Funded by the European Union

Advancing Multi-Dimensional Quantum Computing: Design Automation and Software Tools

Kevin Mato[1], and Robert Wille[1,2]

[1]Technical University of Munich, Munich, Germany
[2]Software Competence Center Hagenberg (SCCH) GmbH, Austria

kevin.mato@tum.de
Technical University of Munich
https://www.cda.cit.tum.de/research/quantum

European Research Council
Established by the European Commission
Munich Quantum Valley
NeQST
scch software competence center hagenberg
universität innsbruck

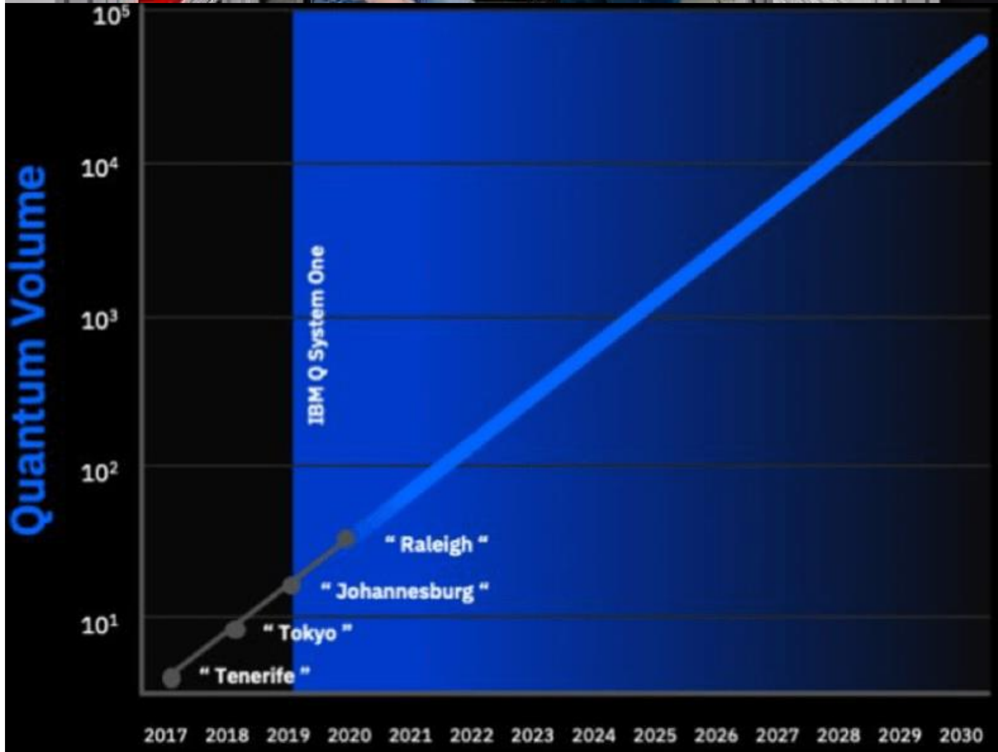
Quantum Computing:
the Next Big thing

- Global Players are heavily investing
  - IBM, Google, Microsoft, Amazon
  - Exciting startups landscape (Quantinuum, Xanadu…)
  - **Exponential improvements** in the best case

- Killer Applications: physics simulation, machine learning, chemistry, unstructured search, …

- Example: Haber-Bosch Process
  - 1-2% of world‘s energy consumption
  - 3-5% of world‘s gas production ($11 Billion)

- Several ambitious roadmaps …



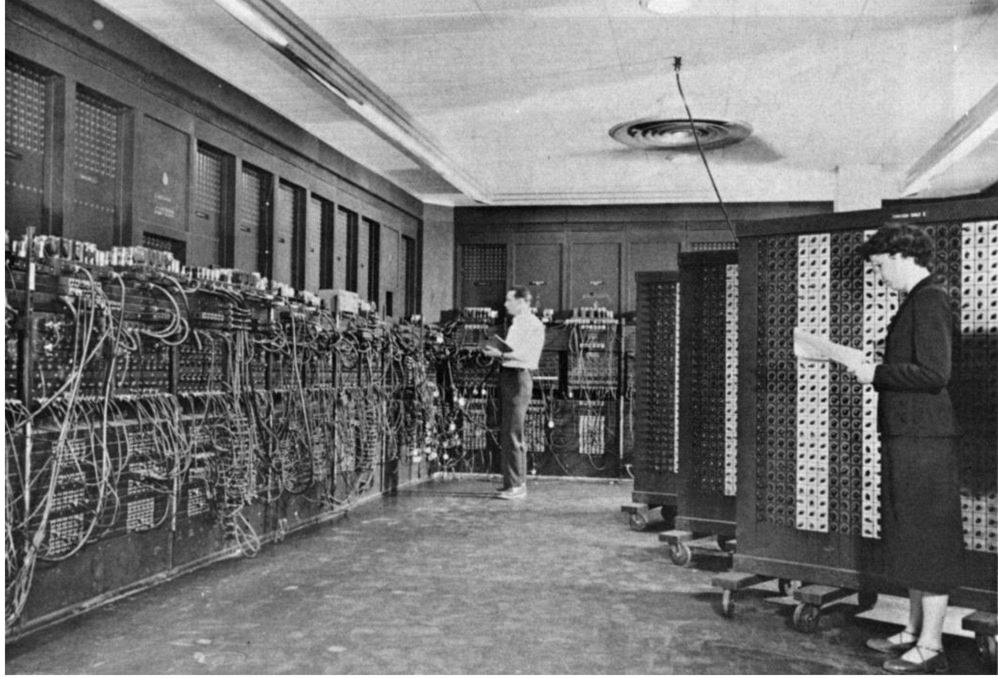
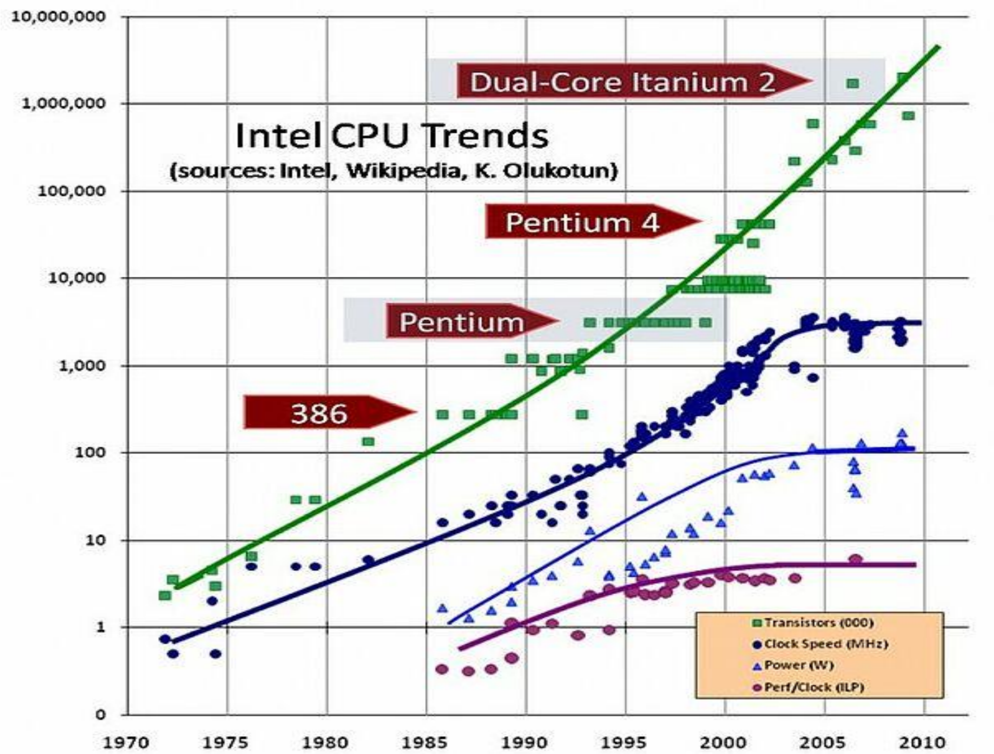
Analogy to Conventional Computers

- Similar picture if we look back in time
  - First, bulky computers
  - Moores law
  - Digital revolution



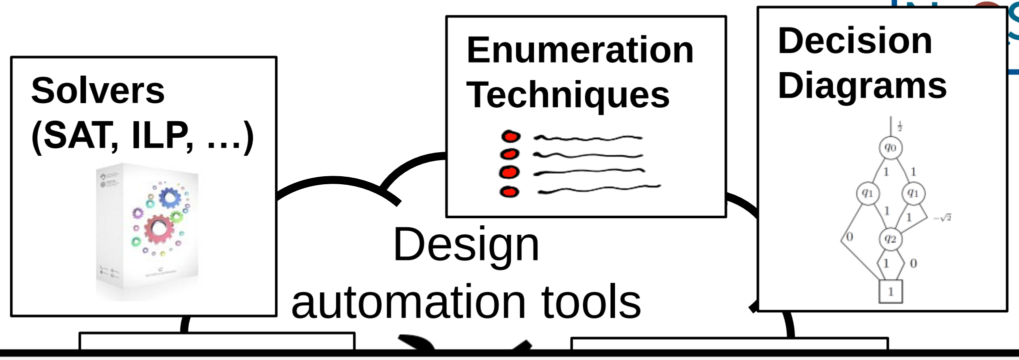




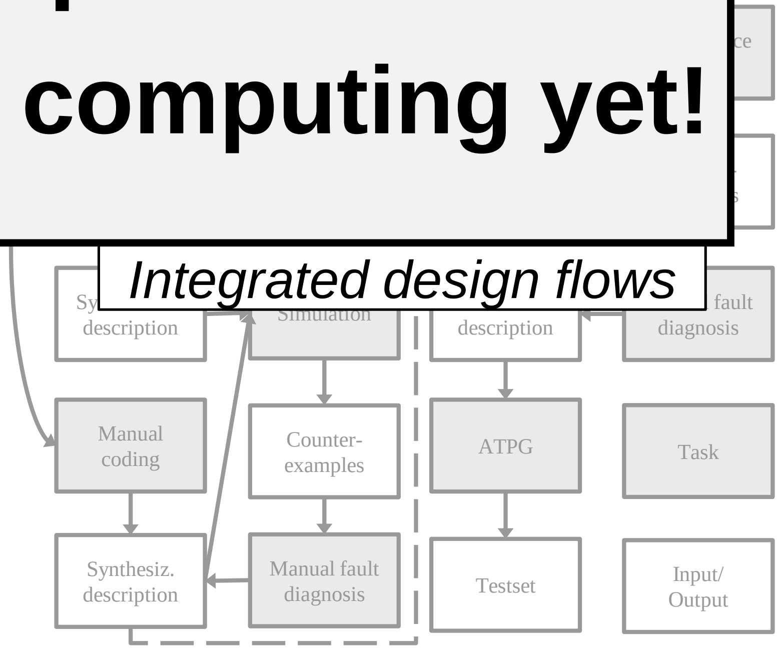
Analogy to Conventional Computers

- Similar picture if we look back in time
  - First, bulky computers



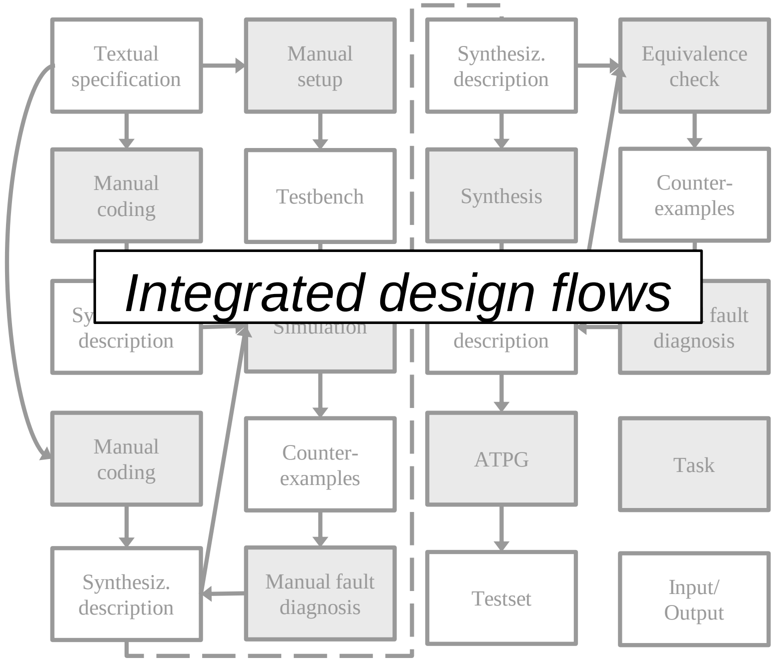
Allowed the design of systems composed of billions of components.

But hardly exploited for quantum computing yet!



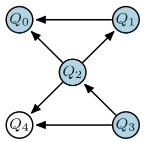
Design Automation and Software for Quantum Computing

MUNICH QUANTUM TOOLKIT

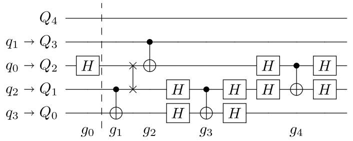


- *Applications*
- *Compilation*
- *Simulation*
- *Verification*
- *Data structures*
- *Visualization*

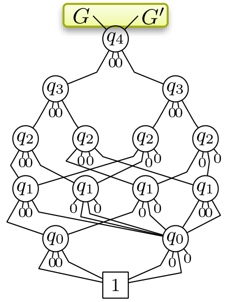
- Problem Solver



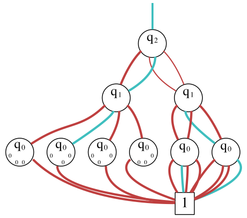
- QMAP



- DDSIM

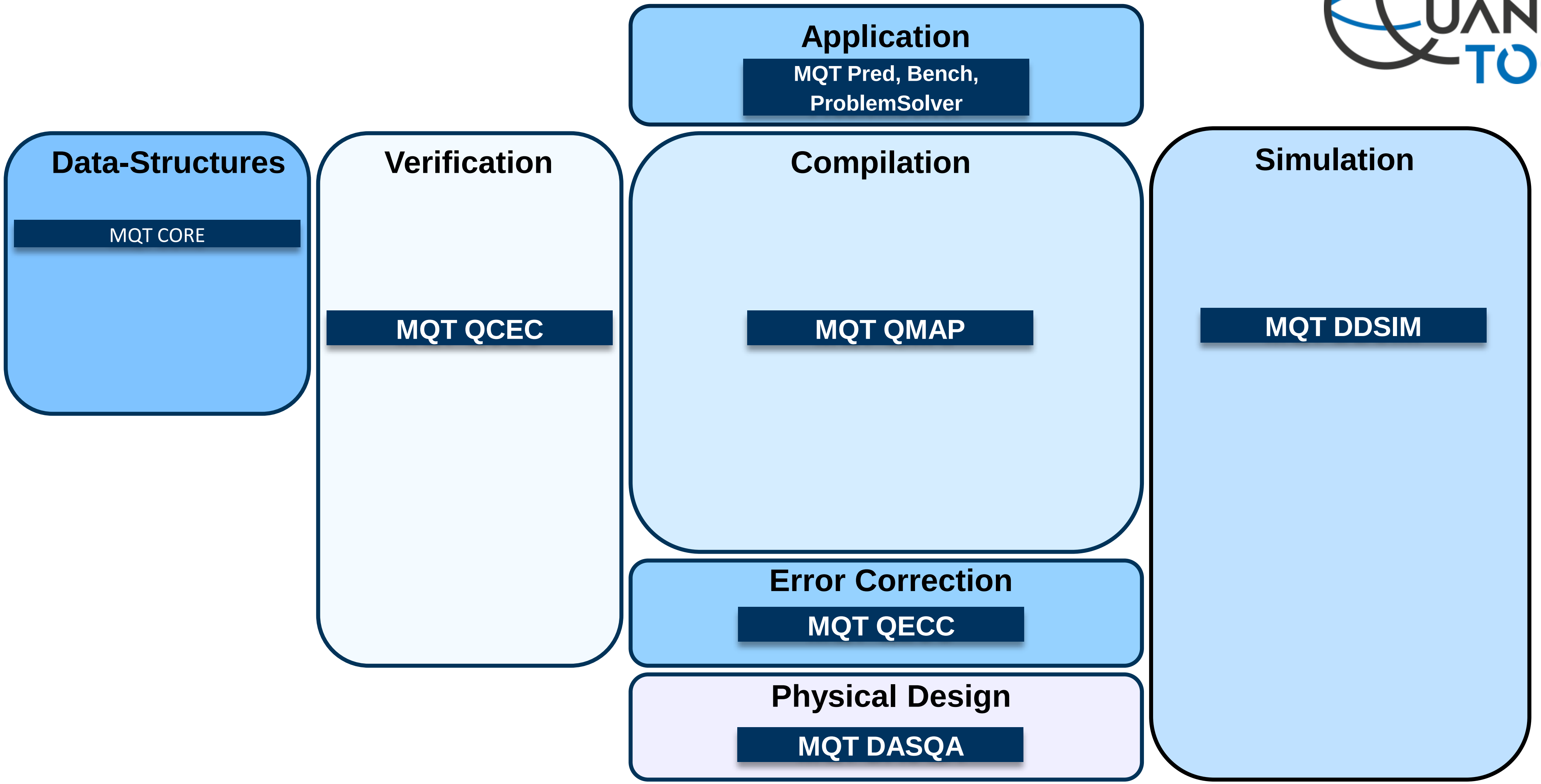


- QCEC
- Decision Diagrams
- Tensor Networks
- ZX

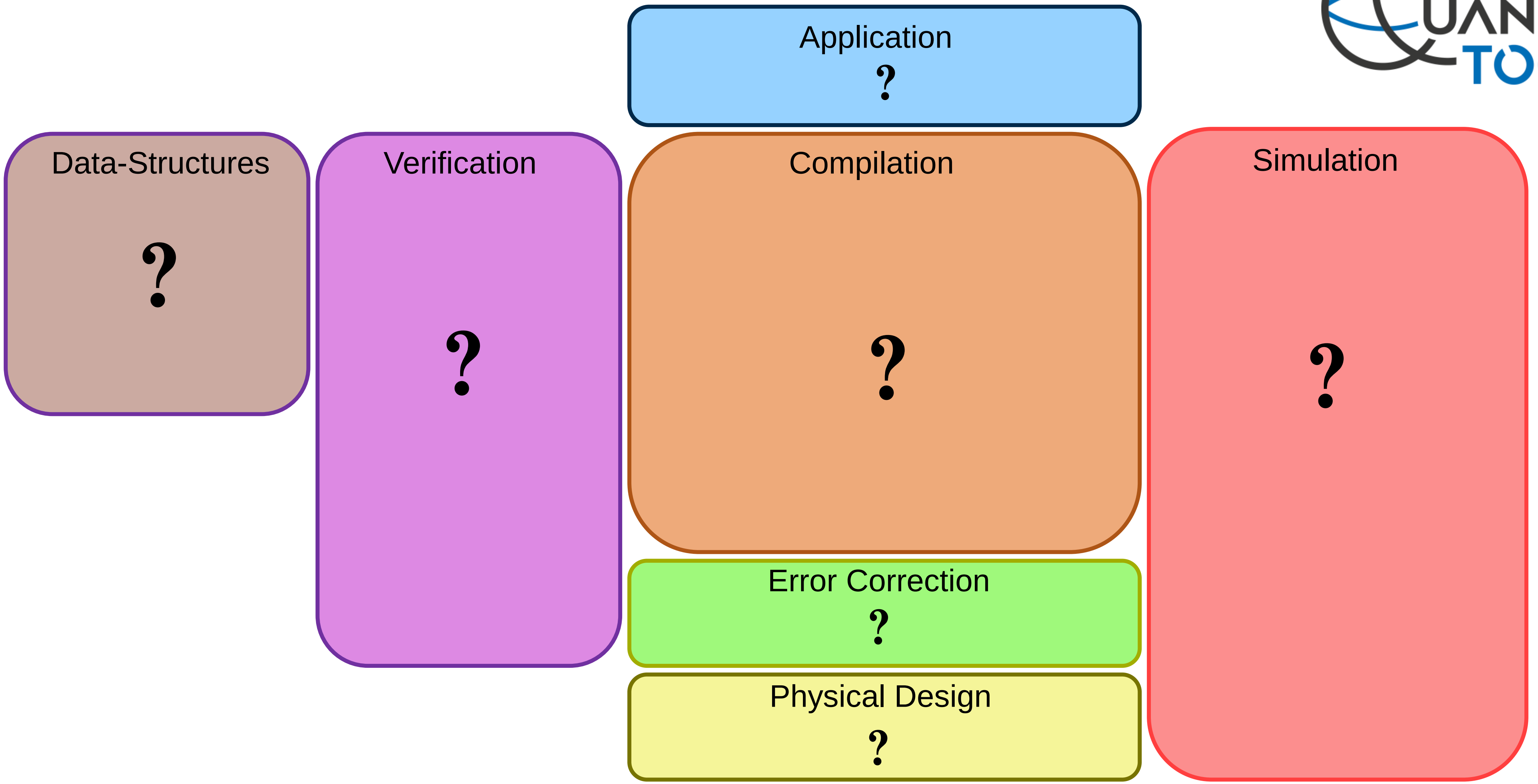


- DDVIS

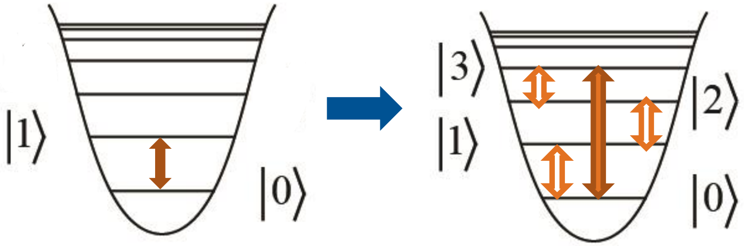
The Stack From Above



The *Qudit* Stack From Above



Introduction: Qudits

*

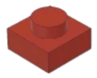
Qubit Qudit

- *Subspace operations*
- *Global Hilbert space operations*

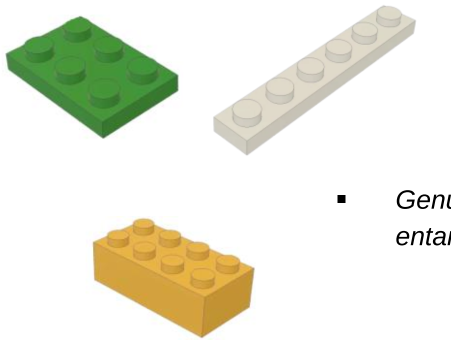
Why Qudits are so interesting?

- Qudits can be implemented on the latest quantum technologies
- Much richer entanglement structure of qudits compared to qubits
- Better circuit complexity and algorithmic efficiency, at an increasing design cost

- Mixed Dimensional Systems



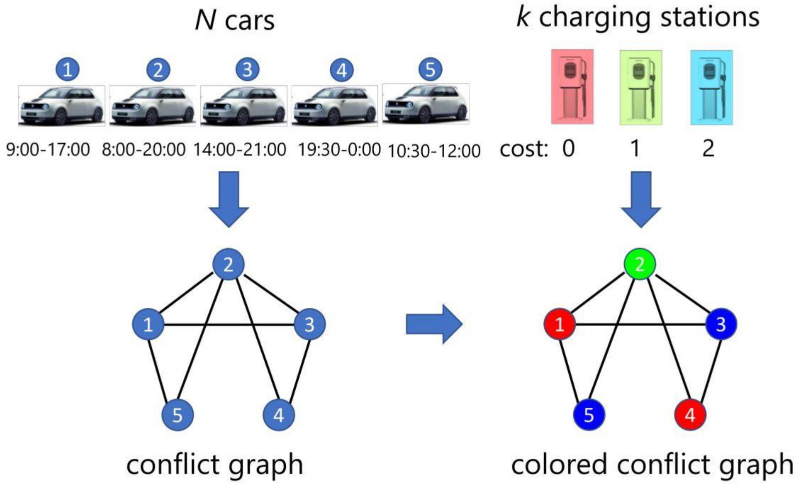
+



- *CNOT.*
- *Genuine qudit entangling gates.*

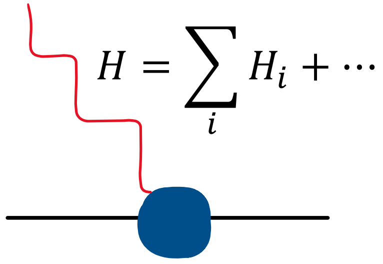
*https://labsites.rochester.edu/bloklab/index.php/qudits/

Mixed-Dimensional Systems



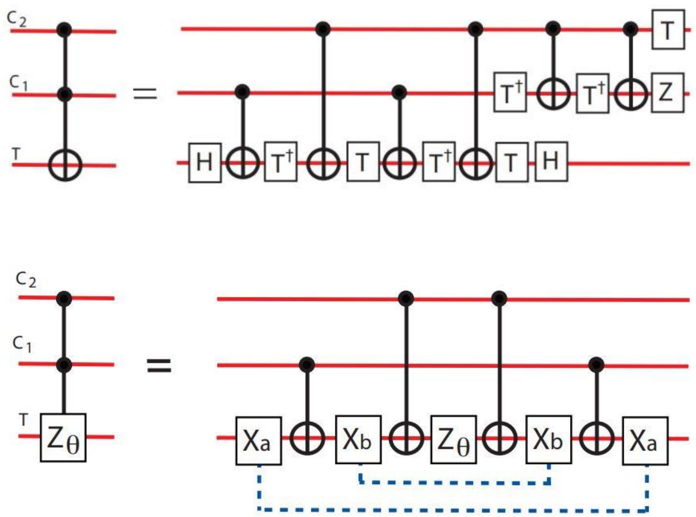
- Integer optimization

arXiv:2204.00340



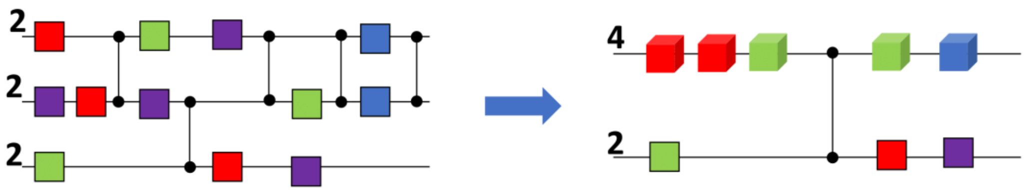
- Fermion-Boson Interactions

arXiv:1312.2849



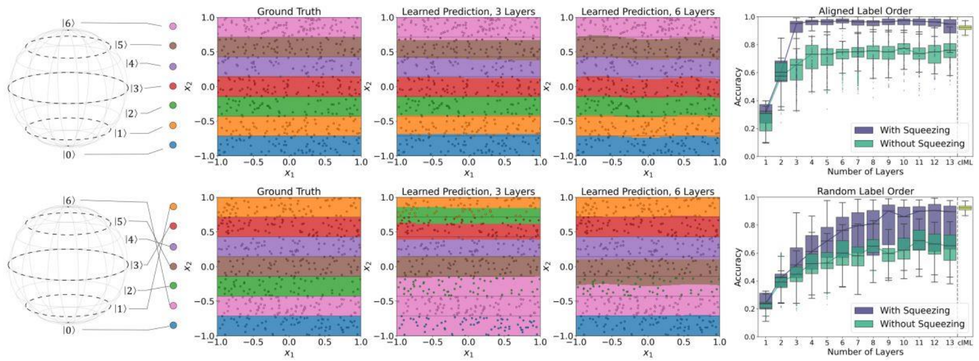
- Compression: Higher Dimensions

arXiv:0804.0272



- Compression: Problem Reincoding

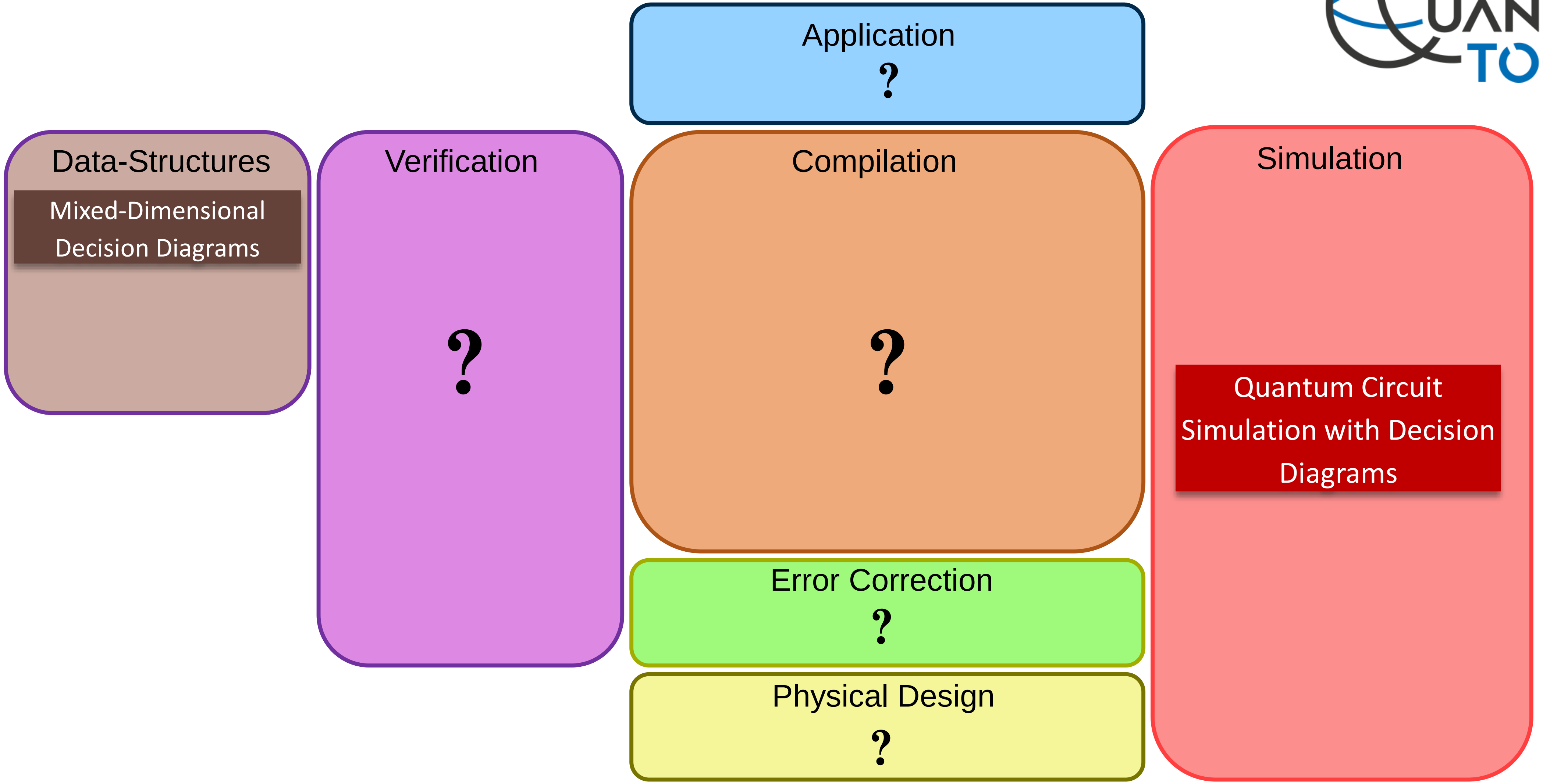
QSW59989.2023.00027



- Quantum Machine Learning

arXiv:2302.13932

The *Qudit* Stack From Above

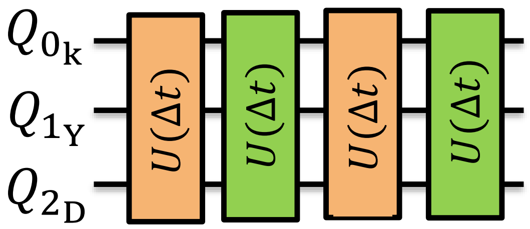


Quantum Circuits

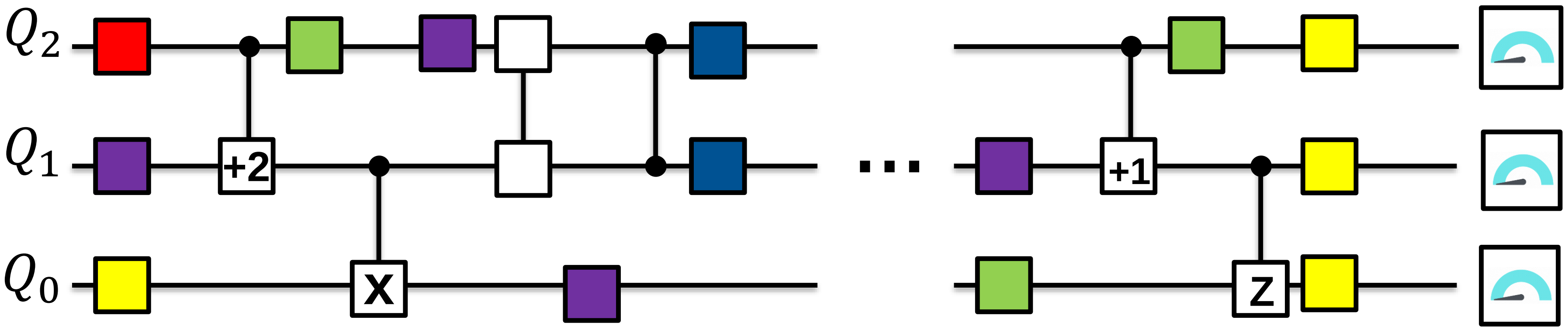
- Objective function or algorithm: $f(x) = \sum_{i=1}^{K} a_i + \sum_{i,j,k=1}^{Y} p_i p_j p_k + \ldots + \sum_{l=1}^{D} c_l$

Compilation

- Quantum algorithm:



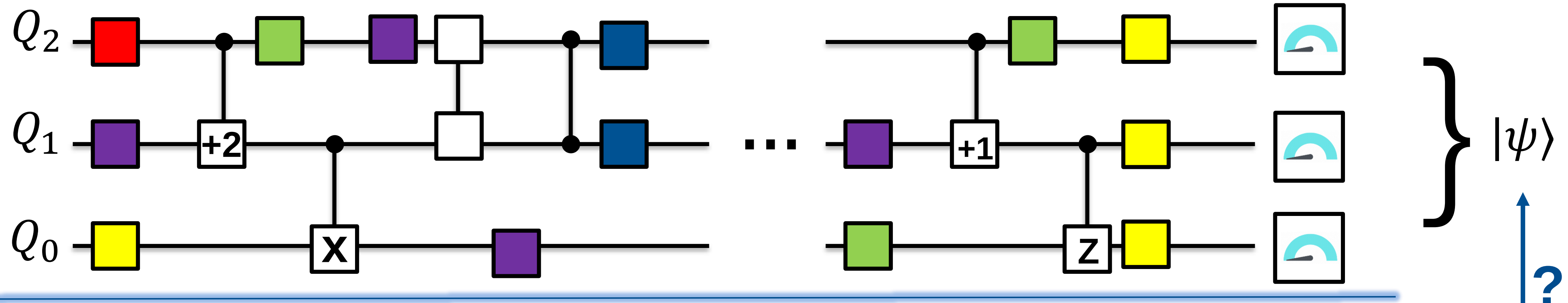
Compilation



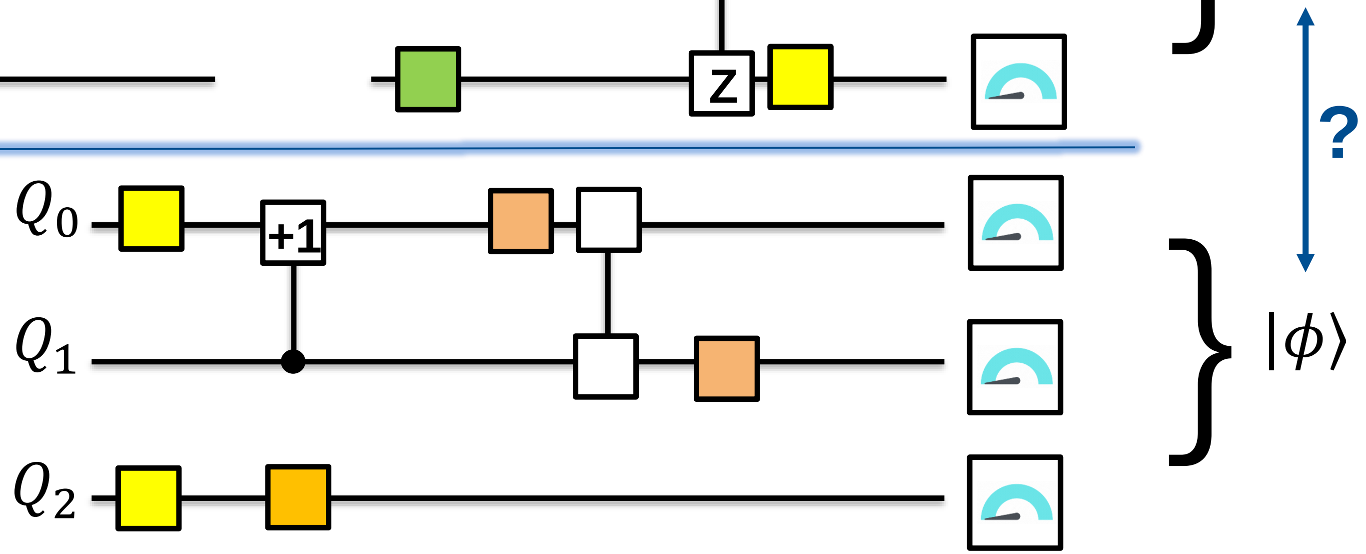
The Need For Simulation

There are four immediate advantages in having an appropriate simulator for mixed-dimensional quantum systems:

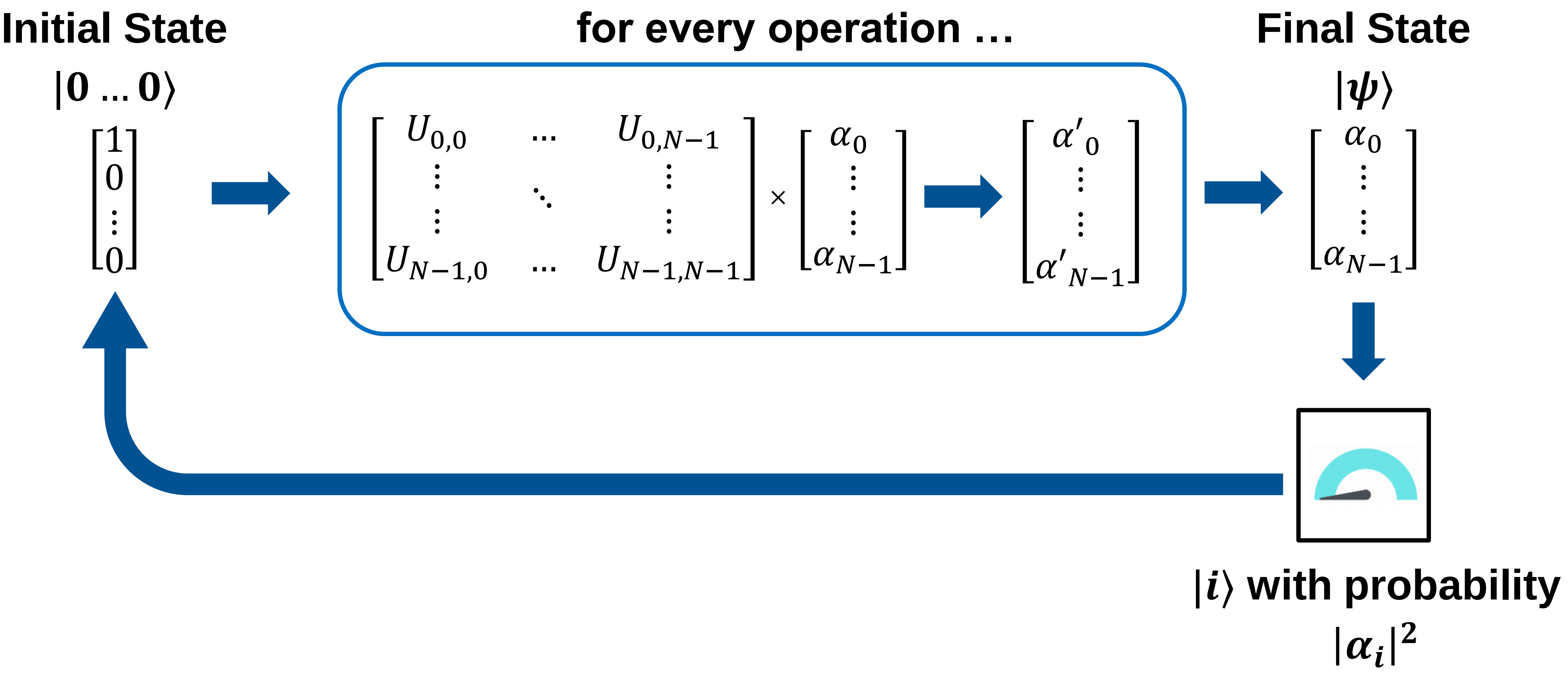
- Getting otherwise opaque information about the quantum state
- Aiding in verification



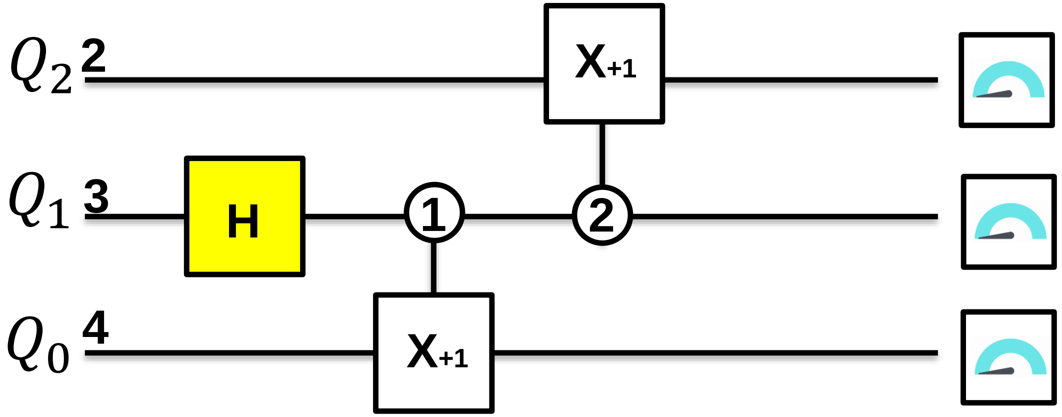
- Enabling design exploration (also noise-aware)
- Identifying potential for compression



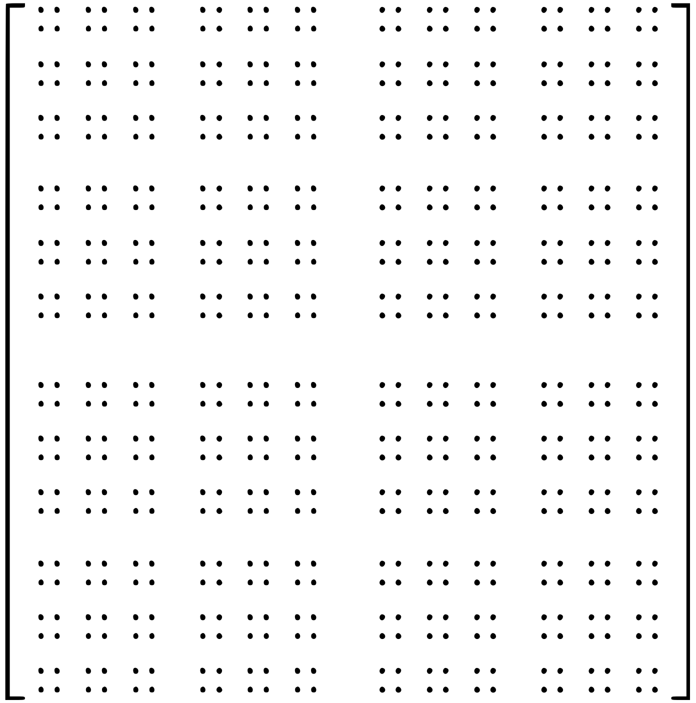
Quantum Circuits Simulation



Quantum Circuits Simulation



$I_2 \otimes H_3 \otimes I_4 =$



- Matrix vector multiplication:

$$H_3 \cdot |0\rangle = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{\frac{2\pi}{3}} & e^{\frac{-2\pi}{3}} \\ 1 & e^{\frac{-2\pi}{3}} & e^{\frac{2\pi}{3}} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

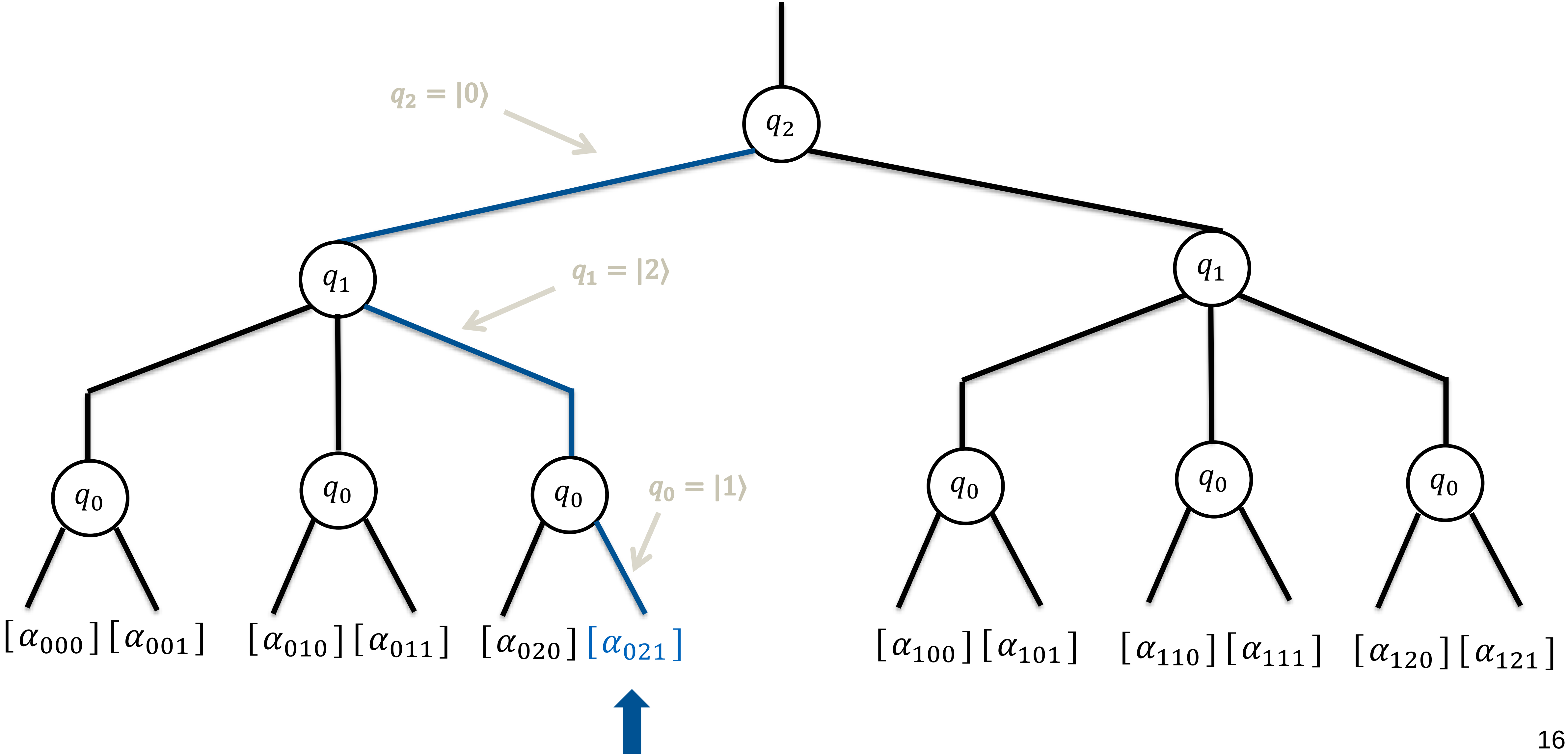
- Exponential complexity

➔ ***Efficient representation required***

Quantum Logic with Decision Diagrams

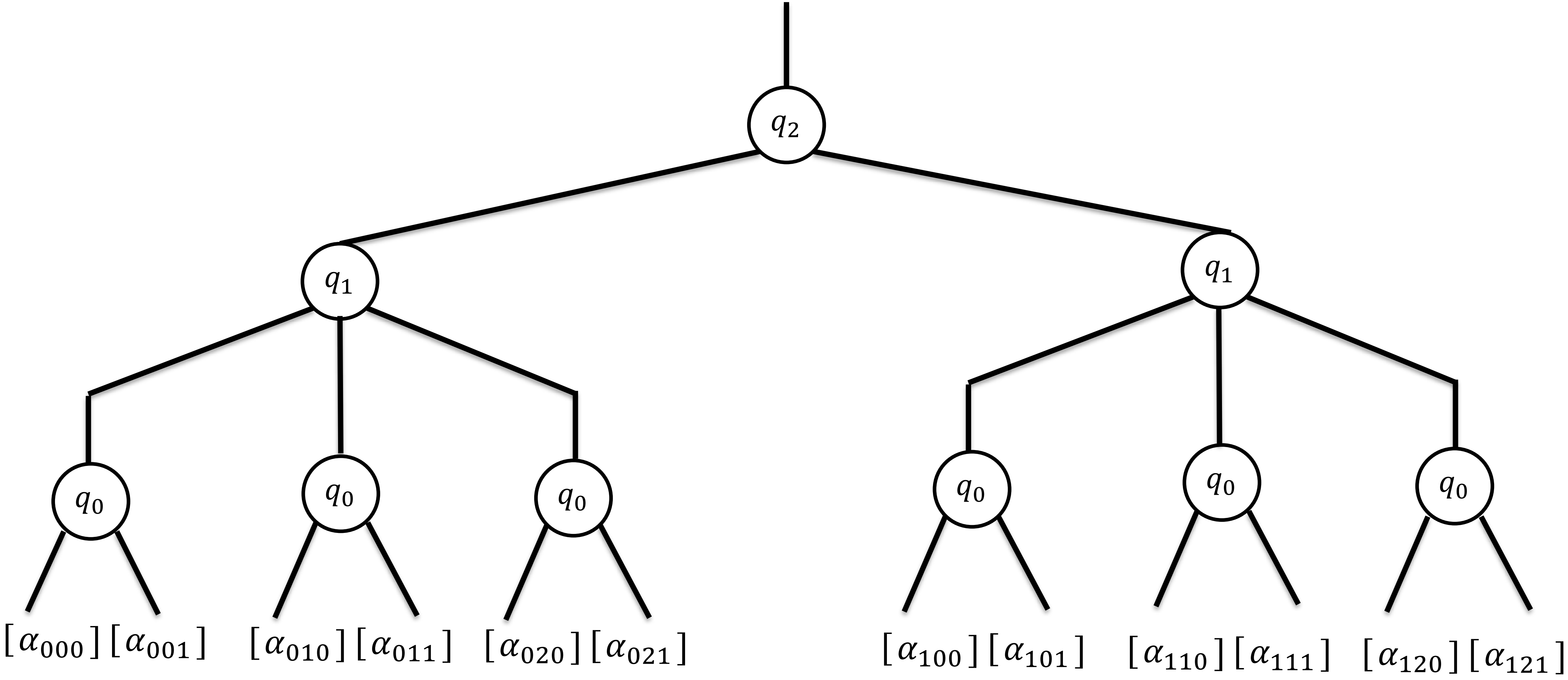
$$[\alpha_{000} \quad \alpha_{001} \quad \alpha_{010} \quad \alpha_{011} \quad \alpha_{020} \quad \alpha_{021} \quad \alpha_{100} \quad \alpha_{101} \quad \alpha_{110} \quad \alpha_{111} \quad \alpha_{120} \quad \alpha_{121}]$$

Structure



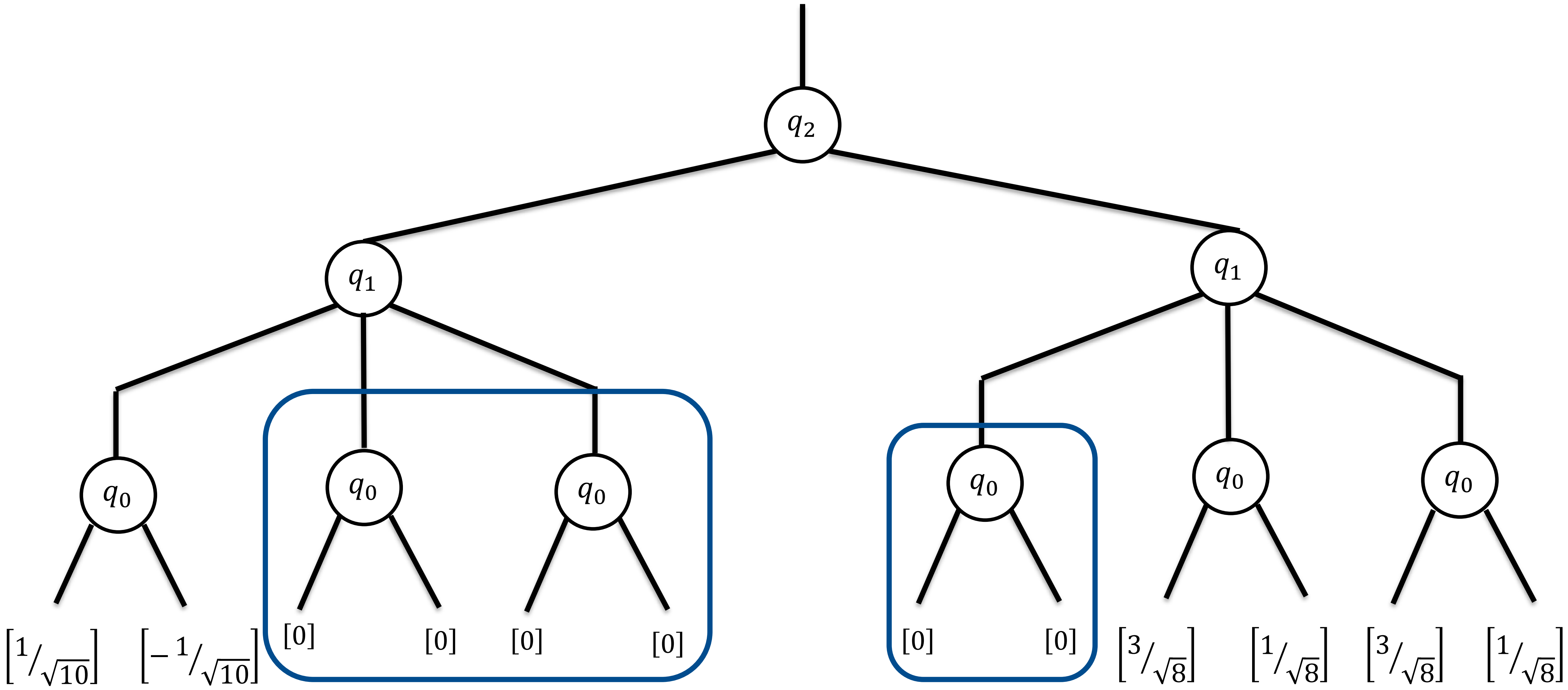
$q_2 = |0\rangle$
q_2
q_1
$q_1 = |2\rangle$
q_1
q_0
q_0
q_0
$q_0 = |1\rangle$
q_0
q_0
q_0
$[\alpha_{000}]\,[\alpha_{001}]$
$[\alpha_{010}]\,[\alpha_{011}]$
$[\alpha_{020}]\,[\alpha_{021}]$
$[\alpha_{100}]\,[\alpha_{101}]$
$[\alpha_{110}]\,[\alpha_{111}]$
$[\alpha_{120}]\,[\alpha_{121}]$

Structure



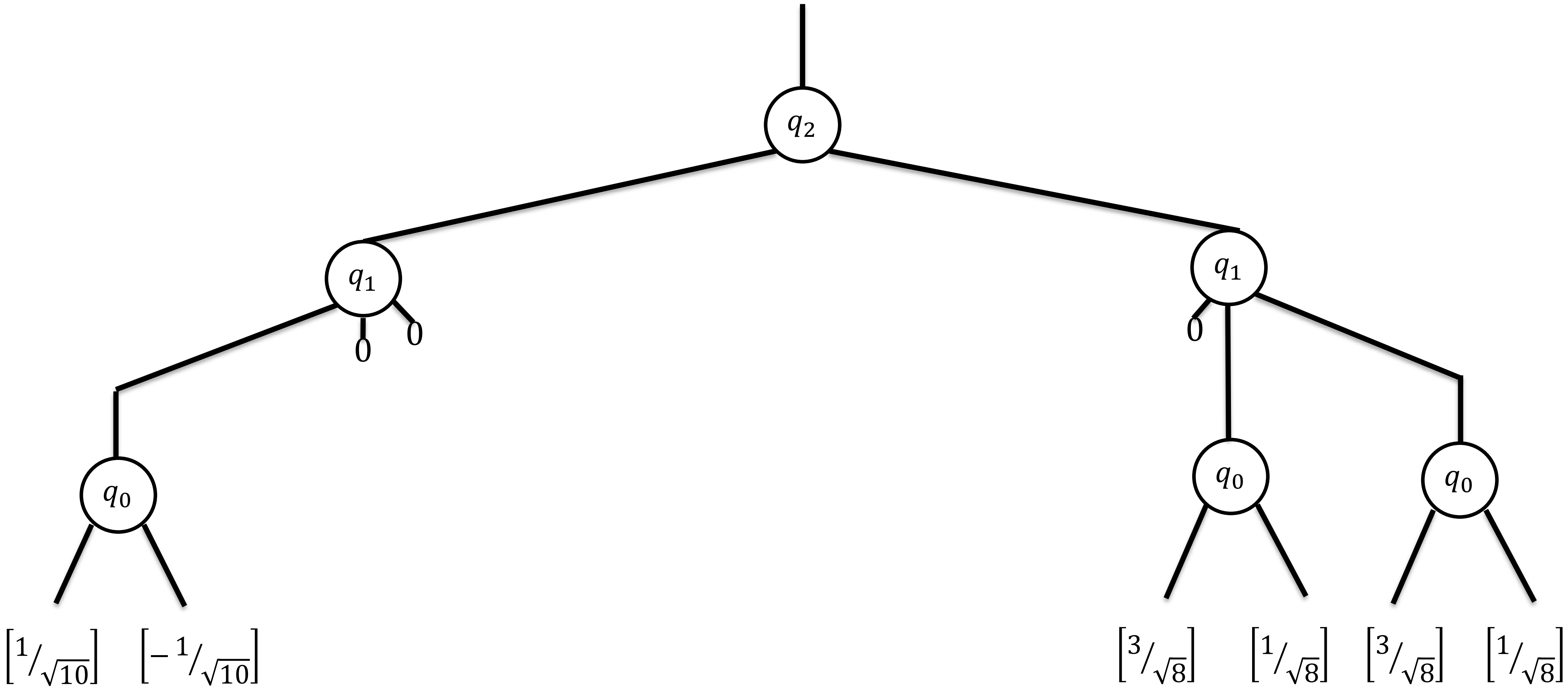
q_2
q_1
q_1
q_0
q_0
q_0
q_0
q_0
q_0
$[\alpha_{000}]$ $[\alpha_{001}]$
$[\alpha_{010}]$ $[\alpha_{011}]$
$[\alpha_{020}]$ $[\alpha_{021}]$
$[\alpha_{100}]$ $[\alpha_{101}]$
$[\alpha_{110}]$ $[\alpha_{111}]$
$[\alpha_{120}]$ $[\alpha_{121}]$

Structure

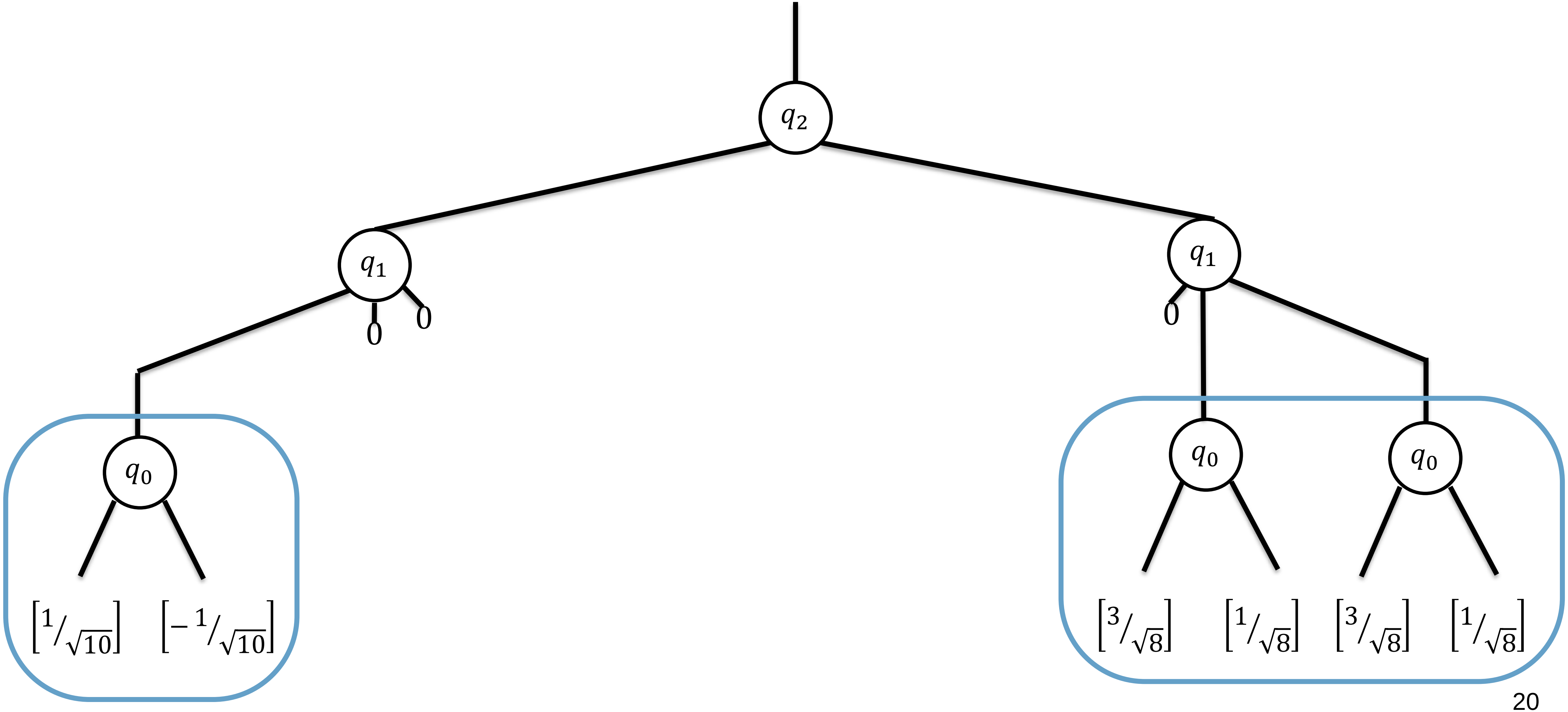


q_2
q_1
q_1
q_0
q_0
q_0
q_0
q_0
q_0
$[1/\sqrt{10}]$
$[-1/\sqrt{10}]$
[0]
[0]
[0]
[0]
[0]
[0]
$[3/\sqrt{8}]$
$[1/\sqrt{8}]$
$[3/\sqrt{8}]$
$[1/\sqrt{8}]$

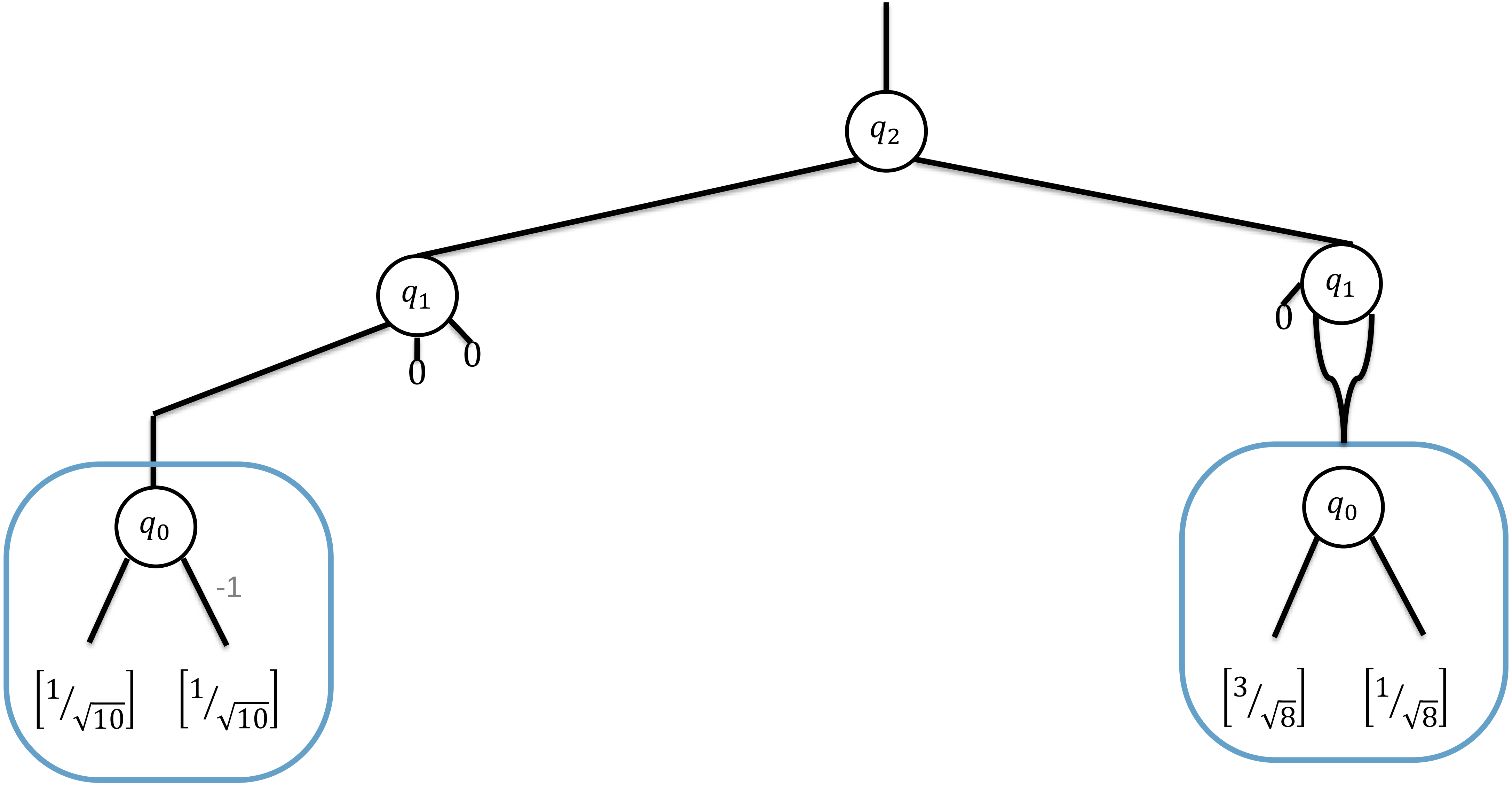
Structure + Sparsity



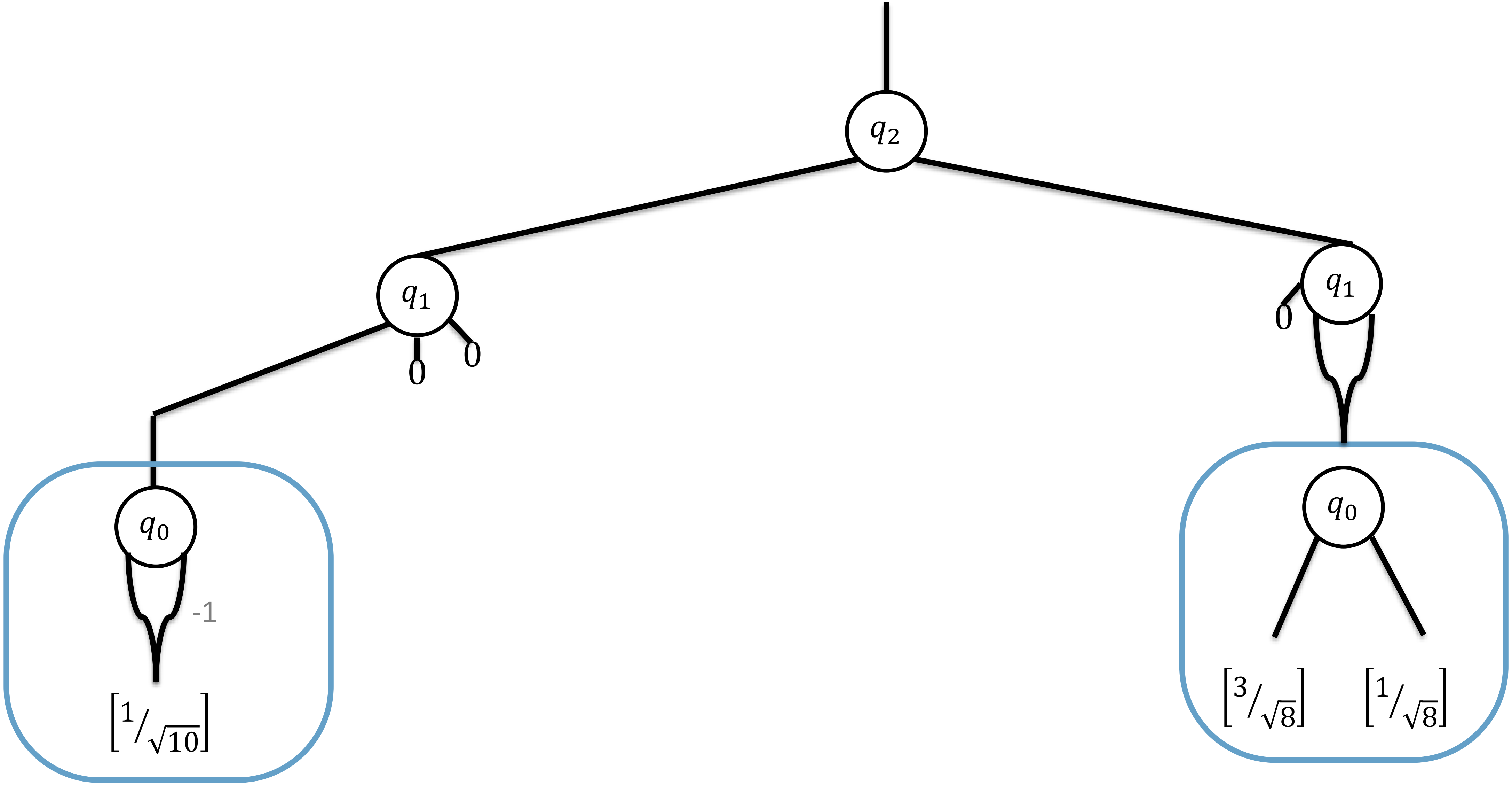
Structure + Sparsity + Redundancy



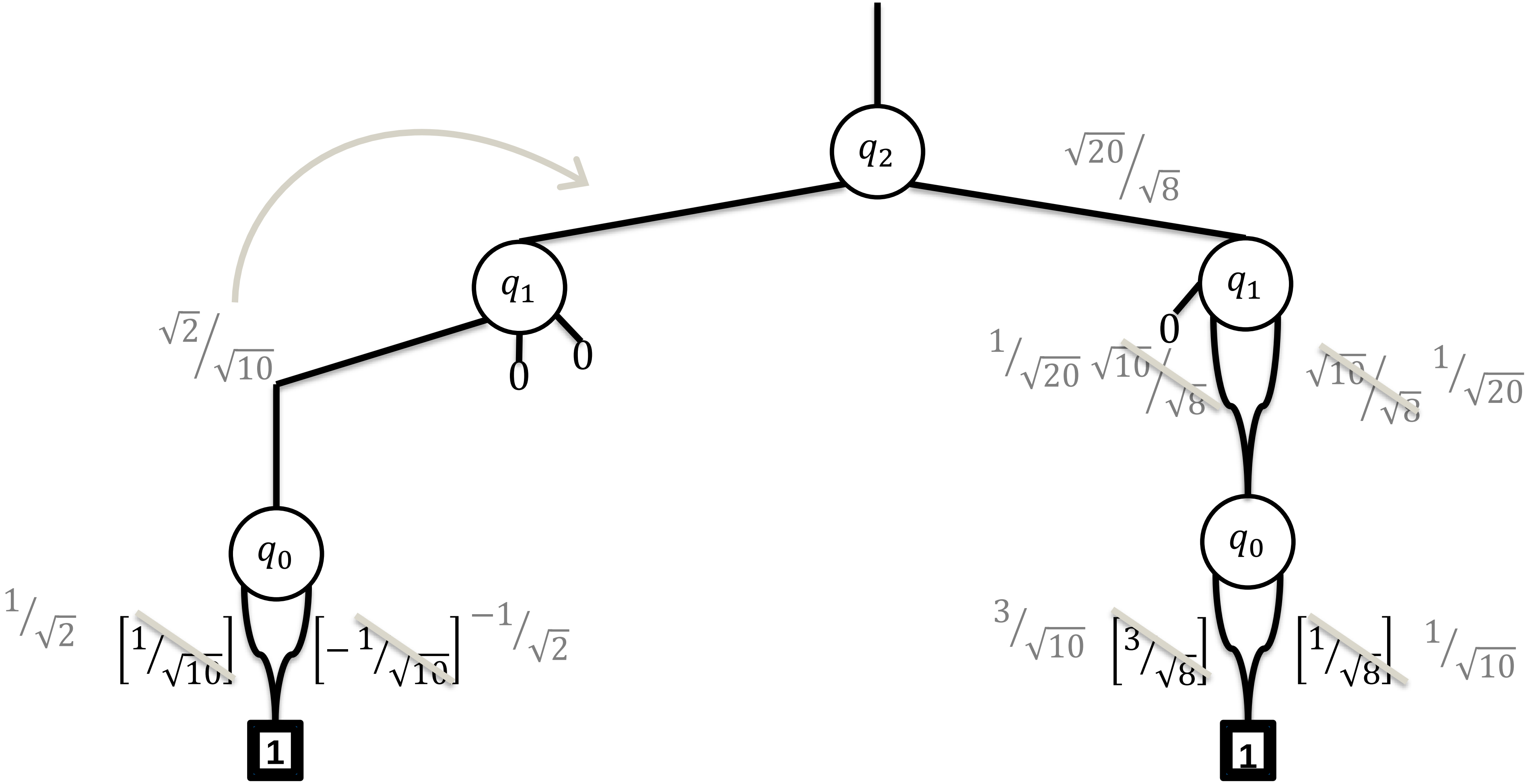
Structure + Sparsity + Redundancy



Structure + Sparsity + Redundancy

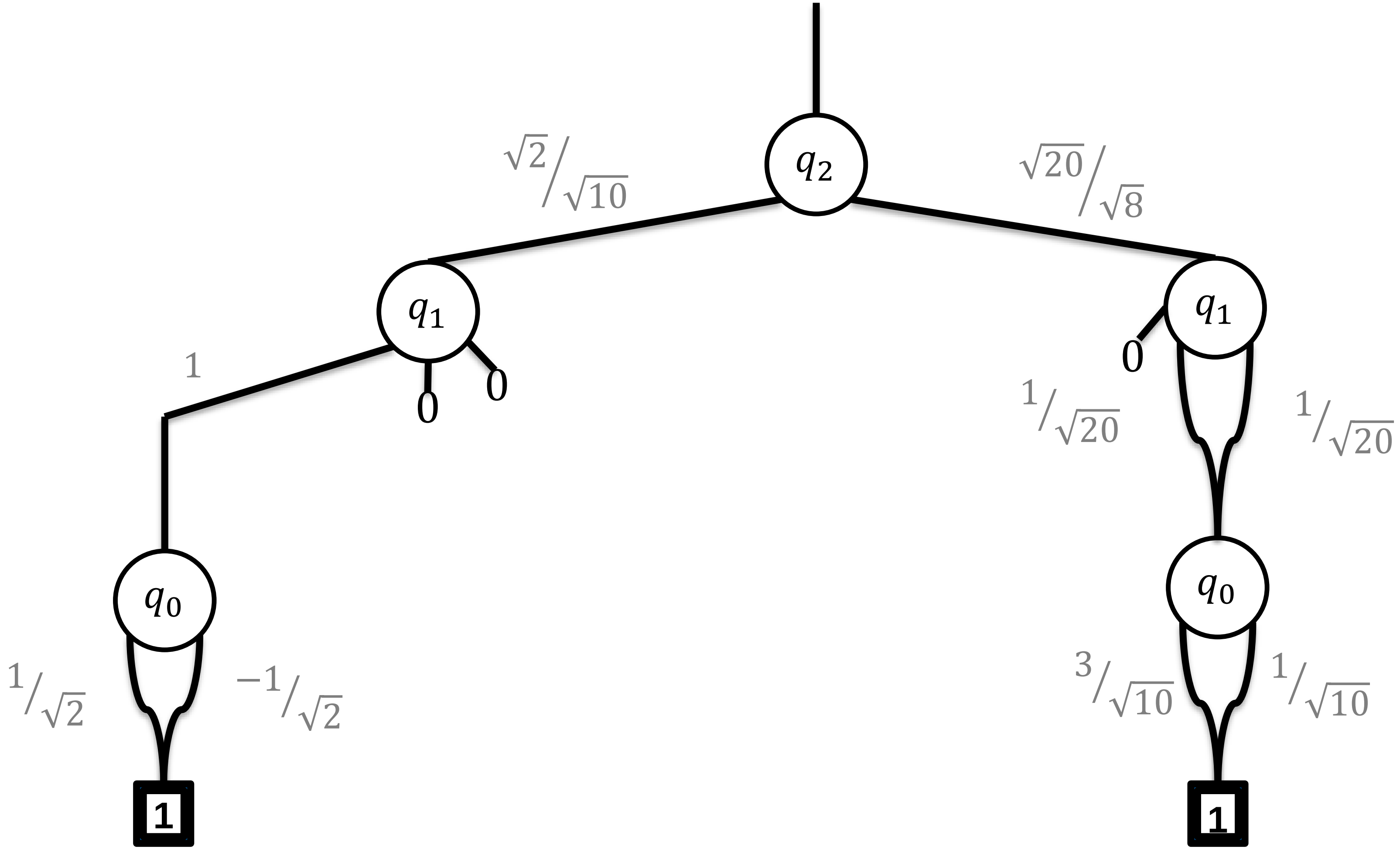


Structure + Sparsity + Redundancy



sparse **reduced**

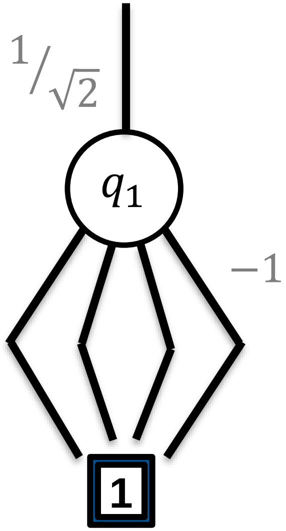
Decision Diagram



sparse **reduced** **normalized**

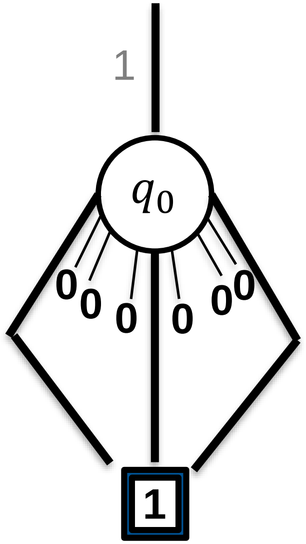
Decision Diagrams of Operations

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

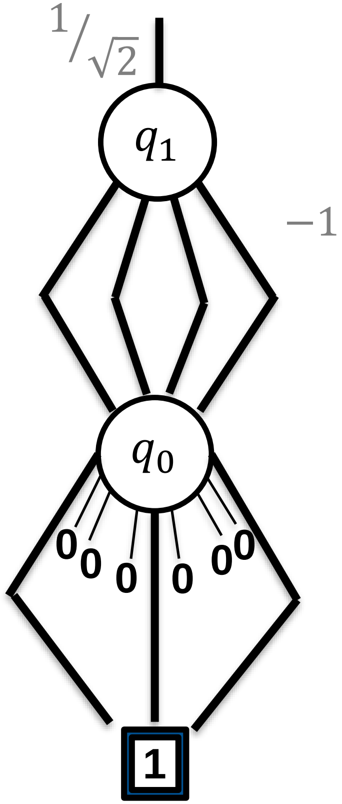


DxD Blocks

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



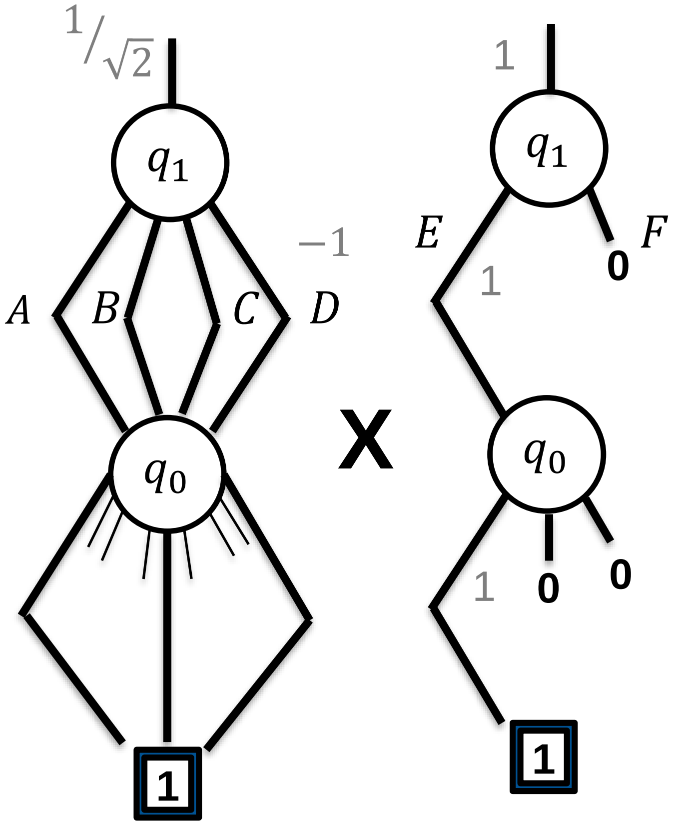
$$U = H \otimes I_3$$



Operations: Efficient Multiplication

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} E \\ F \end{bmatrix} = \begin{bmatrix} A \cdot E + B \cdot F \\ C \cdot E + D \cdot F \end{bmatrix}$$

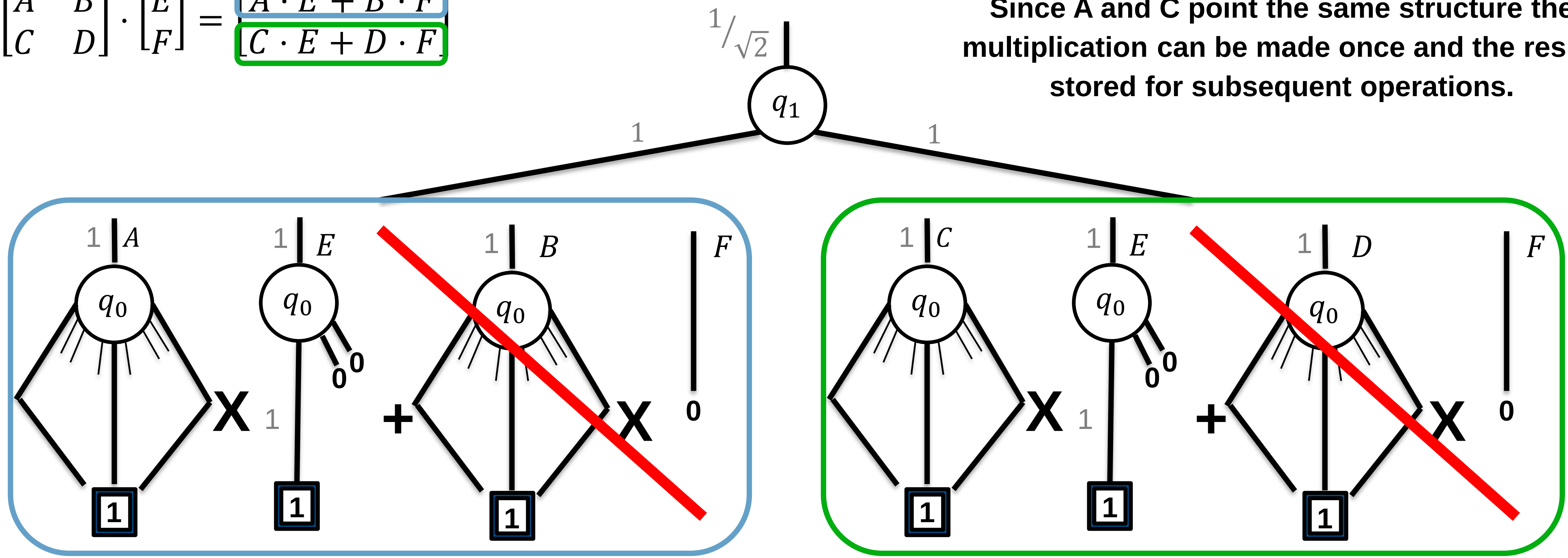
$$(H \otimes I_3) \cdot |00\rangle =$$



Operations: Efficient Multiplication

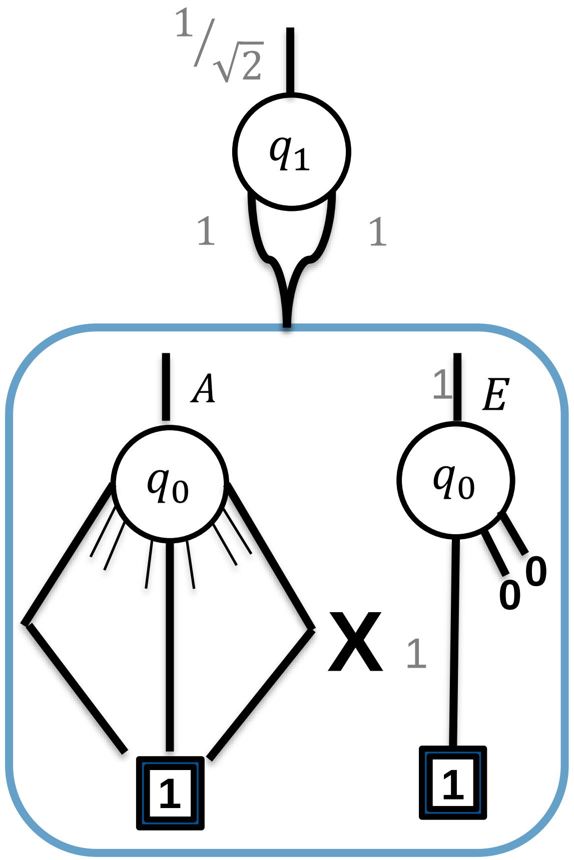
$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} E \\ F \end{bmatrix} = \begin{bmatrix} A \cdot E + B \cdot F \\ C \cdot E + D \cdot F \end{bmatrix}$$

Since A and C point the same structure the multiplication can be made once and the result stored for subsequent operations.



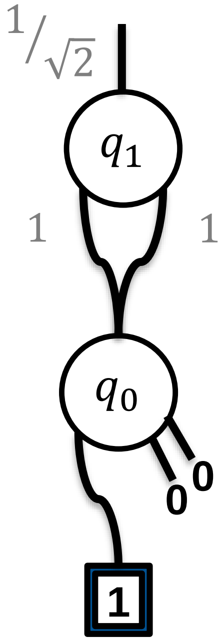
Operations: Efficient Multiplication

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} E \\ F \end{bmatrix} = \begin{bmatrix} A \cdot E + B \cdot F \\ C \cdot E + D \cdot F \end{bmatrix}$$



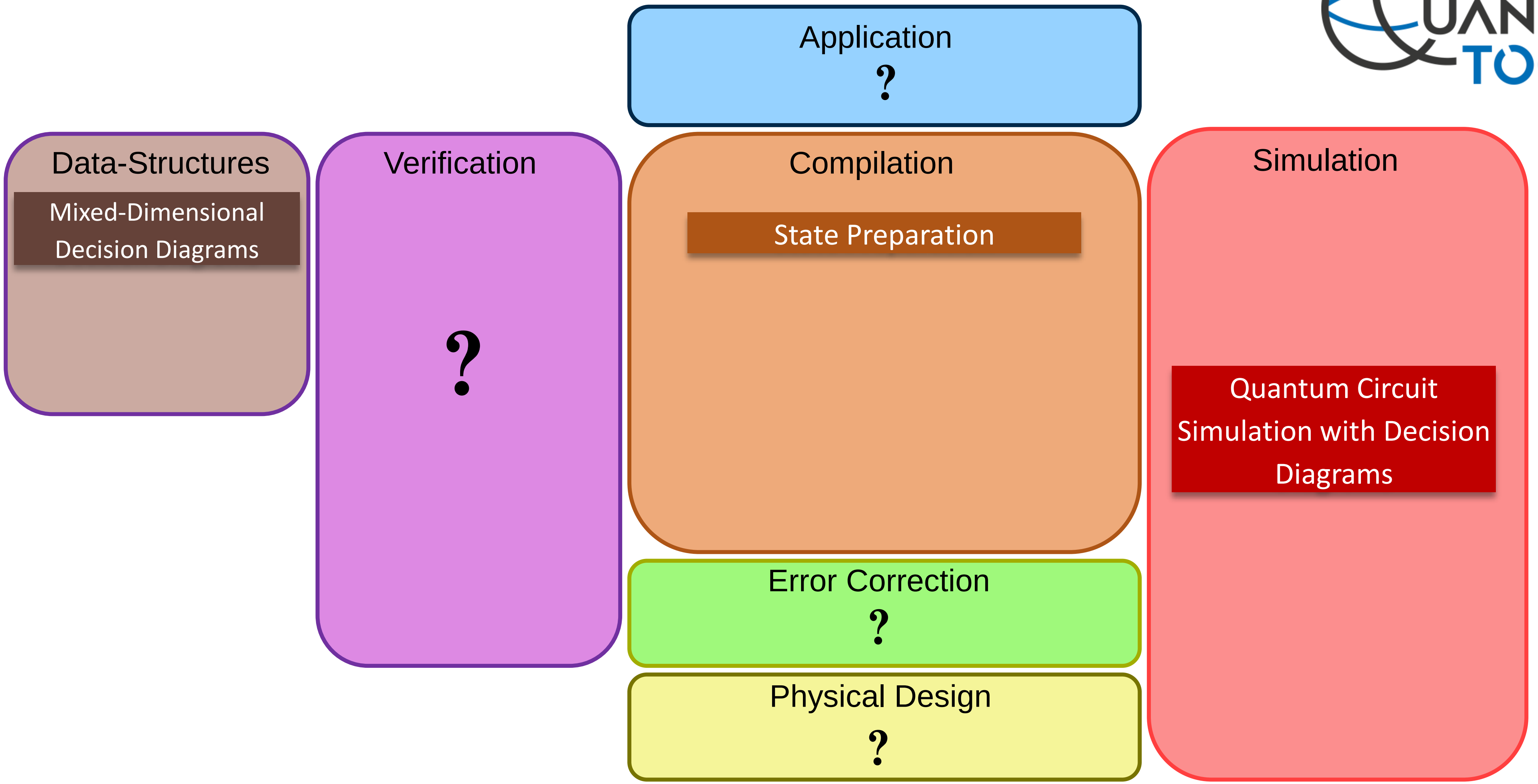
Operations: Efficient Multiplication

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} E \\ F \end{bmatrix} = \begin{bmatrix} A \cdot E + B \cdot F \\ C \cdot E + D \cdot F \end{bmatrix}$$



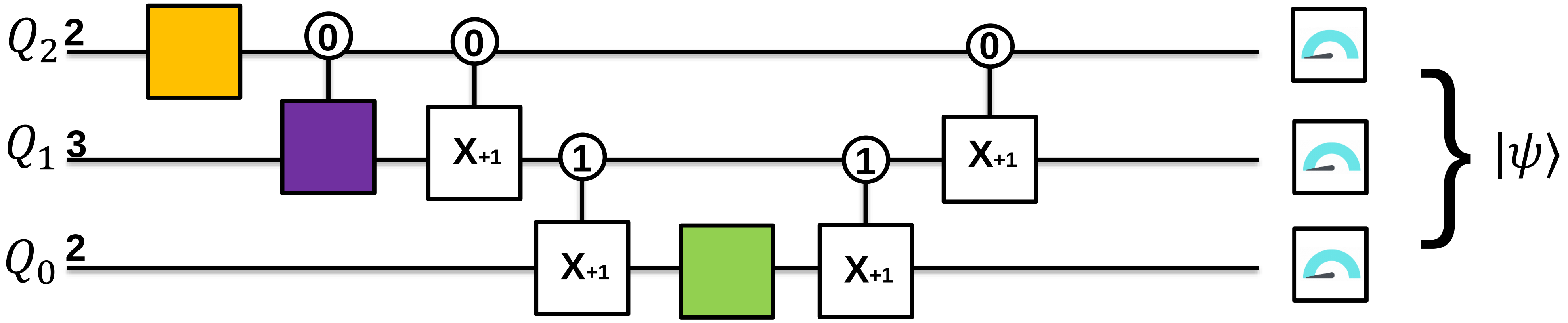
Significant reduction in the number of operations!

The *Qudit* Stack From Above



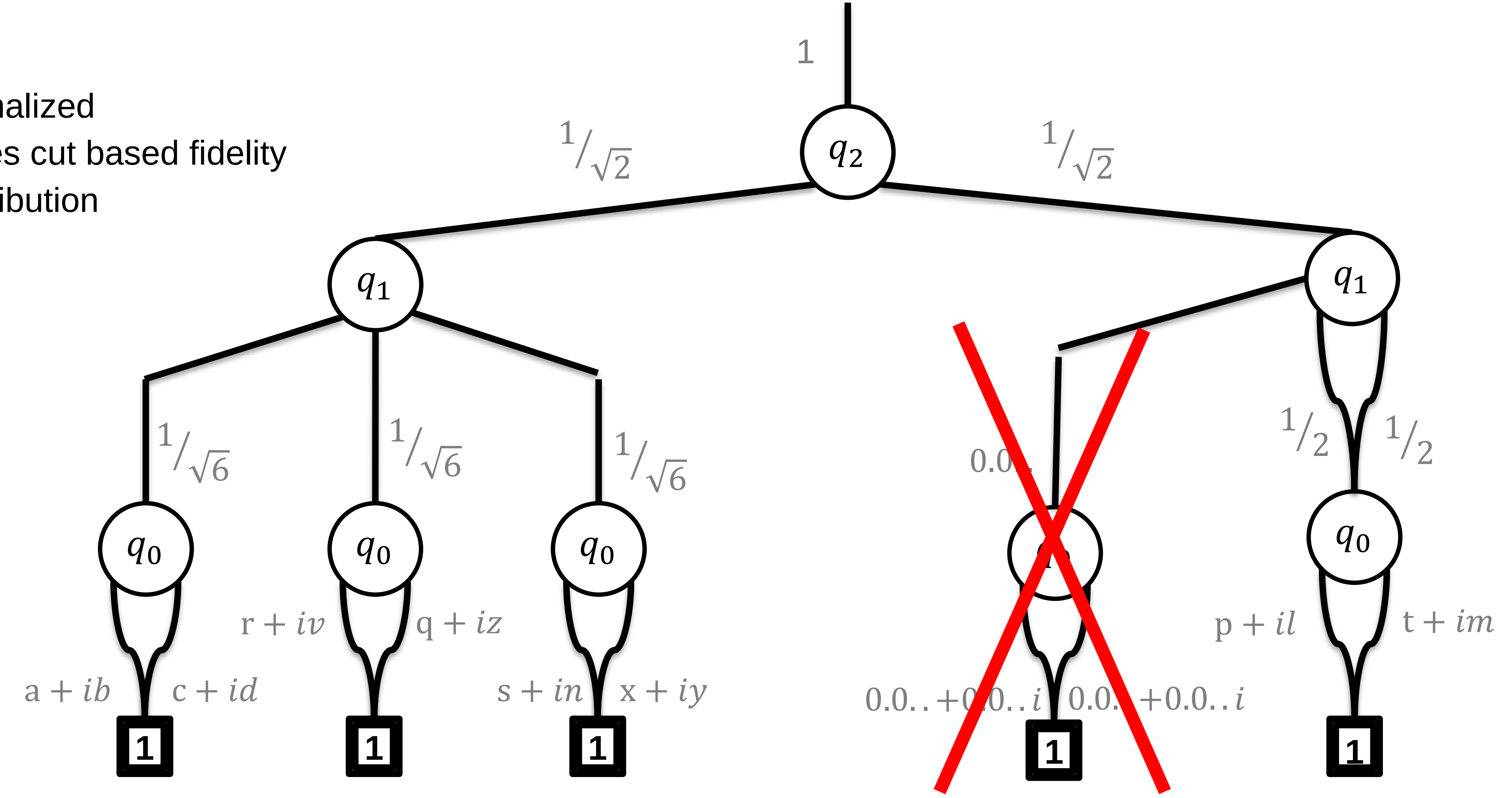
Problem: Mixed-Dimensional Quantum State Preparation

$$\psi = [\alpha_{000} \quad \alpha_{001} \quad \alpha_{010} \quad \alpha_{011} \quad \alpha_{020} \quad \alpha_{021} \quad \alpha_{100} \quad \alpha_{101} \quad \alpha_{110} \quad \alpha_{111} \quad \alpha_{120} \quad \alpha_{121}]$$



State Reduced

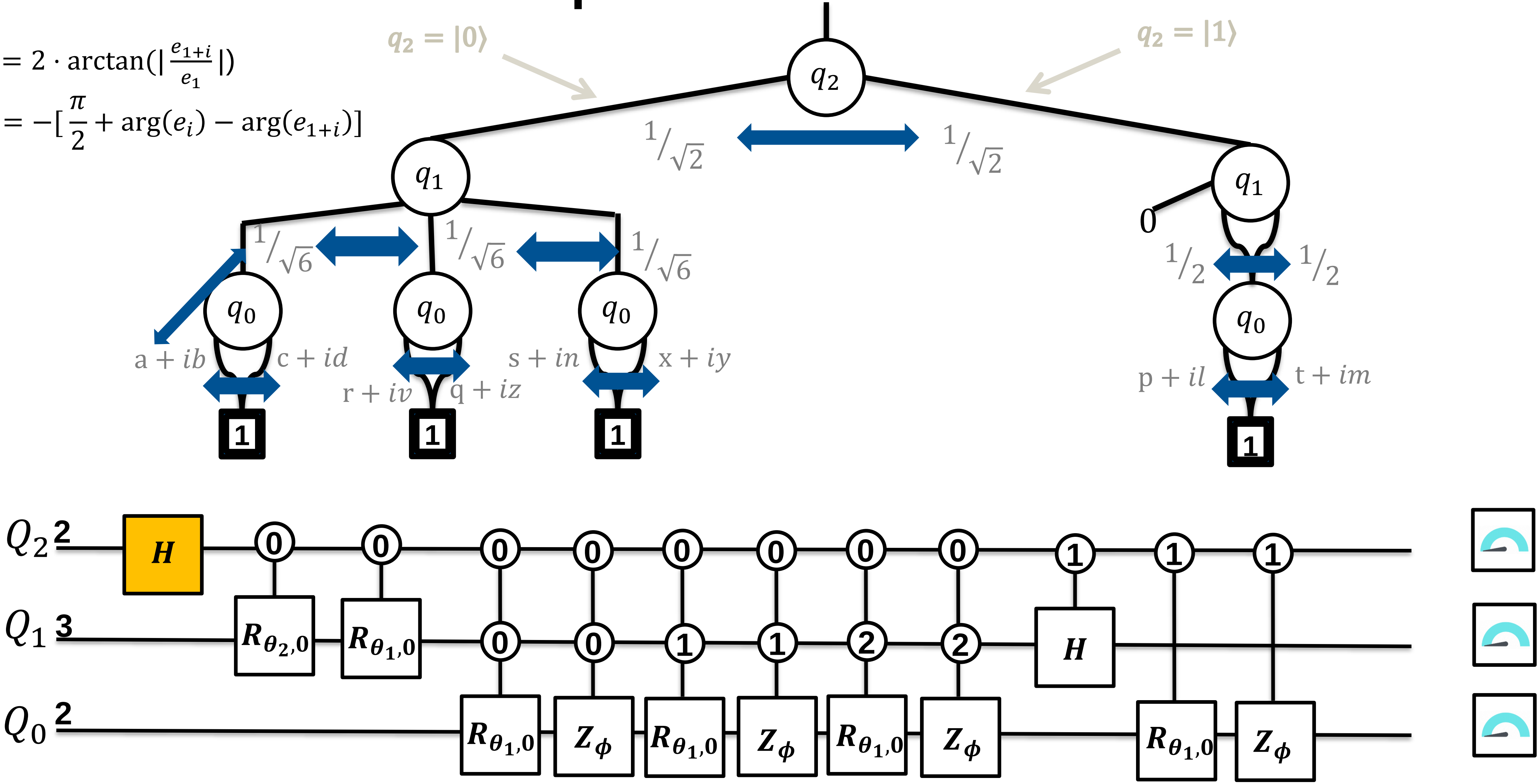
- Normalized
- Nodes cut based fidelity contribution



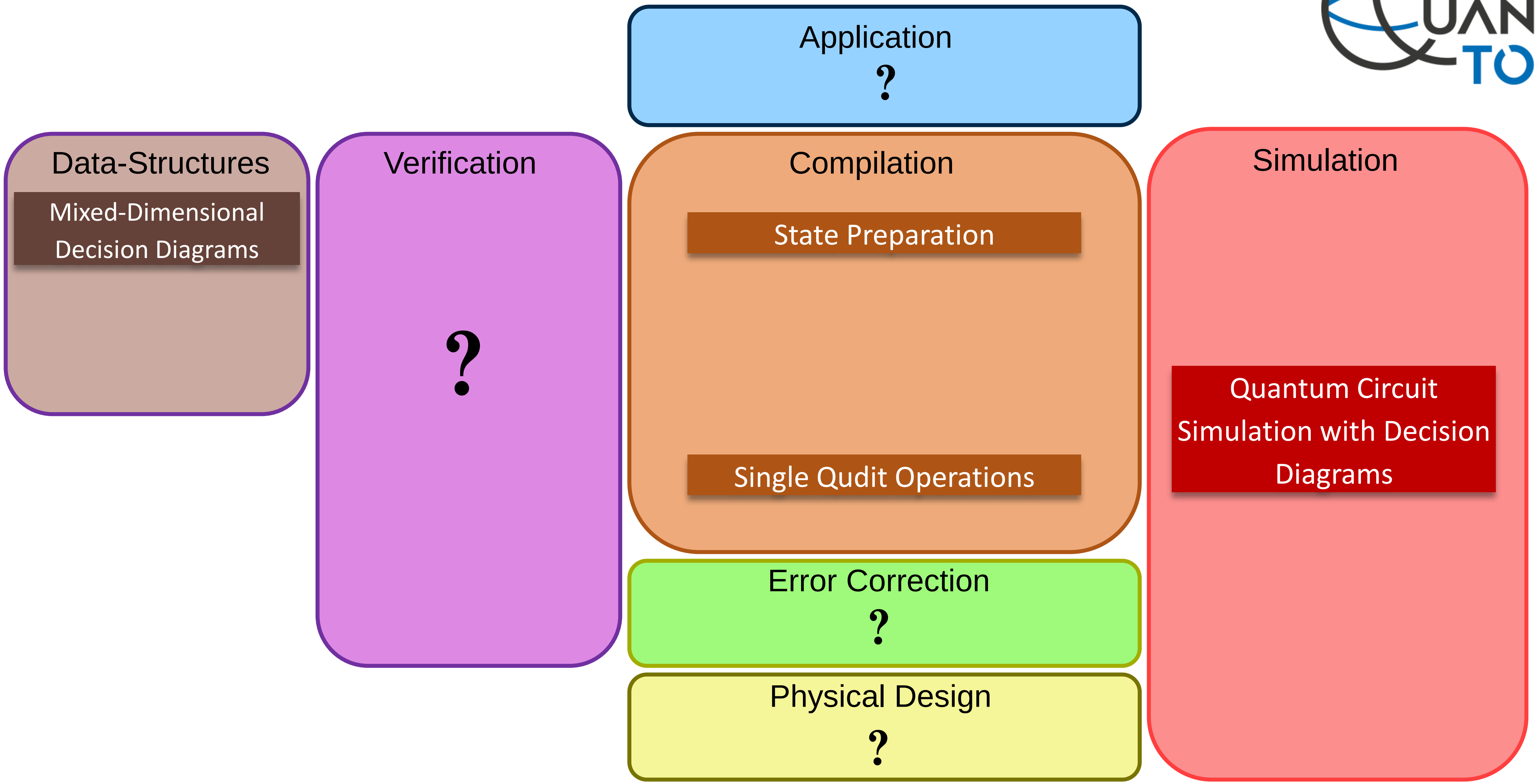
State Reduced and Optimized

$$\theta = 2 \cdot \arctan(|\frac{e_{1+i}}{e_1}|)$$

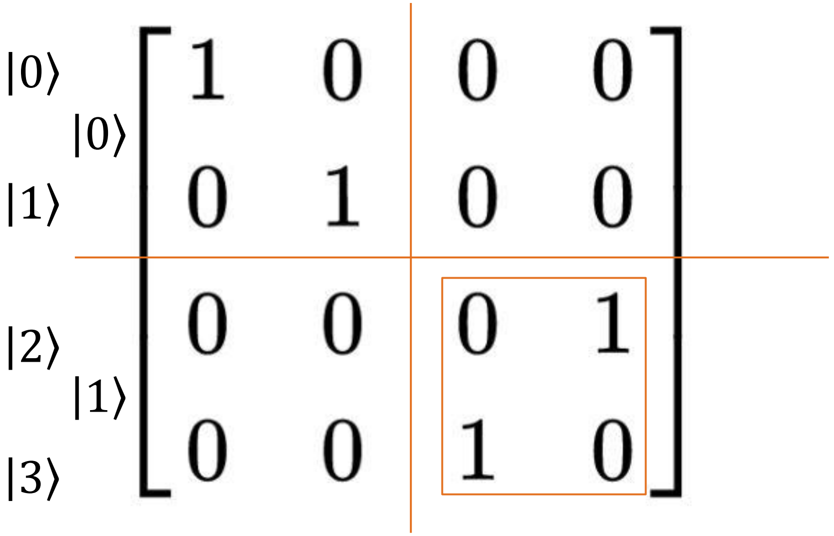
$$\varphi = -[\frac{\pi}{2} + \arg(e_i) - \arg(e_{1+i})]$$



The *Qudit* Stack From Above

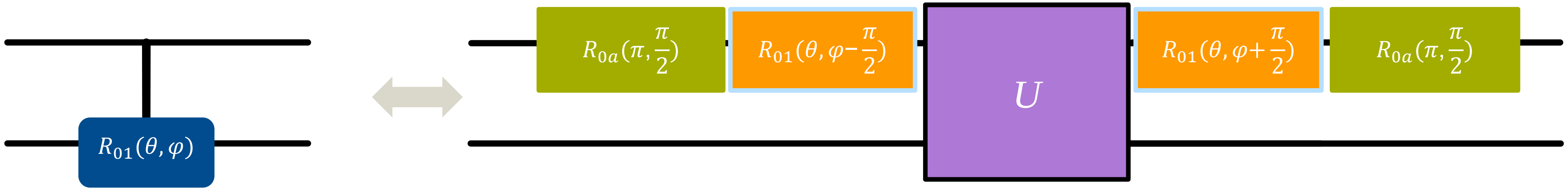


Why a Compiler?



Trading entangling operations for local ones!

- Entangling operations.
- Local operations.



Problem: Compilation of Local Operations

- Improve the error-rate of current state-of-art decomposition algorithms.

Given Unitary U find decomposition: $U = V_k \cdot V_{k-1} \cdots V_1 \cdot \Theta$

- Two-level Rotations
- Arbitrary Phases

Example: QR decomposition

$$U = V_3 \cdot V_2 \cdot V_1 \cdot \Theta_2 \cdot \Theta_1$$

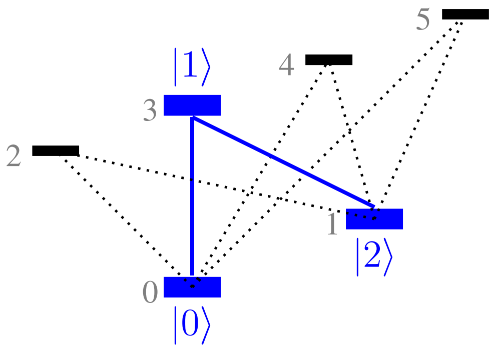
Initial Unitary:

$$\begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} & \frac{e^{\frac{2 i \pi}{3}}}{\sqrt{3}} & \frac{e^{-\frac{2 i \pi}{3}}}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} & \frac{e^{-\frac{2 i \pi}{3}}}{\sqrt{3}} & \frac{e^{\frac{2 i \pi}{3}}}{\sqrt{3}} \end{pmatrix} \rightarrow \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \sqrt{\frac{2}{3}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ 0 & -\frac{i}{\sqrt{2}} & \frac{i}{\sqrt{2}} \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & -\frac{i}{\sqrt{2}} & \frac{i}{\sqrt{2}} \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & i \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & +1 & 0 \\ 0 & 0 & i \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$V_3; L_{12}$ — $V_2; L_{01}$ — $V_1; L_{12}$ — $\Theta_2; L_1$ — $\Theta_1; L_2$

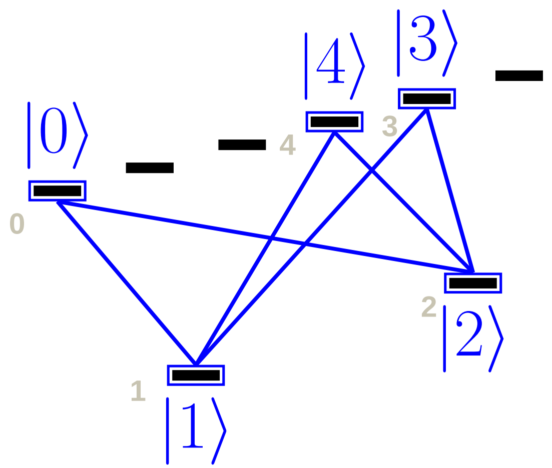
- *Can these rotations be directly and natively implemented?*
- *Is the current sequence the most cost efficient?*

Energy Level Graph

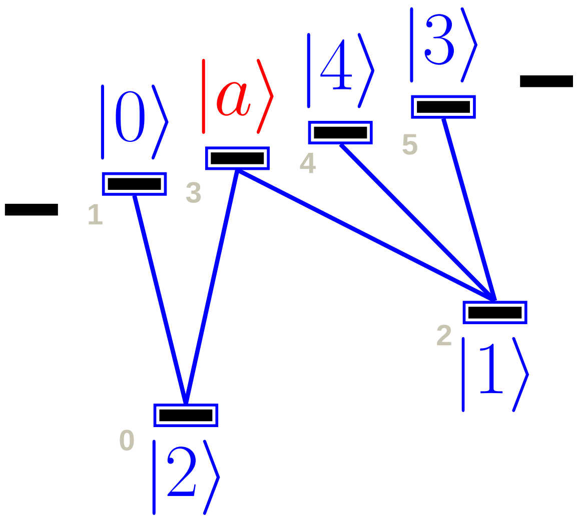


- *Qutrit Energy Level Graph*

- Physical levels connected by couplings.
- Each logic state is mapped to a physical level.



- *Ideal Energy Level Graph*



- *Realistic Energy Level Graph*

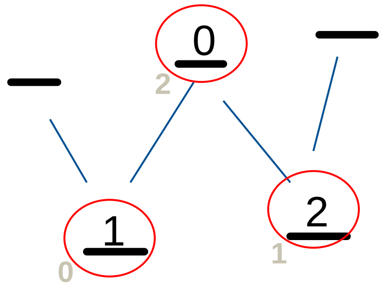
Reality is ***far*** from ideal:

- Subset of couplings.
- Subset of energy levels.
- Ancilla levels.
- Mapping between logical and physical levels is unordered.

Adaptive decomposition

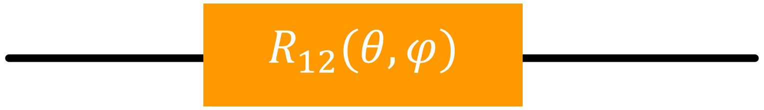
Initial Unitary:

$$\begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} & \frac{e^{\frac{2 i \pi}{3}}}{\sqrt{3}} & \frac{e^{-\frac{2 i \pi}{3}}}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} & \frac{e^{-\frac{2 i \pi}{3}}}{\sqrt{3}} & \frac{e^{\frac{2 i \pi}{3}}}{\sqrt{3}} \end{pmatrix}$$



Snapshot ☺

θ, ϕ of logical 02;



Options for logical operations:

- 1-2 too expensive
- 0-1
- 0-2

- Parameters are calculated as:

$$\theta = 2 \cdot \arctan(|\frac{U_{r2,c}}{U_{r,c}}|)$$

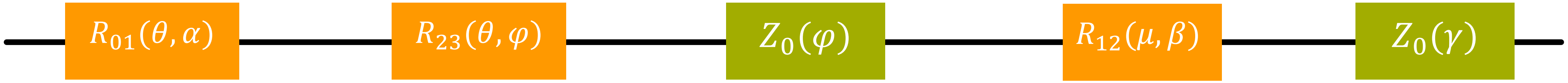
$$\varphi = -[\frac{\pi}{2} + \arg(U_{r,c}) - \arg(U_{r2,c})]$$

  - *Logical operation cost*:

    Cost of previous operations +
    Operations for routing the states +
    Physical rotation implementing the intended operation

Propagating Z rotations

- Final circuit:



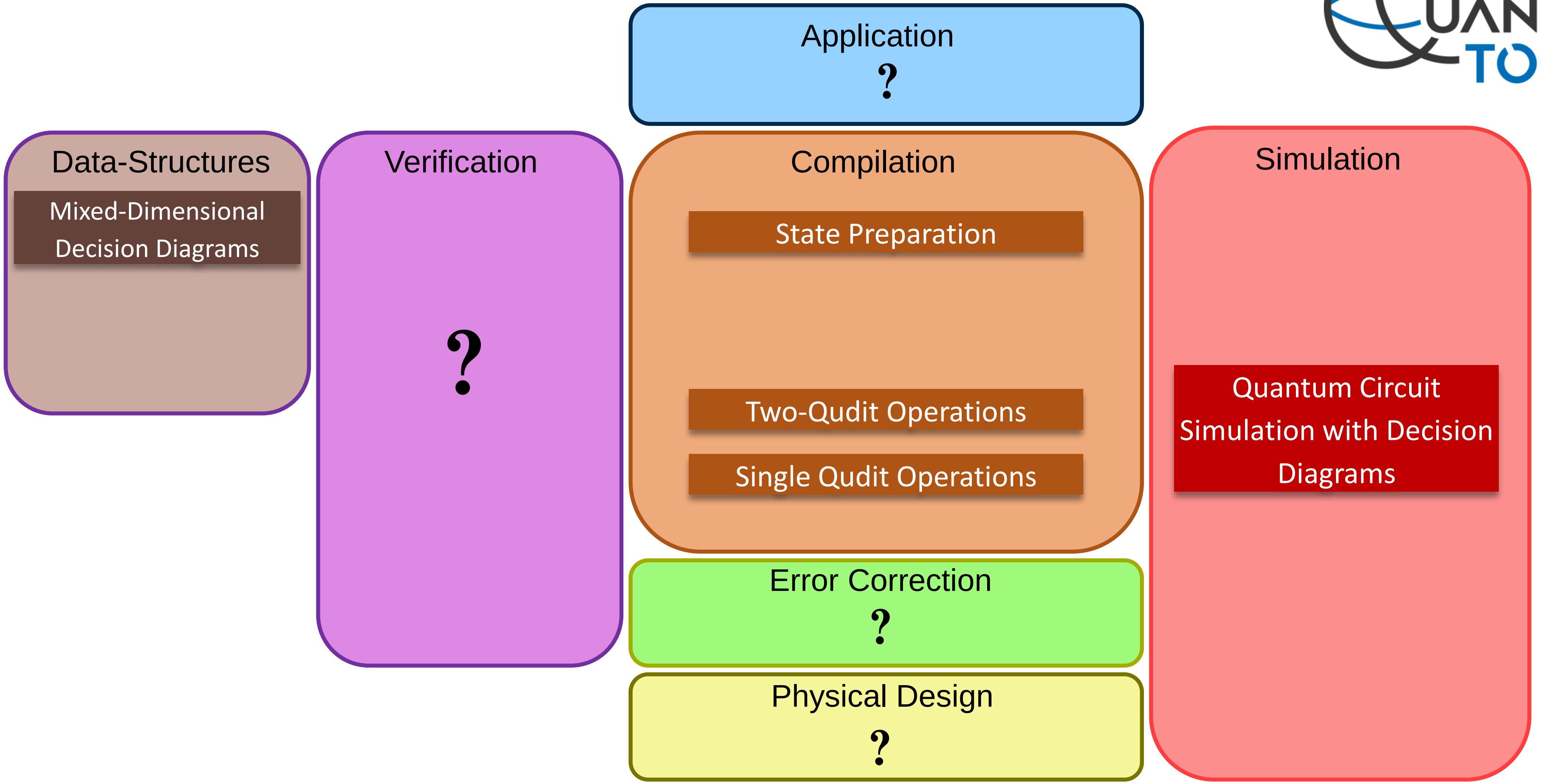
- Rule:

$$\begin{bmatrix} e^{i\phi} & 0 & 0 \\ 0 & e^{i\gamma} & 0 \\ 0 & 0 & e^{i\delta} \end{bmatrix} \cdot R_{0,1}(\theta,\alpha) = R_{0,1}(\theta,\alpha-\phi+\gamma) \cdot \begin{bmatrix} e^{i\phi} & 0 & 0 \\ 0 & e^{\gamma} & 0 \\ 0 & 0 & e^{i\delta} \end{bmatrix}$$

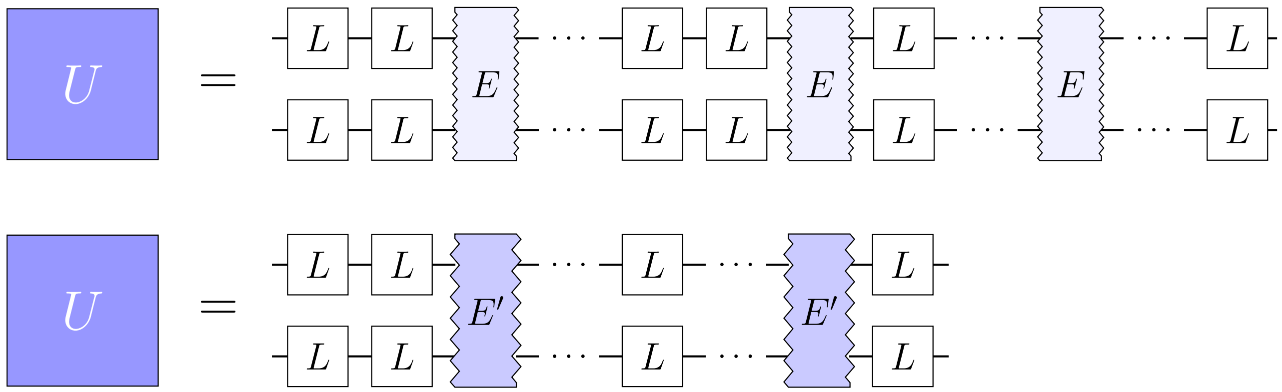
- Backward:

- Forward:

The *Qudit* Stack From Above



Problem: Compilation of Entangling Operations



- Given a unitary U representing an interaction between two qudits of dimension d
  Find a decomposition of U into arbitrary local unitaries and a pre-defined set of entangling gates
  In a way that is as close to the optimum as possible.

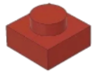
- Each decomposition takes into account the structure of the entangling gate provided by the quantum hardware and the cost of each gate
- A compilation workflow in ***2 steps***

Entanglement Structures

- Much richer entanglement structure of qu*d*its compared to qubits

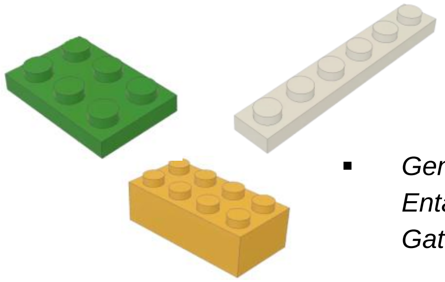
- Quantum Algorithm or Functionality

- *Local Operations*
- *Entangling Operations*



- *CNOT.*

+



- *Genuine Entangling Gates.*

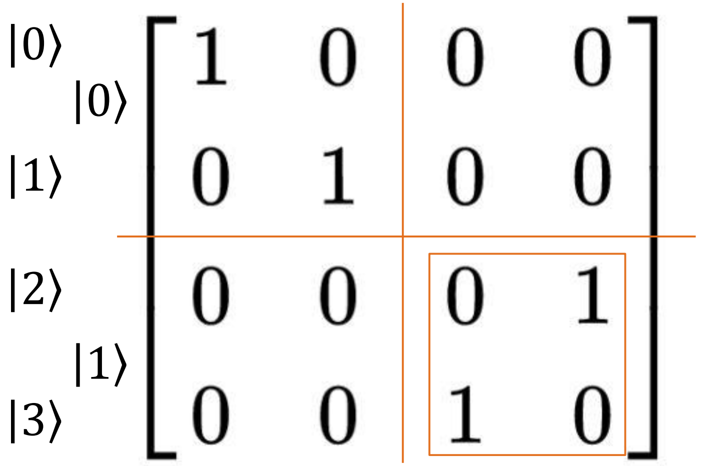
- Challenges in finding suitable gate sets native to hardware and compilation algorithms for these gate sets
- Theory and design methods are insufficient, therefore qudit compilation is still manual

- Once you are given an *unknown arbitrary two-qudit unitary* it is **not** possible to understand beforehand if it is entangling, without performing expensive computations or experiments

- How can you efficiently implement an arbitrary two-qudit unitary given the native gate set of the device?

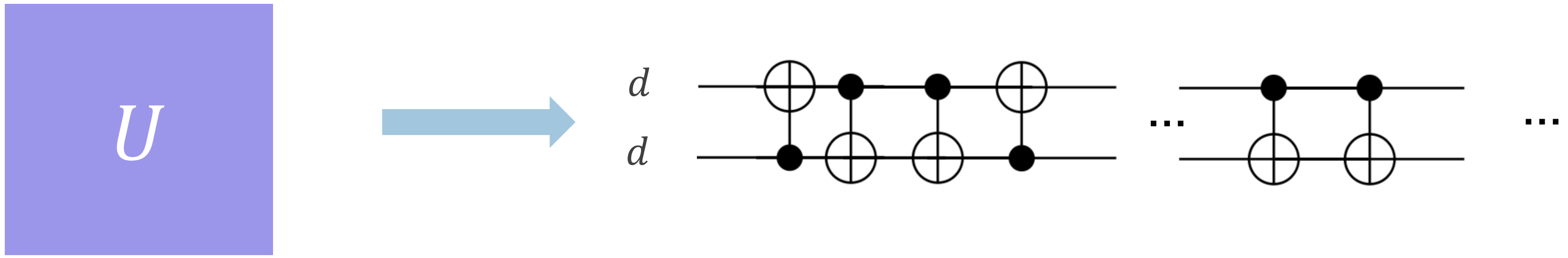
*https://rebrickable.com/mocs/MOC-110625/Simon%20Lenz/back-to-the-future-time-machine-mods-improvements/#details

Decomposition: First Step

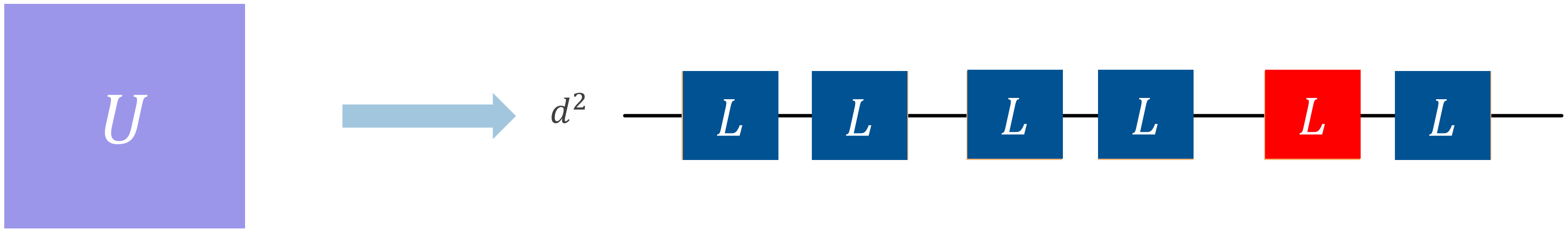


- One of the advantages of using qudits:
  Trading entangling operations for local ones

- We need to quantify the entangling interactions between the two qudits

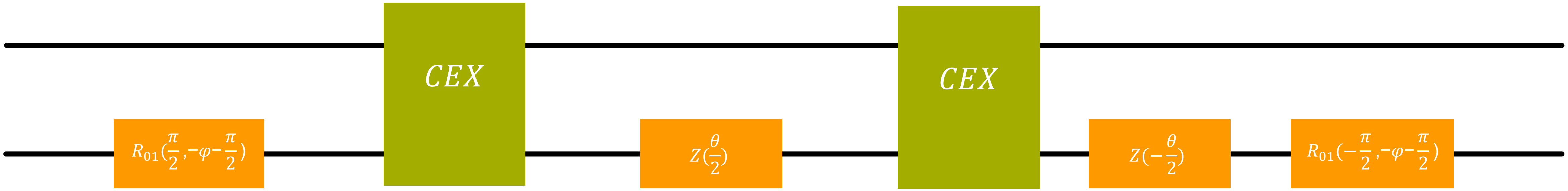


- We map the target unitary on two-qudits, to an appropriate single qudit unitary, by re-encoding

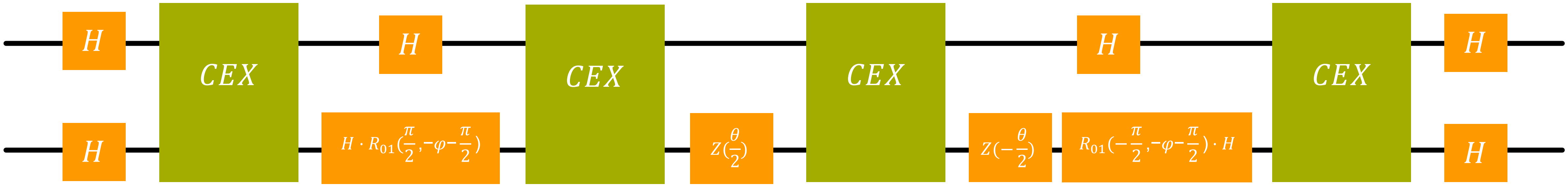


The Basic Brick: CEX Gate

- The decomposition of the chosen ***cRot*** in function of θ and ϕ

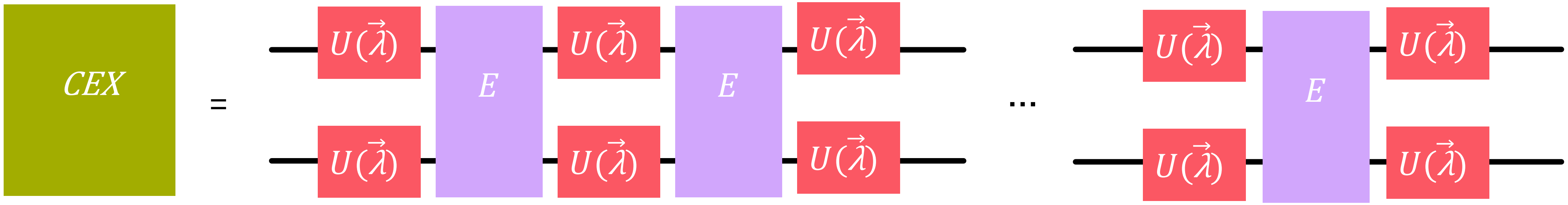


- The decomposition of the chosen ***pSwap*** in function of θ and ϕ



Make your own CEX: Second Step

- Offline
- Parametrized circuit:
  - *Objective function* → $Fidelity(A,B) = \frac{1}{d^2} Tr\langle A^\dagger , B \rangle$
  - *Ansatz Binary Search*

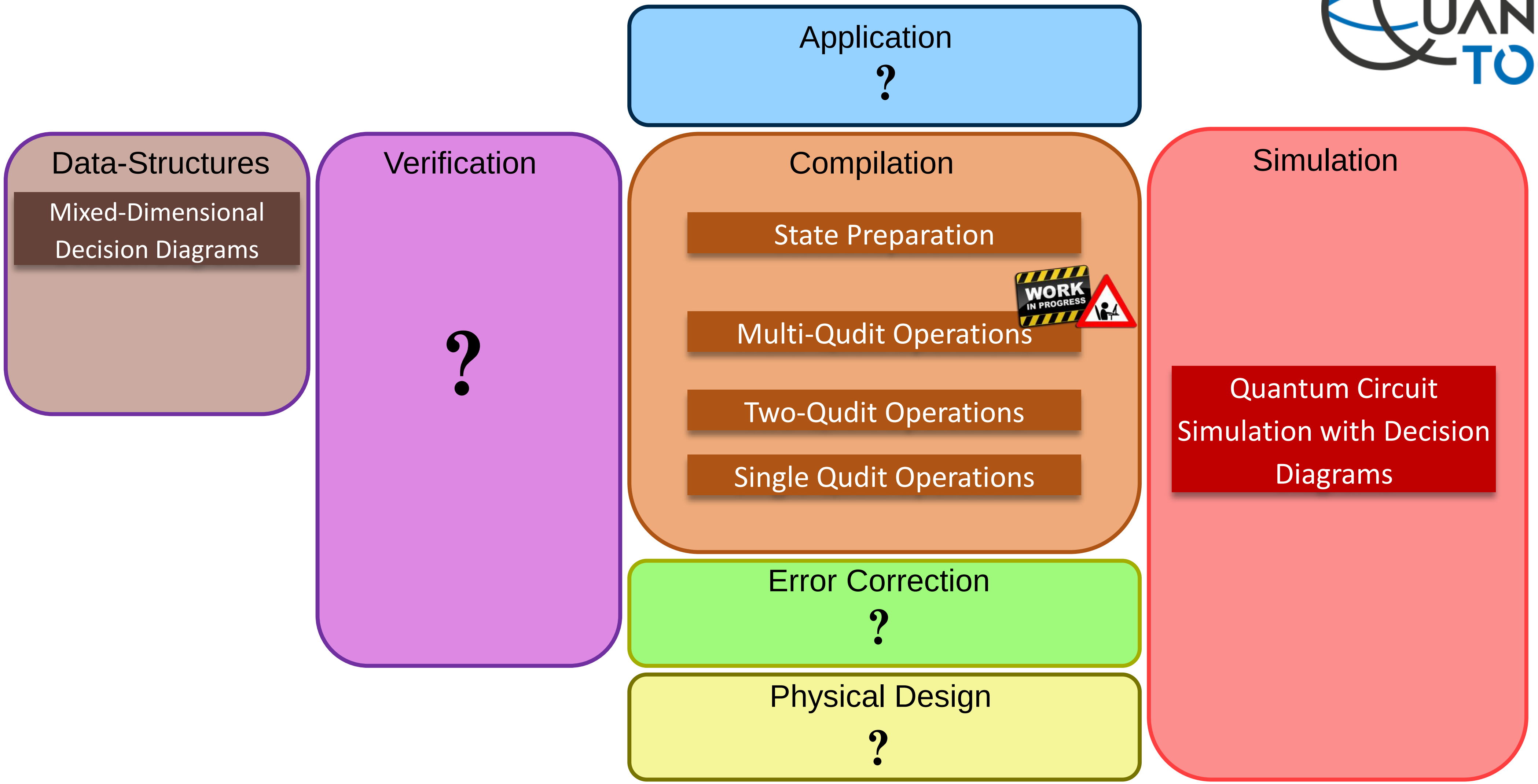


- A genuine qudit entangling gate: *LS gate* $\quad \mathrm{LS}(\theta) = e^{-i\theta \cdot \sum_{i=0}^{d-1} |ii\rangle\langle ii|}$
- Two-level entangling gate: *MS gate* $\quad \mathrm{MS}(\theta) = e^{-i\frac{\theta}{4}\cdot(I\otimes I + \sigma_{x_{01}} \otimes \sigma_{x_{01}})}$
- Custom user input

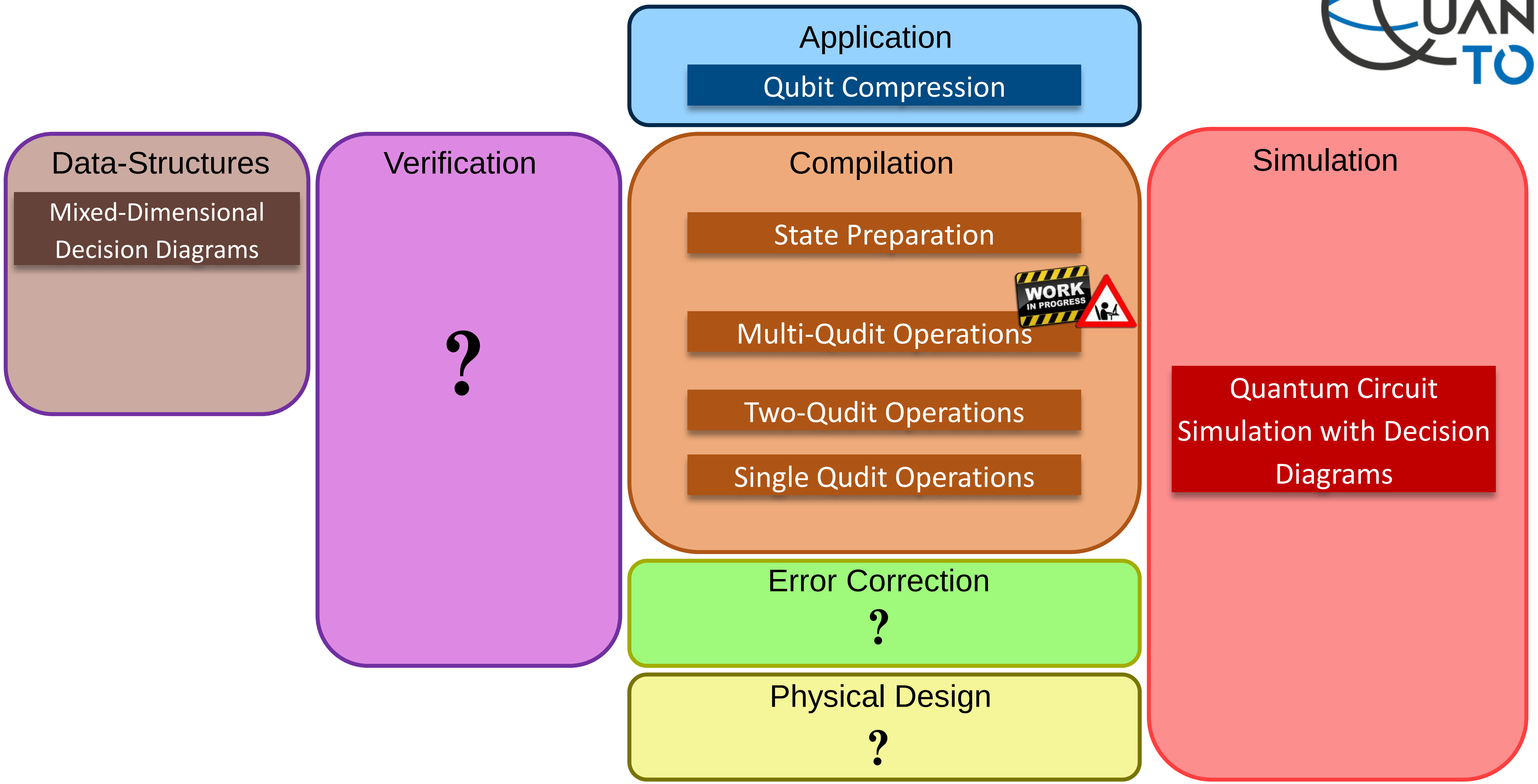
$$U(\vec{\lambda}) = \left[\prod_{m=0}^{d-2} \left(\prod_{n=m+1}^{d-1} \exp(iZ_{m,n}\lambda_{n,m}) \exp(iY_{m,n}\lambda_{m,n})\right)\right] \cdot \left[\prod_{l=1}^{d-1} \exp(iZ_{l,d}\lambda_{l,l})\right]$$

Expressible representation constructed from $d^2 - 1$ parameters

The *Qudit* Stack From Above



The *Qudit* Stack From Above



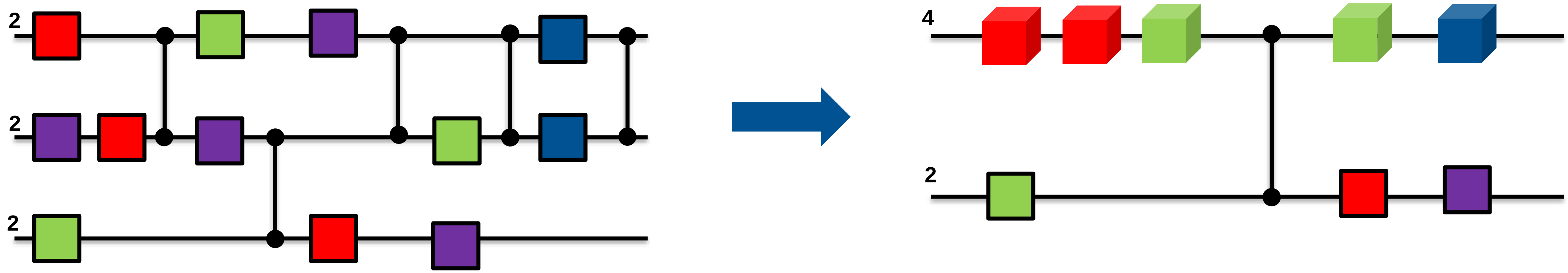
Problem

- The goal is to improve the quality of computation by reducing the number of operations in a sequence, with a focus on non-local operations.

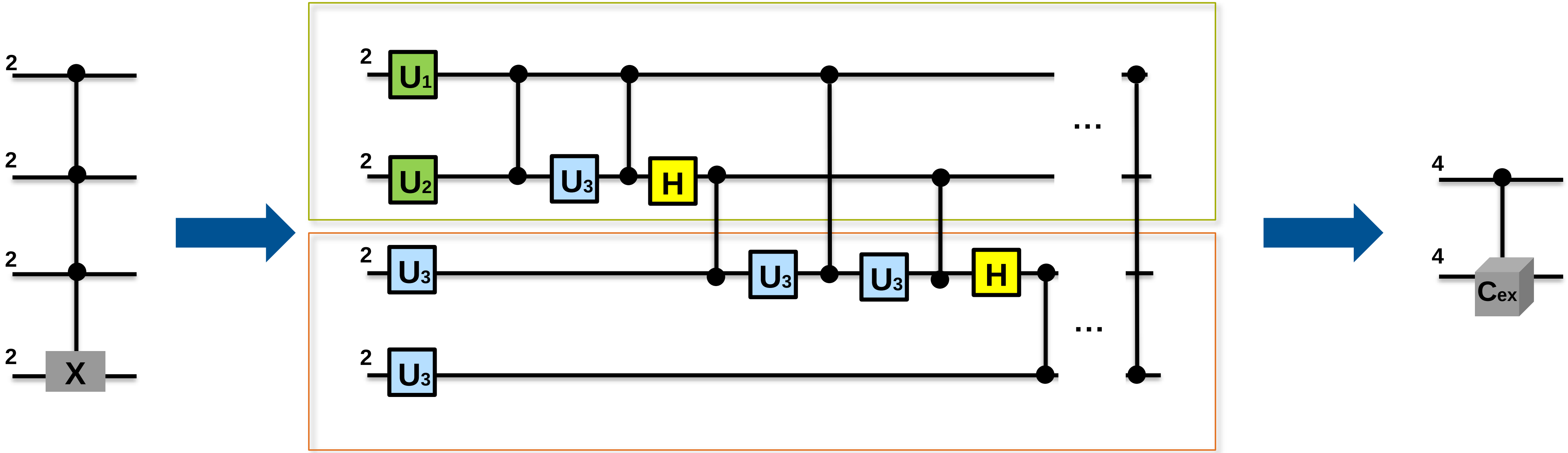
It can be achieved by:

- Rewriting the sequence in a more noise-efficient one.
- Reducing the number of qubits, or qudits.

The process is called ***circuit compression***.

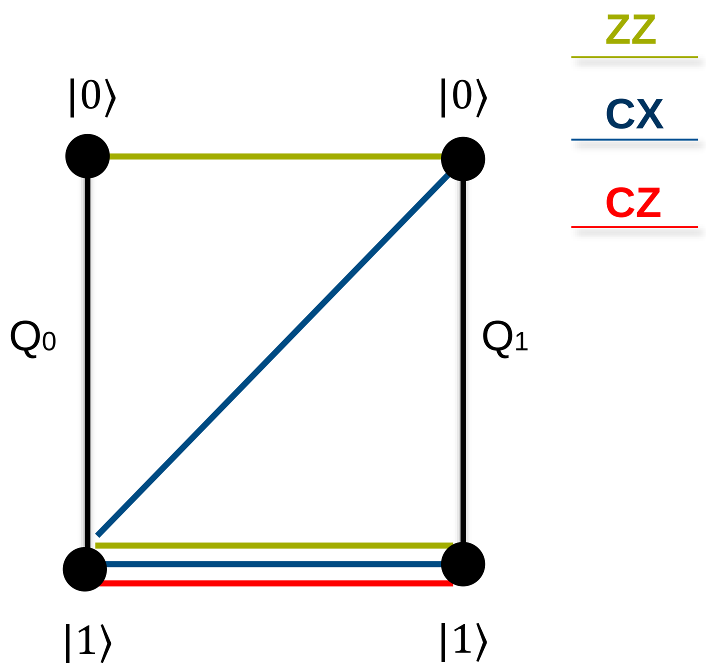


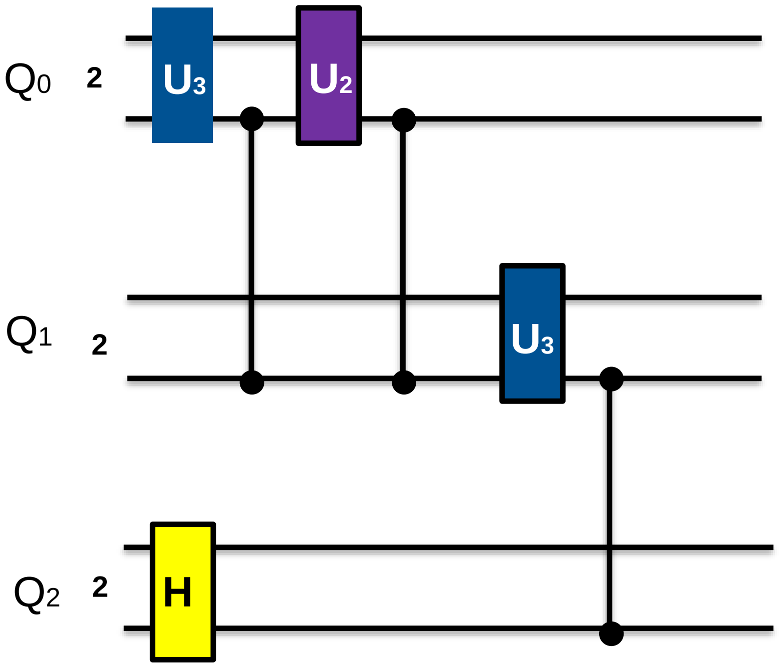
Circuit Compression



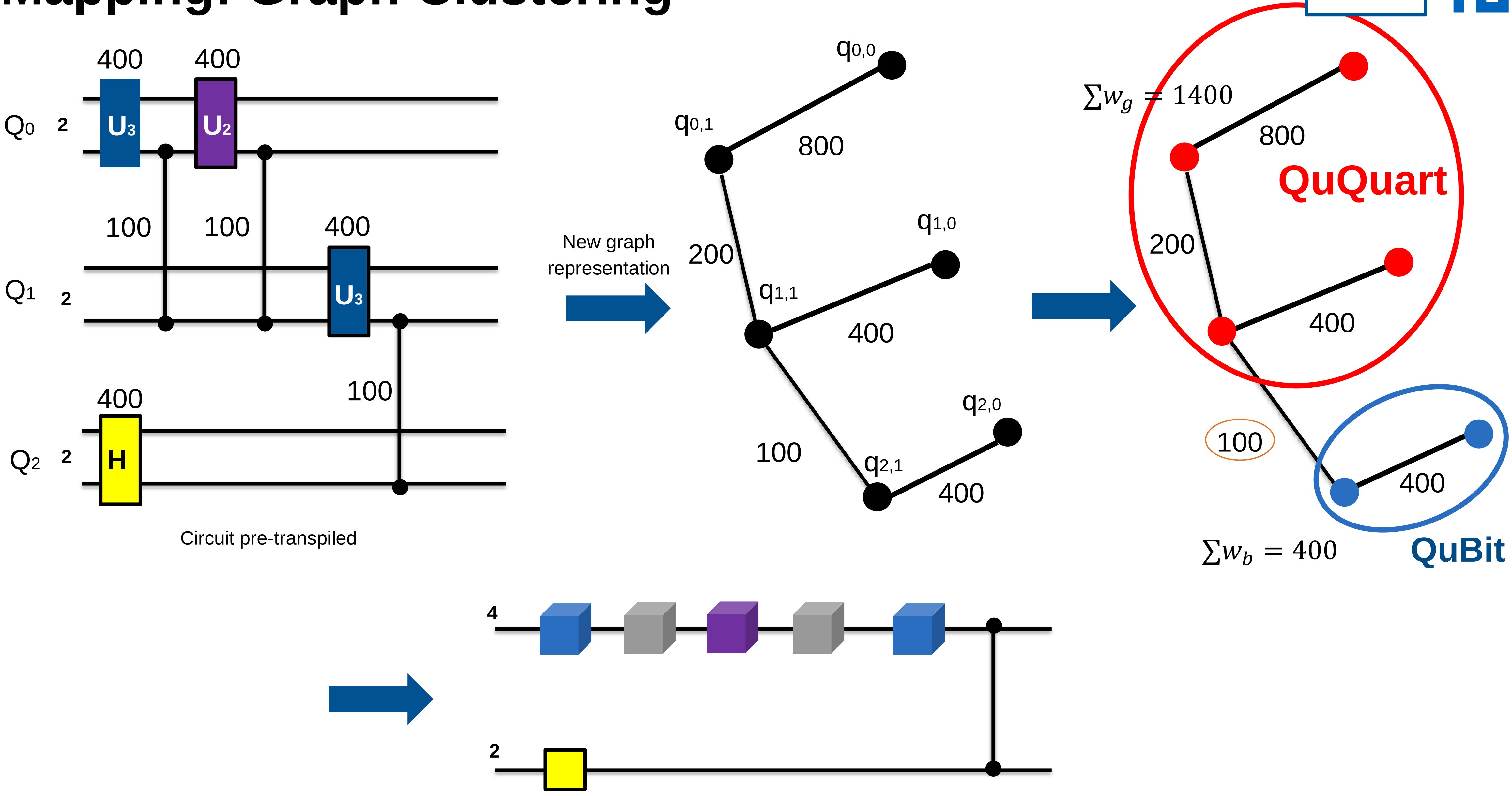
Noise as a Metric

- Encoding, or mapping, the qubit logic into a Hilbert space generated by the combination of suitable higher-dimensional carriers.
- Qubits that are frequently entangled get mapped on the same qudit.

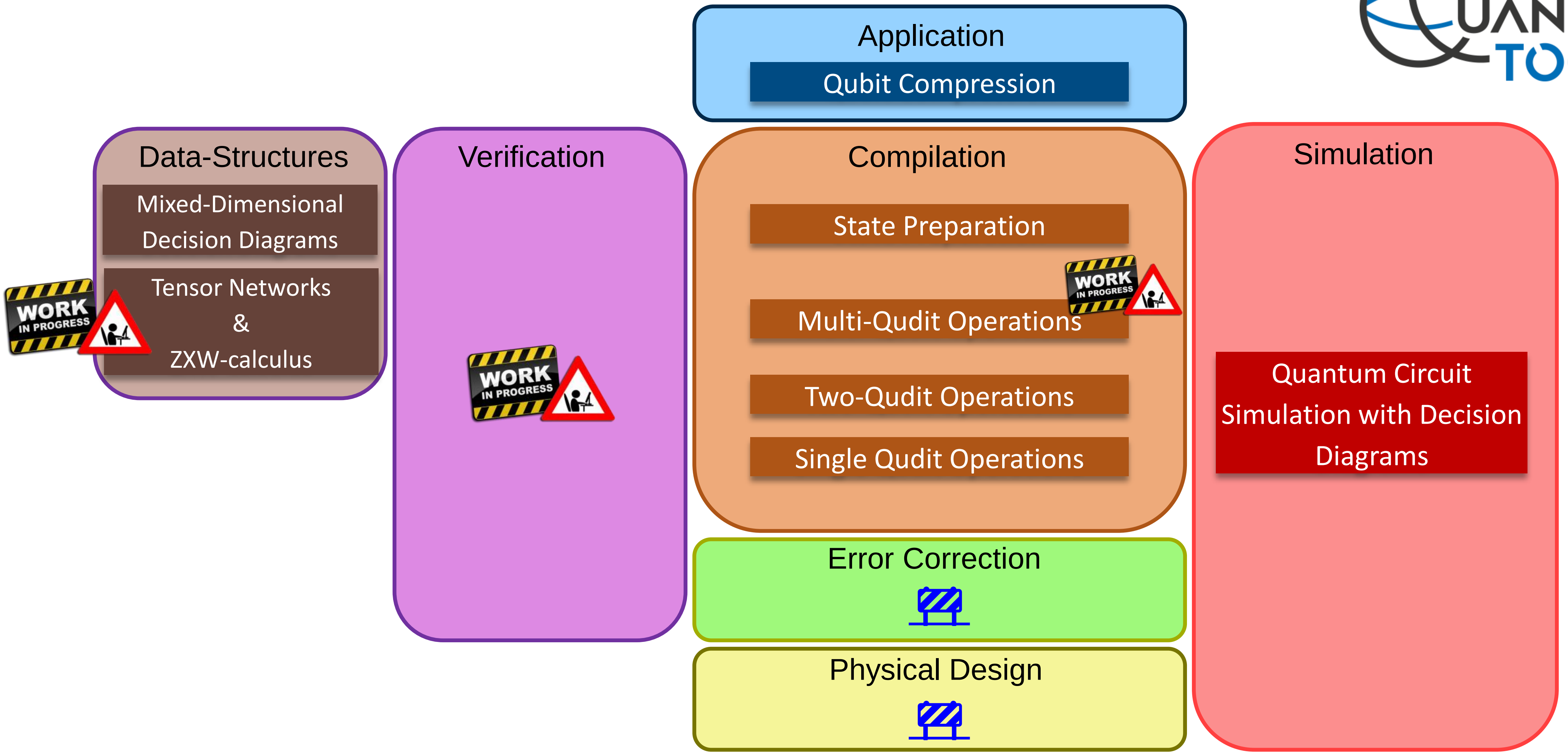




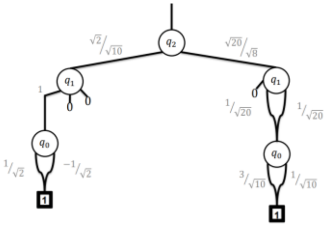
Mapping: Graph Clustering

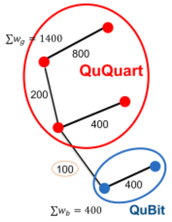


The *Qudit* Stack From Above



The *Qudit* Stack From Above: Conclusions





Application

Qubit Compression

Open QASM 2.0 extended

Data-Structures

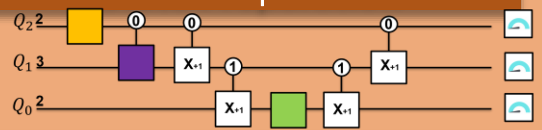
Mixed-Dimensional Decision Diagrams

Tensor Networks & ZXW-calculus

Verification

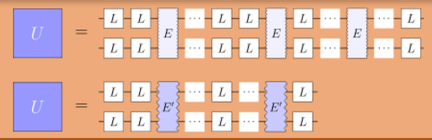
Compilation

State Preparation



Multi-Qudit Operations

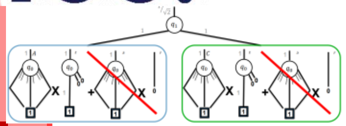
Two-Qudit Operations



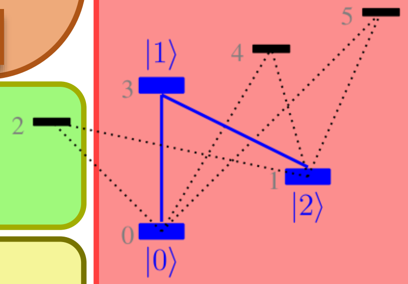
Single Qudit Operations

Error Correction

Physical Design



Quantum Circuit Simulation with Decision Diagrams



- **Available as Open Source**

MUNICH QUANTUM TOOLKIT

- **On GitHub under**
- **https://github.com/cda-tum**

Acknowledgments

Funded by the European Union under Horizon Europe Programme - Grant Agreement 101080086 — NeQST. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.