CSC

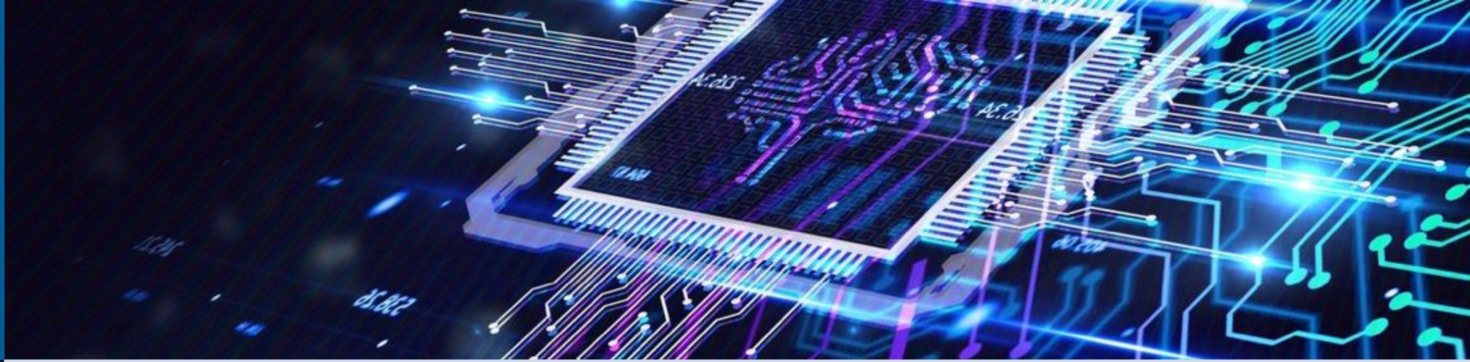
ICT Solutions for Brilliant Minds

QCut, Quantum circuit-knitting on FiQCI

Joonas Nivala, Junior Application Specialist

CSC

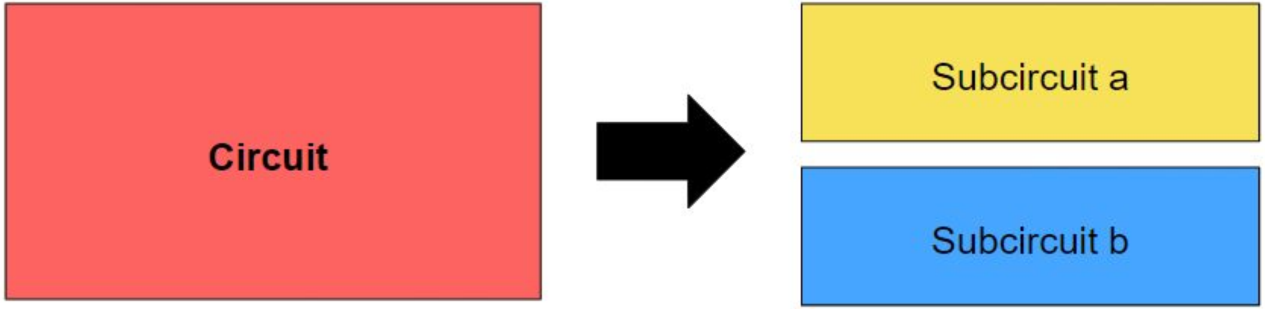
ICT Solutions for Brilliant Minds



Circuit Knitting - Basics

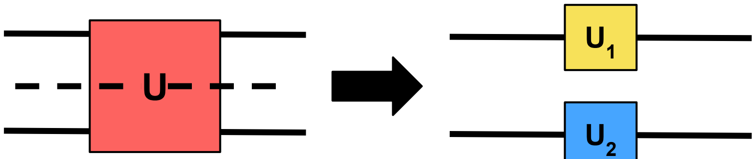
What and why?

- Split a large circuit into multiple smaller pieces
- Simulate large QPUs on smaller ones
- “Increase” number of available qubits
- Parallel Quantum Computing

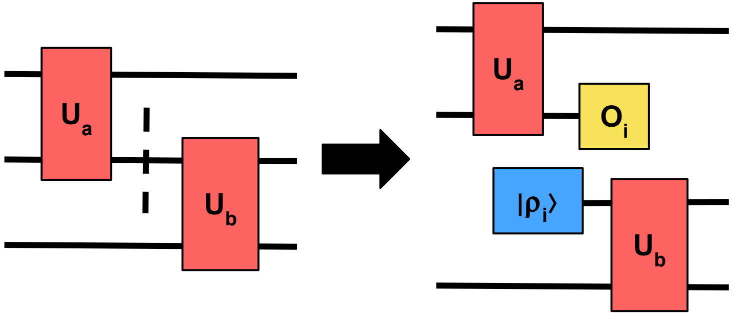


How?

Gate cut



Wire cut



Wire cut

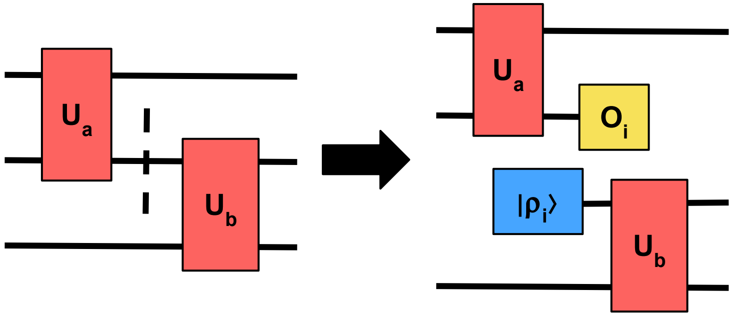
- Quasi probability distribution (QPD) simulation
  - Sample operations from a distribution

Steps:

1. Generate experiment circuits
   a. sample qpd
2. Execute circuits
3. Post-process to obtain expectation values

$$Id(\bullet) = \sum_{i=1}^{8} c_i Tr[O_i(\bullet)]\rho_i$$

O_i	ρ_i	c_i
$O_1 = I$	$\rho_1 = \|0\rangle\langle 0\|$	$c_1 = +1/2$
$O_2 = I$	$\rho_2 = \|1\rangle\langle 1\|$	$c_2 = +1/2$
$O_3 = X$	$\rho_3 = \|+\rangle\langle +\|$	$c_3 = +1/2$
$O_4 = X$	$\rho_4 = \|-\rangle\langle -\|$	$c_4 = -1/2$
$O_5 = Y$	$\rho_5 = \|+i\rangle\langle +i\|$	$c_5 = +1/2$
$O_6 = Y$	$\rho_6 = \|-i\rangle\langle -i\|$	$c_6 = -1/2$
$O_7 = Z$	$\rho_7 = \|0\rangle\langle 0\|$	$c_7 = +1/2$
$O_8 = Z$	$\rho_8 = \|1\rangle\langle 1\|$	$c_8 = -1/2$

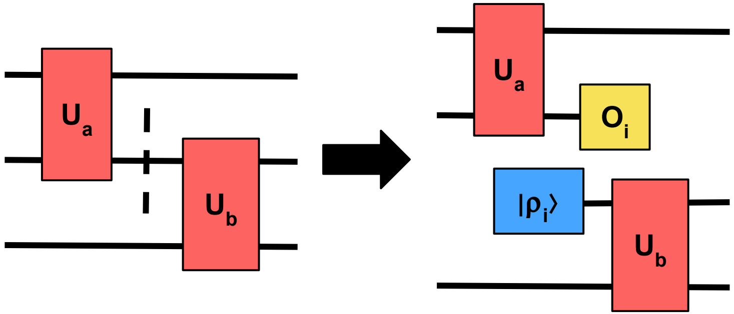


Wire cut - step 1

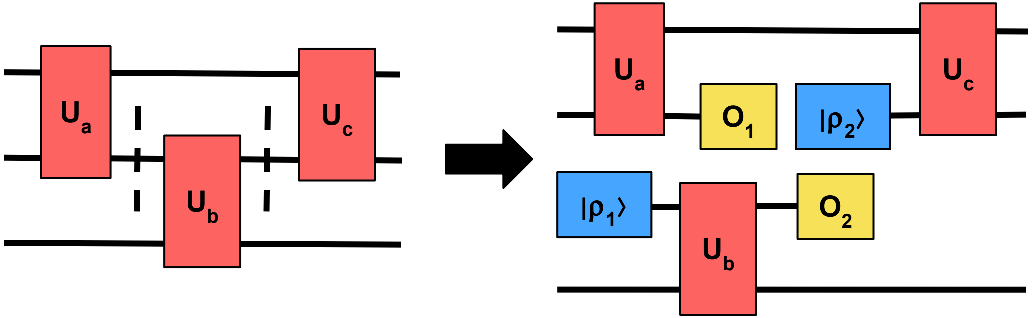
Generate experiment circuits:

- Single cut:
  - Each line in QPD gives a circuit pair
  - 8 pairs so 16 total circuits
- Multiple cuts
  - Generate all QPD combinations of length n
  - 8 groups of k so $k \cdot 8^n$ total circuits

O_i	ρ_i	c_i
$O_1 = I$	$\rho_1 = \|0\rangle\langle 0\|$	$c_1 = +1/2$
$O_2 = I$	$\rho_2 = \|1\rangle\langle 1\|$	$c_2 = +1/2$
$O_3 = X$	$\rho_3 = \|+\rangle\langle +\|$	$c_3 = +1/2$
$O_4 = X$	$\rho_4 = \|-\rangle\langle -\|$	$c_4 = -1/2$
$O_5 = Y$	$\rho_5 = \|+i\rangle\langle +i\|$	$c_5 = +1/2$
$O_6 = Y$	$\rho_6 = \|-i\rangle\langle -i\|$	$c_6 = -1/2$
$O_7 = Z$	$\rho_7 = \|0\rangle\langle 0\|$	$c_7 = +1/2$
$O_8 = Z$	$\rho_8 = \|1\rangle\langle 1\|$	$c_8 = -1/2$

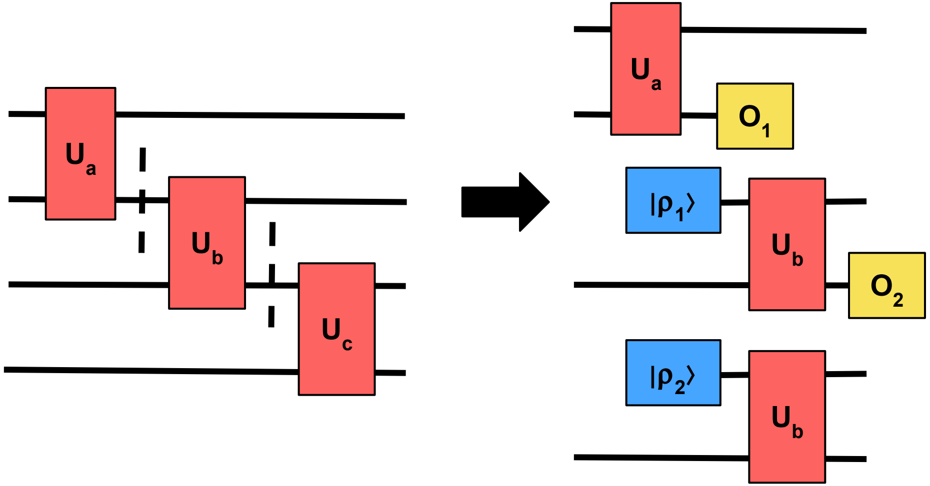


Wire cut - step 1 - example



O_i	ρ_i	c_i
$O_1 = I$	$\rho_1 = \|0\rangle\langle 0\|$	$c_1 = +1/2$
$O_2 = I$	$\rho_2 = \|1\rangle\langle 1\|$	$c_2 = +1/2$
$O_3 = X$	$\rho_3 = \|+\rangle\langle +\|$	$c_3 = +1/2$
$O_4 = X$	$\rho_4 = \|-\rangle\langle -\|$	$c_4 = -1/2$
$O_5 = Y$	$\rho_5 = \|+i\rangle\langle +i\|$	$c_5 = +1/2$
$O_6 = Y$	$\rho_6 = \|-i\rangle\langle -i\|$	$c_6 = -1/2$
$O_7 = Z$	$\rho_7 = \|0\rangle\langle 0\|$	$c_7 = +1/2$
$O_8 = Z$	$\rho_8 = \|1\rangle\langle 1\|$	$c_8 = -1/2$

Wire cut - step 1 - example



O_i	ρ_i	c_i
$O_1 = I$	$\rho_1 = \|0\rangle\langle 0\|$	$c_1 = +1/2$
$O_2 = I$	$\rho_2 = \|1\rangle\langle 1\|$	$c_2 = +1/2$
$O_3 = X$	$\rho_3 = \|+\rangle\langle +\|$	$c_3 = +1/2$
$O_4 = X$	$\rho_4 = \|-\rangle\langle -\|$	$c_4 = -1/2$
$O_5 = Y$	$\rho_5 = \|+i\rangle\langle +i\|$	$c_5 = +1/2$
$O_6 = Y$	$\rho_6 = \|-i\rangle\langle -i\|$	$c_6 = -1/2$
$O_7 = Z$	$\rho_7 = \|0\rangle\langle 0\|$	$c_7 = +1/2$
$O_8 = Z$	$\rho_8 = \|1\rangle\langle 1\|$	$c_8 = -1/2$

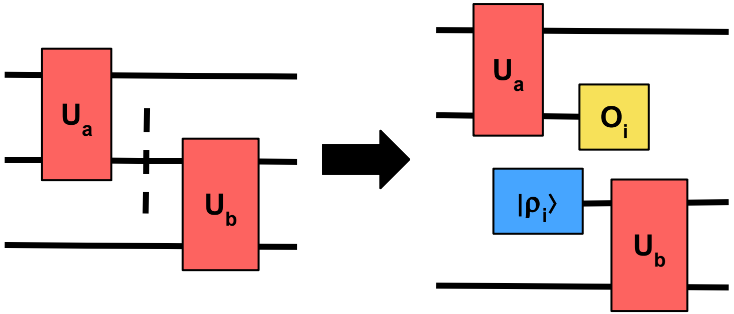
Wire cut - step 2

Execute all circuits

- Independent so order doesn't matter
- Sufficient samples needed:
  - $\gamma^{2n} \cdot \frac{1}{\epsilon^2}$, where

    $$\gamma = \sum_i |c_i|$$
- γ^{2n} is called the overhead
- for wire cuts γ = 4

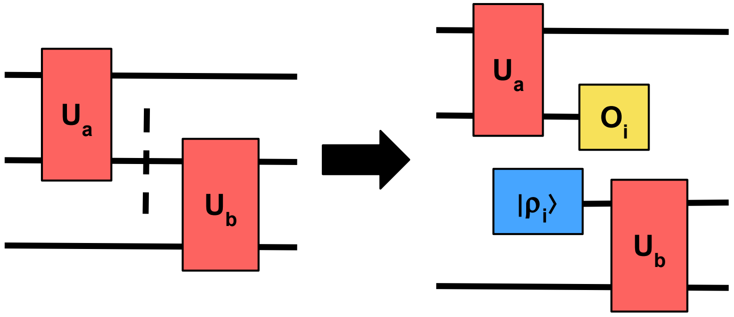
O_i	ρ_i	c_i
$O_1 = I$	$\rho_1 = \|0\rangle\langle 0\|$	$c_1 = +1/2$
$O_2 = I$	$\rho_2 = \|1\rangle\langle 1\|$	$c_2 = +1/2$
$O_3 = X$	$\rho_3 = \|+\rangle\langle +\|$	$c_3 = +1/2$
$O_4 = X$	$\rho_4 = \|-\rangle\langle -\|$	$c_4 = -1/2$
$O_5 = Y$	$\rho_5 = \|+i\rangle\langle +i\|$	$c_5 = +1/2$
$O_6 = Y$	$\rho_6 = \|-i\rangle\langle -i\|$	$c_6 = -1/2$
$O_7 = Z$	$\rho_7 = \|0\rangle\langle 0\|$	$c_7 = +1/2$
$O_8 = Z$	$\rho_8 = \|1\rangle\langle 1\|$	$c_8 = -1/2$



Wire cut - step 3

- Expectation values can be reconstructed with post-processing
  - apply $f:\{0,1\}^N \rightarrow [-1,1]^N$ for each result
  - apply $sgn(c_i)sf(\boldsymbol{y}))$
  - take mean of results

O_i	ρ_i	c_i
$O_1 = I$	$\rho_1 = \|0\rangle\langle 0\|$	$c_1 = +1/2$
$O_2 = I$	$\rho_2 = \|1\rangle\langle 1\|$	$c_2 = +1/2$
$O_3 = X$	$\rho_3 = \|+\rangle\langle +\|$	$c_3 = +1/2$
$O_4 = X$	$\rho_4 = \|-\rangle\langle -\|$	$c_4 = -1/2$
$O_5 = Y$	$\rho_5 = \|+i\rangle\langle +i\|$	$c_5 = +1/2$
$O_6 = Y$	$\rho_6 = \|-i\rangle\langle -i\|$	$c_6 = -1/2$
$O_7 = Z$	$\rho_7 = \|0\rangle\langle 0\|$	$c_7 = +1/2$
$O_8 = Z$	$\rho_8 = \|1\rangle\langle 1\|$	$c_8 = -1/2$



Wire cut - drawbacks

Number of circuits:

- Grows as: $k \cdot 8^n$
  - k = pieces circuit is cut into
  - n = number of cuts

Number of samples

- Grows as: $\gamma^{2n} \cdot \frac{1}{\epsilon^2}$
  - γ = gamma-factor
  - n = number of cuts
  - ϵ = desired error in approximation

Only returns expectation values

Wire cut - drawbacks - solutions

Number of circuits:

- Grows as: $k \cdot 8^n$
  - k = pieces circuit is cut into
  - n = number of cuts

Number of samples

- Grows as: $\gamma^{2n} \cdot \frac{1}{\epsilon^2}$
  - γ = gamma-factor
  - n = number of cuts
  - ϵ = desired error in approximation

Only returns expectation values

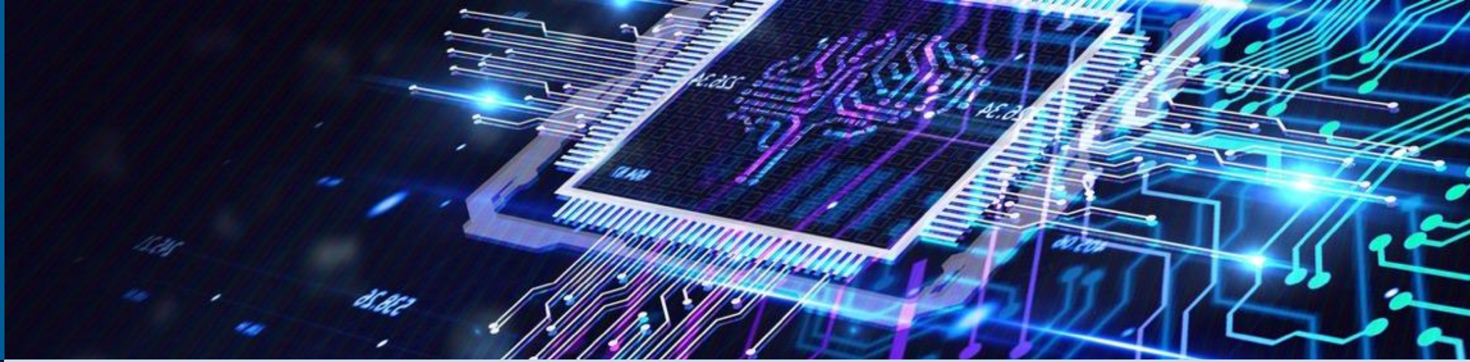
- Only few cuts possible
- Can be combated by using multiple QPUs

- Sample size can be inflated in post-processing
  - Need enough samples so that distribution of results visible

- Still useful for many applications

CSC

ICT Solutions for Brilliant Minds



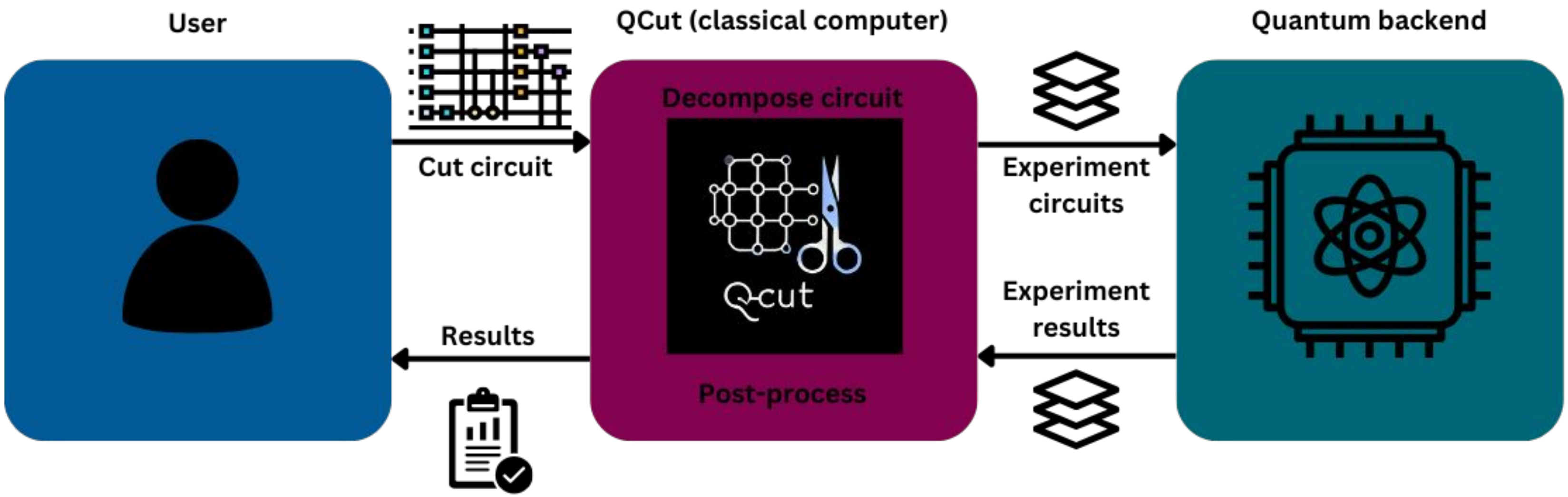
QCut

QCut

- Python package for wire cutting
- No reset gates
- No mid-circuit measurements
    - Runs on FiQCI

```
pip install QCut
```

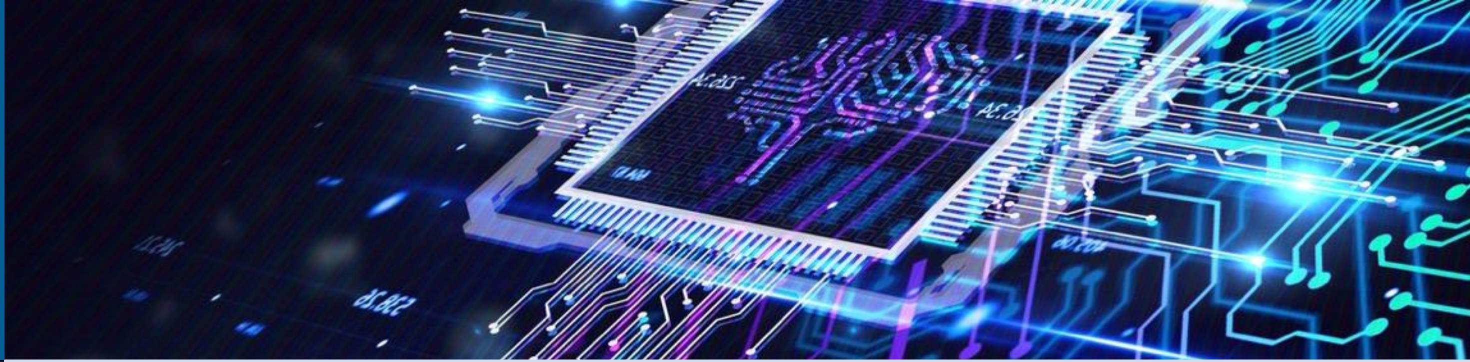
QCut - overview



The user wants to run a circuit that doesn't fit on the backends they have available. User then creates a circuit with cut operations, passes it to QCut with the desired observables and receives expectation values for the observables.

QCut decomposes the cut circuit into a group of circuits and passes them all to a Quantum backend and receives results for the circuits. QCut the applies post-processing to reconstruct expectation values for the original circuit.

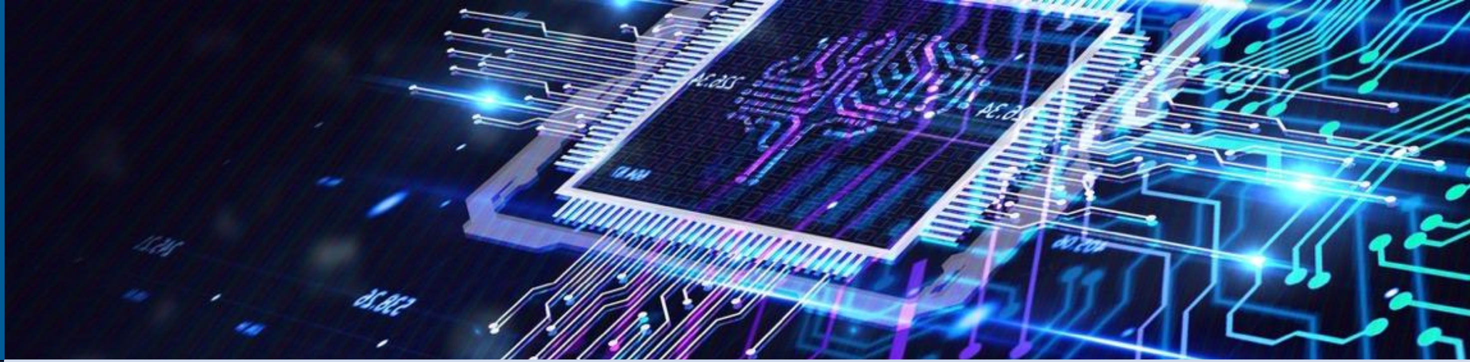
Quantum backend receives a group of circuits, executes them, and returns the results.



QCut - workflow example

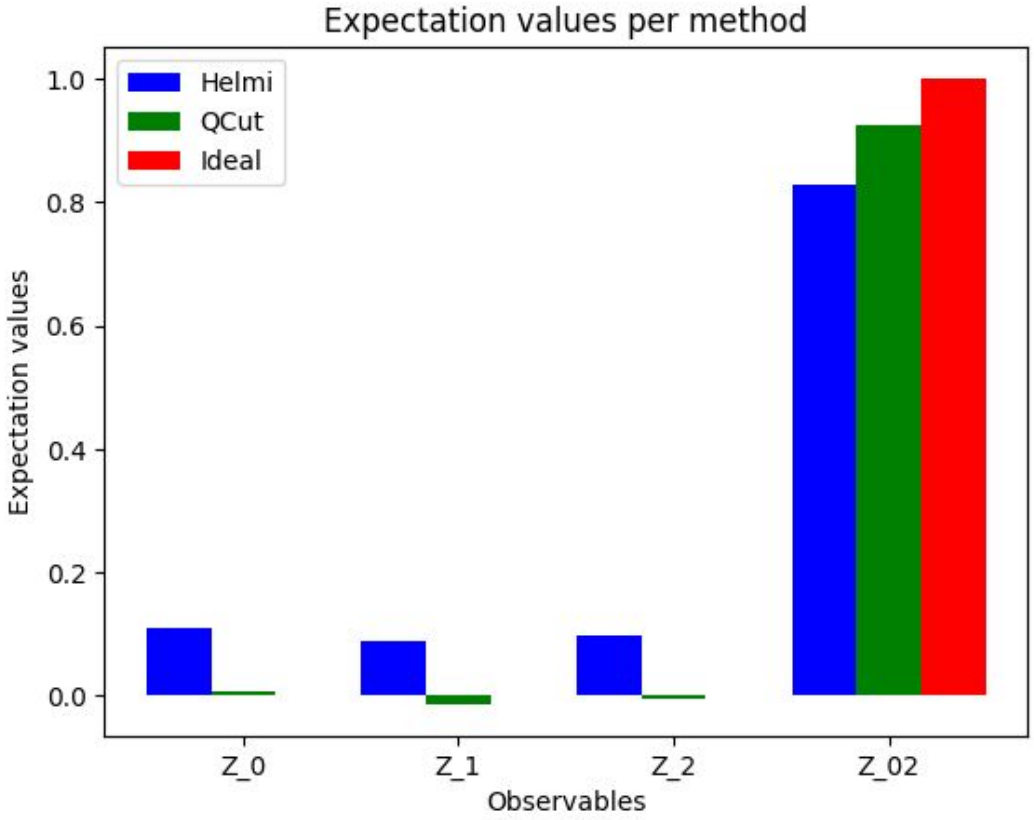
CSC

ICT Solutions for Brilliant Minds



QCut - Helmi

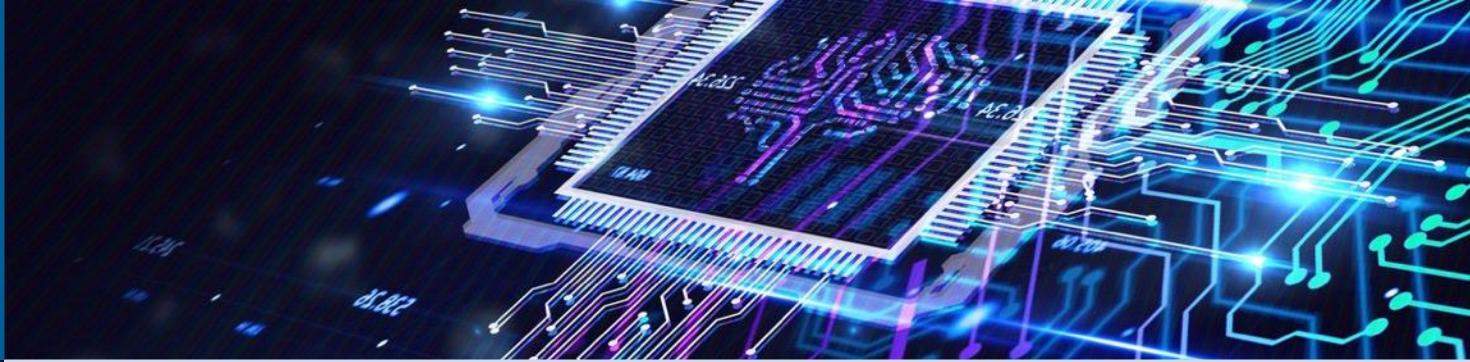
QCut - Helmi



Expectation values per method
Helmi
QCut
Ideal
1.0
0.8
0.6
0.4
0.2
0.0
Expectation values
Z_0
Z_1
Z_2
Z_02
Observables

CSC

ICT Solutions for Brilliant Minds



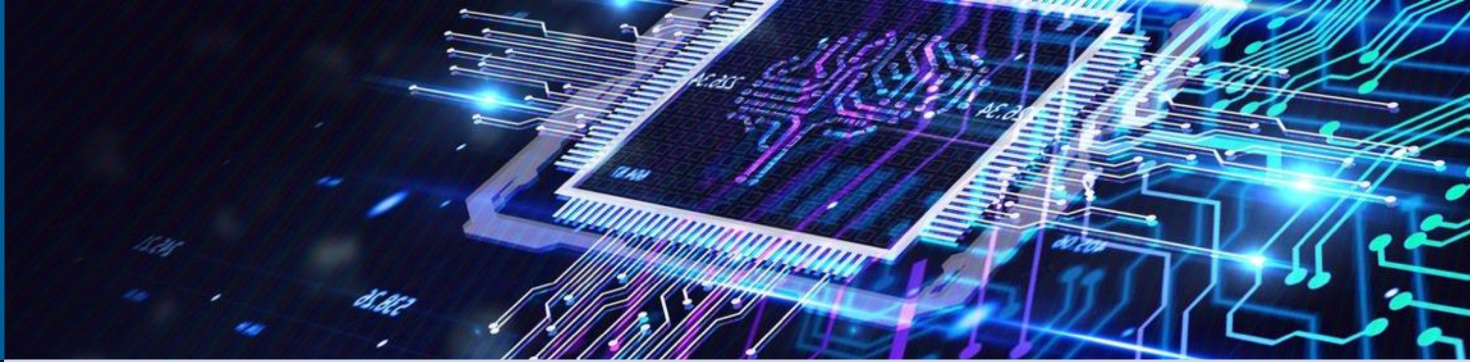
QCut - Applications

QCut - Applications

- Algorithms that return expectation values
    - Optimization problems
    - Variational algorithms
- Problems where circuit can be cut with only a few cuts
    - QAOA is a good and easy example
    - Quantum dynamics simulations

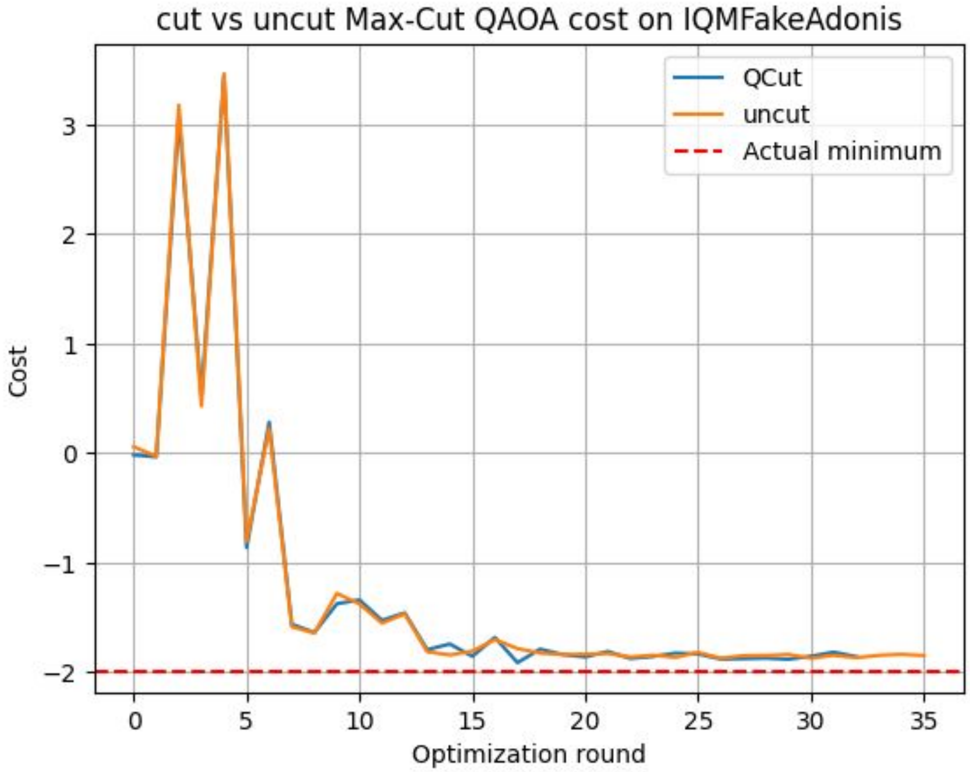
CSC

ICT Solutions for Brilliant Minds

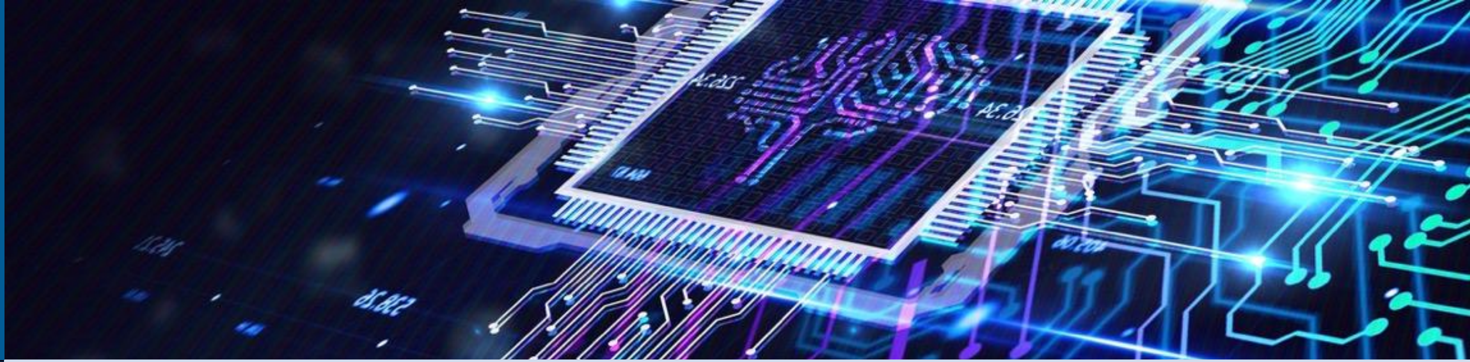


QCut - Applications - QAOA

QCut - QAOA


cut vs uncut Max-Cut QAOA cost on IQMFakeAdonis
QCut
uncut
Actual minimum
Cost
3
2
1
0
−1
−2
0
5
10
15
20
25
30
35
Optimization round

CSC
ICT Solutions for Brilliant Minds



More Circuit Knitting

CSC

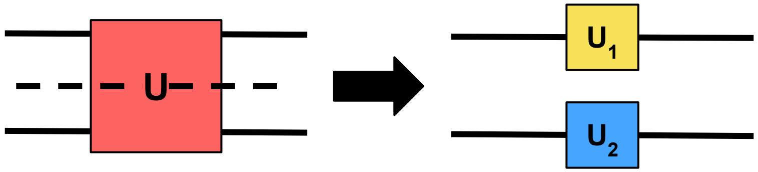
ICT Solutions for Brilliant Minds

Gate cuts

Gate cut

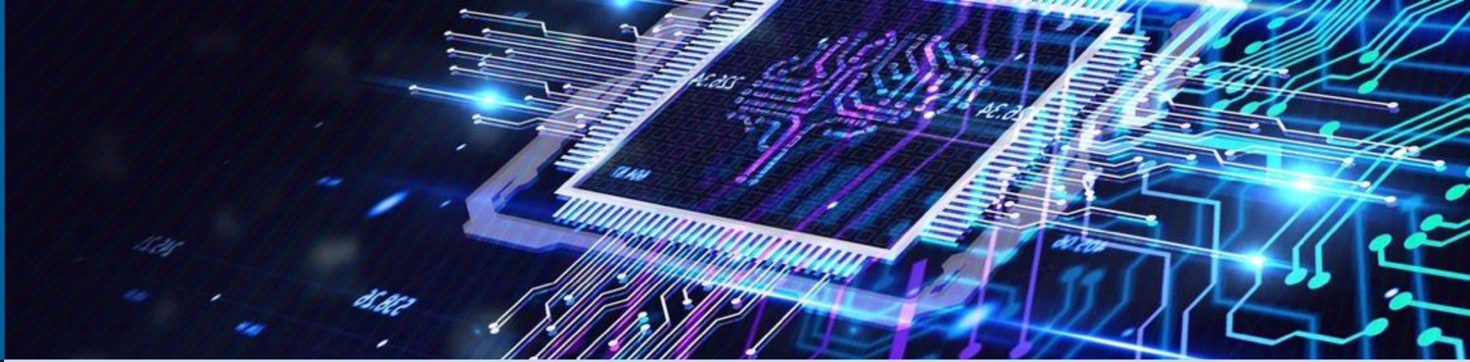
- Works the same as wire cutting
- QPD depends on the gate being cut
- For CX-family gates γ=3
- For SWAP γ=7
- Smaller circuit overhead
- Needs mid-circuit measurements

Operation
$I \otimes I$
$A_1 \otimes A_2$
$M_{A_1} \otimes e^{\pm i\pi A_2/4}$
$e^{\pm i\pi A_1/4} \otimes M_{A_2}$



CSC

ICT Solutions for Brilliant Minds



Classical Communication

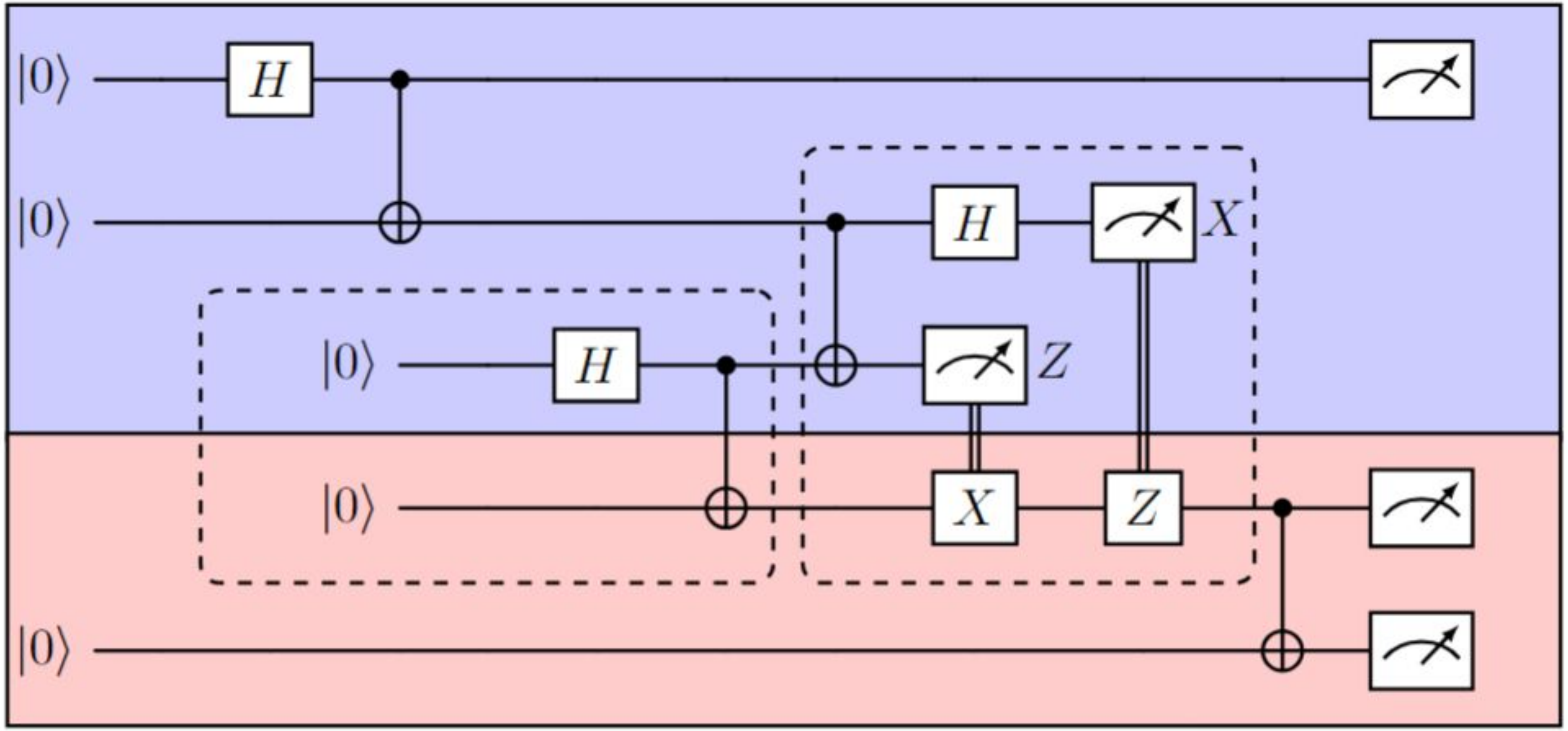
Classical communication

1. Local operations (LO)
    a. circuits completely independent
    b. wire and gate cuts
2. LO and one-way communication
    a. communication can be implemented via
    b. classical processing between circuit pairs
    c. wire cuts
3. LO and two-way communication
    a. real time communication between circuits
    b. gate cuts

One-way communication

- Wire cuts via teleportation
- Multiple simultaneous cuts cheaper than multiple individual ones
- $\gamma = (2^{k+1} - 1)^{1/k}$
  - k is the number of simultaneous cuts
- For a single cut transforms wire cut into cutting the CNOT gate

One-way communication



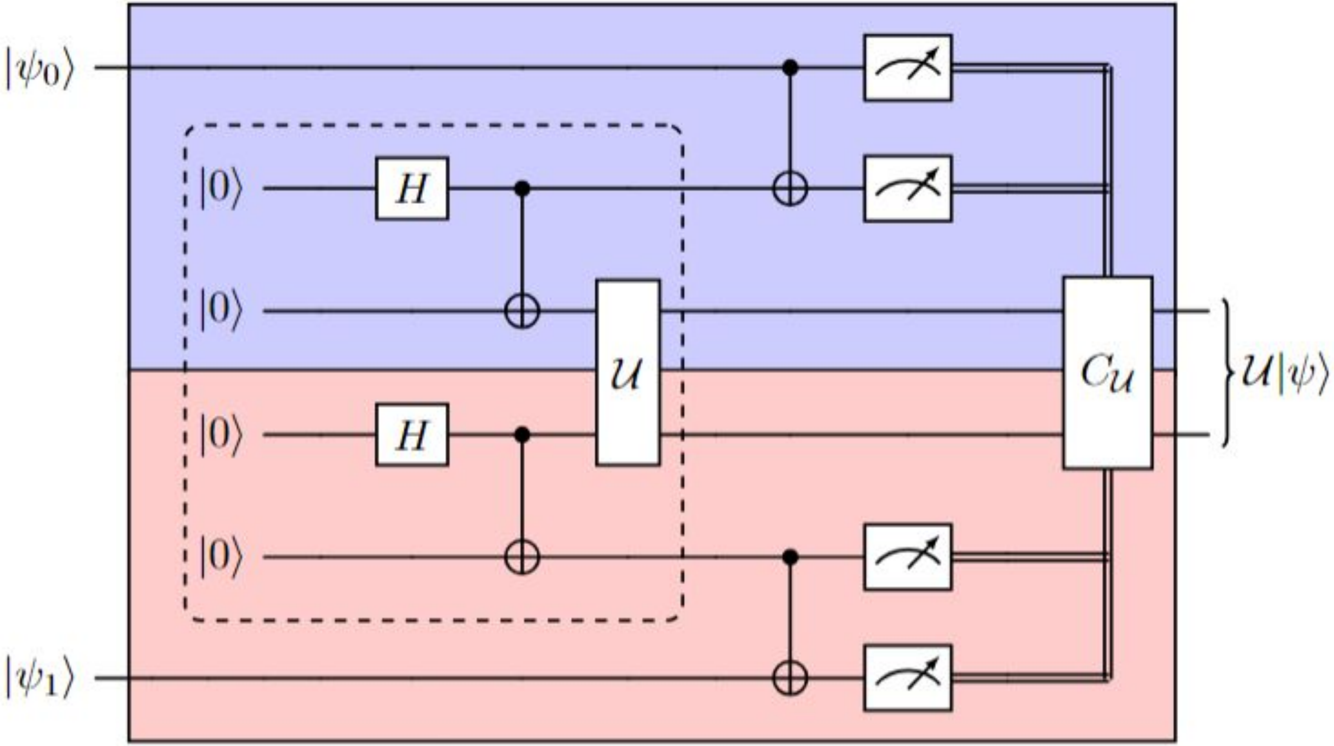
Two-way communication

- Gate cuts with teleportation through Choi state
- Multiple simultaneous cuts cheaper than multiple individual ones

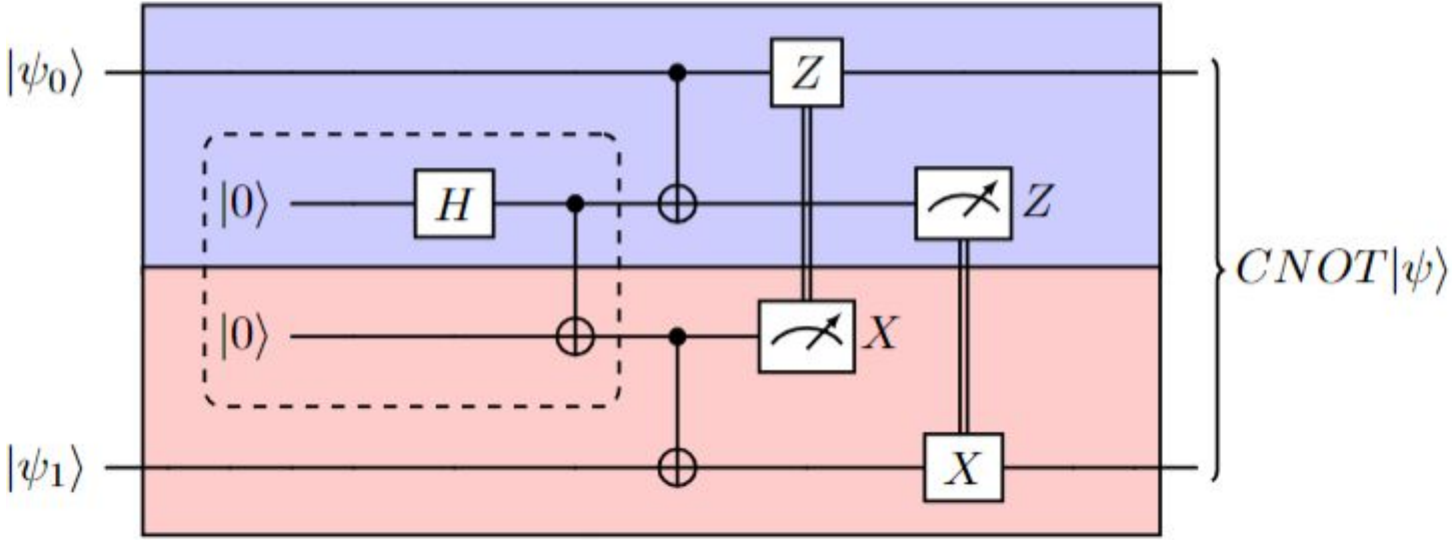
$$\gamma_{LOCC}(\mathcal{U}) = \left(2\left(\sum_i \alpha_i\right)^{2k} - 1\right)^{1/k}$$

- for CNOT γ= $(2^{k+1} - 1)^{1/k}$
  - k is the number of simultaneous cuts

Two-way communication

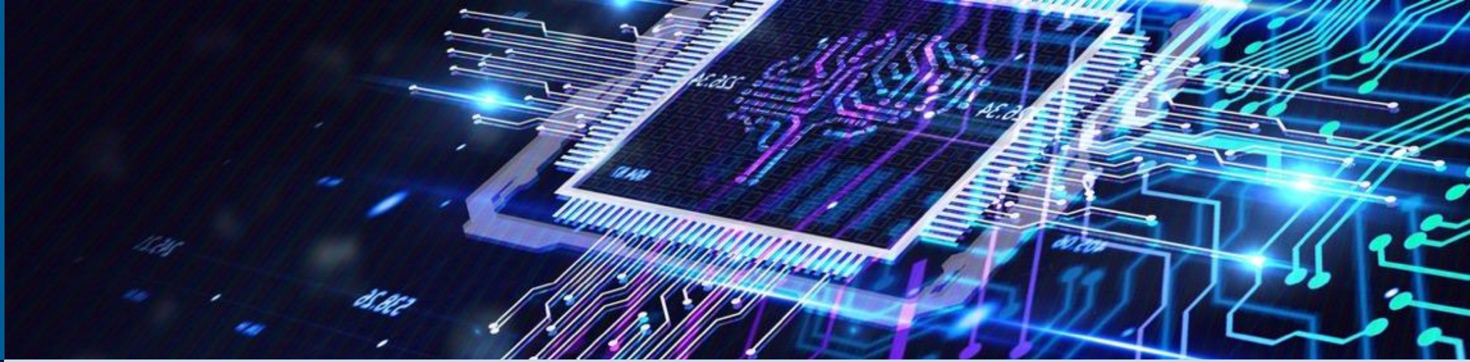


Two-way communication



CSC

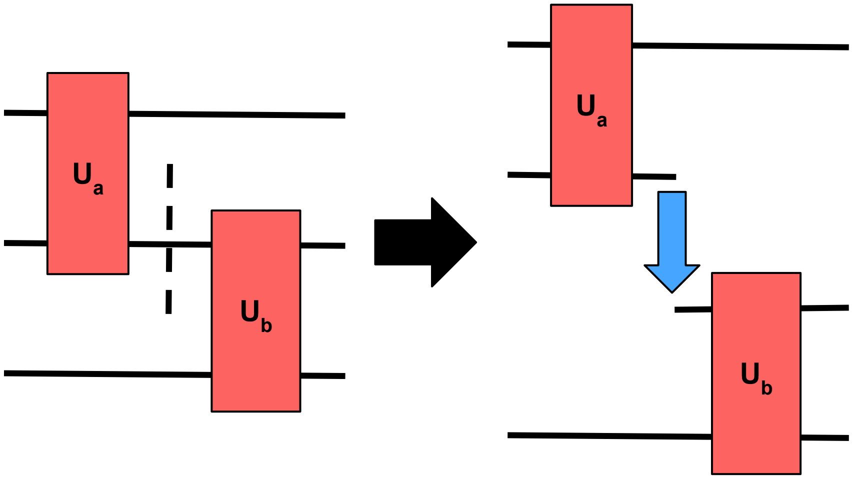
ICT Solutions for Brilliant Minds



Quantum Communication

Quantum communication

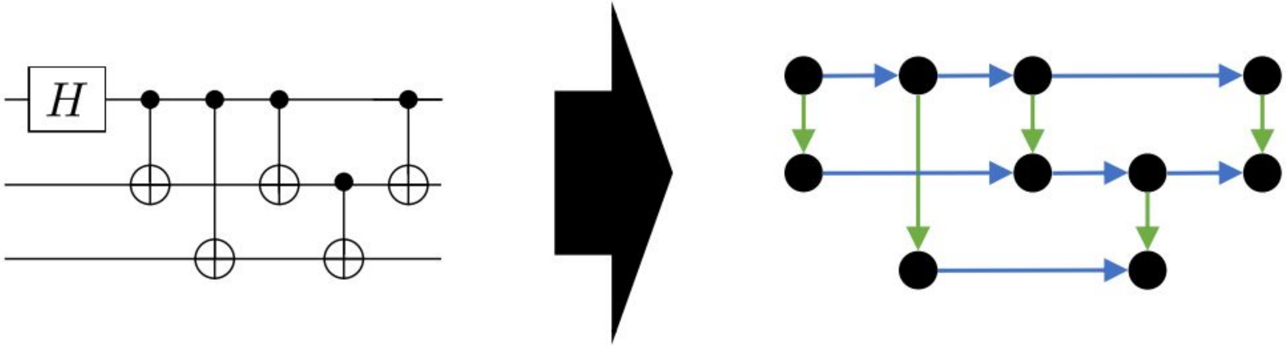
- Greatly simplifies wire cuts
- No QPD sampling required
- Only overhead from transferring a quantum state



Optimal cut locations

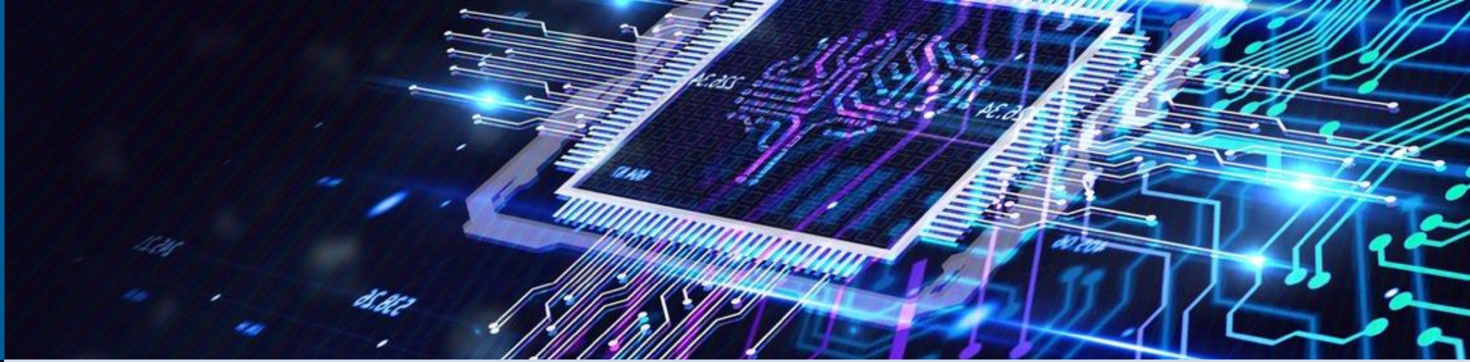
Finding optimal cuts

- Minimize overhead from cuts
- Circuit to a graph
  - for each 2-qubit gate 2 vertices
  - edges are possible wire and gate cuts
  - edges weighed by their overhead
- Finding cuts can be thought of as partitioning the graph



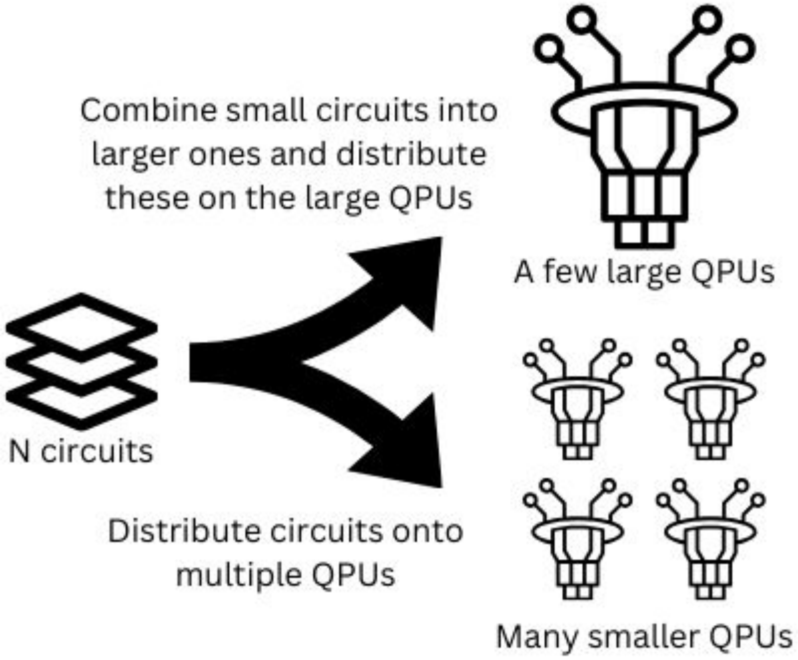
CSC

ICT Solutions for
Brilliant Minds



Parallel quantum computing

Parallel quantum computing



Joonas Nivala

linkedin: @JooNiv
email: joonas.nivala@csc.fi

facebook.com/CSCfi

twitter.com/CSCfi

linkedin.com/company/csc---it-center-for-science

github.com/CSCfi