Project 3: Code Concatenation

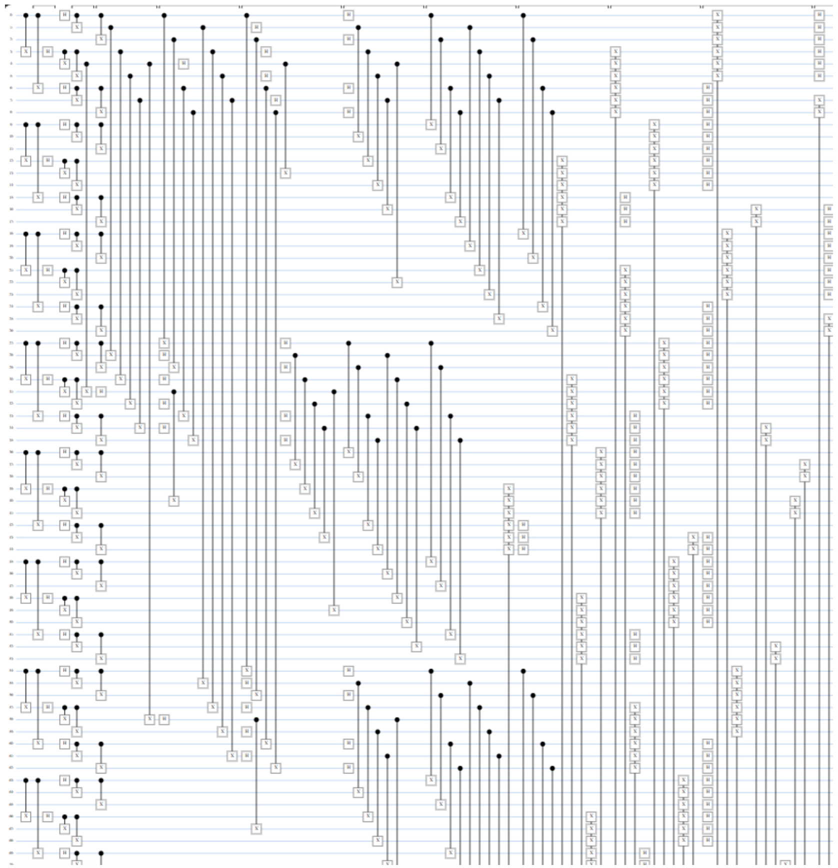
Course: Practical Quantum Computing – C3260

Professor: Alexandru Paler

Developer: Mahtab Hosseini-Dolama

Introduction

- Code concatenation in quantum error correction:
  - Achieve lower logical error rates by concatenating error correcting codes.
  - Encode logical qubits multiple times, using layers of error correcting codes.
  - Correcting code at multiple levels → Reducing overall probability of error propagation.
  - Exponential suppression of logical error rates.
  - Threshold Theorem.



- Shor’s code:
  - Suggested by Peter Shor in 1995.
  - 1 logical qubit → 9 physical qubits.
  - Capable of correcting any single-qubit error.

Project Goals

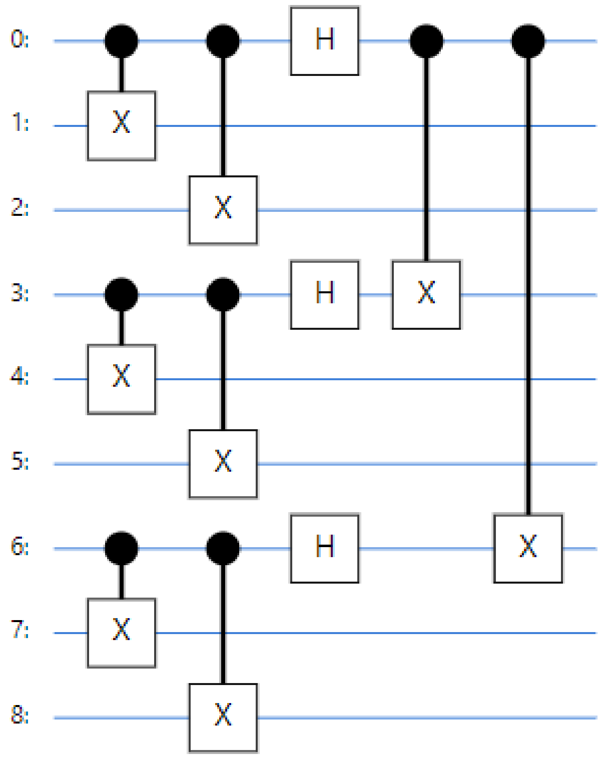
- Implement Shor code and concatenate it with itself (Qualtran).
- Draw the circuit diagrams for the resulting concatenation (Cirq).
- Plot logical error rates for Shor code without concatenation (Cirq).

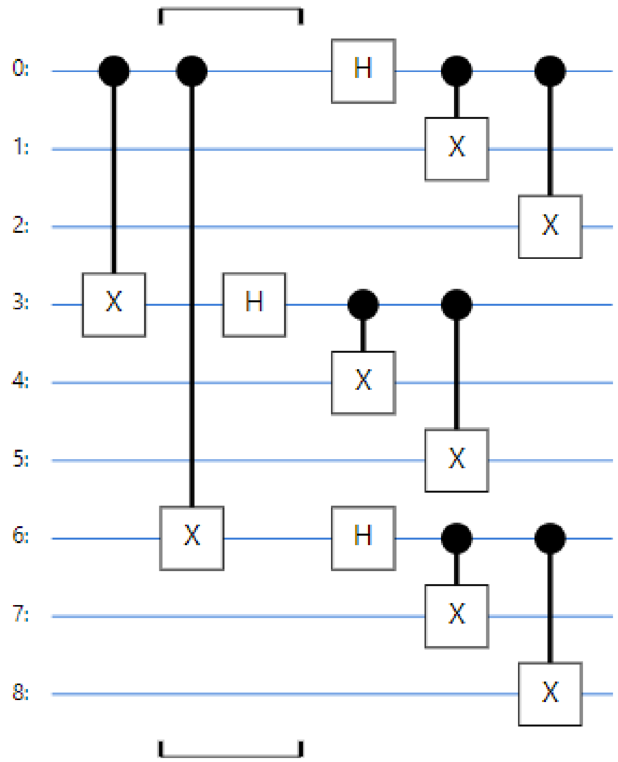
Shor Code in Qualtran

- 9 Physical Qubits
- 8 Ancillary Qubits
- Required Bloqs:
  - Encoder
  - Syndrome “Measurements”
  - Recovery → Principle of Deferred Measurement
  - Decoder
  - Main Function

```
, Z4
8, Z9

(unconcatenated) Shor code
 code "logical" is the Soquet with the 9 qubits, and "qubits" is the form of "logical" that's modified du

frozen
ShorCodeAll(Bloq):
roperty
ef signature(self):
    return Signature.build(logical=9, ancilla=8)

def build_composite_bloq(self, bb: BloqBuilder, *, logical: SoquetT, ancilla: SoquetT) -> Dict[str, SoquetT]:

    # Initialize the data qubit to |+> state
    qubits = bb.split(logical)
    qubits[0] = bb.add(Hadamard(), q=qubits[0])
    qubits = bb.join(qubits)

    # Encoding
    qubits= bb.add_from(ShorEncode(), logical=qubits)[0]

    # syndrome measurements
    qubits, a = bb.add_from(ShorSyndrome(), logical=qubits, ancilla=ancilla)

    # recovery
    qubits, a = bb.add_from(ShorRecovery(), logical=qubits, ancilla=a)

    # decoding step
    qubits= bb.add_from(ShorDecode(), logical=qubits)[0]

    # return the error corrected qubits
    return {'logical': qubits, 'ancilla': a}


ttrs.frozen
ass ShorEncode(Bloq):
@property
def signature(self):
    return Signature.build(logical=9)

uild_composite_bloq(self, bb: BloqBuilder, *, logical: SoquetT) -> Dict[str, SoquetT]:
```

Encoder and Decoder Circuits

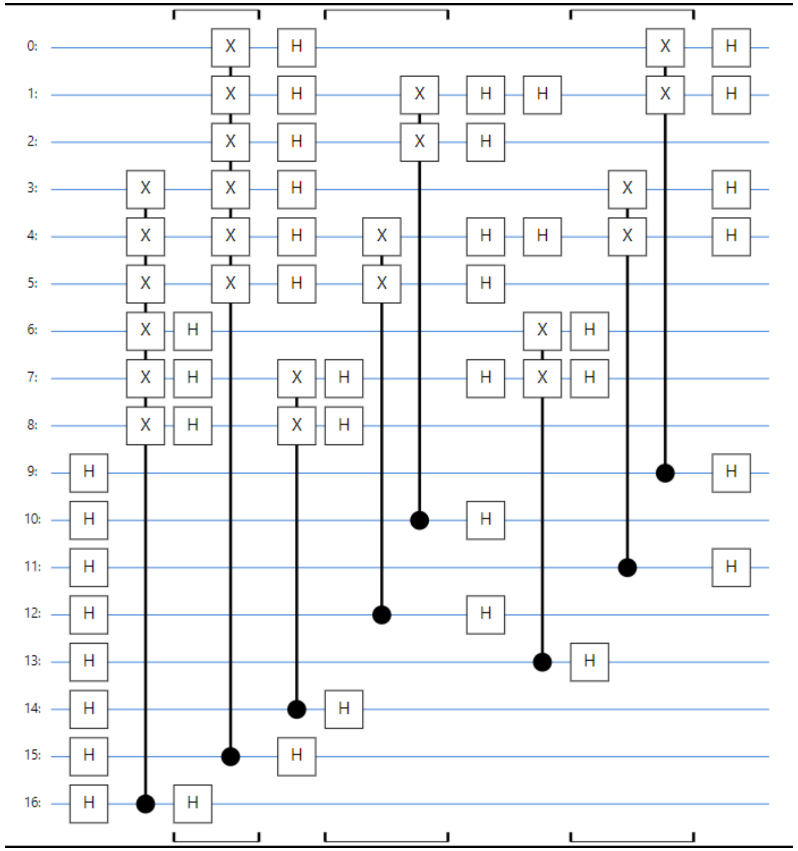




Syndrome "Measurements"

- $S_{[[9,1,3]]} = \{Z_0Z_1, Z_1Z_2, Z_3Z_4, Z_4Z_5, Z_6Z_7, Z_7Z_8, X_0X_1X_2X_3X_4X_5, X_3X_4X_5X_6X_7X_8\}$

Error	Syndrome S	Error	Syndrome S
X_0	10000000	Z_0	00000010
X_1	11000000	Z_1	00000010
X_2	01000000	Z_2	00000010
X_3	00100000	Z_3	00000011
X_4	00110000	Z_4	00000011
X_5	00010000	Z_5	00000011
X_6	00001000	Z_6	00000001
X_7	00001100	Z_7	0000001
X_8	00000100	Z_8	00000001



Recovery

- Principle of Deferred Measurement:

  “Delaying measurements until the end of a quantum computation does not affect the probability distribution of outcomes”.

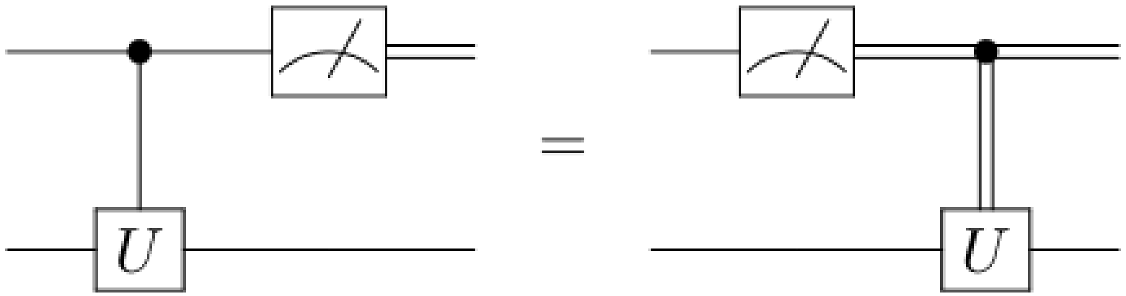
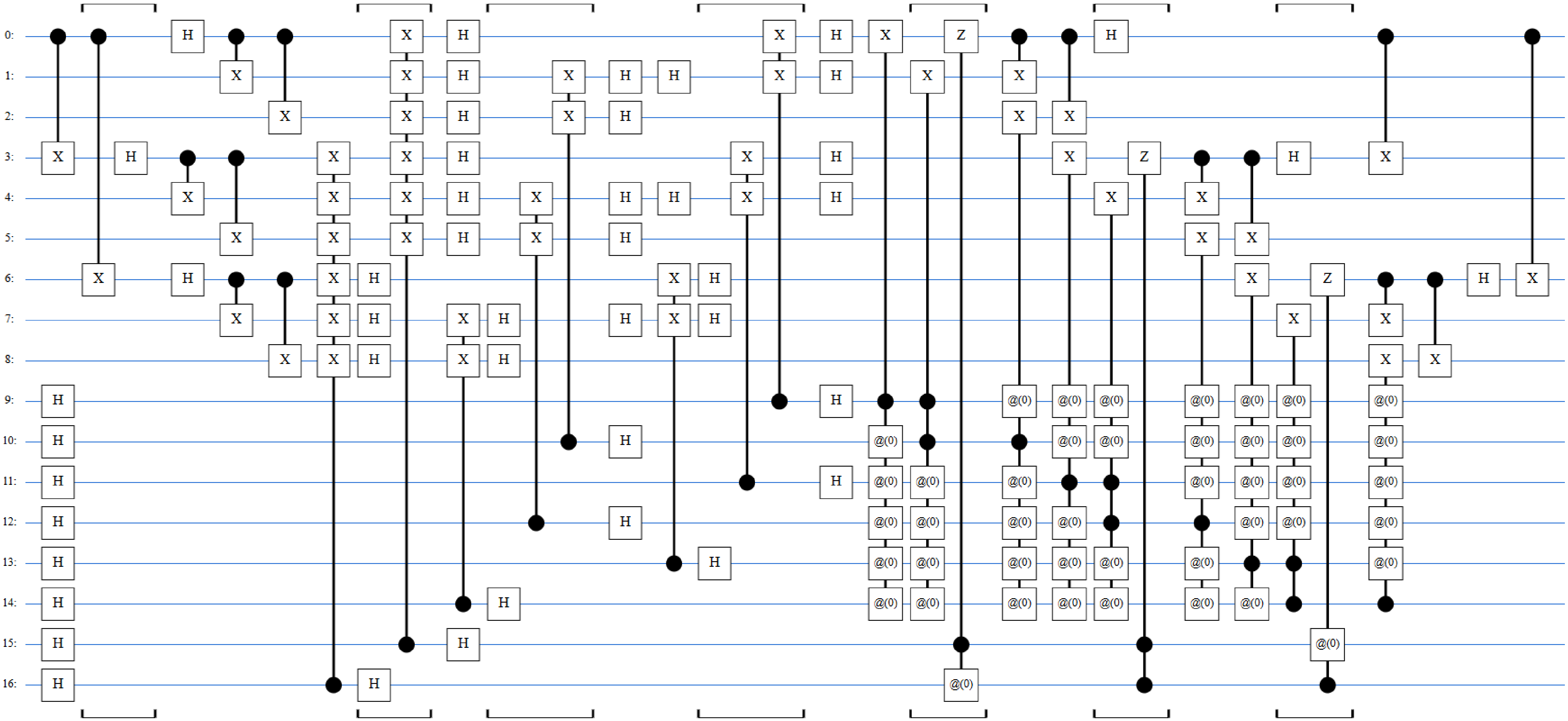


Image from Wikipedia

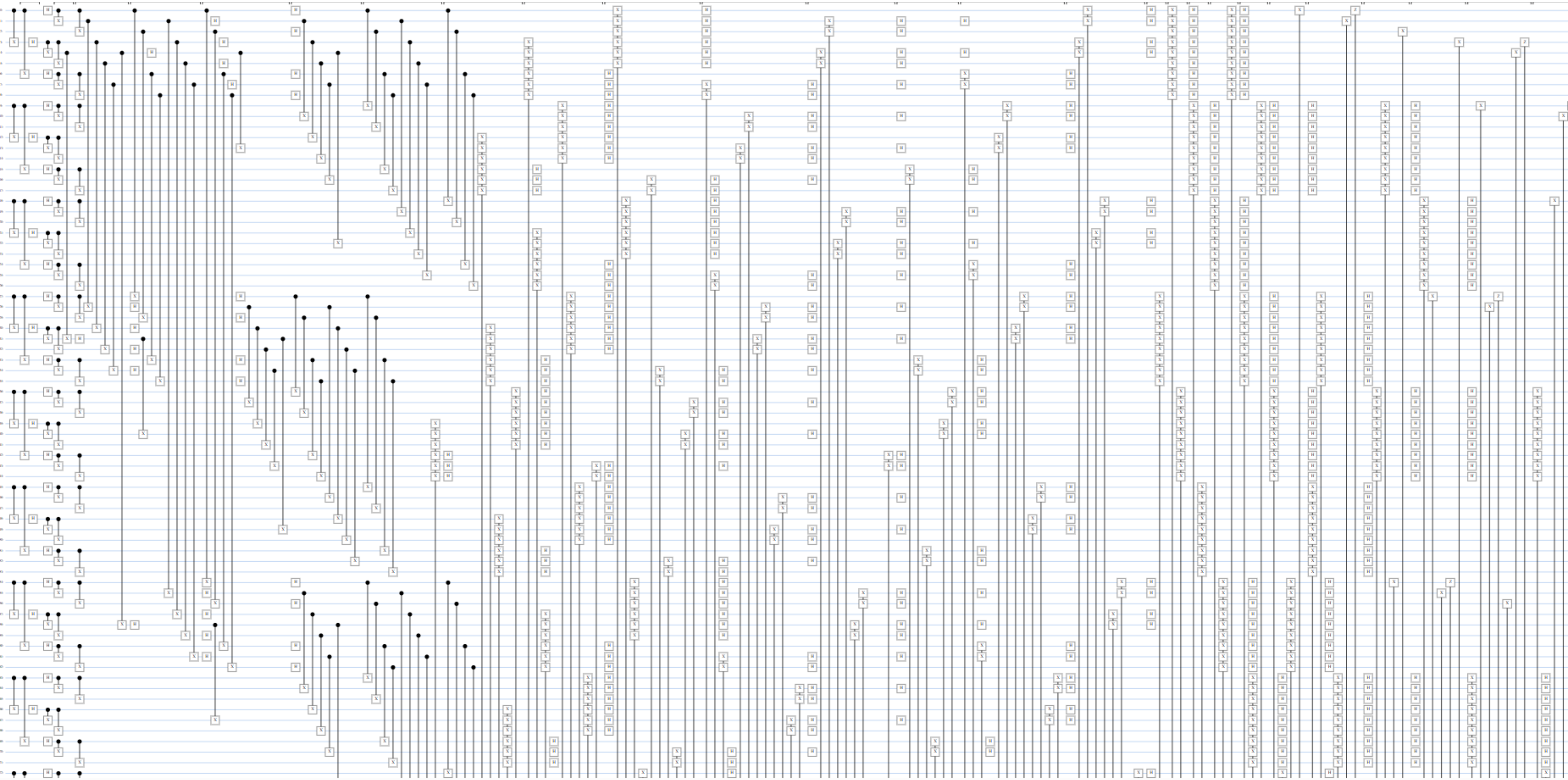
- Multi-control Toffoli gates were used.

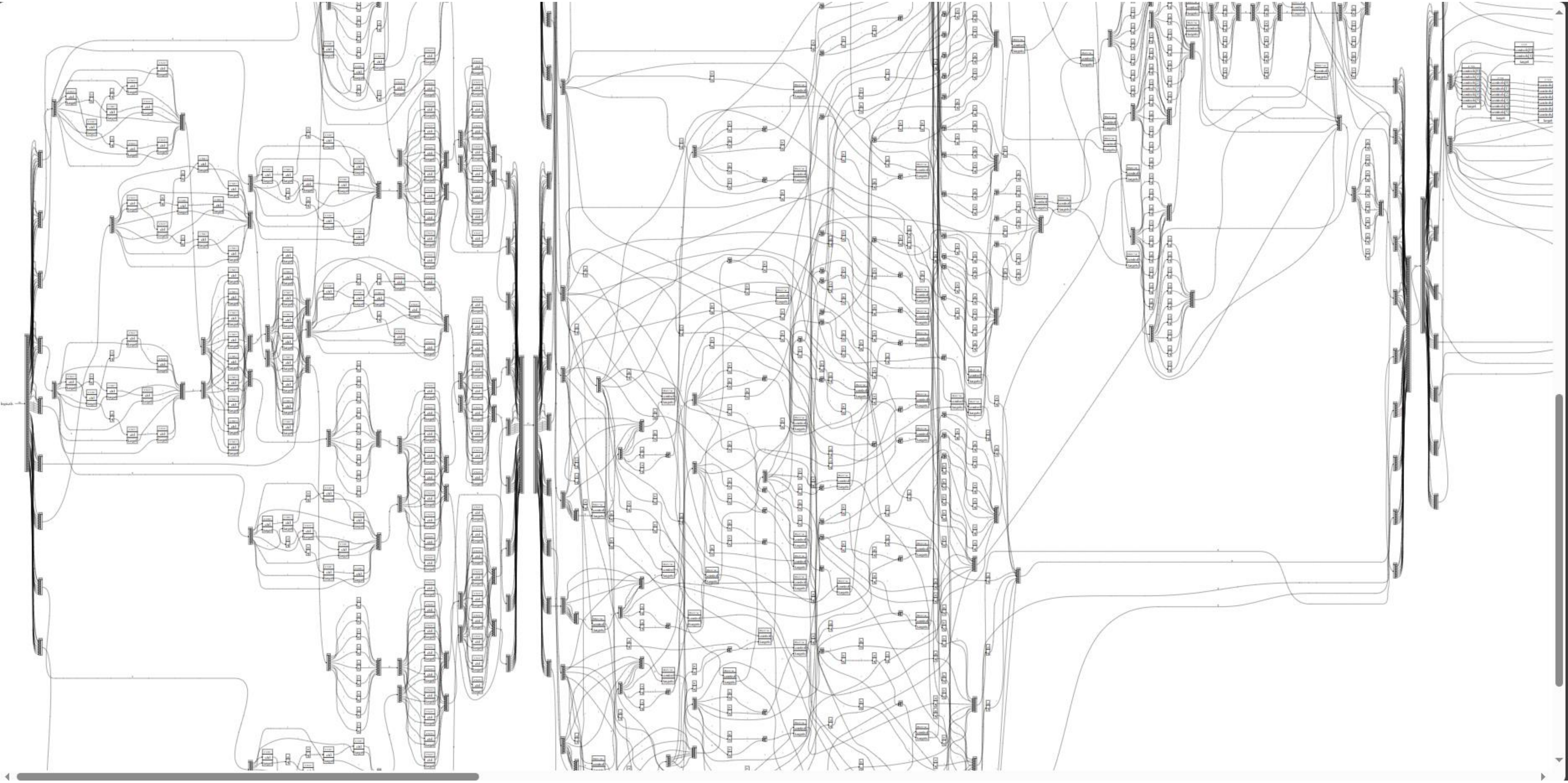

0:
1:
2:
3:
4:
5:
6:
7:
8:
9:
10:
11:
12:
13:
14:
15:
16:
H
X
Z
@(0)

Concatenated Shor Code

- 81 Physical Qubits (9 Logical Qubits)
- 80 Ancillary Qubits

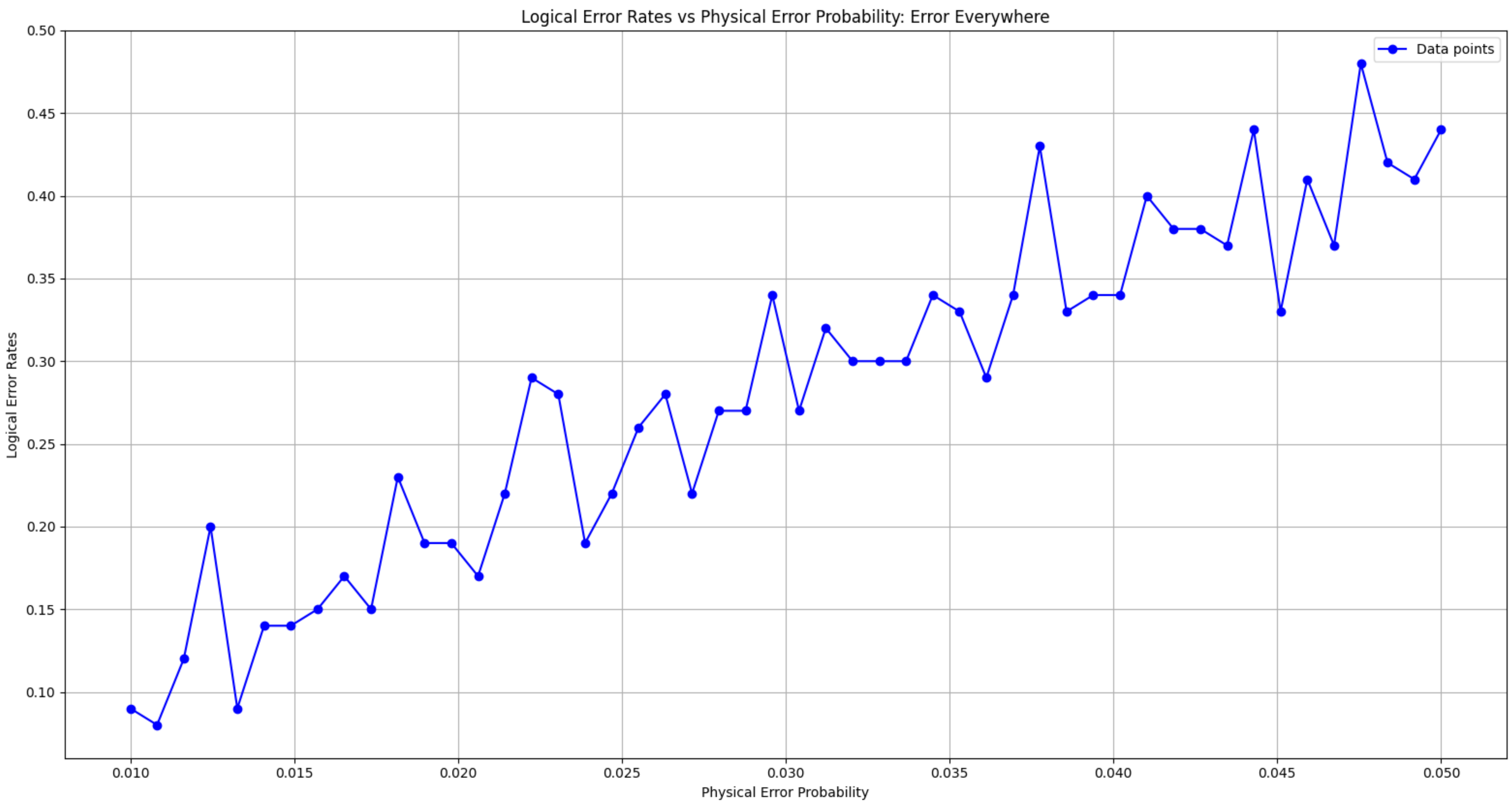
- Required Bloqs:
  - Logical Encoder
  - Logical Syndrome "Measurements"→ same syndromes as the nine-qubit Shor code
  - Logical Recovery → Principle of Deferred Measurement
  - Logical Decoder
  - Main Function
  - Custom Logical Gates (e.g. Logical X, H, CNOT Gate etc.)

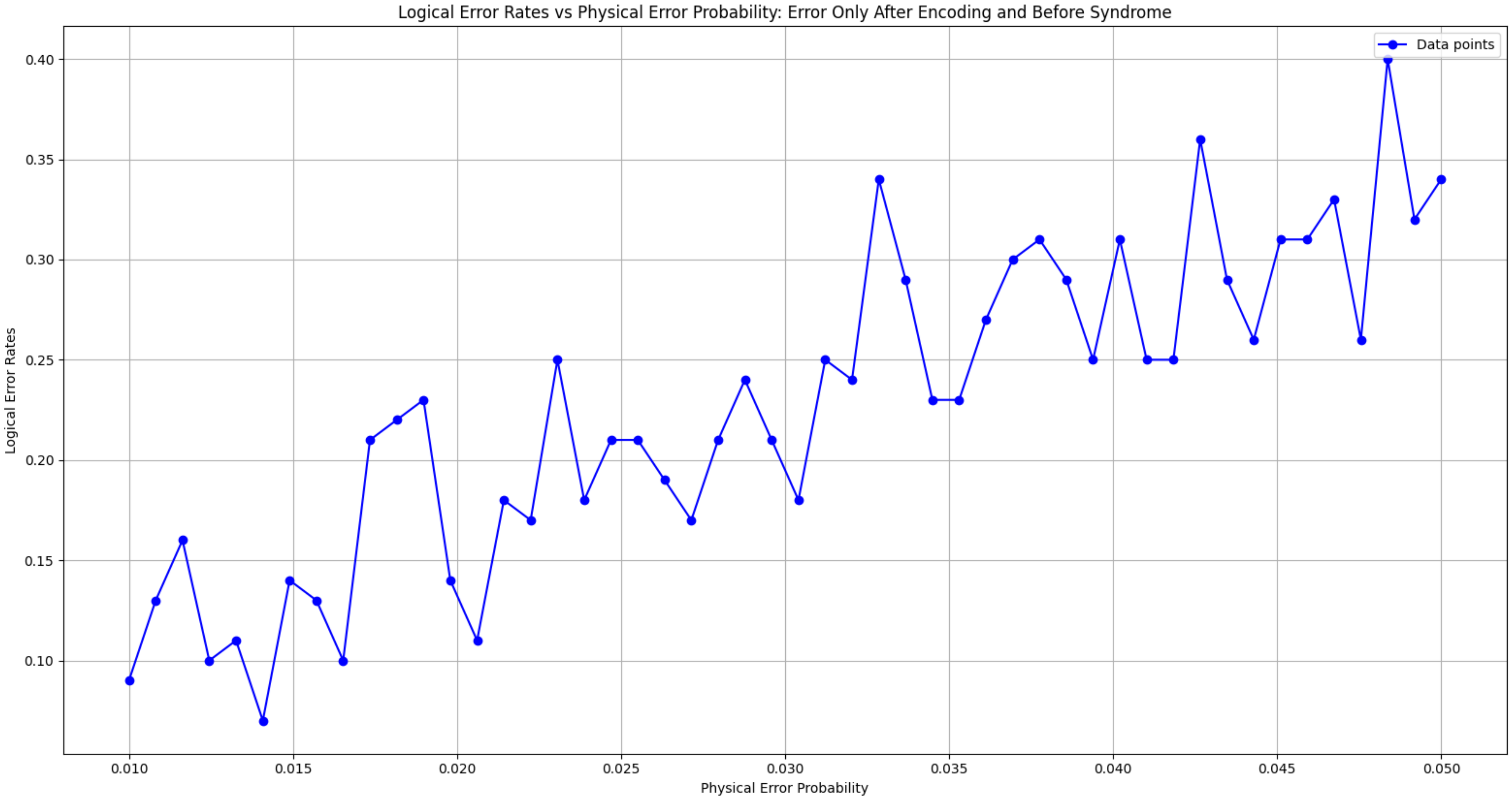




Simulation and Logical Error Rates

- Cirq's pure state simulator.
- Depolarizing channel with error probability range of $[0.01, 0.05]$ with 50 linearly spaced points.
- 100 simulation runs for each physical error rate.
- Logical error rate was computed for each physical error rate.
- Two types of simulations: $\begin{cases} \text{Errors occurring only between encode and syndrome} \\ \text{Errors occuring everywhere in the circuit} \end{cases}$


Logical Error Rates vs Physical Error Probability: Error Everywhere
Data points
Logical Error Rates
Physical Error Probability
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.010
0.015
0.020
0.025
0.030
0.035
0.040
0.045
0.050


Logical Error Rates vs Physical Error Probability: Error Only After Encoding and Before Syndrome
Data points
0.40
0.35
0.30
0.25
0.20
0.15
0.10
Logical Error Rates
0.010
0.015
0.020
0.025
0.030
0.035
0.040
0.045
0.050
Physical Error Probability

Conclusion

- The plots demonstrate lower logical error rates on average (and maximum logical error rate) for the situation when the error occurs only after encoding, as is expected.

- Under both conditions logical error rates increase linearly as the physical error rates increase.

- Shor code offers some degree of robustness to error channels acting on all qubits, despite only correcting for single-qubit errors.

- According to the non-negligible logical error rates resulting from high depolarizing error rates, it is expected that the concatenated scheme would show increased robustness against logical errors with a similar linearly increasing trend.