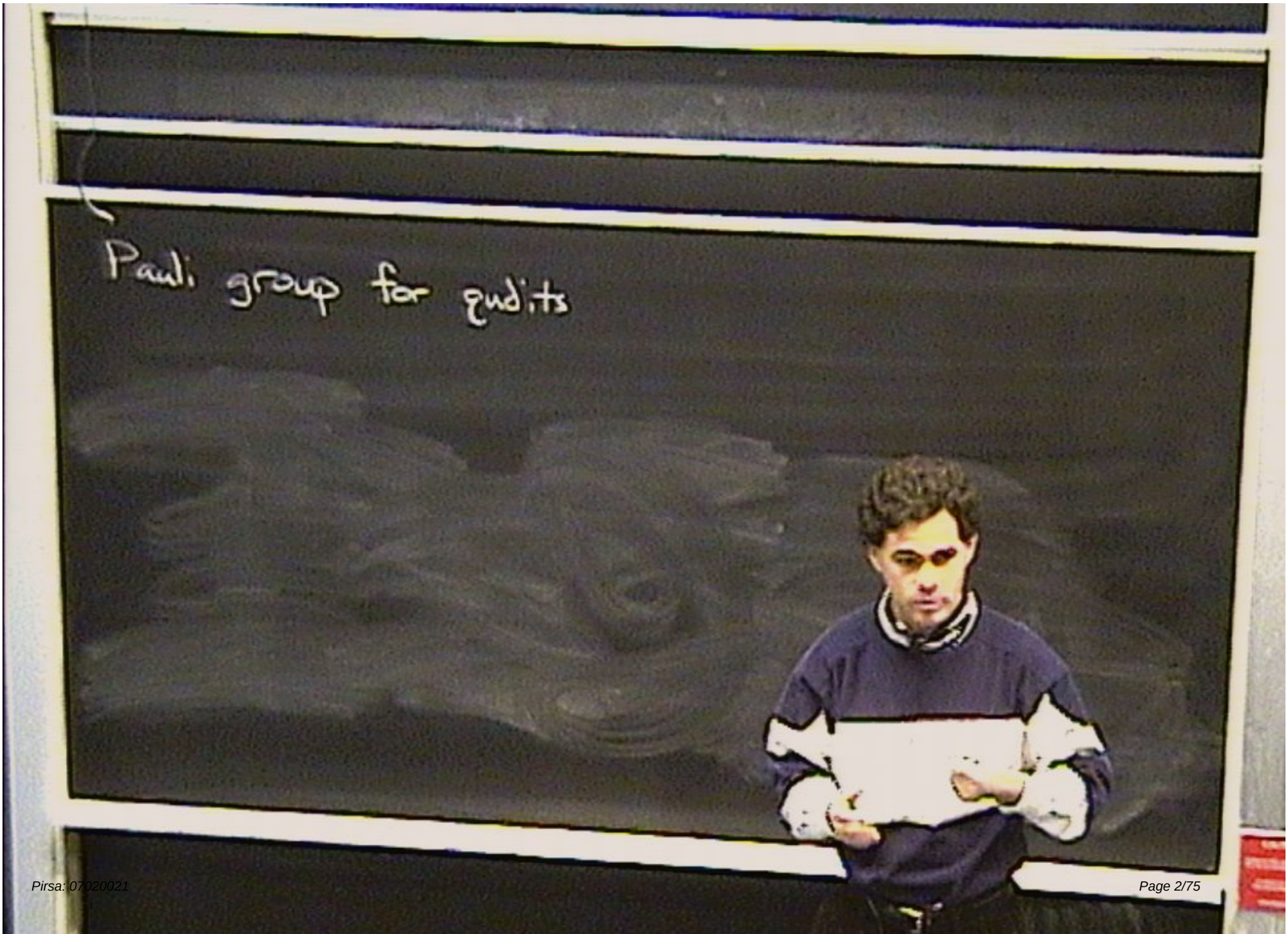
Title: Quantum Error Correction 6A

Date: Feb 13, 2007 03:30 PM

URL: http://pirsa.org/07020021

Abstract: Examples of qudit stabilizer codes, polynomial codes, Clifford group for qudits, introduction to fault-tolerance, definition of transversal gates, definition of fault-tolerant gates


Pauli group for qudits

Pauli group for qudits (dimension p prime):

$$\mathcal{P}_n = \{\omega^a X^b Z^c\} \quad \omega = e^{2\pi i/p}, \; X|j\rangle = |j+1\rangle, \; Z = \omega^j |j\rangle$$

Pauli group for qudits (dimension p prime):

$$P_n = \{\omega^a X^b Z^c\} \quad \omega = e^{2\pi i/p}, \; X|j\rangle = |j+1\rangle, \; Z = \omega^j |j\rangle$$

$$XZ = \omega^{-1} ZX$$

Pauli group for qudits (dimension p prime):

$$\mathcal{P}_n = \{\omega^a X^b Z^c\} \quad \omega = e^{2\pi i/p}, \; X|j\rangle = |j+1\rangle, \; Z = \omega^j |j\rangle$$

$$XZ = \omega^{-1} ZX$$

Stabilizer codes for qudits

Pauli group for qudits (dimension p prime):

$$\mathcal{P}_n = \{\omega^a X^b Z^c\} \quad \omega = e^{2\pi i/p}, \; X|j\rangle = |j+1\rangle, \; Z = \omega^j |j\rangle$$

$$XZ = \omega^{-1} ZX$$

Stabilizer codes for qudits

Example:

Pauli group for qudits (dimension p prime):

$$\mathcal{P}_n = \{\omega^a X^b Z^c\} \quad \omega = e^{2\pi i/p}, \; X|j\rangle = |j+1\rangle, \; Z = \omega^j |j\rangle$$

$$XZ = \omega^{-1} ZX$$

Stabilizer codes for qudits

Example: $[[3,1,2]]_3$

Pauli group for qudits (dimension p prime):

$$\mathcal{P}_n = \{\omega^a X^b Z^c\} \quad \omega = e^{2\pi i/p}, \; X|j\rangle = |j+1\rangle, \; Z = \omega^j |j\rangle$$

$$XZ = \omega^{-1} ZX$$

Stabilizer codes for qudits

Example: $[[3,1,2]]_3$

$$X \otimes X \otimes X$$
$$Z \otimes Z \otimes Z$$

Pauli group for qudits (dimension p prime):

$$\mathcal{P}_n = \{\omega^a X^b Z^c\} \quad \omega = e^{2\pi i/p},\ X|j\rangle = |j+1\rangle,\ Z = \omega^j |j\rangle$$

$$XZ = \omega^{-1} ZX$$

Stabilizer codes for qudits

Example: $[[3,1,2]]_3$

$$X \otimes X \otimes X$$
$$Z \otimes Z \otimes Z$$

$$(X \otimes X \otimes X)(Z \otimes Z \otimes Z) = \omega^3 (Z \otimes Z \otimes Z)(X \otimes X \otimes X)$$

Pauli group for qudits (dimension p prime):

$$\mathcal{P}_n = \{\omega^a X^b Z^c\} \quad \omega = e^{2\pi i/p},\ X|j\rangle = |j+1\rangle,\ Z = \omega^j |j\rangle$$

$$XZ = \omega^{-1} ZX$$

Stabilizer codes for qudits

Example: $[[3,1,2]]_3$

$$X \otimes X \otimes X$$
$$Z \otimes Z \otimes Z$$

Z error $\Rightarrow \omega X \otimes X \otimes X$

Z^2 error $\Rightarrow \omega^2 X \otimes X \otimes X$

$$(X \otimes X \otimes X)(Z \otimes Z \otimes Z) = \omega^3 (Z \otimes Z \otimes Z)(X \otimes X \otimes X)$$

Pauli group for qudits (dimension p prime):

$$\mathcal{P}_n = \{\omega^a X^b Z^c\} \qquad \omega = e^{2\pi i/p},\ X|j\rangle = |j+1\rangle,\ Z = \omega^j |j\rangle$$

$$XZ = \omega^{-1} ZX$$

Stabilizer codes for qudits

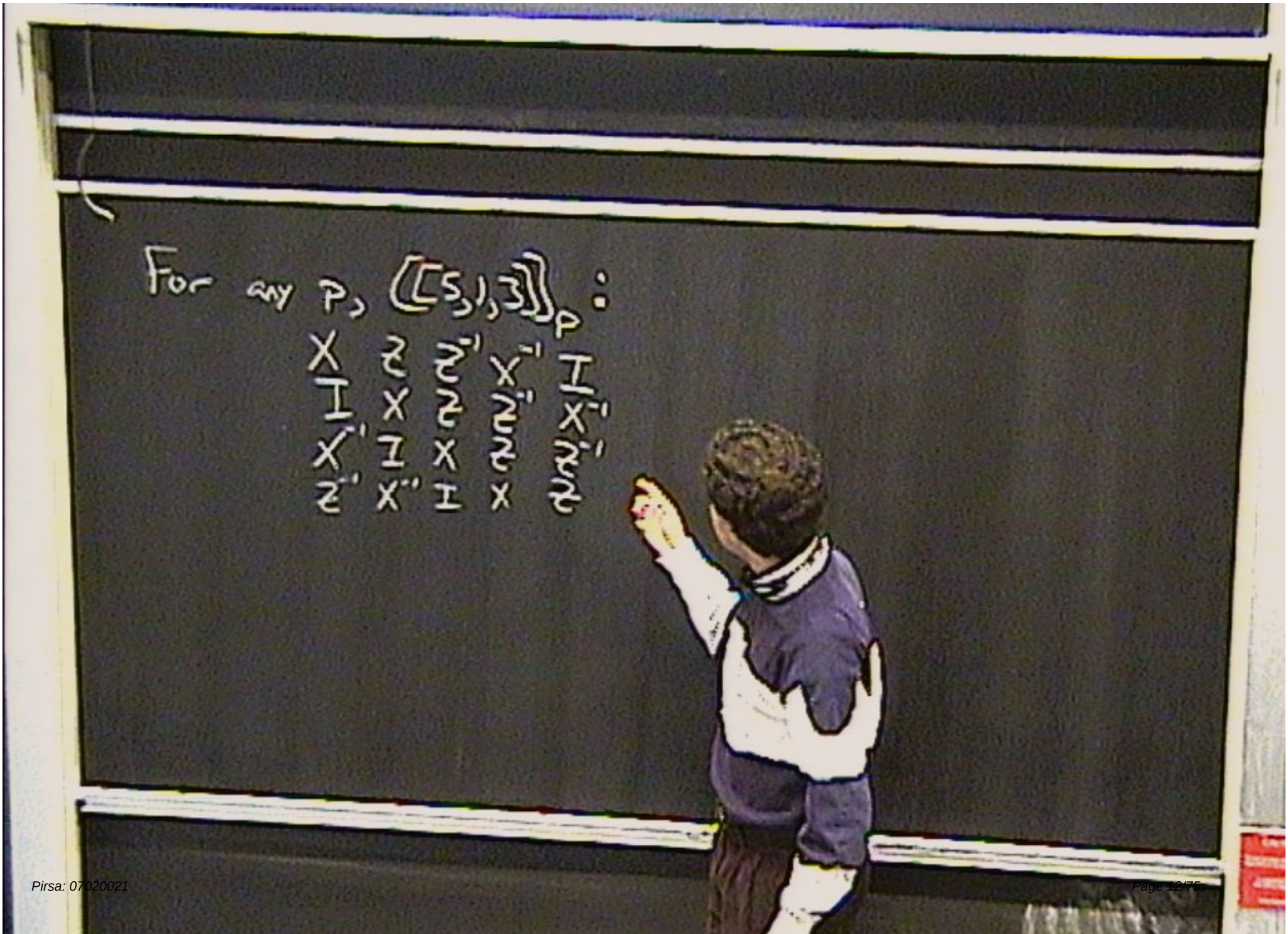
Example: $[[3,1,2]]_3$

$$X \otimes X \otimes X$$
$$Z \otimes Z \otimes Z$$

Z error $\Rightarrow \omega X \otimes X \otimes X$

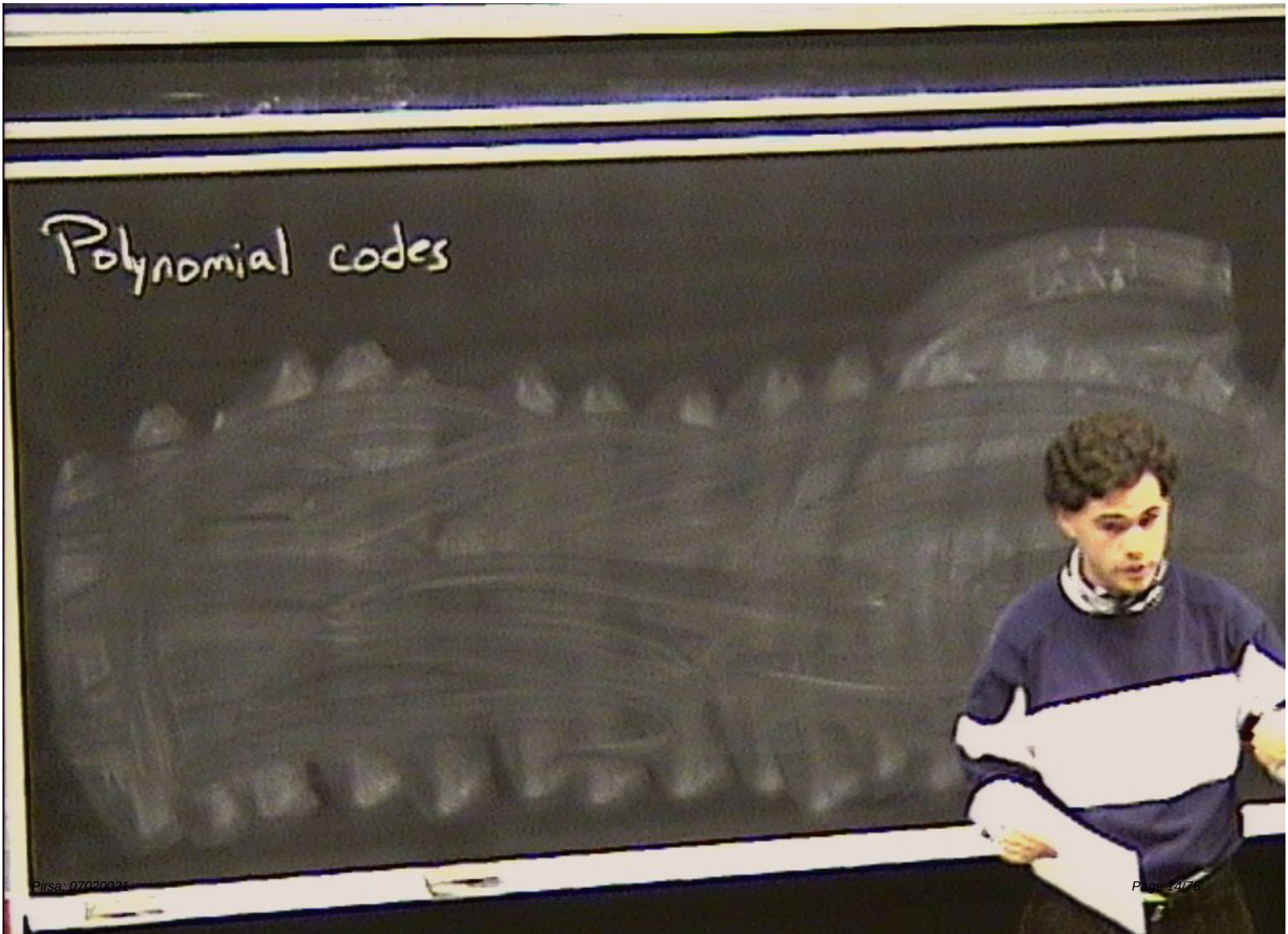
Z^2 error $\Rightarrow \omega^2 X \otimes X \otimes X$

$$(X \otimes X \otimes X)(Z \otimes Z \otimes Z) = \omega^3 (Z \otimes Z \otimes Z)(X \otimes X \otimes X)$$


For any p, ([S,1],3]]p :
X Z Z⁻¹ X⁻¹ I
I X Z Z⁻¹ X⁻¹
X⁻¹ I X Z Z⁻¹
Z⁻¹ X⁻¹ I X Z

For any p, $([S_1, S_3])_p$:

$$
\begin{array}{ccccc}
X & Z & Z^{-1} & X^{-1} & I \\
I & X & Z & Z^{-1} & X^{-1} \\
X^{-1} & I & X & Z & Z^{-1} \\
Z^{-1} & X^{-1} & I & X & Z
\end{array}
$$


Polynomial codes

Polynomial codes (based on classical Reed-Solomon codes):

Polynomial codes (based on classical Reed-Solomon codes):

$$|a_0, a_1, \ldots, a_{k-1}\rangle \longmapsto \sum_f |f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{n-1})\rangle$$

Polynomial codes (based on classical Reed-Solomon codes):

$$|a_0, a_1, \ldots, a_{k-1}\rangle \longmapsto \sum_f |f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{n-1})\rangle$$

$\alpha_0, \ldots, \alpha_{n-1}$

Polynomial codes (based on classical Reed-Solomon codes):

$$|a_0, a_1, \ldots, a_{k-1}\rangle \longmapsto \sum_f |f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{n-1})\rangle$$

$\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{Z}_p \setminus \{0\}$, f is polynomial of degree $\leq g$ $(g > k)$

$f = \sum_{j=0}^{g} c_j x^j$, $c_j = a_j$ $(0 \leq j \leq k-1)$, sum over

Polynomial codes (based on classical Reed-Solomon codes):

$$|a_0, a_1, \ldots, a_{k-1}\rangle \longmapsto \sum_f |f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{n-1})\rangle$$

$\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{Z}_p \setminus \{0\}$, f is polynomial of degree $\leq g$ $(g > k)$

$f = \sum_{j=0}^{g} c_j x^j$, $c_j = a_j$ $(0 \leq j \leq k-1)$, sum over c_j $(j \geq k)$

Polynomial codes (based on classical Reed-Solomon codes):

$$|a_0, a_1, \ldots, a_{k-1}\rangle \longmapsto \sum_f |f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{n-1})\rangle$$

$\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{Z}_p \setminus \{0\}$, f is polynomial of degree $\leq g$ $(g > k)$

$f = \sum_{j=0}^{g} c_j x^j$, $c_j = a_j$ $(0 \leq j \leq k-1)$, sum over c_j $(j \geq k)$

Polynomial codes (based on classical Reed-Solomon codes):

$$|a_0, a_1, \ldots, a_{k-1}\rangle \longmapsto \sum_f |f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{n-1})\rangle$$

$\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{Z}_p \setminus \{0\}$, f is polynomial of degree $\leq g$ $(g > k)$

$(\alpha_i \neq \alpha_j)$

$f = \sum_{j=0}^{g} c_j x^j$, $c_j = a_j$ $(0 \leq j \leq k-1)$, sum over c_j $(j \geq k)$

Polynomial codes (based on classical Reed-Solomon codes):

$$|a_0, a_1, \ldots, a_{k-1}\rangle \longmapsto \sum_f |f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{n-1})\rangle$$

$\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{Z}_p \setminus \{0\}$, f is polynomial of degree $\leq g$ $(g > k)$

$(\alpha_i \neq \alpha_j)$

$f = \sum_{j=0}^{g} c_j x^j$, $c_j = a_j$ $(0 \leq j \leq k-1)$, sum over c_j $(j \geq k)$

Example: $k=1$, $g=2$, $n=3$, $p=5$

Polynomial codes (based on classical Reed-Solomon codes):

$$|a_0, a_1, \ldots, a_{k-1}\rangle \longmapsto \sum_f |f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{n-1})\rangle$$

$\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{Z}_p \setminus \{0\}$, f is polynomial of degree $\leq g$ $(g > k)$

$(\alpha_i \neq \alpha_j)$

$(p > n)$ $\quad f = \sum_{j=0}^{g} c_j x^j$, $c_j = a_j$ $(0 \leq j < k-1)$, sum over c_j $(j > k)$

Example: $k=1$, $g=2$, $n=3$, $p=5$

Polynomial codes (based on classical Reed-Solomon codes):

$$|a_0, a_1, \ldots, a_{k-1}\rangle \longmapsto \sum_f |f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{n-1})\rangle$$

$\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{Z}_p \setminus \{0\}$, f is polynomial of degree $\leq g$ $(g > k)$

$(\alpha_i \neq \alpha_j)$

$(p > n)$ $\quad f = \sum_{j=0}^{g} c_j x^j$, $c_j = a_j$ $(0 \leq j \leq k-1)$, sum over c_j $(j \geq k)$

Example: $k=1$, $g=2$, $n=3$, $p=5$

$$|a\rangle \longmapsto |$$

$f = a + c_1 x + c_2 x^2$

$\alpha_0 = 1$, $\alpha_1 = 2$, $\alpha_2 = 3$

Polynomial codes (based on classical Reed-Solomon codes):

$$|a_0, a_1, \ldots, a_{k-1}\rangle \longmapsto \sum_f |f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{n-1})\rangle$$

$\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{Z}_p \setminus \{0\}$, f is polynomial of degree $\leq g$ $(g > k)$

$(\alpha_i \neq \alpha_j)$

$(p > n)$ $\quad f = \sum_{j=0}^{g} c_j x^j$, $c_j = a_j$ $(0 \leq j < k-1)$, sum over c_j $(j \geq k)$

Example: $k=1$, $g=2$, $n=3$, $p=5$

$$|a\rangle \longmapsto |a, a, a\rangle + |a+1, a+2, a+3\rangle + \ldots + |a+c_1, a+2c_1, a+3c_1\rangle$$

$f = a + c_1 x + c_2 x^2$

$\alpha_0 = 1$, $\alpha_1 = 2$, $\alpha_2 = 3$

Polynomial codes (based on classical Reed-Solomon codes):

$$|a_0, a_1, \ldots, a_{k-1}\rangle \longmapsto \sum_f |f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{n-1})\rangle$$

$\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{Z}_p \setminus \{0\}$, f is polynomial of degree $\leq g$ $(g > k)$

$(\alpha_i \neq \alpha_j)$

$(p > n)$ $\quad f = \sum_{j=0}^{g} c_j x^j$, $c_j = a_j$ $(0 \leq j \leq k-1)$, sum over c_j $(j \geq k)$

Example: $k=1$, $g=2$, $n=3$, $p=5$

$$|a\rangle \longmapsto |a, a, a\rangle + |a+1, a+2, a+3\rangle + \ldots + |a+c_1, a+2c_1, a+3c_1\rangle + \ldots$$

$$f = a + c_1 x + c_2 x^2 \quad + |a+1, a+4, a+4\rangle + |a+2, a+1, a+2\rangle + \ldots$$

$$+ \ldots |a+c_1+c_2, a+2c_1+4c_2, a+3c_1+4c_2\rangle + \ldots$$

$\alpha_0 = 1$, $\alpha_1 = 2$, $\alpha_2 = 3$

CSS code: $C_1 = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} < g\}$

$C_2^\perp = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \leq g,\ c_j = 0\ (j<k)\}$

CSS code: $C_1 = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} < g\}$

$C_2^\perp = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} < g,\ c_j = 0\ (j<k)\}$

$d(C_1)$:

CSS code: $C_1 = \{(f(\alpha_0), \dots, f(\alpha_{n-1})) \mid f \text{ degree} < g\}$

$C_2^\perp = \{(f(\alpha_0), \dots, f(\alpha_{n-1})) \mid f \text{ degree} \leq g, c_j = 0 \ (j<k)\}$

$d(C_1)$: Need $g+1$ points to reconstruct f

$\Rightarrow n-(g+1)$ erasures can be corrected

$d_1 = n-g$

CSS code: $C_1 = \{(f(\alpha_0), \dots, f(\alpha_{n-1})) \mid f \text{ degree} < g\}$

$C_2^\perp = \{(f(\alpha_0), \dots, f(\alpha_{n-1})) \mid f \text{ degree} < g,\ c_j = 0\ (j<k)\}$

$d(C_1)$: Need $g+1$ points to reconstruct f

$\Rightarrow n-(g+1)$ erasures can be corrected

$d_1 = n-g$

$d(C_1^\perp)$

CSS code: $C_1 = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \leq g\}$

$C_2^\perp = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \leq g, c_j = 0 \ (j<k)\}$

$d(C_1)$: Need $g+1$ points to reconstruct f

$\Rightarrow n-(g+1)$ erasures can be corrected

$d_1 = n-g$

$d(C_1^\perp)$:

CSS code: $C_1 = \{(f(\alpha_0), \dots, f(\alpha_{n-1})) \mid f \text{ degree} \le g\}$

$C_2^\perp = \{(f(\alpha_0), \dots, f(\alpha_{n-1})) \mid f \text{ degree} \le g,\ c_j = 0\ (j<k)\}$

$d(C_1)$: Need $g+1$ points to reconstruct f

$\Rightarrow n-(g+1)$ erasures can be corrected

$d_1 = n-g$

$d(C_1^\perp)$: $g+1$ points of C_1 are arbitrary

CSS code: $C_1 = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \leq g\}$

$C_2^{\perp} = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \leq g, c_j = 0 \ (j<k)\}$

$d(C_1)$: Need $g+1$ points to reconstruct f

$\Rightarrow n-(g+1)$ erasures can be corrected

$d_1 = n-g$

$d(C_1^{\perp})$: $g+1$ points of C_1 are arbitrary, $g+2$ are constrained

$\Rightarrow d = g+2$

CSS code: $C_1 = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \leq g\}$

$C_2^\perp = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \leq g,\ c_j = 0\ (j<k)\}$

$d(C_1)$: Need $g+1$ points to reconstruct f

$\Rightarrow n-(g+1)$ erasures can be corrected

$d_1 = n-g$

$d(C_1^\perp)$: $g+1$ points of C_1 are arbitrary, $g+2$ are constrained

$\Rightarrow d = g+2$

$d(C_2)$:

$(C_1^\perp \subset C_2)$

CSS code: $C_1 = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \leq g\}$

$C_2^\perp = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \leq g,\ c_j = 0\ (j<k)\}$

$d(C_1)$: Need $g+1$ points to reconstruct f

$\Rightarrow n-(g+1)$ erasures can be corrected

$d_1 = n-g$

$d(C_1^\perp)$: $g+1$ points of C_1 are arbitrary, $g+2$ are constrained

$\Rightarrow d = g+2$

$d(C_2)$: $g+1-k$ arbitrary points in $C_2^\perp \Rightarrow d_2 = g+2-k$

$(C_1^\perp \subset C_2)$

$C_2 = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \le g,\ c_j = 0\ (j<k)\}$

$d(C_1)$: Need $g+1$ points to reconstruct f

$\Rightarrow n-(g+1)$ erasures can be corrected

$d_1 = n-g$

$d(C_1^\perp)$: $g+1$ points of C_1 are arbitrary, $g+2$ are constrained

$\Rightarrow d = g+2$

$d(C_2)$: $g+1-k$ arbitrary points in $C_2^\perp \Rightarrow d_2 = g+2-k$

$(C_1^\perp \subset C_2)$ $d_Q = \min(d_1, d_2) = \min(n-g, g+2-k)$

Choose g so that $n-g = g+2-k \Rightarrow 2(g+1)$

$C_2 = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \le g,\ c_j = 0\ (j<k)\}$

$d(C_1)$: Need $g+1$ points to reconstruct f

$\Rightarrow n-(g+1)$ erasures can be corrected

$d_1 = n-g$

$d(C_1^\perp)$: $g+1$ points of C_1 are arbitrary, $g+2$ are constrained

$\Rightarrow d = g+2$

$d(C_2)$: $g+1-k$ arbitrary points in $C_2^\perp \Rightarrow d_2 = g+2-k$

$(C_1^\perp \subset C_2)$ $d_Q = \min(d_1, d_2) = \min(n-g, g+2-k)$

Choose g so that $n-g = g+2-k \Rightarrow (g+1) = \frac{n+k}{2}$

$C_2 = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \le g,\ c_j = 0\ (j<k)\}$

$d(C_1)$: Need $g+1$ points to reconstruct f

$\Rightarrow n-(g+1)$ erasures can be corrected

$d_1 = n-g$

$d(C_1^\perp)$: $j+1$ points of C_1 are arbitrary, $g+2$ are constrained

$\Rightarrow d = g+2$

$d(C_2)$: $g+1-k$ arbitrary points in $C_2^\perp \Rightarrow d_2 = g+2-k$

$(C_1^\perp \subset C_2)$ $d_Q = \min(d_1, d_2) = \min(n-g, g+2-k) = \frac{n-k}{2}+1$

Choose g so that $n-g = g+2-k \Rightarrow (g+1) = \frac{n+k}{2}$

CSS code: $C_1 = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \leq g\}$

$C_2^\perp = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \leq g,\ c_j = 0\ (j<k)\}$

$d(C_1)$: Need $g+1$ points to reconstruct f

$\Rightarrow n-(g+1)$ erasures can be corrected

$d_1 = n-g$

$d(C_1^\perp)$: $g+1$ points of C_1 are arbitrary, $g+2$ are constrained

$\Rightarrow d = g+2$

$d(C_2)$: $g+1-k$ arbitrary points in $C_2^\perp \Rightarrow d_2 = g+2-k$

$(C_1^\perp \subset C_2)$ $d_Q = \min(d_1, d_2) = \min(n-g, g+2-k) = \frac{n-k}{2}+1$

Choose g so that $n-g = g+2-k \Rightarrow (g+1) = \frac{n+k}{2}$

CSS code: $C_1 = \{(f(\alpha_0), \dots, f(\alpha_{n-1})) \mid f \text{ degree} \le g\}$

$C_2^\perp = \{(f(\alpha_0), \dots, f(\alpha_{n-1})) \mid f \text{ degree} \le g,\ c_j = 0\ (j<k)\}$

$d(C_1)$: Need $g+1$ points to reconstruct f

$\Rightarrow n-(g+1)$ erasures can be corrected

$d_1 = n-g$

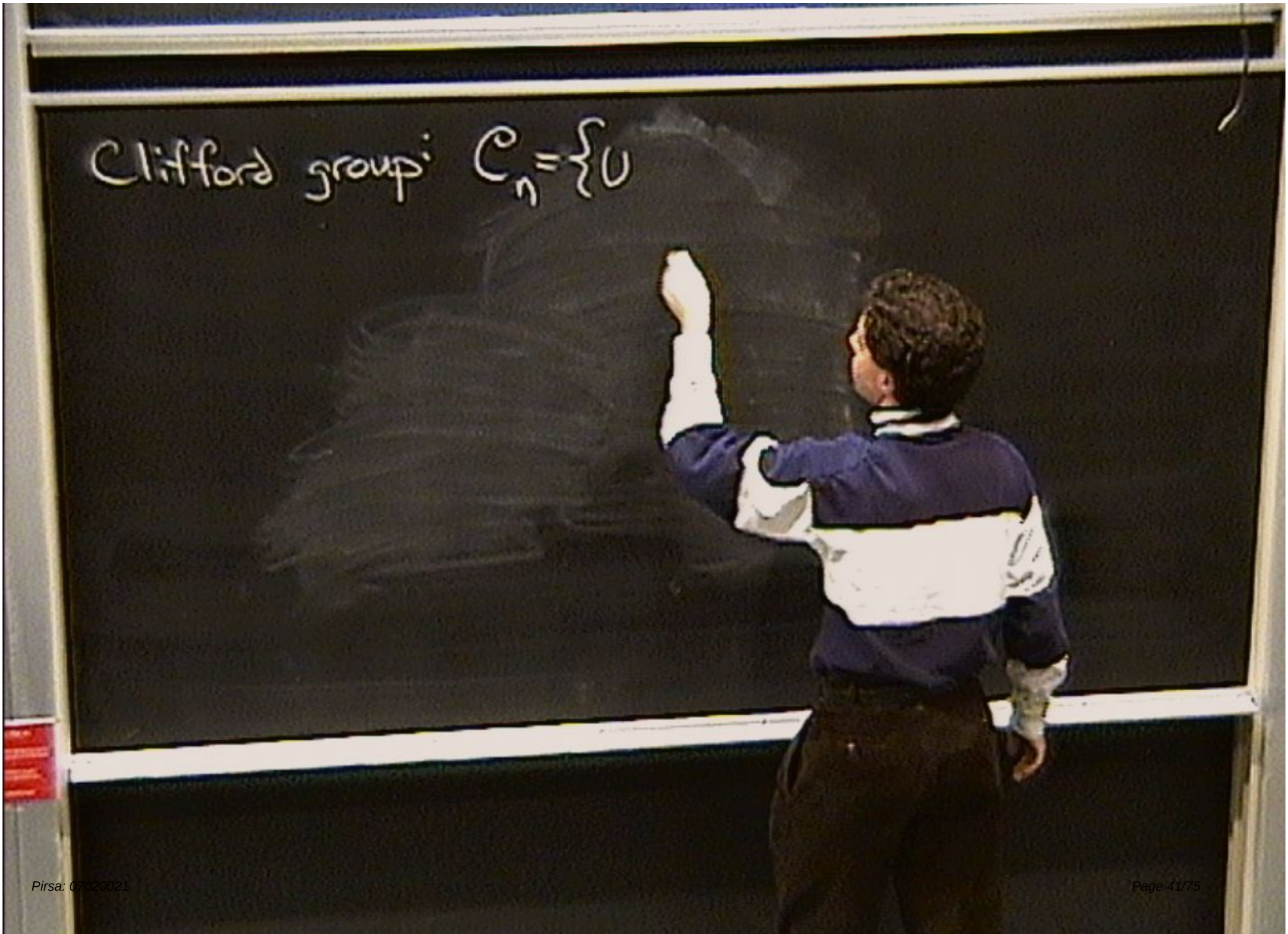
$d(C_1^\perp)$: $g+1$ points of C_1 are arbitrary, $g+2$ are constrained

$\Rightarrow d = g+2$

$d(C_2)$: $g+1-k$ arbitrary points in $C_2^\perp \Rightarrow d_2 = g+2-k$

$(C_1^\perp \subset C_2)$ $d_Q = \min(d_1, d_2) = \min(n-g, g+2-k) = \frac{n-k}{2}+1$ (Singleton bound $n-k \ge 2(d-1)$)

Choose g so that $n-g = g+2-k \Rightarrow (g+1) = \frac{n+k}{2}$

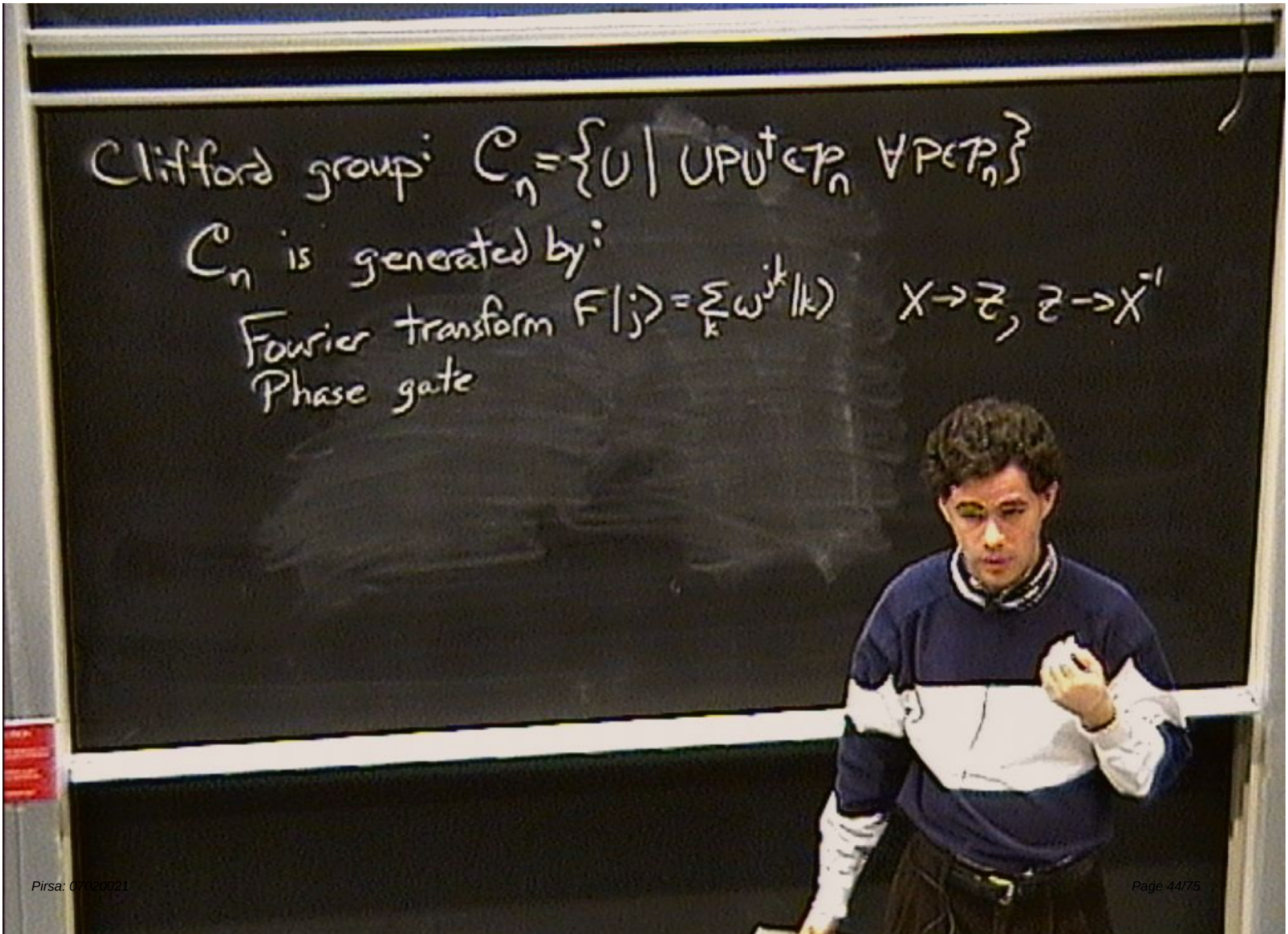

Clifford group: $\mathcal{C}_n = \{U$

Clifford group: $\mathcal{C}_n = \{U \mid UPU^\dagger \in \mathcal{P}_n \ \forall P \in \mathcal{P}_n\}$

Clifford group: $\mathcal{C}_n = \{U \mid UPU^\dagger \in \mathcal{P}_n \ \forall P \in \mathcal{P}_n\}$

$\mathcal{C}_n$ is generated by:

Fourier transform $F|j\rangle = \sum_k \omega^{jk} |k\rangle$


Clifford group: $\mathcal{C}_n = \{U \mid UPU^\dagger \in \mathcal{P}_n \ \forall P \in \mathcal{P}_n\}$
$\mathcal{C}_n$ is generated by:
Fourier transform $F|j\rangle = \sum_k \omega^{jk}|k\rangle$
$X \to Z$, $Z \to X^{-1}$
Phase gate

Clifford group: $\mathcal{C}_n = \{U \mid UPU^\dagger \in \mathcal{P}_n \ \forall P \in \mathcal{P}_n\}$

$\mathcal{C}_n$ is generated by:

Fourier transform $F|j\rangle = \sum_k \omega^{jk}|k\rangle$ $\quad X \to Z,\ Z \to X^{-1}$

Phase gate $R|j\rangle = \omega^{j(j-1)/2}|j\rangle$ $\quad X \to XZ,\ Z \to Z$

Clifford group: $\mathcal{C}_n = \{U \mid UPU^\dagger \in \mathcal{P}_n \ \forall P \in \mathcal{P}_n\}$

$\mathcal{C}_n$ is generated by:

Fourier transform: $F|j\rangle = \sum_k \omega^{jk}|k\rangle$ $\quad X \to Z,\ Z \to X^{-1}$

Phase gate: $R|j\rangle = \omega^{j(j-1)/2}|j\rangle$ $\quad X \to XZ,\ Z \to Z$

SUM: $\mathrm{SUM}|j\rangle|k\rangle = |j\rangle|k-j \pmod p\rangle$

$X \otimes I \to X \otimes X$

$I \otimes X \to I \otimes X$

$Z \otimes I \to Z \otimes I$

$I \otimes Z \to$

Clifford group: $\mathcal{C}_n = \{U \mid UPU^\dagger \in \mathcal{P}_n \ \forall P \in \mathcal{P}_n\}$

$\mathcal{C}_n$ is generated by:

Fourier transform: $F|j\rangle = \sum_k \omega^{jk}|k\rangle$ $\quad X \to Z,\ Z \to X^{-1}$

Phase gate: $R|j\rangle = \omega^{j(j-1)/2}|j\rangle$ $\quad X \to XZ,\ Z \to Z$

SUM: $\mathrm{SUM}|j\rangle|k\rangle = |j\rangle|k+j \pmod p\rangle$

$X \otimes I \to X \otimes X$

$I \otimes X \to I \otimes X$

$Z \otimes I \to Z \otimes I$

$I \otimes Z \to Z^{-1} \otimes Z$

Scalar multiplication:

Clifford group: $\mathcal{C}_n = \{U \mid UPU^\dagger \in \mathcal{P}_n \ \forall P \in \mathcal{P}_n\}$

$\mathcal{C}_n$ is generated by:

Fourier transform: $F|j\rangle = \sum_k \omega^{jk}|k\rangle$ $\quad X \to Z,\ Z \to X^{-1}$

Phase gate: $R|j\rangle = \omega^{j(j-1)/2}|j\rangle$ $\quad X \to XZ,\ Z \to Z$

SUM: $\mathrm{SUM}|j\rangle|k\rangle = |j\rangle|k-j \pmod p\rangle$

$X \otimes I \to X \otimes X$

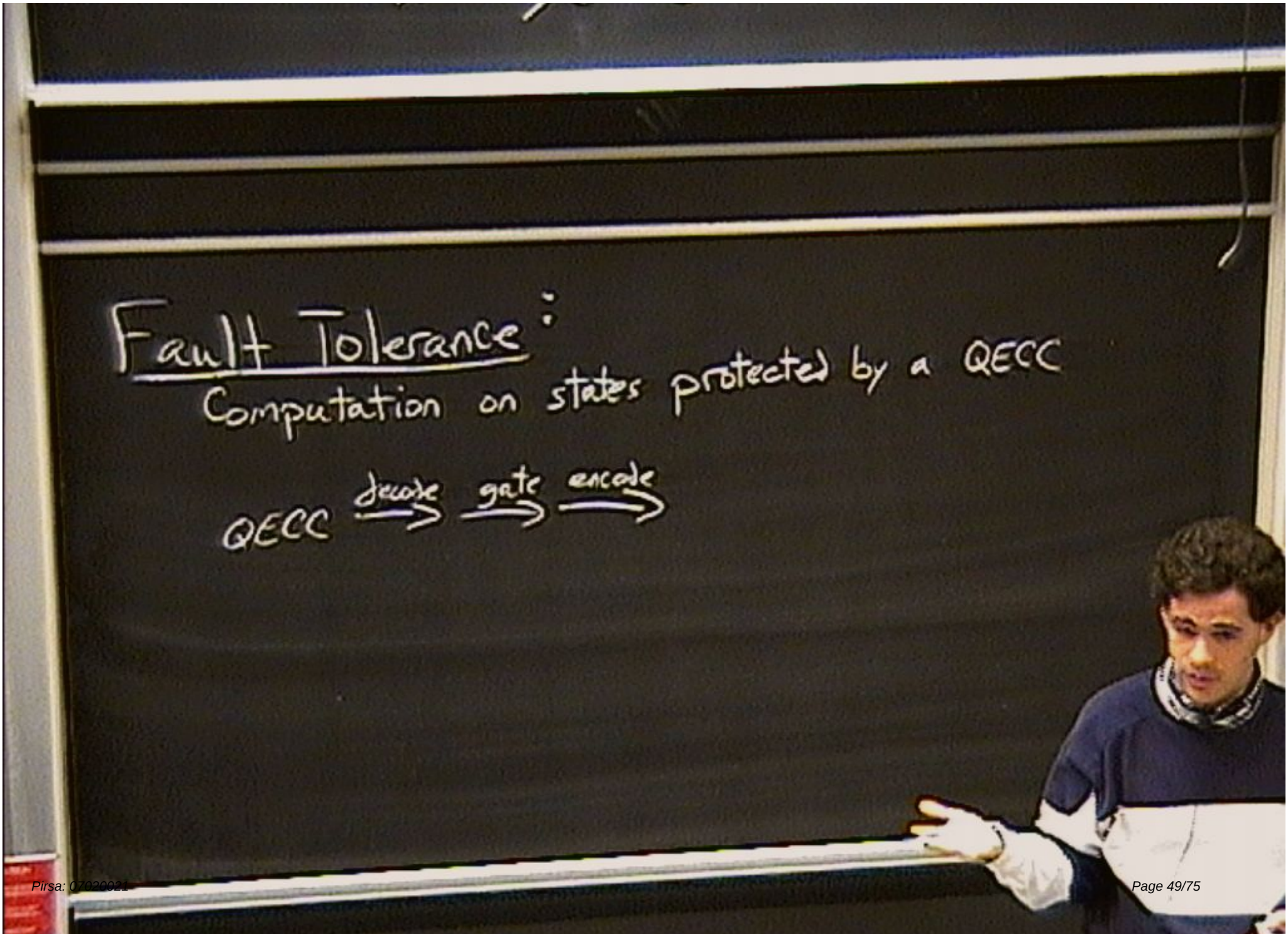
$I \otimes X \to I \otimes X$

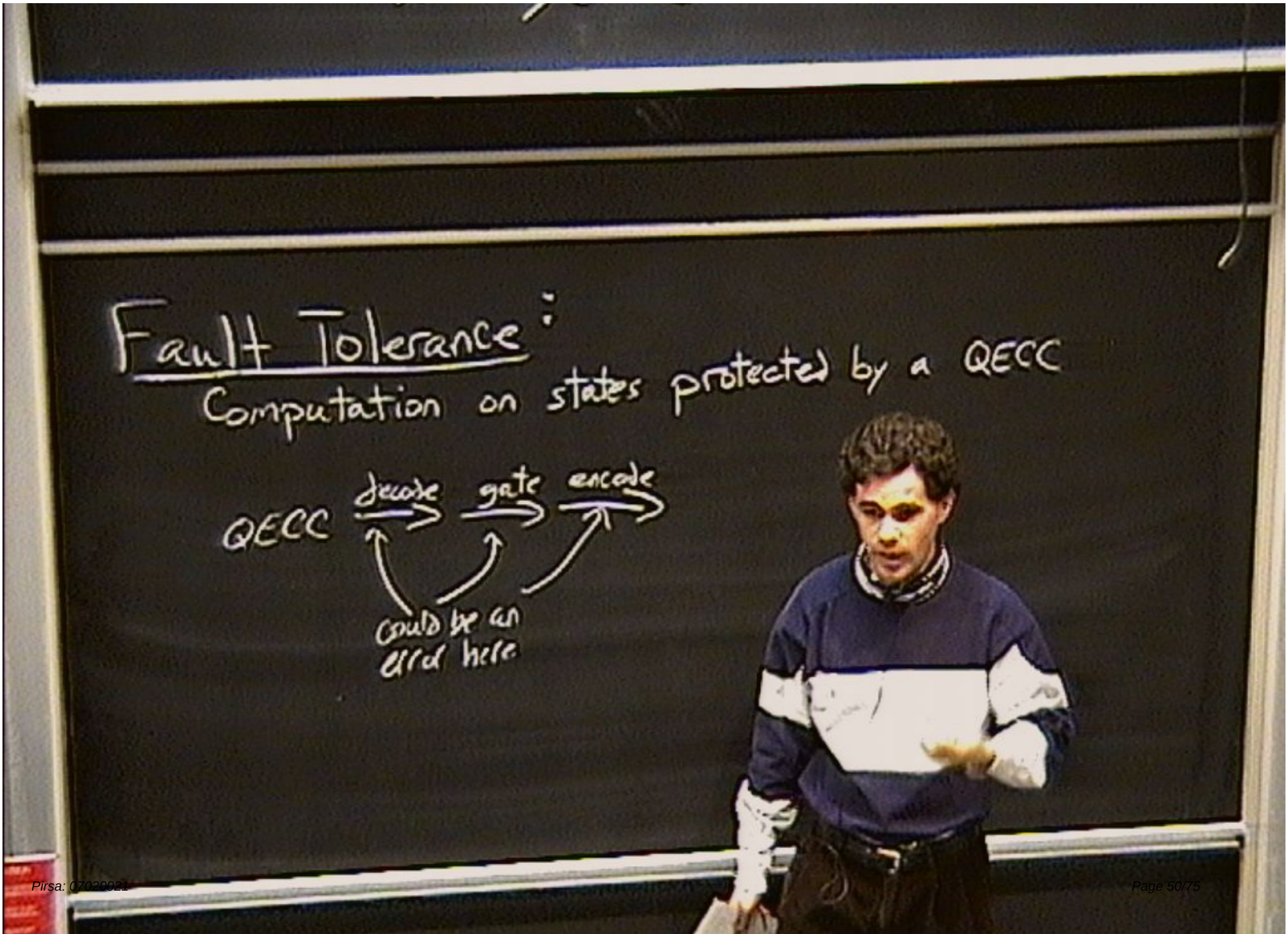
$Z \otimes I \to Z \otimes I$

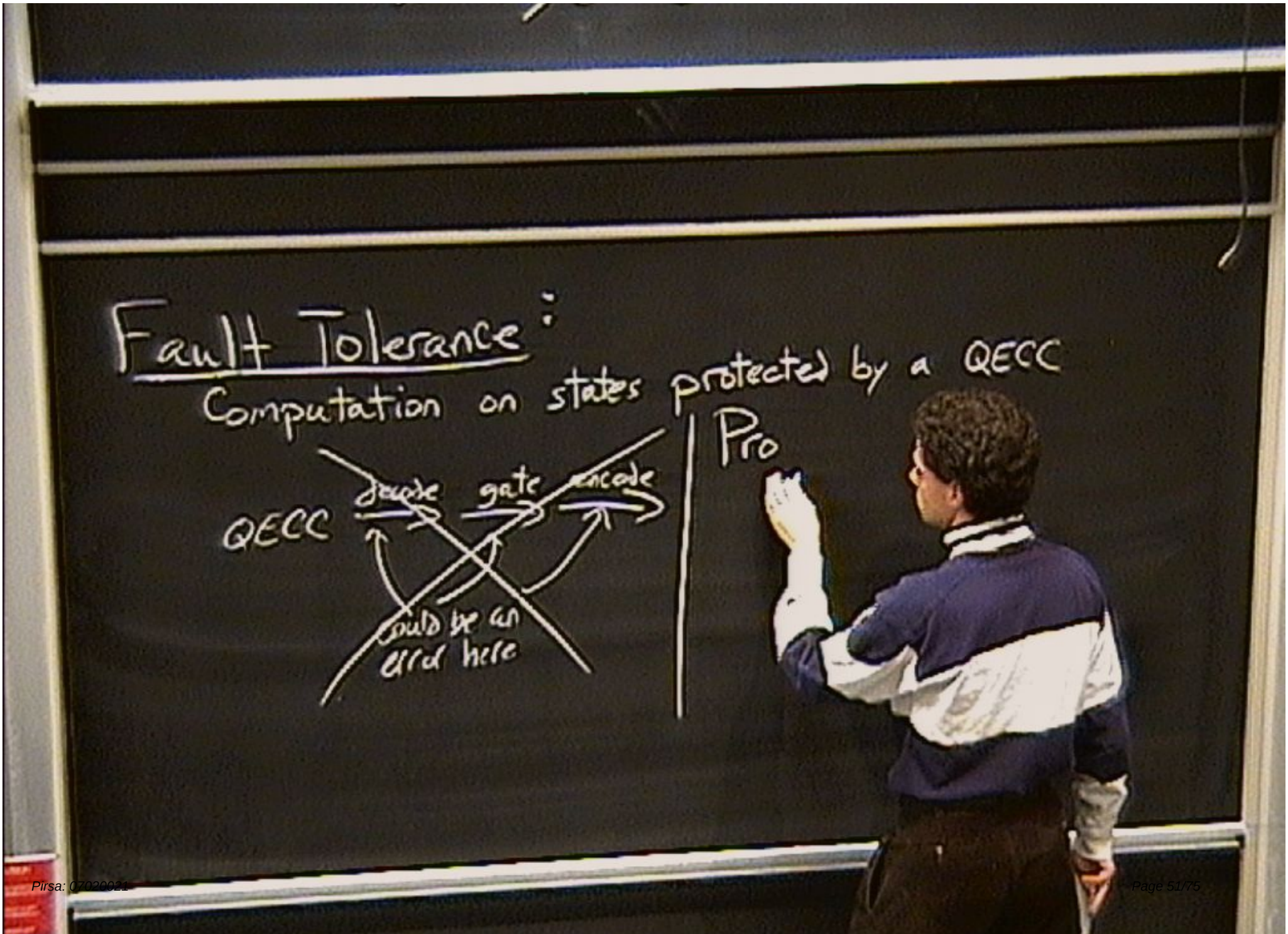
$I \otimes Z \to Z^{-1} \otimes Z$

Scalar multiplication: $S_c|j\rangle = |cj\rangle$ $(c \neq 0)$

$X \to X^c,\ Z \to Z^{1/c}$

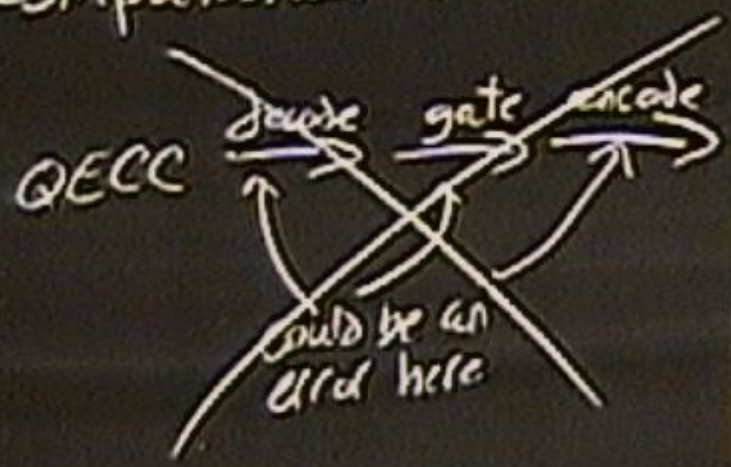

Fault Tolerance:
Computation on states protected by a QECC
QECC $\xrightarrow{\text{decode}}$ $\xrightarrow{\text{gate}}$ $\xrightarrow{\text{encode}}$


Fault Tolerance:
Computation on states protected by a QECC
QECC
decode
gate
encode
could be an error here


Fault Tolerance:
Computation on states protected by a QECC
QECC
decode
gate
encode
could be an error here
Pro

Fault Tolerance:

Computation on states protected by a QECC

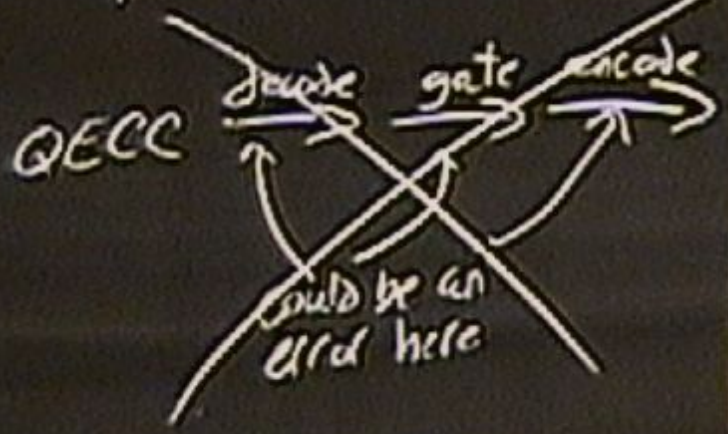


Propagation of errors:

$X_1|\bar{\psi}\rangle$

Fault Tolerance:

Computation on states protected by a QECC



Propagation of errors:

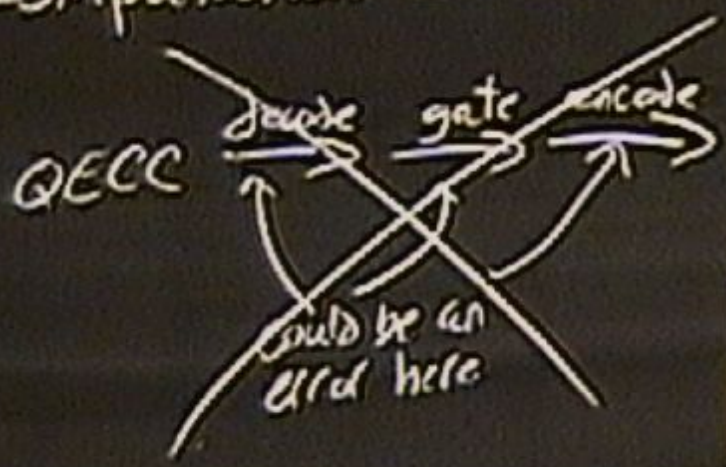
$X_i|\bar{\psi}\rangle \xrightarrow{CNOT(i,j)} (X_i \otimes X_j)\, CNOT\, |\bar{\psi}\rangle$

X propagates forward in CNOT

Z propagates backwards

Fault Tolerance:

Computation on states protected by a QECC

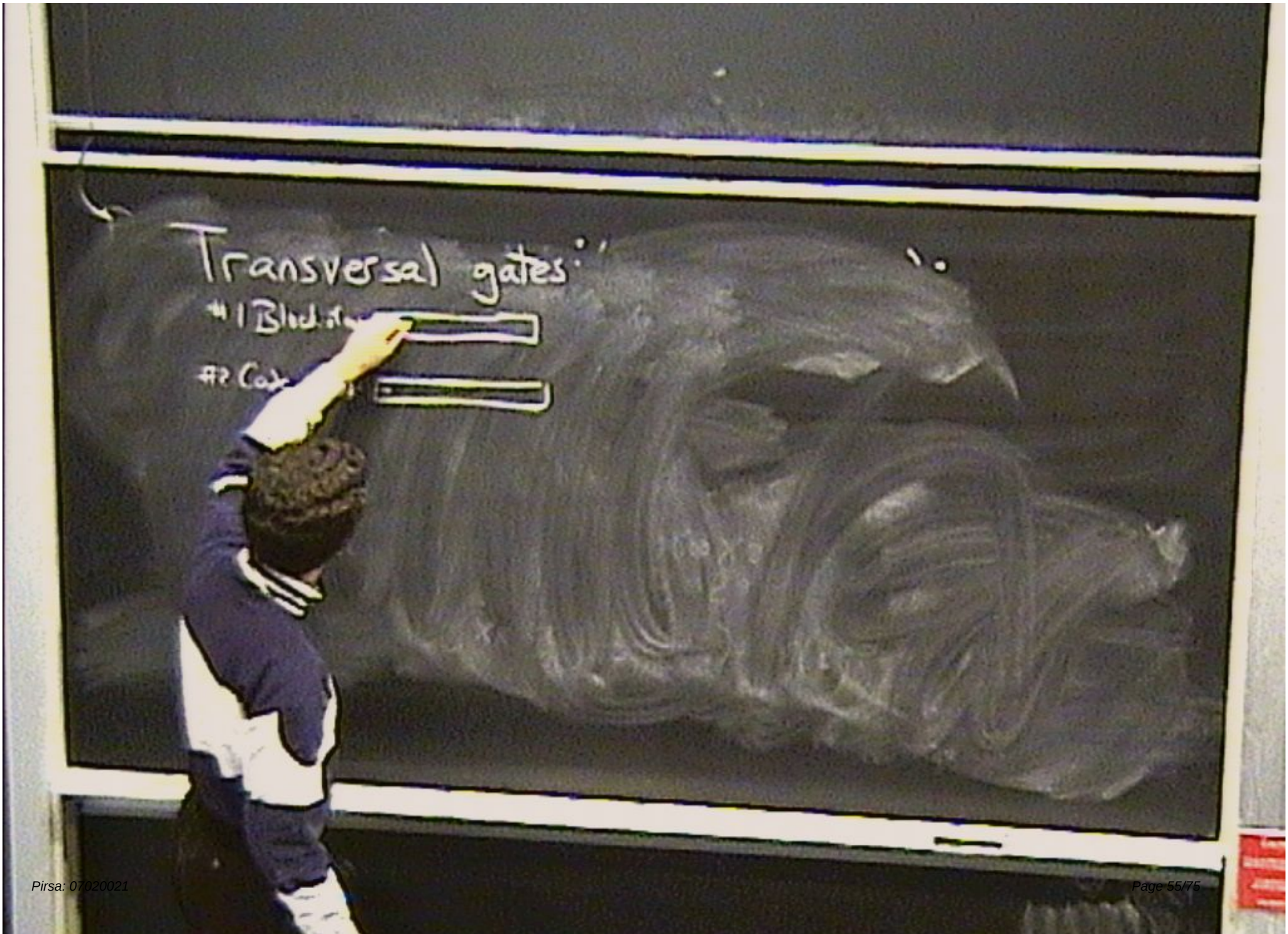


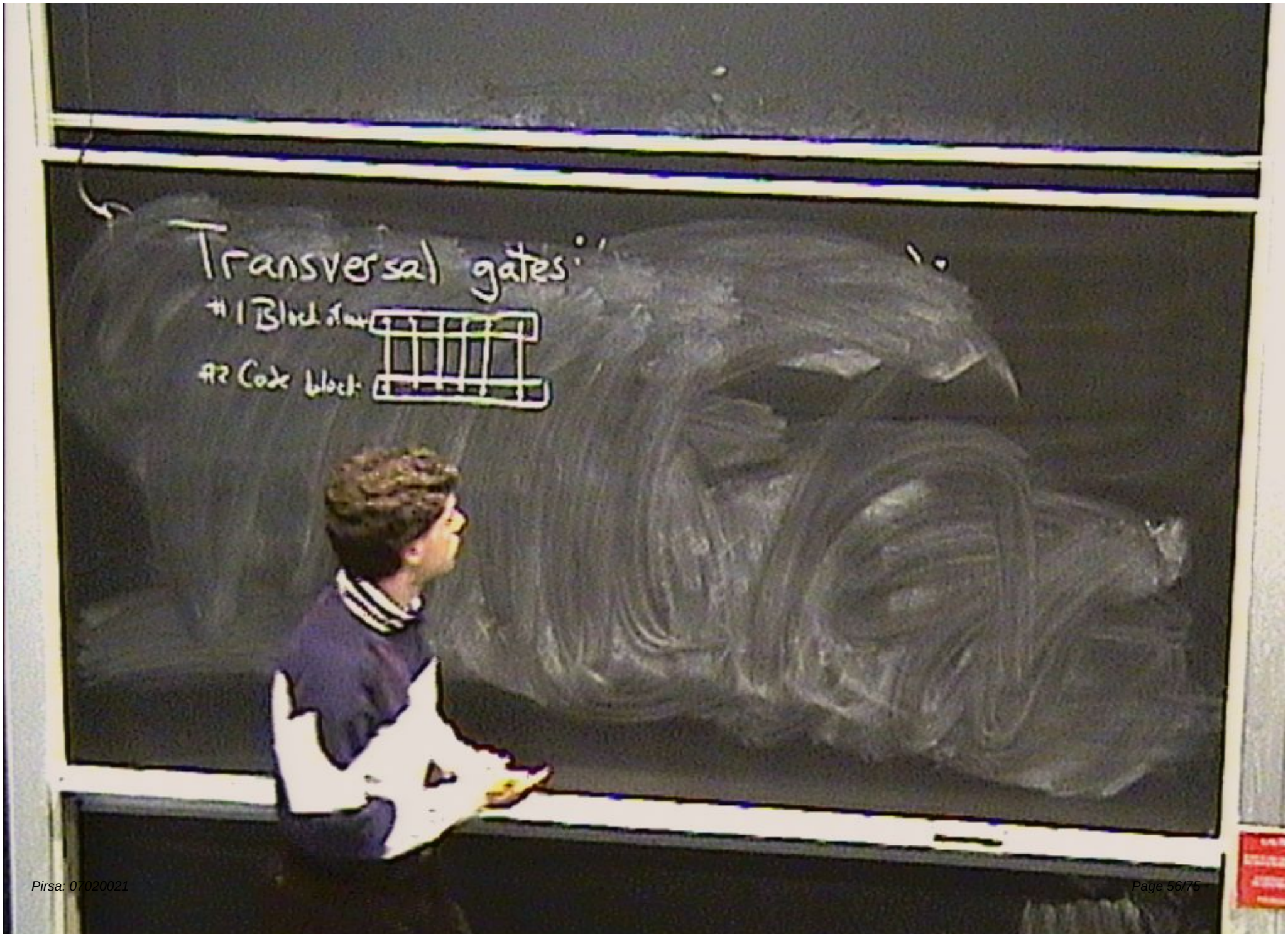
Propagation of errors:

$$X_i|\bar{\psi}\rangle \xrightarrow{CNOT(i,j)} (X_i \otimes X_j)\, CNOT|\bar{\psi}\rangle$$

X propagates forward in CNOT

Z propagates backwards


Transversal gates:

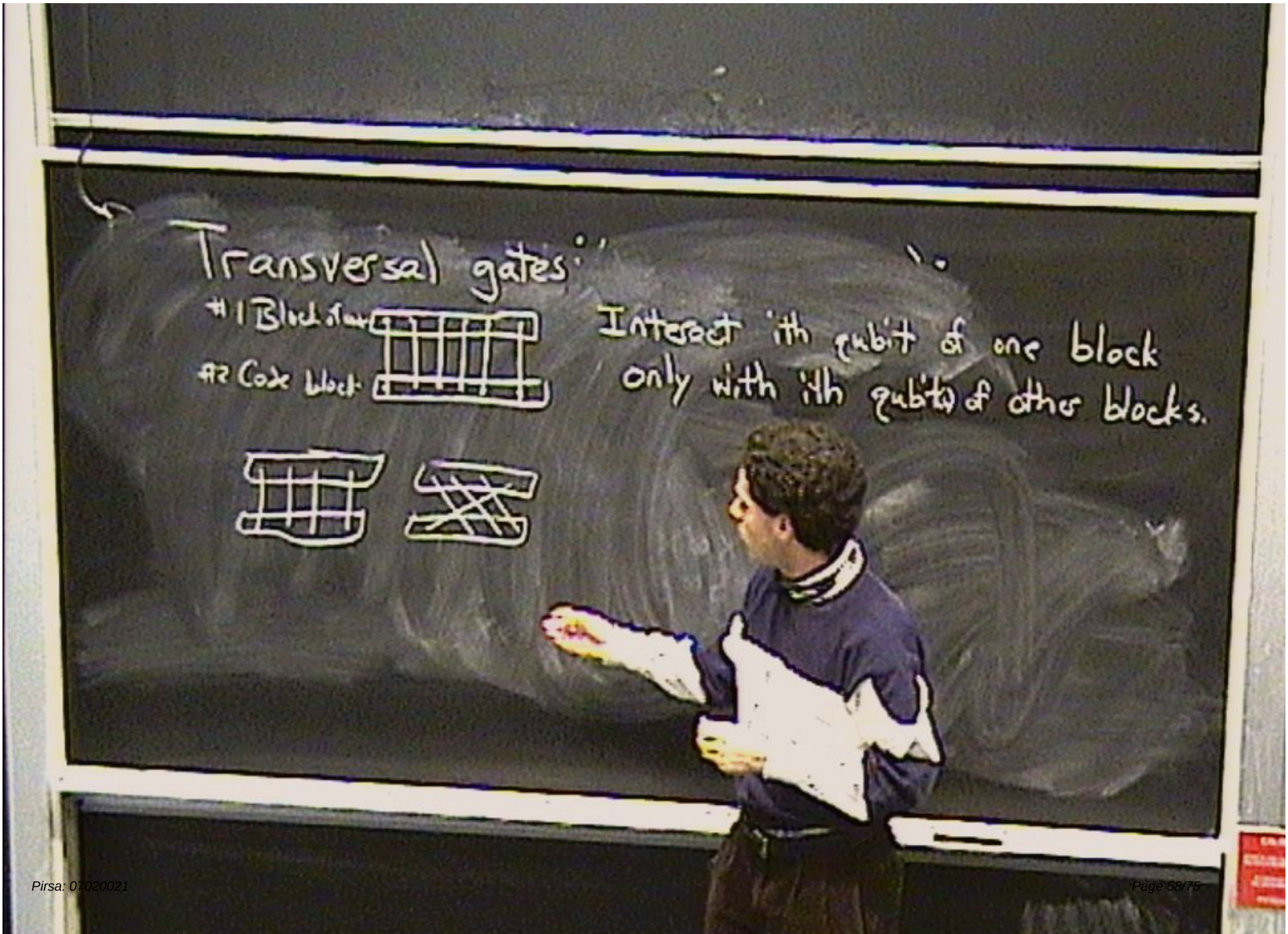

Transversal gates:
#1 Block of one
#2 Code block

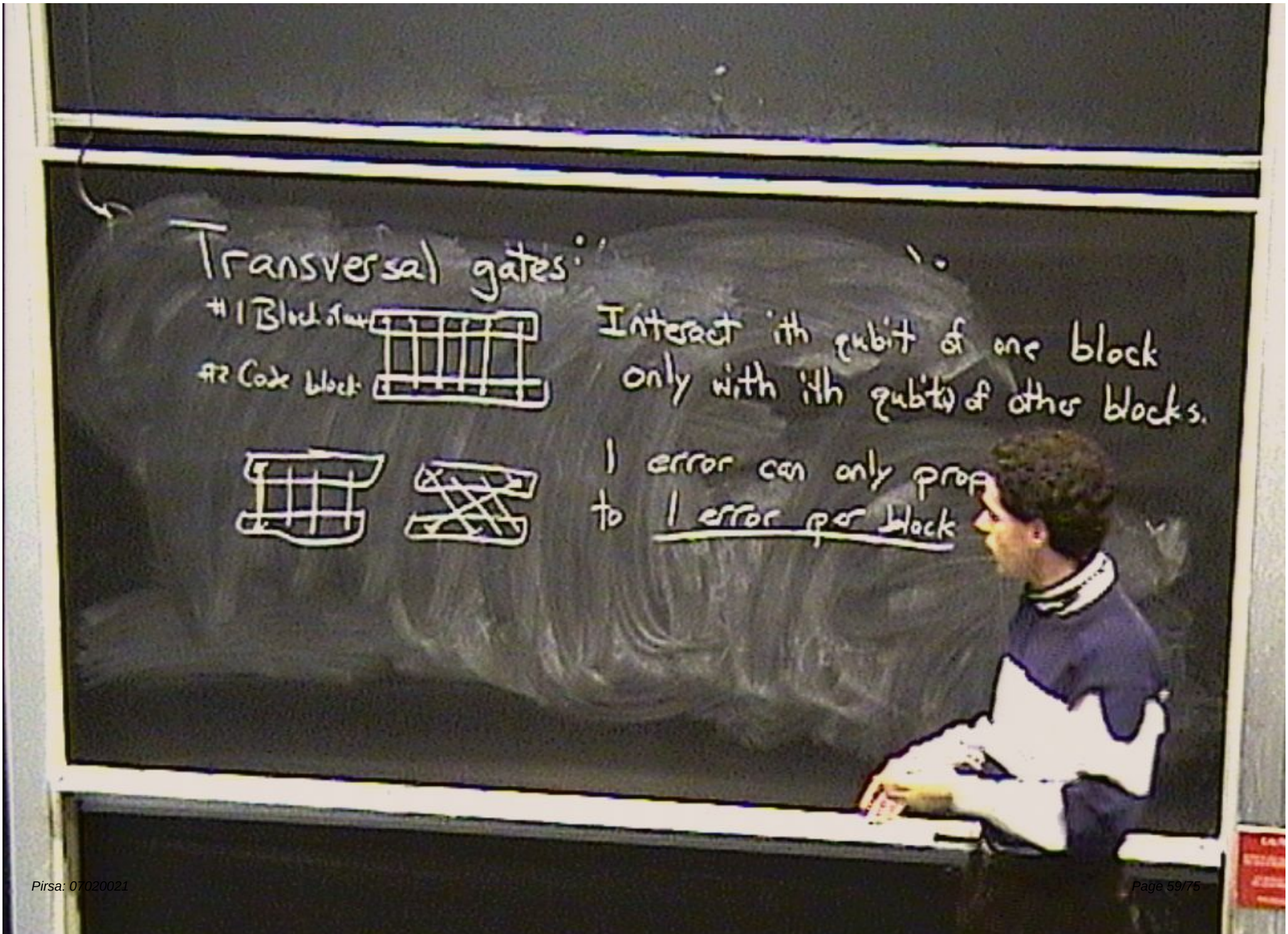
Transversal gates:

#1 Block of code

#2 Code block

Interact ith qubit of one block only with ith qubit(s) of other blocks.


Transversal gates:
#1 Block
#2 Code block
Interact ith qubit of one block
only with ith qubits of other blocks.


Transversal gates:
#1 Block of
#2 Code block
Interact ith qubit of one block
only with ith qubit of other blocks.
1 error can only prop
to 1 error per block

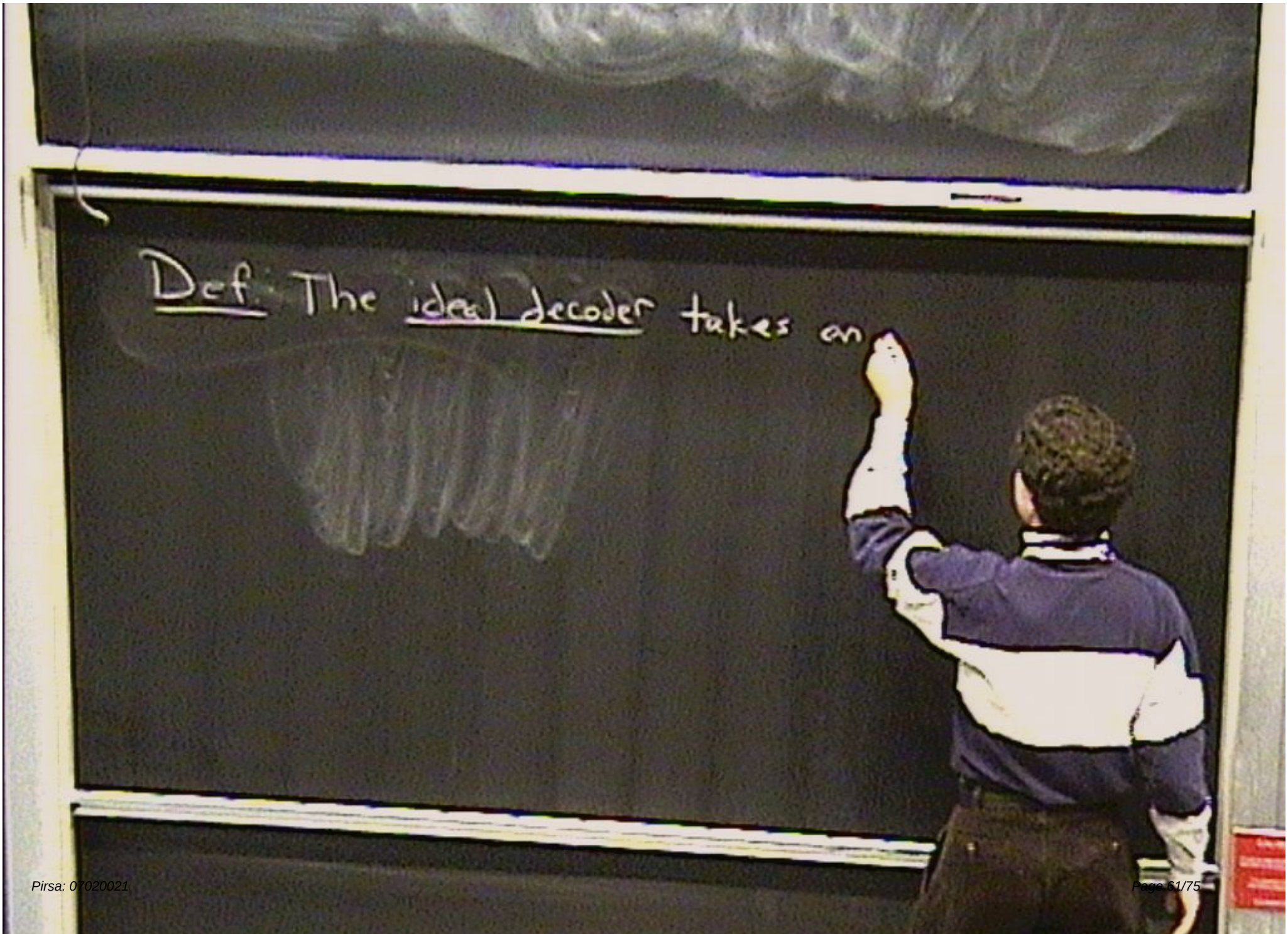
Transversal gates:

#1 Block of one
#2 Code block

Interact ith qubit of one block only with ith qubits of other blocks.

1 error can only propagate to <u>1 error per block</u>

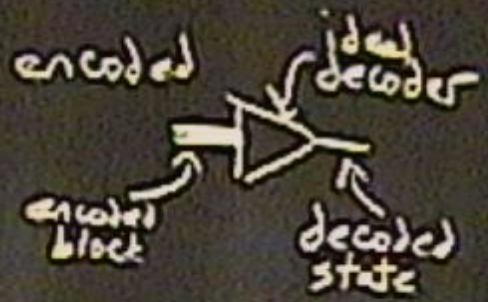
(even in a sequence of transversal gates)


Def: The ideal decoder takes an

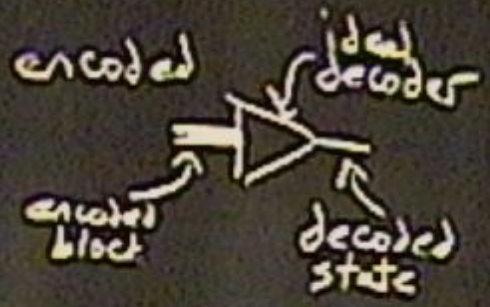
Def: The <u>ideal decoder</u> takes an encoded state, corrects any errors, & decodes.

Def: The <u>ideal decoder</u> takes an encoded state, corrects any errors, & decodes.

Def: The <u>ideal decoder</u> takes an encoded state, corrects any errors, & decodes.



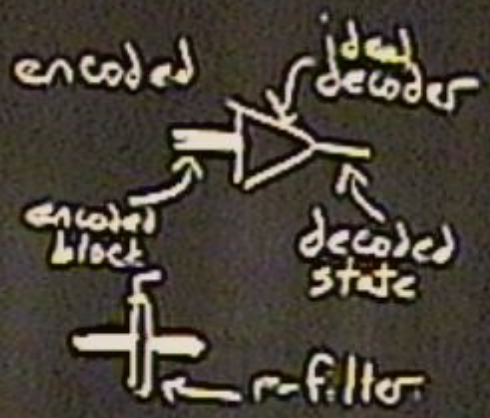
Def: The <u>ideal decoder</u> takes an encoded state, corrects any errors, & decodes.



Def: Let the <u>r-filter</u> be the projector on subspace spanned by codewords w/ up to r errors.

Def: The <u>ideal decoder</u> takes an encoded state, corrects any errors, & decodes.

Def: Let the <u>r-filter</u> be the projector on subspace spanned by codewords w/ up to r errors.



CSS code: $C_1 = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \leq e\}$

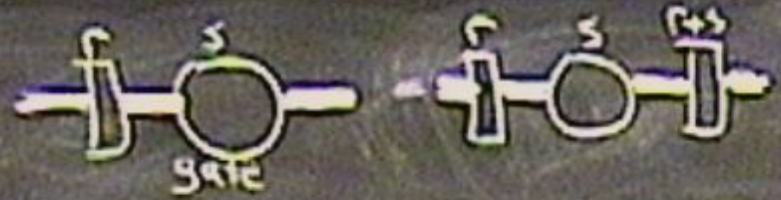
Gate property 1:

CSS code: $C_1 = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree } \leq r\}$

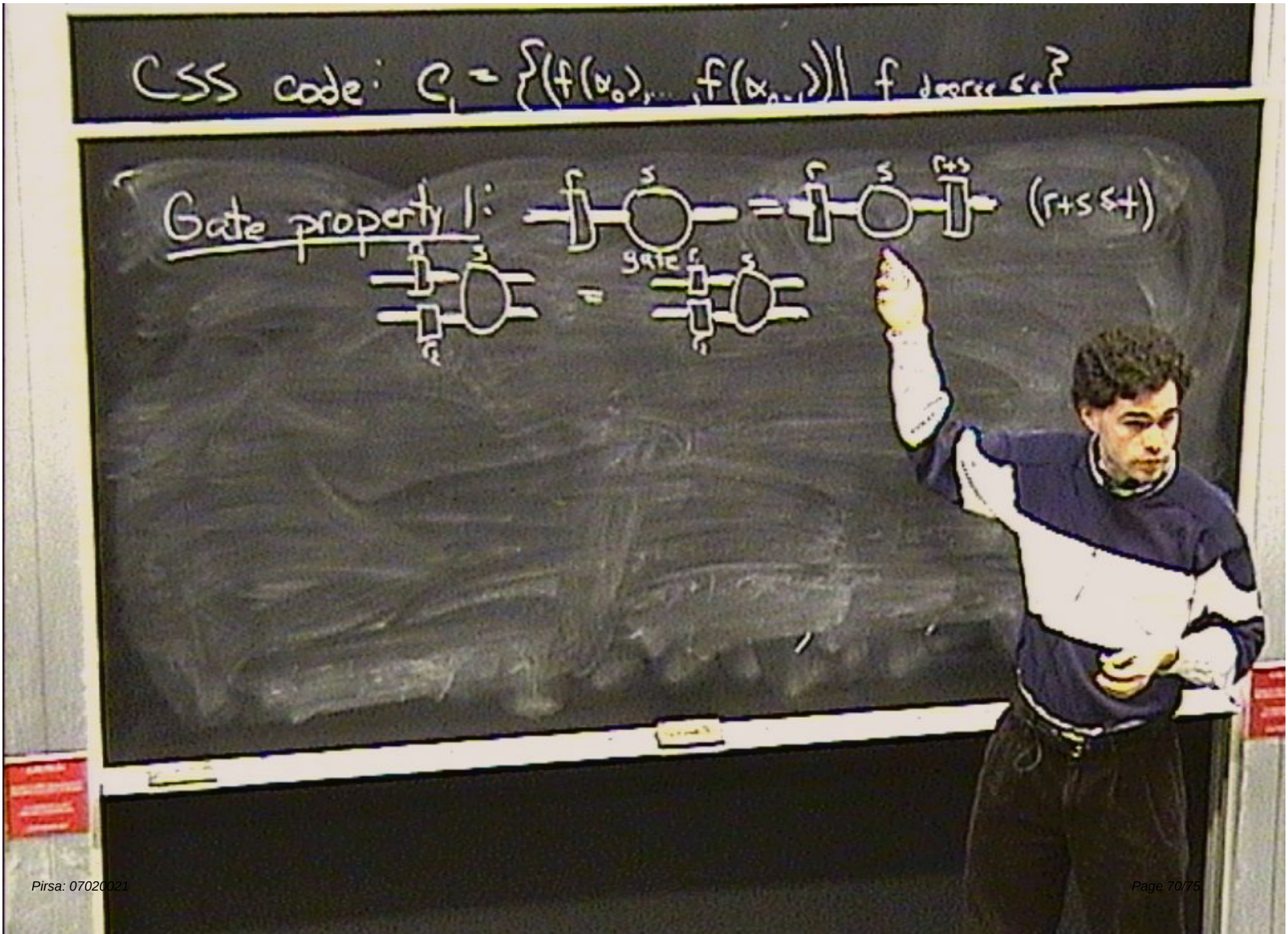
Gate property 1:

$(r+$

CSS code: $C_1 = \{(f(x_0), \ldots, f(x_{n-1})) \mid f \text{ degree} \leq t\}$

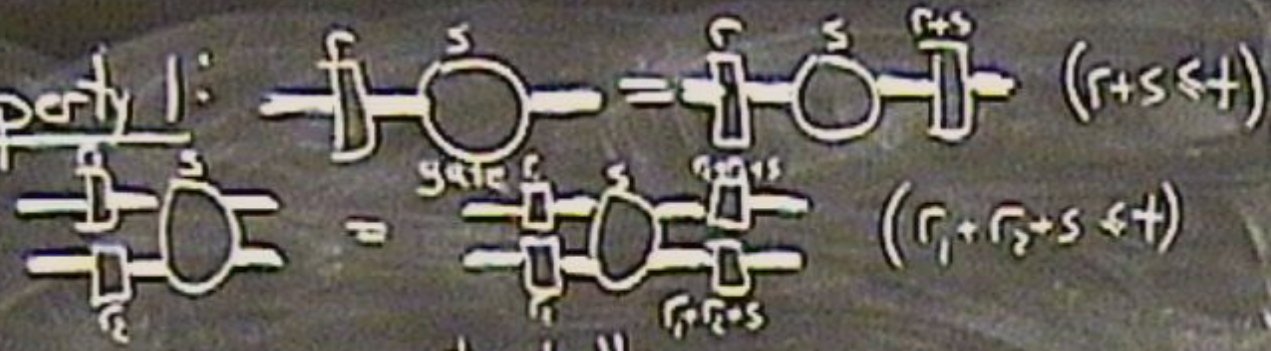
<u>Gate property 1:</u> 

$(r+s \leq t)$


CSS code: $C_1 = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \leq t\}$
Gate property 1:
(r+s≤t)
gate

CSS code: $C_1 = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \leq t\}$

Gate property 1:



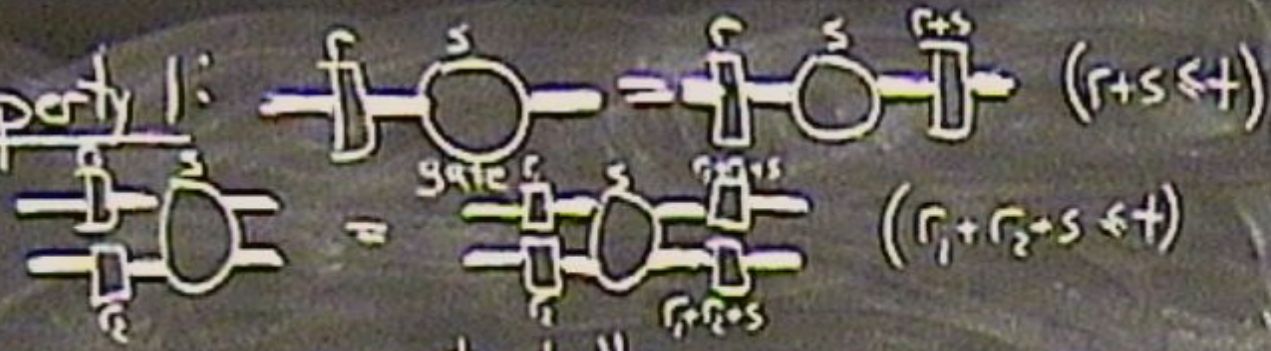
$(r+s \leq t)$

$(r_1 + r_2 + s \leq t)$

Errors do not propagate badly.

CSS code: $C_1 = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \text{ degree} \leq t\}$

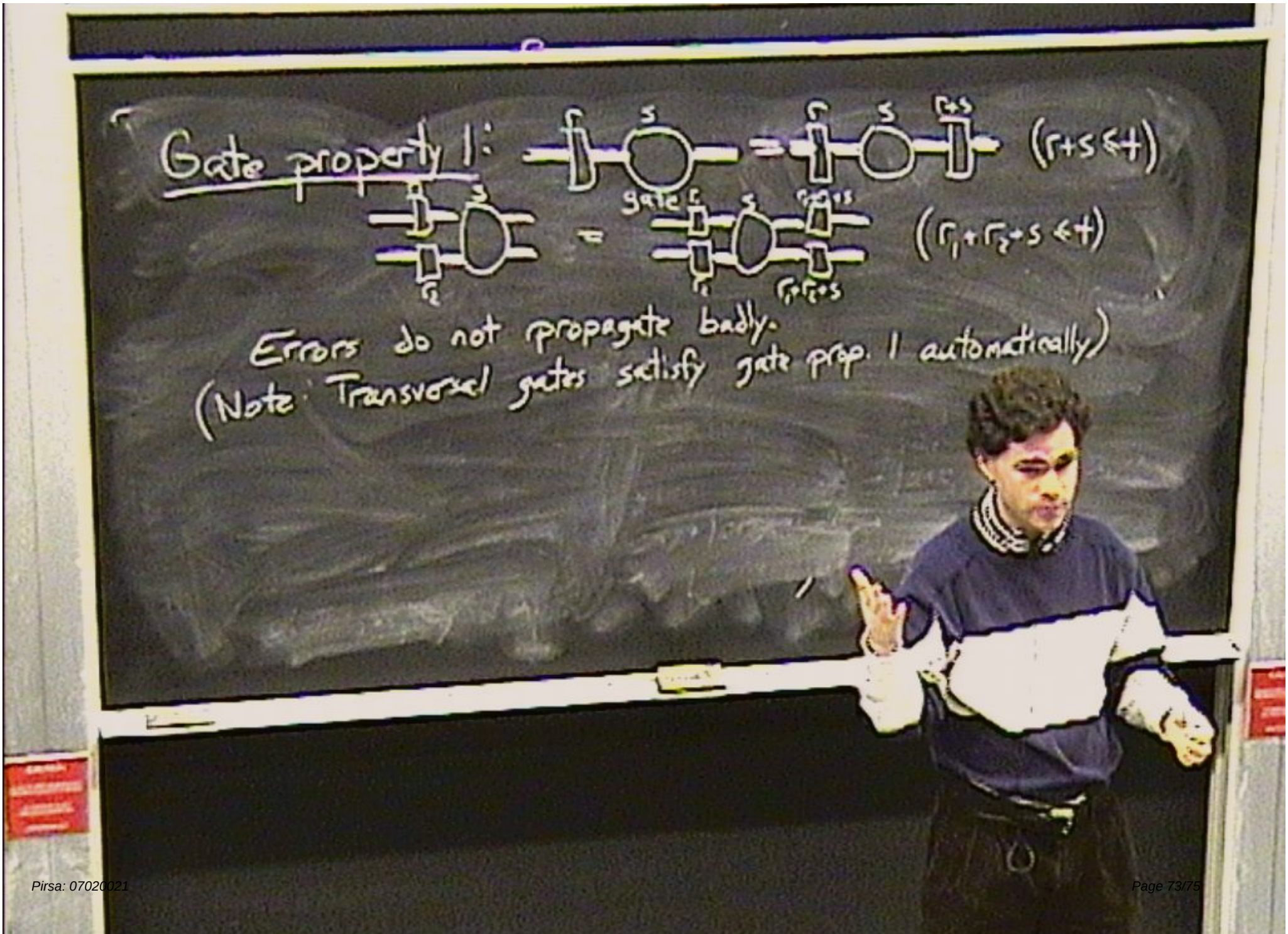
Gate property 1:

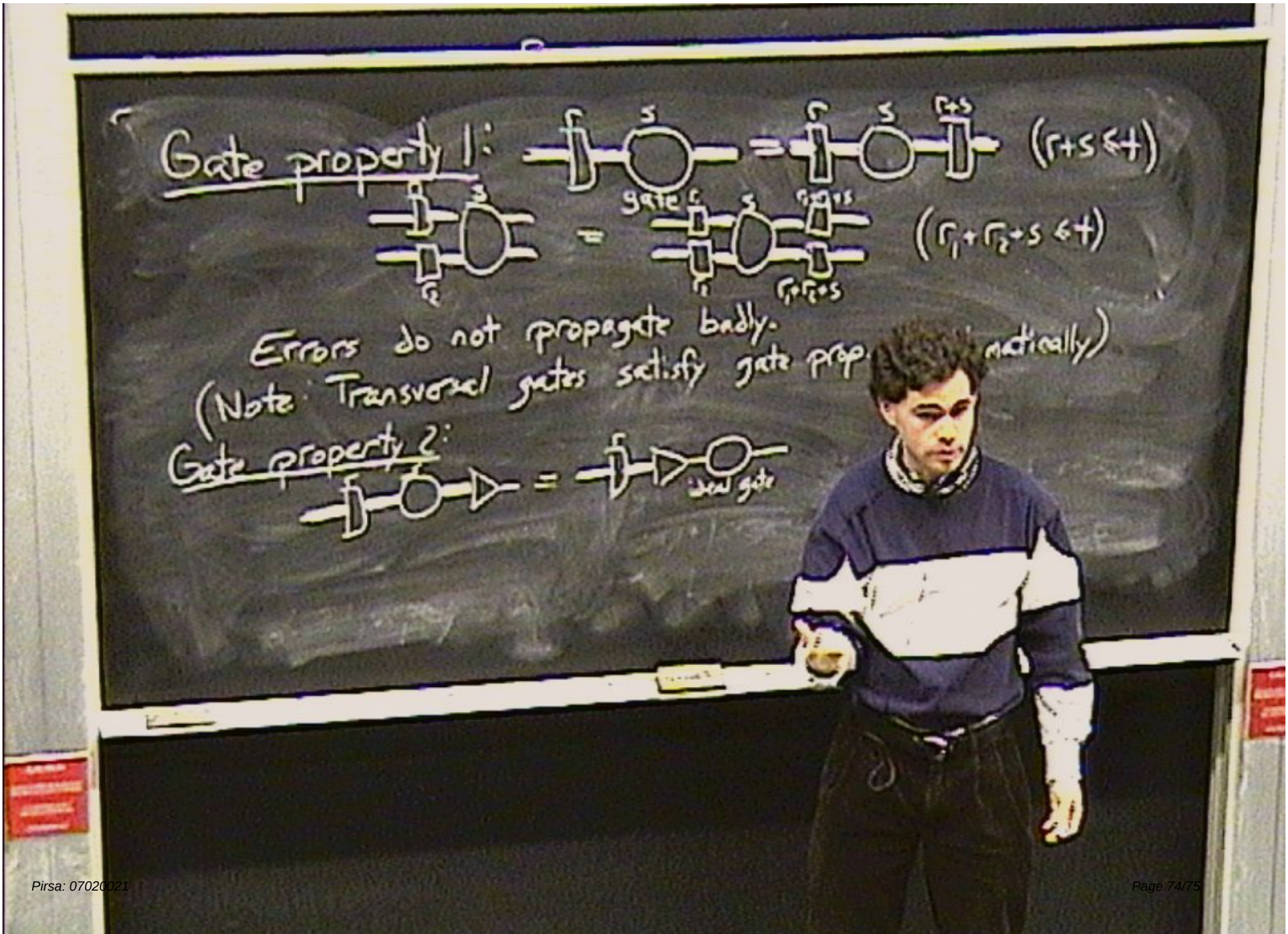


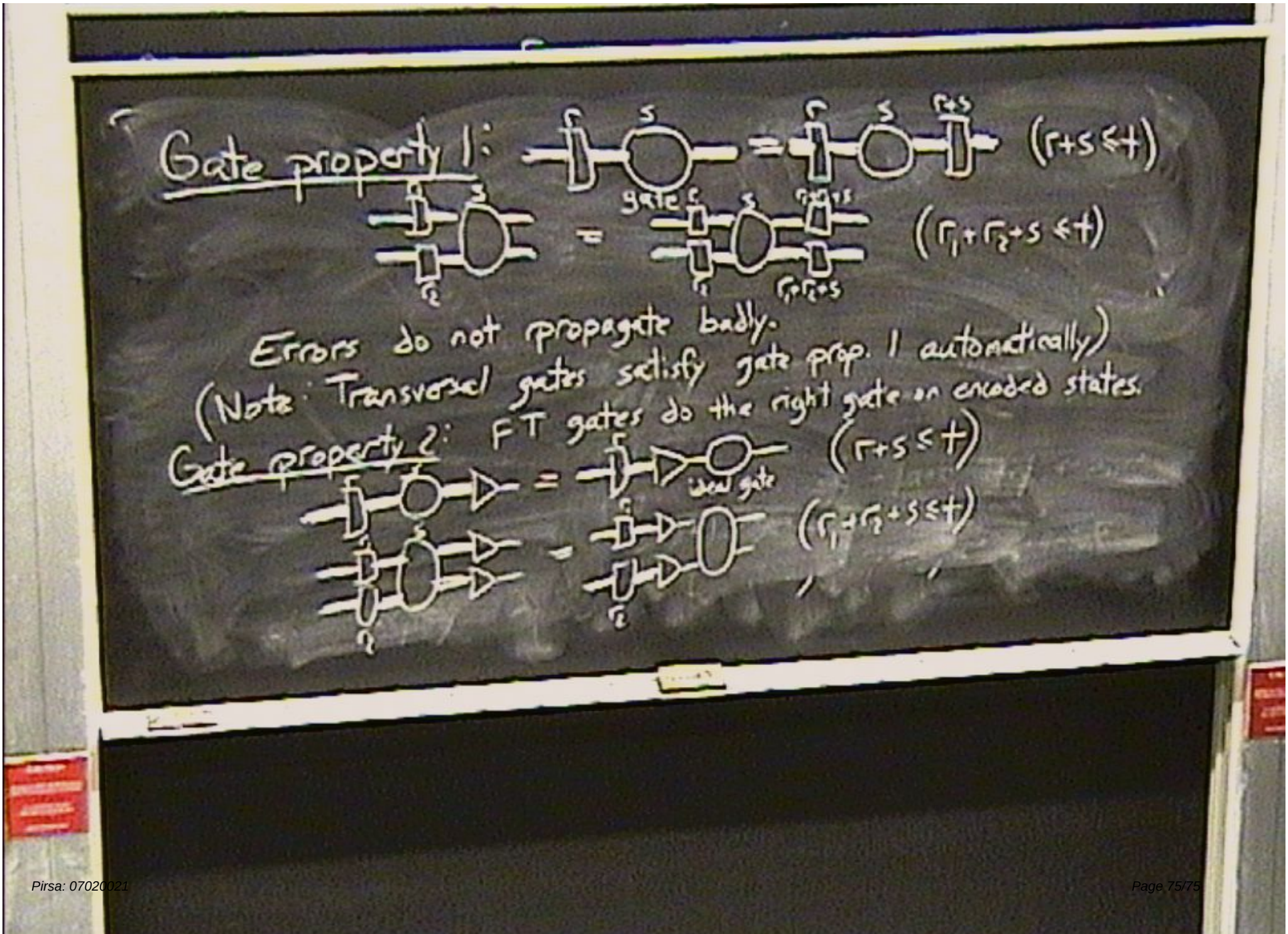
$(r+s \leq t)$

$(r_1 + r_2 + s \leq t)$

Errors do not propagate badly.


Gate property 1:
(r+s ≤ t)
gate
(r1+r2+s ≤ t)
Errors do not propagate badly.
(Note: Transversal gates satisfy gate prop. 1 automatically)


Gate property 1:
(r+s ≤ t)
gate
(r1+r2+s ≤ t)
Errors do not propagate badly.
(Note: Transversal gates satisfy gate prop
nationally)
Gate property 2:
seal gate


Gate property 1:
(r+s≤t)
gate
(r1+r2+s≤t)
Errors do not propagate badly.
(Note: Transversal gates satisfy gate prop. 1 automatically)
Gate property 2: FT gates do the right gate on encoded states.
(r+s≤t)
ideal gate
(r1+r2+s≤t)