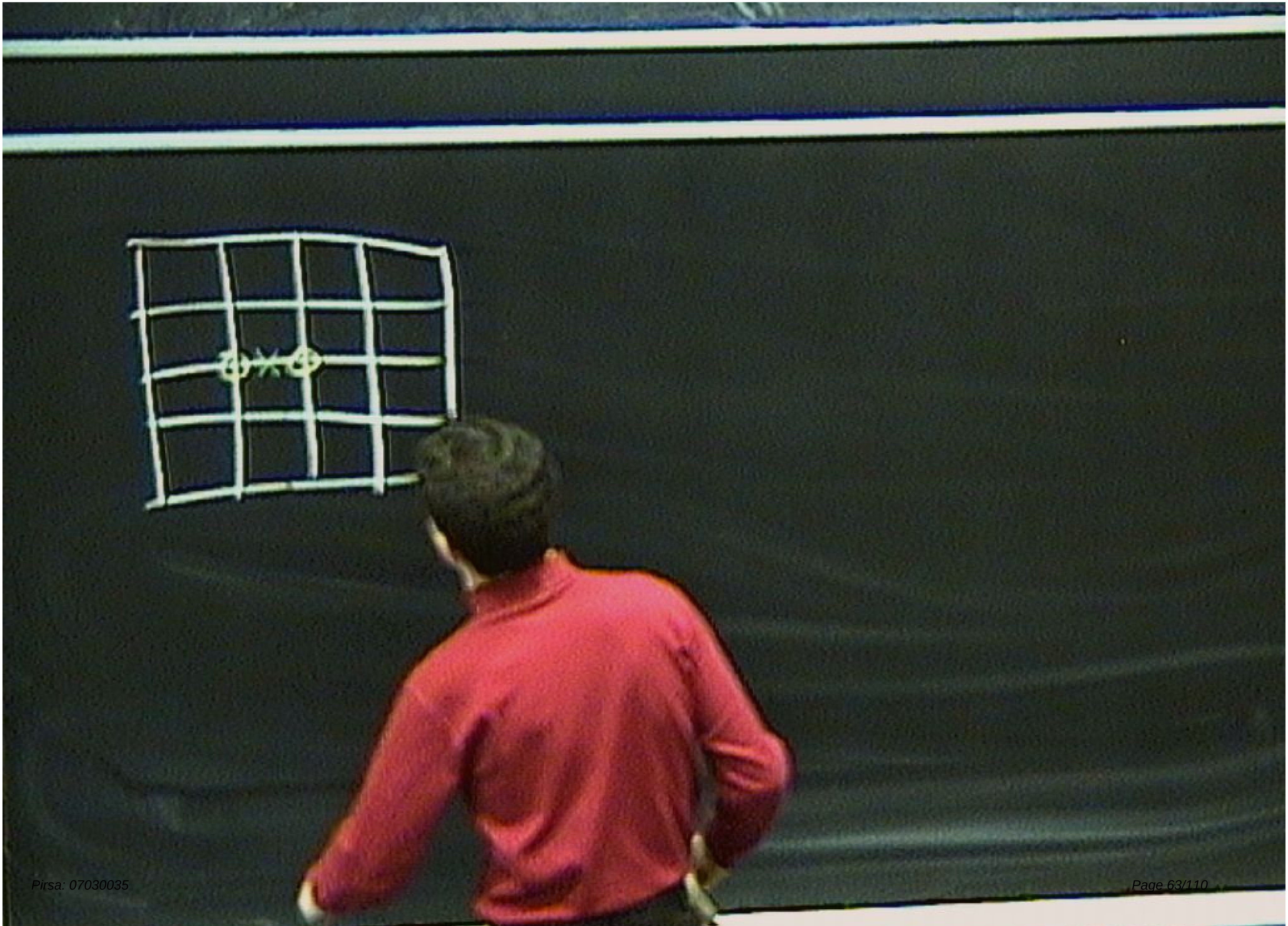
Grow torus by adding rows & columns of qubits.

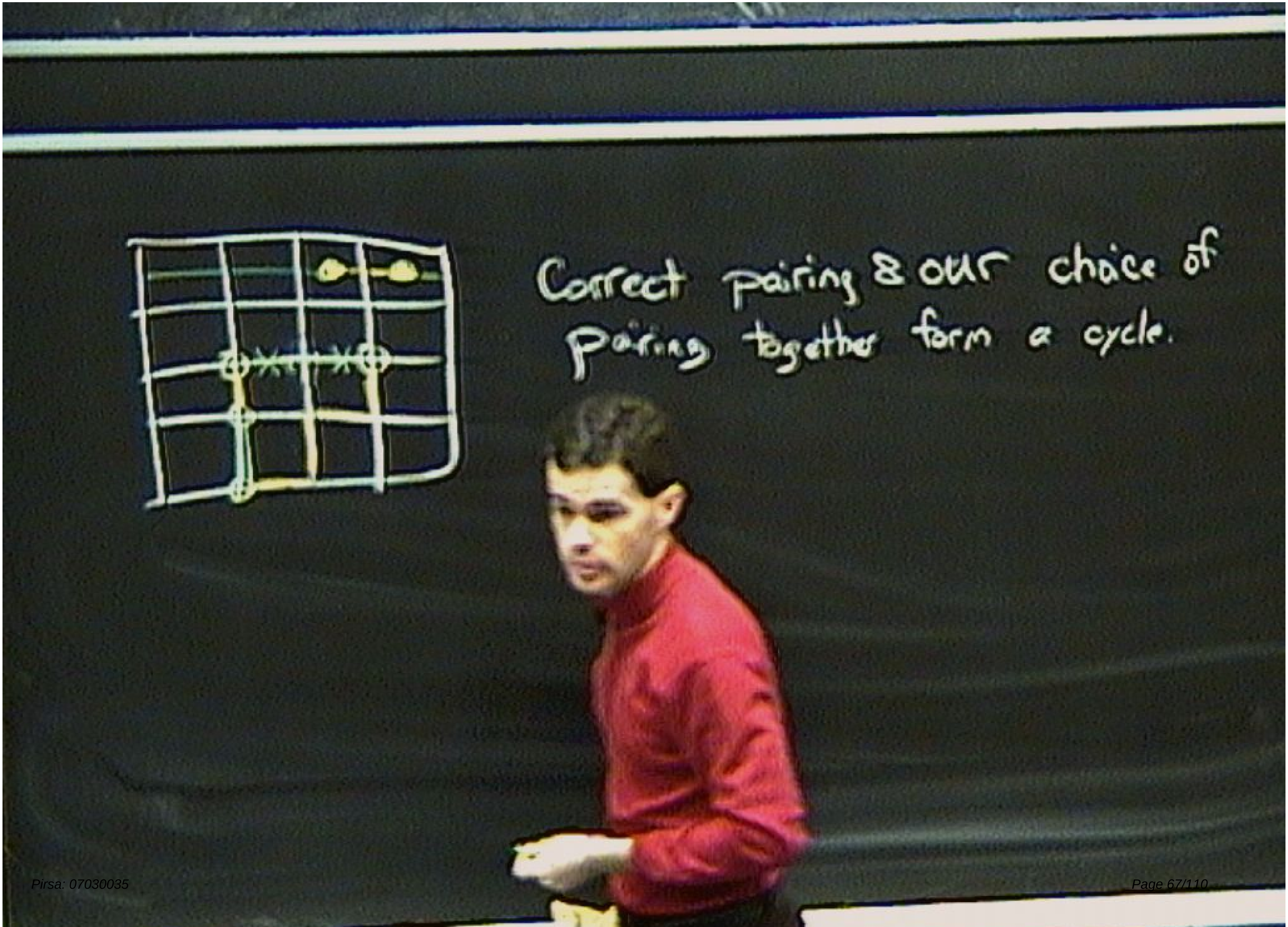
Distill ancillas to improve reliability

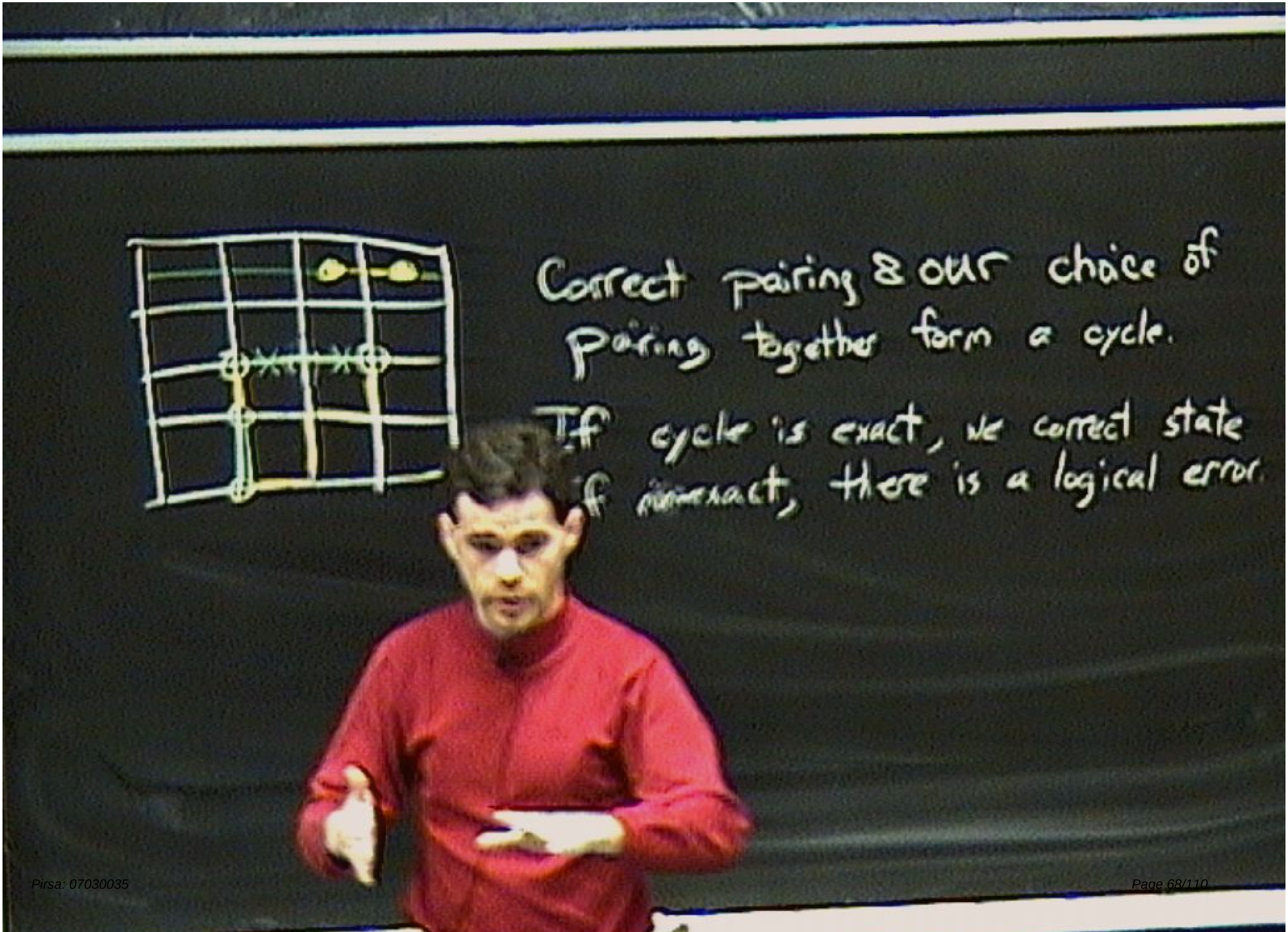
Get threshold result w/ large toric codes

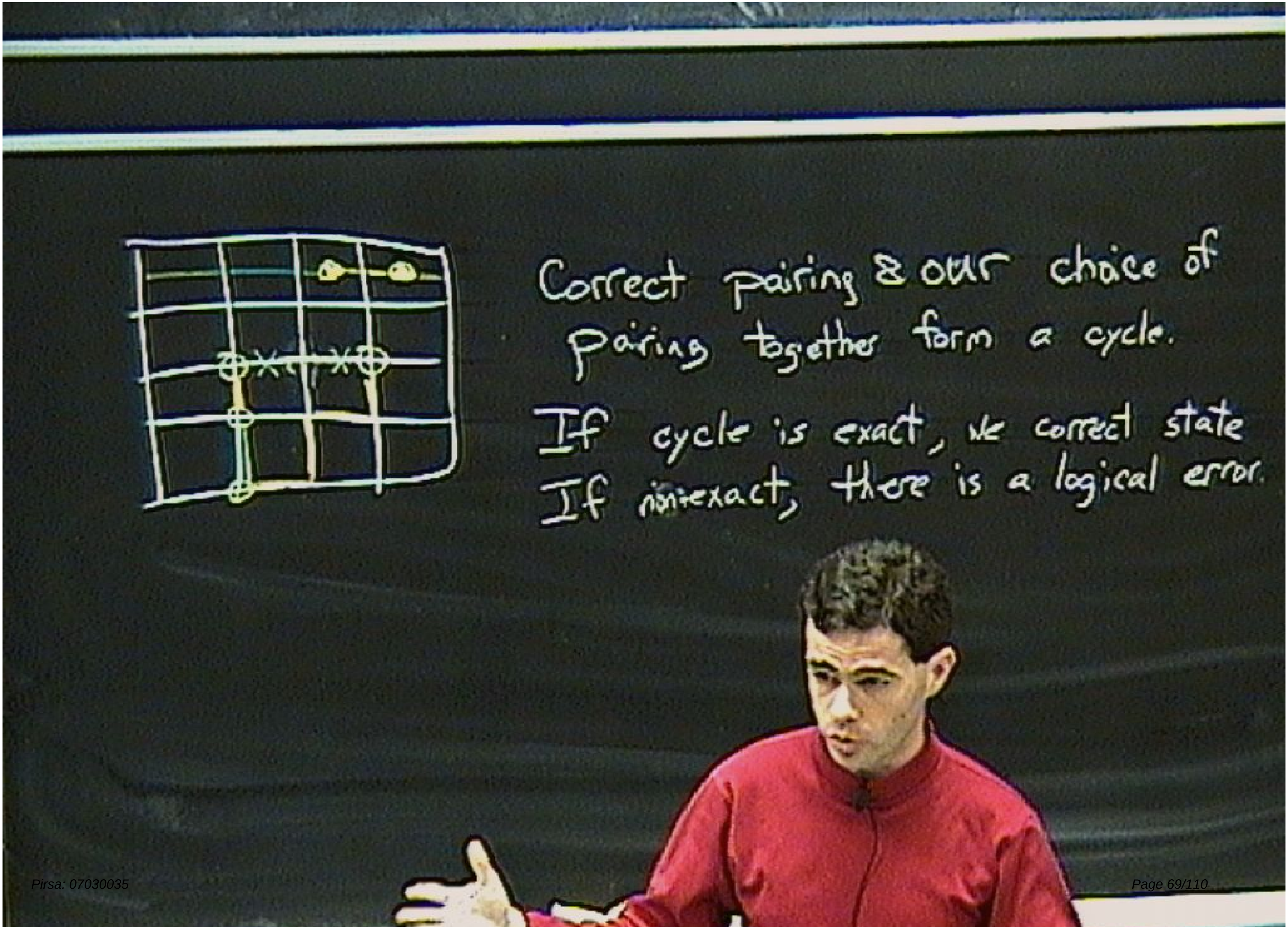


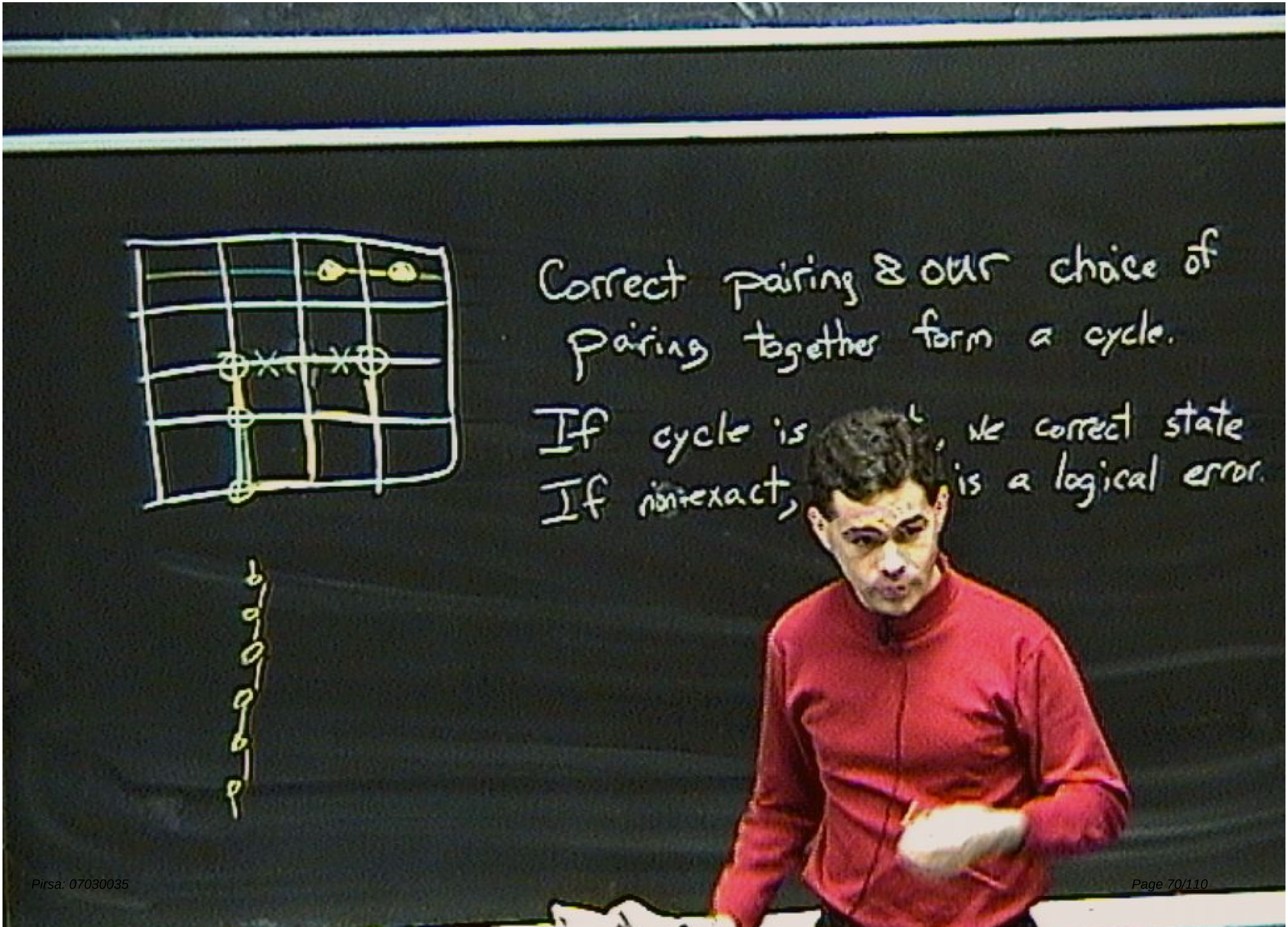


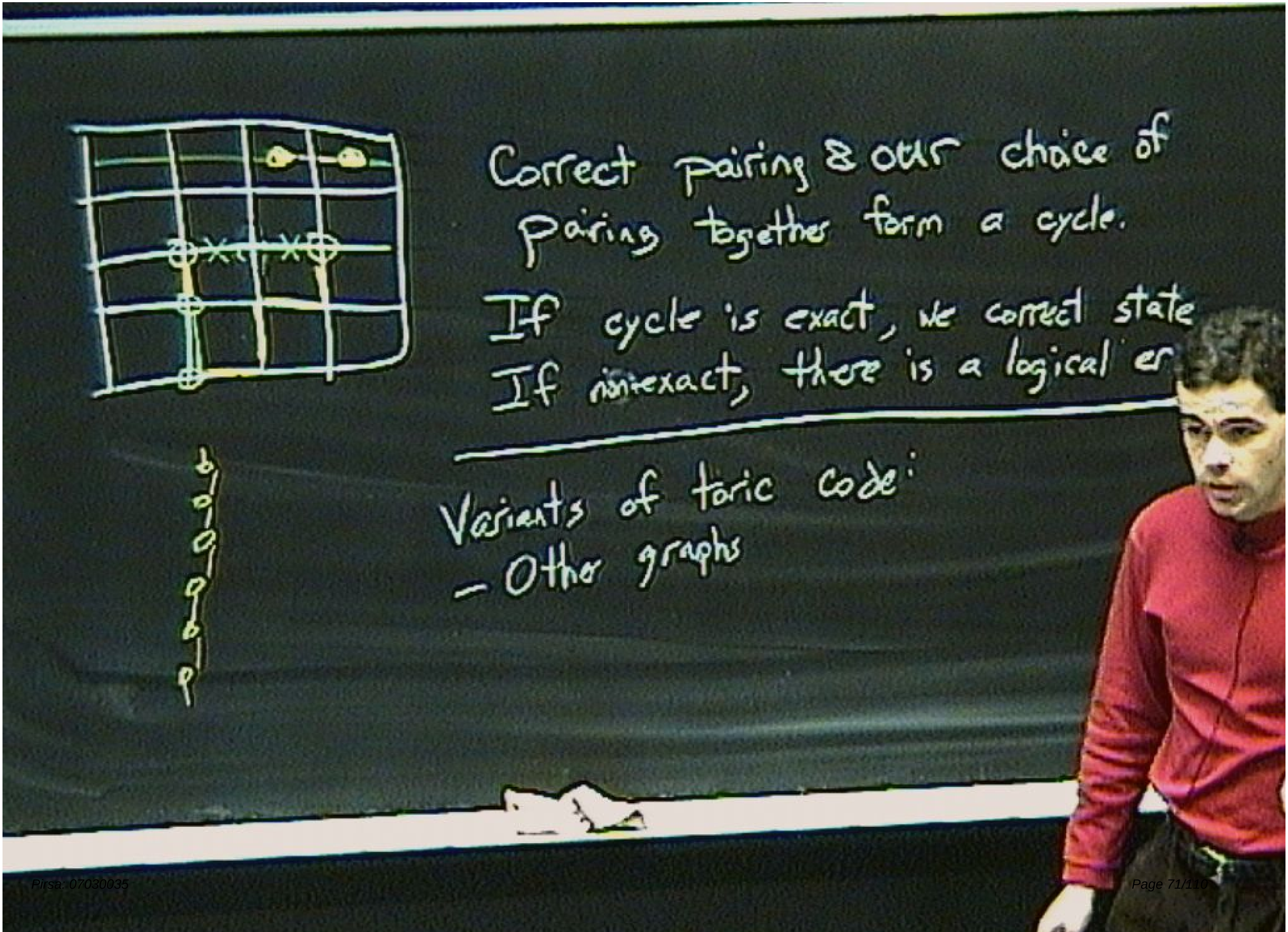


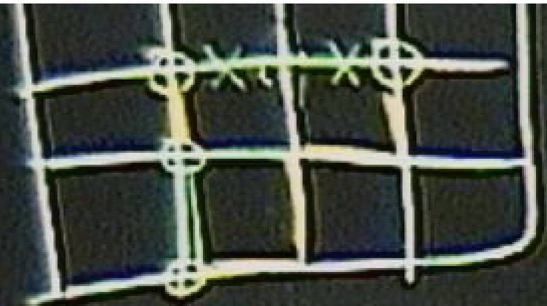

Correct pairing & our choice of
pairing together form a cycle.


Correct pairing & our choice of
pairing together form a cycle.
If cycle is exact, we correct state
If inexact, there is a logical error.


Correct pairing & our choice of pairing together form a cycle.
If cycle is exact, we correct state
If nonexact, there is a logical error.


Correct pairing & our choice of
pairing together form a cycle.
If cycle is , we correct state
If nonexact, is a logical error.


Correct pairing & our choice of
pairing together form a cycle.
If cycle is exact, we correct state
If nonexact, there is a logical er
Variants of toric code:
- Other graphs



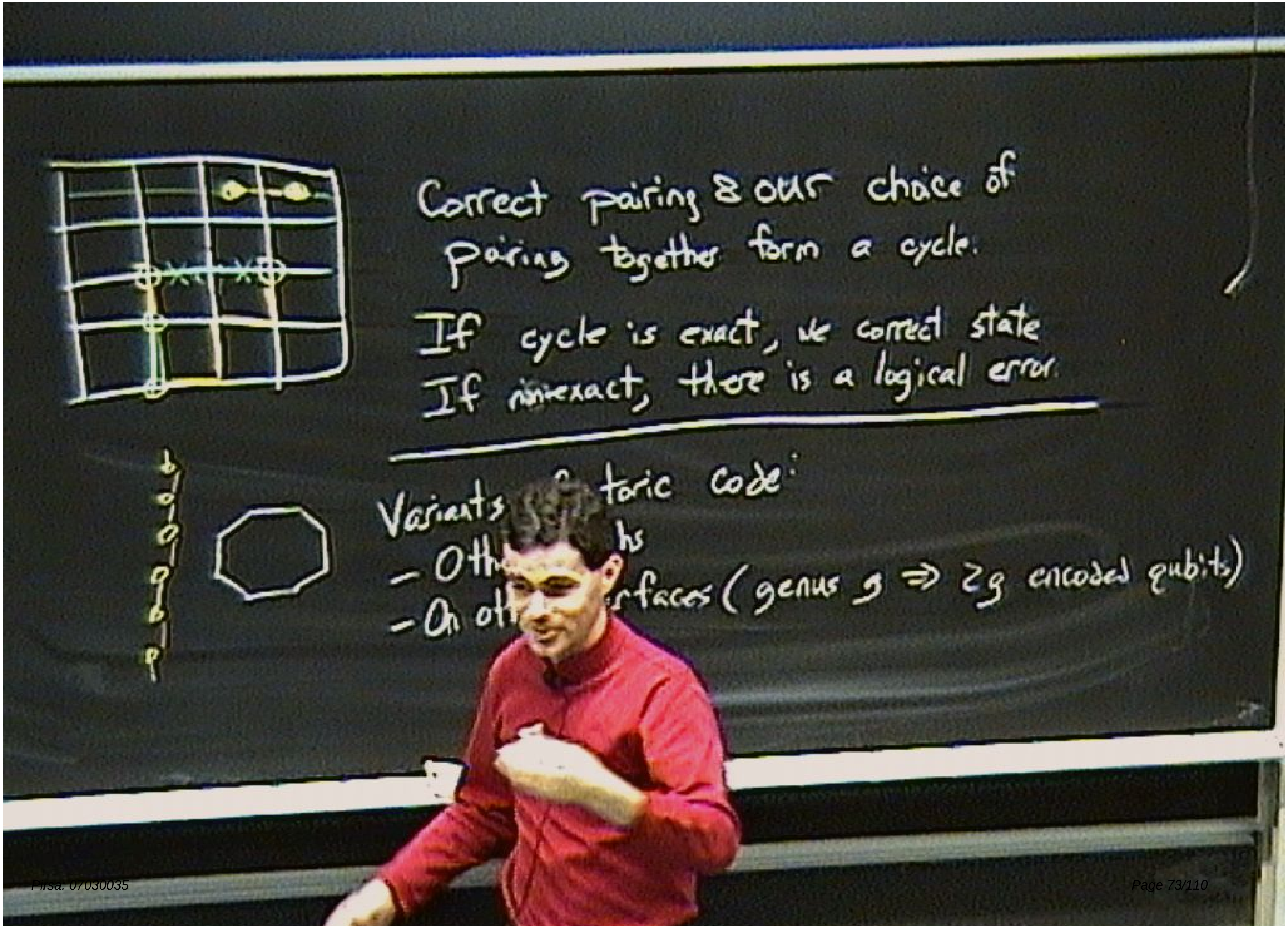
pairing together form a cycle.

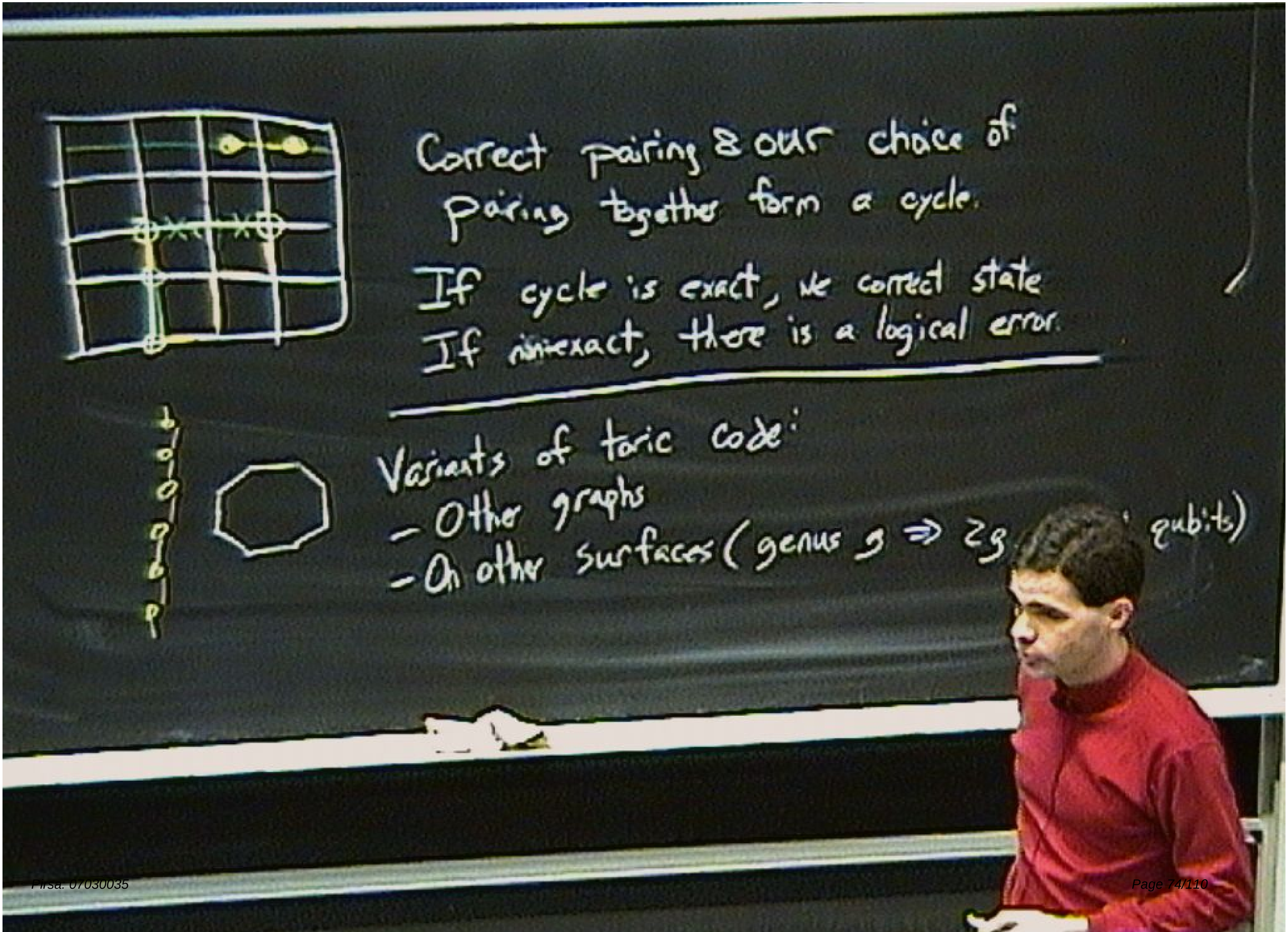
If cycle is exact, we correct state

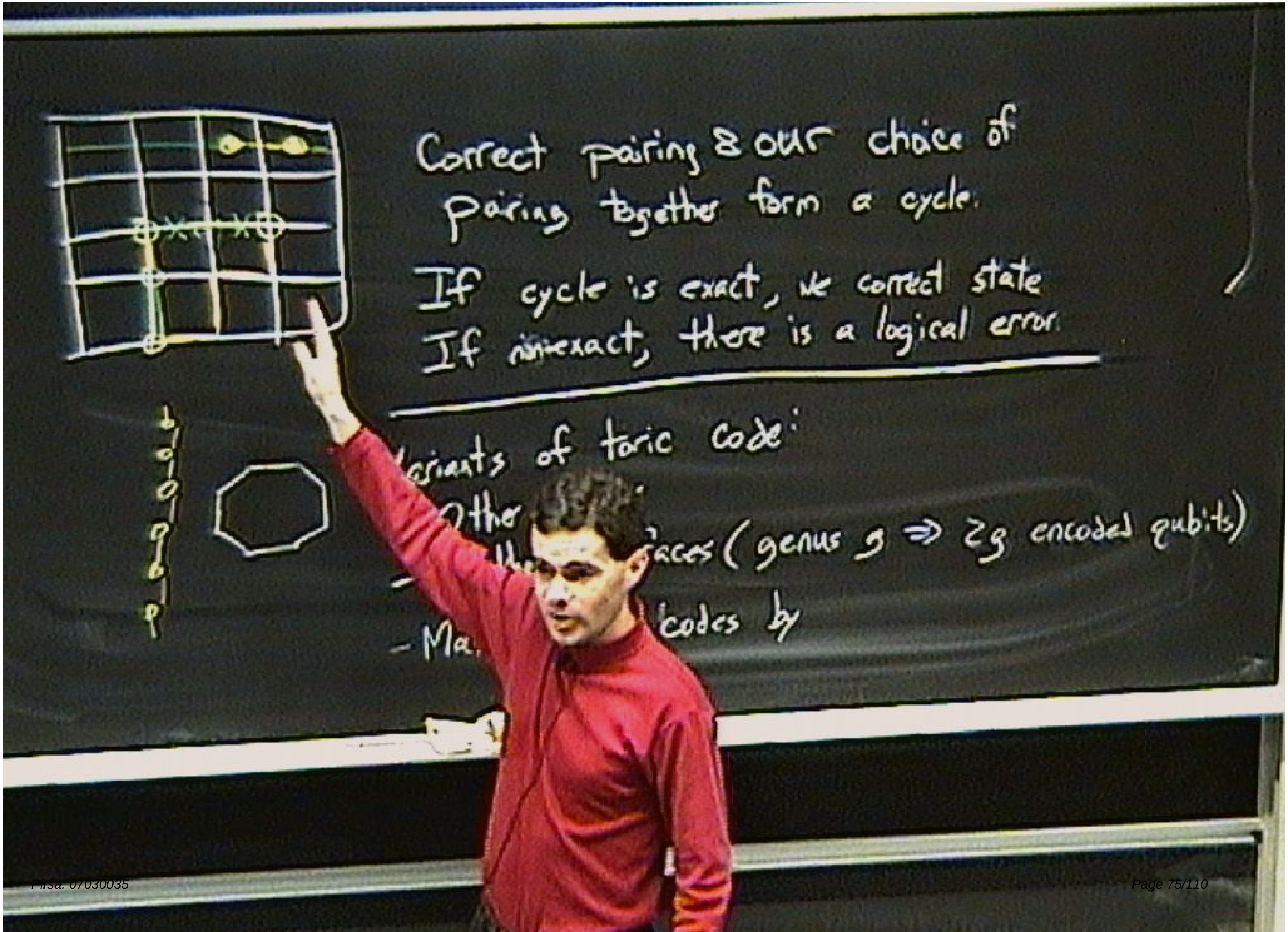
If nonexact, there is a logical error.

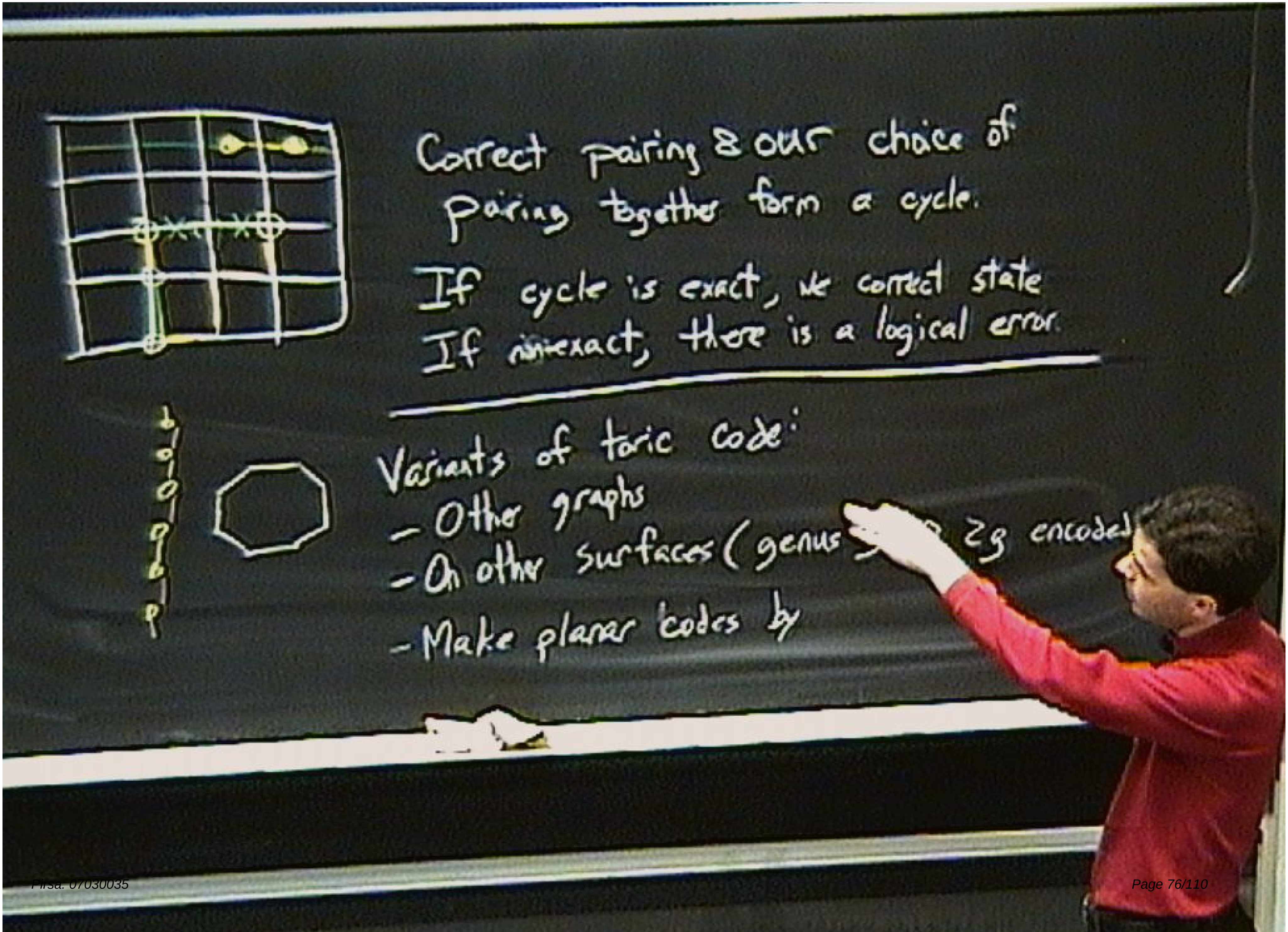
Variants of toric code:

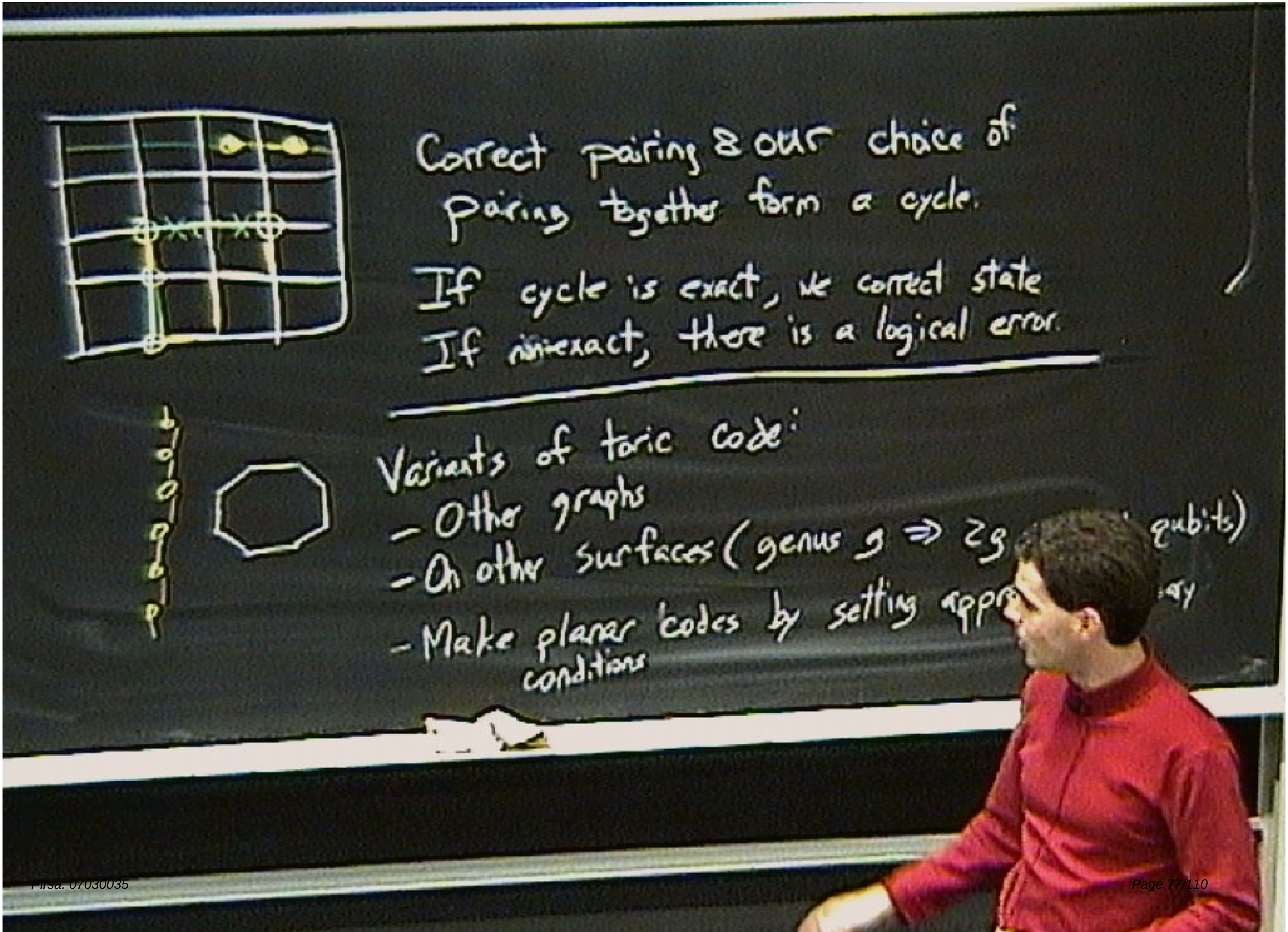
- Other graphs
- On other surfaces (genus g $\Rightarrow$

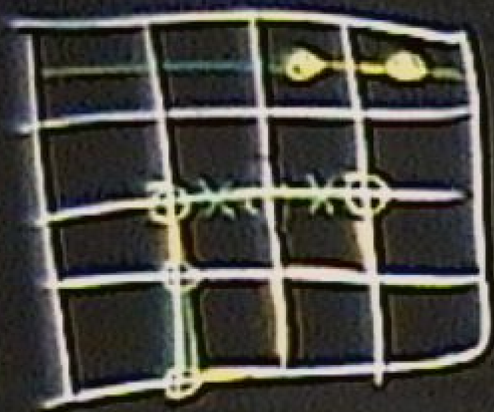

Correct pairing & our choice of pairing together form a cycle.
If cycle is exact, we correct state
If nonexact, there is a logical error.
Variants
toric code:
- Oth
hs
- On ot
rfaces (genus g ⇒ 2g encoded qubits)


Correct pairing & our choice of
pairing together form a cycle.
If cycle is exact, we correct state
If nonexact, there is a logical error.
Variants of toric code:
- Other graphs
- On other surfaces (genus g ⇒ 2g
qubits)


Correct pairing & our choice of
pairing together form a cycle.
If cycle is exact, we correct state
If nonexact, there is a logical error.
ariants of toric code:
Other
aces (genus g ⇒ 2g encoded qubits)
codes by
- Ma


Correct pairing & our choice of
pairing together form a cycle.
If cycle is exact, we correct state
If nonexact, there is a logical error.
Variants of toric code:
- Other graphs
- On other surfaces (genus
2g encoded
- Make planar codes by


Correct pairing & our choice of
pairing together form a cycle.
If cycle is exact, we correct state
If nonexact, there is a logical error.
Variants of toric code:
- Other graphs
- On other surfaces (genus g ⇒ 2g
qubits)
- Make planar codes by setting
conditions



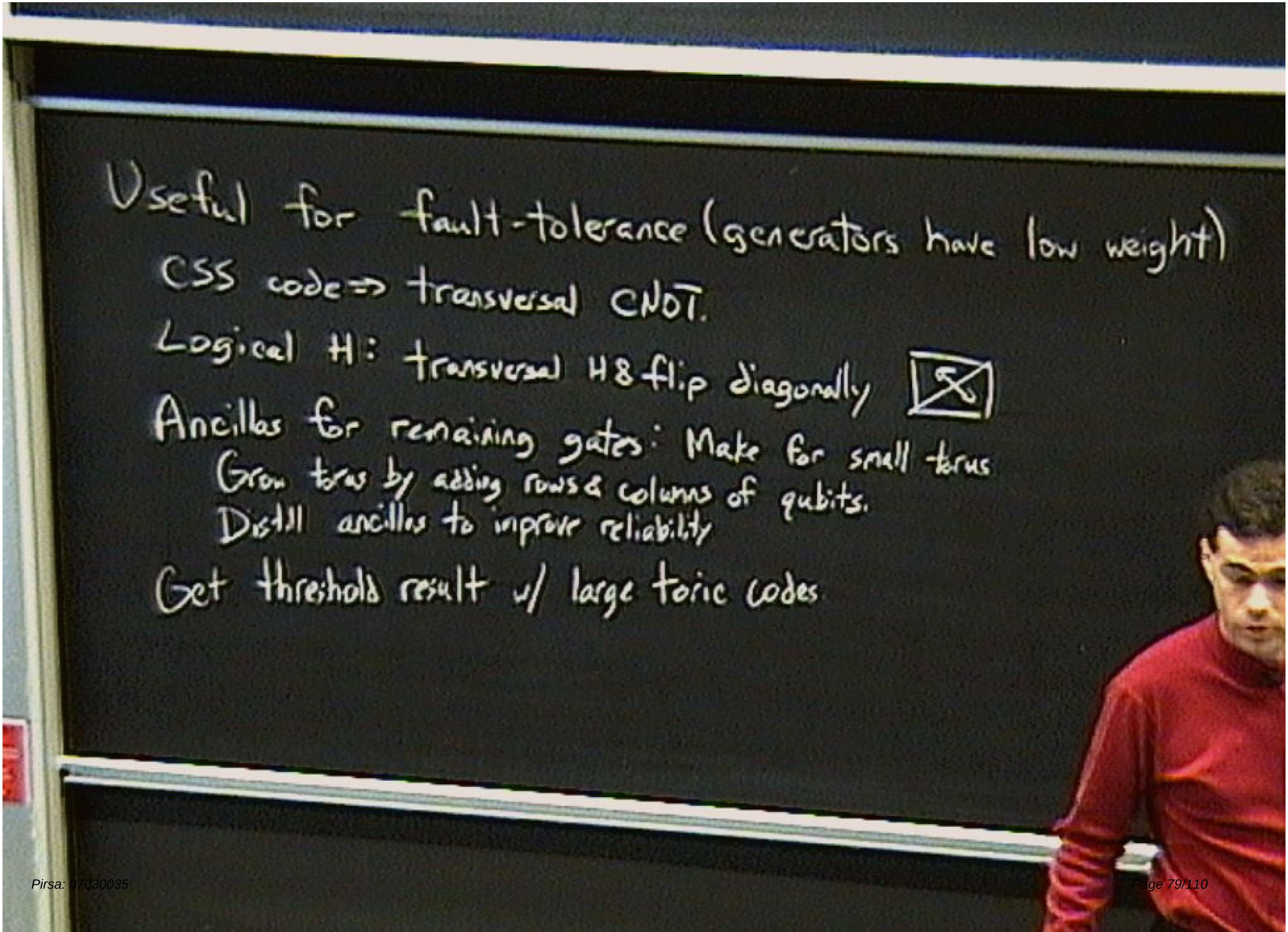
Correct pairing & our choice of pairing together form a cycle.

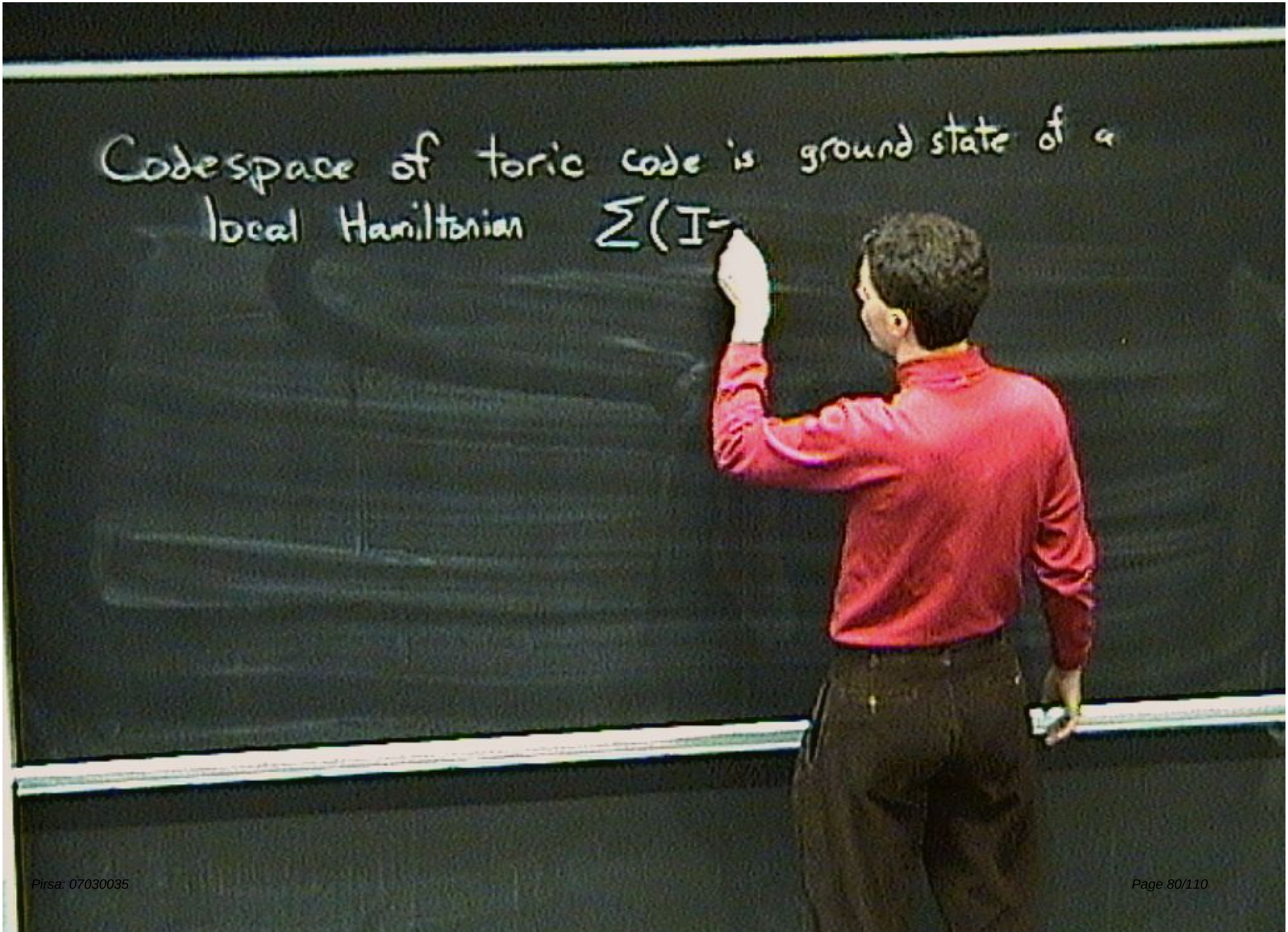
If cycle is exact, we correct state

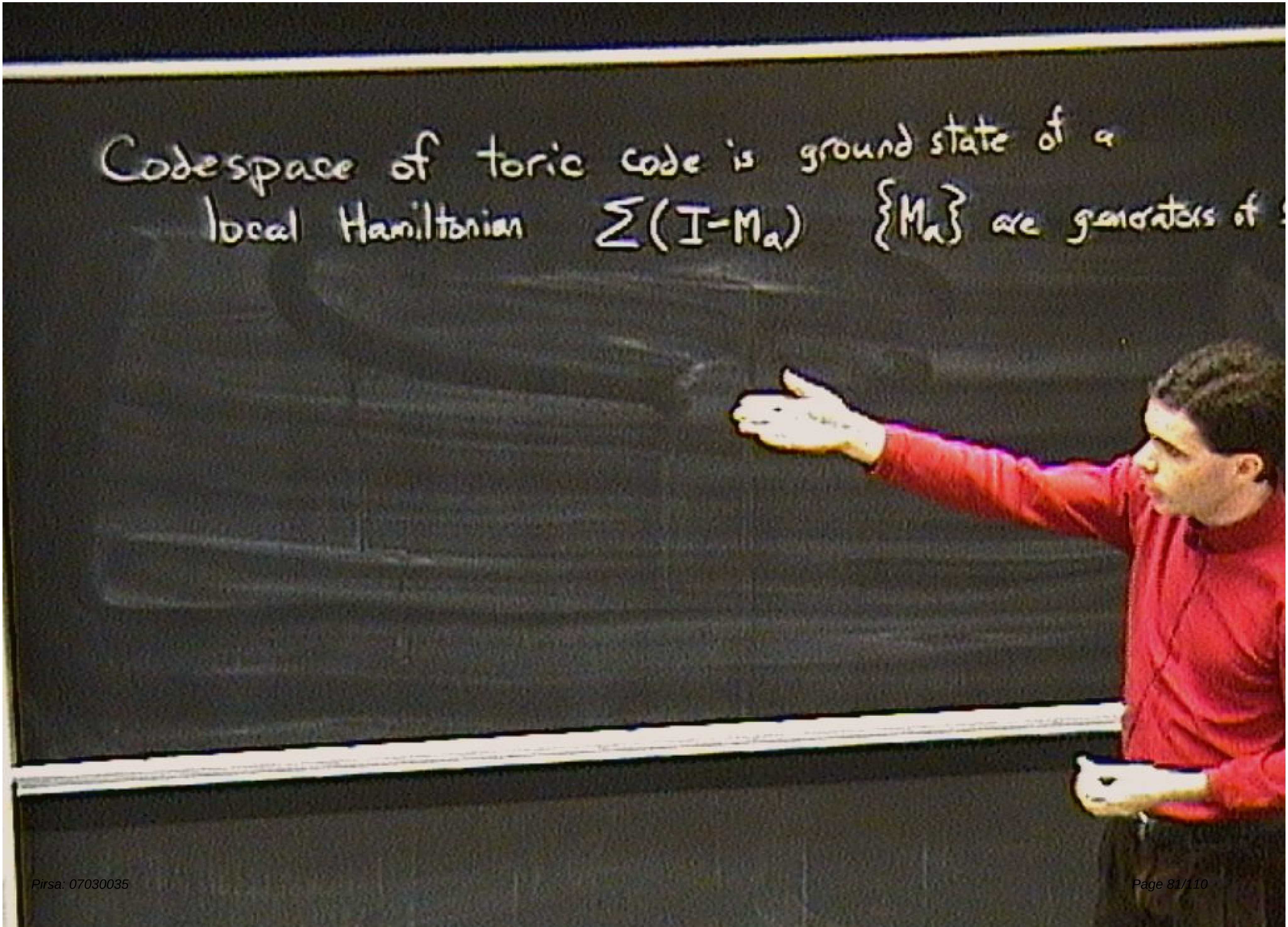
If nonexact, there is a logical error.

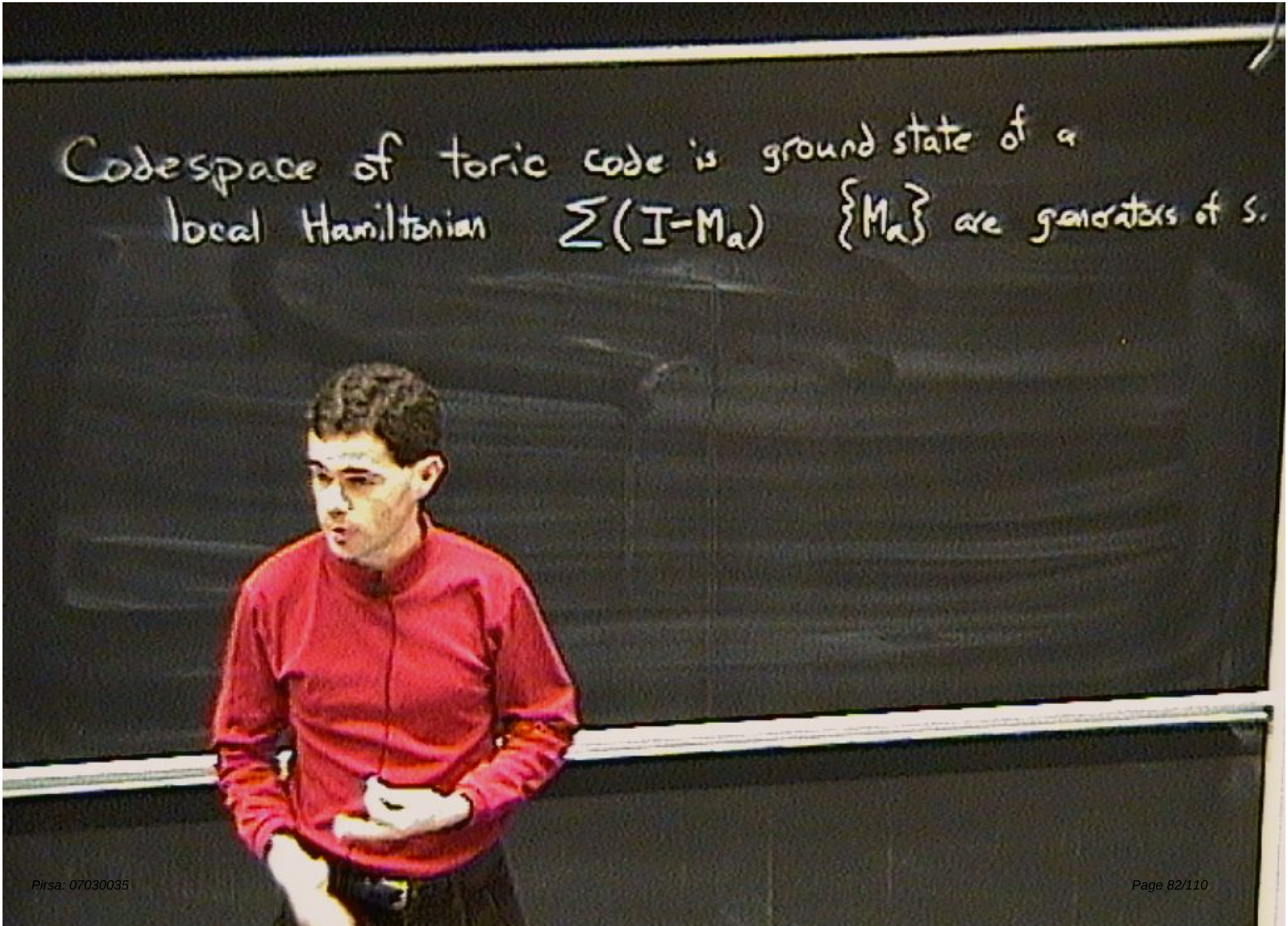
Variants of toric code:

- Other graphs
- On other surfaces (genus $g \Rightarrow 2g$ encoded qubits)
- Make planar codes by setting appropriate boundary conditions


Useful for fault-tolerance (generators have low weight)
CSS code ⇒ transversal CNOT.
Logical H: transversal H & flip diagonally
Ancillas for remaining gates: Make for small torus
Grow torus by adding rows & columns of qubits.
Distill ancillas to improve reliability
Get threshold result w/ large toric codes

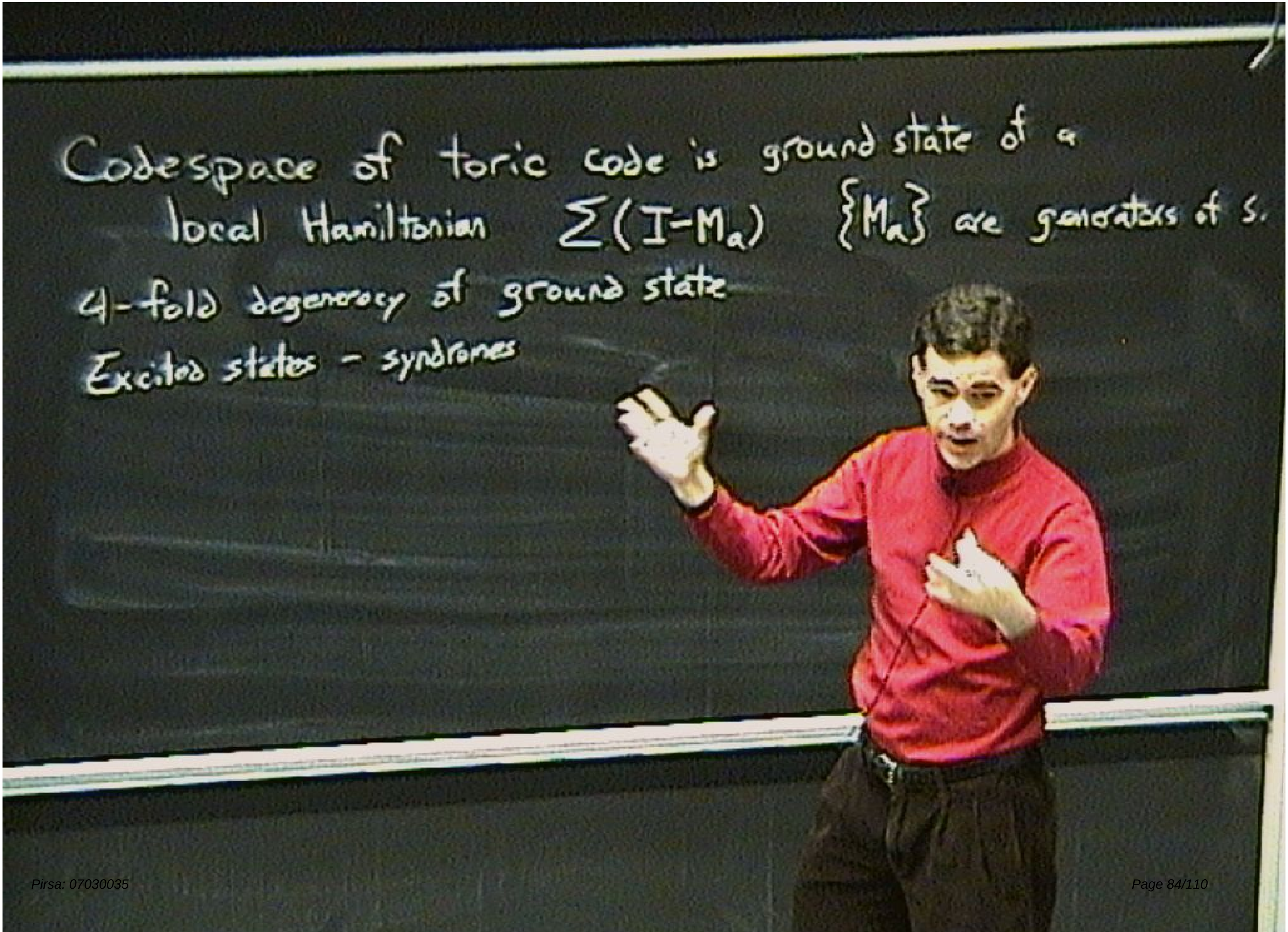

Codespace of toric code is ground state of a
local Hamiltonian $\Sigma(I-$

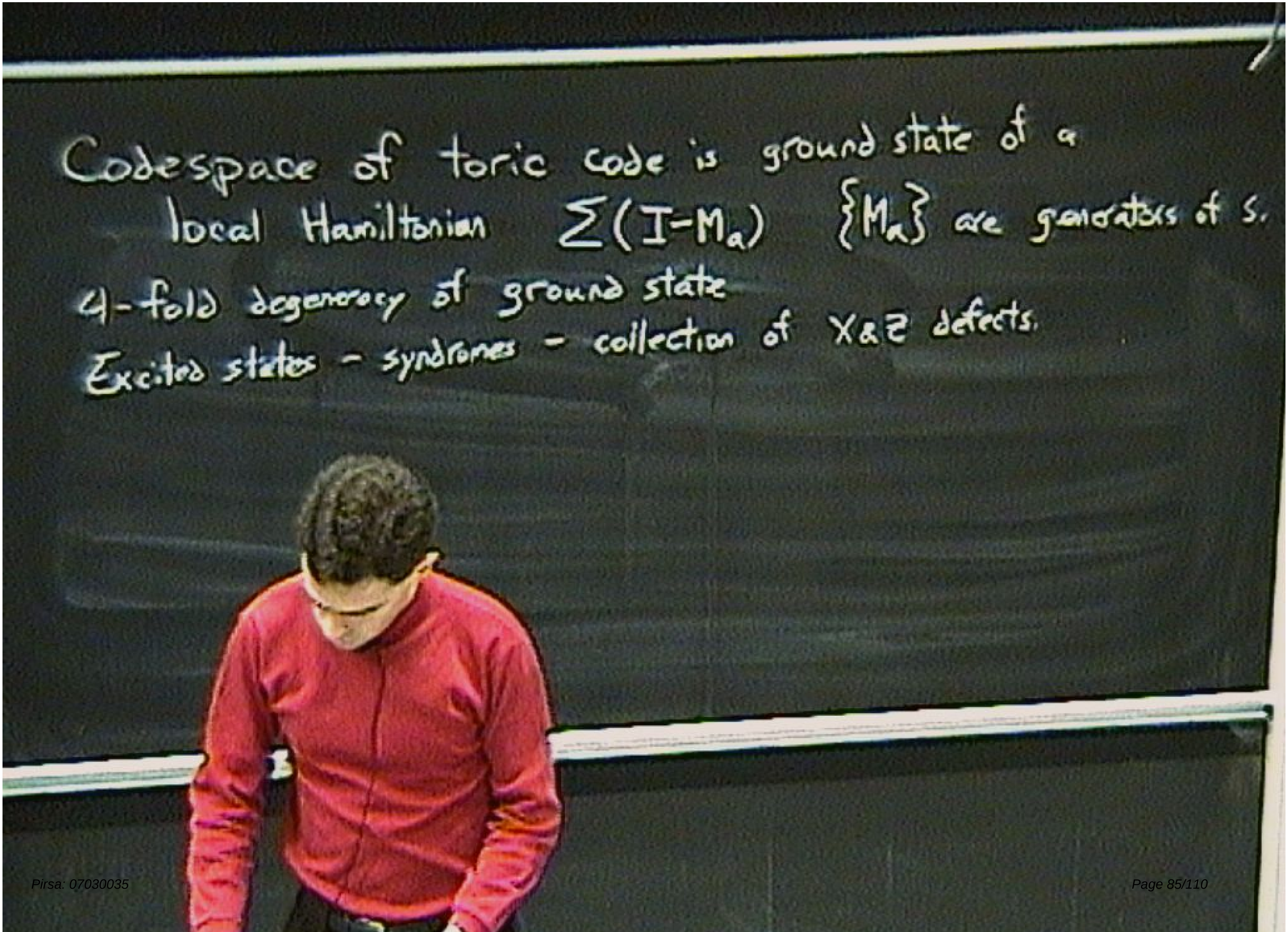

Codespace of toric code is ground state of a
local Hamiltonian $\sum(I-M_a)$ $\{M_a\}$ are generators of

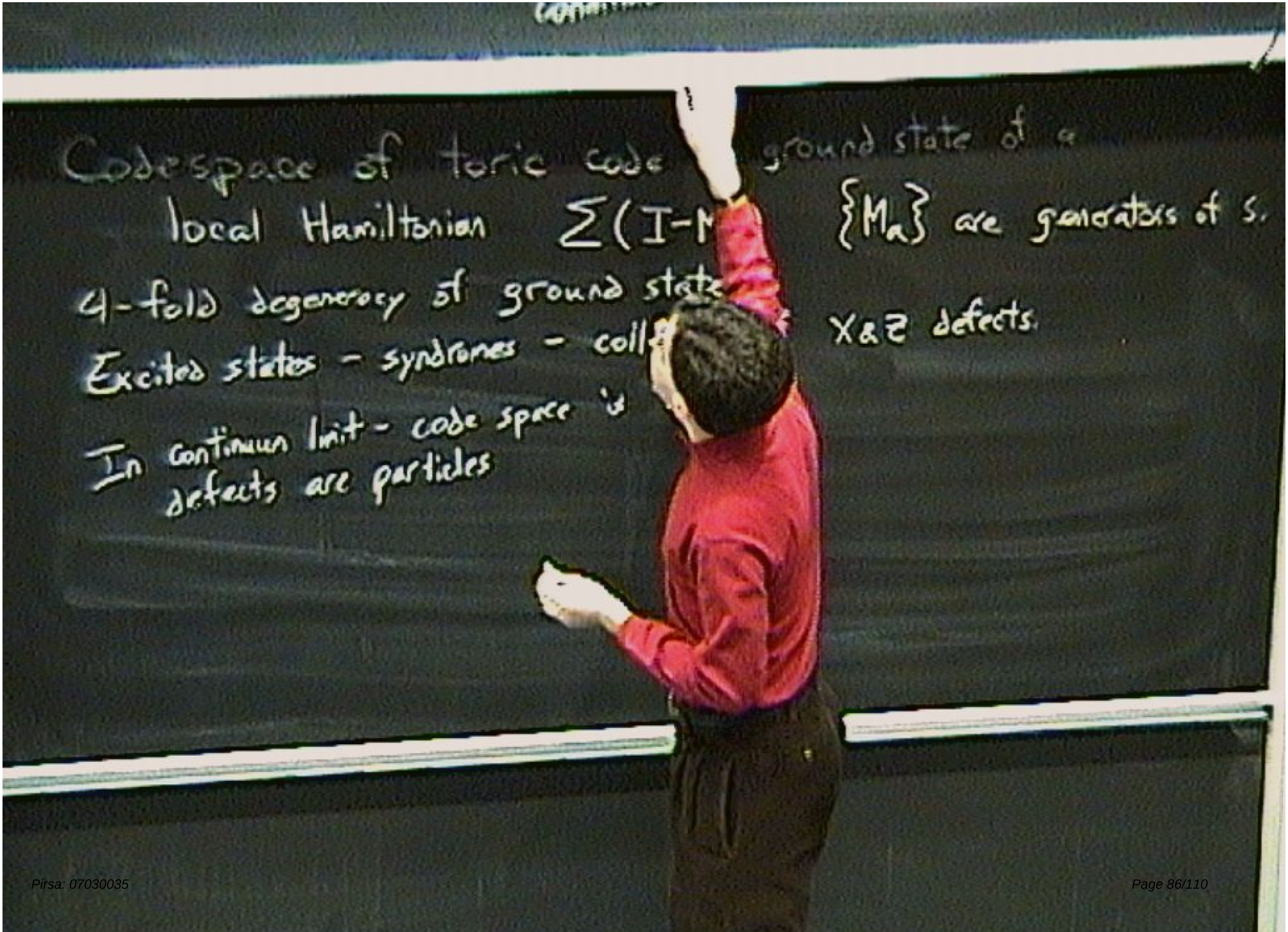

Codespace of toric code is ground state of a
local Hamiltonian $\sum (I - M_a)$ $\{M_a\}$ are generators of S.

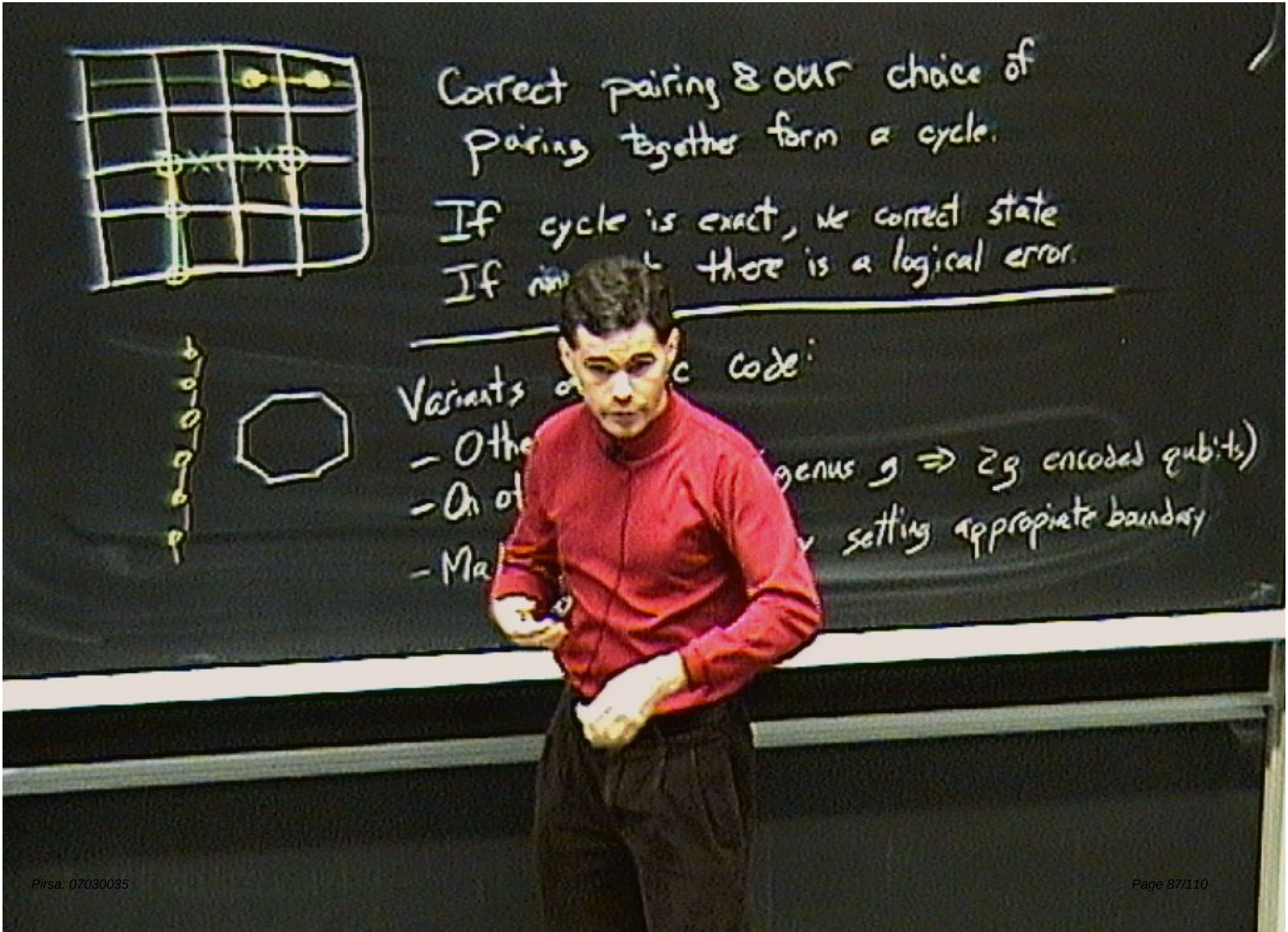
Codespace of toric code is ground state of a local Hamiltonian $\sum (I - M_a)$ $\{M_a\}$ are generators of S.

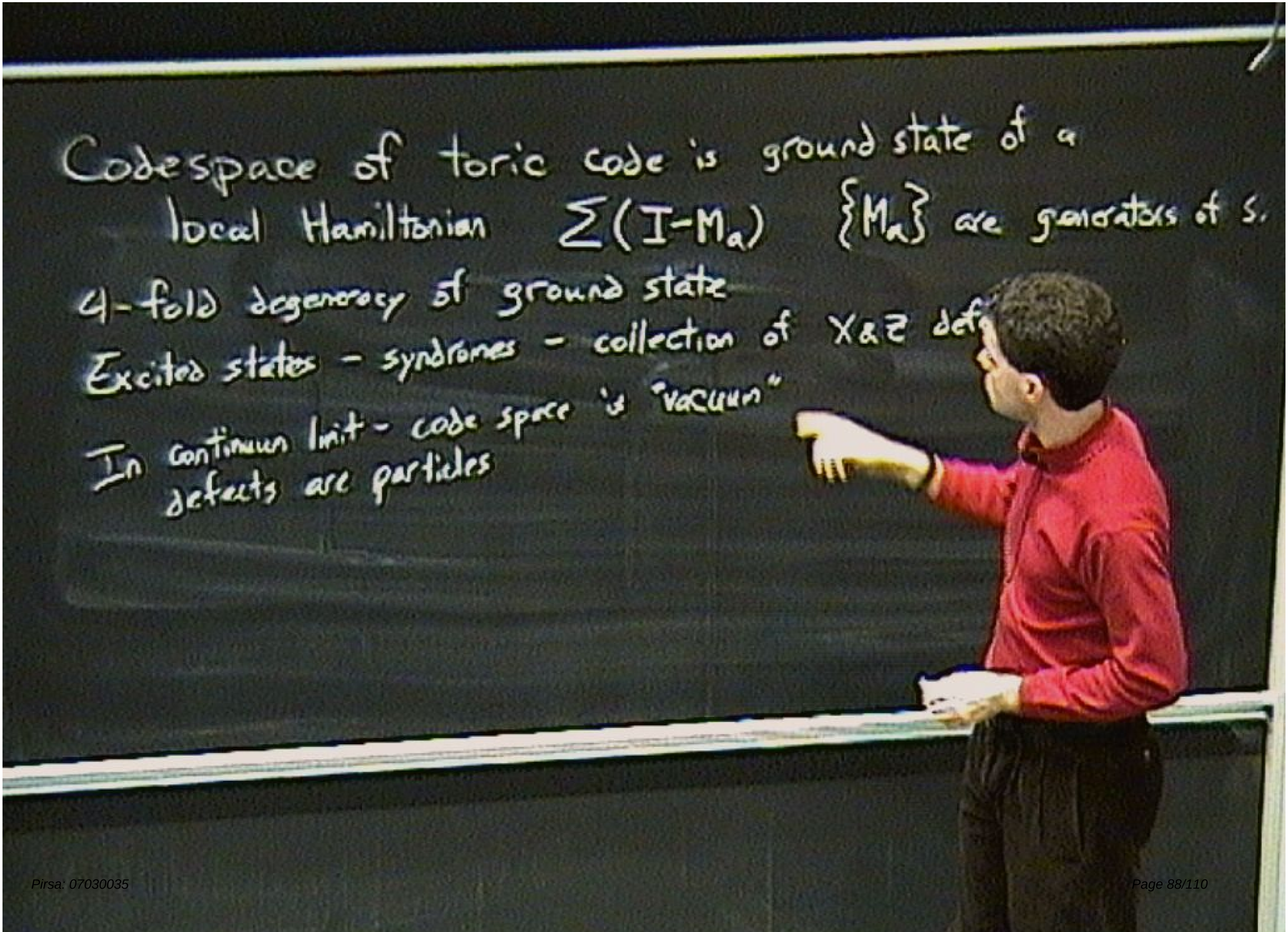
4-fold degeneracy of ground state

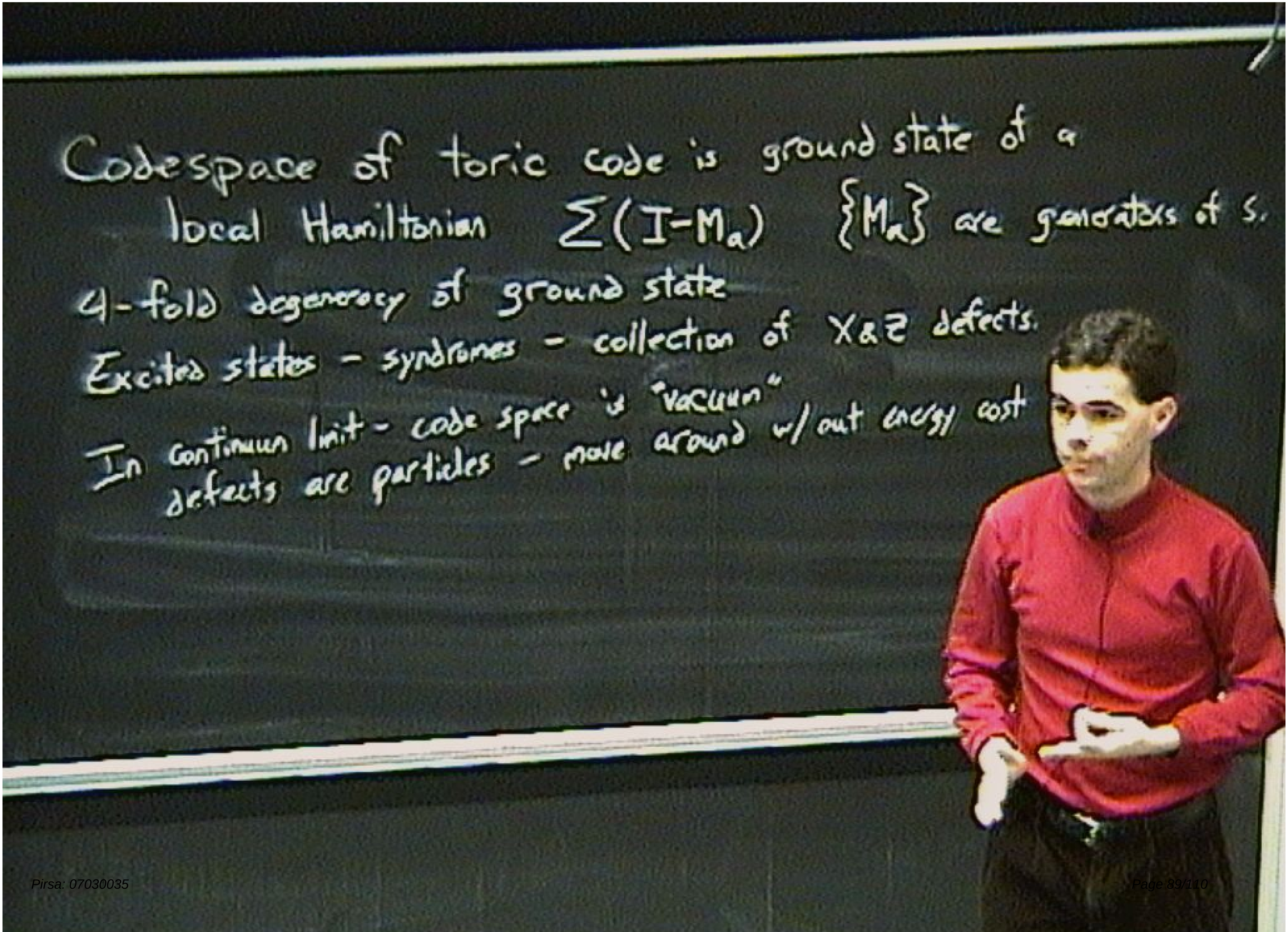

Codespace of toric code is ground state of a
local Hamiltonian $\sum(I-M_a)$ $\{M_a\}$ are generators of S.
4-fold degeneracy of ground state
Excited states - syndromes

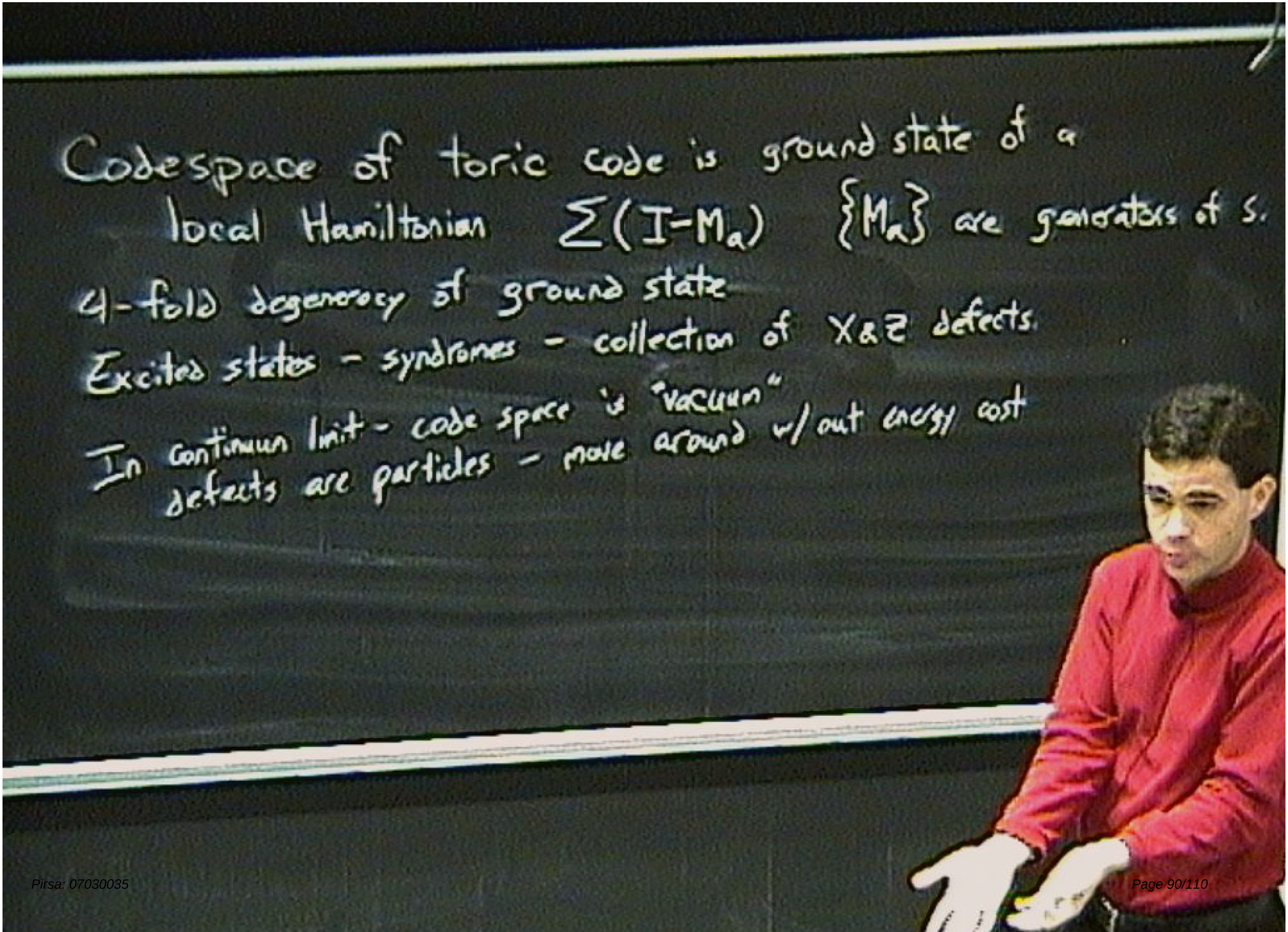

Codespace of toric code is ground state of a
local Hamiltonian $\sum(I-M_a)$ $\{M_a\}$ are generators of S.
4-fold degeneracy of ground state
Excited states - syndromes - collection of X&Z defects.

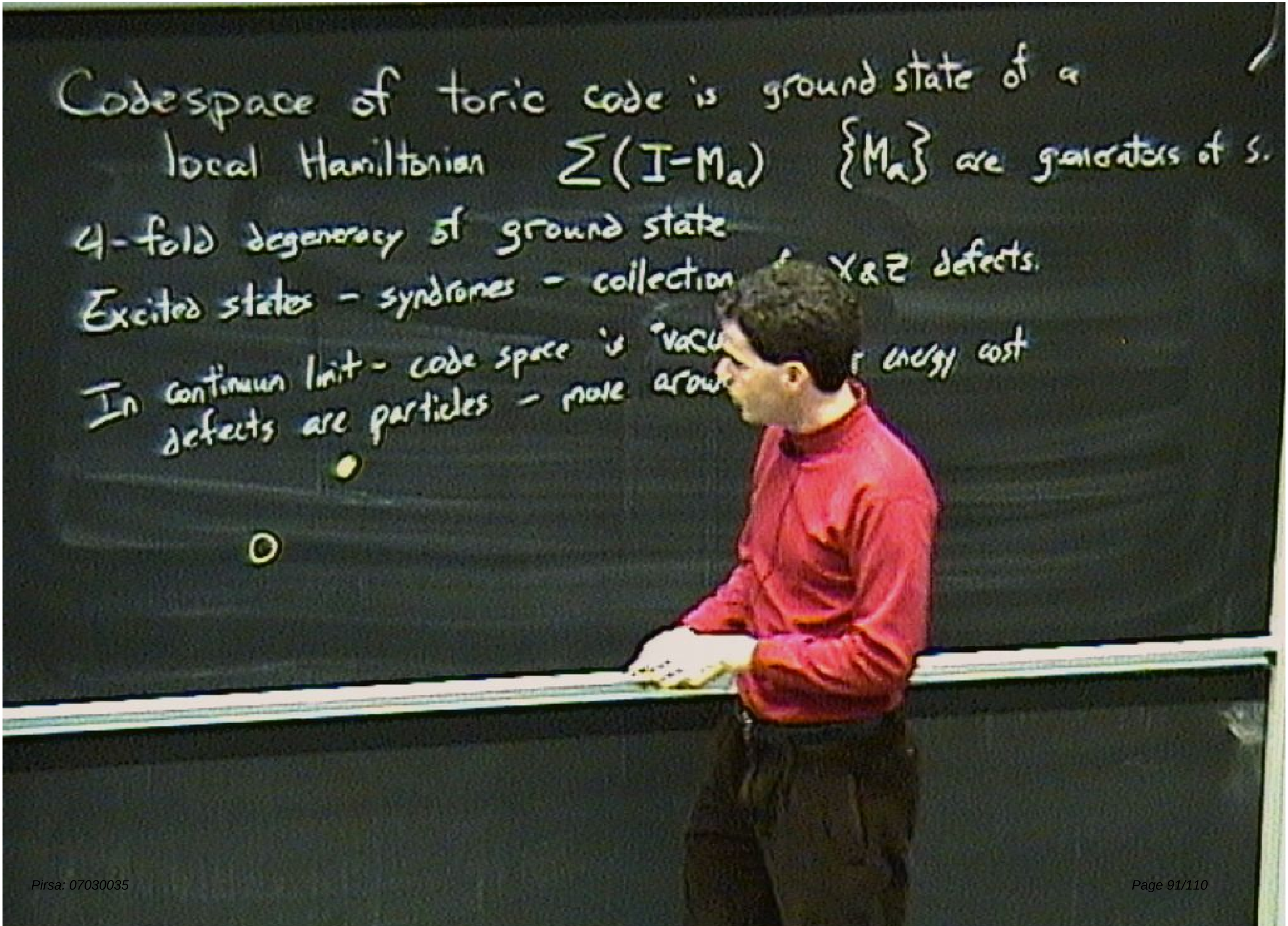

Codespace of toric code ground state of a
local Hamiltonian Σ(I-M {Ma} are generators of S.
4-fold degeneracy of ground state
Excited states - syndromes - coll X&Z defects.
In continuum limit - code space is
defects are particles

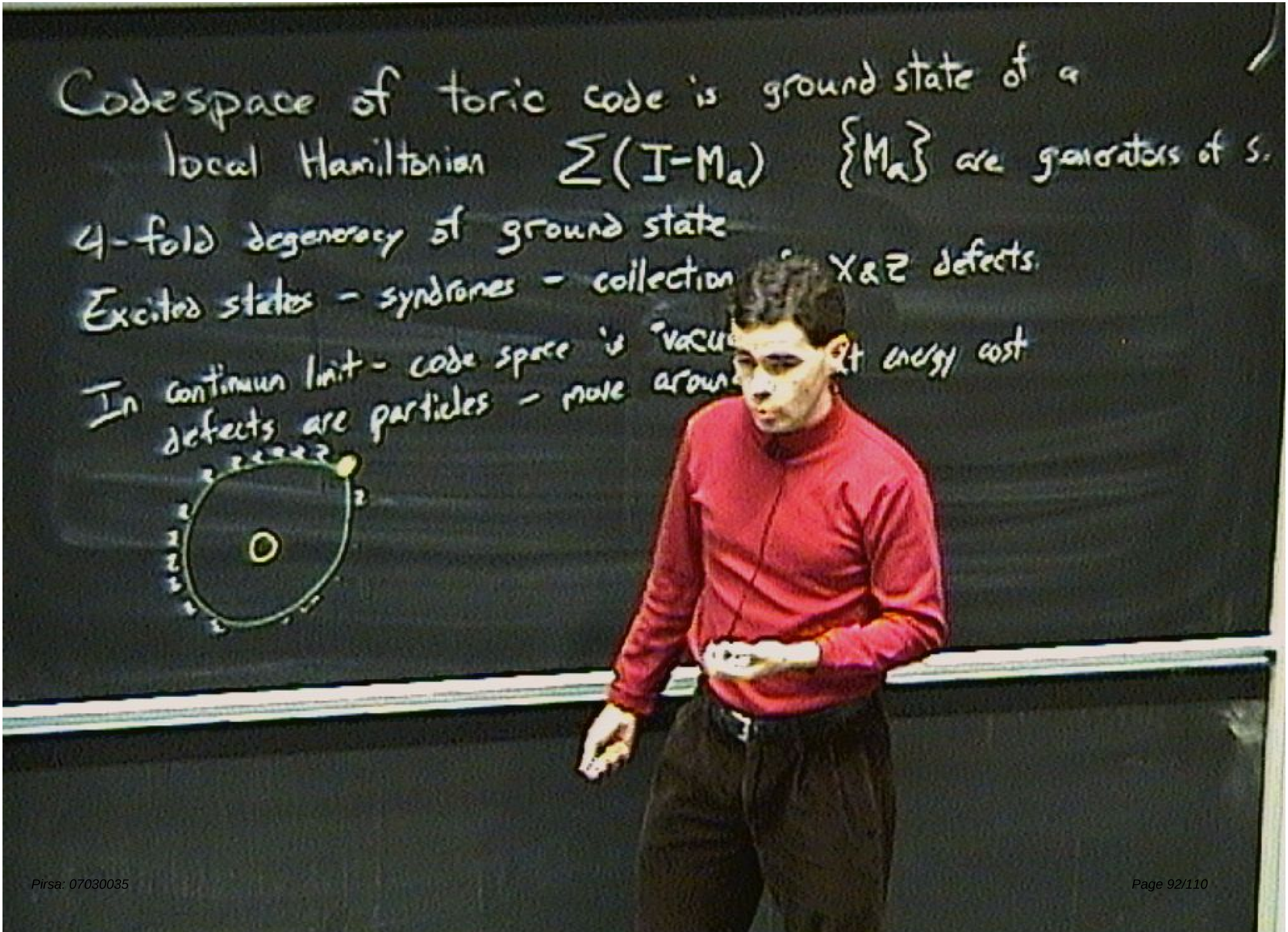

Correct pairing & our choice of
pairing together form a cycle.
If cycle is exact, we correct state
If
there is a logical error.
Variants
code:
genus g ⇒ 2g encoded qubits)
setting appropriate boundary

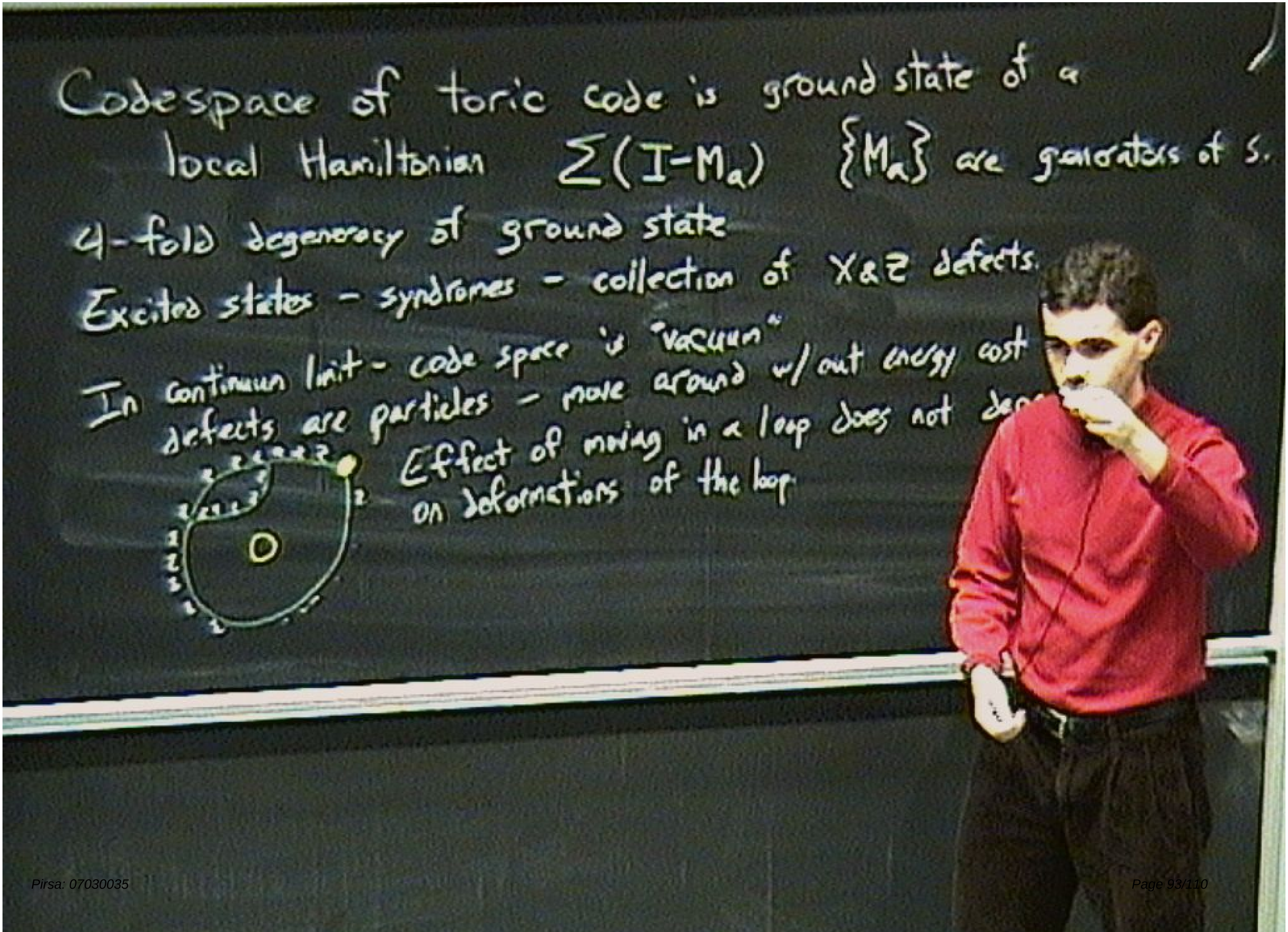

Codespace of toric code is ground state of a
local Hamiltonian $\sum(I-M_a)$ $\{M_a\}$ are generators of S.
4-fold degeneracy of ground state
Excited states - syndromes - collection of X&Z def
In continuum limit - code space is "vacuum"
defects are particles

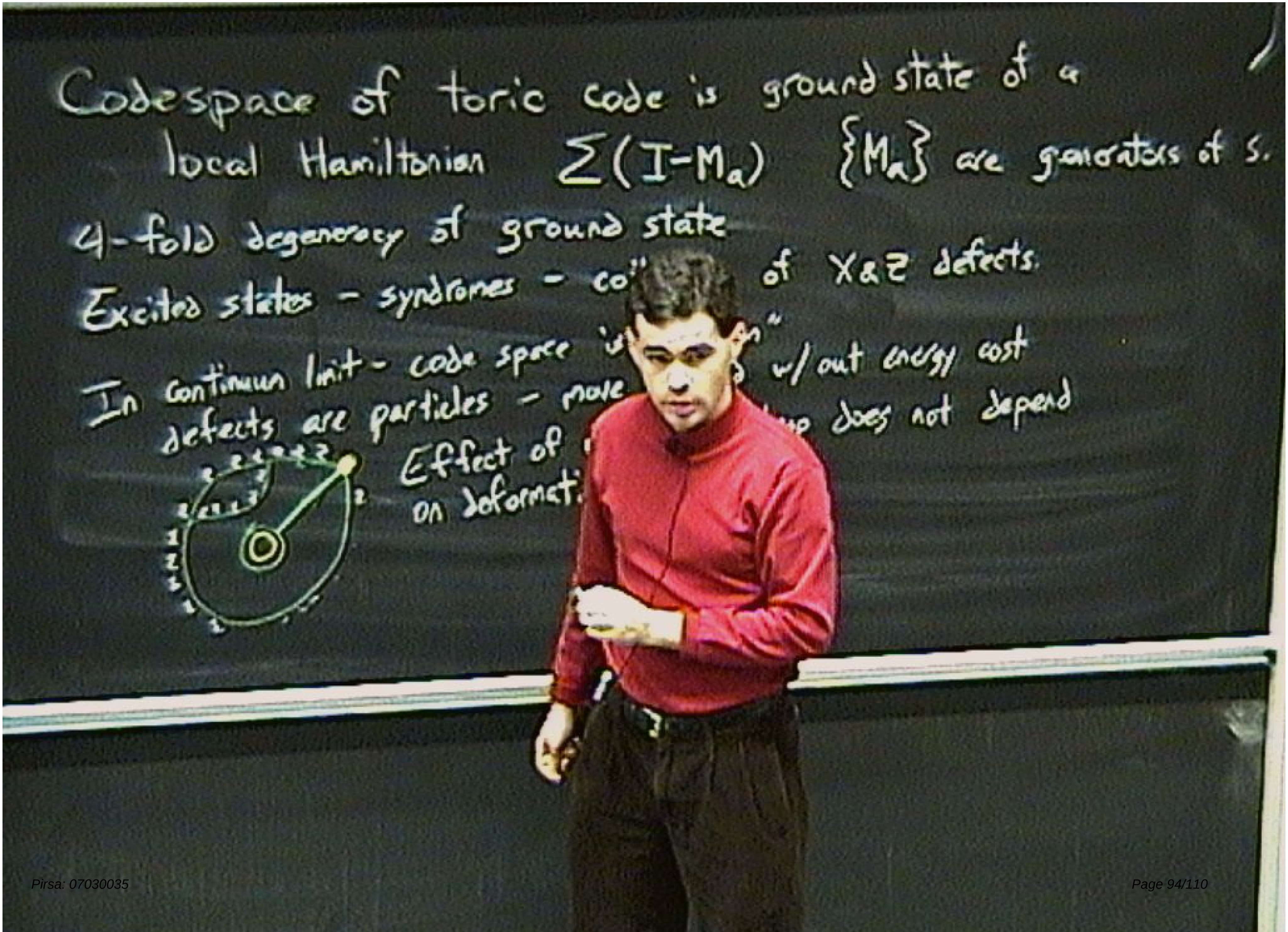

Codespace of toric code is ground state of a
local Hamiltonian $\sum(I-M_a)$ $\{M_a\}$ are generators of S.
4-fold degeneracy of ground state
Excited states - syndromes - collection of X&Z defects.
In continuum limit - code space is "vacuum"
defects are particles - move around w/out energy cost


Codespace of toric code is ground state of a
local Hamiltonian $\sum(I-M_a)$ $\{M_a\}$ are generators of S.
4-fold degeneracy of ground state
Excited states - syndromes - collection of X&Z defects.
In continuum limit - code space is "vacuum"
defects are particles - move around w/out energy cost

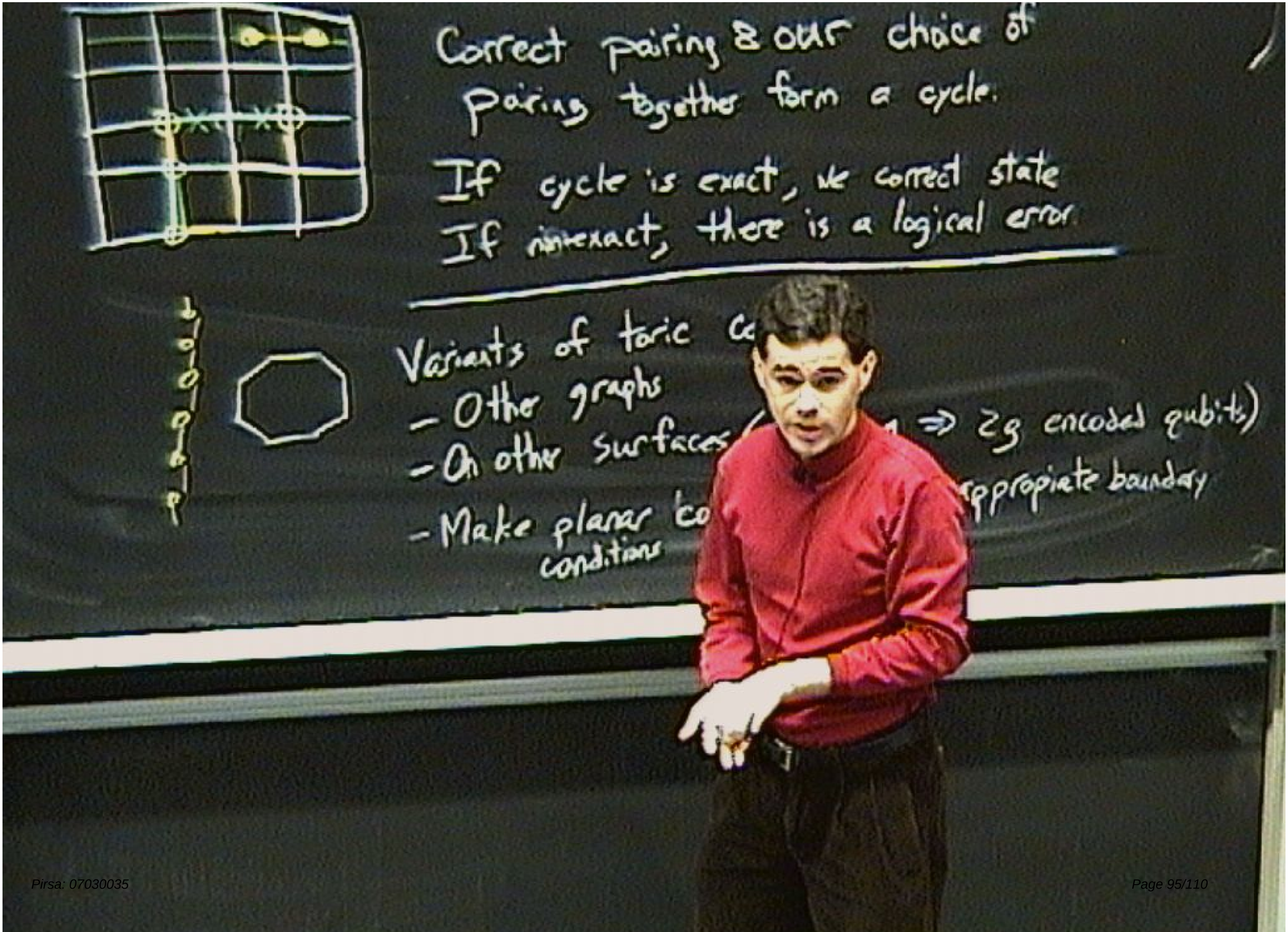

Codespace of toric code is ground state of a
local Hamiltonian Σ(I-M_a) {M_a} are generators of S.
4-fold degeneracy of ground state
Excited states - syndromes - collection
X&Z defects.
In continuum limit - code space is "vacu
energy cost
defects are particles - move arou


Codespace of toric code is ground state of a
local Hamiltonian $\sum(I-M_a)$ $\{M_a\}$ are generators of S.
4-fold degeneracy of ground state
Excited states - syndromes - collection X&Z defects.
In continuum limit - code space is "vacu t energy cost
defects are particles - move aroun

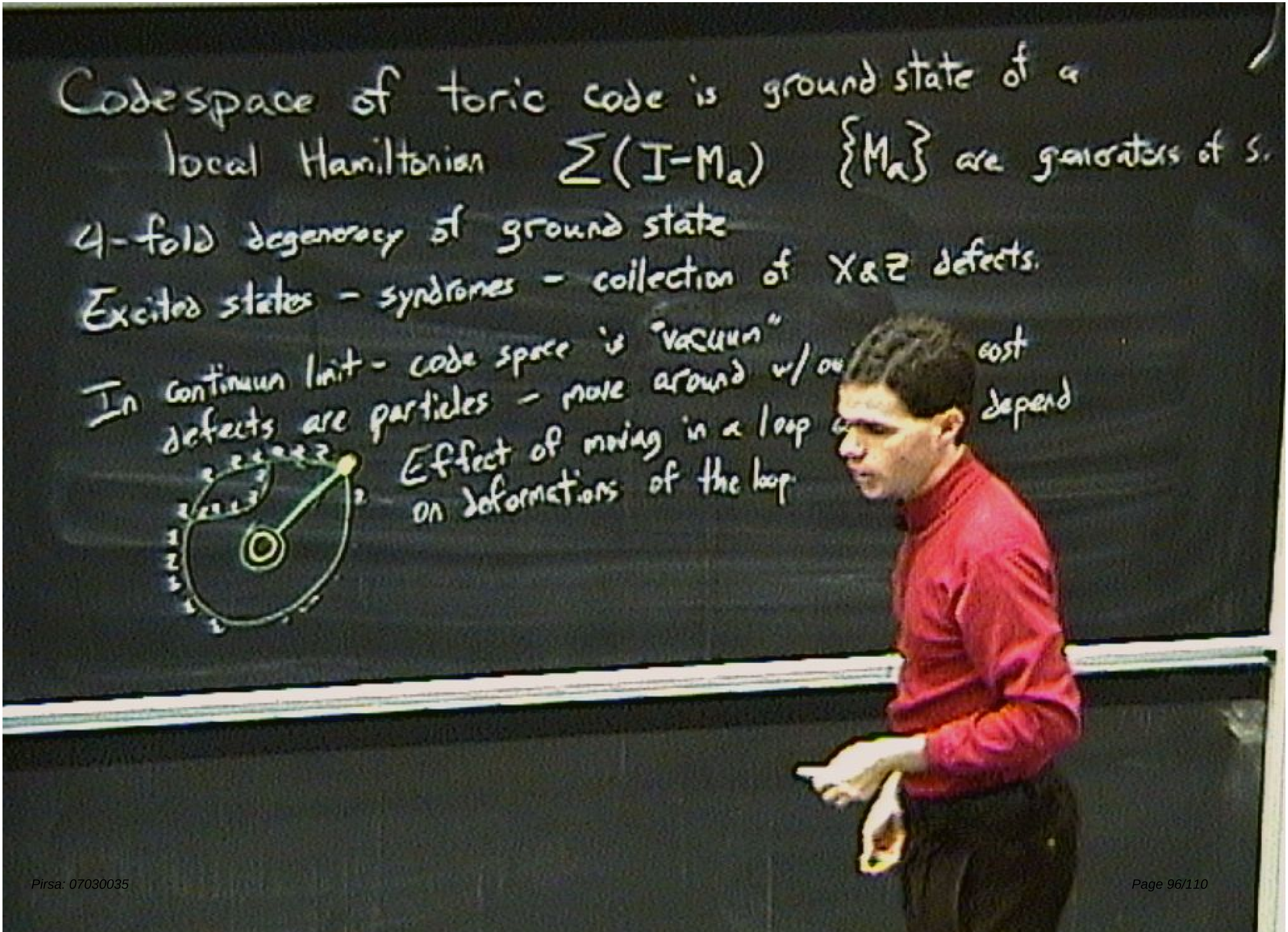

Codespace of toric code is ground state of a
local Hamiltonian $\sum(I - M_a)$ $\{M_a\}$ are generators of S.
4-fold degeneracy of ground state
Excited states - syndromes - collection of X&Z defects.
In continuum limit - code space is "vacuum"
defects are particles - move around w/out energy cost
Effect of moving in a loop does not dep
on deformations of the loop

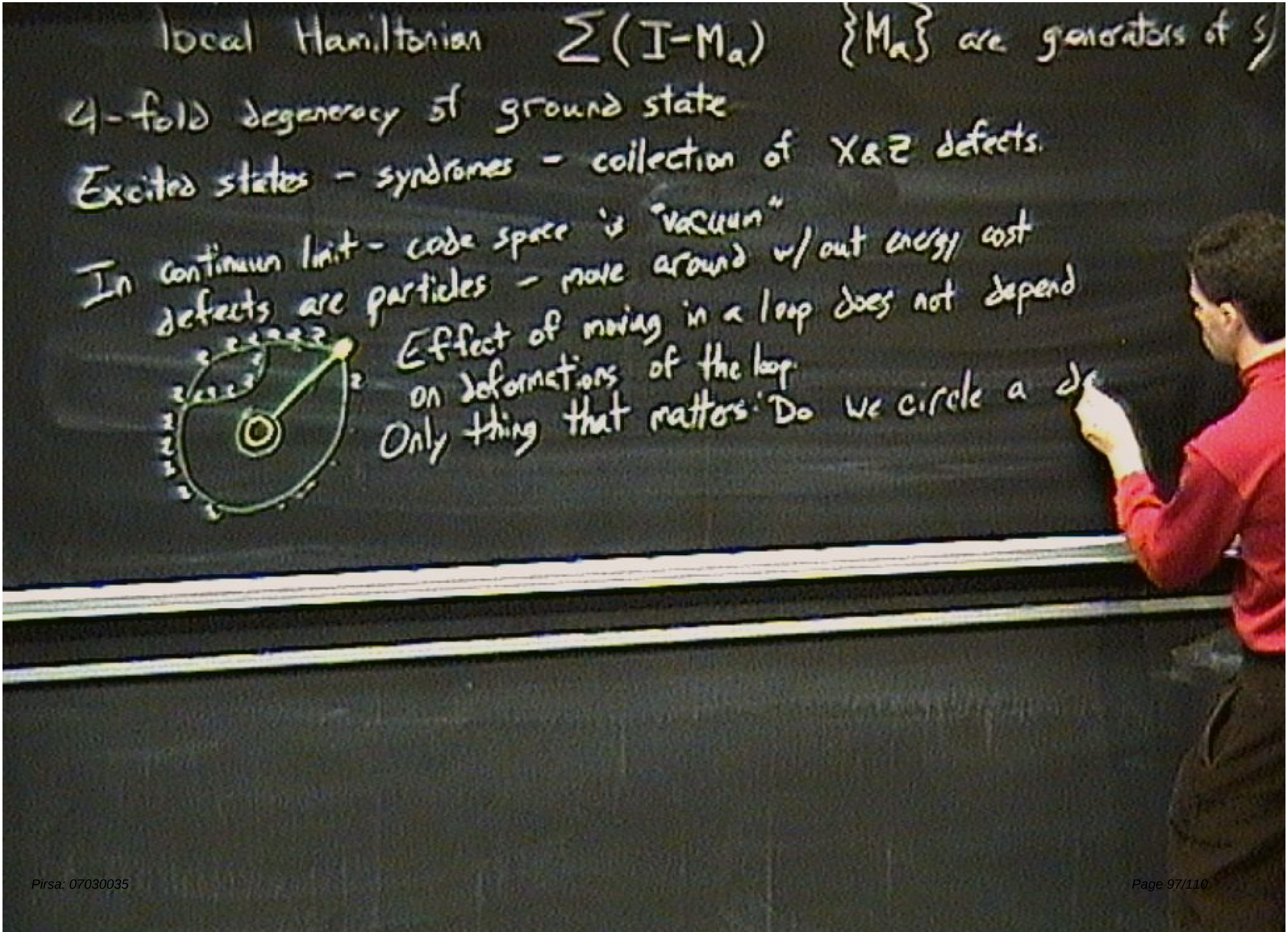


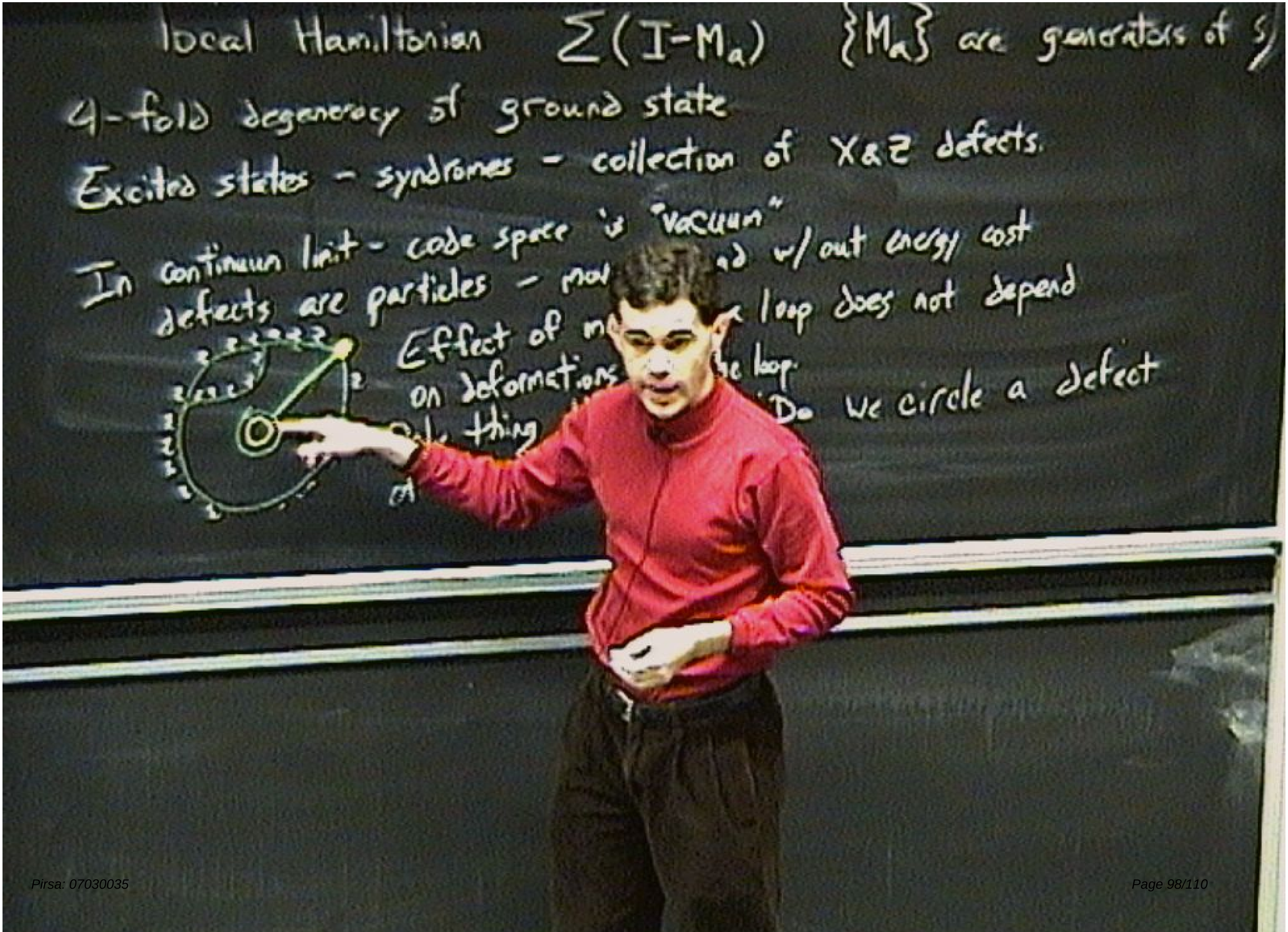
Codespace of toric code is ground state of a
local Hamiltonian $\sum(I-M_a)$ $\{M_a\}$ are generators of S.
4-fold degeneracy of ground state
Excited states - syndromes - co of X&Z defects.
In continuum limit - code space w/out energy cost
defects are particles - move does not depend
Effect of
on deformati



Correct pairing & our choice of
pairing together form a cycle.
If cycle is exact, we correct state
If inexact, there is a logical error
Variants of toric co
– Other graphs
– On other surfaces
⇒ 2g encoded qubits)
– Make planar co
appropriate boundary
conditions


Codespace of toric code is ground state of a
local Hamiltonian $\sum(I-M_a)$ $\{M_a\}$ are generators of S.
4-fold degeneracy of ground state
Excited states - syndromes - collection of X&Z defects.
In continuum limit - code space is "vacuum"
cost
defects are particles - move around w/
depend
Effect of moving in a loop
on deformations of the loop


local Hamiltonian $\sum(I-M_a)$ $\{M_a\}$ are generators of S
4-fold degeneracy of ground state
Excited states - syndromes - collection of X&Z defects.
In continuum limit - code space is "vacuum"
defects are particles - move around w/out energy cost
Effect of moving in a loop does not depend
on deformations of the loop.
Only thing that matters: Do we circle a de


local Hamiltonian $\sum(I-M_a)$ $\{M_a\}$ are generators of
4-fold degeneracy of ground state
Excited states - syndromes - collection of X&Z defects.
In continuum limit - code space is "vacuum"
defects are particles
w/out energy cost
loop does not depend
Effect of
on deformations
the loop.
Do we circle a defect

local Hamiltonian $\sum (I - M_a)$ $\{M_a\}$ are generators of S

4-fold degeneracy of ground state

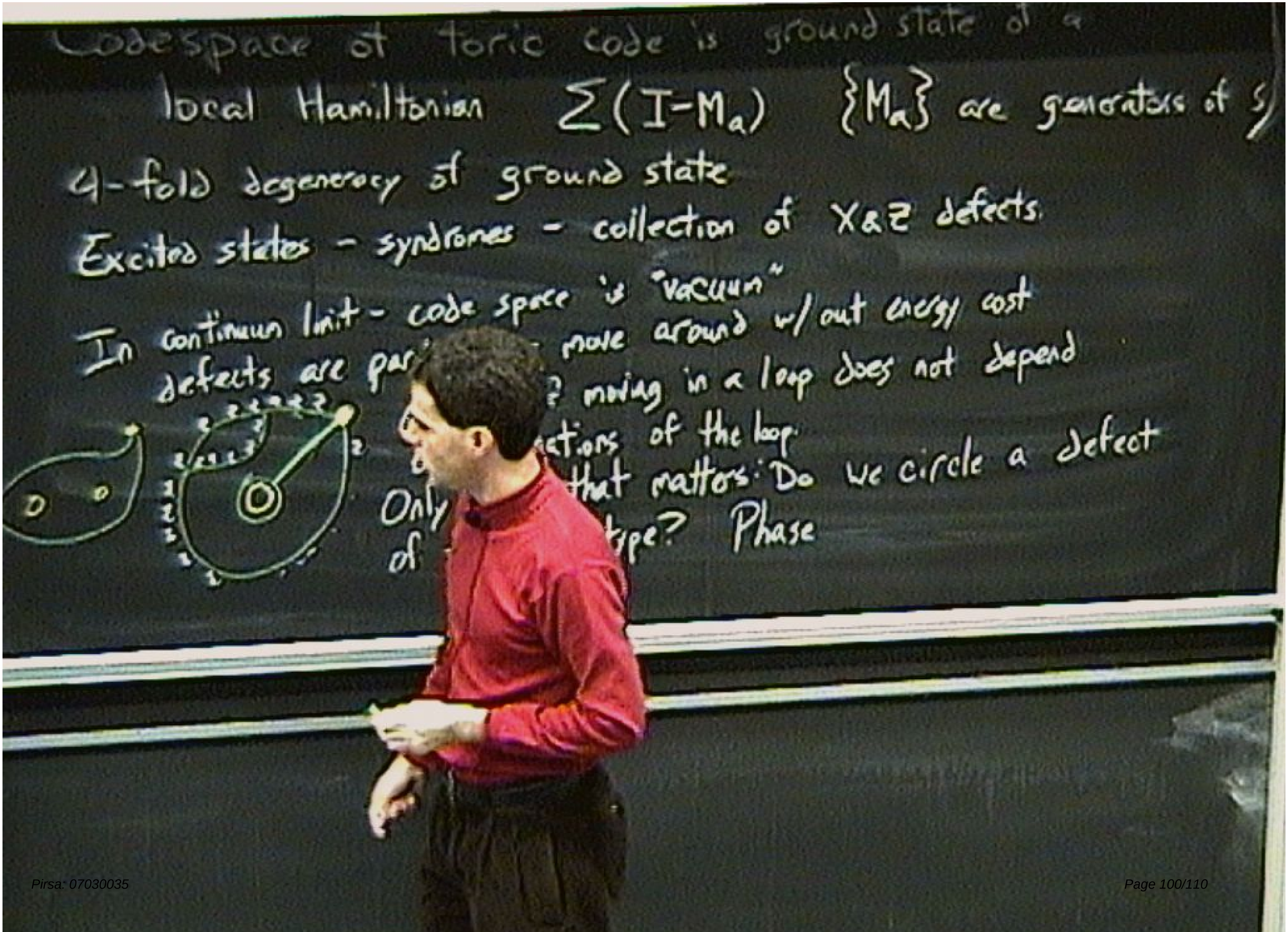
Excited states - syndromes - collection of X & Z defects.

In continuum limit - code space is "vacuum"

defects are particles - move around w/out energy cost

Effect of moving in a loop does not depend on deformations of the loop.

Only thing that matters: Do we circle a defect of the other type? Phase



Codespace of toric code is ground state of a
local Hamiltonian Σ(I-M_a) {M_a} are generators of S
4-fold degeneracy of ground state
Excited states - syndromes - collection of X&Z defects.
In continuum limit - code space is "vacuum"
defects are par
move around w/out energy cost
moving in a loop does not depend
ctions of the loop
that matters: Do we circle a defect
Only
of
type? Phase

Codespace of toric code is ground state of a

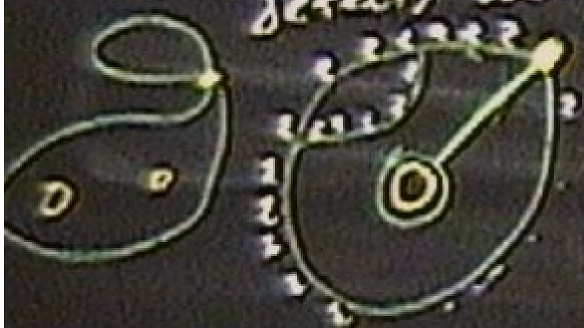
local Hamiltonian $\sum(I-M_a)$ $\{M_a\}$ are generators of S

4-fold degeneracy of ground state

Excited states - syndromes - collection of X&Z defects.

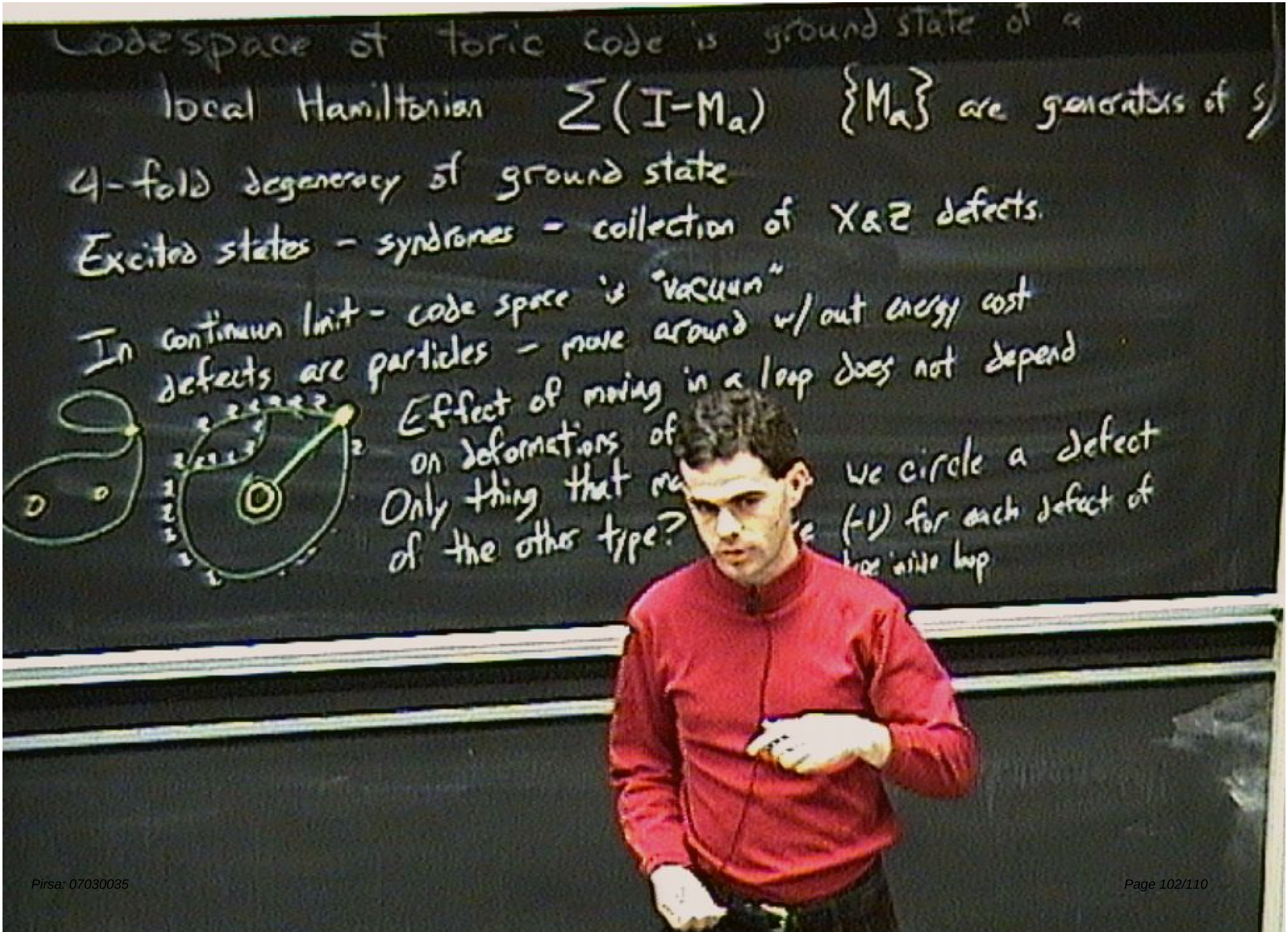
In continuum limit - code space is "vacuum"

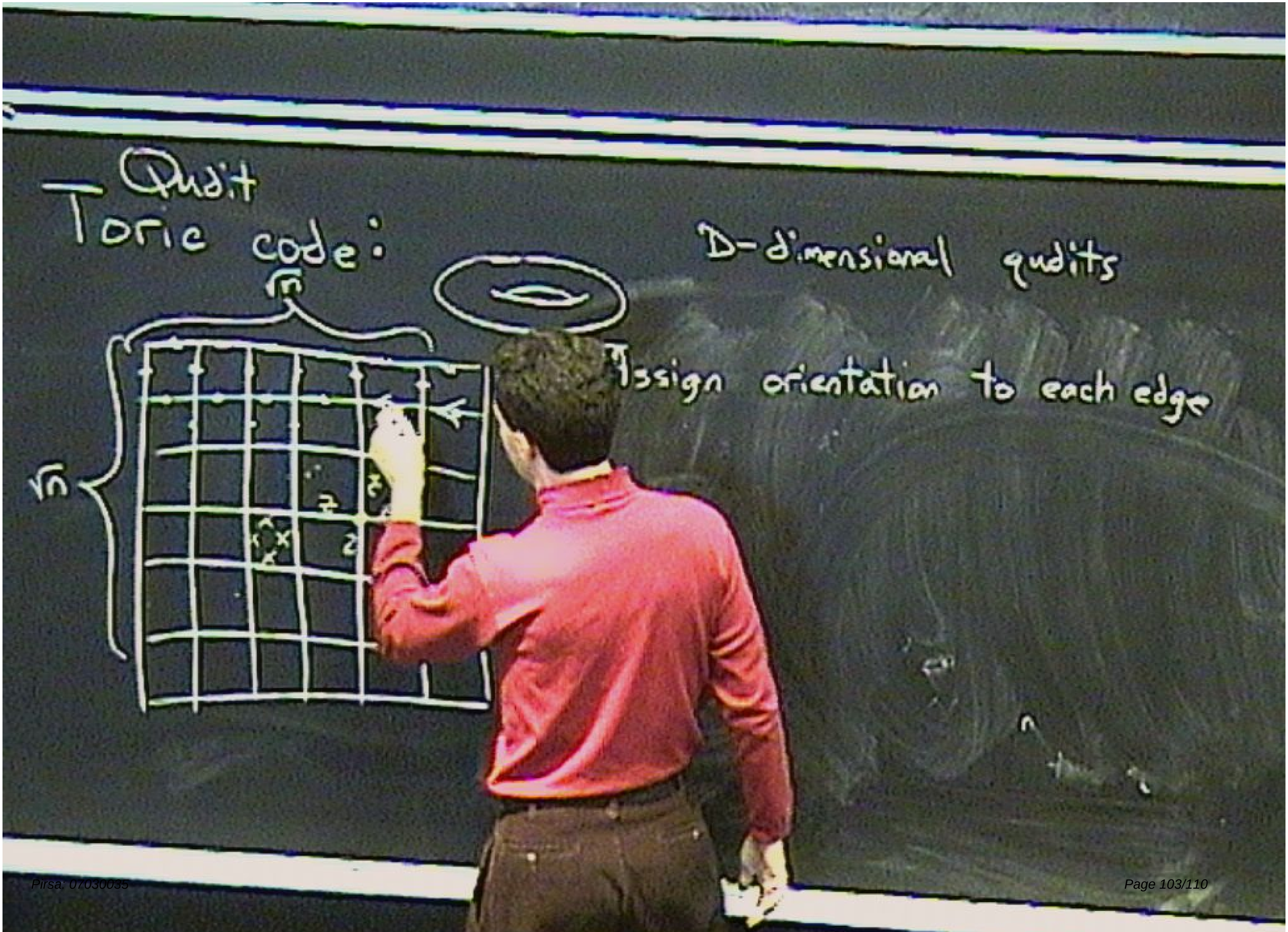
defects are particles - move around w/out energy cost

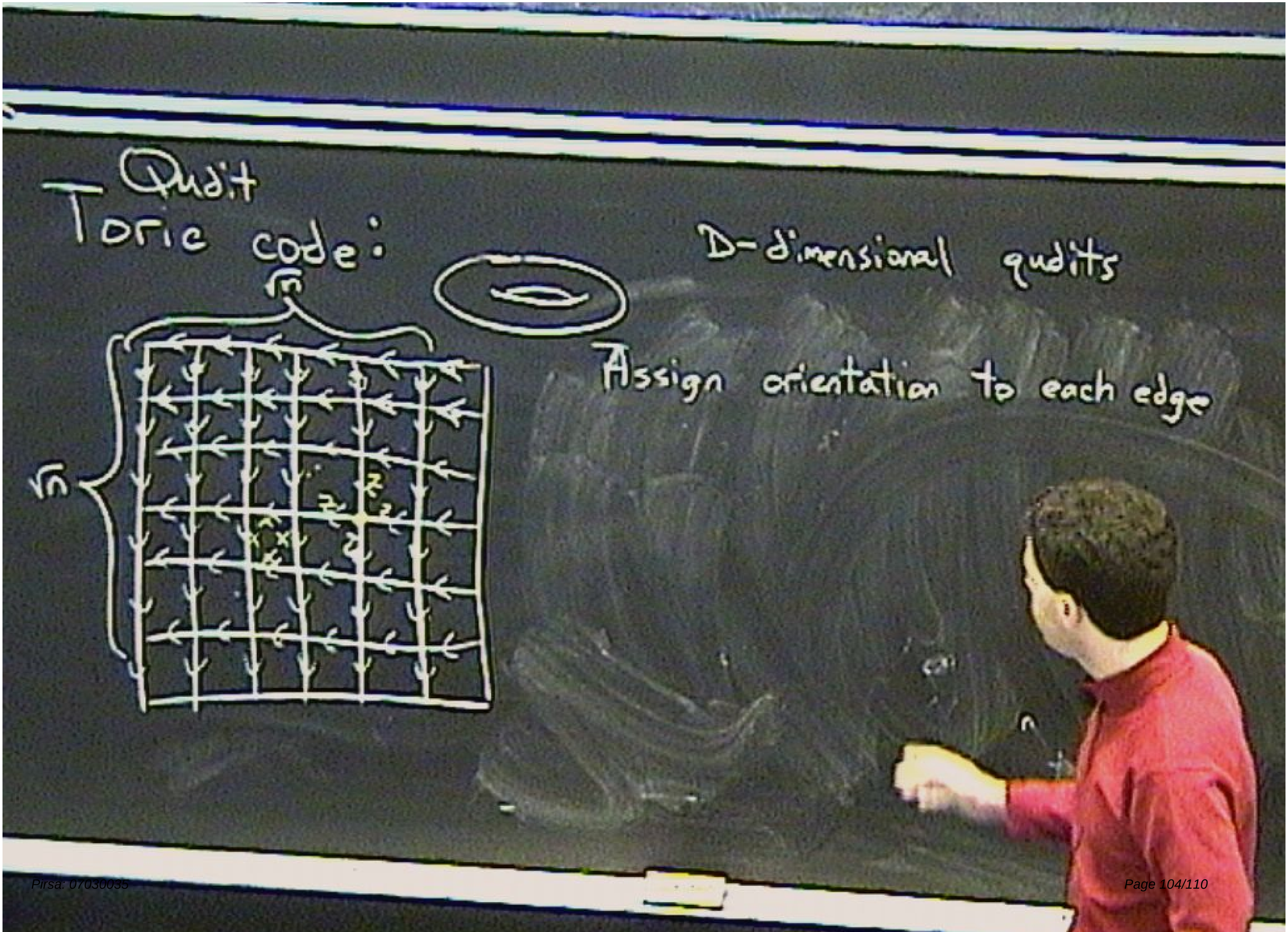


Effect of moving in a loop does not depend on deformations of the loop

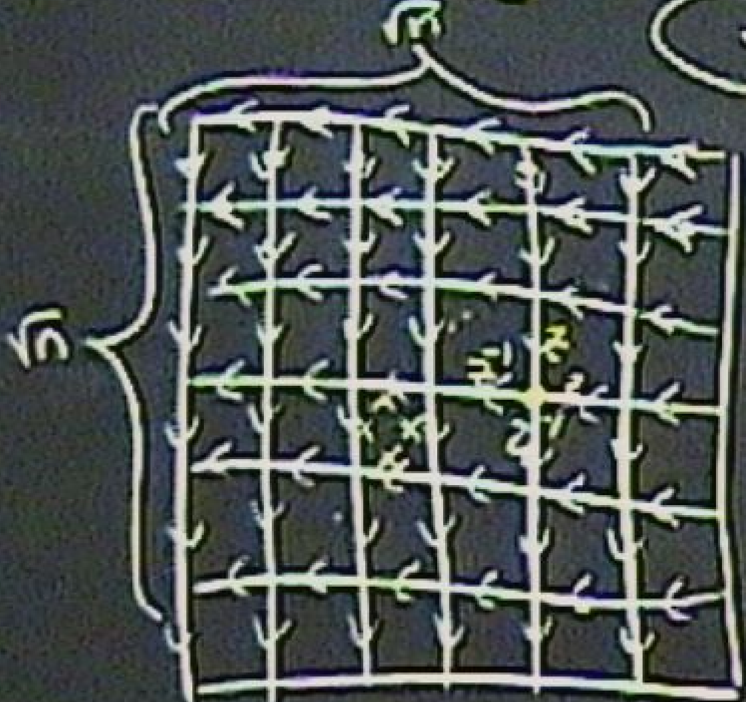
Only thing that matters: Do we circle a defect of the other type? Phase (-1) for each defect of other type inside loop


Codespace of toric code is ground state of a
local Hamiltonian Σ(I-Ma) {Ma} are generators of S
4-fold degeneracy of ground state
Excited states - syndromes - collection of X&Z defects.
In continuum limit - code space is "vacuum"
defects are particles - move around w/out energy cost
Effect of moving in a loop does not depend
on deformations of
Only thing that
we circle a defect
of the other type?
(-1) for each defect of
inside loop


Qudit
Toric code:
√n
√n
D-dimensional qudits
Assign orientation to each edge


Qudit
Toric code:
D-dimensional qudits
√n
√n
Assign orientation to each edge

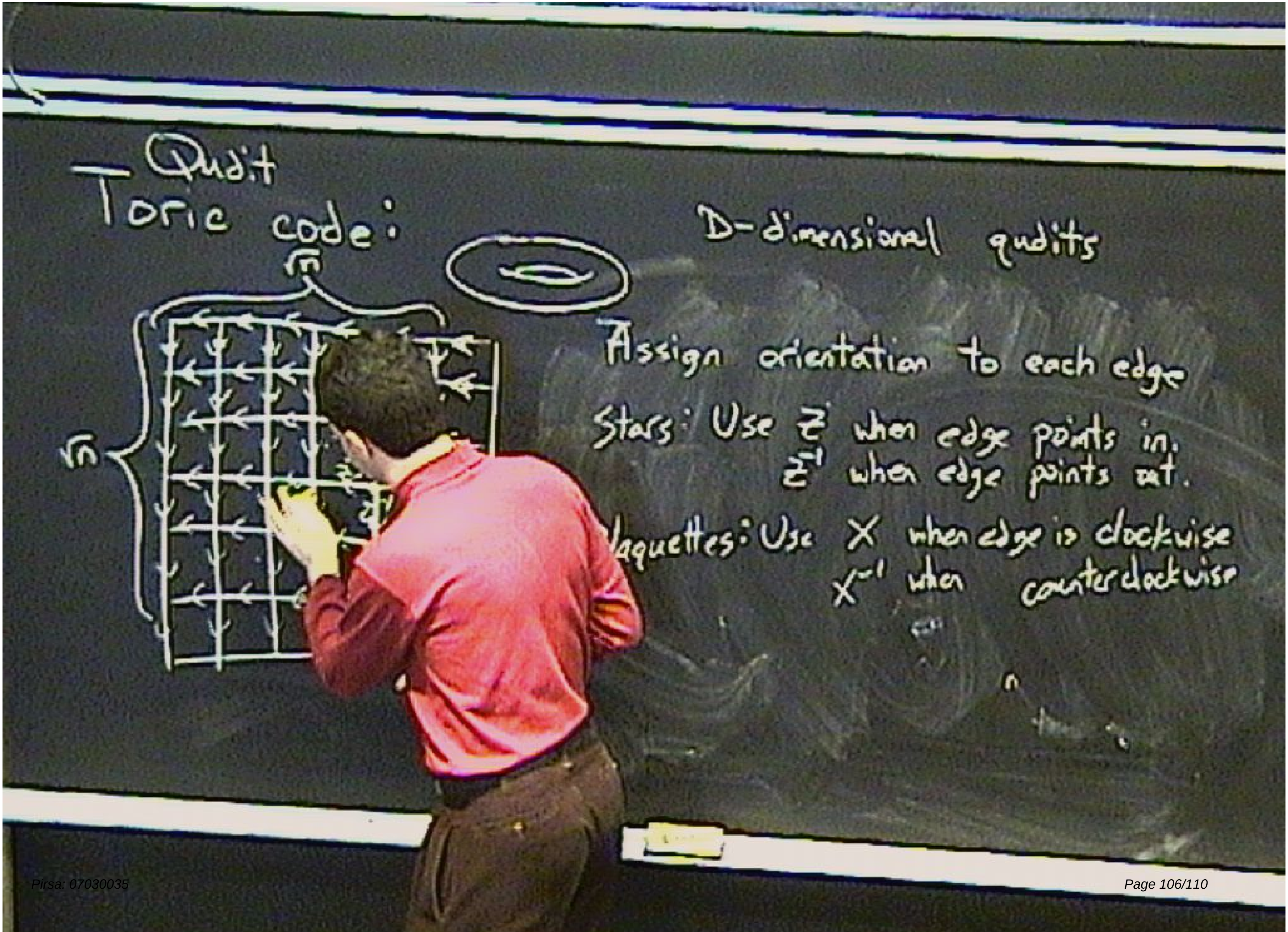
Qudit Toric code:



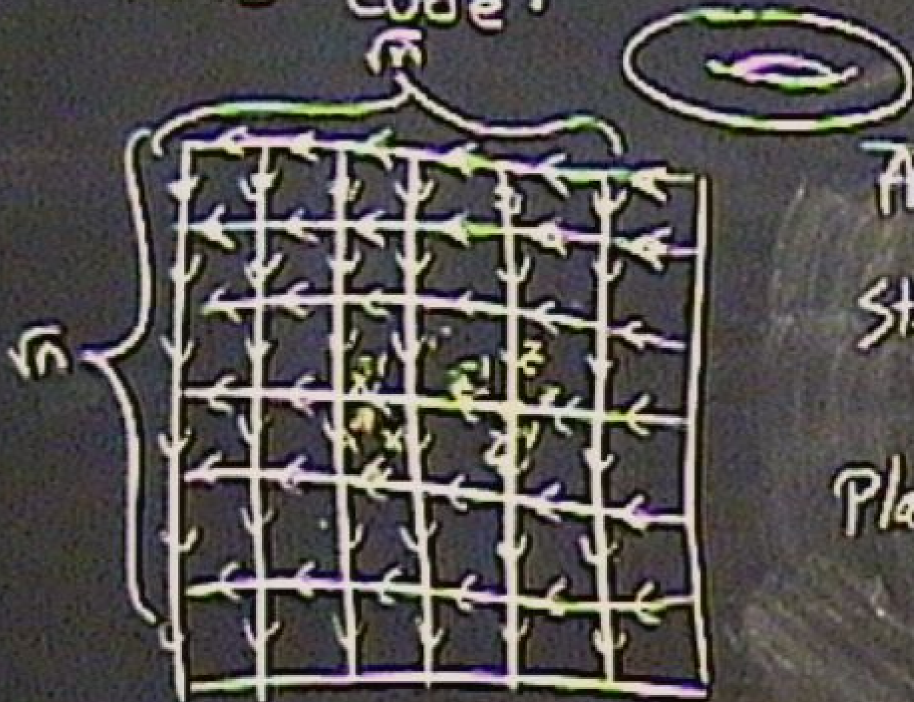
D-dimensional qudits

Assign orientation to each edge

Stars: Use Z when edge points in.
Z^{-1} when edge points out.


Qudit
Toric code:
√n
√n
D-dimensional qudits
Assign orientation to each edge
Stars: Use Z when edge points in.
Z^-1 when edge points out.
laquettes: Use X when edge is clockwise
X^-1 when counterclockwise

Qudit
Toric code:

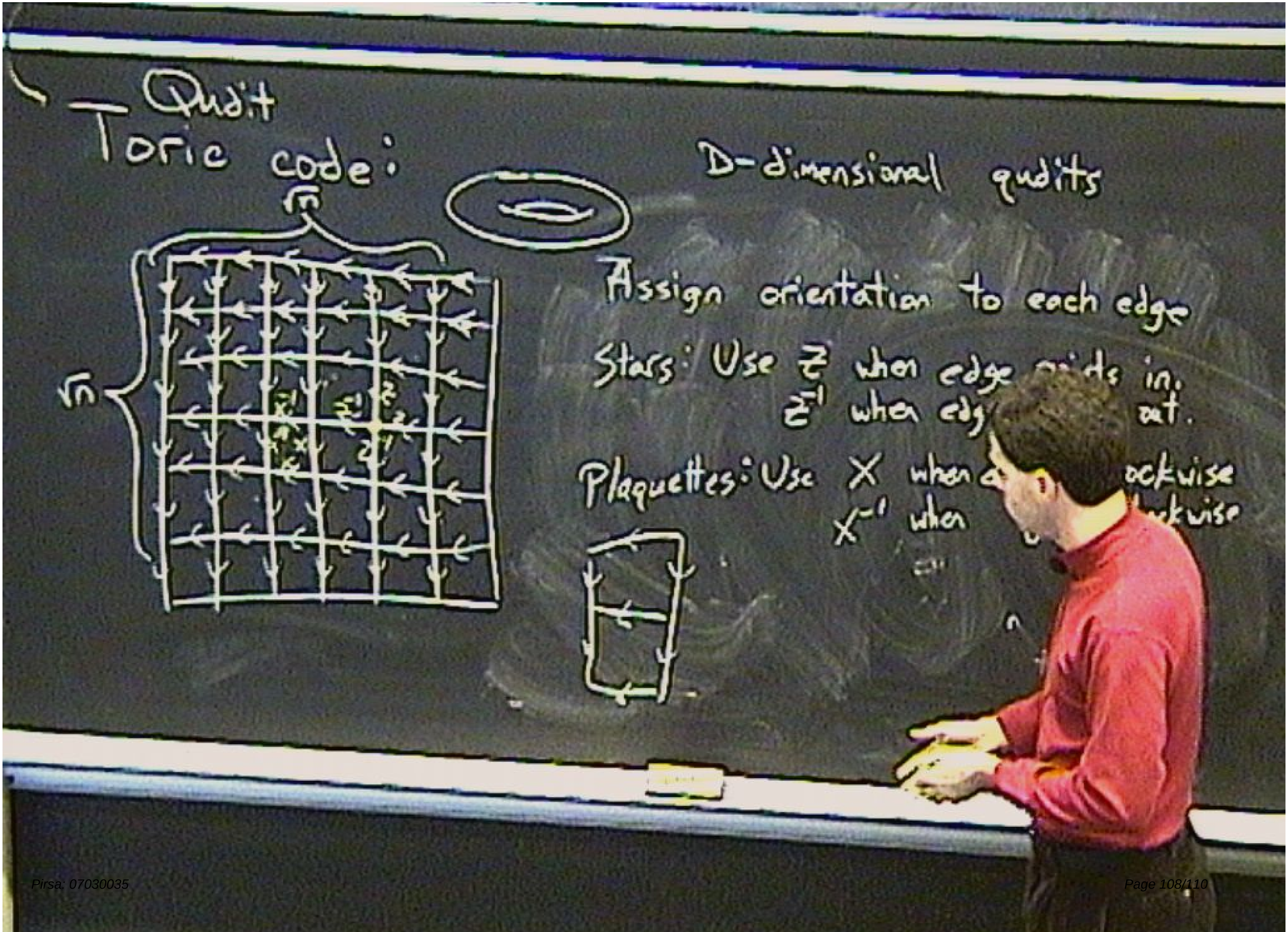


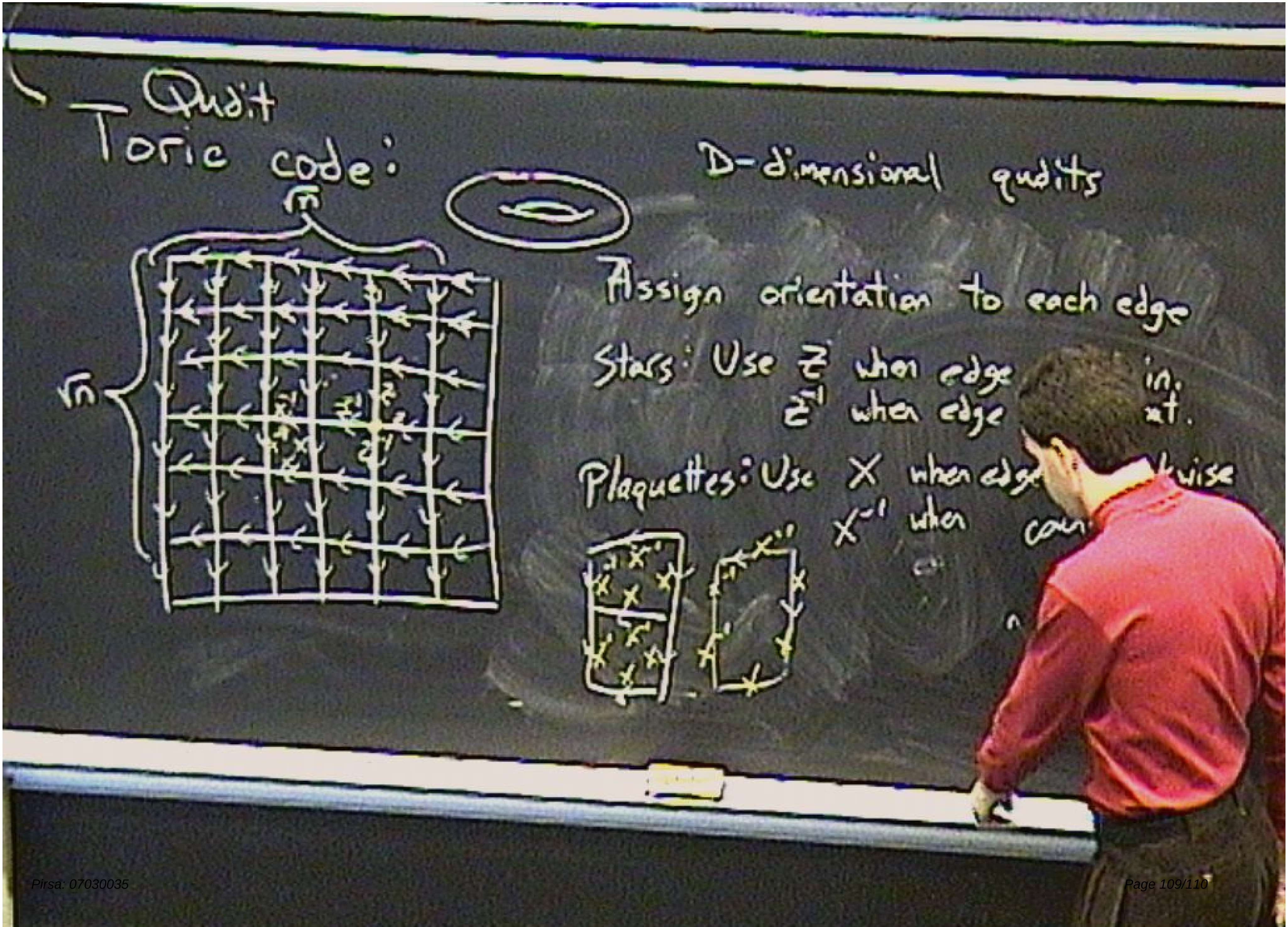
D-dimensional qudits

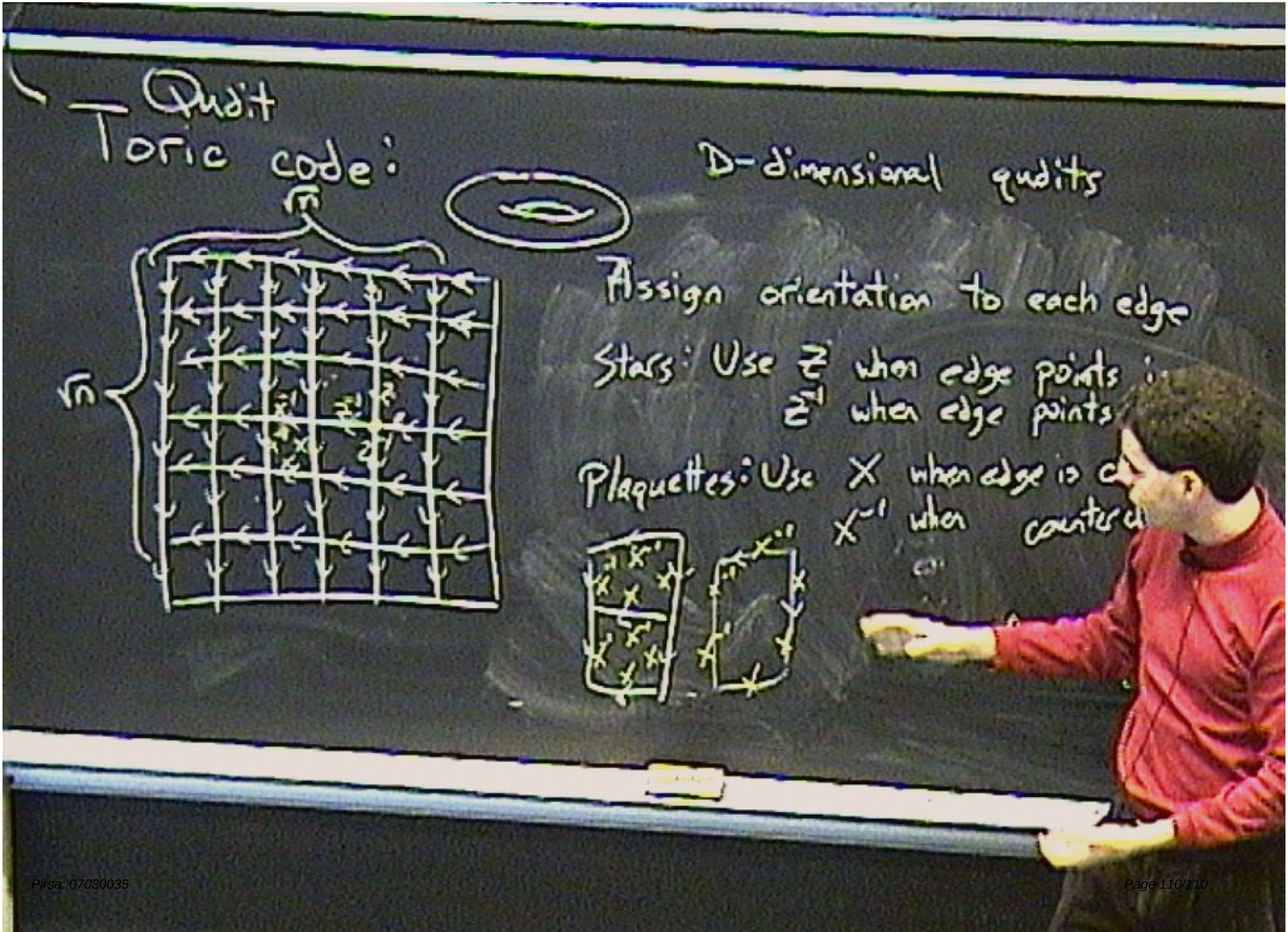
Assign orientation to each edge

Stars: Use Z when edge points in.
Z^{-1} when edge points out.

Plaquettes: Use X when edge is clockwise
X^{-1} when counterclockwise


Qudit
Toric code:
√n
√n
D-dimensional qudits
Assign orientation to each edge
Stars: Use Z when edge
in.
Z⁻¹ when edg
out.
Plaquettes: Use X when
ockwise
X⁻¹ when


Qudit
Toric code:
D-dimensional qudits
$\sqrt{n}$
$\sqrt{n}$
Assign orientation to each edge
Stars: Use Z when edge
in.
Z^{-1} when edge
ut.
Plaquettes: Use X when edge
wise
X^{-1} when
coun


Qudit
Toric code:
√n
√n
D-dimensional qudits
Assign orientation to each edge
Stars: Use Z when edge points
Z⁻¹ when edge points
Plaquettes: Use X when edge is c
X⁻¹ when counterc