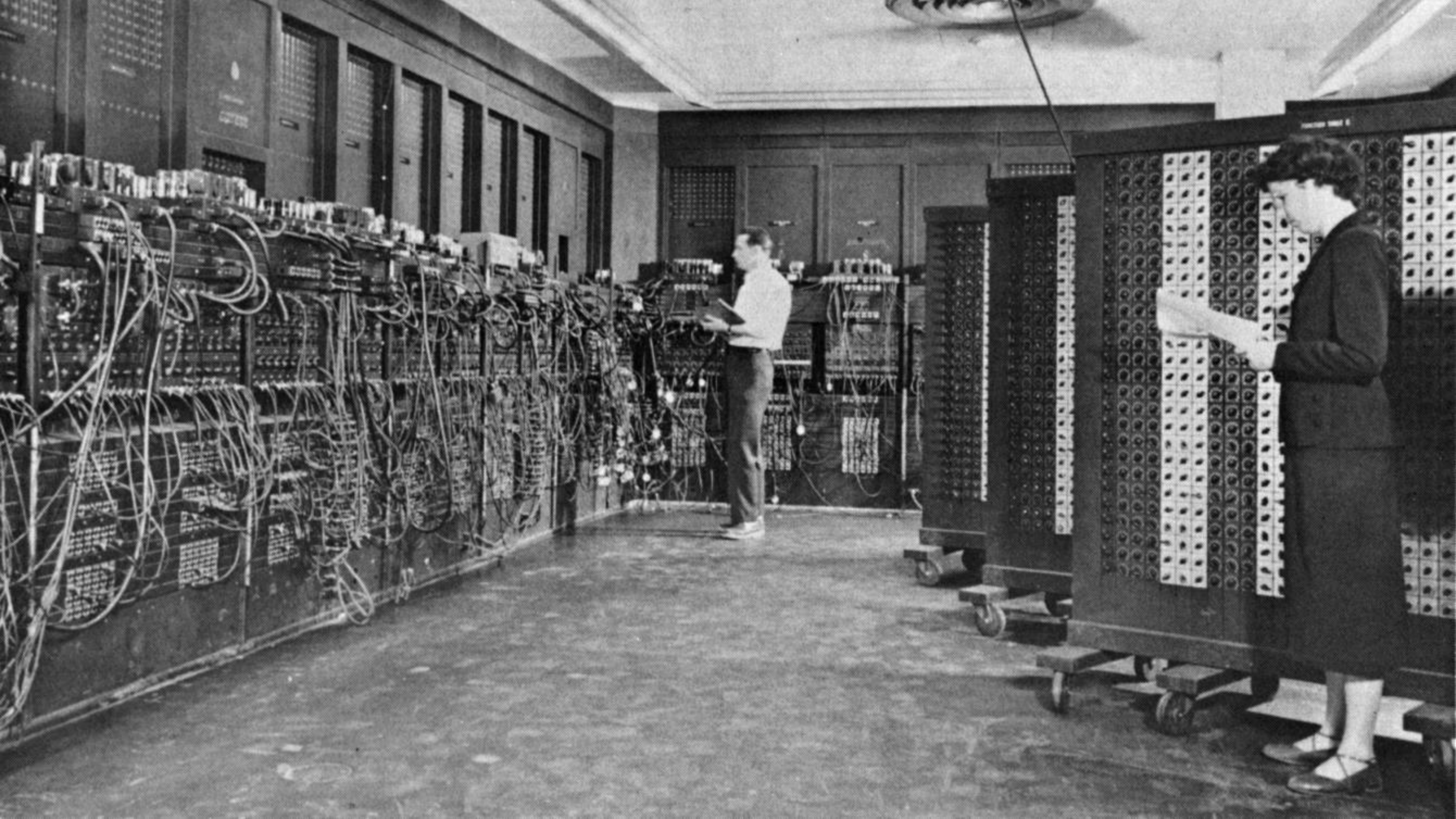
QUANTUM COMPUTING

Vision and Reality

Bettina Heim
Microsoft, QuArC



Ammonia production

Use: fertilizer, cleaning products…

Haber process: Nitrogen ⟶ Ammonia

- High temperature and pressure
- 1-2% of global energy

Nature via Bacteria

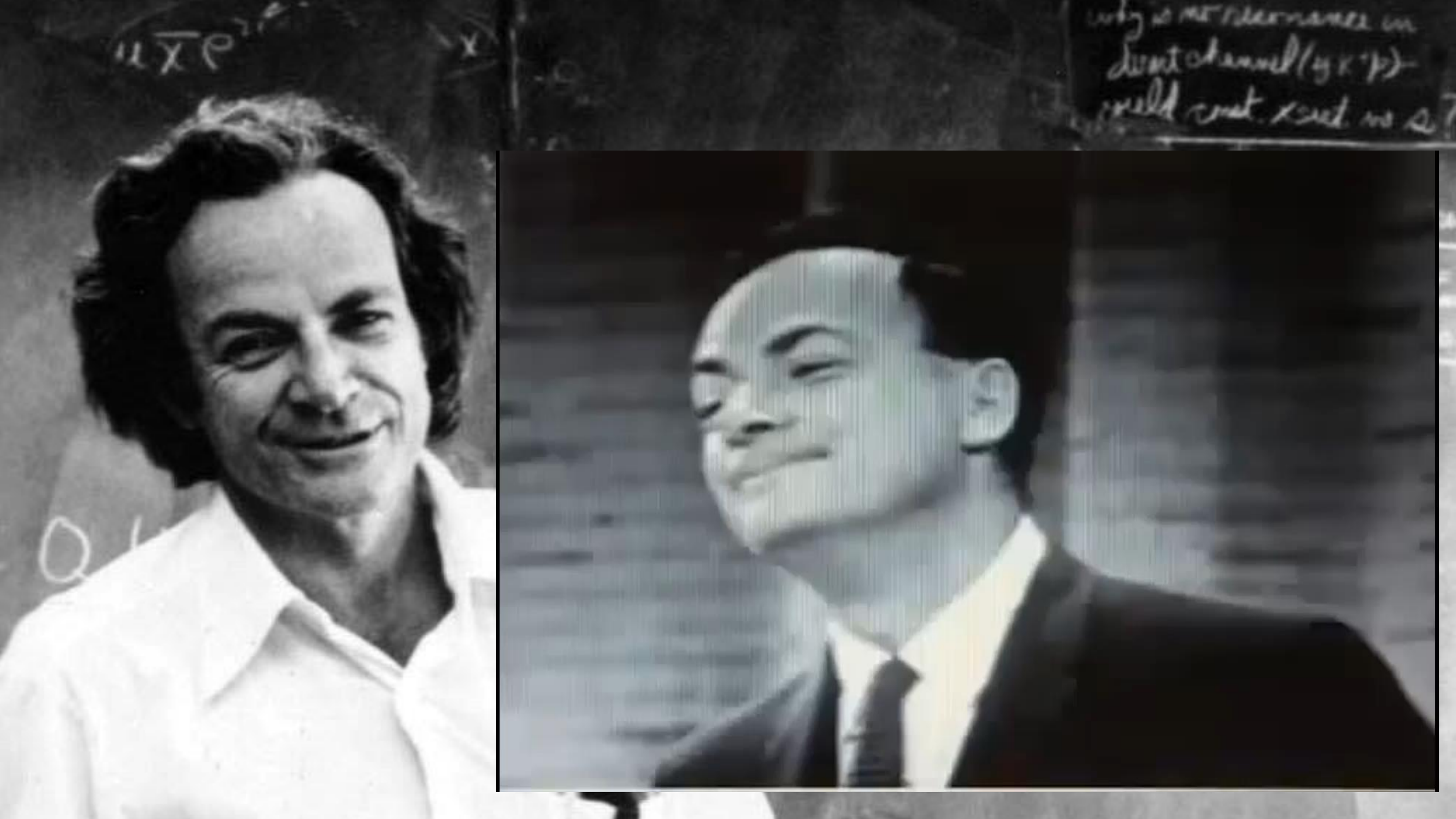
- Ambient temperature and pressure
- Catalyst: Nitrogenase enzyme

FeMoco molecule

Climate Change

Food Production

Antibiotic Resistance



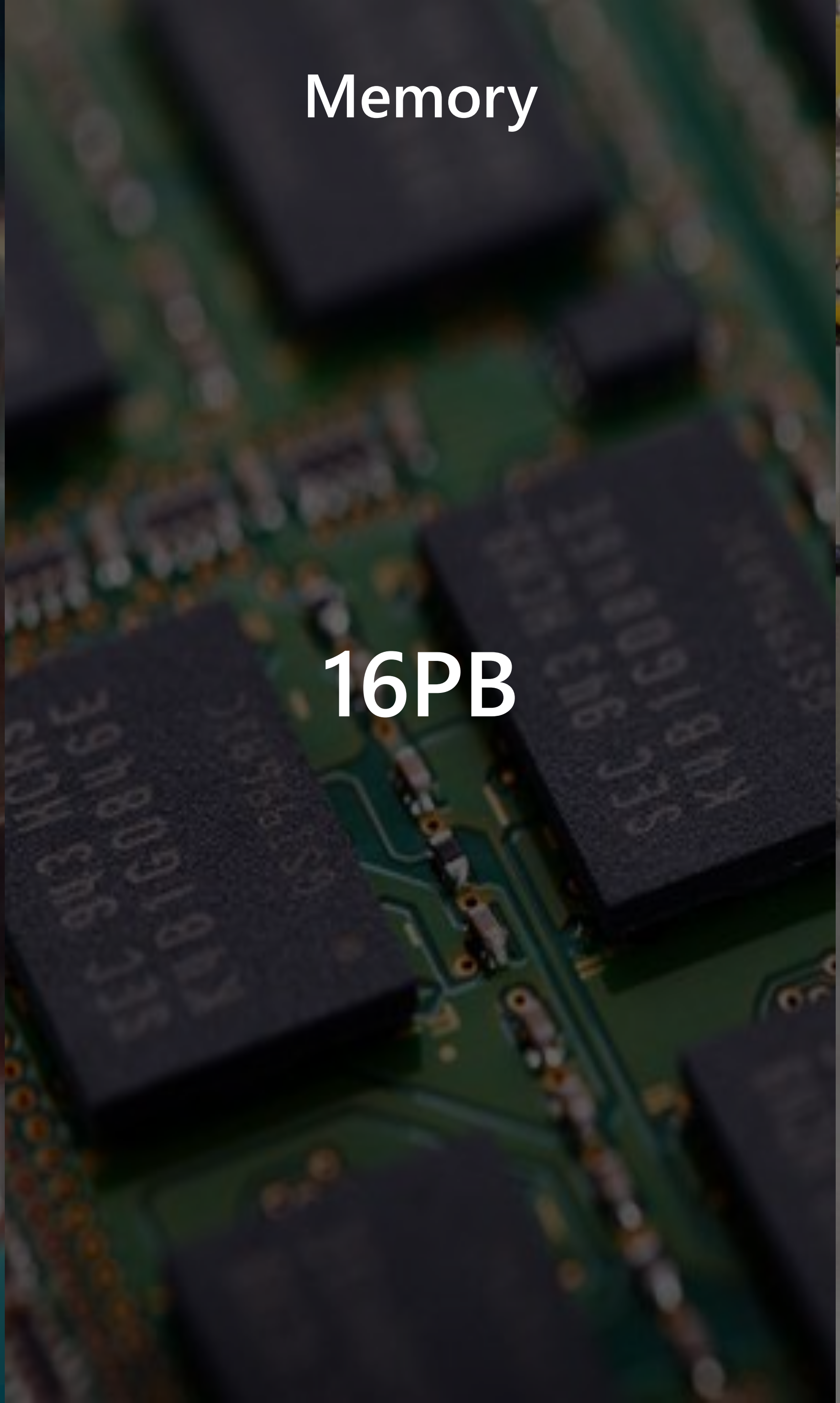

B-36
summit

What is a Qubit?

Mathematical Concept:

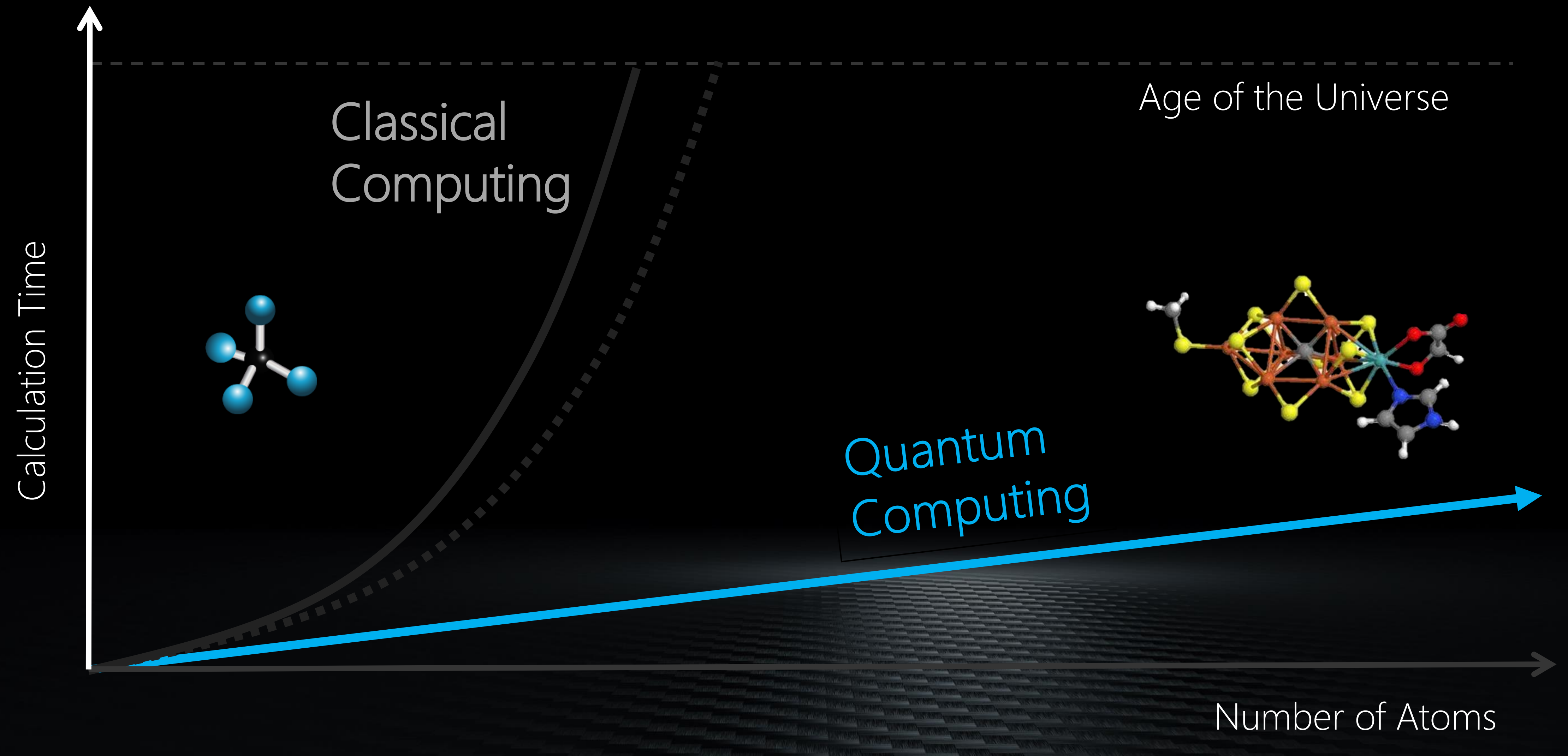
- 2-level quantum system
- each level is describe by a complex number


Qubits
50


Memory
16PB


Machine

250 qubits -
more states than atoms in the visible universe



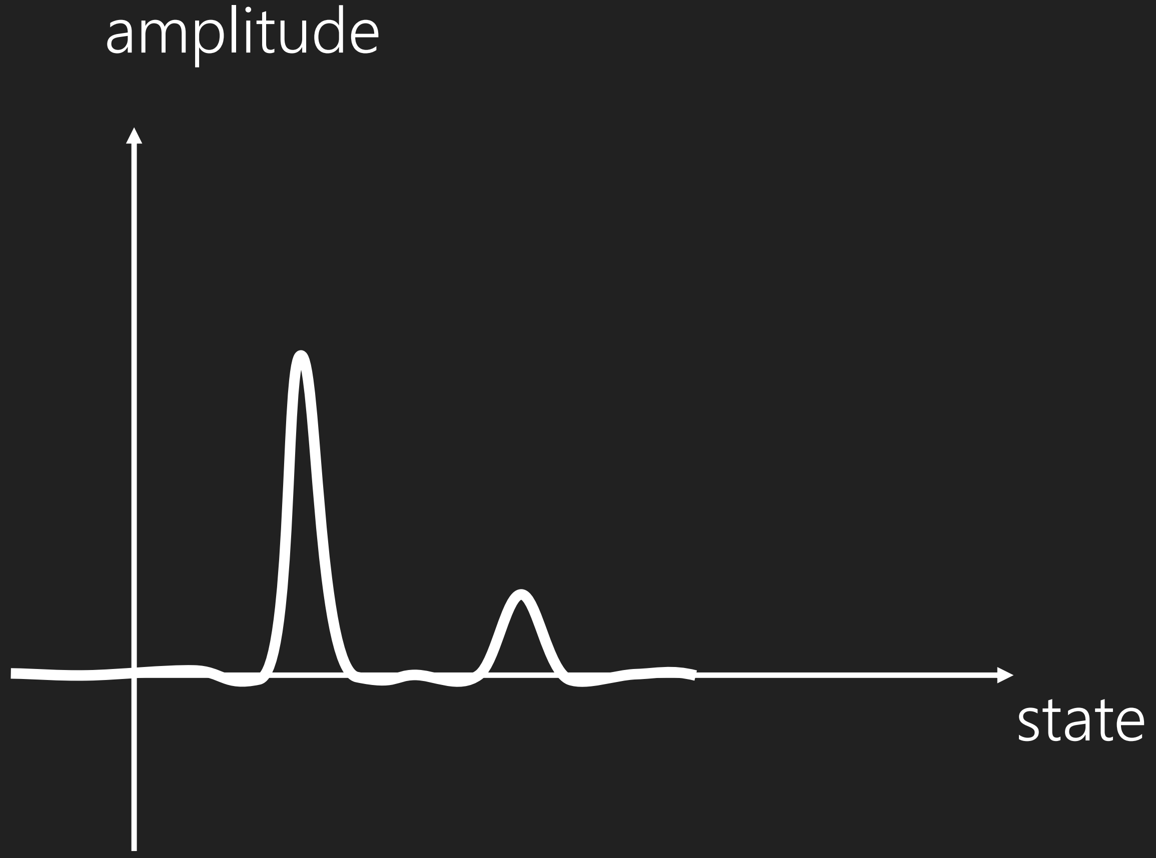
Addressing classically intractable problems

Quantum States & Interference

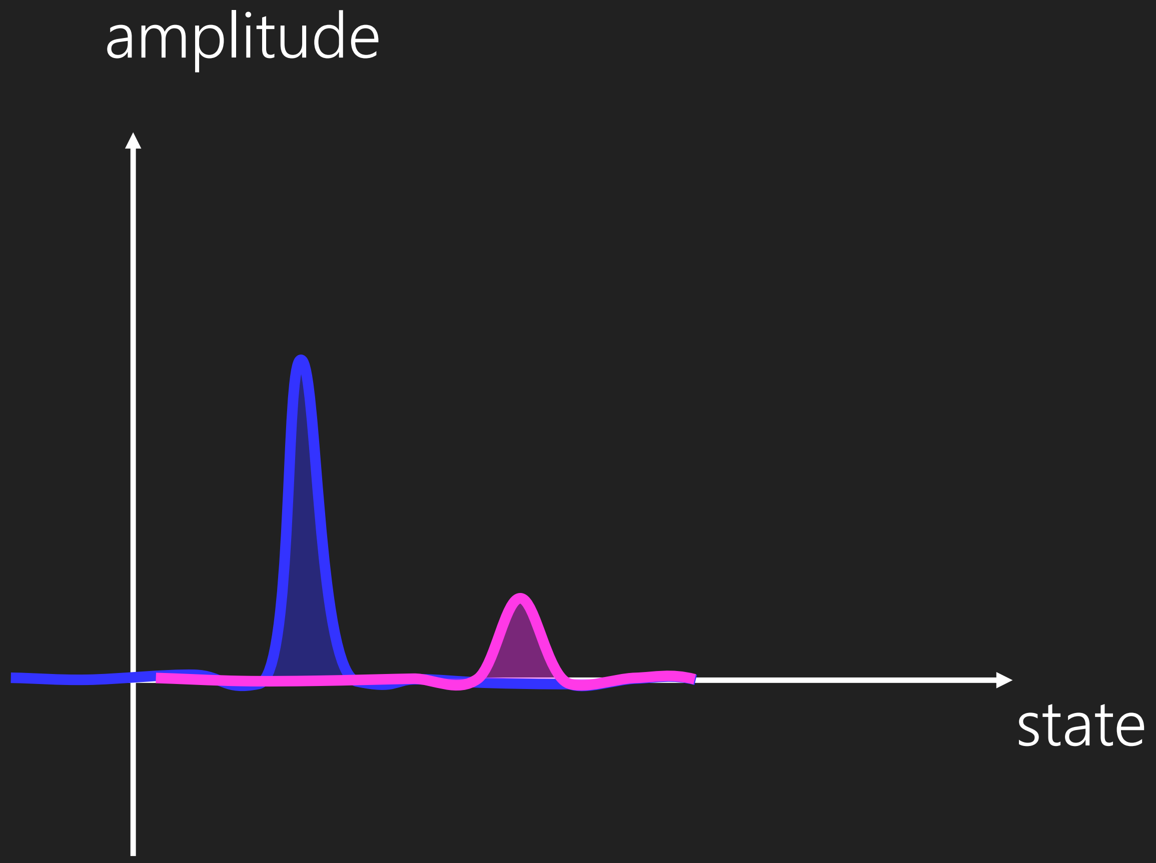
Wikipedia:
https://en.wikipedia.org/wiki/Double-slit_experiment

Animation by
G. Mikaberidze

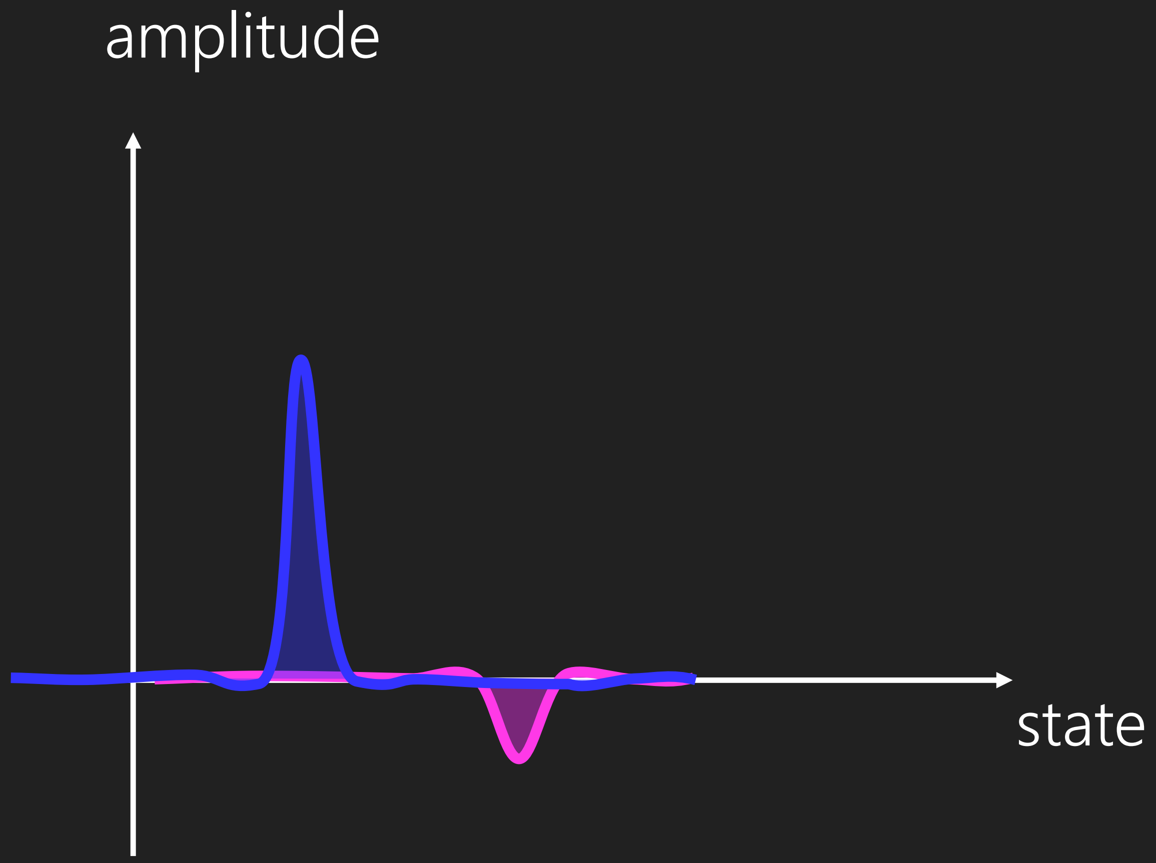
Quantum States & Interference



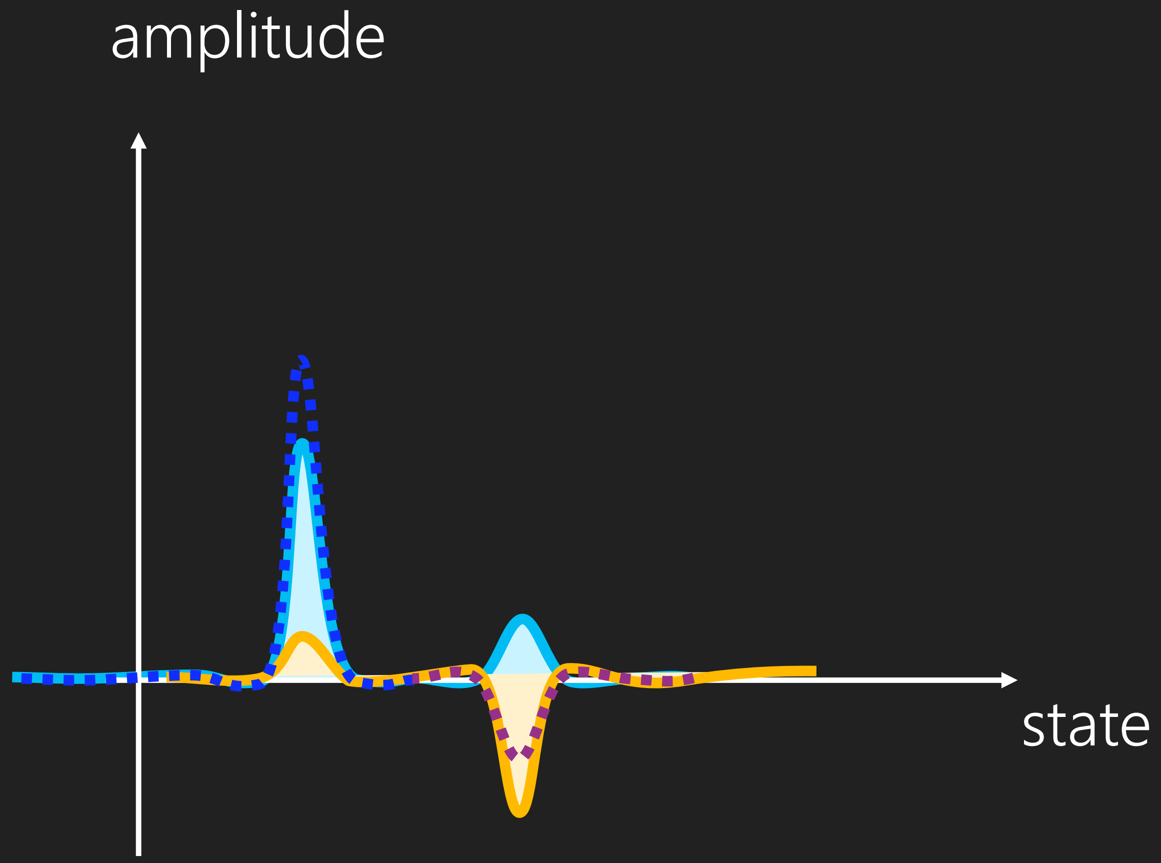
Quantum States & Interference



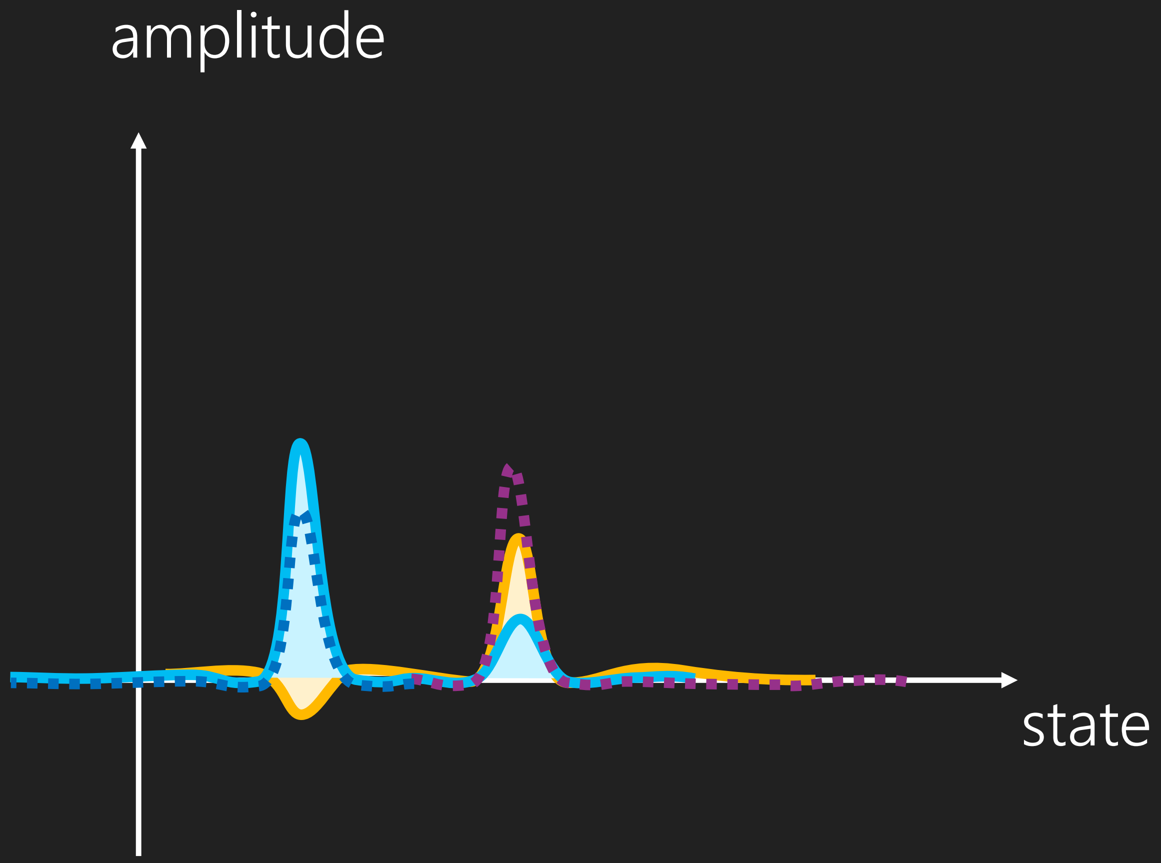
Quantum States & Interference



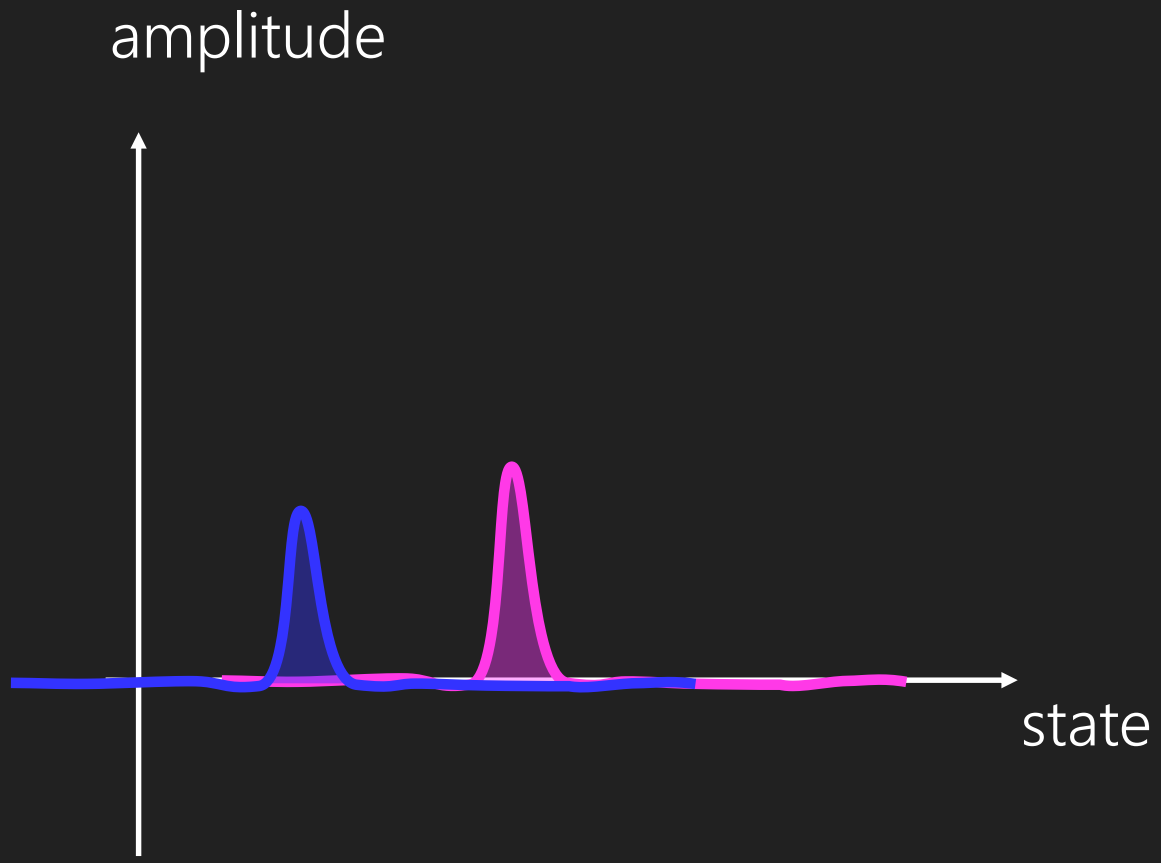
Quantum States & Interference



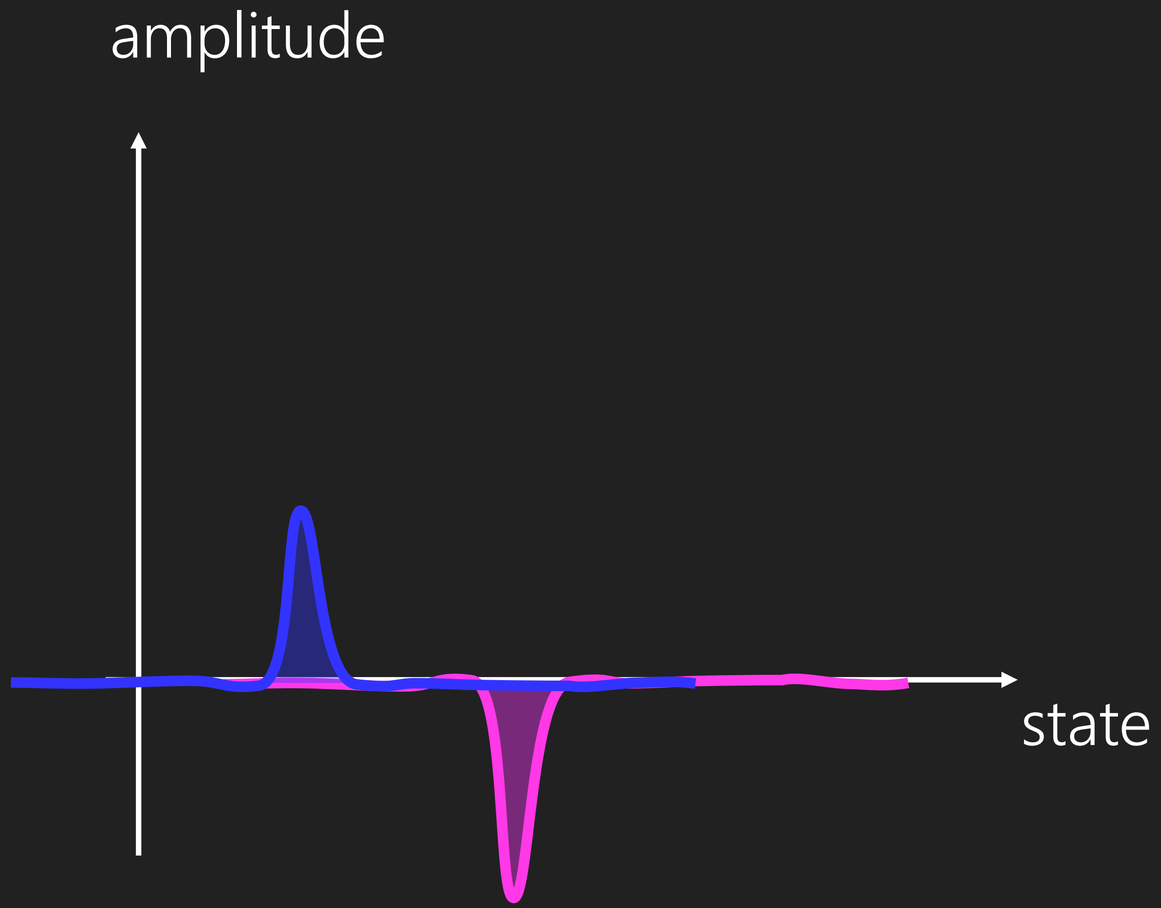
Quantum States & Interference



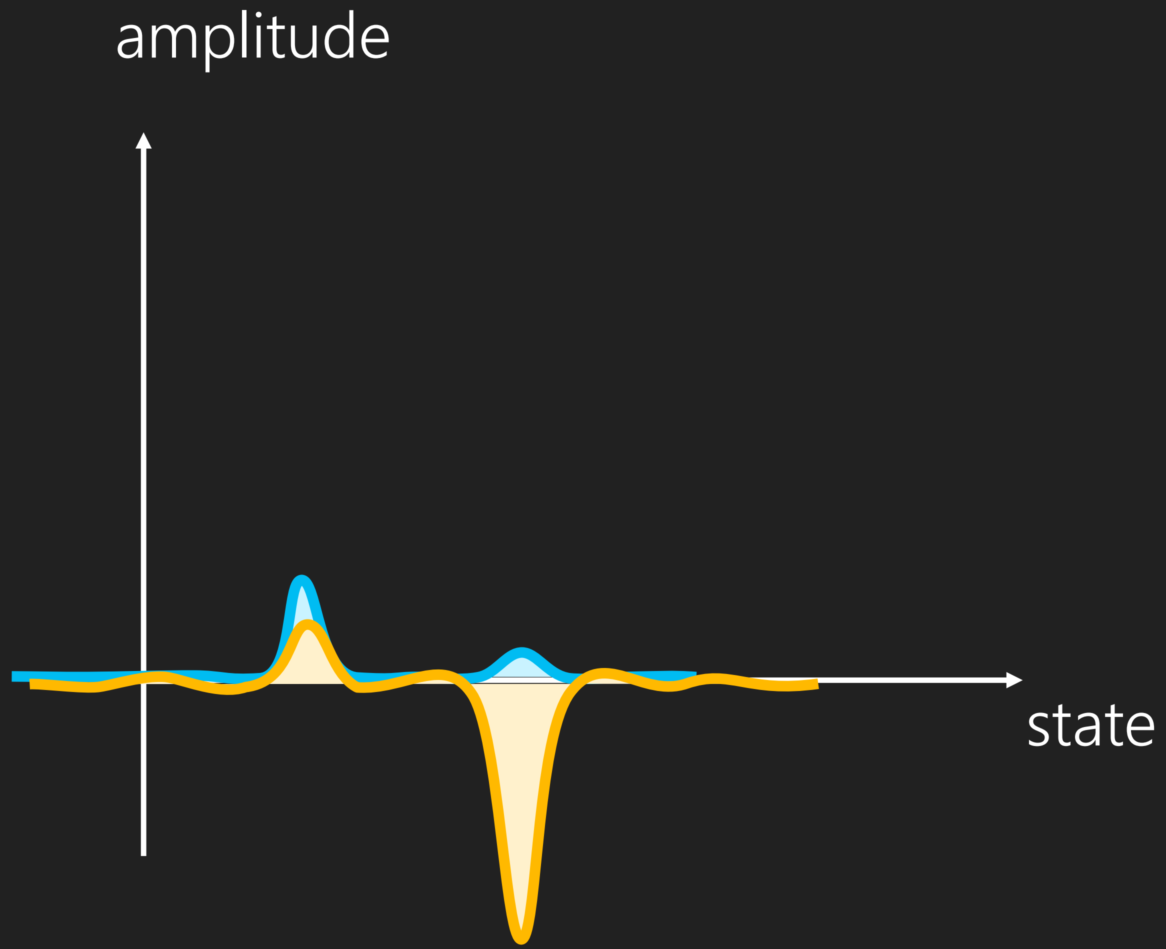
Quantum States & Interference



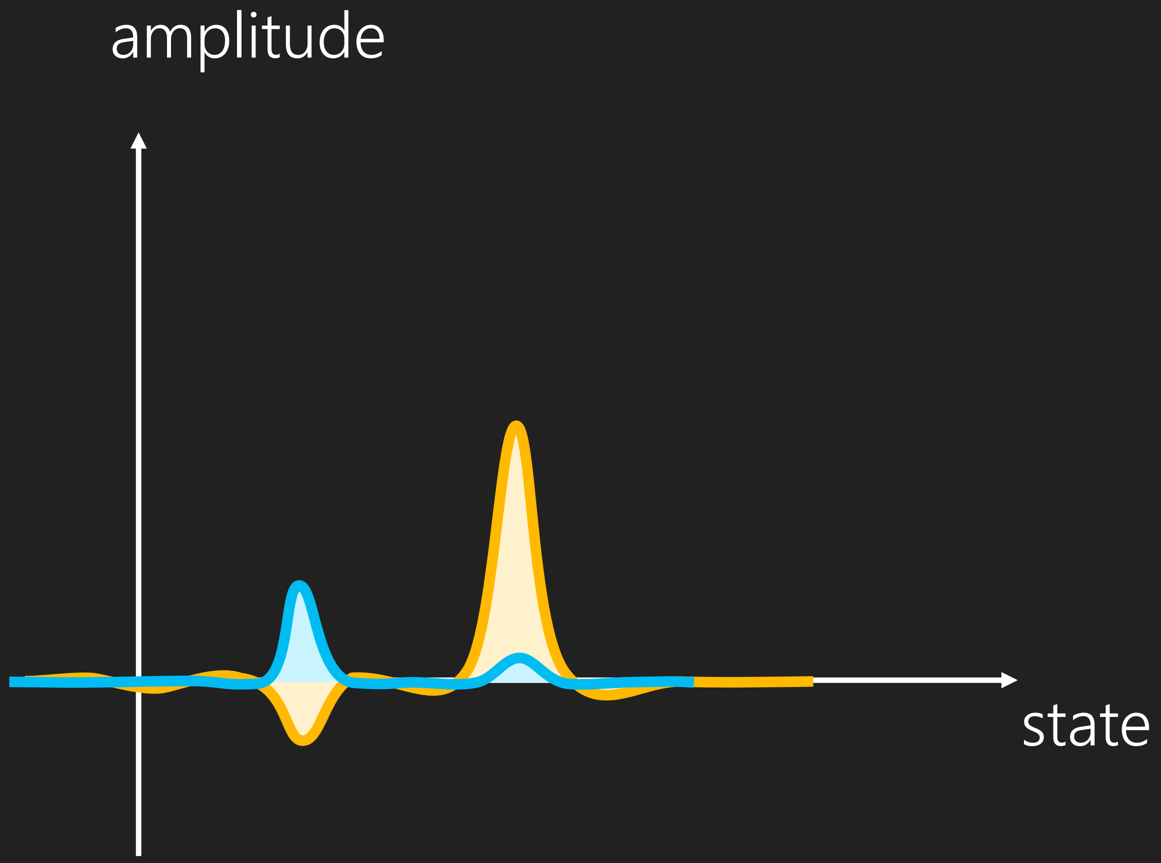
Quantum States & Interference



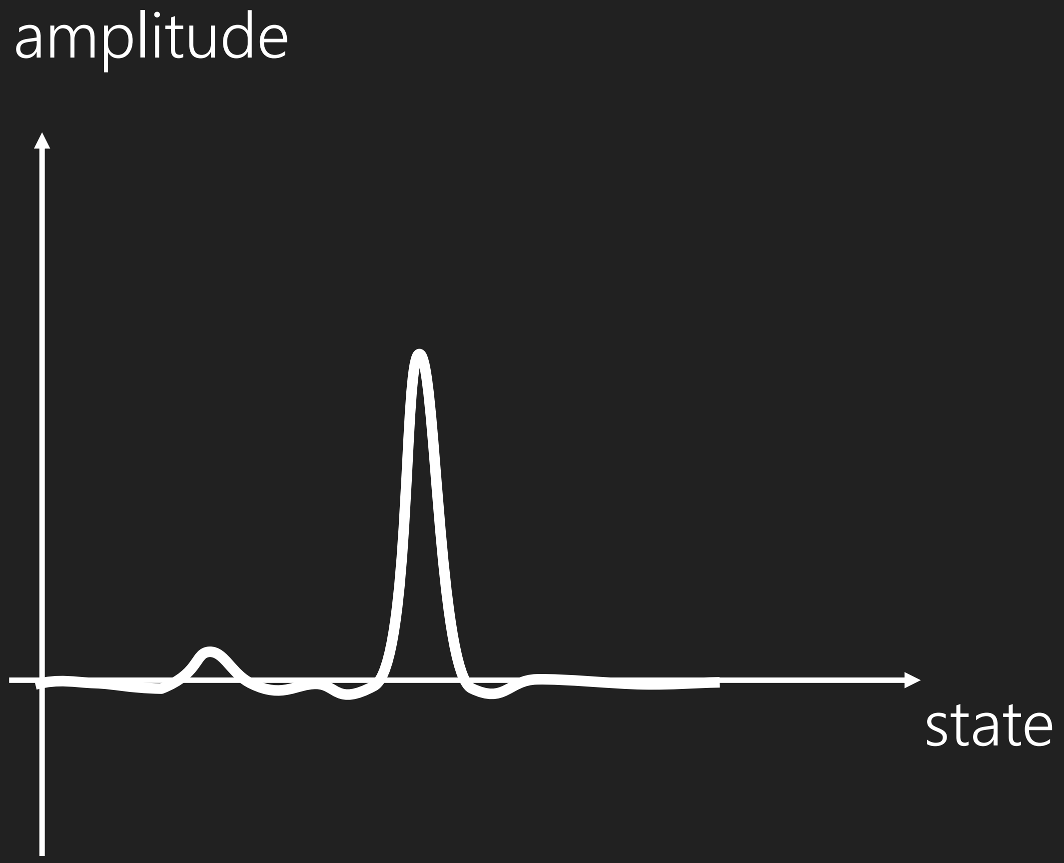
Quantum States & Interference



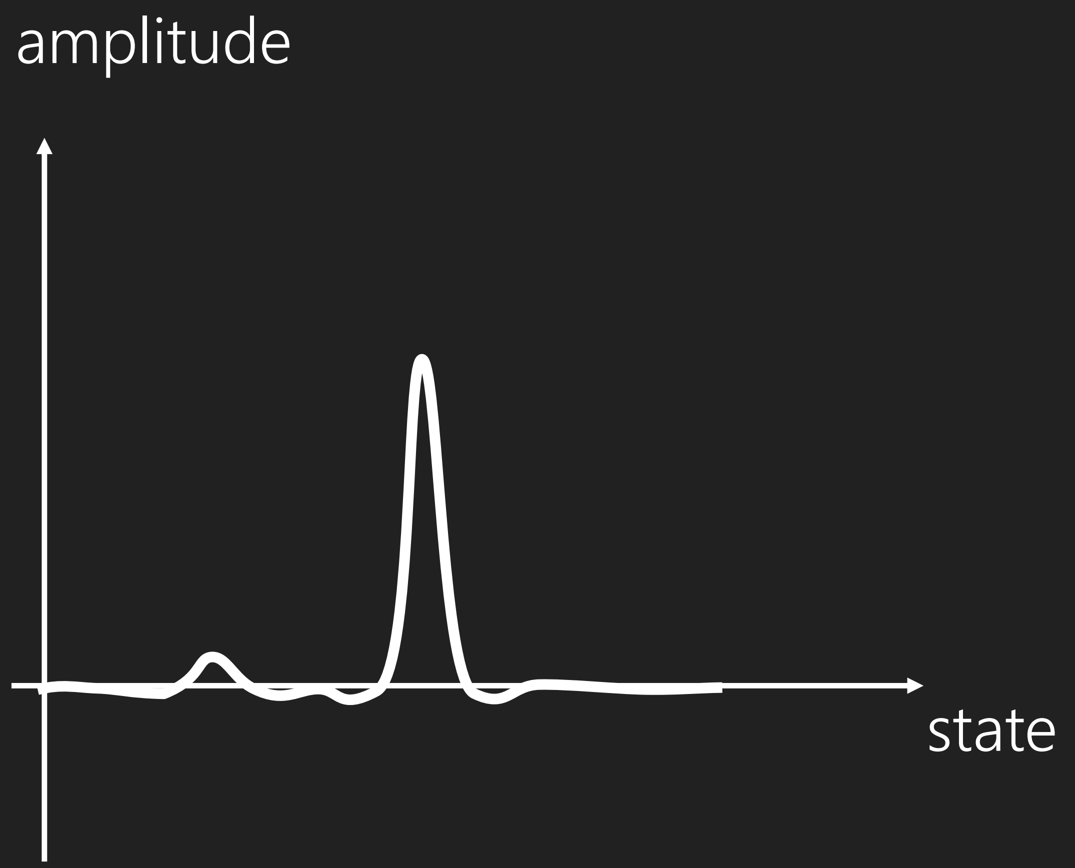
Quantum States & Interference

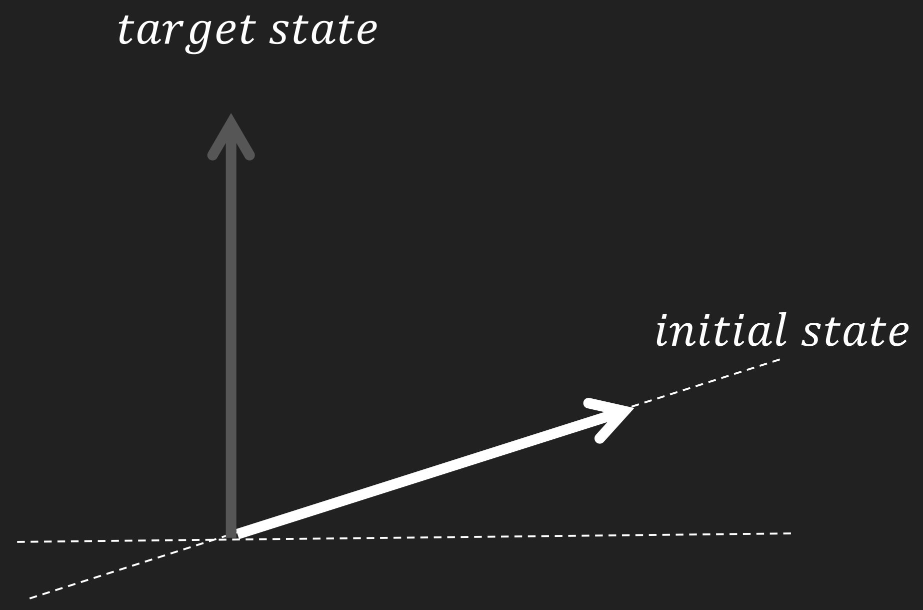


Quantum States & Interference

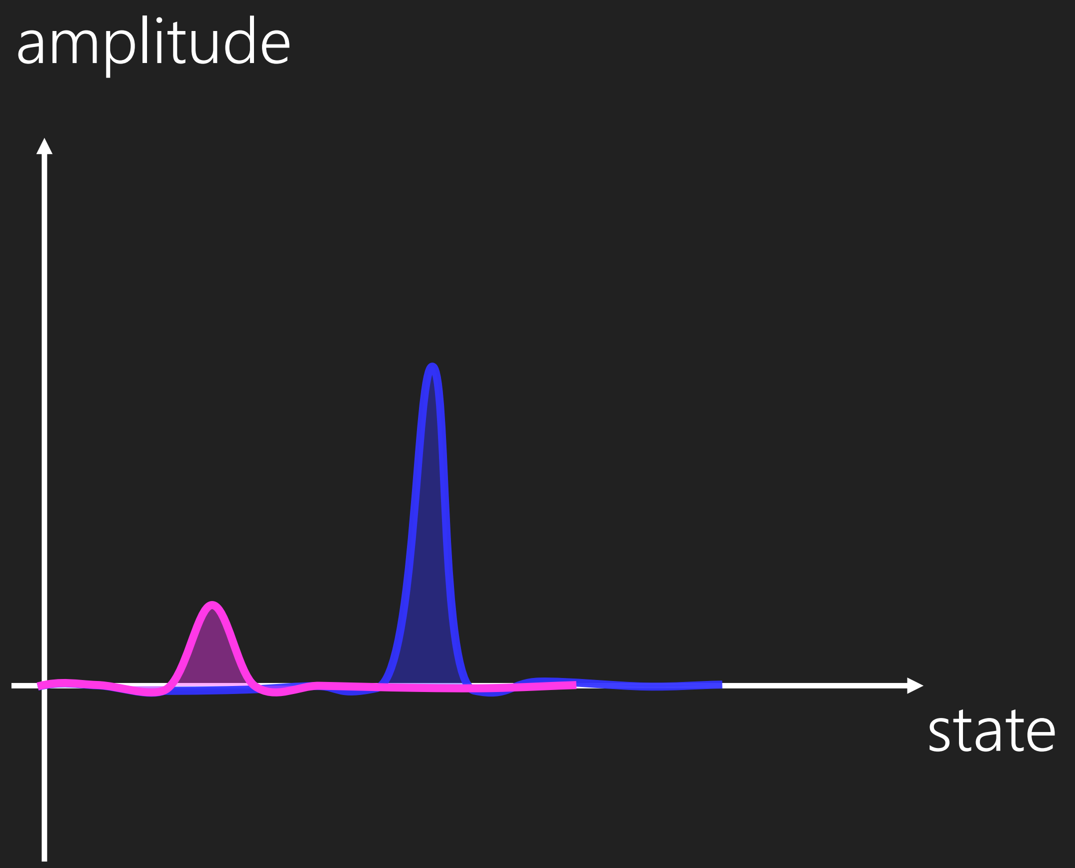


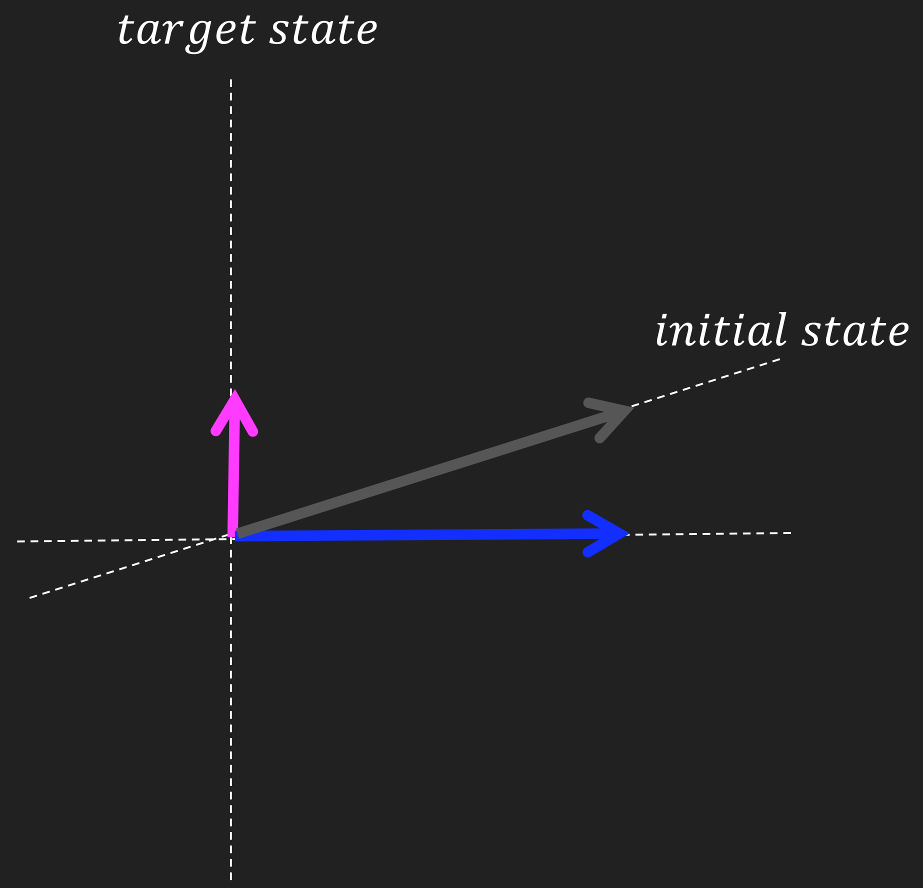
Quantum States & Interference



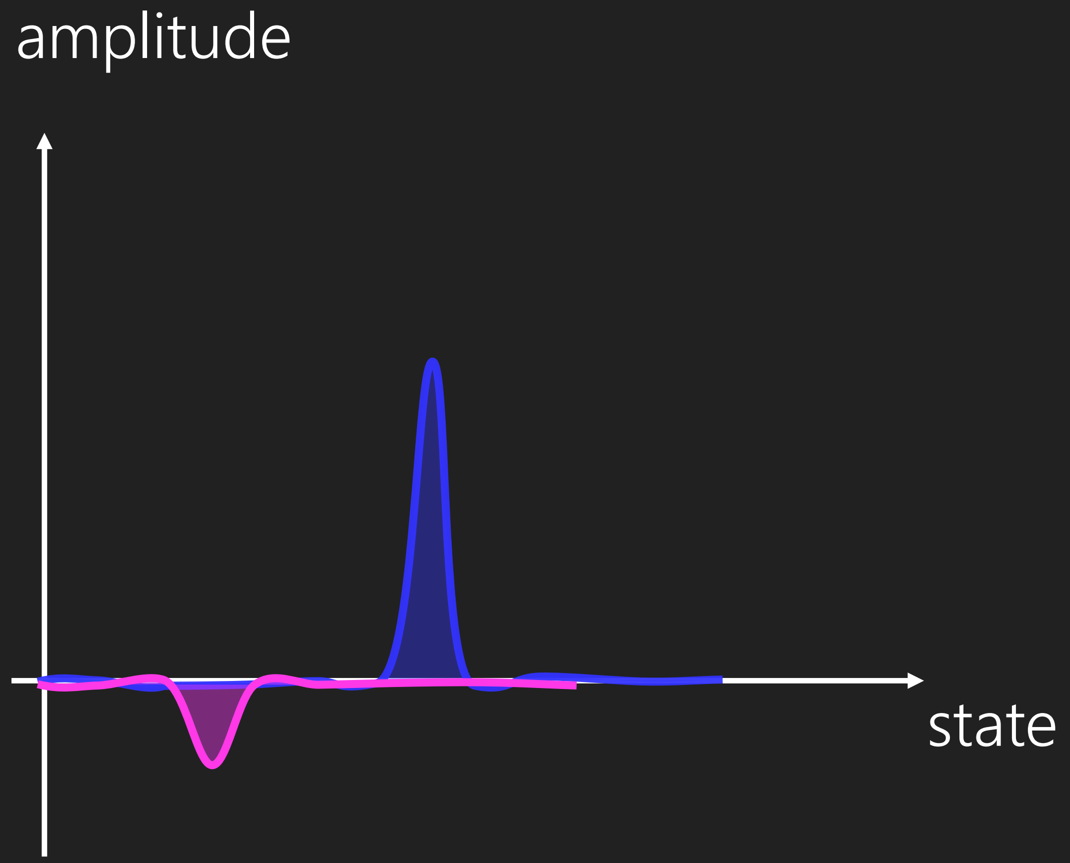


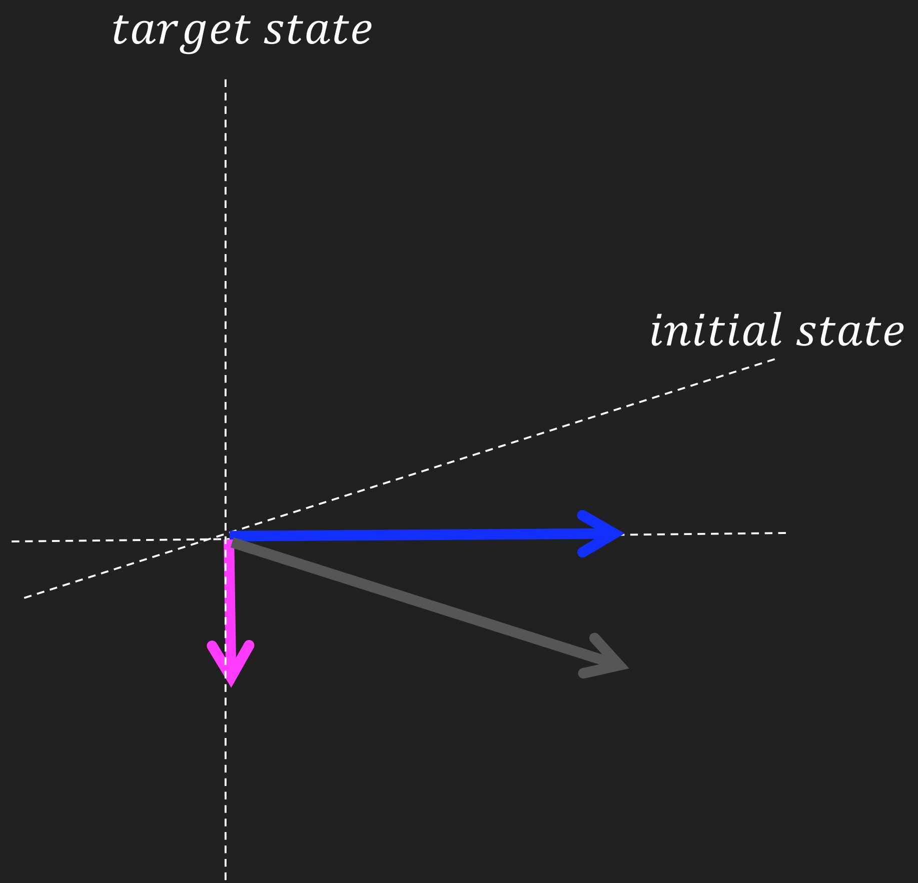
Quantum States & Interference



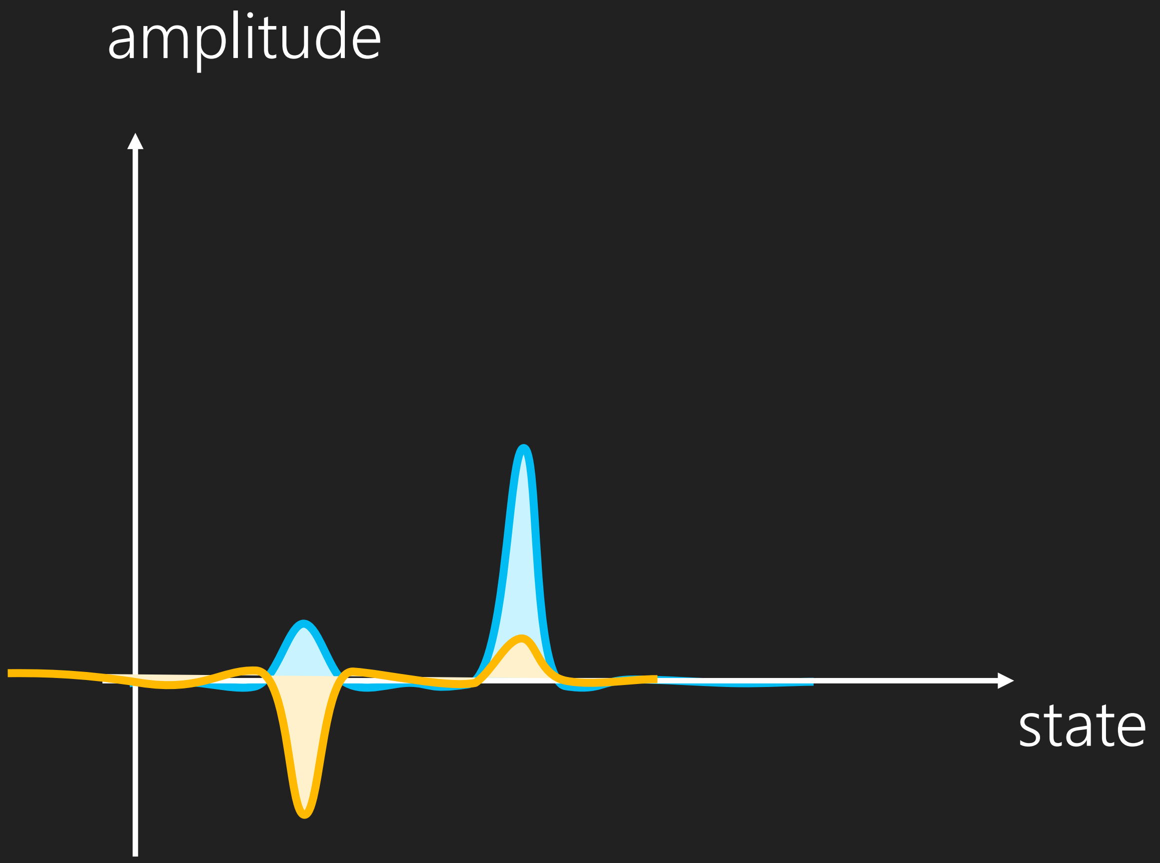


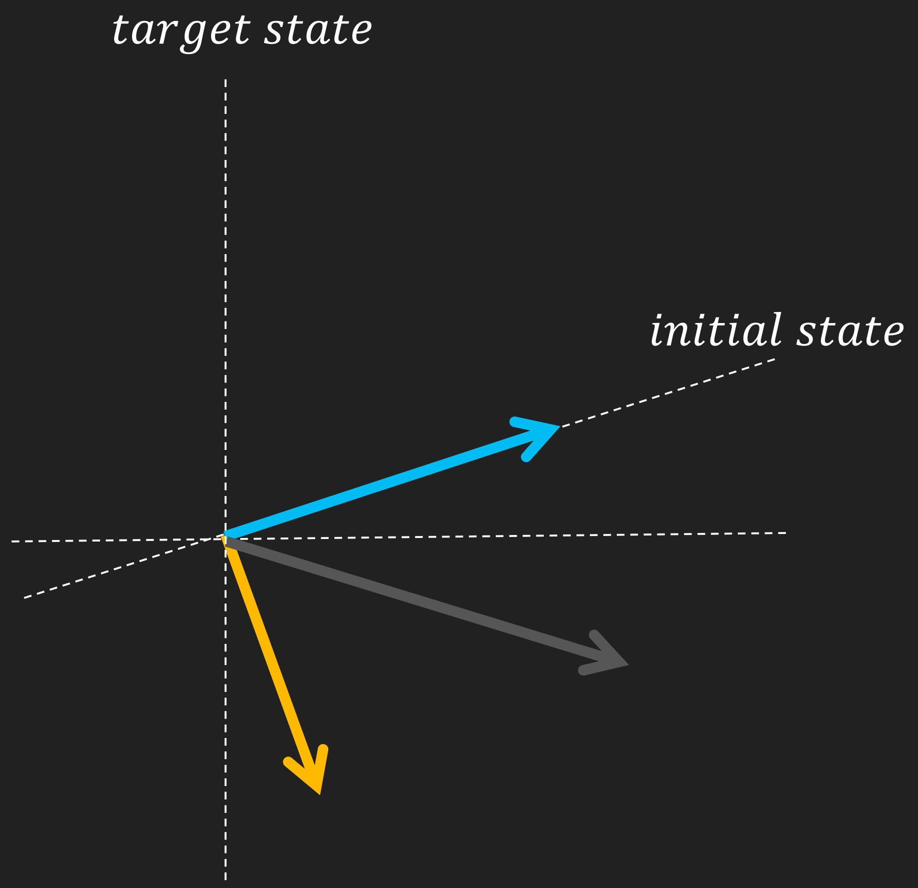
Quantum States & Interference



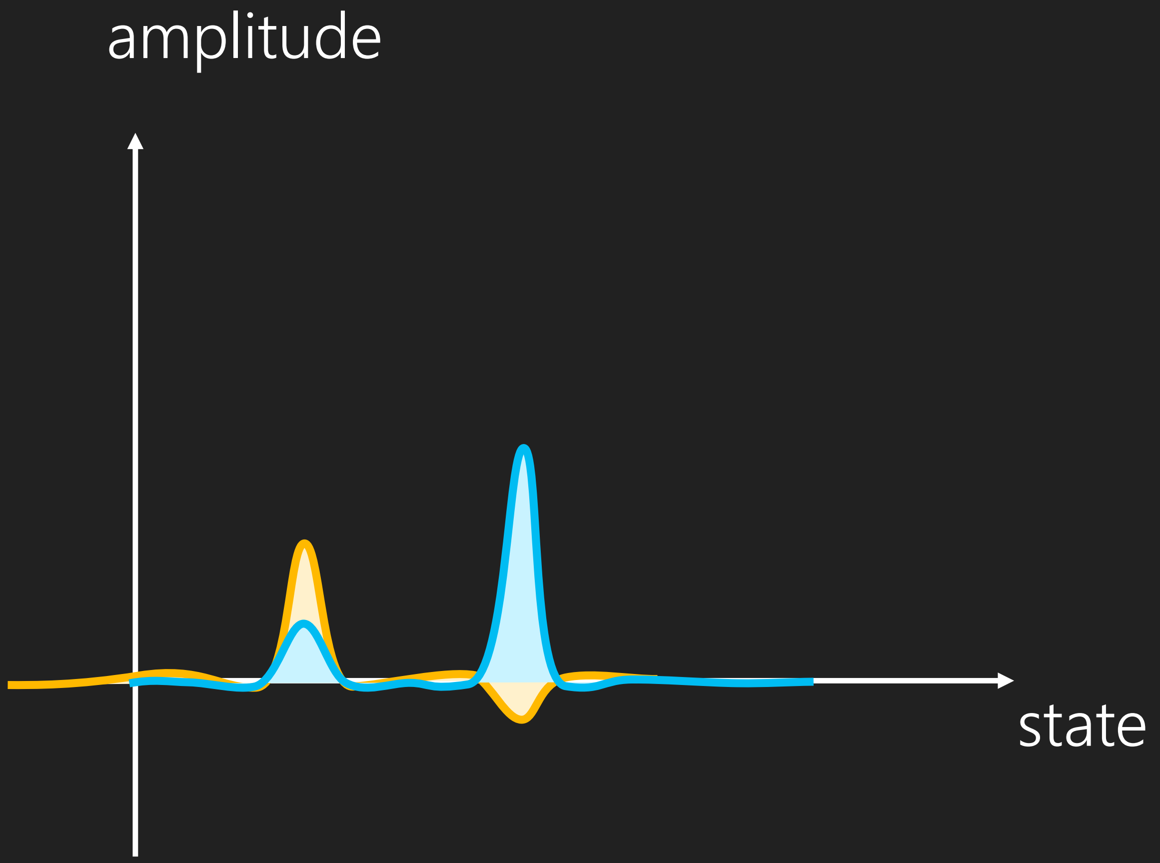


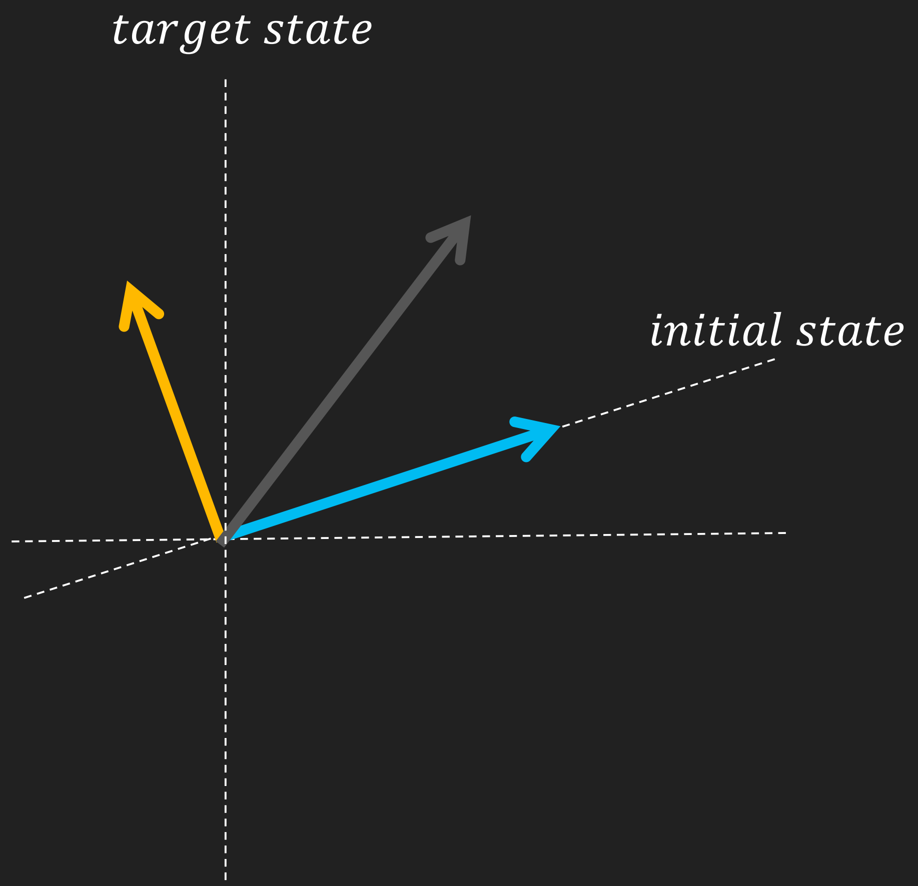
Quantum States & Interference



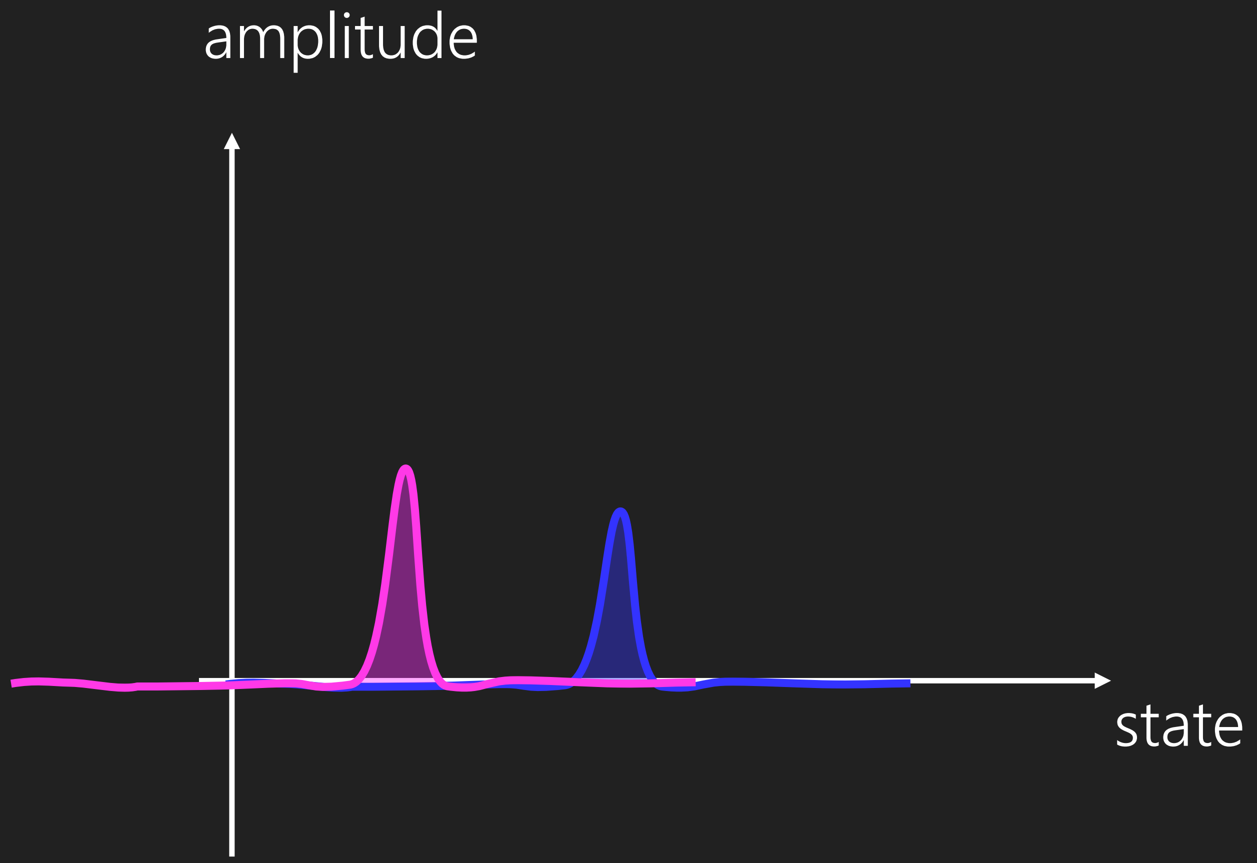


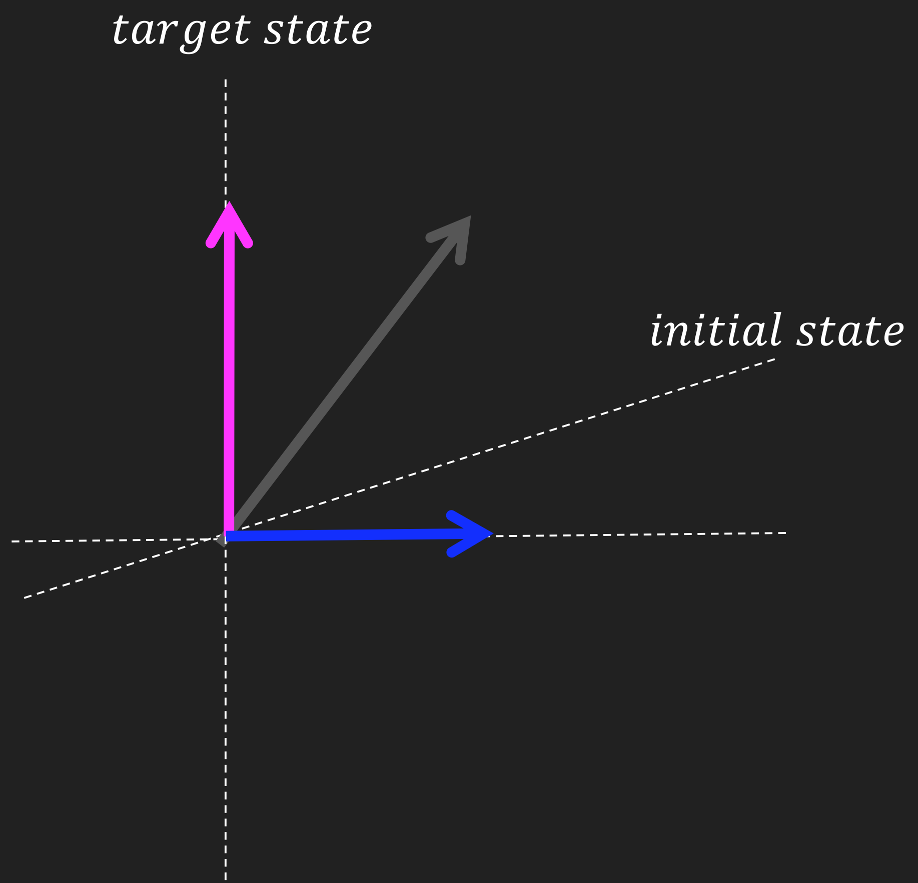
Quantum States & Interference



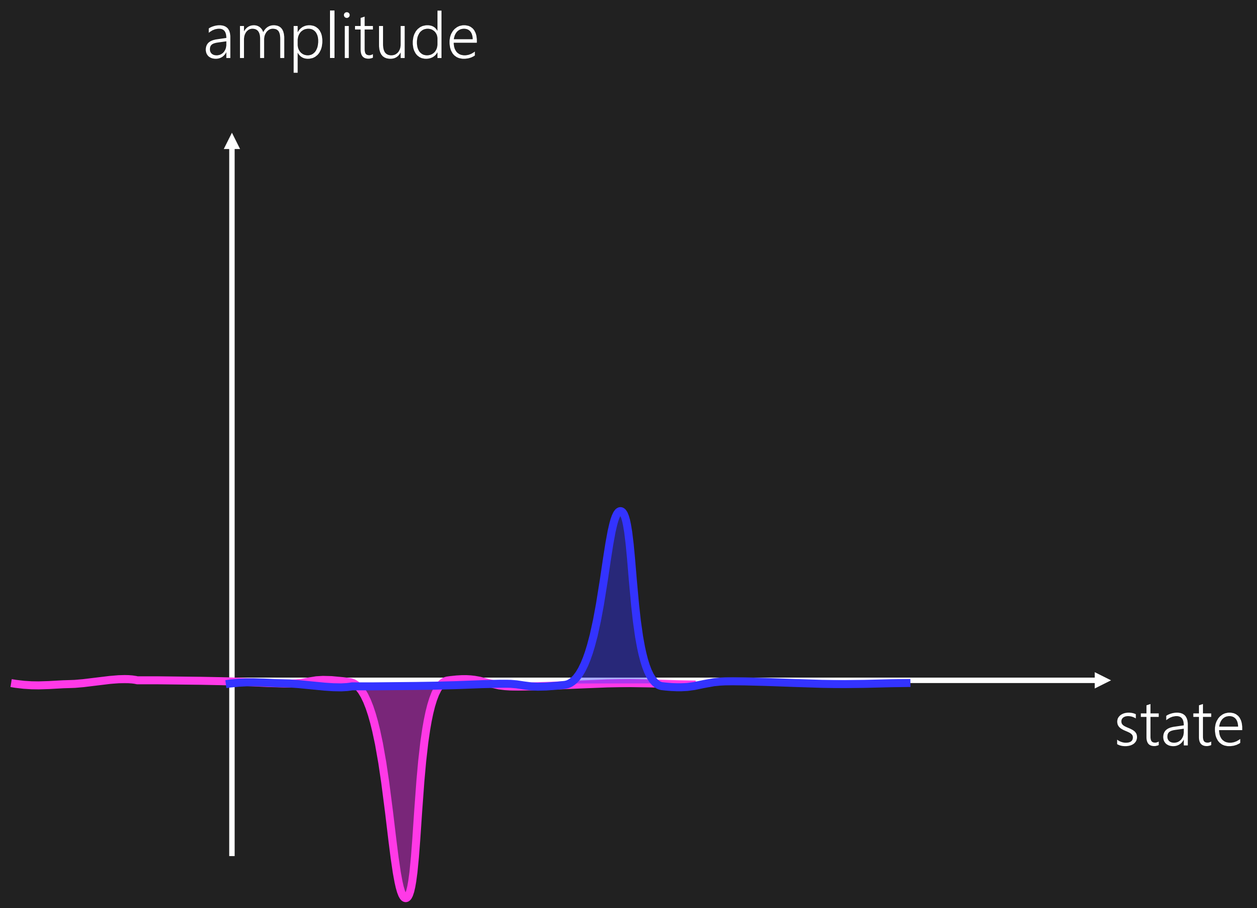


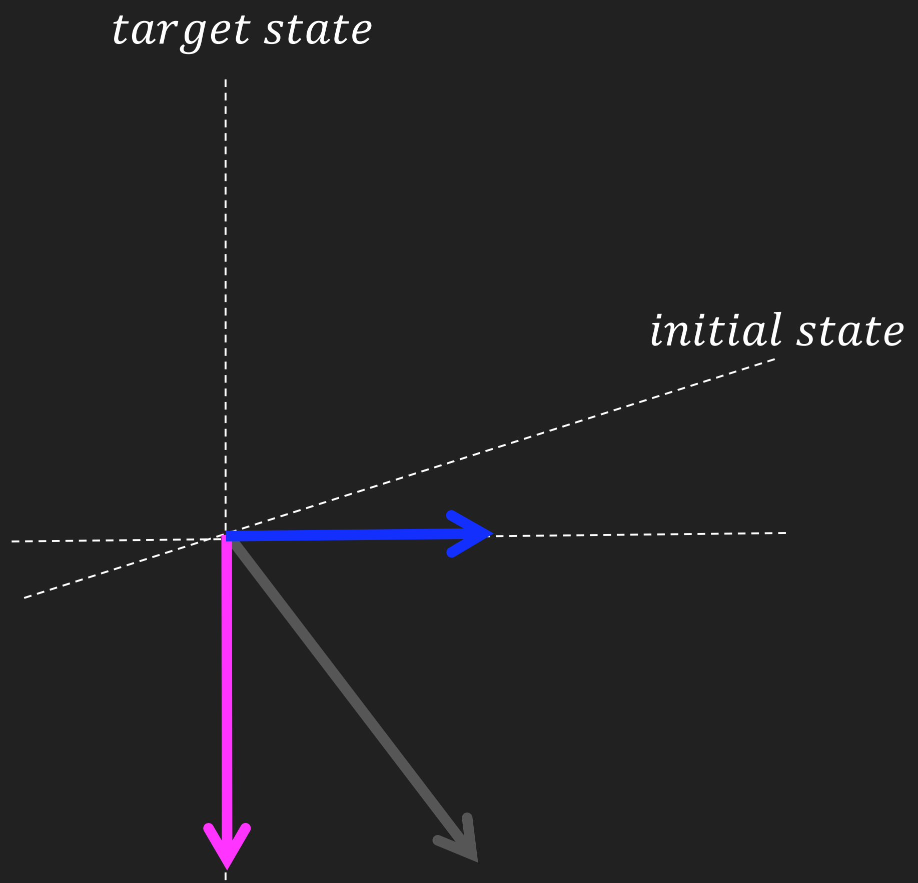
Quantum States & Interference



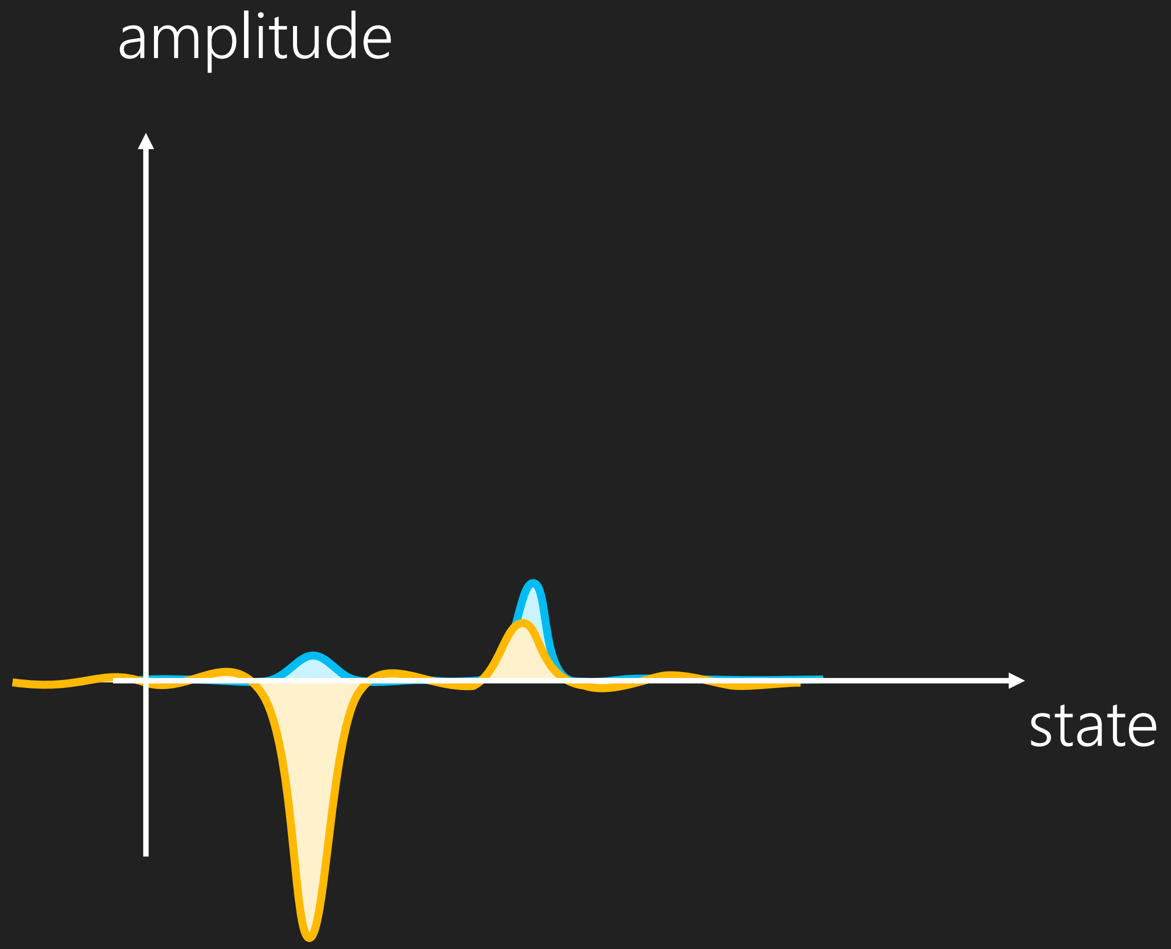


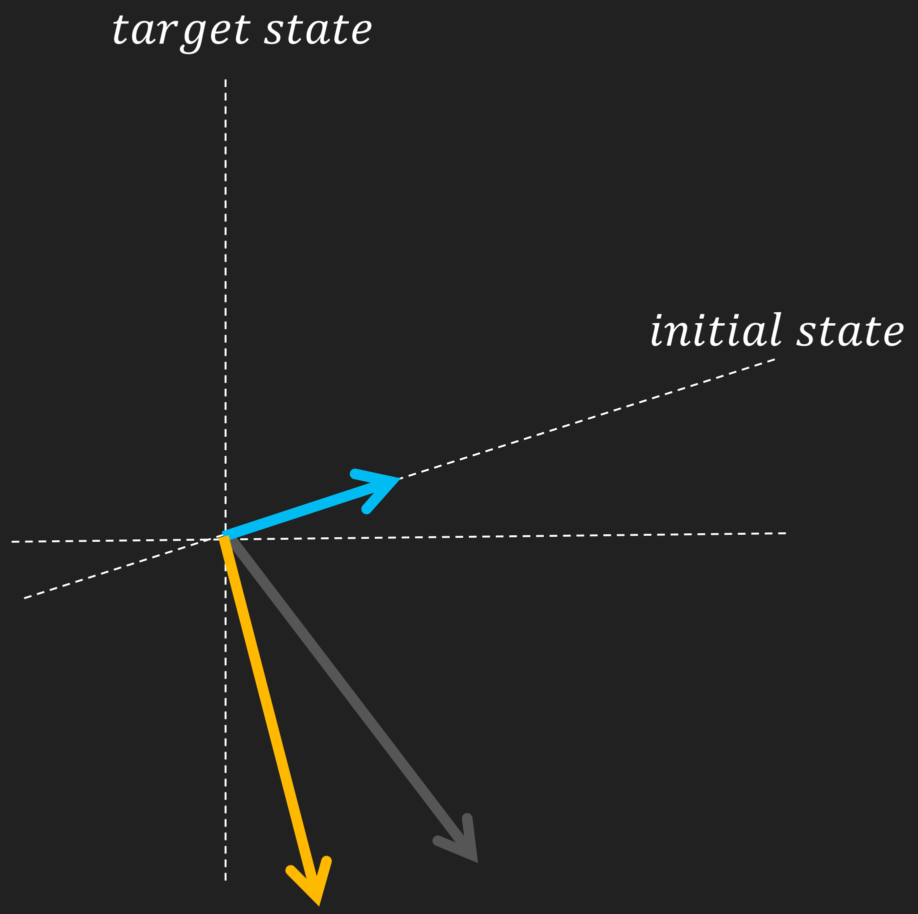
Quantum States & Interference



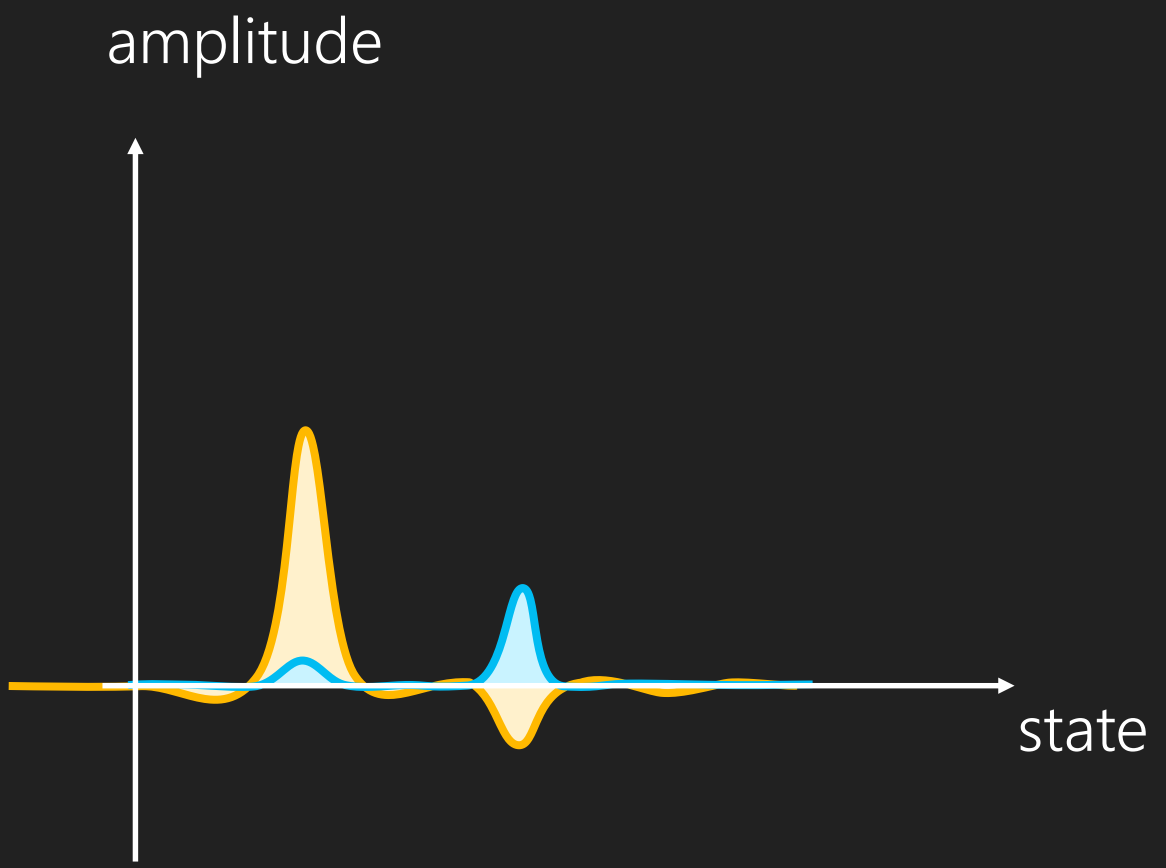


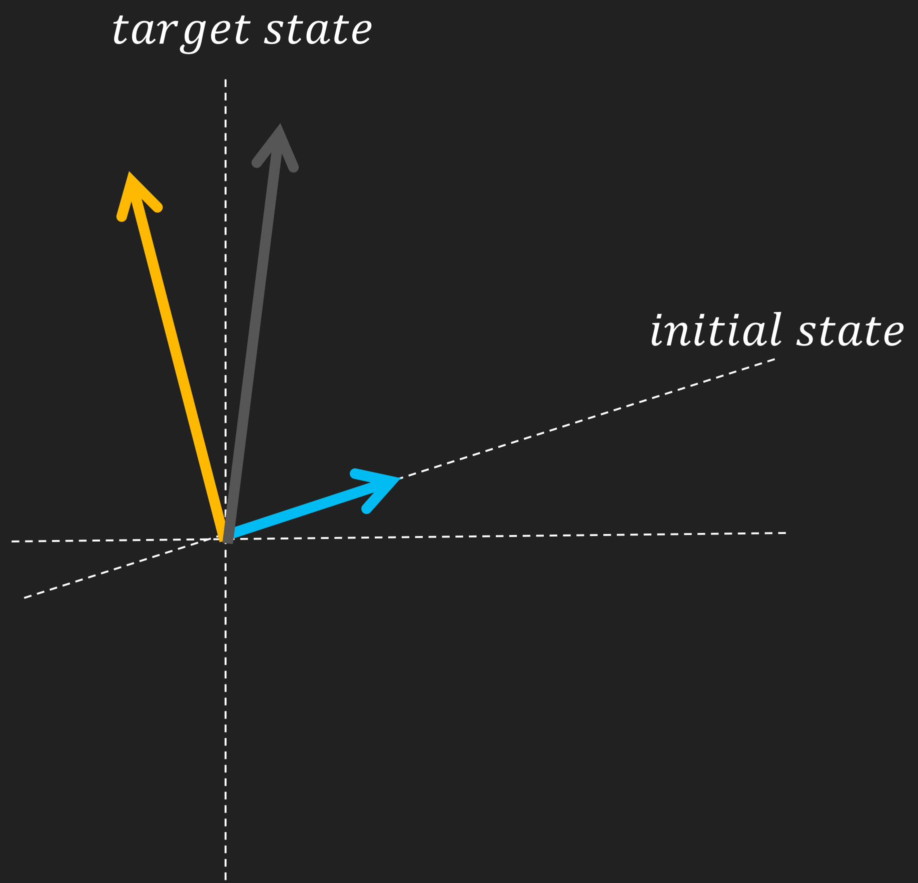
Quantum States & Interference



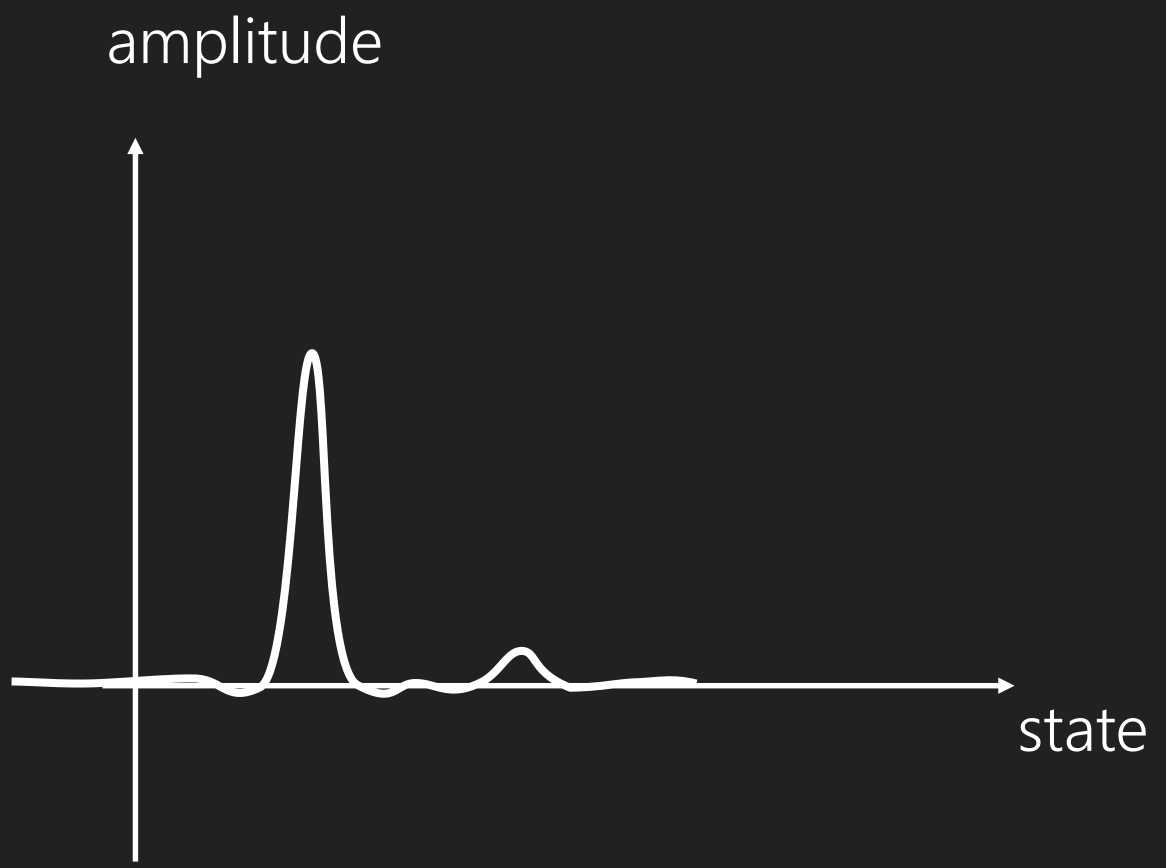


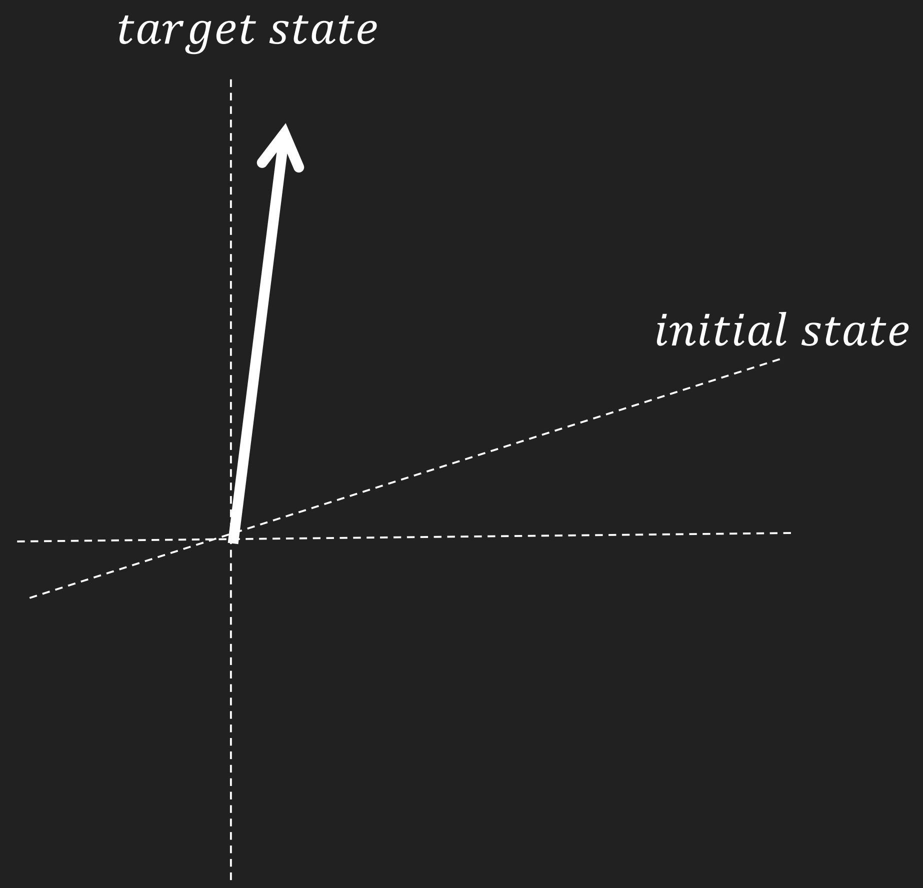
Quantum States & Interference





Quantum States & Interference

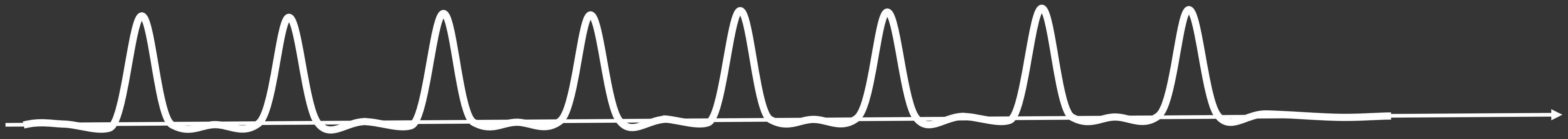




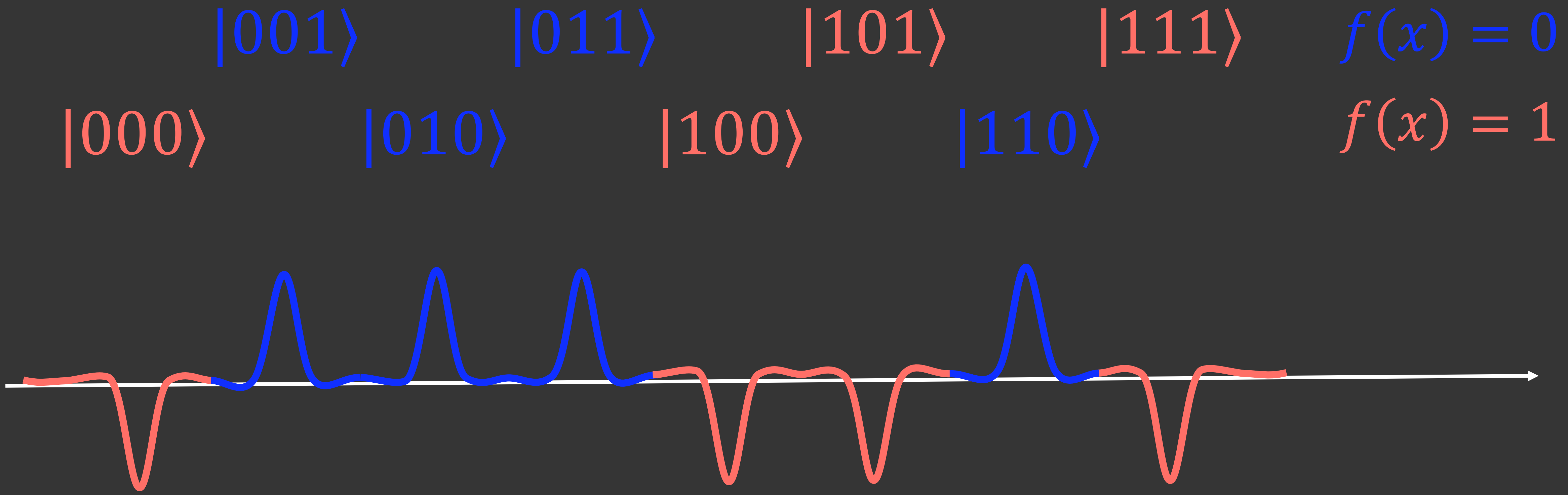
Quantum “Parallelism”

$|001\rangle$ $|011\rangle$ $|101\rangle$ $|111\rangle$

$|000\rangle$ $|010\rangle$ $|100\rangle$ $|110\rangle$



Quantum "Parallelism"



Quantum “Parallelism”

Applications

- The amount of information to extract is far smaller than the information needed to compute the relevant properties
- Even multiplied by the effort needed to extract the relevant information, the quantum algorithm is still faster
- Input is hard to capture classically

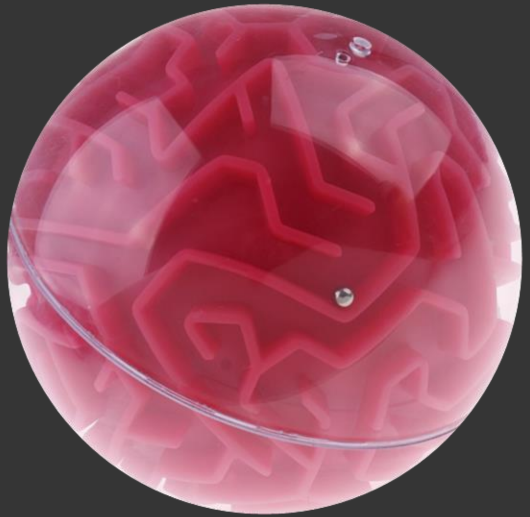
Computation Description

Describe what to do with the state

- State of the computation:
  quantum state (Schroedinger picture)
- Program description:
  discrete sequence of actions (digital)

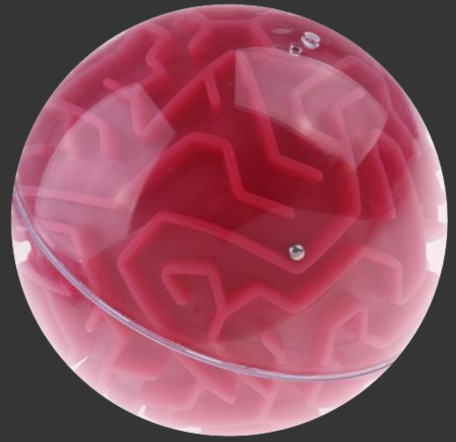
Describe the desired effect on the state

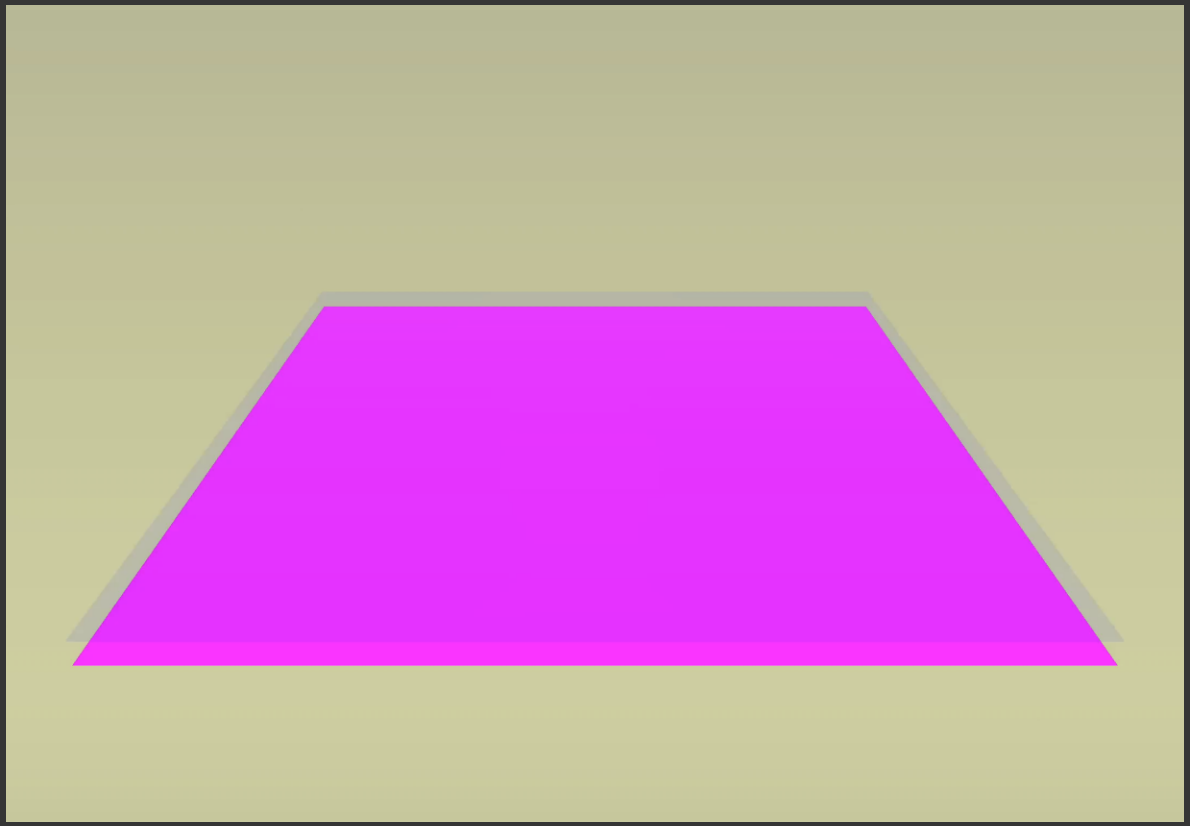
- State of the computation:
  currently applied actions (Heisenberg Picture)
- Program description:
  function describing how to change
  the continuously applied actions (analog)



Analog Computation

Program description:
function describing how to change
the continuously applied actions (analog)





- Qubit lattice with fixed geometry
- Fixed set of unable couplings and fields
- program describes how field and coupling strengths vary over time
- can in principle be universal

Analog Computation

Computing by time evolution (annealing)

→ heuristic solver for Quadratic Unconstrained Binary Optimization Problems

Applications:

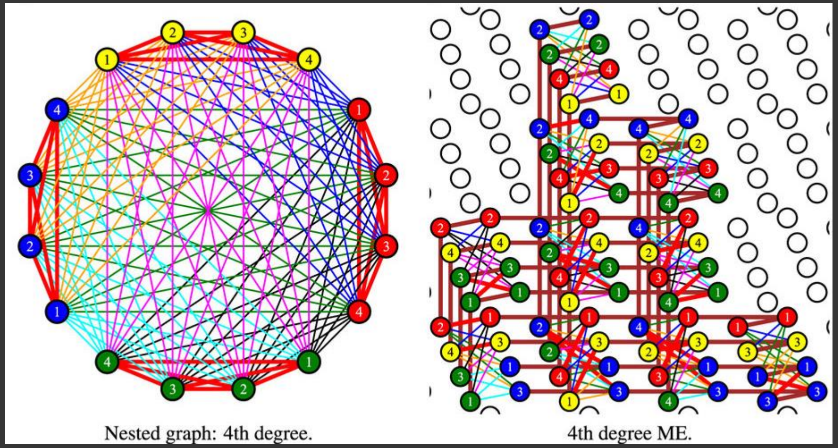
- NP-complete combinatorial optimization problems
- sampling and machine learning
- chemistry, biology & materials simulations

Challenges on Analog Devices:

- overhead for non-native problems
- mostly restricted to quadratic optimization
- embedding and problem engineering
- limited possibility for error correction
- required coupling precision

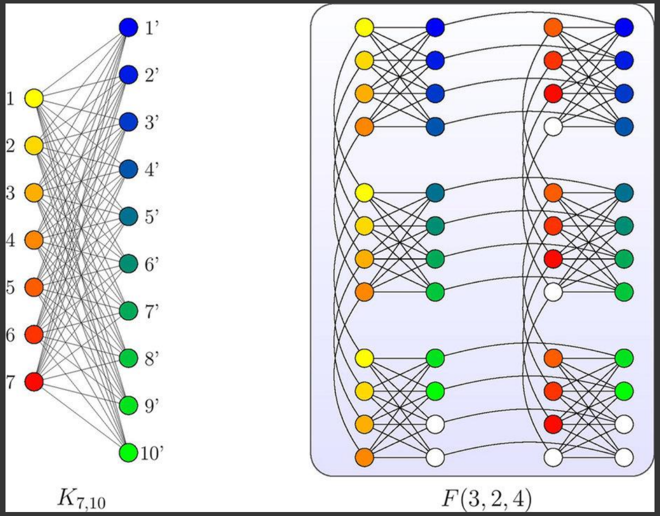
nested quantum annealing correction

Vinci, Albash, Lidar, 2016



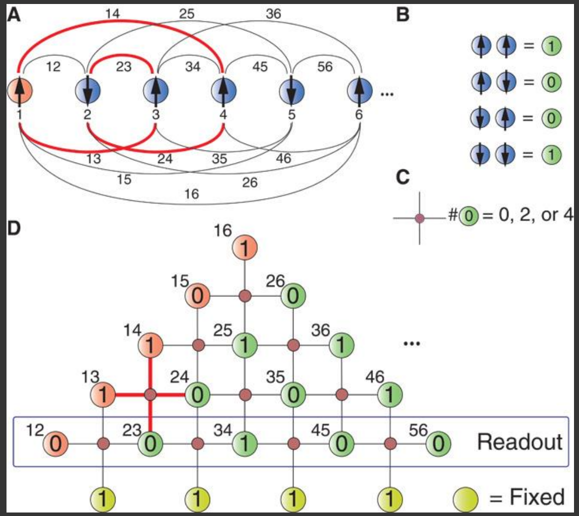
Cao, Jiang, Perouli, Kais, 2016

solving set cover with pairs problem by QA



embedding for all-to-all two-body interactions

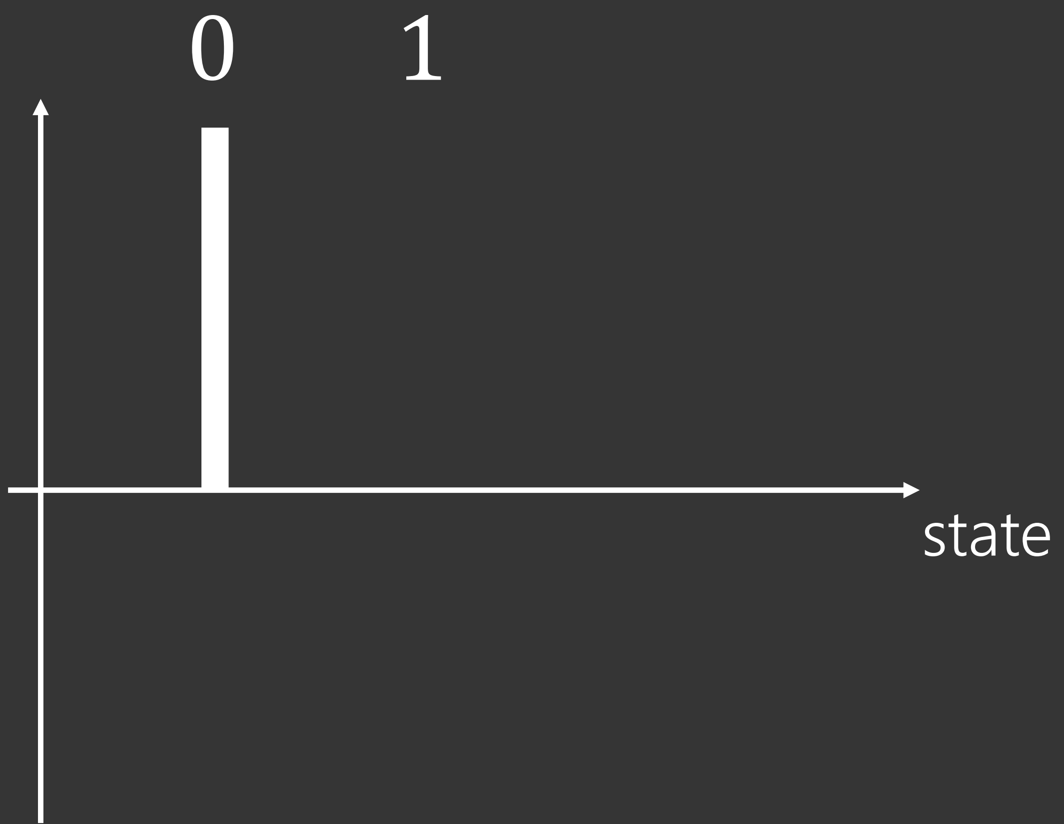
Lechner, Hauke, Zoller, 2015



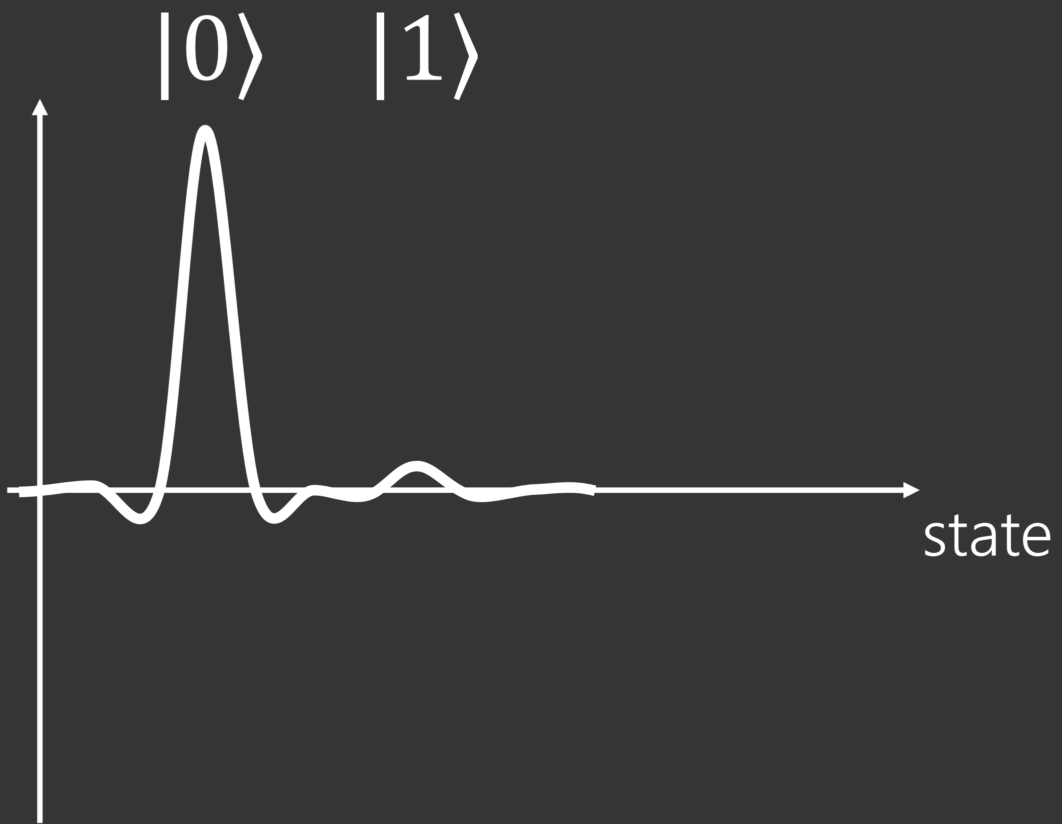
Digital Computation

Program description:
discrete sequence of actions (digital)

Classical:



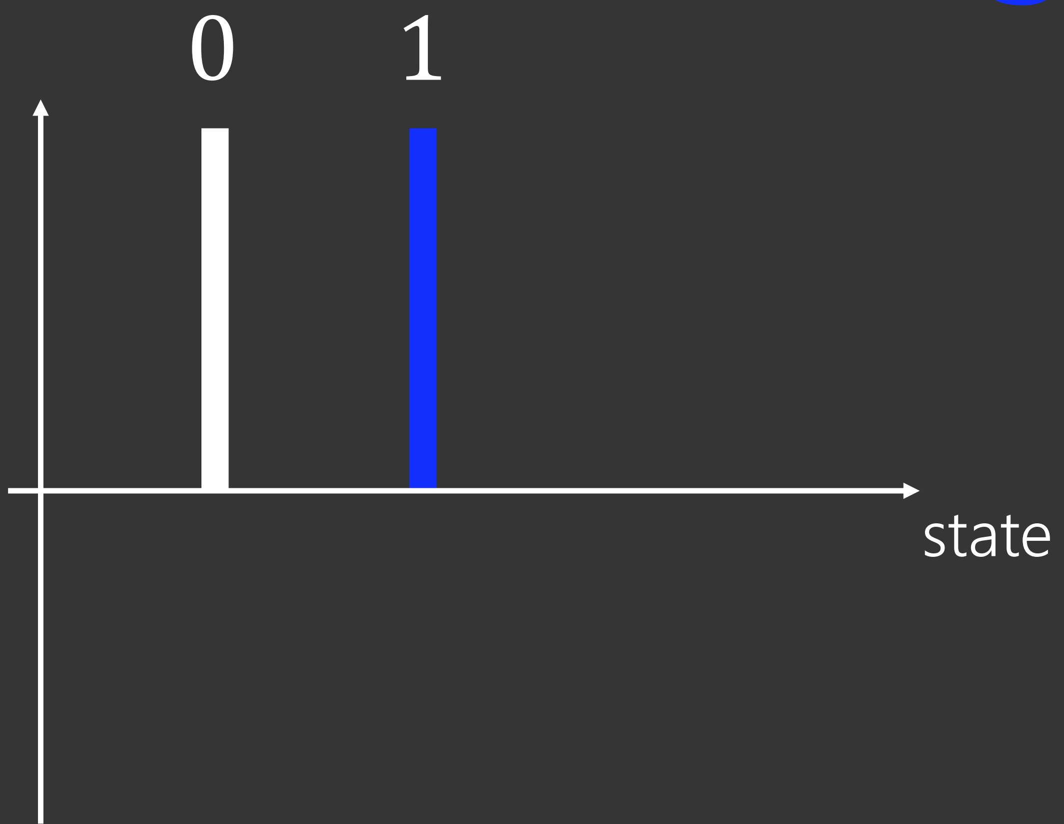
Quantum:



Digital Computation

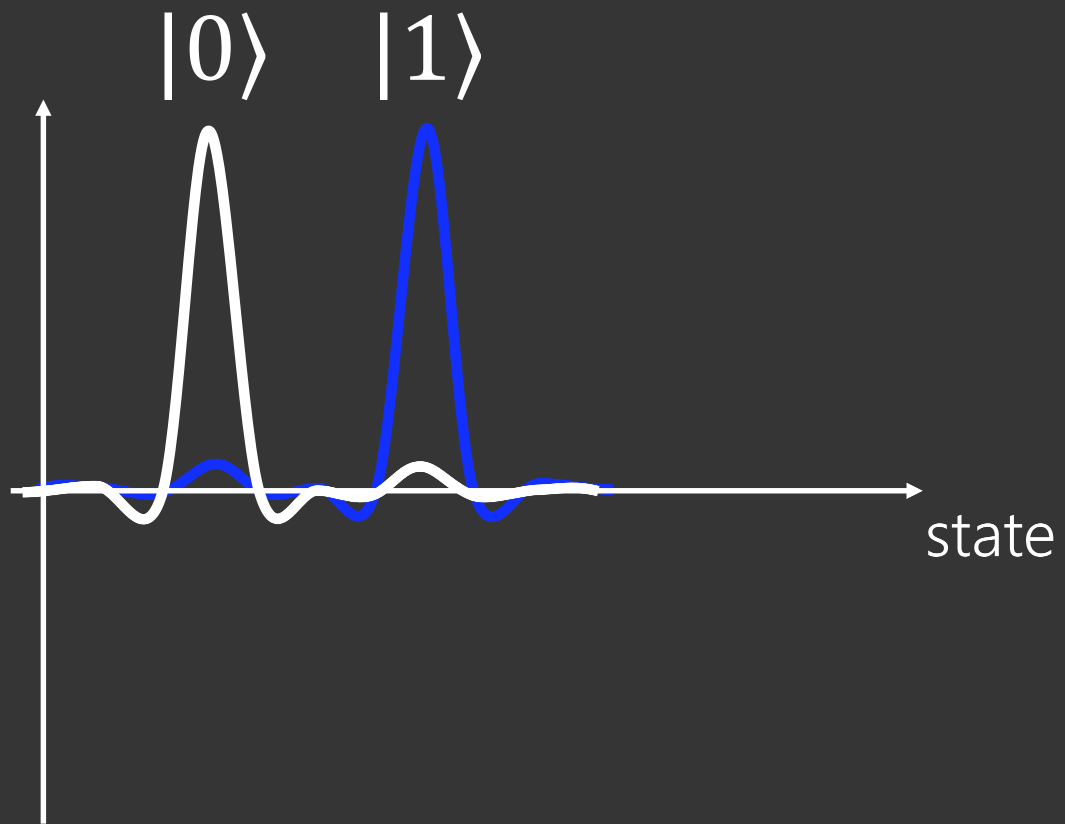
Program description:
discrete sequence of actions (digital)

Classical:



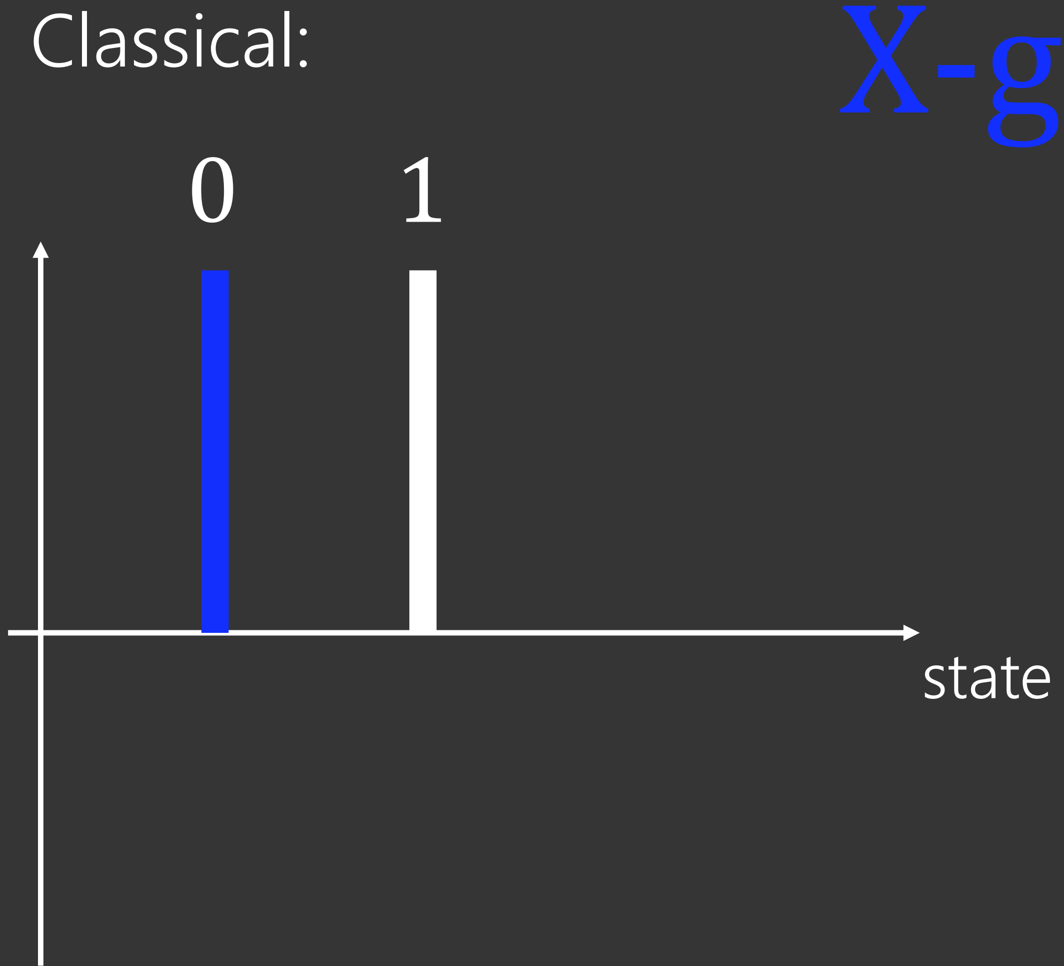
X-gate

Quantum:

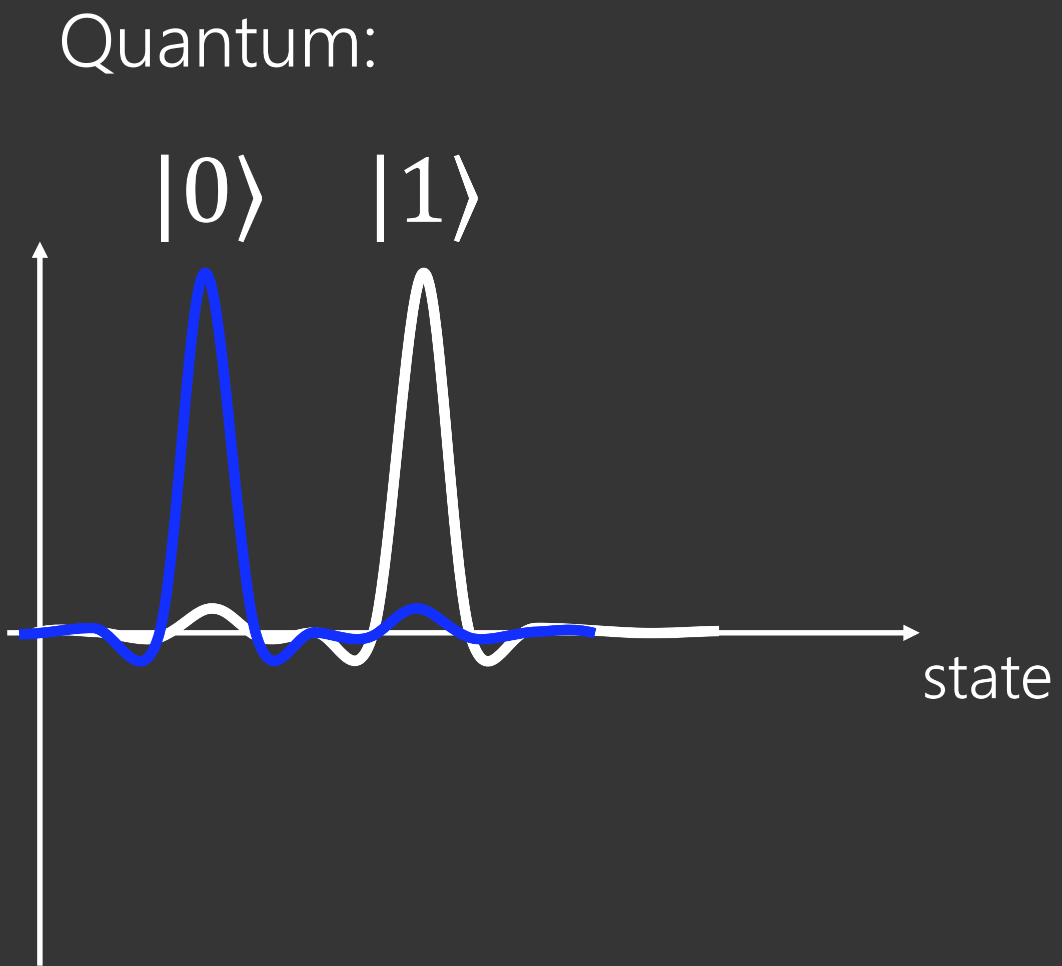


Digital Computation

Program description:
discrete sequence of actions (digital)



X-gate



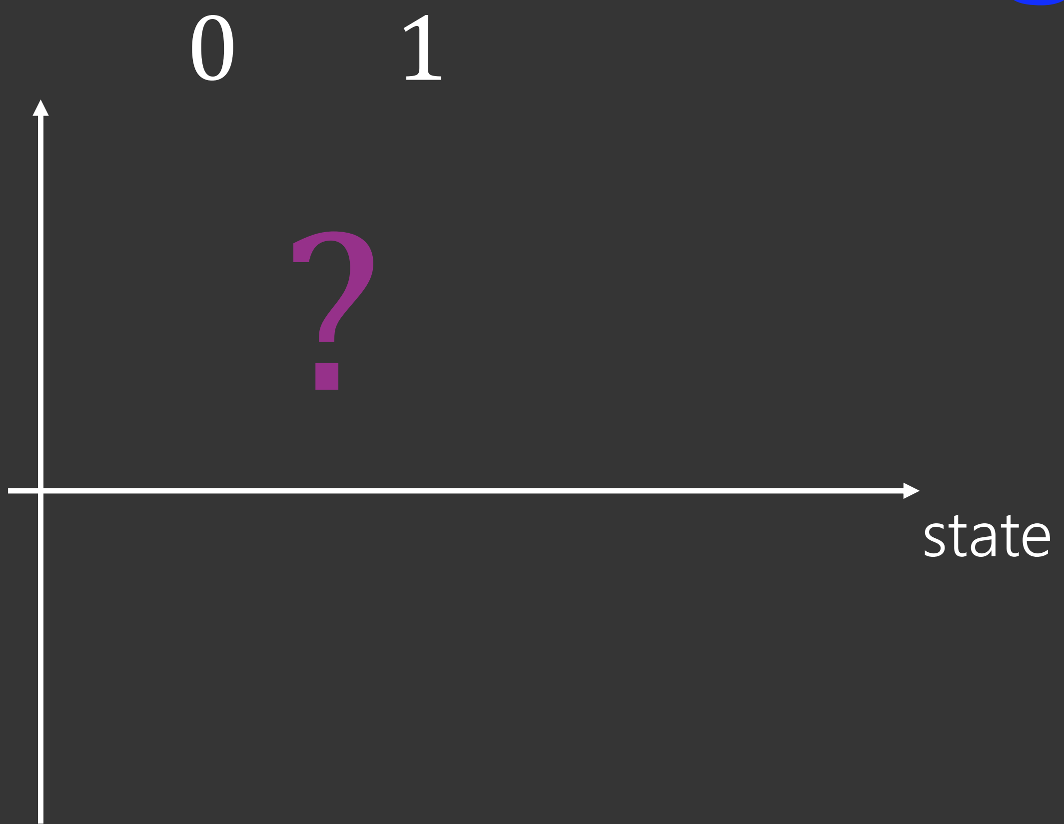
Digital Computation

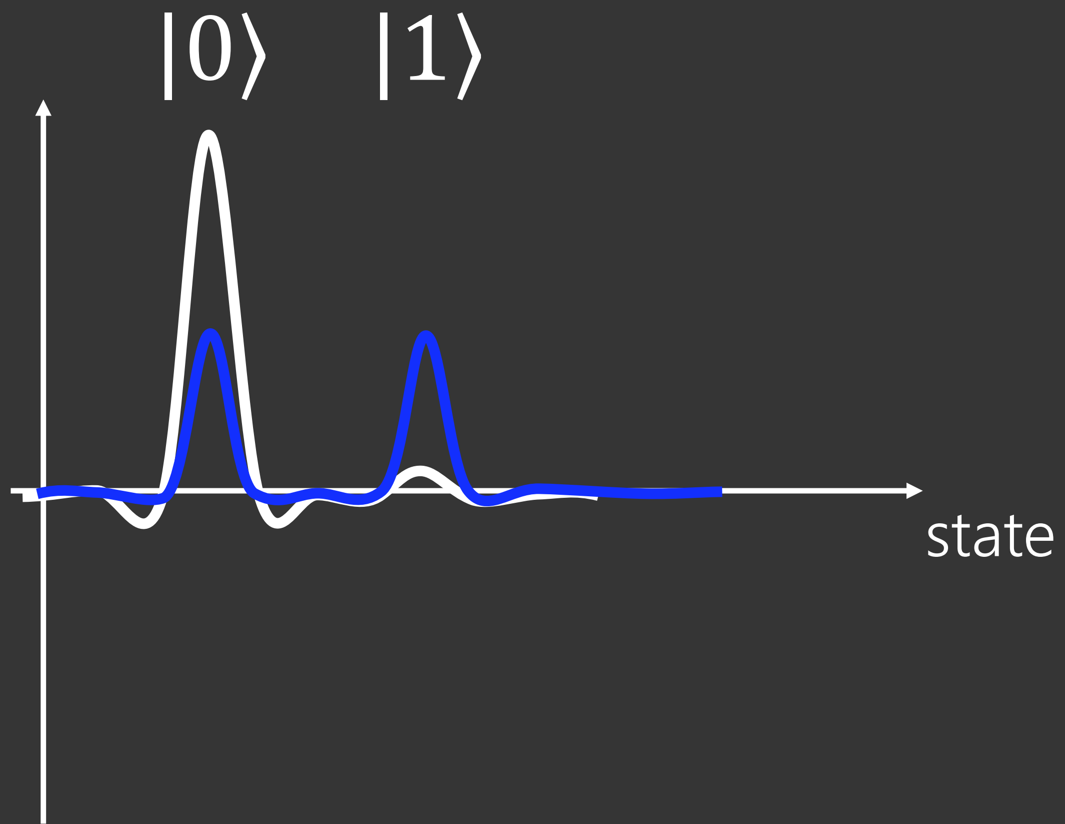
Program description:
discrete sequence of actions (digital)

Classical:

H-gate

Quantum:

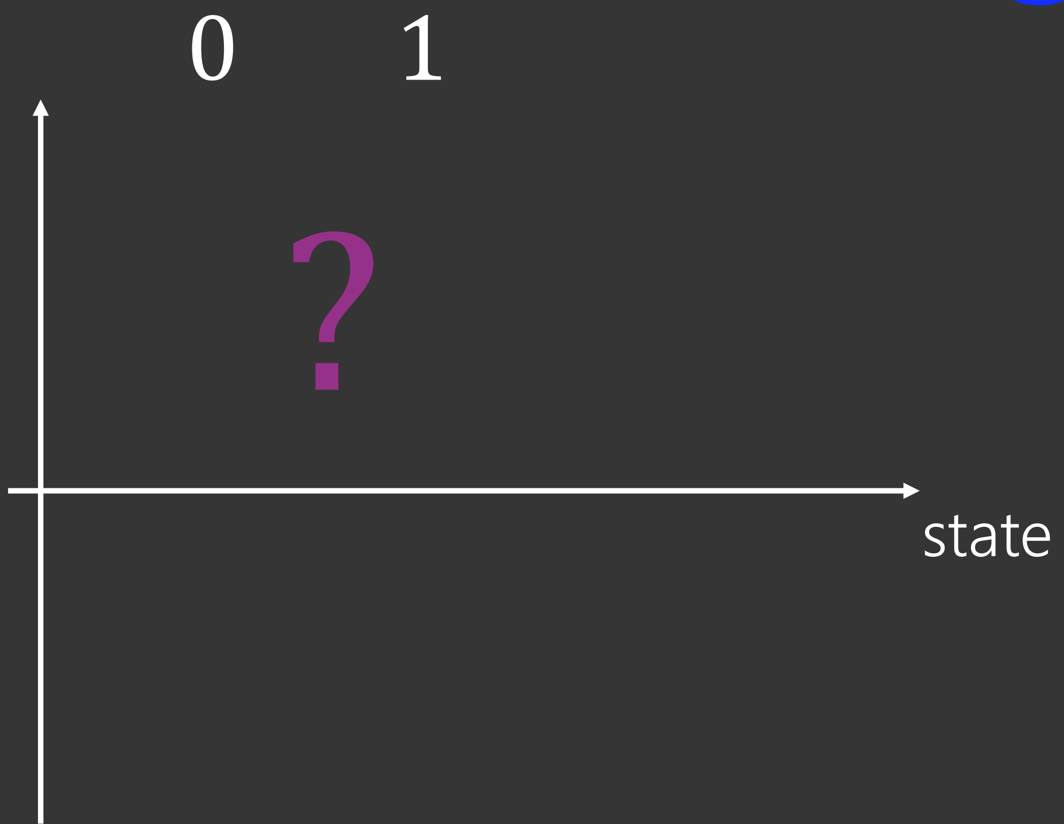




Digital Computation

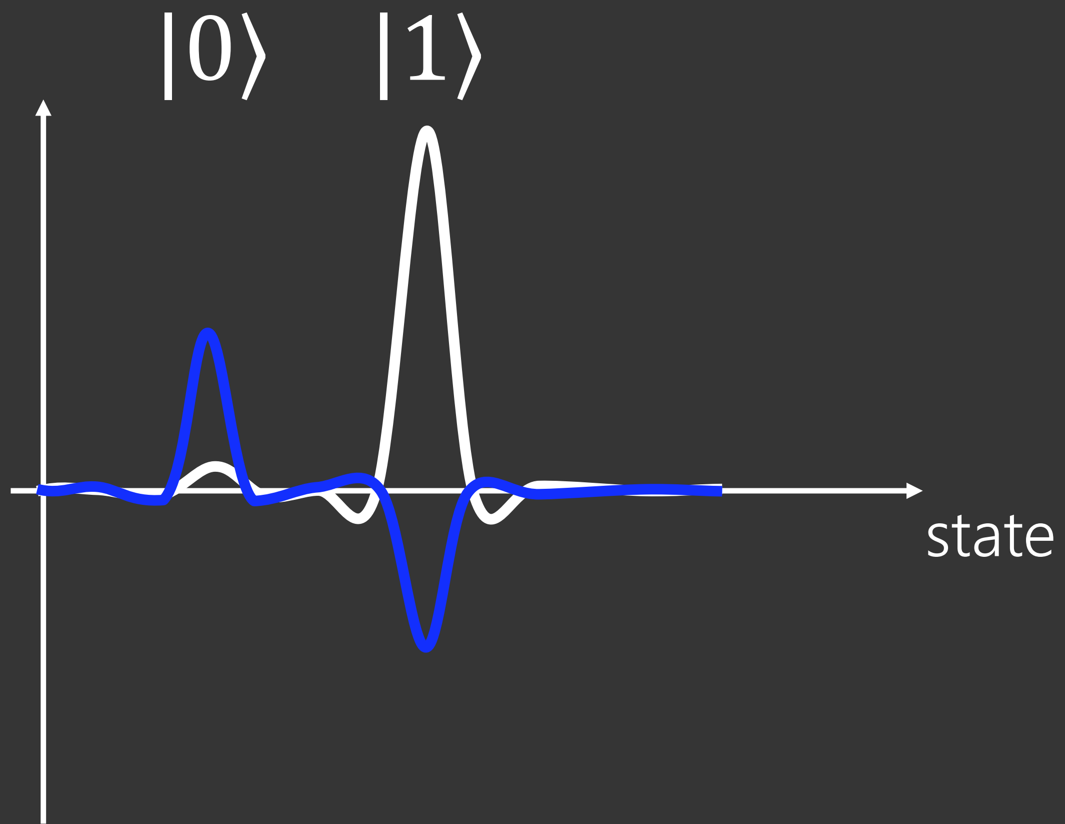
Program description:
discrete sequence of actions (digital)

Classical:



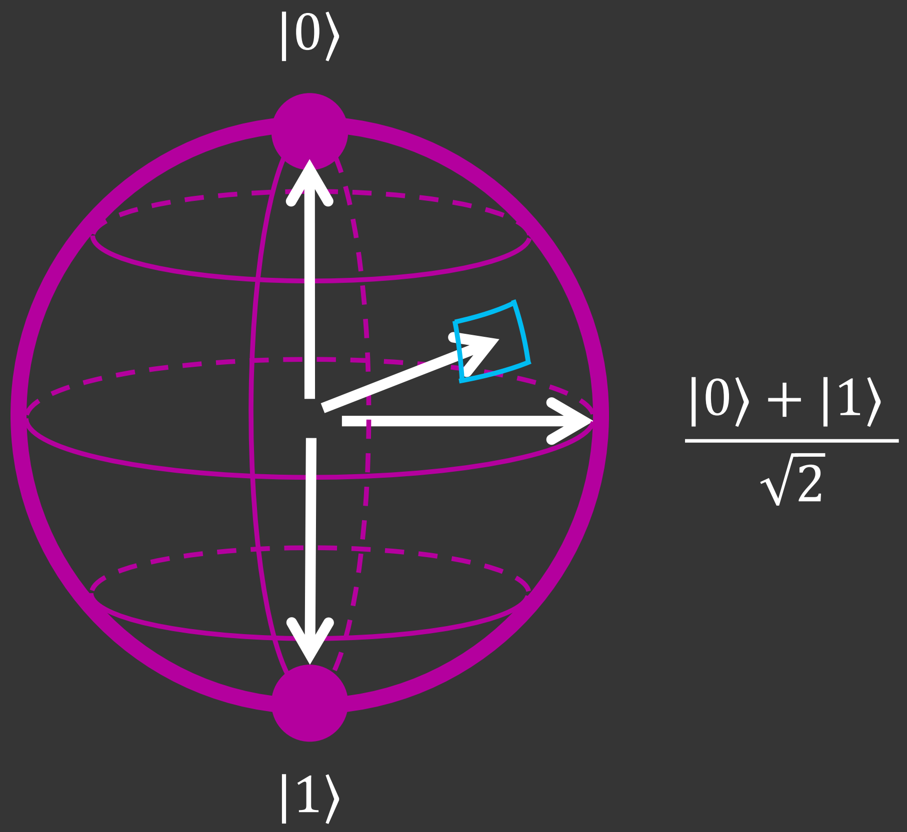
H-gate

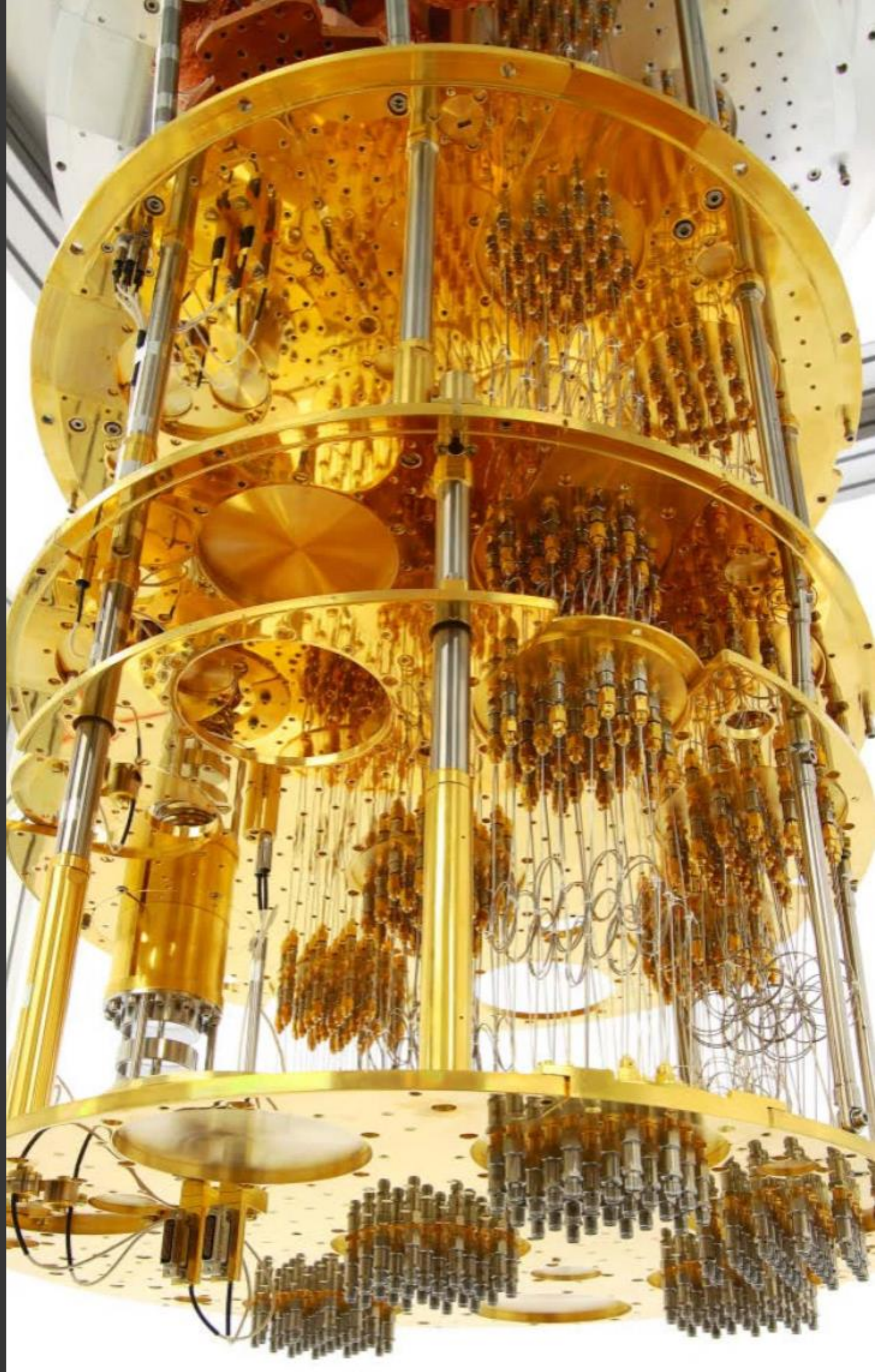
Quantum:



Digital Computation

Program description:
discrete sequence of actions (digital)



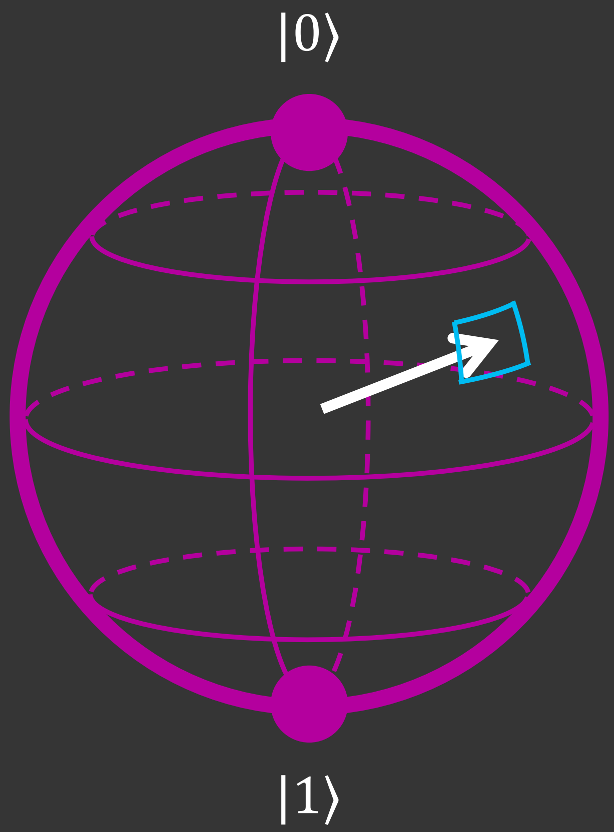


Rotation Synthesis

```
Input: n,φ,δ ⊲ n – T-count, Rz(φ) – target rotation
m ←⌊(n + 1)/2⌋+ 2
for k = 0,1 do
    Lre,k ← FIND-HALVES(cos(φ−πk/8),m,δ)
    Lim,k ← FIND-HALVES(sin(φ−πk/8),m,δ)
end for
Interval I ← [0,α] ⊲ Pick α > 0 based on Lre,k,Lim,k
while I ∩[0,δ] 6= ∅ do
    Find an array A of tuples (ε,a0,b0,a1,b1,k) s.t.:
        • (εre,a0,b0) from Lre,k
        • (εim,a1,b1) from Lim,k
        • ε = εre + εim and ε ∈ I ∩[0,δ]
    Sort A by ε in ascending order
    ε1 < ... < εM ← all distinct ε that occur in A
    for j = 1 to M do
        ∂ ← ∅
        for all (εj,a0,b0,a1,b1) ∈ A do
            x′ ← a0 + b0√2 + i(a1 + b1√2)
            n0 ← MIN-T-COUNT(x′,m,k) ⊲ (computes Tk(x′/√2m)
            if n = n0 then
                ∂ ← ∂ ∪ALL-UNITARIES(x′,m,k) ⊲ minimal unitaries
            end if
        end for
        if ∂ 6= ∅ then
            return (εj,∂) ⊲ Solution
        end if
    end for
    Replace I = [α0,α1] by I = [α1,2α1 −α0]
end while
return (δ,∅) ⊲ No solution
```

[Kliuchnikov, Maslov, Mosca (2014)]

Error Correction



Challenges:

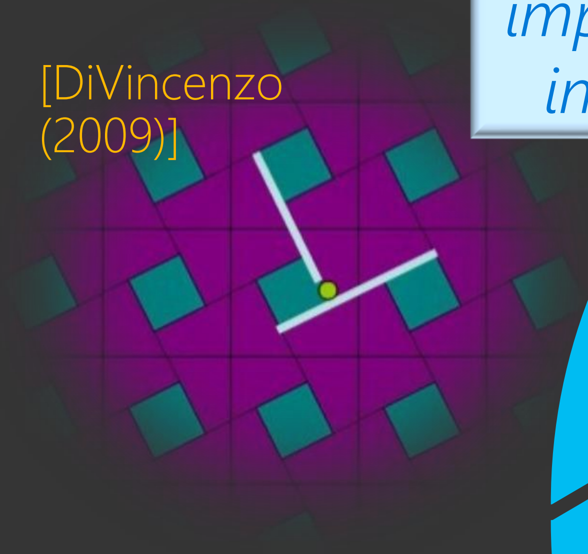
- Detecting errors – we can't look at the state
- Correcting errors – an erroneous state cannot easily be reset
- No duplication or easy comparison of arbitrary quantum states
- The physical space within which the computation takes place is not clearly defined

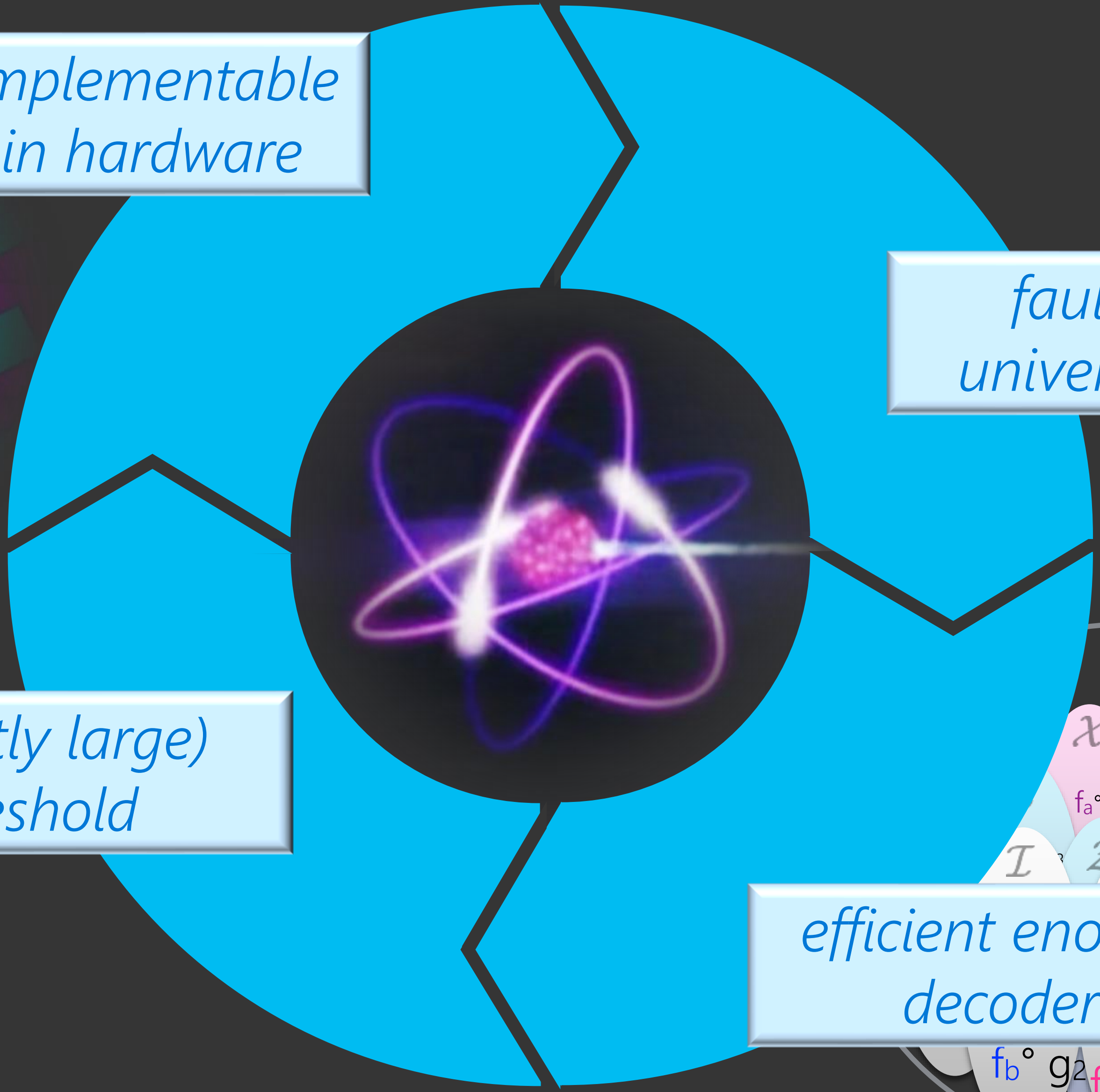
Error correction

Basic idea remains the same:

- *Information is encoded in redundant form into global degrees of freedom.*
- *Computations take place within a clearly defined subspace, and (any) local noise causes a perturbation outside that subspace.*
- *Suitable projective measurements*
  a. *allow to detect these perturbations*
  b. *project any error onto a discrete set of errors*
  c. *give information that can be used to (attempt to) classify the error*
  d. *leave the computation space unchanged*

Error correction



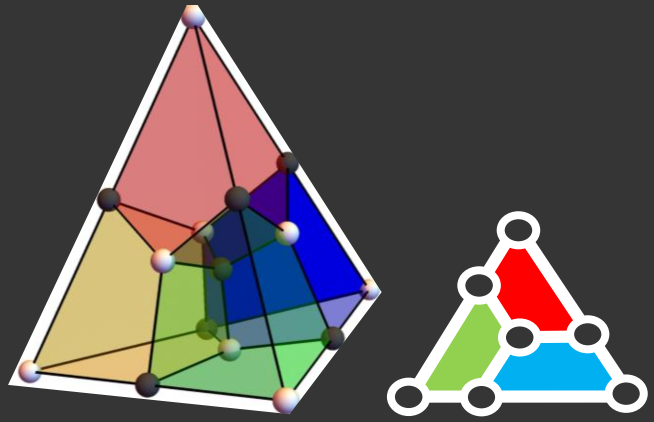


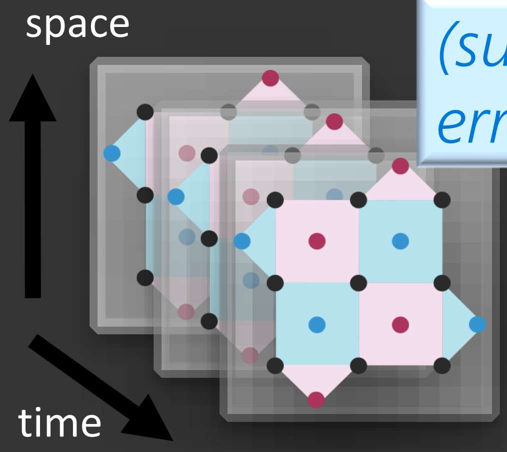
implementable in hardware

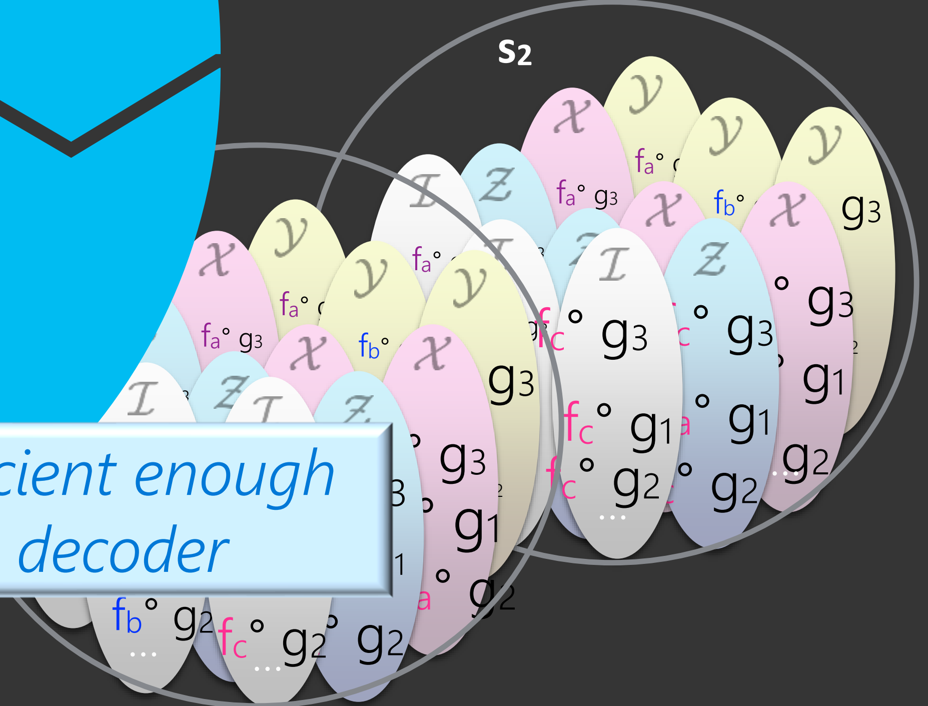
fault-tolerant universal gate set

(sufficiently large) error threshold

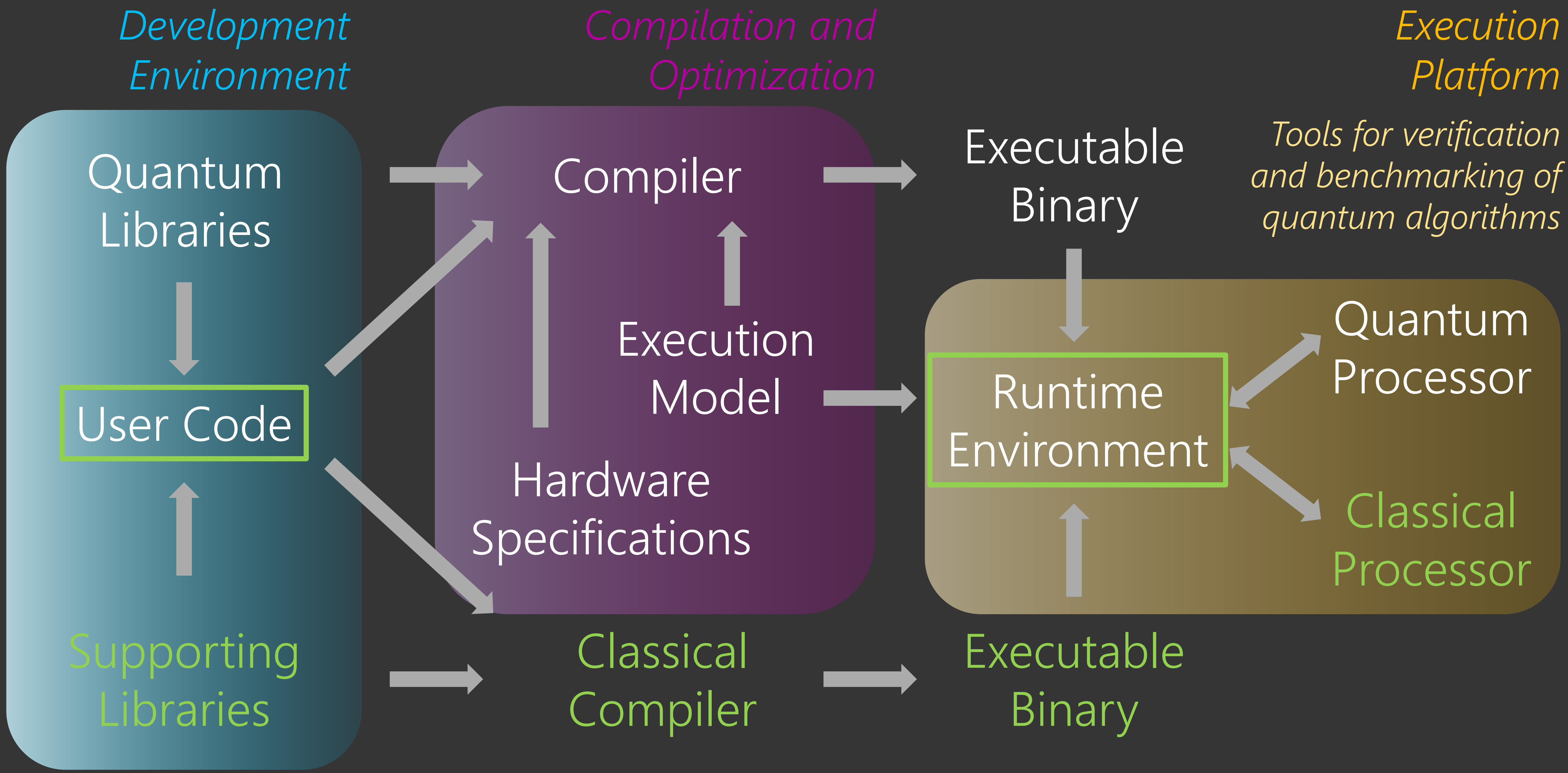
efficient enough decoder







Quantum Software Framework



Quantum Software Framework

Abstraction

- Hardware independent formulation of mathematical concepts
- Algorithm formulation on a logical level
- Encapsulation

Validation

- Resource requirements
- Correctness of the algorithm
- Verifiable behavior

Error Sources

- Algorithmic Errors
- Approximation Errors
- Hardware Errors

Resource Management

- Memory management
- Asynchronous execution
- Classical processing
- Hardware specific optimization

→ Hardware specifications
→ Classical/quantum coordination
→ Precision distribution
→ Available information

→ Context dependent dispatch
→ Performance metrics
→ Static vs. runtime
→ Heuristics

Compilation process

Library: variations for each quantum (sub-)routine

User code defining an algorithm, optimization of algorithmic errors

Dependency model of subroutines, constant folding, optimization of the overall error

Subroutine dispatch based on hardware, erasure of subroutine boundaries

Exploiting (de-facto) commutation relations to reduce algorithm cost

Optimization of synthesis errors

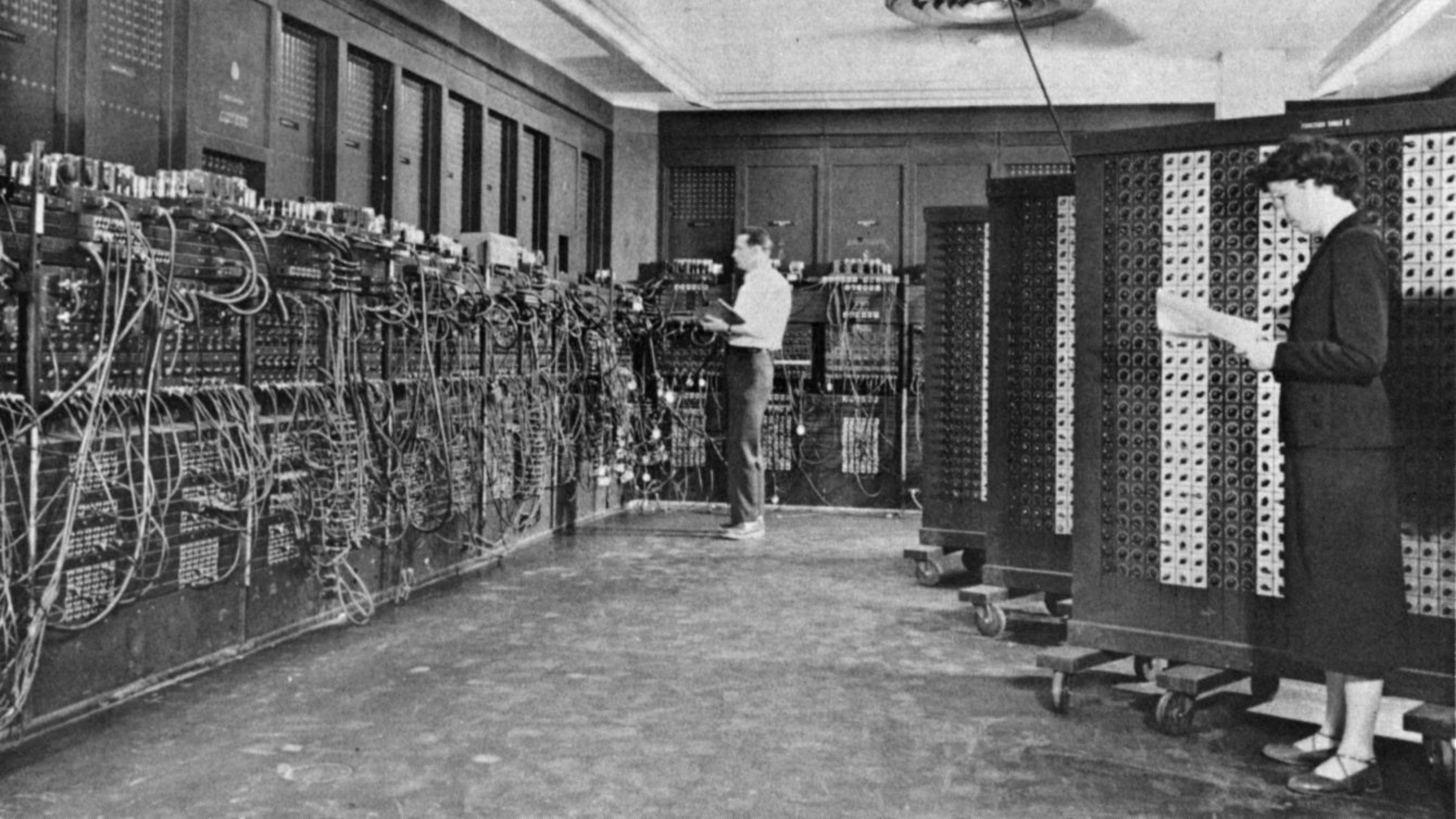
Determine state distillation routines (possibly dynamic)

Physical layout, “routing” (dynamic and/or look-up)

Applying or tracking error correction, communication for runtime compilation

Choice of error correction code

- *What is the relevant information?*
- *How do we obtain the necessary information?*
- *How do we represent that information?*
- *How do we use that information?*
- *How do we generalize this process?*



WWW.MICROSOFT.COM/QUANTUM

Learn more about our approach

Get started with Quantum

Invent the future