Quantum Software

Aleks Kissinger, Stefano Gogioso


Oxford, Hillary Term 2024

Quantum software

1. := the code the runs on a quantum computer

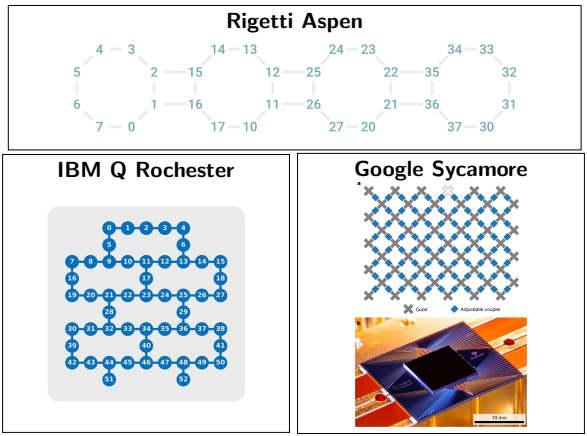
factoring

search

physical simulation
optimisation problems
linear systems & codes
network flows
natural language processing
...

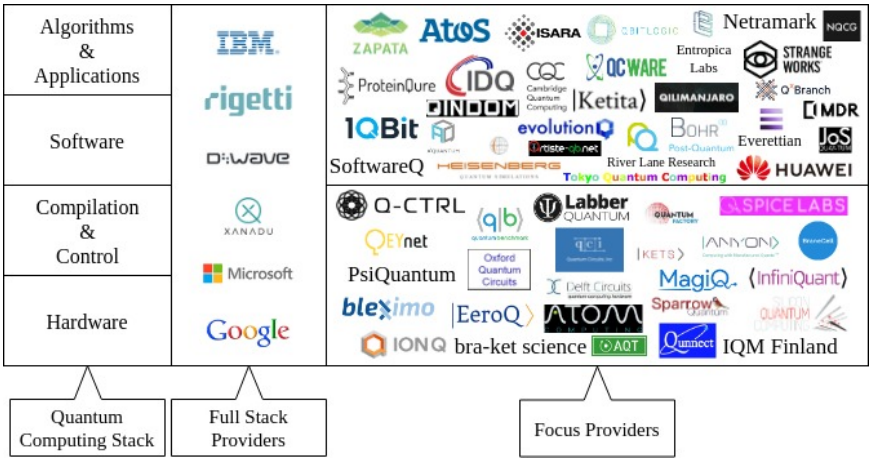
2. := the code that makes that code (better)
   - **compilers**
   - **optimisation**
   - **verification**

Problem: ~~no quantum computers~~



(+ Oxford, Vienna, Delft, Sussex, Maryland...)

Problem: ~~no quantum computers~~



[1]

[1] https://quantumcomputingreport.com/review-of-the-cirq-quantum-software-framework

Problem: limited quantum computers

NISQ := "noisy intermediate-scale quantum"

NISQ devices have:

- short coherence times
- low numbers of qubits
- noisy operations
- limited connectivity
- ...

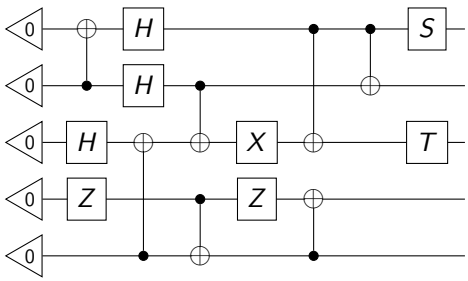
⇒ **small** advances in software give **big** gains on NISQ hardware!

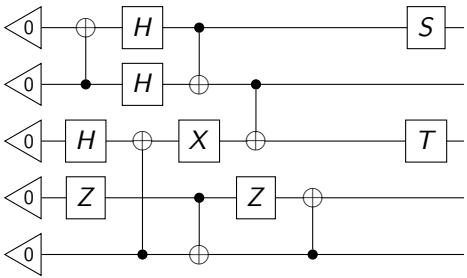
Quantum circuits

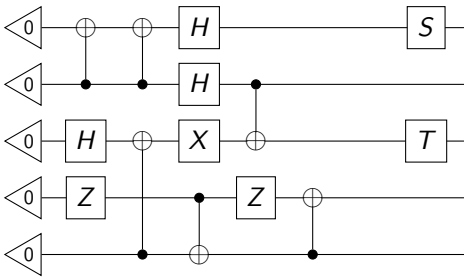
- := the 'assembly language' of quantum computation, e.g.

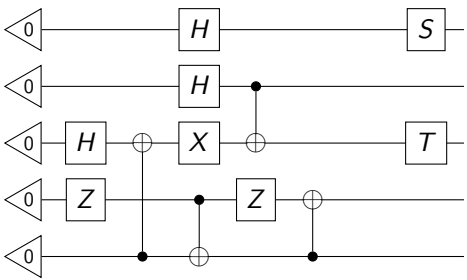
```
INIT 5
CNOT 1 0
H 2
Z 3
H 0
H 1
CNOT 4 2
...
```

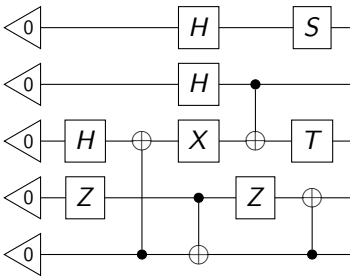
$\Leftrightarrow$

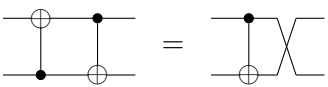


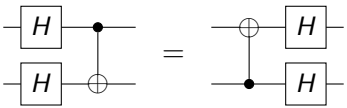

0
0
0
0
0
H
H
H
X
Z
Z
S
T

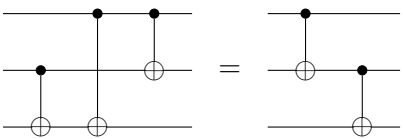

0
0
0
0
0
H
H
H
X
Z
Z
S
T

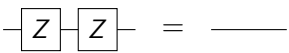

0
0
0
0
0
H
H
H
X
Z
Z
S
T


0
0
0
0
0
H
H
H
X
S
T
Z
Z



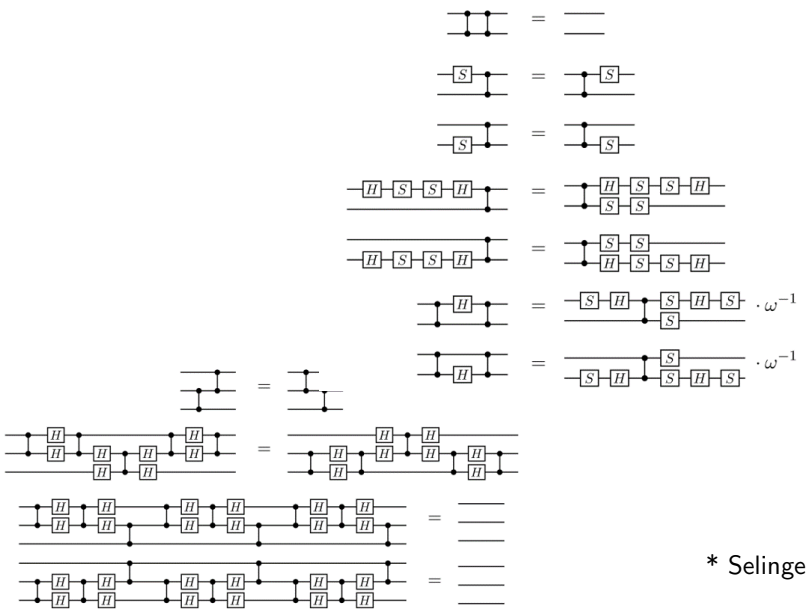

H
H
=
H
H




Z
Z
=

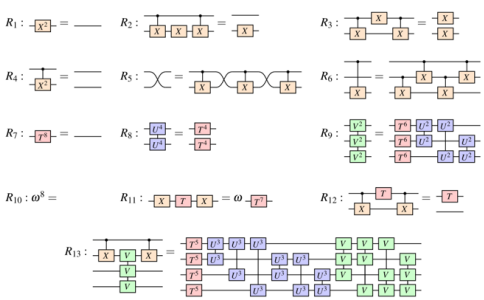

H
S
H
S
H
=
H

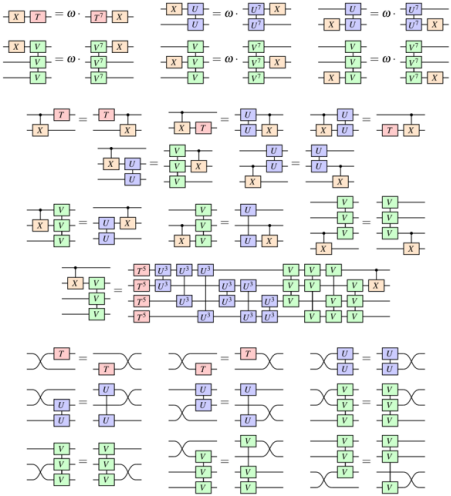
. . .



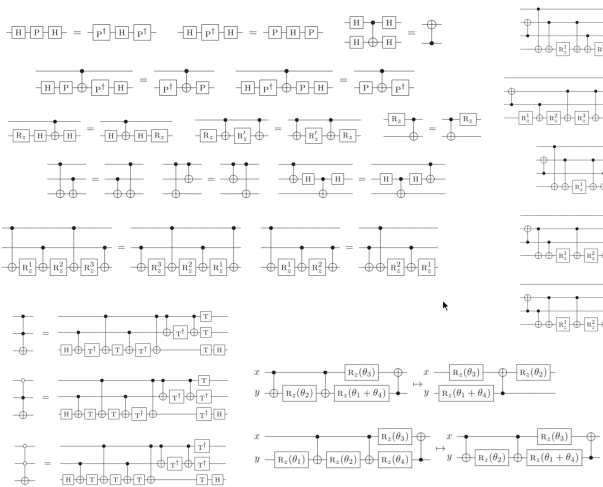
* Selinger 2015

* Selinger 2015

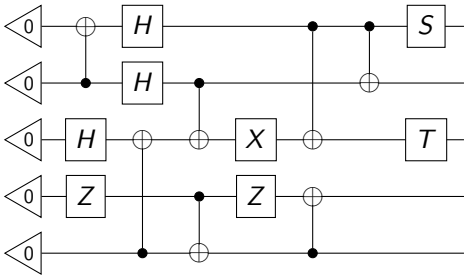


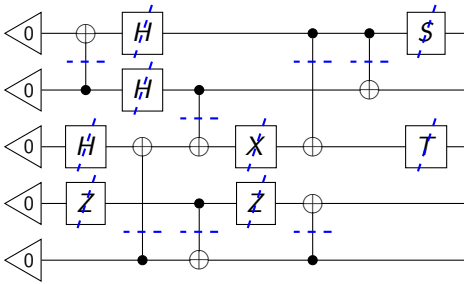


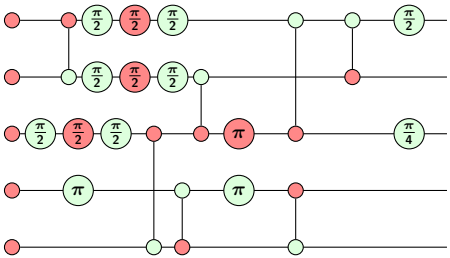
* Amy, Chen, & Ross 2018

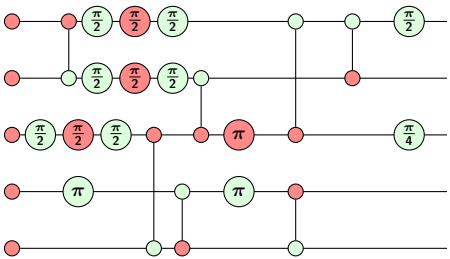


* Nam *et al* 2018


0
0
0
0
0
H
H
H
Z
X
Z
S
T


0
0
0
0
0
H
H
H
Z
Z
X
S
T


π/2
π/2
π/2
π/2
π/2
π/2
π/2
π/2
π/2
π/2
π/2
π
π/4
π
π



ZX-diagram

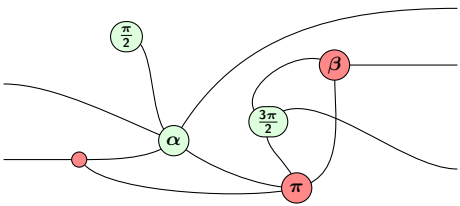
ZX-diagrams

...are like circuits, but made from **spiders** instead of gates:

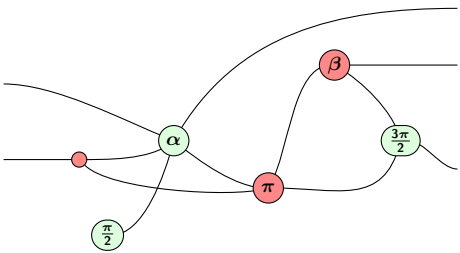
$$Z_\alpha \;=\; \vdots\ \overset{\alpha}{\bullet}\ \vdots$$

$$X_\alpha \;=\; \vdots\ \overset{\alpha}{\bullet}\ \vdots$$

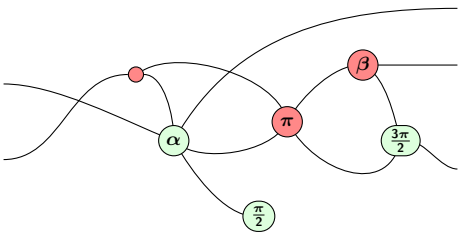
ZX-diagrams are bendy



ZX-diagrams are bendy

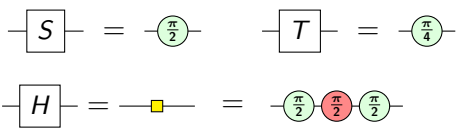


ZX-diagrams are bendy

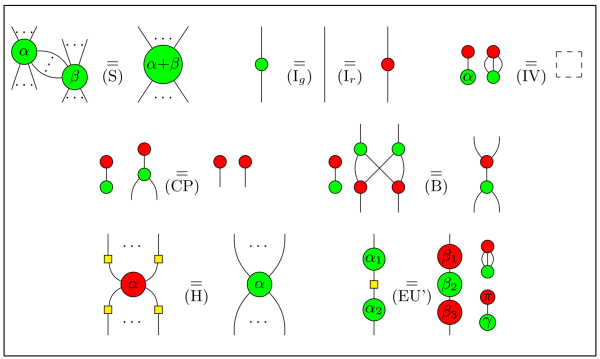


Why spiders?

- They generate **all** linear maps $\mathbb{C}^{2^m} \to \mathbb{C}^{2^n}$.
- Handy for building most common gates, e.g.



- ....and....

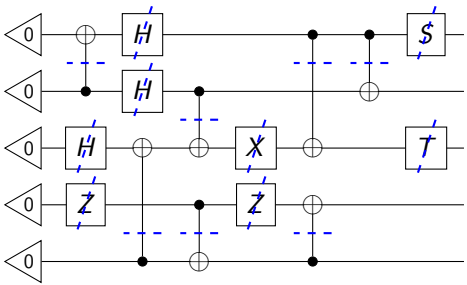


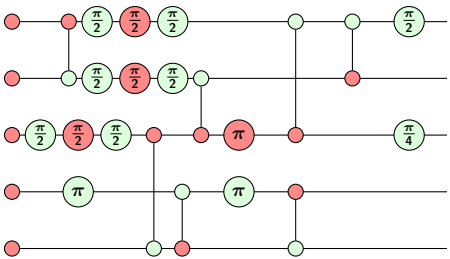
2

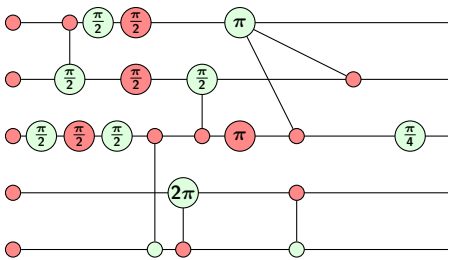
ZX calculus

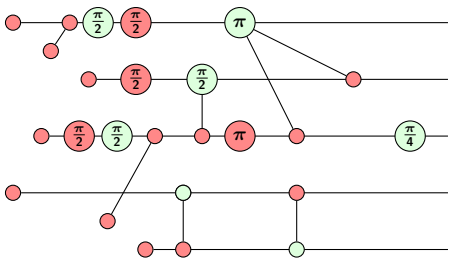
these 8 rules $\Longrightarrow$ everything before

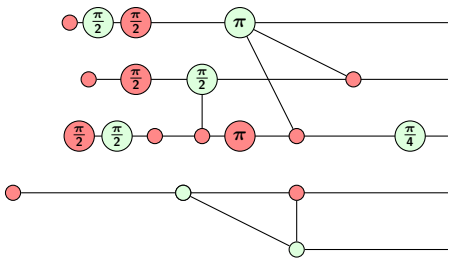
[2] Vilmart 2018. arXiv:1812.09114

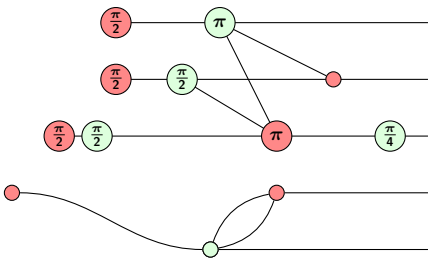

0
0
0
0
0
H
H
H
Z
X
Z
S
T

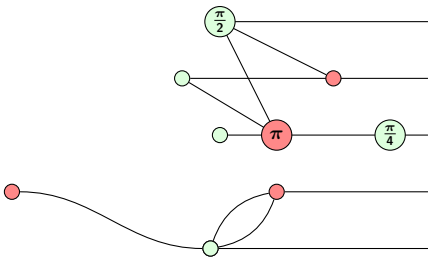

π/2
π/2
π/2
π/2
π/2
π/2
π/2
π/2
π/2
π/2
π/2
π
π/4
π
π

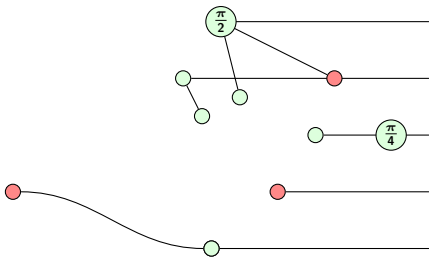

$\frac{\pi}{2}$
$\frac{\pi}{2}$
π
$\frac{\pi}{2}$
$\frac{\pi}{2}$
$\frac{\pi}{2}$
$\frac{\pi}{2}$
$\frac{\pi}{2}$
$\frac{\pi}{2}$
π
$\frac{\pi}{4}$
2π

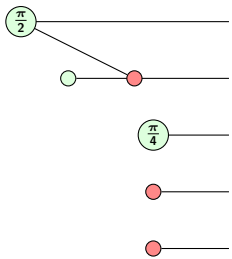

π/2
π/2
π
π/2
π/2
π/2
π/2
π
π/4

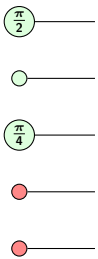

π/2
π/2
π
π/2
π/2
π/2
π/2
π
π/4

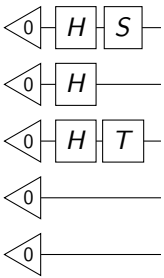

π/2
π
π/2
π/2
π/2
π/2
π
π/4


π/2
π
π/4

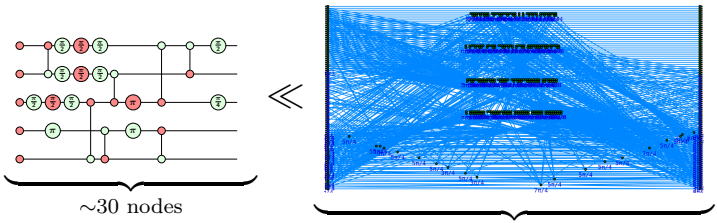

π/2
π/4


$\frac{\pi}{2}$
$\frac{\pi}{4}$


π/2
π/4

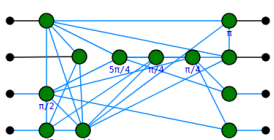

0
H
S
0
H
0
H
T
0
0

Q: How do we scale up?



A: Automation.

```
In [11]: zx.simplify.clifford_simp(g)
         g.normalise()
         zx.d3.draw(g)
```



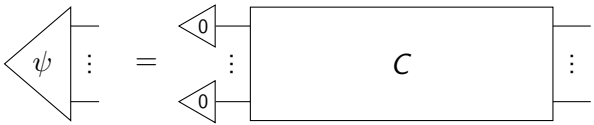
In this course...

- We'll look at **fundamental structures** underlying quantum computations:

$$\textbf{ZX-diagrams} \quad \longleftrightarrow \quad \begin{cases} \text{graph states} \\ \text{stabiliser groups} \\ \text{exponentiated Paulis} \\ \text{phase polynomials} \\ \dots \end{cases}$$

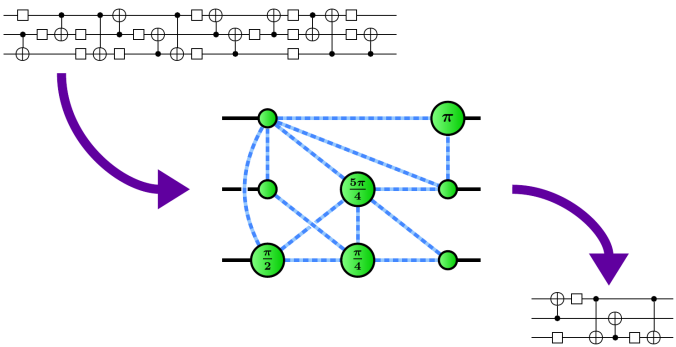
- And apply them to...

Classical simulation

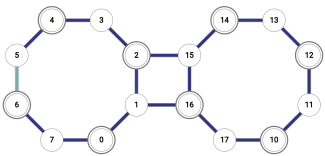


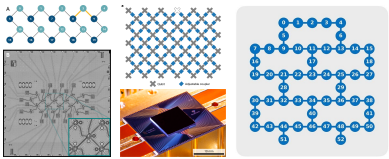
$$\mathrm{Prob}\big(i \,\big|\, |\psi\rangle\big) = ???$$

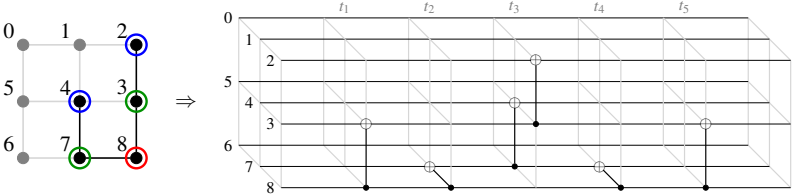
Circuit optimisation



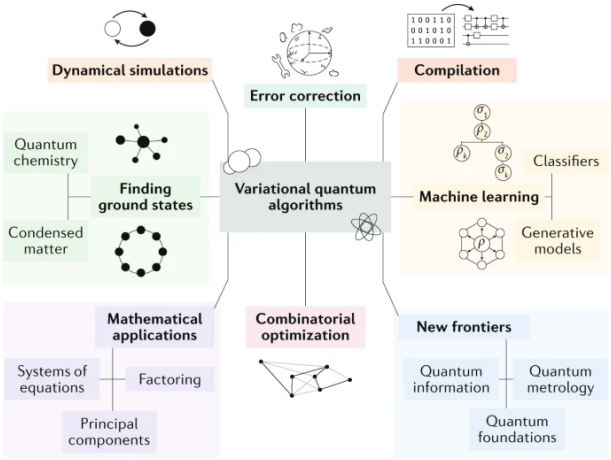
Circuit routing







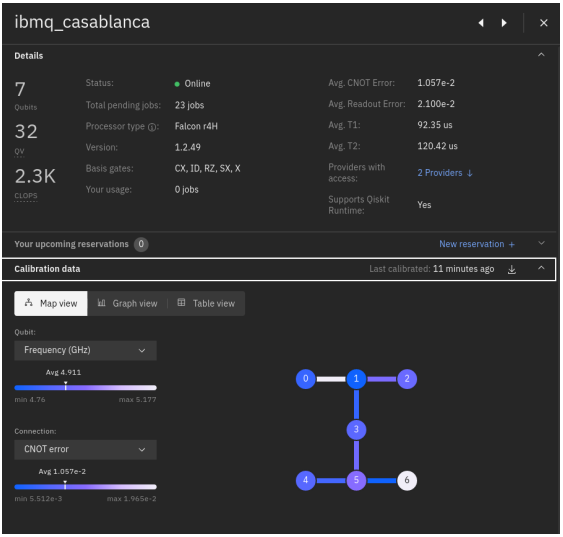
(NISQ) quantum algorithms



[3]

[3] https://www.nature.com/articles/s42254-021-00348-9

Implementation



[4]

[4] IBMQ calibration data. https://quantum-computing.ibm.com

Format of the course

- 8 weeks, 24 lectures, 6 problem sheets + classes (week 3, 4, 5, 6, 7, 8)
- 5 weeks theory (Aleks)
  - *basics, ZX, classical simulation, quantum compilation, quantum error correction*
- 3 weeks live coding (Stefano)
  - *algorithms, implementation on IBMQ*
- Materials:
  - *handwritten lecture notes, matching Aleks' lectures*
  - *textbook: preprint of Picturing Quantum Software. Kissinger & van de Wetering*
  - *Jupyter notebooks from live coding*
  - *Picturing Quantum Processes*. Coecke & Kissinger. CUP (optional)
  - *Quantum Computation and Quantum Information*. Nielsen & Chuang. CUP (optional)
- Exam is by take-home miniproject
  - *expect a theory and a (Python) coding component*