Quantum Logic 2025-26
– TP projects –

Nico Wittrock

April 6, 2026

This file contains the project descriptions and summarizes the procedure:

- we have an online kick-off meeting on *Tuesday, April 7, at 14:30*
- project teams should consist of *1-2 students*
- every team should tackle a *different project*
- and present their solution *on June 8*; time and venue: tba
- a report is not necessary
- If you have doubts, questions or suggestions, please do not hesitate to contact me.

In project 1, you will basically implement our notions from the TP classes in Haskell; in all other projects, you may choose between zxlive and the python-package PyZX, in order to implement a specific (quantum) algorithm.

Projects

1 ZX calculus in Haskell

Recall how we derived the ZX calculus in the TP classes: starting from "classical" maps and rules (in the category of sets), we applied the *free-Hilbert-space-functor* to translate this structure to the quantum realm (the category of Hilbert spaces and linear functions).
Now Haskell has the built-in notion of functors [3]. Use this functionality to implement our approach in Haskell:

1. on the classical side, you may want to replace the type of finite Sets [10] by *(finite) cardinal numbers* [11], which only consists of sets of the form $\{0, \ldots n\}$; this will facilitate the next steps 2-5; define the set of bits and our functions *swap*, *delete*, *copy*, $falsum$, *xor*, *not*;

2. test at least three rules (out of the 13 rules in the TP classes) with random inputs;

3. for the quantum realm, you may replace the category of Hilbert spaces by the (very simple) category of *complex matrices* [8]: it has natural numbers m, n as objects and $m \times n$-matrices as morphisms (see also [1, Exercise 6]);

4. this sets the ground to define the free-Hilbert-space-functor. It's built-in functions [3] translate all functions from step 1 to complex matrices;

5. test at least three rules in the quantu realm with random inputs.

2 Quantum Error Correction I

Parity checks are among the easiest error detection and correction techniques in (classical and quantum) signal processing. An example is the following scenario from fig. 1, which is explained in slightly more detail in [12]:

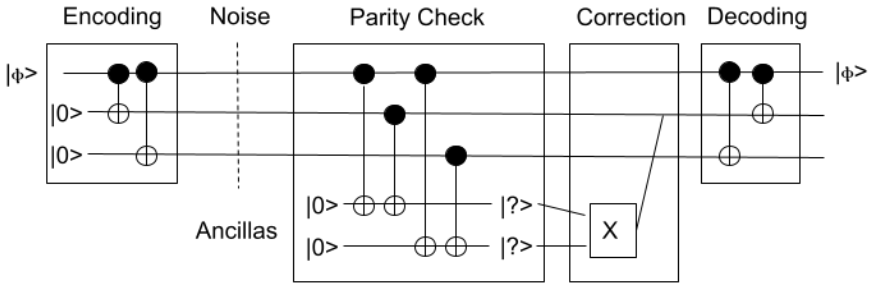

Figure 1: noisy communication via parity measurement, picture from [12]

- the sender wants to communicate the state ϕ; they *encode* and send it through a noise channel, which can be simulated by adding independent phases to the three qbits;
- the receiver receives these (potentially disturbed) three signals and tries to recover the original message ϕ, using *parity check*, *error correction* and *decoding*.

In this project, you are going to

1. complete the algorithm by finding an appropriate correction step;
2. implement this scenario without noise in zxlive or PyZX;
3. simplify the string diagram / circuit; what does the result mean?
4. optional: add noise (via Z or X rotation) in one of the qbits. How does it propagate?

3 Quantum Fourier Transform

As in classical signal processing, the Fourier Transform is crucial for many quantum algorithms, most notably Shor's algorithm[2].
In this project, you will derive it's quantum circuit using the ZX calculus:

1. follow the steps from [5, Exercise 7.5] (except for the last one) to derive a ZX diagram for the quantum Fourier transform on 3 qbits;
2. translate your result to a quantum circuit.

4 Measurement-based Quantum Computing

Quantum algorithms are commonly expressed by *quantum circuits*. An alternative approach to quantum computing is *measurement-based quantum computing* (MBQC).
In this project, you will take this approach via an easy example:

1. Implement the quantum gate teleportation protocol [5, Section 3.3.2] in zxlive or PyZX;
2. identify the three steps of MBQC: entanglement, measurement, and correction
3. why do we need the correction step?

5 Quantum Phase Estimation

The *quantum phase estimation algorithm*[6] is an important tool in quantum computing, e.g in Shor's algorithm. Given a unitary U on m qbits, with eigenvector $\psi \in (\mathbb{C}^2)^{\otimes m}$, the algorithm estimates the corresponding eigenvalue λ, using n auxiliary qbits; it's general quantum circuit is shown in figure 2. See also [7, Section 5.2] for a detailed description.

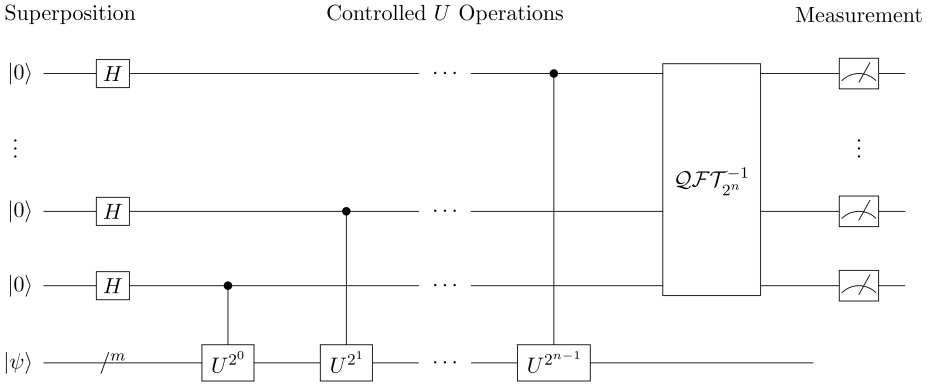

Figure 2: General quantum circuit of phase estimation; picture from Wikipedia

In this project, you will prove it's correctness for the simple special case $U = Z(\alpha)$ (i.e. $m = 1$) for general $\alpha \in [0, 2\pi)$ and $n = 1$ auxiliary qbits (such that the inverser Fourier transform in fig. 2 collapses to an H-gate).

1. determine the non-trivial eigenvalue $\lambda \in \mathbb{C}$ and vector $\psi \in \mathbb{C}^2$ (depending on α)
2. adapt the circuit from fig. 2 to your scenario and implement it in zxlive, using Paramtetric rules [15] and [5, Exercise 7.5 1.] for the controlled U-gate; ignore the measurment process;
3. simplify the ZX circuit, using the ZX ules
4. explain how repeated measurement processes would estimate the eigenvalue λ; (the #Toy examples in[13] might help).

6 Quantum Error Correction II (difficult)

Warning: this project is much more advanced than the others (but also very interesting)!

Surface codes are a group of quantum error correction algorithms that are actually applied (by experimental physicists and tech companies[9]), but also subject to theoretical research (in computer science and mathematical physics); the oldest and most prominent member of this group is the *toric code* [6]. By a recent

result [4] the ZX-calculus is an appropriate framework for CSS codes.
Follow the demonstration of [4, Section 7] to explain the toric code for a 2×2 grid in the ZX calculus, using the graphical editor in zxlive.

7 Classical vs Quantum Error Correction (difficult)

Warning: this project is not difficult per se, but requires background on stabilizer theory (see [5, chapter 6])

Hamming codes are well studied error correction algorithms in classical communication; they generalize to *Steane codes* in quantum computing.
Implement the 7-qubit Steane code in zxlive of PyZX by spelling out the diagrammatic calculations of [5, Section 12.2.3].

Optional: how does this recover the old Hamming code [14]?

References

[1] Luís Soares Barbosa. *Lecture Notes 1.* URL: `https://lmf.di.uminho.pt//quantum-logic-2526/LQ-Lec1.pdf` (visited on 04/03/2026).

[2] Austin G Fowler, Simon J Devitt, and Lloyd CL Hollenberg. "Implementation of Shor's algorithm on a linear nearest neighbour qubit array". In: *arXiv preprint quant-ph/0402196* (2004).

[3] *HaskellWiki.* URL: `https://wiki.haskell.org/Functor` (visited on 04/03/2026).

[4] Aleks Kissinger. "Phase-free ZX diagrams are CSS codes (... or how to graphically grok the surface code)". In: *arXiv preprint arXiv:2204.14038* (2022).

[5] Aleks Kissinger and John van de Wetering. *Picturing Quantum Software: An Introduction to the ZX-Calculus and Quantum Compilation.* Preprint, 2024.

[6] A Yu Kitaev. "Quantum communication, computing, and measurement". In: *Proceedings of the 3rd International Conference of Quantum Communication and Measurement, New York: Plenum.* 1997.

[7] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information.* Cambridge university press, 2010.

[8] *nlab.* URL: `https://ncatlab.org/nlab/show/matrix` (visited on 04/03/2026).

[9] "Quantum error correction below the surface code threshold". In: *Nature* 638.8052 (2025), pp. 920–926.

[10] *The Haskell Package Repository*. URL: https://hackage-content.haskell.org/package/containers-0.8/docs/Data-Set.html (visited on 04/03/2026).

[11] *Wikipedia*. URL: https://en.wikipedia.org/wiki/Cardinal_number (visited on 04/03/2026).

[12] *Wikipedia*. URL: https://en.wikipedia.org/wiki/Parity_measurement#Example (visited on 04/03/2026).

[13] *Wikipedia*. URL: https://en.wikipedia.org/wiki/Quantum_phase_estimation (visited on 04/03/2026).

[14] *Wikipedia*. URL: https://en.wikipedia.org/wiki/Hamming(7,4) (visited on 04/03/2026).

[15] *zxlive*. URL: https://zxlive.readthedocs.io/en/latest/adding-rewrites.html#parametric-rules (visited on 04/03/2026).