Basics of Quantum Computing in CT

Marc Kachelrieß

marc.kachelriess@dkfz.de

German Cancer Research Center (DKFZ)
Heidelberg, Germany
www.dkfz.de/ct

BASICS OF BASICS

Recap

Complex conjugate	$z^* = x - iy$
Symmetric matrix	$M^{\mathrm{T}} = M$
Orthogonal matrix	$O^{\mathrm{T}} = O^{-1}$
Hermitian adjoint = conjugate transpose	$M^\dagger = (M^{\mathrm{T}})^* = (M^*)^{\mathrm{T}}$
Hermitian operator (e.g. Hermitian matrix)	$H^\dagger = H$
Unitary matrix	$U^\dagger = U^{-1}$

Hermitian operators have real eigenvalues $H|n\rangle = \lambda_n|n\rangle \qquad \lambda_n \in \mathbb{R}$

The matrix exponential of a hermitian is unitary $U = e^{iH}$

Important Statements

$$i\hbar\frac{d}{dt}|\Psi(t)\rangle = H|\Psi(t)\rangle$$

Schrödinger equation

- Quantum algorithms[1] are those that are
  - inherently quantum
  - use some essential quantum feature such as superposition or entanglement
- The Schrödinger equation is a linear differential equation in time and position. Thus any linear combination of all the eigenfunctions of the Schrödinger equation can be a state of the system.
- Its solution is described by a unitary matrix

$$|\Psi(t)\rangle = U(t)|\Psi(0)\rangle = e^{\frac{1}{i\hbar}Ht}|\Psi(0)\rangle$$

- A qubit is a two-basis-state QM system with basis $|0\rangle$ and $|1\rangle$. Its state is

$$|\Psi\rangle = c_0|0\rangle + c_1|1\rangle$$

$$|c_0|^2 + |c_1|^2 = 1$$

- Two qubits are described by a four-dimensional Hilbert space.

[1]All classical algorithms can be performed on a quantum computer. However, it may not make sense to do so.

dkfz.

Pure vs. Mixed States

- **Pure states are fully specified quantum states with intrinsic measurement uncertainty.**

$$|\Psi\rangle = c_0|0\rangle + c_1|1\rangle$$

$$\rho = |\Psi\rangle\langle\Psi|$$

Measuring in the computational basis $\{|0\rangle, |1\rangle\}$ yields $|i\rangle$ with probability $|c_i|^2$.

If we want to estimate the value of $|c_i|^2$ we have to measure often and compute the relative frequency of having measured $|i\rangle$.

- **Mixed states describe statistical mixtures of quantum states and therefore include classical uncertainty.**

$$\rho = \sum_k p_k |\Psi_k\rangle\langle\Psi_k|$$

Tensor Product Notation

- Equivalent notations for a tensor product of, say, three qubits:

$$|q_2\rangle \otimes |q_1\rangle \otimes |q_0\rangle = |q_2\rangle|q_1\rangle|q_0\rangle = |q_2, q_1, q_0\rangle = |q_2 q_1 q_0\rangle = |\sum_{b=0}^{2} q_b 2^b\rangle.$$

most significant qubit

least significant qubit

- By convention, lexicographical ordering is used:

$$\begin{pmatrix} |0,0,0\rangle \\ |0,0,1\rangle \\ |0,1,0\rangle \\ |0,1,1\rangle \\ |1,0,0\rangle \\ |1,0,1\rangle \\ |1,1,0\rangle \\ |1,1,1\rangle \end{pmatrix} = \begin{pmatrix} |000\rangle \\ |001\rangle \\ |010\rangle \\ |011\rangle \\ |100\rangle \\ |101\rangle \\ |110\rangle \\ |111\rangle \end{pmatrix} = \begin{pmatrix} |0\rangle \\ |1\rangle \\ |2\rangle \\ |3\rangle \\ |4\rangle \\ |5\rangle \\ |6\rangle \\ |7\rangle \end{pmatrix}$$

Degrees of Freedom for Pure States

- A single qubit state $|\Psi\rangle = c_0|0\rangle + c_1|1\rangle$ is described by two complex-valued coefficients. These must lie on the unit circle $|c_0|^2 + |c_1|^2 = 1$. A global phase factor is irrelevant. Therefore it has two degrees of freedom (two free real values).
- An *N* qubit state

$$|\Psi\rangle = \sum_{k=0}^{2^N-1} c_k|k\rangle$$

with $\sum_k |c_k|^2 = 1$ therefore has $2 \cdot 2^N - 2$ degrees of freedom.

- However, there is Holevo‘s bound: You can maximally access *N* classical bits of this information.

Hadamard Gate

$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} |0\rangle \\ |1\rangle \end{pmatrix}$

- **Switches to Hadamard base**

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$$

- **Generates all basis values with equal amplitude**

$$H^{\otimes n}|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} |y\rangle$$

- **For x ∈ {0, 1}**

$$H|x\rangle = \frac{1}{\sqrt{2}} \sum_{y \in \{0,1\}} (-)^{x \cdot y} |y\rangle$$

- **For x ∈ {0, 1}*ⁿ***

$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-)^{x \cdot y} |y\rangle$$

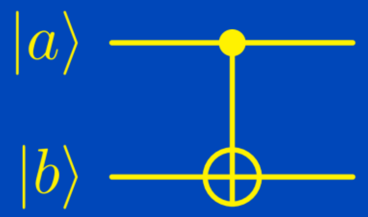
CNOT Gate

$$P_{\text{CNOT}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \overset{|ab\rangle}{\begin{pmatrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{pmatrix}}$$

- **Two qubits *a* and *b***

$$\text{CNOT}(|a, b\rangle) = |a, a \oplus b\rangle$$

XOR or
plus modulo 2



$$\begin{aligned} &\text{CNOT}(c_{00}|00\rangle + c_{01}|01\rangle + c_{10}|10\rangle + c_{11}|11\rangle) \\ &= \; c_{00}|00\rangle + c_{01}|01\rangle + c_{10}|11\rangle + c_{11}|10\rangle \end{aligned}$$

- **Note:**
  - **NOT** $a \oplus 1 = \bar{a}$
  - **XOR** $a \oplus b$

Toffoli Gate (CCNOT)

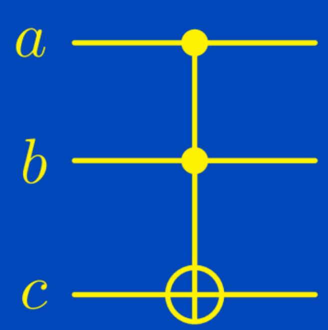
$$P_{\text{Toffoli}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \cdot \overset{|abc\rangle}{\begin{pmatrix} |000\rangle \\ |001\rangle \\ |010\rangle \\ |011\rangle \\ |100\rangle \\ |101\rangle \\ |110\rangle \\ |111\rangle \end{pmatrix}}$$

- **Three qubits *a*, *b* and *c***

$$\text{Toff}(|a, b, c\rangle) = |a, b, c \oplus (a \wedge b)\rangle$$

XOR or plus mod 2 ($\oplus$); AND or times ($\wedge$)

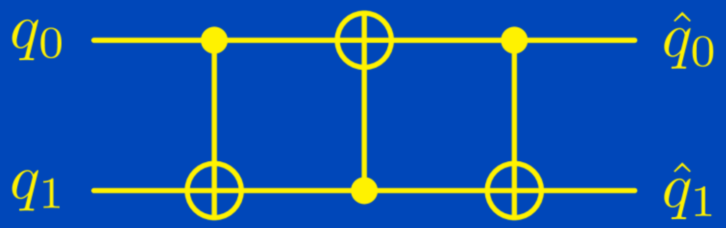
$$\text{Toff}\left(\begin{pmatrix} a \\ b \\ c \end{pmatrix}\right) = \begin{pmatrix} a \\ b \\ c \oplus (a \wedge b) \end{pmatrix}$$



- **Note:**
  - **AND** $0 \oplus (a \wedge b) = a \wedge b$
  - **NOT** $1 \oplus (a \wedge 1) = \bar{a}$
  - **OR requires four Toffoli gates (NOT, NOT, AND, NOT), since** $a \vee b = \overline{\bar{a} \wedge \bar{b}}$ **.**
  - **XOR** $c \oplus (1 \wedge b) = c \oplus b$

SWAP Gate

$$\mathrm{SWAP}(|a, b\rangle) = |b, a\rangle$$



q_0 in	q_1 in	q_0 out	q_1 out
0	0	0	0
0	1	1	0
1	0	0	1
1	1	1	1

Short notation:

Note:

A SWAP gate implements a global permutation of the computational basis, not a local memory access.

E.g., swapping qubits 1 and 2 maps the basis state

|5⟩ = |101⟩

to the basis state

|3⟩ = |011⟩.

ISING MODEL

Ising Model and Quantum Annealing

- The Ising model is a universal formulation of optimization problems as energy minimization of binary variables – originally for magnetism, today the backbone of quantum optimization.
- Find $s_i \in \{-1, +1\}$ that minimizes the Hamiltonian

$$H_\mathrm{I}(s) = \sum M_{ij} s_i s_j + \sum m_i s_i$$

In ideal quantum annealing, the ground-state bitstring is obtained deterministically.

In real devices, repeated runs are needed because they are noisy, thermal, or may have degenerate ground states.

- Can be directly … by setting the matrix values {… and running the solver using a … schedule:

$$H(t) = \alpha(t) H_0 + \beta(t) H_\mathrm{I} = \alpha(t) \sum_i \sigma_i^x + \beta(t) \Big(\sum_{i<j} M_{ij} \sigma_i^z \sigma_j^z + \sum_i m_i \sigma_i^z \Big)$$

annealing term (decreases in time) — Ising term (increases in time)

Pauli operators

$$\sigma^x|0\rangle = |1\rangle$$
$$\sigma^x|1\rangle = |0\rangle$$
$$\sigma^z|0\rangle = |0\rangle$$
$$\sigma^z|1\rangle = -|1\rangle$$

Terms must not commute. Annealing requires a non-commuting driver. X is the simplest and physically natural choice. It represents quantum tunneling between |0⟩ and |1⟩. $[\sigma^x, \sigma^z] = 2i\sigma^y \neq 0$

The device is cooled to mK temperatures.
At $t = 0$ only the transverse-field Hamiltonian H_0 is active. The system relaxes thermally into its ground state $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$.

Quantum Annealing vs. Gate-Based QC

Quantum Annealing	Gate-Based Quantum Computing
Superposition is the thermal ground state, e.g. strong transverse field	Superposition created by gates, e.g. Hadamards on each qubit
Analog Hamiltonian physics	Digital control
Ising is the hardware that works by adiabatic annealing $\|\Psi(T)\rangle = \mathcal{T}\exp\left(-i\int_0^T H(t)dt\right)\|\Psi(0)\rangle$	Ising needs to be emulated as a quantum operator (of depth *L*) … $\|\Psi(T)\rangle = \mathcal{T}\exp\left(-i\sum_{l=1}^{L}(\alpha_l H_0 + \beta_l H_\mathrm{I})\right)\|\Psi(0)\rangle$ … and to be iteratively updated by invoking several runs of the quantum program, i.e. QAOA.
Physics-guided with very limited freedom.	Freedom with very little physics guidance.
More or less restricted to Ising	Not restricted to Ising **General Hamiltonians** (Heisenberg, Hubbard, quantum chemistry, lattice gauge theories) – **True quantum dynamics** (Real-time evolution, spectra, transitions, scattering) – **Quantum algorithms (**Phase estimation, amplitude amplification, linear systems) – **Scalability** (digital error correction and, potentially, fault tolerance) – …

$\mathcal{T}$ = time ordering operator, QAOA = quantum approximate optimization algorithm

Ising Model and QUBO

- The Ising model is a universal formulation of spin energies and spin spin interactions. With $s_i \in \{-1, +1\}$ the Hamiltonian is

$$H_{\mathrm{I}}(s) = \sum_{i<j} M_{ij} s_i s_j + \sum_i m_i s_i$$

- Quadratic unconstrained binary optimization (QUBO) is the search for $b_i \in \{0, 1\}$ that minimize the cost function

$$C(b) = \sum_{i<j} K_{ij} b_i b_j + \sum_i k_i b_i$$

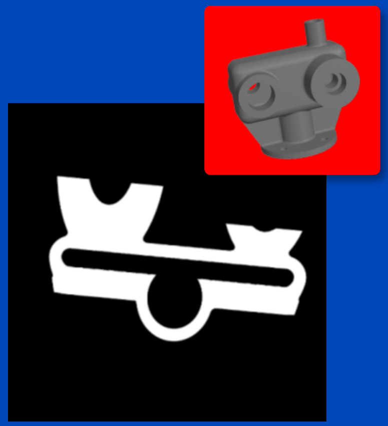
- Both problems are equivalent (up to an irrelevant constant) and therefore a quantum system realizing H_{I} may be useful to minimize C.

SOME EXAMPLES IN OUR FIELD

Binary Tomography

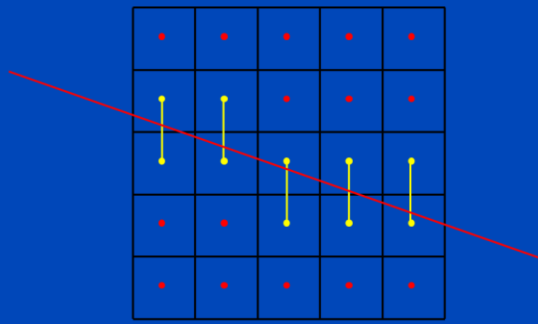
- Binary image $b(x, y) = b_i \in \{0, 1\}$. X-ray transform operator X. Projection data p with

$$\boldsymbol{p} = \boldsymbol{X} \cdot \boldsymbol{b}$$



- Find b by minimizing

$$C_1 = (\boldsymbol{X} \cdot \boldsymbol{b} - \boldsymbol{p})^2 = \boldsymbol{b}^{\mathrm{T}} \cdot \boldsymbol{X}^{\mathrm{T}} \cdot \boldsymbol{X} \cdot \boldsymbol{b} - 2\,\boldsymbol{p}^{\mathrm{T}} \cdot \boldsymbol{X} \cdot \boldsymbol{b} + \boldsymbol{p}^2 .$$



Joseph's forward projector

- If we want to find the sparsest solution, also minimize

$$C_2 = \boldsymbol{1}^{\mathrm{T}} \cdot \boldsymbol{b} = \sum_j b_j .$$

- Minimizing $C_1 + \lambda\, C_2$ means

$$\boldsymbol{b} = \arg\min_{\boldsymbol{b}} \left(\boldsymbol{b}^{\mathrm{T}} \cdot \boldsymbol{X}^{\mathrm{T}} \cdot \boldsymbol{X} \cdot \boldsymbol{b} - 2\,\boldsymbol{p}^{\mathrm{T}} \cdot \boldsymbol{X} \cdot \boldsymbol{b} + \lambda \boldsymbol{1}^{\mathrm{T}} \cdot \boldsymbol{b}\right) = \arg\min_{\boldsymbol{b}} \left(\sum_{ij} b_i K_{ij} b_j + \sum_i k_i b_i\right)$$

This is solvable by quadratic unconstrained binary optimization (QUBO).

Enforcing Gradient Sparsity
in Binary Tomography

- Instead of minimizing the L_0 norm of the image we could also minimize the gradient‘s L_0 norm, and thereby enforce homogeneous image regions with few edges in-between.
- Let *i* index a pixel and *j* index a pixel in its neighborhood $\mathcal{N}$. Then

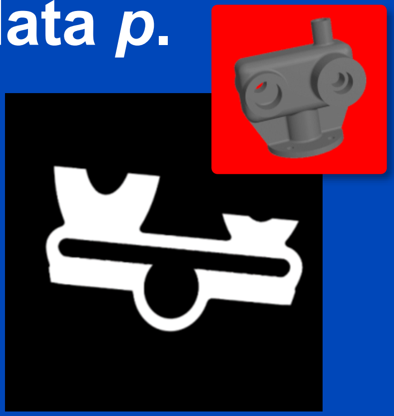
$$C_2 = \sum_i \sum_{j \in \mathcal{N}(i)} (b_i - b_j)^2 = \sum_i \sum_{j \in \mathcal{N}(i)} (b_i - 2b_i b_j + b_j)$$

would be an adequate cost term, counting the pixels at object edges.

Binary Tomography with Projection Selection

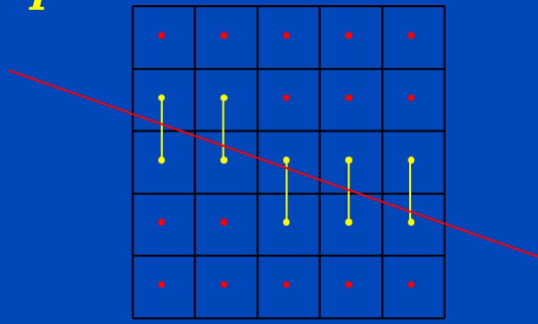
- **Binary volume $f_i \in \{0, 1\}$. X-ray transform operator X. Projection data p. Projection selection binaries $s_n \in \{0, 1\}$.**

$$\boldsymbol{S} \cdot \boldsymbol{p} = \boldsymbol{S} \cdot \boldsymbol{X} \cdot \boldsymbol{f} \qquad \boldsymbol{S} = \operatorname{diag} \boldsymbol{s} = \begin{pmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & s_N \end{pmatrix}$$



- **Find f_i and s_n by minimizing**

$$C_1 = (\boldsymbol{S} \cdot \boldsymbol{X} \cdot \boldsymbol{f} - \boldsymbol{S} \cdot \boldsymbol{p})^2 = \boldsymbol{f}^{\mathrm{T}} \cdot \boldsymbol{X}^{\mathrm{T}} \cdot \boldsymbol{S} \cdot \boldsymbol{X} \cdot \boldsymbol{f} - 2\,\boldsymbol{p}^{\mathrm{T}} \cdot \boldsymbol{S} \cdot \boldsymbol{X} \cdot \boldsymbol{f} + \boldsymbol{p}^2$$



Joseph's forward projector

- **To enforce gradient and view sparsity, also minimize**

$$C_2 = \sum_i \sum_{j \in \mathcal{N}(i)} (f_i - f_j)^2 \qquad C_3 = \sum_n s_n$$

- **Minimizing $C_1 + \lambda_2 C_2 + \lambda_3 C_3$ means (after renaming binaries f_i and s_n)**

$$\boldsymbol{b} = \arg\min_{\boldsymbol{b}} \Big(\sum_{ijk} A_{ijk} b_i b_j b_k + \sum_{ij} A_{ij} b_i b_j + \sum_i A_i b_i \Big)$$

This is cubic and therefore not (yet) solvable by quadratic unconstrained binary optimization (QUBO).

Making Cubic Terms be Quadratic

- Let *a*, *b* and *c* be binary variables.
- The ancilla trick with a quadratic penalty does not work:

$$\arg\min_{a,b,c} abc = \arg\min_{a,b,c,d} \left(ad + \lambda(\underbrace{d - bc}_{\text{unfortunately still cubic}})^2\right)$$

- Instead, one uses a different penalty:

$$\arg\min_{a,b,c} abc = \arg\min_{a,b,c,d} \left(ad + \lambda(\underbrace{bc - 2bd - 2cd + 3d}_{\text{0 iff } d=bc\text{, otherwise positive}})\right)$$

b	c	d	$bc - 2bd - 2cd + 3d$
0	0	0	0 i.e. $bc = d$
0	0	1	3 i.e. $bc \neq d$
0	1	0	0 i.e. $bc = d$
0	1	1	1 i.e. $bc \neq d$
1	0	0	0 i.e. $bc = d$
1	0	1	1 i.e. $bc \neq d$
1	1	0	1 i.e. $bc \neq d$
1	1	1	0 i.e. $bc = d$

Ternary Tomography with One-Hot Coding

- Ternary volume $f_i \in \{v_1, v_2, v_3\}$, binary $o_{gi} \in \{0, 1\}$ for one-hot coding:

$$f_i = \sum_{g=1}^{3} v_g o_{gi} \text{ with } \sum_{g=1}^{3} o_{gi} = 1 \text{ enforced by penalty } (o_{1i} + o_{2i} + o_{3i} - 1)^2$$

- X-ray transform operator X. Projection data p with $\boldsymbol{p} = \boldsymbol{X} \cdot \boldsymbol{f}$.
- Find f by minimizing

$$C_1 = \sum_n \Big(\sum_i X_{ni} f_i - p_n\Big)^2$$

- Enforce gradient sparsity

$$C_{\hat{2}} = \sum_i \sum_{j \in \mathcal{N}(i)} (f_i - f_j)^2 \qquad C_2 = \sum_i \sum_{j \in \mathcal{N}(i)} \Big(1 - \sum_g o_{gi} o_{gj}\Big)$$

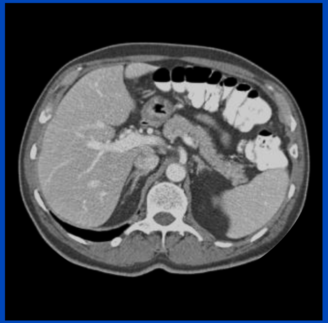
- To find the desired o_{gi}, minimize

$$C = \lambda_{\text{one}} \sum_i \Big(\sum_g o_{gi} - 1\Big)^2 + \lambda_{\text{raw}} \sum_n \Big(\sum_i X_{ni} \sum_g v_g o_{gi} - p_n\Big)^2 + \lambda_{\text{grad}} \sum_i \sum_{j \in \mathcal{N}(i)} \Big(1 - \sum_g o_{gi} o_{gj}\Big)$$

Integer and Real Tomography

- **Integer- or real-valued image *f*(*x, y*). Radon transform operator *R*. Projection data *p* with**

$$\boldsymbol{p} = \boldsymbol{R} \cdot \boldsymbol{f}$$



- **Find *f* by minimizing**

$$C_1 = (\boldsymbol{R} \cdot \boldsymbol{f} - \boldsymbol{p})^2 = \boldsymbol{f}^{\mathrm{T}} \cdot \boldsymbol{R}^{\mathrm{T}} \cdot \boldsymbol{R} \cdot \boldsymbol{f} - 2\,\boldsymbol{p}^{\mathrm{T}} \cdot \boldsymbol{R} \cdot \boldsymbol{f} + \boldsymbol{p}^2.$$

- **Replace f_i by its binary expansion**

$$f_i = c \sum_{p=0}^{P-1} b_{ip} 2^p$$

- **Minimizing C_1 (plus other quadratic constraints $C_2 + C_3 + \ldots$) means**

$$\boldsymbol{b} = \arg\min_{\boldsymbol{b}} \Big(\sum_{ij} b_i K_{ij} b_j + \sum_i k_i b_i \Big)$$

Enforcing Pixel Sparsity
in Integer or Real Tomography

- Starting from

$$f_i = c \sum_{p=0}^{P-1} b_{ip} 2^p$$

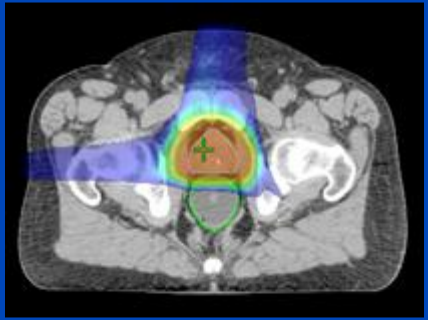
- To enforce pixel sparsity it is not sufficient to minimize the sum over all b_{ip}. Instead, we need to minimize

$$C_{\hat{2}} = \sum_i \min(1, \sum_p b_{ip})$$

- Since this is a hard constraint and not (yet) quadratic, we introduce binary slacks and minimize

$$C_2 = \sum_i \left(\hat{b}_i + \eta((1 - \hat{b}_i) \sum_p b_{ip}\right).$$

RT Treatment Plan Optimization
(or CT Dose Minimization)



- X-ray or treatment beam index *i* (orientation, energy, intensity, ...)
- Beam off or on: $b_i \in \{0, 1\}$
- Voxel index *j*
- Dose d_{ij} of beam *i* to voxel *j*.
- Prescribed target dose minimum D^{T} and organ at risk maximum D^{R}
- The optimal treatment is given by

$$\boldsymbol{b} = \arg\min_{\boldsymbol{b}} \Big(\sum_{j \in T} \max(0, D_j^{\mathrm{T}} - \sum_i b_i d_{ij}) + \sum_{j \in R} \max(0, \sum_i b_i d_{ij} - D_j^{\mathrm{R}}) \Big)$$

- How can this be converted into a QUBO problem?

Convert Hard Constraints into QUBO

- A sub problem of the previous slide is the constrained problem

$$\boldsymbol{b} = \arg\min_{\boldsymbol{b}} \max(0, K - \sum_i k_i b_i)$$

- We introduce two slack variables *u* ≥ 0 and *v* ≥ 0

$$u - v = K - \sum_i k_i b_i$$

- Since neither *u* nor *v* must be negative, we have

$$u = \max(0, K - \sum_i k_i b_i) \quad \textbf{and} \quad v = \max(0, \sum_i k_i b_i - K)$$

- Our sub problem can now be formulated as QUBO

$$\boldsymbol{b} = \arg\min_{\boldsymbol{b},\boldsymbol{u},\boldsymbol{v}} \left(u + \lambda\Big((u - v) - (K - \sum_i k_i b_i)\Big)^2\right)$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = c \sum_{p=0}^{P-1} \begin{pmatrix} u_p \\ v_p \end{pmatrix} 2^p$$

$$u_p, v_p \in \{0, 1\}$$

QUANTUM HARDWARE

Quantum Hardware Today

- Today's quantum hardware splits into
  - large-scale analog annealers (Ising problems)
  - smaller but universal gate-based processors

Quantum Annealing Computers

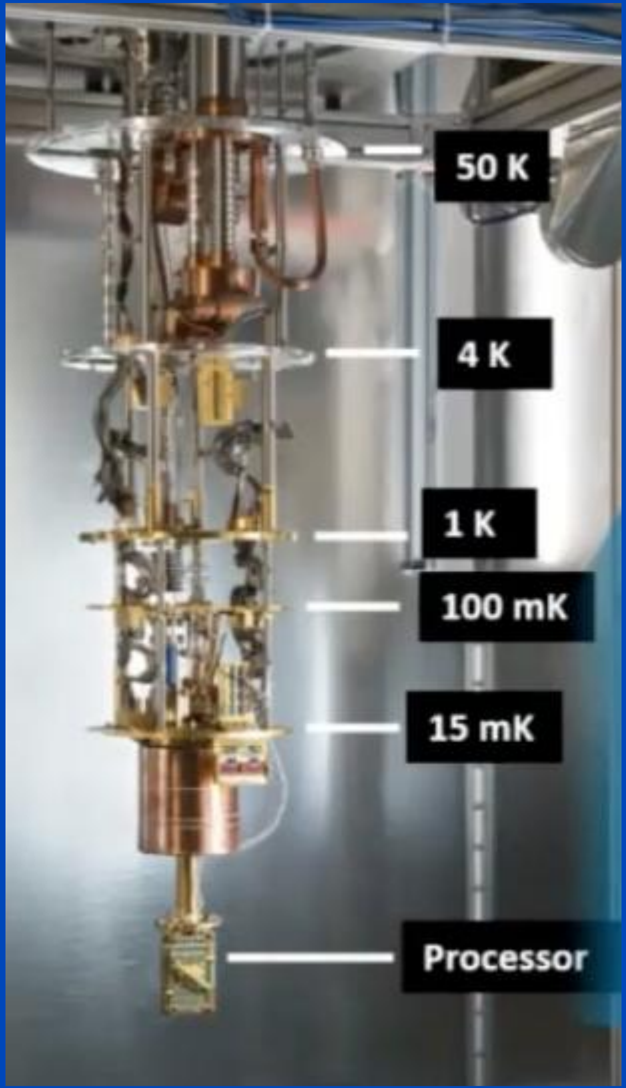
- Computational model
  - Based on the Ising Hamiltonian (or QUBO formulation)
  - Evolves adiabatically from an initial Hamiltonian H_0 to a problem-specific Hamiltonian H_I.
  - Aims at finding low-energy configurations

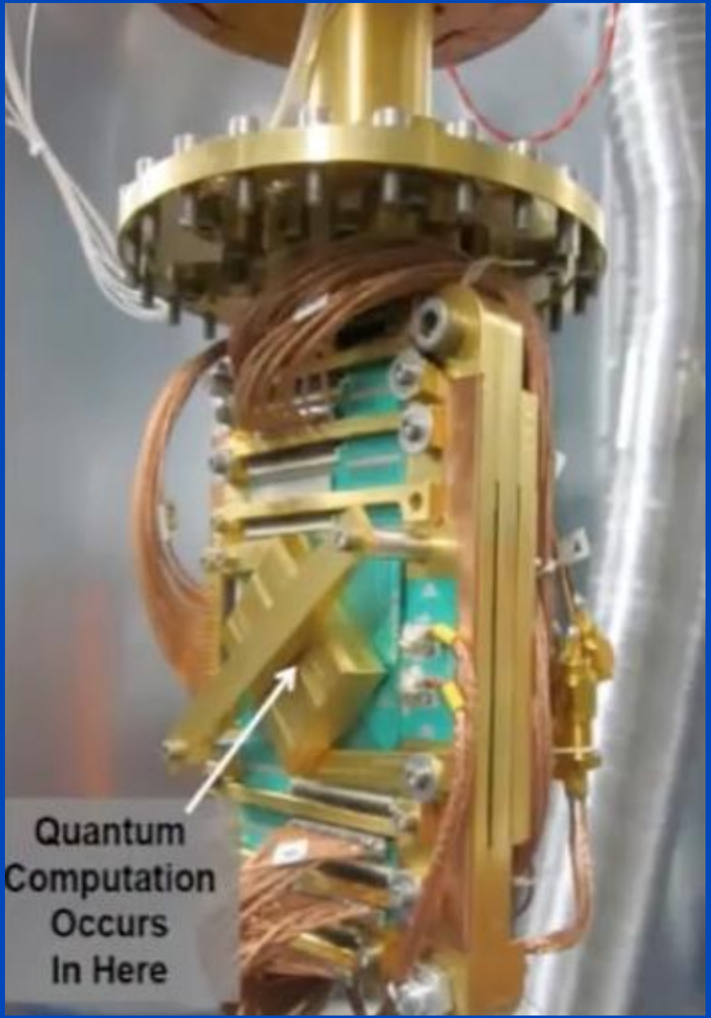
$$H_{\mathrm{I}} = \sum_{i<j} M_{ij}\sigma_i^z\sigma_j^z + \sum_i m_i\sigma_i^z$$

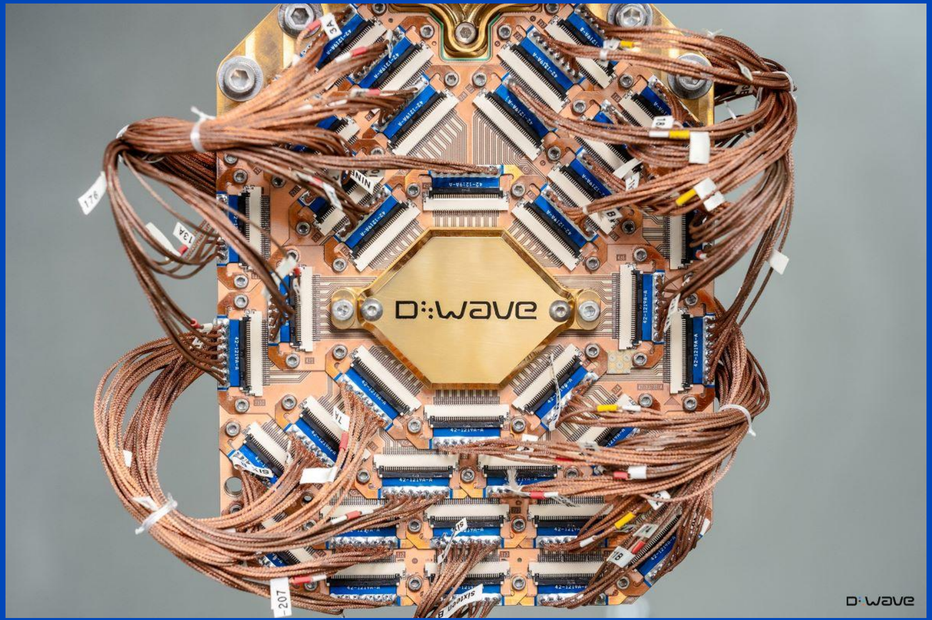
- Not a universal quantum computer
  - No execution of arbitrary quantum circuits.
- Typical hardware characteristics
  - “Large” number of physical qubits (thousands)
  - Sparse but structured connectivity graph
  - Analog control parameters M_{ij} and m_i
  - Noise and effective temperature are intrinsic to operation

Quantum Annealing Computers

- Key parameters
  - Number of physical qubits
  - Connectivity degree (e.g. 15-way, 20-way)
  - Coupler precision and range
  - Annealing time (typically μs to ms)
  - Effective temperature / noise floor
  - Embedding overhead (logical variables → chains of physical qubits)
  - Hybrid quantum and classical workflows
- Representative system
  - D-Wave Advantage (Pegasus topology)
    - » 5000 superconducting flux qubits
    - » Pegasus graph with ~15 connections per qubit
    - » Operated via cloud
- Annealers trade universality for scale.


50 K
4 K
1 K
100 mK
15 mK
Processor


Quantum
Computation
Occurs
In Here



D-Wave

Gate-Based Quantum Computers

- Computational model
  - Quantum circuit model
  - Computation defined by sequences of 1- and 2-qubit gates, followed by measurement
  - Universal for quantum algorithms (QFT, Grover, VQE, QAOA, error correction, …)

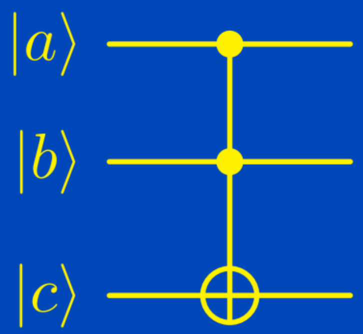
$$|\psi_{\text{out}}\rangle = U_N \cdots U_3 U_2 U_1 |\psi_{\text{in}}\rangle$$

- Core hardware requirements
  - High-fidelity single- and two-qubit gates
  - Stable coherence
  - Precise calibration and crosstalk suppression
  - Measurement and reset (often mid-circuit)
  - Clear scaling path toward quantum error correction (QEC)

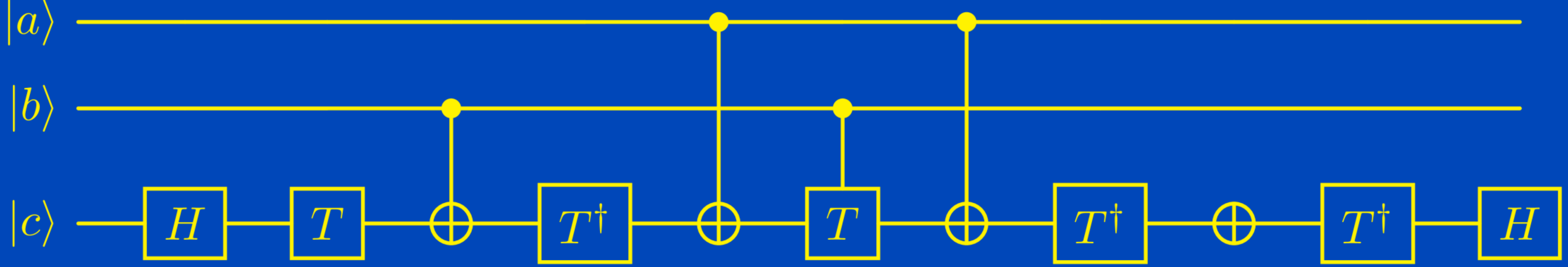
$$T = R_z(\frac{\pi}{4}) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

Gate-Based Quantum Computers

- **Key performance parameters**
  - **Number of physical qubits**
  - **Connectivity graph (nearest-neighbor, heavy-hex, all-to-all)**
  - **Single-qubit gate fidelity**
  - **Two-qubit gate fidelity *(dominant bottleneck)***
  - **Readout fidelity**
  - **Coherence times (T_1, T_2)**
  - **Native gate set**



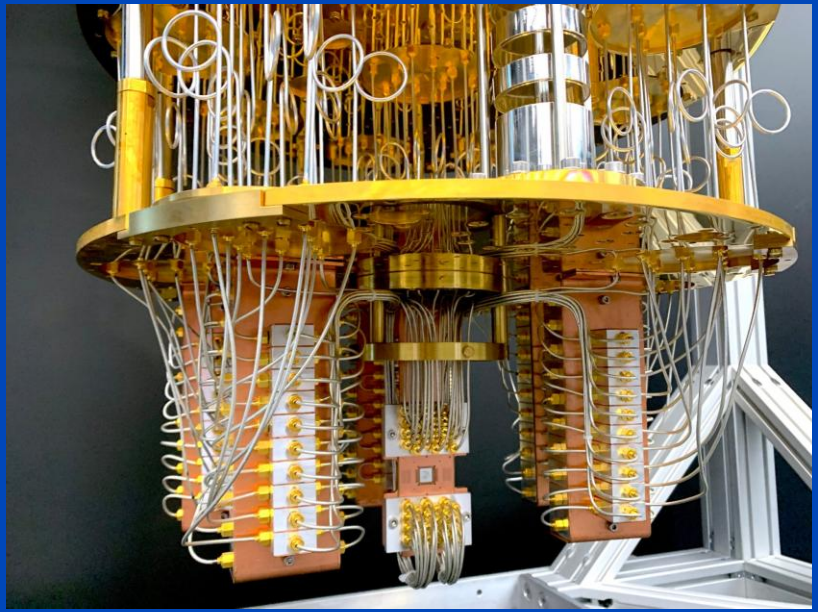
Toffoli gate

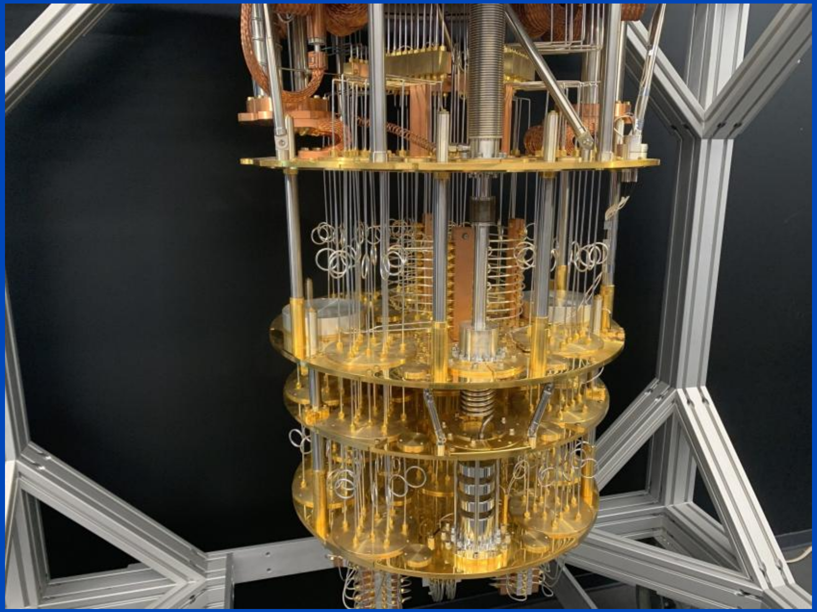


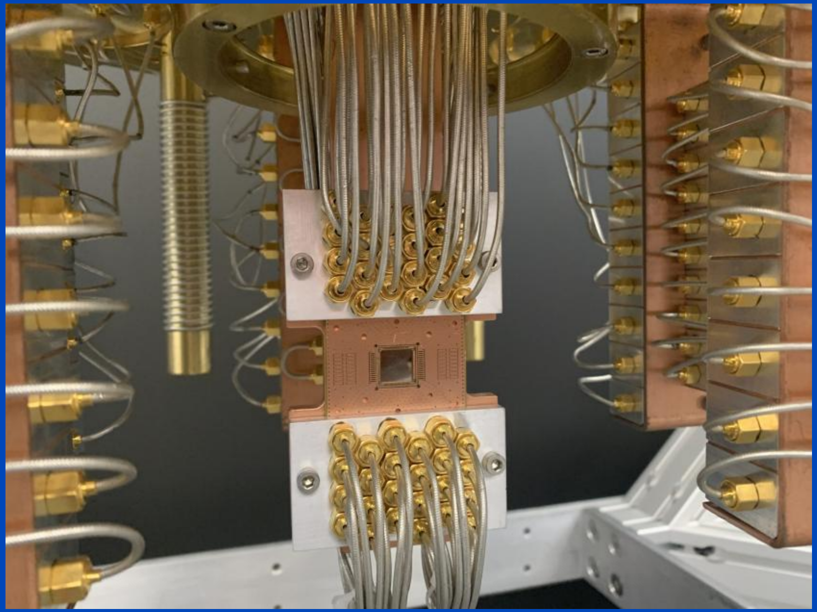
Toffoli gate decomposed into one- and two-qubit gates.

Superconducting Gate-Based Systems

- Characteristics
  - Fast gates (ns to µs)
  - Lithographically scalable
  - Limited connectivity (planar layouts)
  - Strong focus on surface-code-compatible architectures
- Example: IBM Quantum – Heron processors
  - 133 to 156 superconducting transmon qubits
  - Heavy-hex connectivity
  - Explicit system metrics (EPLG, CLOPS)
  - Integrated into modular “System Two” architecture
- These systems are controlled open quantum systems with strong microwave engineering constraints and continuous recalibration.







IBM Q

dkfz.

Trapped-Ion Quantum Computers

- Characteristics
  - Qubits: internal states of ions
  - All-to-all logical connectivity
  - Very high gate fidelities
  - Slower gate speeds (µs to ms)
  - Excellent platform for early logical qubits
- Example 1: Quantinuum H2
  - 56 trapped-ion qubits
  - All-to-all connectivity
  - Emphasis on fault-tolerant primitives
- Example 2: IonQ Forte Enterprise
  - Performance expressed in algorithmic qubits
  - Focus on usable circuit depth rather than raw qubit count

Neutral-Atom Platforms

- Characteristics
  - Qubits: neutral atoms in optical tweezers
  - Reconfigurable geometries
  - Often used for analog Hamiltonian simulation
  - Digital gate sets are emerging but still limited
- Example: QuEra Aquila
  - 256 neutral-atom qubits
  - Programmable interaction graphs
  - Strong overlap with many-body physics and quantum simulation
-

MORE DETAILS ON ISING

One Ising QAOA Layer for All *N* Qubits

$$
\begin{aligned}
U(\alpha, \beta) &= e^{-i\alpha H_0} e^{-i\beta H_\mathrm{I}} \\
&= \prod_i e^{-i\alpha \sigma_i^x} \prod_{i<j} e^{-i\beta M_{ij} \sigma_i^z \sigma_j^z} \prod_i e^{-i\beta m_i \sigma_i^z}
\end{aligned}
$$

Many Ising QAOA Layers to Add Depth

$$|\psi(\boldsymbol{\alpha}, \boldsymbol{\beta}\rangle = \Big(\prod_{l=1}^{L} U(\alpha_l, \beta_l)\Big) \, |+\rangle^{\otimes N}$$

How to Emulate the Ising Hamiltonian

- The Ising Hamiltonian, which is diagonal in the computational basis, is

$$H_{\mathrm{I}} = \sum_{i<j} M_{ij}\sigma_i^z\sigma_j^z + \sum_i m_i\sigma_i^z$$

- A gate-based quantum computer cannot “run” a Hamiltonian directly.
- Instead, it can implement unitary operators of the form $U = e^{-i\beta H}$.
- Due to $[\sigma_i^z, \sigma_j^z] = 0$ and $[\sigma_i^z, \sigma_j^z\sigma_k^z] = 0$ we get

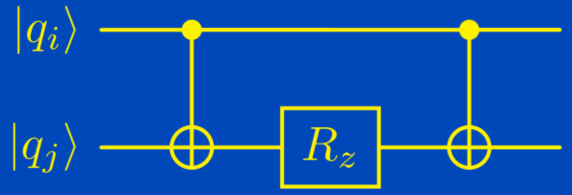
$$U = e^{-i\beta H_{\mathrm{I}}} = \prod_{i<j} e^{-i\beta M_{ij}\sigma_i^z\sigma_j^z} \prod_i e^{-i\beta m_i\sigma_i^z}$$

and therefore we can implement each term independently.

How to Emulate the Ising Hamiltonian

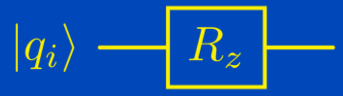
- **ZZ interaction, i.e coupling. Map parity onto qubit *j*, apply a phase, undo the parity mapping:**

$$e^{-i\beta M_{ij}\sigma_i^z\sigma_j^z} = \mathrm{CNOT}_{ij}\, R_z^{(j)}(2\beta M_{ij})\, \mathrm{CNOT}_{ij}$$



- **Z rotation (phase rotation, no entanglement, one gate per qubit):**

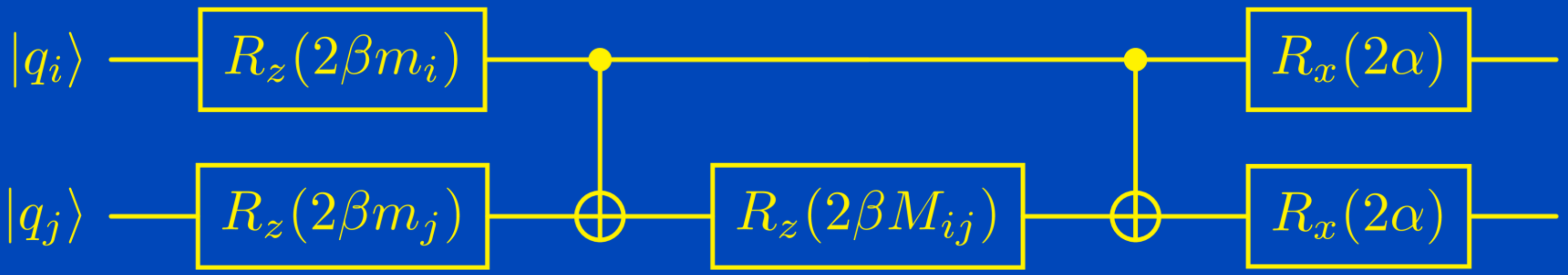
$$e^{-i\beta m_i\sigma_i^z} = R_z^{(i)}(2\beta m_i)$$



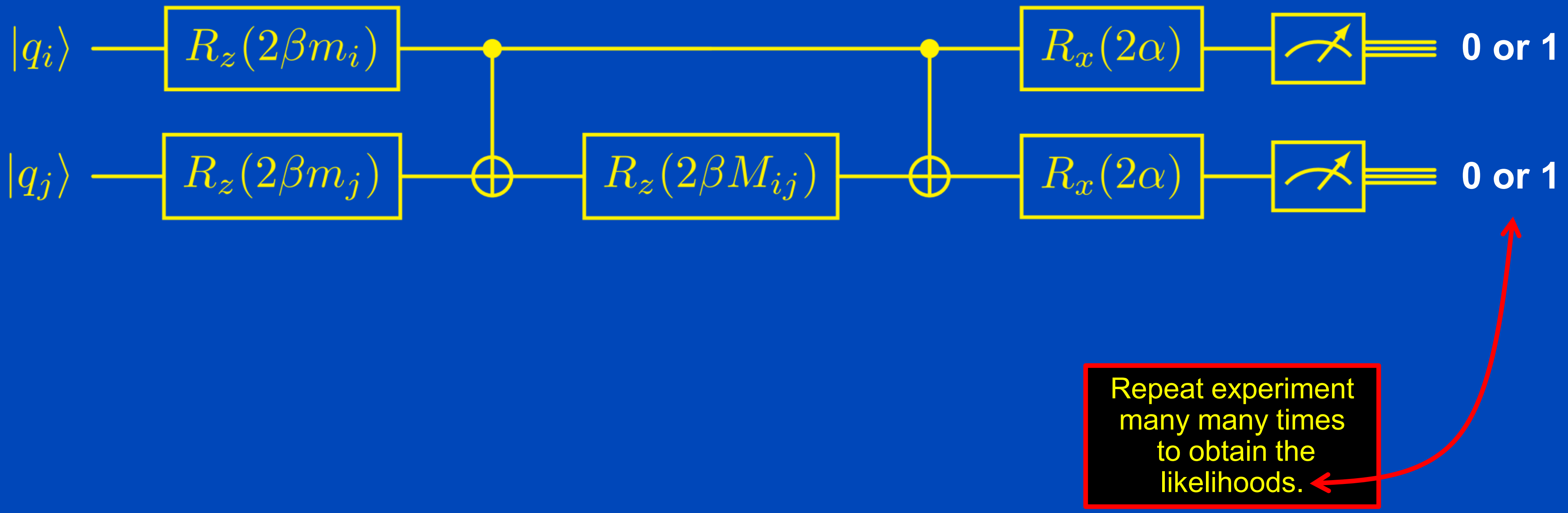
- **We also need to consider the mixing Hamiltonian $H_0 = \sum_i \sigma_i^x$ that flips bits, creates superpositions, plays the role of "quantum mixing":**

$$e^{-i\alpha H_0} = \prod_i e^{-i\alpha\sigma_i^x} = \prod_i R_x^{(i)}(2\alpha\sigma_i^x)$$

Ising Building Block for two Interacting Qubits

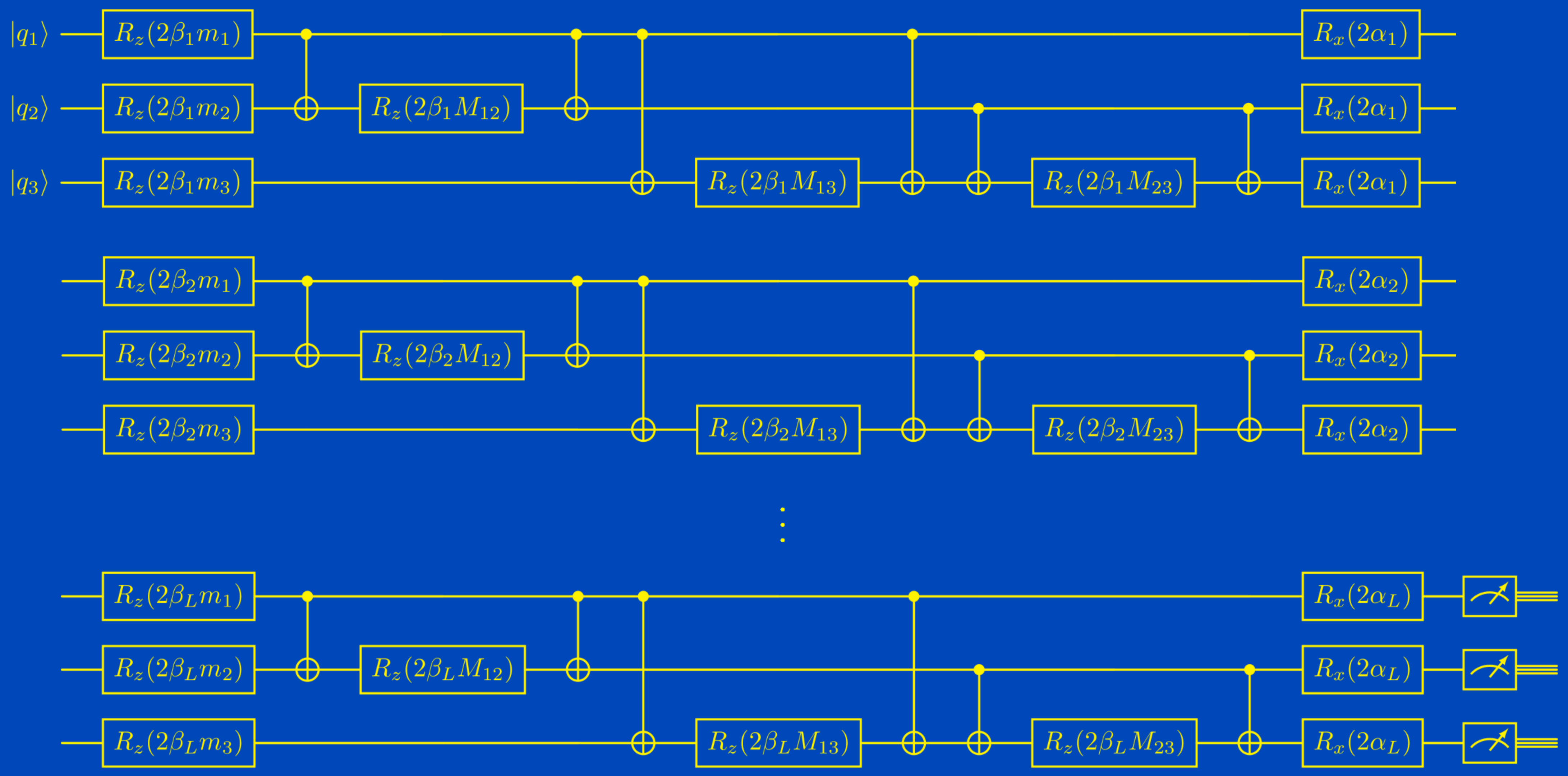


- Rotations are physically realized as carefully timed and phased control pulses.
- A rotation gate on a quantum computer is not applied as a matrix. It rather emerges from controlled time evolution under a physically engineered Hamiltonian.
- R_z is usually implemented virtually, without any physical pulse, i.e. z-rotations are software-defined. The phase reference frame of subsequent pulses is shifted.



$|q_i\rangle$
$|q_j\rangle$
$R_z(2\beta m_i)$
$R_z(2\beta m_j)$
$R_z(2\beta M_{ij})$
$R_x(2\alpha)$
$R_x(2\alpha)$
0 or 1
0 or 1
Repeat experiment many many times to obtain the likelihoods.

Example for 3 Qubits and *L* Layers



How to Run QAOA

- **Classical**
  - **Choose (α, β)**
- **Quantum** (repeat measurement *M* times):
  - **Initialize qubits** $|0\rangle^{\otimes N}$
  - **Create maximum superposition** $|+\rangle^{\otimes N} = H^{\otimes N} |0\rangle^{\otimes N}$
  - **Apply unitaries** $|\psi(\boldsymbol{\alpha}, \boldsymbol{\beta})\rangle = U_L \cdot \ldots \cdot U_2 U_1 \, |+\rangle^{\otimes N}$
  - **Measure** $\left\{ P_b = |b\rangle\langle b| \, \middle| \, b \in \{0,1\}^{\otimes N} \right\}$
  - **to obtain a bit string with *N* bits** $\boldsymbol{b} = \left(b_N, \ldots, b_2, b_1 \right)$
- **Classical** (draw conclusions from the *M* bit strings):
  - **Calculate the cost function, given the bit strings that have been obtained in the *M* runs with the latest (α, β) choice**

$$C(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{M} \sum_{m=1}^{M} C(\boldsymbol{b}_m)$$

  - **Use, e.g., Nelder Mead to update (α, β) and do another *M* quantum runs and measurements.**

dkfz.

Thank You!

This presentation will soon be available at www.dkfz.de/ct.

Job opportunities through marc.kachelriess@dkfz.de or through DKFZ’s international PhD or Postdoctoral Fellowship programs.

Parts of the reconstruction software were provided by RayConStruct® GmbH, Nürnberg, Germany.