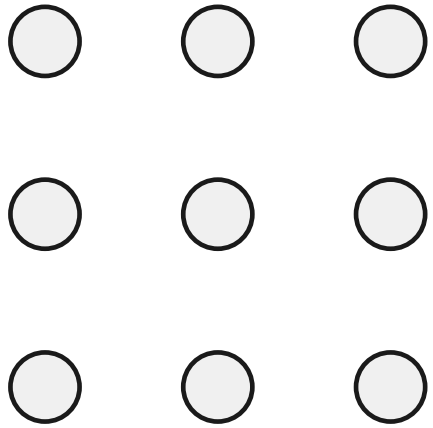
SAJID Lab
Smart & Advanced Junction of Intelligent Devices

EEE 6516 · QUANTUM COMPUTING

Stabilisers, the Surface Code and Fault Tolerance

A language for codes, a code for chips, and the threshold

Chapter 8 · Lecture 18



Sajid Muhaimin Choudhury

Copyright © 2026 Sajid Muhaimin Choudhury. All rights reserved.

Today

1. **Recall**
2. **Stabiliser codes**
3. **Steane and five-qubit codes**
4. **Surface code**
5. **Fault tolerance**
6. **Experiments**

By the end of this lecture you can

- identify stabiliser generators, syndromes, logical operators and [[n, k, d]]
- build the Steane code from the Hamming code
- describe the rotated surface code: qubits, checks, distance, logical strings
- state the threshold theorem and use the scaling model to estimate distance and overhead

YOUR TURN · RECALL

Recall: what leaves the code state alone?

Recall · 45 seconds

Which of these leave every state $\alpha|000\rangle + \beta|111\rangle$ unchanged?

A Z_0Z_1

B Z_0Z_2

C $X_0X_1X_2$

D Z_0

ANSWER

A and B fix it; C and D change it

Z_0Z_1, Z_0Z_2: +1 on both $|000\rangle$ and $|111\rangle$ (so is their product Z_1Z_2).

$X_0X_1X_2$ swaps $|000\rangle \leftrightarrow |111\rangle$; Z_0 flips the sign of β.

C and D map code states to code states, but different ones: logical operators.

1 Stabiliser codes

Can we describe a code by its checks alone?

DERIVATION

Pauli products commute or anticommute

The n-qubit Pauli group: products of I, X, Y, Z, times ±1 or ±i.

$XZ = -ZX, \qquad XY = -YX, \qquad YZ = -ZY$

two different Paulis anticommute

$P = P_0 \otimes P_1 \otimes \cdots \otimes P_{n-1}$

a Pauli product; weight = number of non-I factors

$PQ = (-1)^m QP$

m = positions where the factors anticommute

$(Z_0Z_1)(X_0X_1) = (-1)^2 (X_0X_1)(Z_0Z_1)$

two positions: they commute

Even number of clashes: they commute. Odd number: they anticommute.

A code defined by its checks

Stabiliser code

$$S = \langle g_1, \ldots, g_m \rangle$$

commuting Pauli products,
closed under products, −I not included

$$g|\psi\rangle = |\psi\rangle \text{ for all } g \in S$$

$$k = n - m$$

Stabiliser group S: commuting Pauli products, closed under products, without −I.

Code space: the states with g|ψ⟩ = |ψ⟩ for every g in S.

m independent generators on n qubits leave $k = n - m$ logical qubits.

Bit-flip code: $S = \langle Z_0Z_1, Z_1Z_2 \rangle$, n = 3, m = 2, k = 1.

DERIVATION

Syndromes are anticommutation patterns

$g_i E|\psi\rangle = \pm E g_i|\psi\rangle = \pm E|\psi\rangle$ — move the generator past the error

$s_i = 1 \;\Leftrightarrow\; E$ anticommutes with g_i — minus sign: odd parity

X_1: clashes with Z_0Z_1 and $Z_1Z_2 \;\rightarrow\; 11$ — Lecture 17's table, by counting clashes

The syndrome follows from commutation alone, without writing a single state vector.

Logical operators and distance

Logical operator

commutes with every element of S,
but is not in S: invisible, yet harmful

Distance d

smallest weight of a logical operator

$$[[n, k, d]]: \quad t = \lfloor (d-1)/2 \rfloor$$

In S: acts as the identity on the code (harmless).

Commutes with S, not in S: a logical operator $\overline{X}$, $\overline{Z}$: trivial syndrome, wrong state.

Distance d: the lightest logical operator.

Corrects t = ⌊(d − 1)/2⌋ errors; detects up to d − 1.

Worked example: the bit-flip code as a stabiliser code

error on qubit	0	1	2
X	10	11	01
Y	10	11	01
Z	00	00	00

syndrome (s_0, s_1) from Z_0Z_1, Z_1Z_2

$S = \langle Z_0Z_1, Z_1Z_2 \rangle$: n = 3, m = 2, k = 1.

$\overline{X} = X_0X_1X_2$; $\overline{Z} = Z_0$ (or Z_1, Z_2).

Z errors are invisible because they are logical operators.

Weight-1 logical: as a code for arbitrary errors it is $[[3,1,1]]$.

2 The Steane and five-qubit codes

Can a classical code give us a quantum one?

From Hamming to Steane

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

row of H	Z-type	X-type
0001111	IIIZZZZ	IIIXXXX
0110011	IZZIIZZ	IXXIIXX
1010101	ZIZIZIZ	XIXIXIX

column j of H = binary form of j + 1

Each row of H gives a Z-type generator ($1 \to Z$) and an X-type generator ($1 \to X$).

Z-type checks find X errors exactly as the classical code finds flips; X-type find Z errors.

Every pair of rows shares an even number of 1s: all six generators commute.

n = 7, m = 6, k = 1; $\overline{X} = X^{\otimes 7}$, $\overline{Z} = Z^{\otimes 7}$; d = 3: the $[[7,1,3]]$ code.

YOUR TURN · CALCULATE

Your turn: syndromes in the Steane code

Think · 1 minute

Using the table, find the Z-check syndrome of X_4 and the X-check syndrome of Z_4.

Which position does each syndrome point to?

ANSWER

Both give 101: binary 5, position 4

X_4 clashes with $IIIZZZZ$ and $ZIZIZIZ$: Z-check syndrome 101.

101 = 5, pointing to position 5 − 1 = 4; likewise Z_4 gives X-check syndrome 101.

Every X_j, Z_j has its own syndrome; Y_j triggers both halves: d ≥ 3.

The syndrome is the Hamming address of the faulty qubit, once for X and once for Z.

The smallest code, and when a code works

Five-qubit code [[5,1,3]]

$$XZZXI, IXZZX, XIXZZ, ZXIXZ$$

Knill–Laflamme condition

$$PE_a^\dagger E_b P = c_{ab} P$$

Four generators: $2^4 = 16$ syndromes = no error + 3×5 single-qubit Paulis, each distinct.

No smaller code corrects an arbitrary error on one qubit.

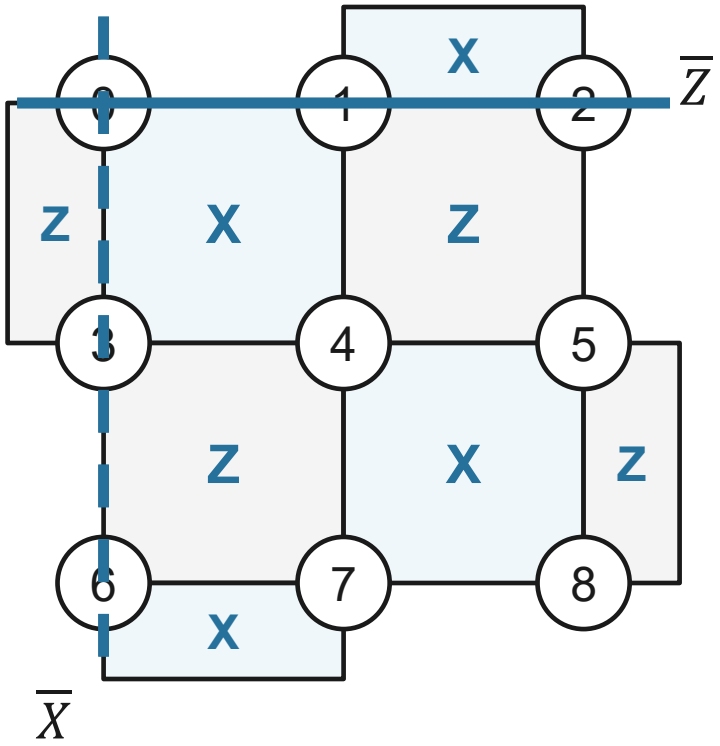
Knill–Laflamme: errors $\{E_a\}$ are correctable if and only if $PE_a^\dagger E_b P = c_{ab} P$.

Errors may move the state, but must not reveal the encoded information.

3 The surface code

Which code fits a chip with nearest-neighbour couplings?

The rotated surface code, distance 3



9 data qubits; 8 checks, each with its own measure qubit.

Weight-4 checks form a chessboard; weight-2 checks close the edges: X top and bottom, Z left and right.

$\overline{Z} = Z_0 Z_1 Z_2$: a row, left edge to right edge.

$\overline{X} = X_0 X_3 X_6$: a column, top to bottom; they cross once and anticommute.

Counting qubits

d	qubits	data	measure	detectors
3	17	9	8	24
5	49	25	24	72
7	97	49	48	144

Simulation 8.5 (Stim surface_code:rotated_memory_z, 3 rounds)

d² data + (d − 1)² + 2(d − 1) checks: $n_{total} = 2d^2 - 1$.

Stim's generated circuits use 17, 49, 97 qubits for d = 3, 5, 7.

Detectors at d = 3: 4 + 8 + 8 + 4 = 24 for 3 rounds.

COMMON MISTAKE

“A distance-d surface code has d² qubits”

✗ Count the grid

$$d = 5\text{:}\;\; 5^2 = 25 \text{ qubits}$$

counts only the data qubits

Forgets the measure qubits.

✓ Data plus measure qubits

$$2 \cdot 5^2 - 1 = 49 \text{ qubits}$$

each check needs its own measure qubit

d² data + d² − 1 measure qubits.

The patch uses 2d² − 1 qubits: 49, not 25, at d = 5.

4 Fault tolerance and the threshold

What if the checks themselves are noisy?

Error correction makes errors too

(a) X on the control

$$\mathrm{CNOT}\,(X \otimes I) = (X \otimes X)\,\mathrm{CNOT}$$

(b) Z on the target

$$\mathrm{CNOT}\,(I \otimes Z) = (Z \otimes Z)\,\mathrm{CNOT}$$

(c) one X fault on the ancilla of an X-check, halfway through: X on two data qubits

A CNOT copies errors as well as values.

One fault in a syndrome circuit can leave a weight-2 error, which a distance-3 code in general cannot correct.

Fault-tolerant: one fault anywhere causes at most a correctable error.

Surface code: one measure qubit per check; CNOTs ordered so weight-2 errors lie across the logical strings.

Repeated rounds and the threshold theorem

Threshold theorem

Noise weak and local: there is $p_{th} > 0$.

If every operation fails with $p < p_{th}$,

a computation of any length runs with arbitrarily small logical error,

at polylogarithmic overhead.

A measure qubit can lie: one round cannot tell a data error from a measurement error.

So checks are repeated, typically d rounds; a decoder finds the likeliest faults from the whole history.

Correction runs continuously alongside the computation.

Below threshold larger codes help; above it they hurt.

DERIVATION

How fast does the logical error fall?

$$p_L \approx A\left(\frac{p}{p_{th}}\right)^{(d+1)/2}$$

about (d + 1)/2 faults must line up

$$p_L(d+2) = p_L(d) \cdot \frac{p}{p_{th}}$$

each step of 2 in distance

$$\Lambda = \frac{p_L(d)}{p_L(d+2)} = \frac{p_{th}}{p} \text{ (model)}$$

the suppression factor

Λ > 1 means below threshold: every step in distance divides the logical error by Λ.

A threshold needs its noise model

noise model (surface code)	threshold	source
independent X, Z on data; perfect checks	≈ 10 %	Roffe (10.3 %, matching)
faulty gates, preparation, measurement	≈ 1 %	Roffe
early concatenated-code estimates	~10^{-4}	Nielsen & Chuang §10.6

Code capacity (perfect measurements): about 10 %. Circuit level: about 1 %.

The repetition code's break-even p = ½ is not a threshold of any quantum code.

YOUR TURN · CALCULATE

Your turn: how many qubits per logical qubit?

Pairs · 90 seconds

Model: $A = 0.1$, $p_{th} = 1\%$, physical $p = 0.1\%$.

Smallest odd d with $p_L < 10^{-9}$ per round?

How many qubits does that patch use?

ANSWER

d = 17: 577 qubits

$p_L = 0.1 \cdot 0.1^{(d+1)/2} = 10^{-1-(d+1)/2}$

10^{-9}: d = 15 gives exactly 10^{-9}; d = 17 gives 10^{-10}.

Targets 10^{-6}, 10^{-9}, 10^{-12}: d = 11, 17, 23; 241, 577, 1057 qubits.

Of the order of a thousand physical qubits per logical qubit, before gates and routing.

Experiments: reading Λ

2023: d = 5 vs d = 3

2.914 % vs 3.028 % per cycle

Λ ≈ 1.04: modestly better

2024 (Nature 2025): d = 7, 101 qubits

0.143 % per cycle; Λ = 2.14 ± 0.02

1.1 μs cycle; lifetime 2.4× best physical qubit

Extrapolate with constant Λ: 0.067 % at d = 9, 0.031 % at d = 11.

Λ = 2.14 means $p/p_{th} \approx 0.47$: about half the threshold.

Rare correlated events (about once an hour) set a floor that does not shrink with d.

YOUR TURN · PREDICT

Predict, then simulate: a repetition-code memory

Think · 1 minute

Each data qubit ends flipped with probability q, a few times p.

Using $p_L \approx 3q^2$ (d = 3) and $p_L \approx 10q^3$ (d = 5), predict $p_L(3)/p_L(5)$ at $q \approx 0.03$.

How does the ratio change as q grows?

SIMULATION

Simulation: a repetition-code memory in Stim

```
c = stim.Circuit.generated(
"repetition_code:memory", distance=d,
after_clifford_depolarization=p,
m = c.compile_sampler(seed=6586)
data = m[:, -d:]
fail = data.sum(axis=1) > d // 2
```

Excerpt; one round, 200 000 shots per entry.

Decoder: majority of the final data readout.

p	d = 3	d = 5	d = 7
0.01	2.02e-03	2.45e-04	5.00e-06
0.02	7.87e-03	1.50e-03	2.50e-04
0.05	4.12e-02	1.74e-02	7.45e-03
0.10	1.27e-01	8.61e-02	5.80e-02
0.20	3.08e-01	2.76e-01	2.52e-01

Recap

1

Stabiliser codes

checks commute; syndrome = anticommutation; $k = n - m$

2

Steane and five-qubit

[[7,1,3]] from Hamming; [[5,1,3]] is the smallest

3

Surface code

local checks; strings across; $2d^2 - 1$ qubits

4

Threshold

below it $\Lambda > 1$; ≈ 1 % circuit level; ~10^3 qubits per logical

Next week: Lectures 19–20 (Chapter 9)

Physical realisations: where p comes from

Read: handout 8.4–8.7 before next week; 8.8 is self-study, 8.9 reference.

Exit ticket

A patch at $p/p_{th} = 0.2$ has $p_L = 10^{-4}$ at d = 5. Estimate p_L at d = 9 and the qubits needed.

Copyright © 2026 Sajid Muhaimin Choudhury. All rights reserved.

Rights and attributions

Copyright (c) 2026 Sajid Muhaimin Choudhury. All rights reserved.

Public edition of Lecture 18: Stabilisers, the Surface Code and Fault Tolerance. Release qc-2026-10-15, prepared 2026-10-15 from source revision 3df9449aa6.

This edition omits speaker notes, instructor-only material and every figure whose publication rights are not cleared for public distribution.