Boson sampling simulation enhanced by FPGA based data-flow engines

Peter Rakyta

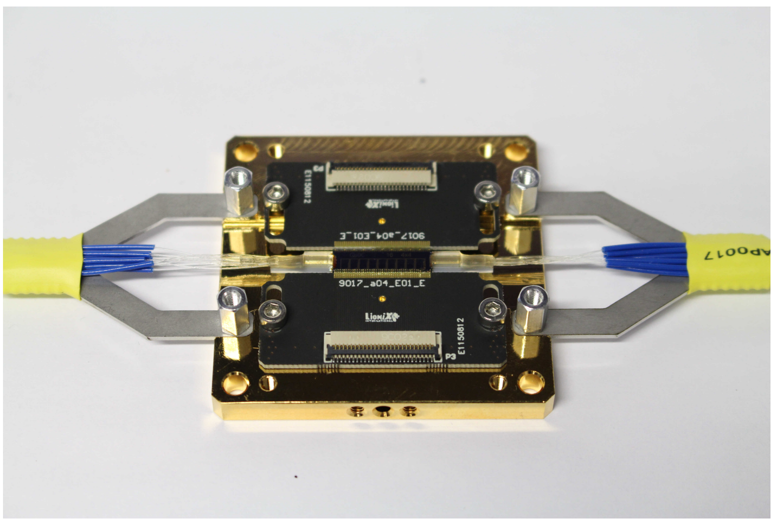
Department of Physics of Complex Systems

Optical Quantum Computing

Quantum Information
National Laboratory
HUNGARY

optical interferometers

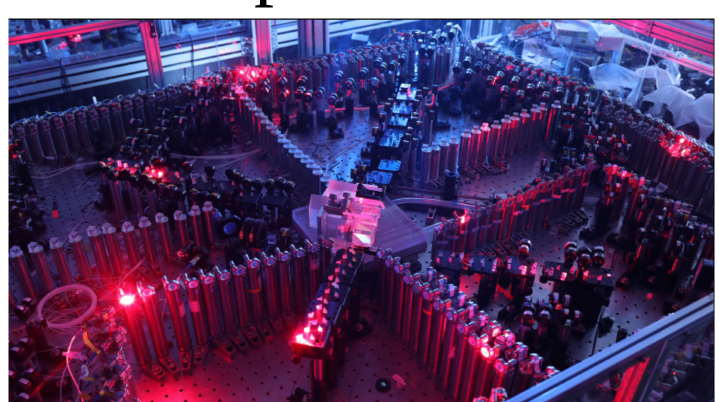
on chip



- N optical modes occupied by 0,1,2, ... N photons (instead of qubits)
- single photon sources or Gaussian sates

generated from coherent laser beams

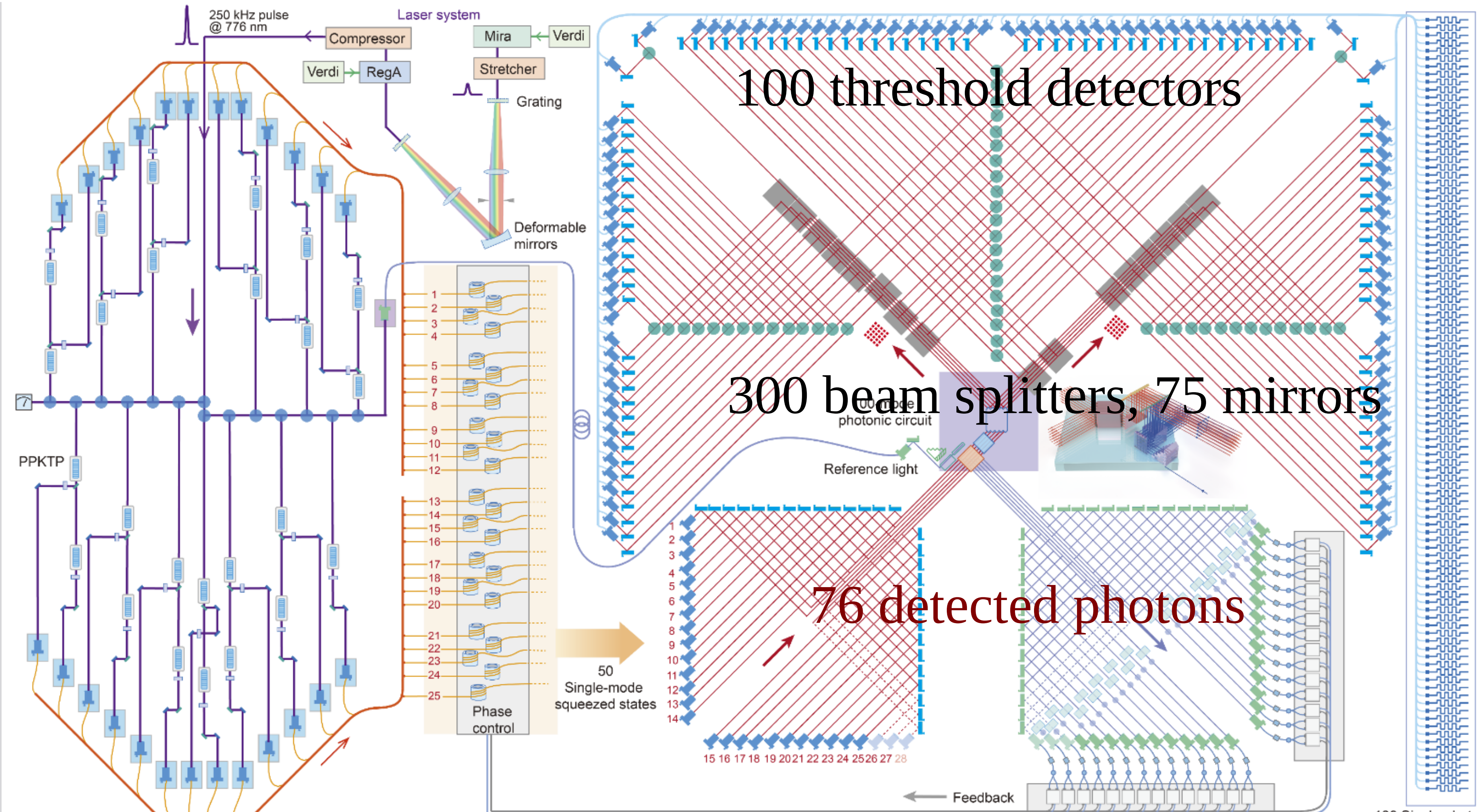
on optical table



ELTE
EÖTVÖS LORÁND
UNIVERSITY

Quantum advantage showed on Boson Sampling from a Gaussian state

Quantum Information National Laboratory HUNGARY



Quantum computational advantage using photons. Science, 370(6523):1460–1463, 2020.

University of Science and Technology of China
Chinese Academy of Sciences, Tsinghua University, China

ELTE EÖTVÖS LORÁND UNIVERSITY

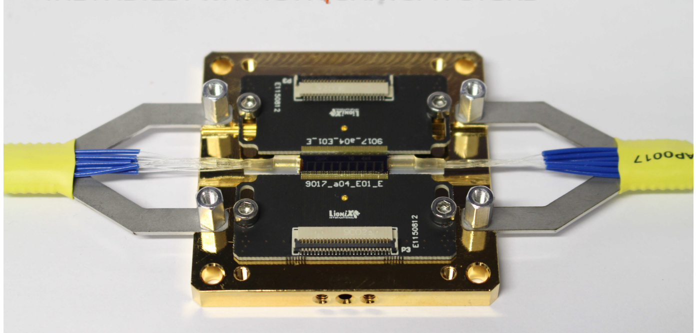
Programable optical integrated circuits

Quantum Information
National Laboratory
HUNGARY

Spin-off University of Twente

QUIX

THE FASTEST WAY TO A QUANTUM FUTURE



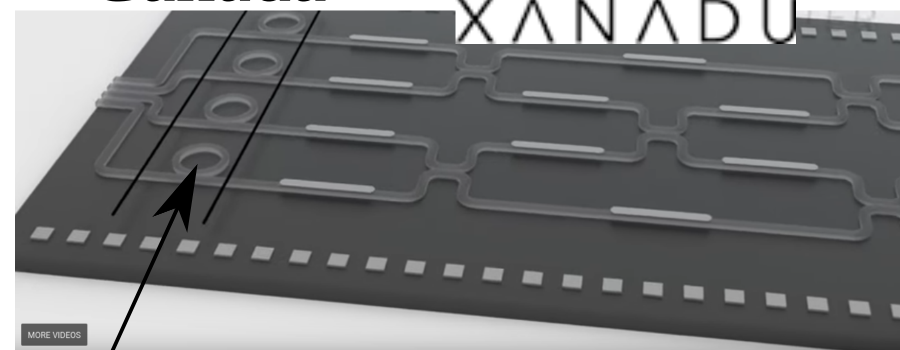
Netherlands

conencted to single photon
sources and PNR detectors

12 photonic modes

Canada

XANADU



creates squeezed state from
classical laser input - this is the qubit

24 photonic modes

Strawberry Fields:
quantum computer simulator

PennyLane:
first library for quantum machine learning

ELTE
EÖTVÖS LORÁND
UNIVERSITY

The Piquasso project

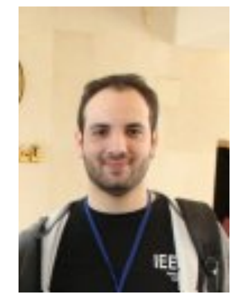
Quantum Information
National Laboratory
HUNGARY

PIQUASSO

Why don't we put together a bosonic quantum simulator?

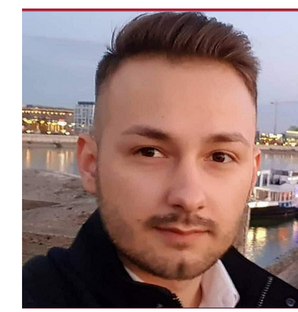
Zoltán Zimborás
Wigner

Kozsik tamás



Kareem El Safty

Poór Boldizsár

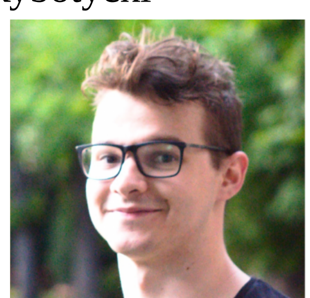


Jóczik Szabolcs

Michał Oszmaniec

Tomasz Rybotycki

Kaposi Ágoston



Kolarovszki Zoltán

MAXELER
Technologies
Maximum Performance Computing

Supported by
ERICSSON

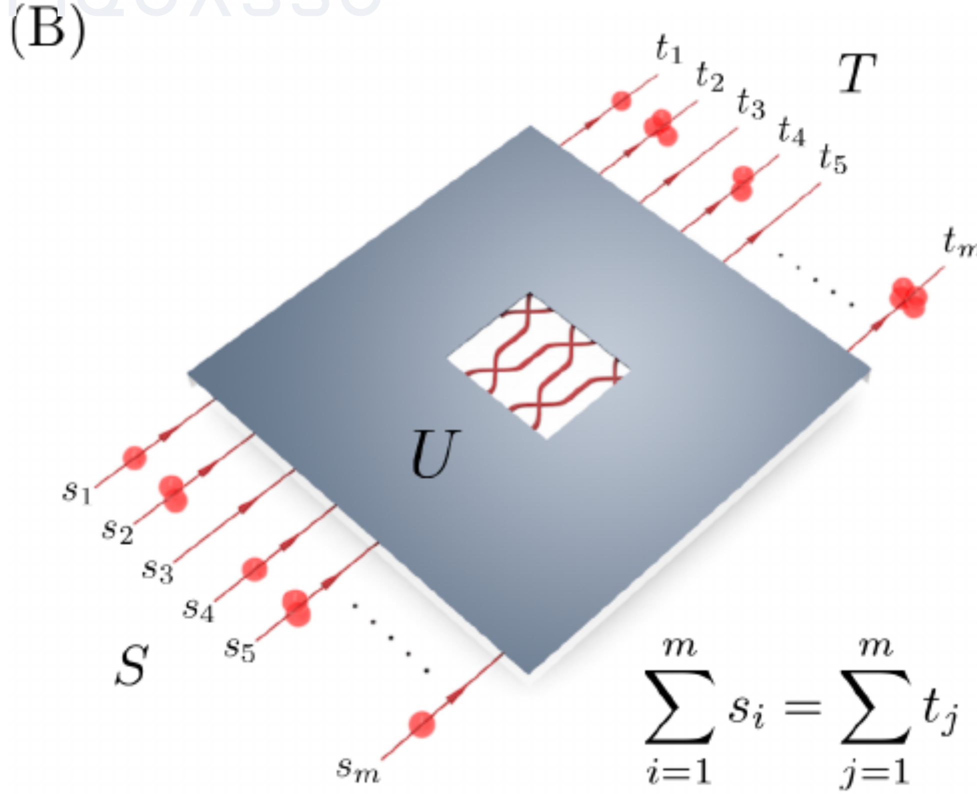
ELTE
EÖTVÖS LORÁND
UNIVERSITY

Boson Sampling

Quantum Information National Laboratory HUNGARY

The Computational Complexity of Linear Optics

Scott Aaronson* Alex Arkhipov†



$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{m} t_j = n$$

$$P[S \to T] = \frac{|Per(U_{S,T})|^2}{s_1! \ldots s_m! t_1! \ldots t_m!}$$

$$\mathrm{Per}\,(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)},$$

Set of permutations of (1,2,3 .... n)

$\mathcal{O}((n-1)\cdot n!)$

$\mathcal{O}(n^2 \cdot 2^n)$

Ryser's formula:

$$\mathrm{Per}(A) = (-1)^n \sum_{S \subseteq \{1,2,\ldots,n\}} (-1)^{|S|} \prod_{i=1}^{n} \sum_{j \in S} a_{ij}$$

BB/FG's formula:

$$\mathrm{Per}(A) = \frac{\sum_\delta \left(\prod_{k=1}^n \delta_k\right) \prod_{i=1}^n \sum_{j=1}^n \delta_j a_{ij}}{2^{n-1}}$$

$\delta = \{\delta_1, \delta_2, ..., \delta_n\}$ $\delta_1 = 1$ and $\delta_i \in \{-1, 1\}$ for $2 \leq i \leq n$

PIQUASSO

ELTE EÖTVÖS LORÁND UNIVERSITY

Permanent calculation complexity

Quantum Information National Laboratory HUNGARY

PIQUASSO

Further complexity improvements by Gray code ordering:

0	000	0	000
1	001	1	001
2	010	3	011
3	011	2	010
4	100	6	110
5	101	7	111
6	110	5	101
7	111	4	100

$\mathcal{O}(n^2 \cdot 2^n)$ complexity reduced to $\mathcal{O}(n \cdot 2^n)$

data recycling in expressions:

$$\sum_{j \in S} a_{ij} \qquad \sum_{j=1}^{n} \delta_j a_{ij}$$

in exchange of scalability reduction?

Strawberry Fields

Ryser's formula: $\mathrm{Per}(A) = (-1)^n \sum_{S \subseteq \{1,2,\ldots,n\}} (-1)^{|S|} \prod_{i=1}^{n} \sum_{j \in S} a_{ij}$

Piquasso

BB/FG's formula: $\mathrm{Per}(A) = \dfrac{\sum_\delta \left(\prod_{k=1}^{n} \delta_k\right) \prod_{i=1}^{n} \sum_{j=1}^{n} \delta_j a_{ij}}{2^{n-1}}$

ELTE EÖTVÖS LORÁND UNIVERSITY

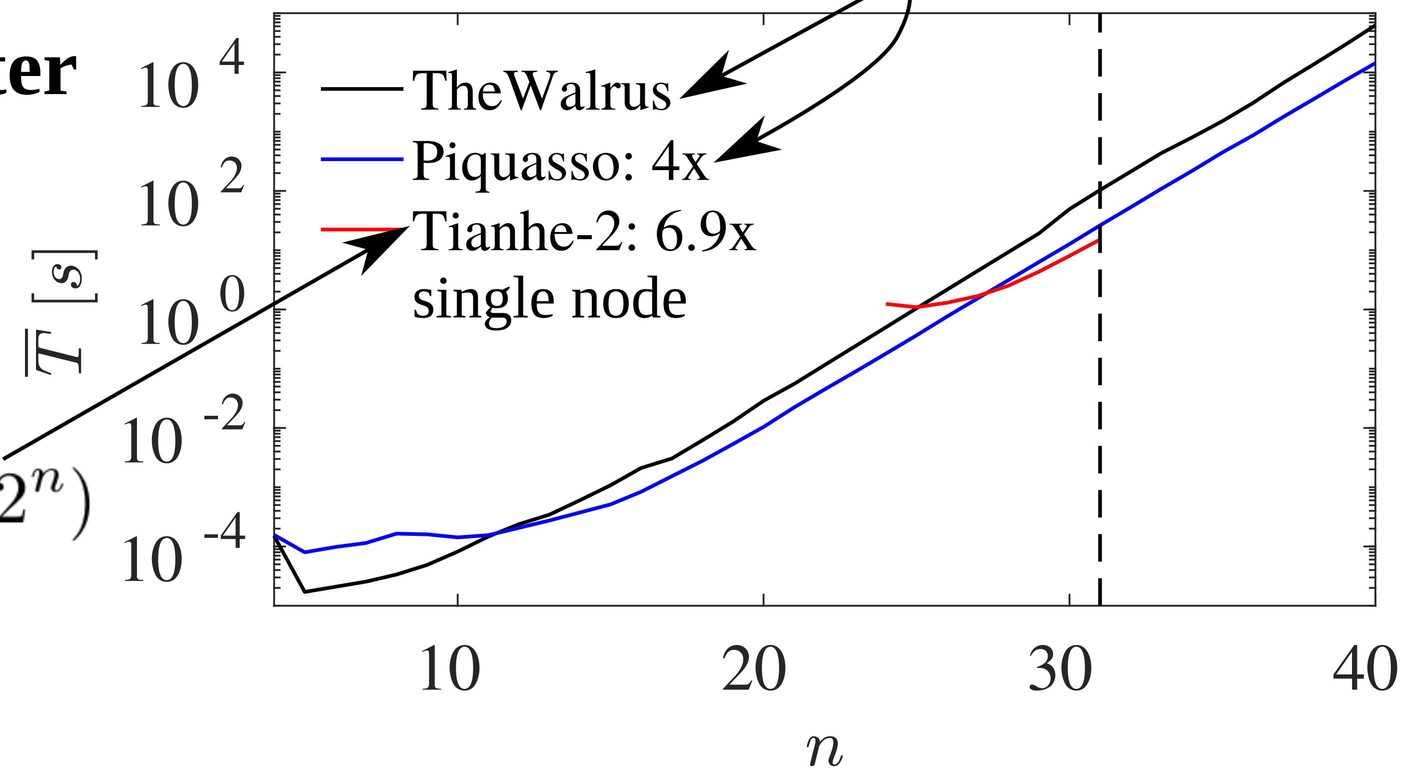
Permanent calculation benchmark

$\mathcal{O}(n \cdot 2^n)$

Intel Xeon Gold 6130, 24 threads

Tianhe-2 supercomputer

$\mathcal{O}(n^2 \cdot 2^n)$



16 000 computing nodes

Intel Xeon E5: dual socet x 12 cores x 2 multithread = 48 threads

Xeon Phi 31S1P: 3 PCI cards x 57 cores x 4 multithreads = 684 threads

total: 33.86 petaflops

limit: permanent of n=50 matrix in 1 hour

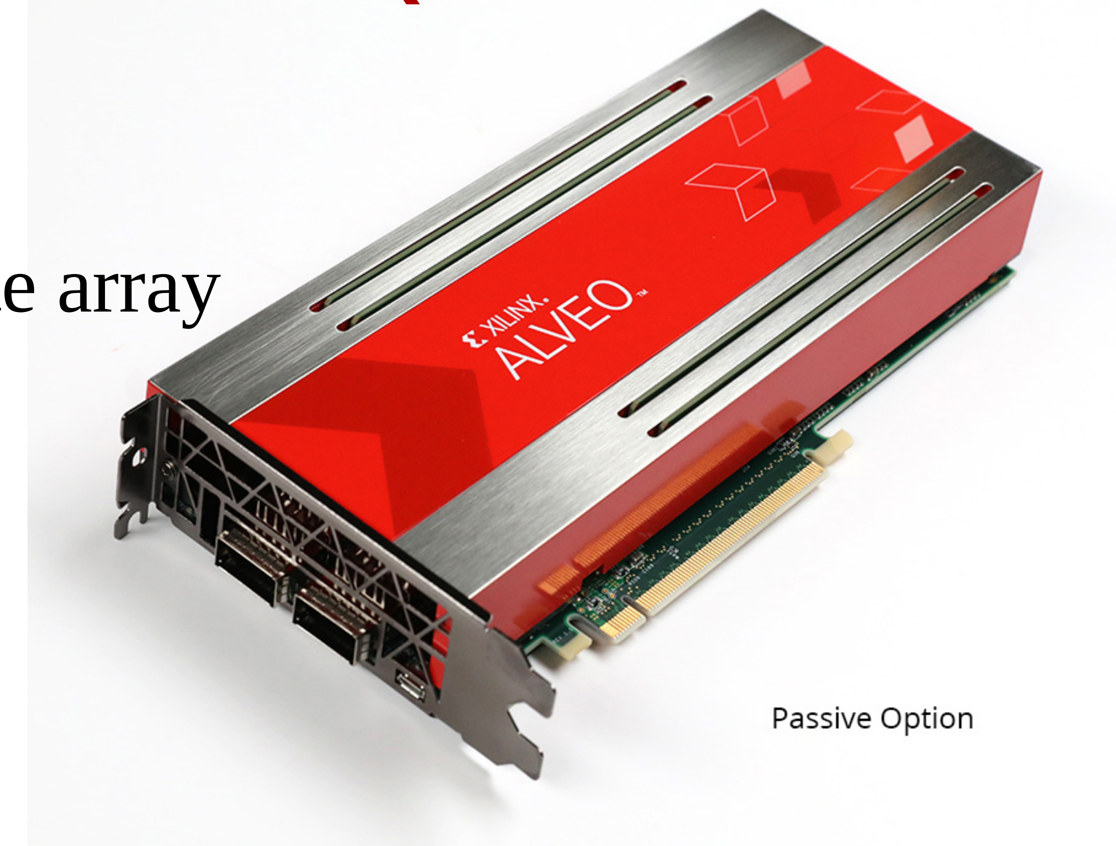
National Science Review, Volume 5, Issue 5, September 2018, Pages 715–720

FPGA based data-flow engines (DFE's)

PIQUASSO

Quantum Information
National Laboratory
HUNGARY

FPGA: field-programmable gate array



programmable logic gates,
memory units,
arithmeic units,
or other elements

particularly useful for prototyping application-specific integrated circuits (ASICs) or processors.

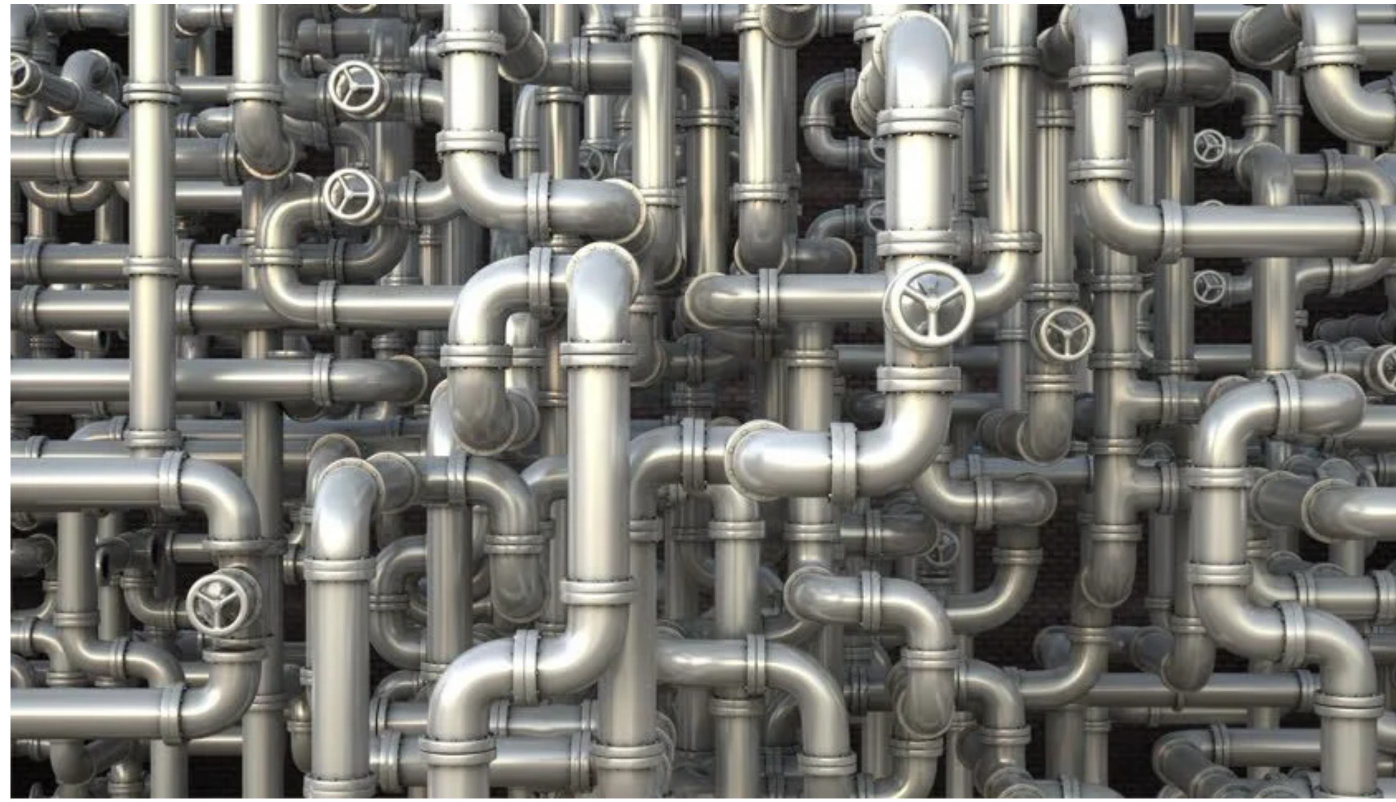
Quamtum Computing inspired ASICs?
Coprocessors for real Quantum Hardware?

ELTE
EÖTVÖS LORÁND
UNIVERSITY

FPGA based data-flow engines (DFE's)

Quantum Information National Laboratory HUNGARY

MAXELER Technologies
Maximum Performance Computing



Data streams flowing through the FPGA chip

automatized time and space constraints

FPGA hardware + data-flow programming model = DFE

ELTE
EÖTVÖS LORÁND UNIVERSITY

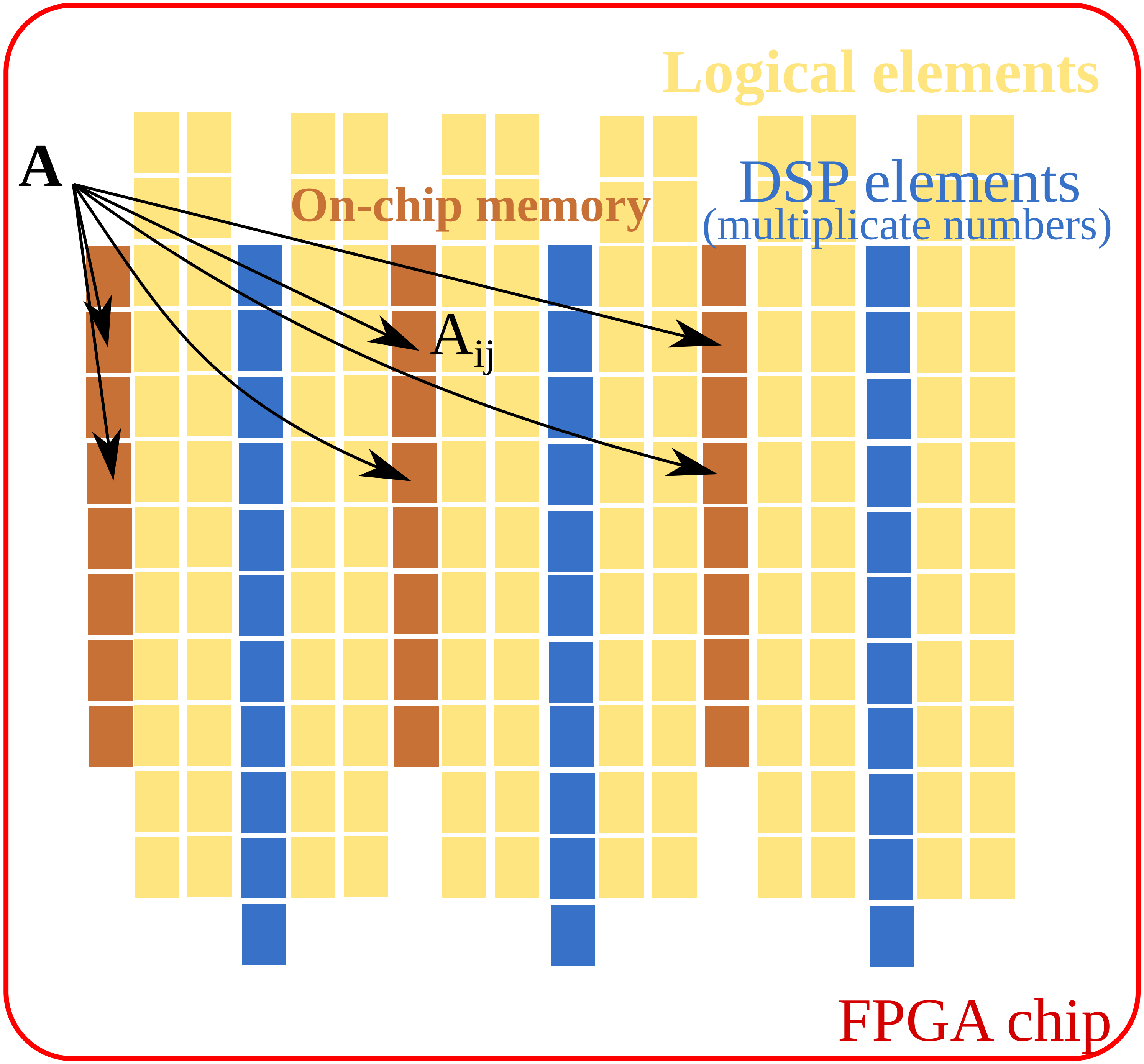
supported by Xilinx University Program

DFE implementation of permanent calculation

$$\mathrm{Per}(A) = \frac{\sum_{\delta}\left(\prod_{k=1}^{n}\delta_k\right)\prod_{i=1}^{n}\sum_{j=1}^{n}\delta_j a_{ij}}{2^{n-1}}$$

spread over the FPGA chip

step 1:
upload matrix A into on-chip memories



DFE implementation of permanent calculation

$$\mathrm{Per}(A) = \frac{\sum_\delta \left(\prod_{k=1}^n \delta_k\right) \prod_{i=1}^n \sum_{j=1}^n \delta_j a_{ij}}{2^{n-1}}$$

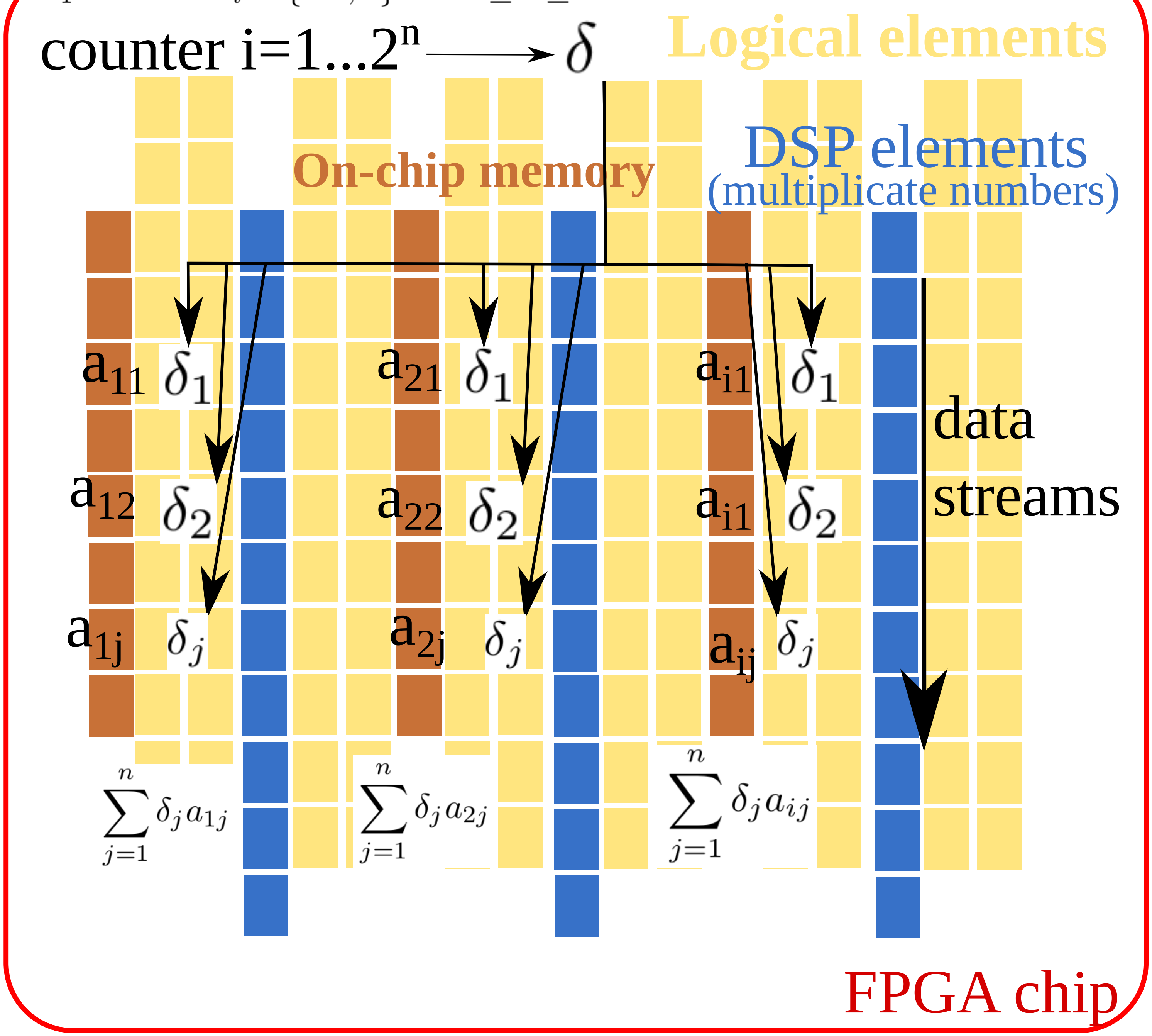
$\delta_1 = 1$ and $\delta_i \in \{-1, 1\}$ for $2 \leq i \leq n$

step 2:

create counter for δ

split it into bits: δ_j

calculate for each i $\sum_{j=1}^{n} \delta_j a_{ij}$



PIQUASSO

ELTE
EÖTVÖS LORÁND UNIVERSITY

DFE implementation of permanent calculation

$$\mathrm{Per}(A) = \frac{\sum_\delta \left(\prod_{k=1}^n \delta_k\right) \prod_{i=1}^n \sum_{j=1}^n \delta_j a_{ij}}{2^{n-1}}$$

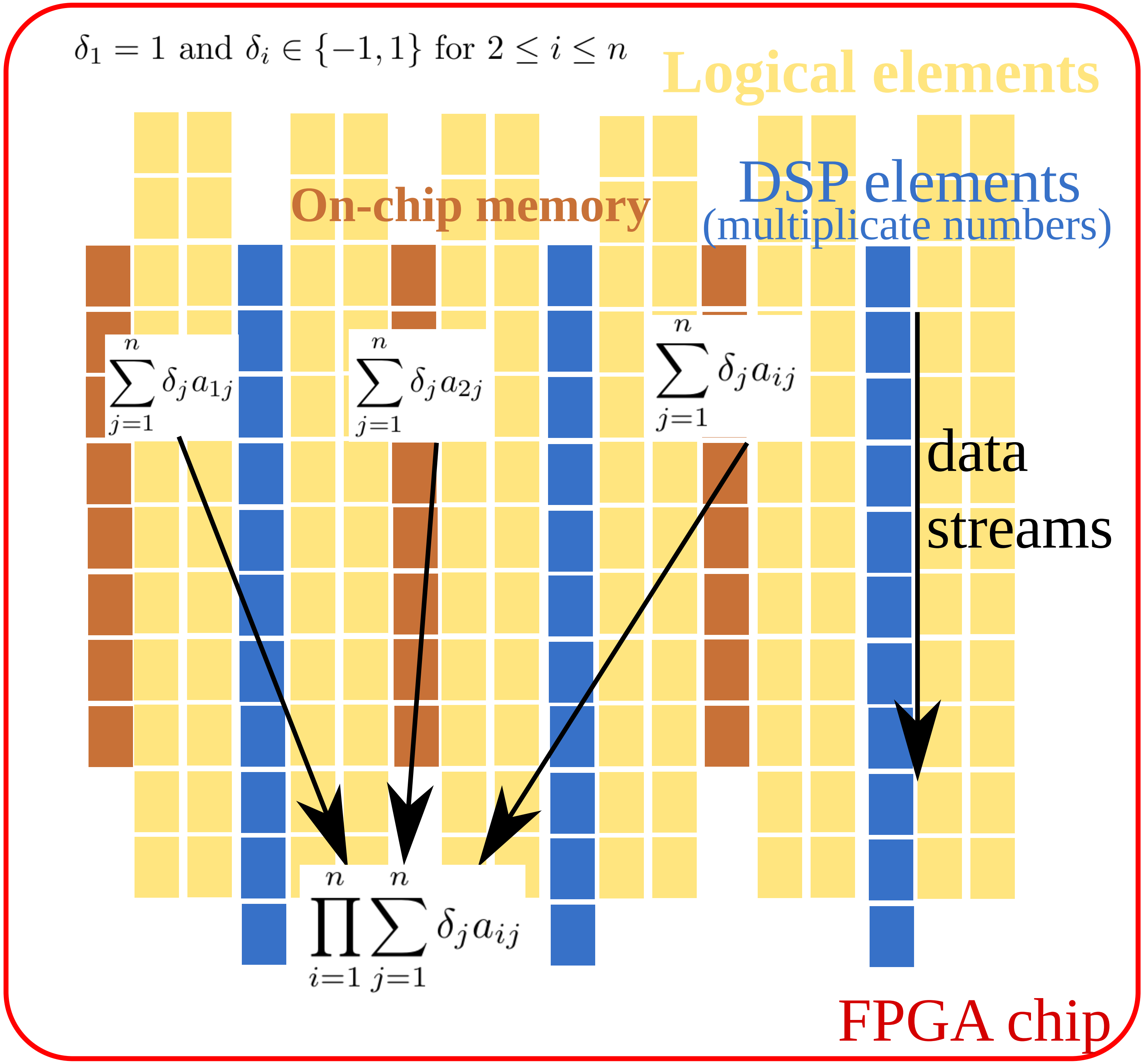
step 3:

calculate product

$$\prod_{i=1}^n \sum_{j=1}^n \delta_j a_{ij}$$

step 4:

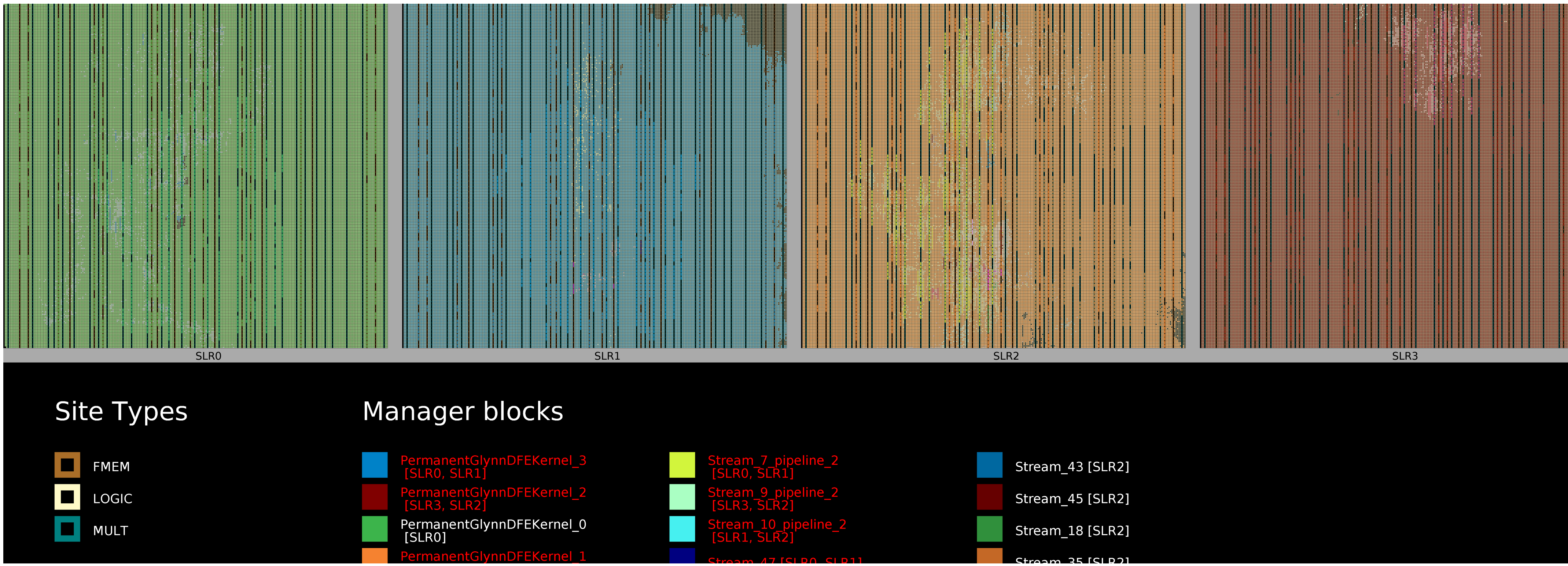
sum up the addends and return to CPU



DFE implementation of permanent calculation

Quantum Information
National Laboratory
HUNGARY

clock frequency: 180 MHz

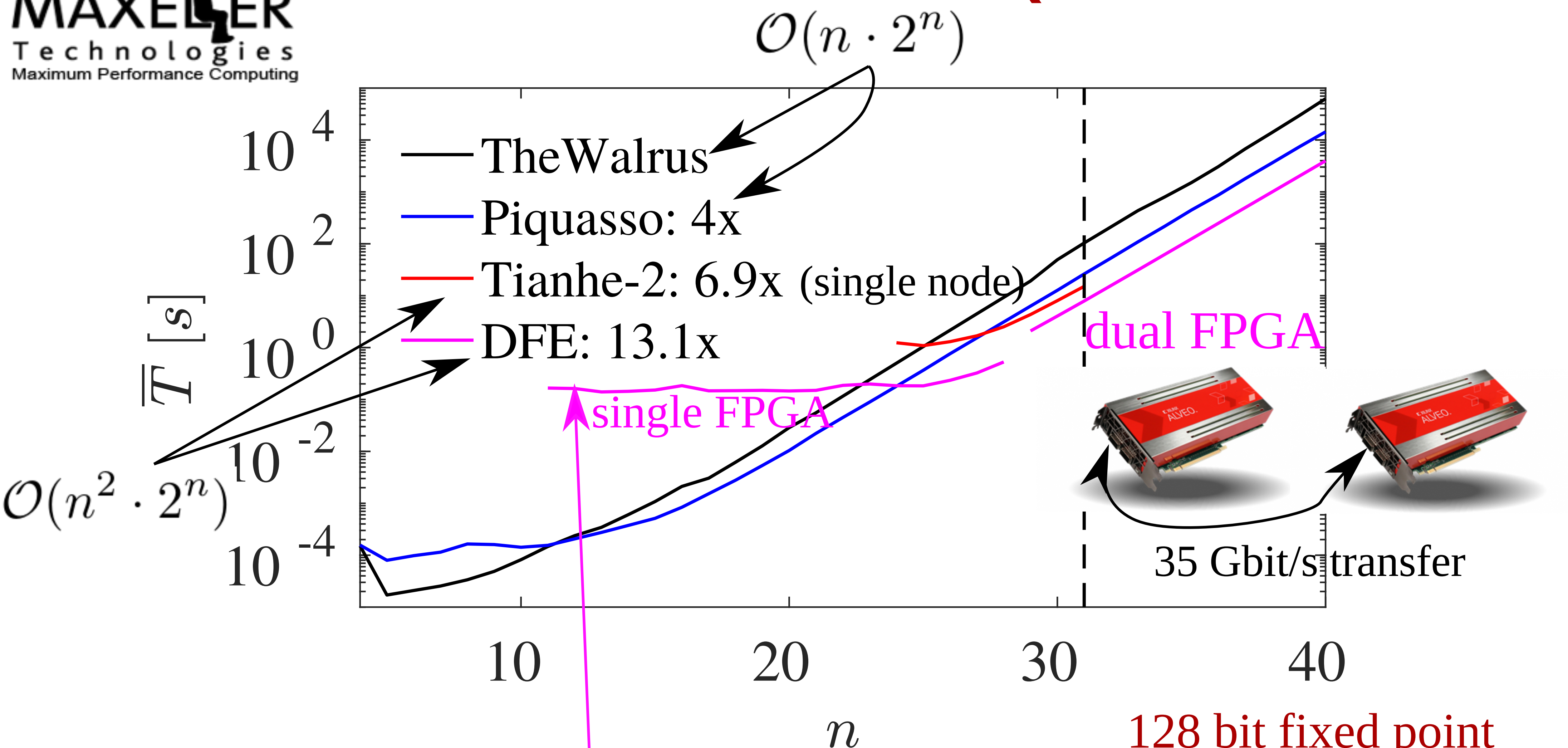


MAXELER Technologies
Maximum Performance Computing

ELTE
EÖTVÖS LORÁND
UNIVERSITY

Logic utilization:	3750968 / 5184000 (72.36%)
LUTs:	1665146 / 1728000 (96.36%)
Primary FFs:	2085822 / 3456000 (60.35%)
DSP blocks:	4000 / 12288 (32.55%)
Block memory (BRAM18):	3365 / 5376 (62.59%)
Block memory (URAM):	12 / 1280 (0.94%)

Permanent benchmark again



uploading the program to FPGA takes time

128 bit fixed point number representation

Time complexity on DFE: $\mathcal{O}(C \cdot 2^n)$

n^2 operations are dispersed in space over the FPGA chip

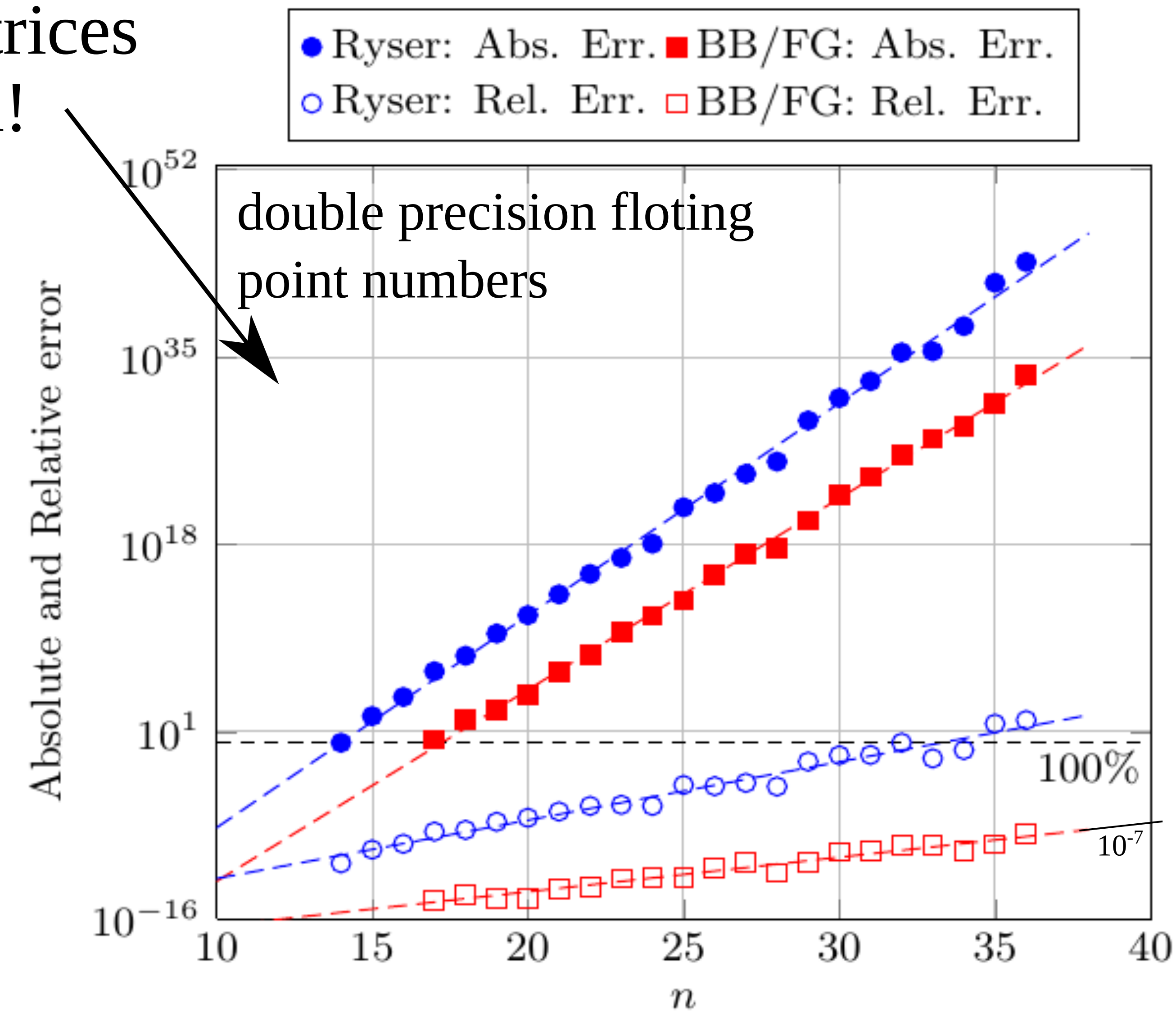
Numerical error of permanent calculation

Quantum Information
National Laboratory
HUNGARY

Junjie Wu, Yong Liu, Baida Zhang, Xianmin Jin, Yang Wang, Huiquan Wang, Xuejun Yang, National Science Review, Volume 5, Issue 5, September 2018

n x n all-ones matrices

Perm(A) = n!



If A = 40 x 40 unitary,

$$a_{ij} \sim \frac{1}{\sqrt{40}}$$

$$\mathrm{Perm}(A) \sim 10^{-12}$$

ELTE
EÖTVÖS LORÁND
UNIVERSITY

Aknowledgement

Quantum Information
National Laboratory
HUNGARY

This research was supported by the Ministry of Innovation and Technology and the National Research, Development and Innovation Office within the Quantum Information National Laboratory of Hungary

contact: Peter Rakyta, peter.rakyta@ttk.elte.hu

ELTE
EÖTVÖS LORÁND
UNIVERSITY