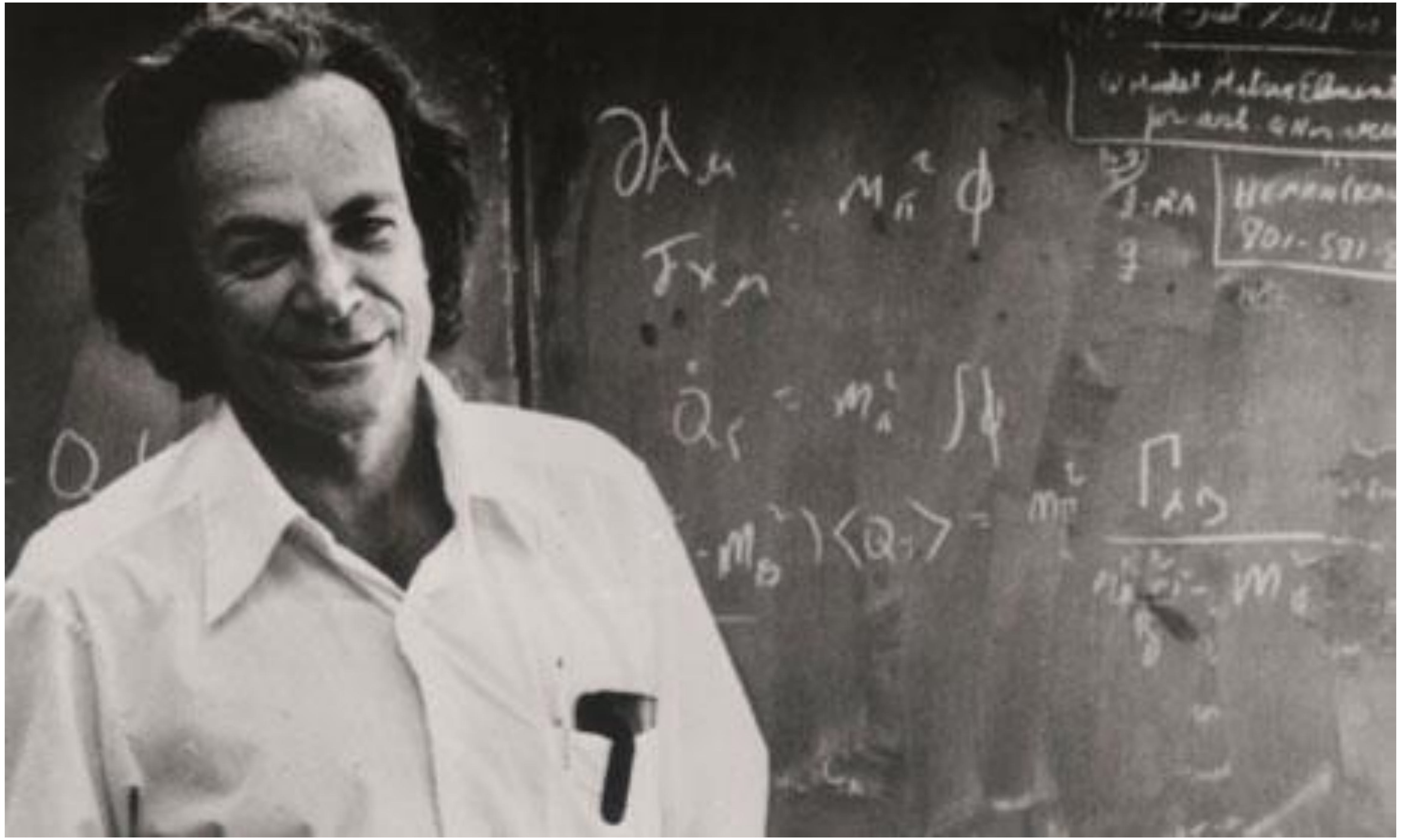
<QICrypton>

(The Quantum Security Evaluation Platform for Cryptographic Algorithms)

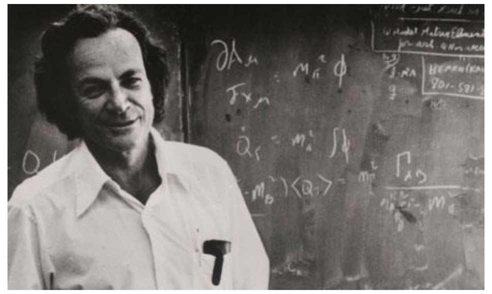
July. 21th, 2021
Sokjoon Lee

ETRI
KOREA UNIVERSITY
PUSAN NATIONAL UNIVERSITY
HANYANG UNIVERSITY
HANSUNG UNIVERSITY

Quantum Computer – History



Quantum Computer – History



Richard Feynman ('81)
Proposal of Quantum Computer (for Simulating Quantum Mechanics)

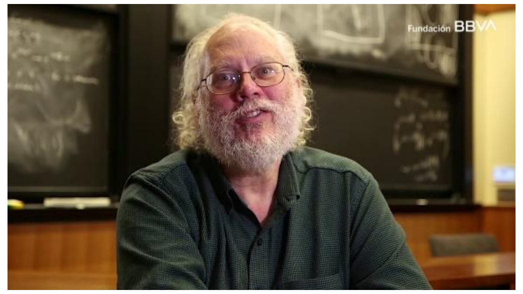
David Deutsch ('85)
The First Algorithm of Exponential Performance Improvement

Bernstein & Vazirani ('93)
Quantum-Classical Separation in Computational Complexity

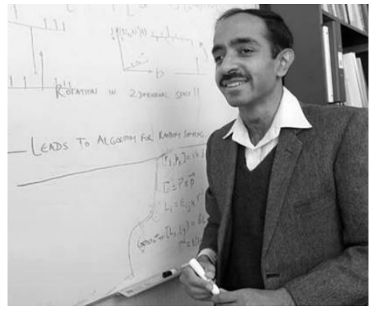
The first motivation : Quantum algorithm for attacking cryptography

Daniel Simon ('94)
Simon Algorithm (Period Finding Problem)

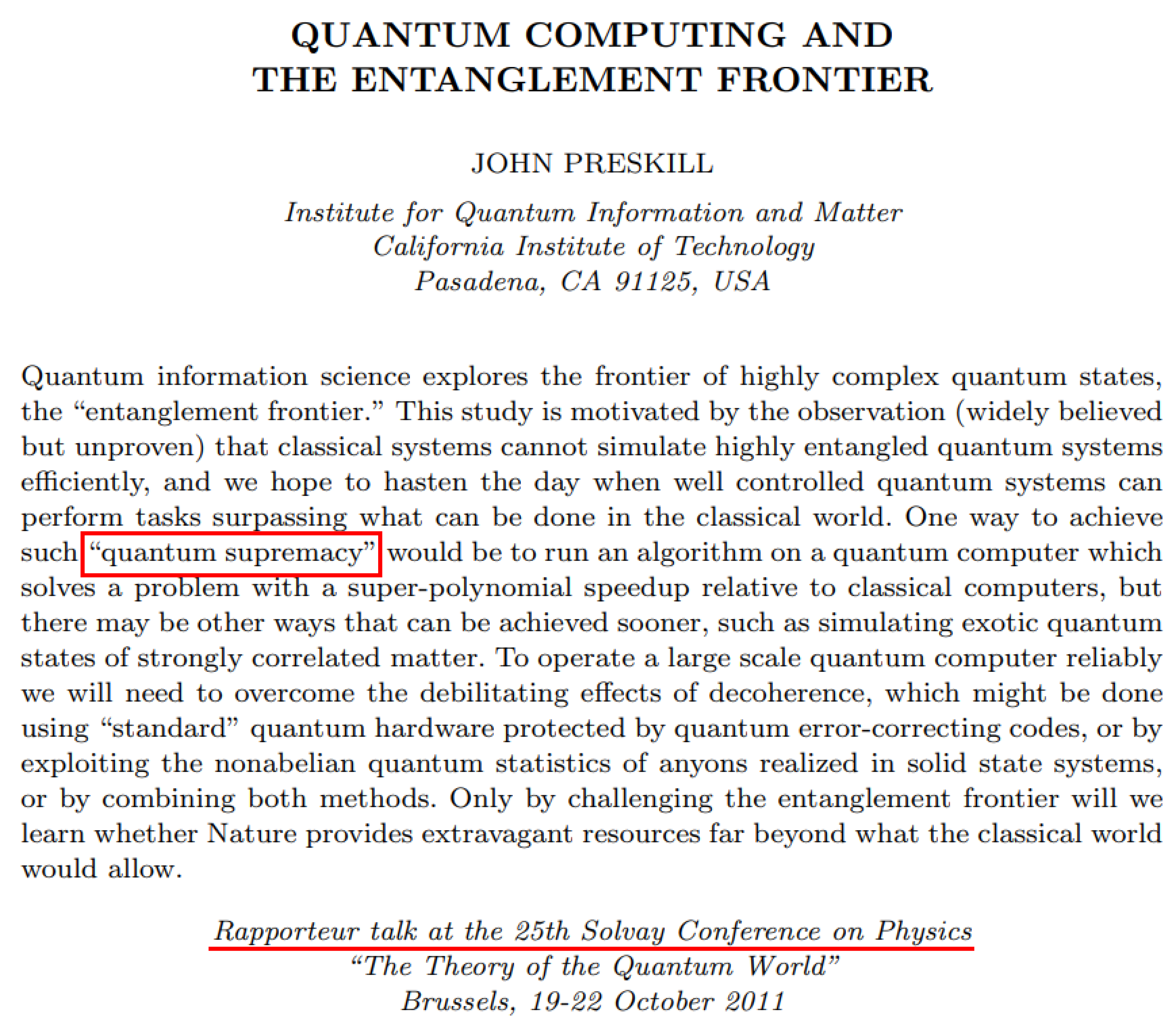
Peter Shor ('94)
Shor Algorithm (Factoring in Polynomial Time)



Lov Grover ('96)
Grover Algorithm (Quantum Search Algorithm in $O(\sqrt{N})$)



Quantum Computer – History


QUANTUM COMPUTING AND
THE ENTANGLEMENT FRONTIER

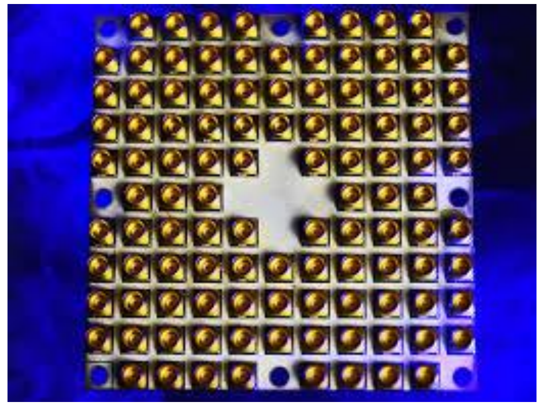
JOHN PRESKILL

Institute for Quantum Information and Matter
California Institute of Technology
Pasadena, CA 91125, USA

Quantum information science explores the frontier of highly complex quantum states, the "entanglement frontier." This study is motivated by the observation (widely believed but unproven) that classical systems cannot simulate highly entangled quantum systems efficiently, and we hope to hasten the day when well controlled quantum systems can perform tasks surpassing what can be done in the classical world. One way to achieve such "quantum supremacy" would be to run an algorithm on a quantum computer which solves a problem with a super-polynomial speedup relative to classical computers, but there may be other ways that can be achieved sooner, such as simulating exotic quantum states of strongly correlated matter. To operate a large scale quantum computer reliably we will need to overcome the debilitating effects of decoherence, which might be done using "standard" quantum hardware protected by quantum error-correcting codes, or by exploiting the nonabelian quantum statistics of anyons realized in solid state systems, or by combining both methods. Only by challenging the entanglement frontier will we learn whether Nature provides extravagant resources far beyond what the classical world would allow.

Rapporteur talk at the 25th Solvay Conference on Physics
"The Theory of the Quantum World"
Brussels, 19-22 October 2011

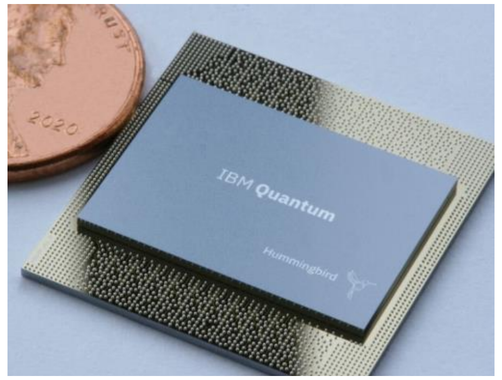
Quantum Computer – History



Intel Tangle Lake (Q49, '18)

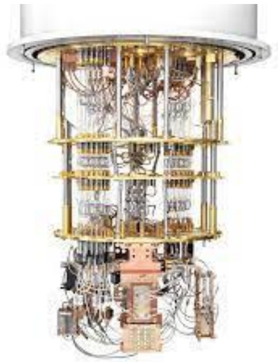
Google Bristlecone (Q72, '18)

IBM Q System One (Q20, '19)

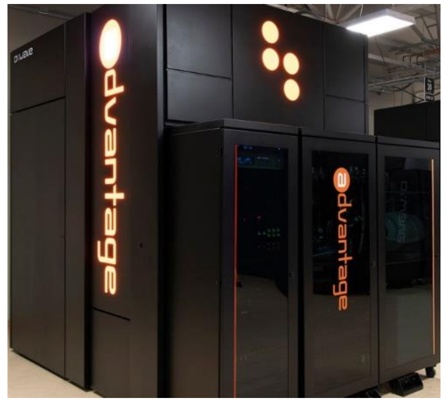


IBM Hummingbird (Q65, '20)

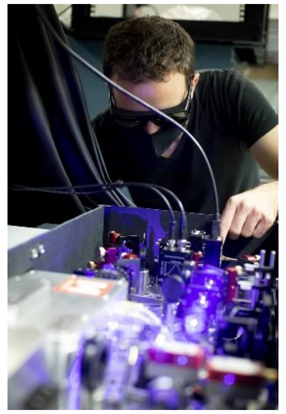
IonQ (Q32, '20)



Rigetti Aspen-8 (Q31, '20)



D-wave (Q5000+, '20)



Harvard & MIT (Q256, '21)

not for universal quantum computing

Quantum Computer – History

Quantum Computing in the NISQ era and beyond

John Preskill

Institute for Quantum Information and Matter and Walter Burke Institute for Theoretical Physics, California Institute of Technology, Pasadena CA 91125, USA

30 July 2018

Noisy Intermediate-Scale Quantum (NISQ) technology will be available in the near future. Quantum computers with 50-100 qubits may be able to perform tasks which surpass the capabilities of today's classical digital computers, but noise in quantum gates will limit the size of quantum circuits that can be executed reliably. NISQ devices will be useful tools for exploring many-body quantum physics, and may have other useful applications, but the 100-qubit quantum computer will not change the world right away — we should regard it as a significant step toward the more powerful quantum technologies of the future. Quantum technologists should continue to strive for more accurate quantum gates and, eventually, fully fault-tolerant quantum computing.

QUANTIZED COLUMNS

Why I Called It 'Quantum Supremacy'

John Preskill

Contributing Columnist

October 2, 2019

Researchers finally seem to have a quantum computer that can outperform a classical computer. But what does that really mean?

The quantum supremacy milestone allegedly achieved by Google is a pivotal step in the quest for practical quantum computers. I thought it would be useful to have a word for the era that is now dawning, so I recently made one up: NISQ. (It rhymes with risk.) This stands for "noisy intermediate-scale quantum." Here "intermediate-scale" refers to the size of quantum computers that are now becoming available: potentially large enough to perform certain highly specialized tasks beyond the reach of today's supercomputers. "Noisy" emphasizes that we have imperfect control over the qubits, resulting in small errors that accumulate over time; if we attempt too long a computation, we're not likely to get the right answer.

https://www.quantamagazine.org/john-preskill-explains-quantum-supremacy-20191002/

Quantum Computer – Current Trends

The Era of Quantum Supremacy, started by Google ('19)

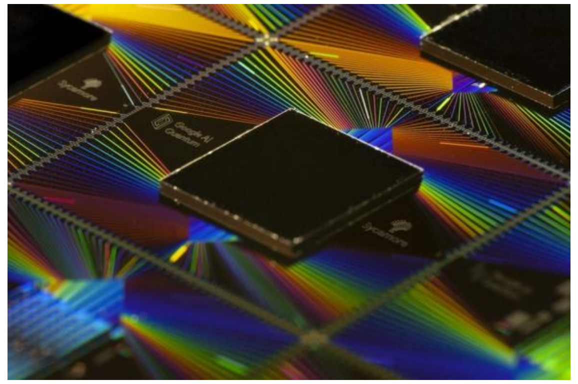
nature

Article | Published: 23 October 2019

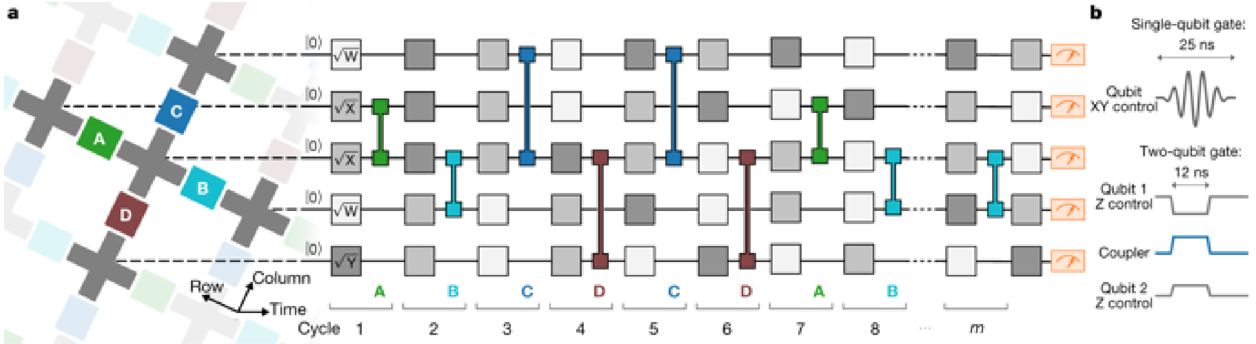
Quantum supremacy using a programmable superconducting processor

Frank Arute, Kunal Arya, [...] John M. Martinis

Our Sycamore processor takes about 200 seconds to sample one instance of a quantum circuit a million times—our benchmarks currently indicate that the equivalent task for a state-of-the-art classical supercomputer would take approximately 10,000 years.



Google's Sycamore quantum processor (Q54)



Control operations for the quantum supremacy circuits
(53 qubit, 1,113 single-qubit gates and 430 two-qubit gates)

cf. **IBM argued** that an ideal simulation of the same task can be performed **on a classical system in 2.5 days**

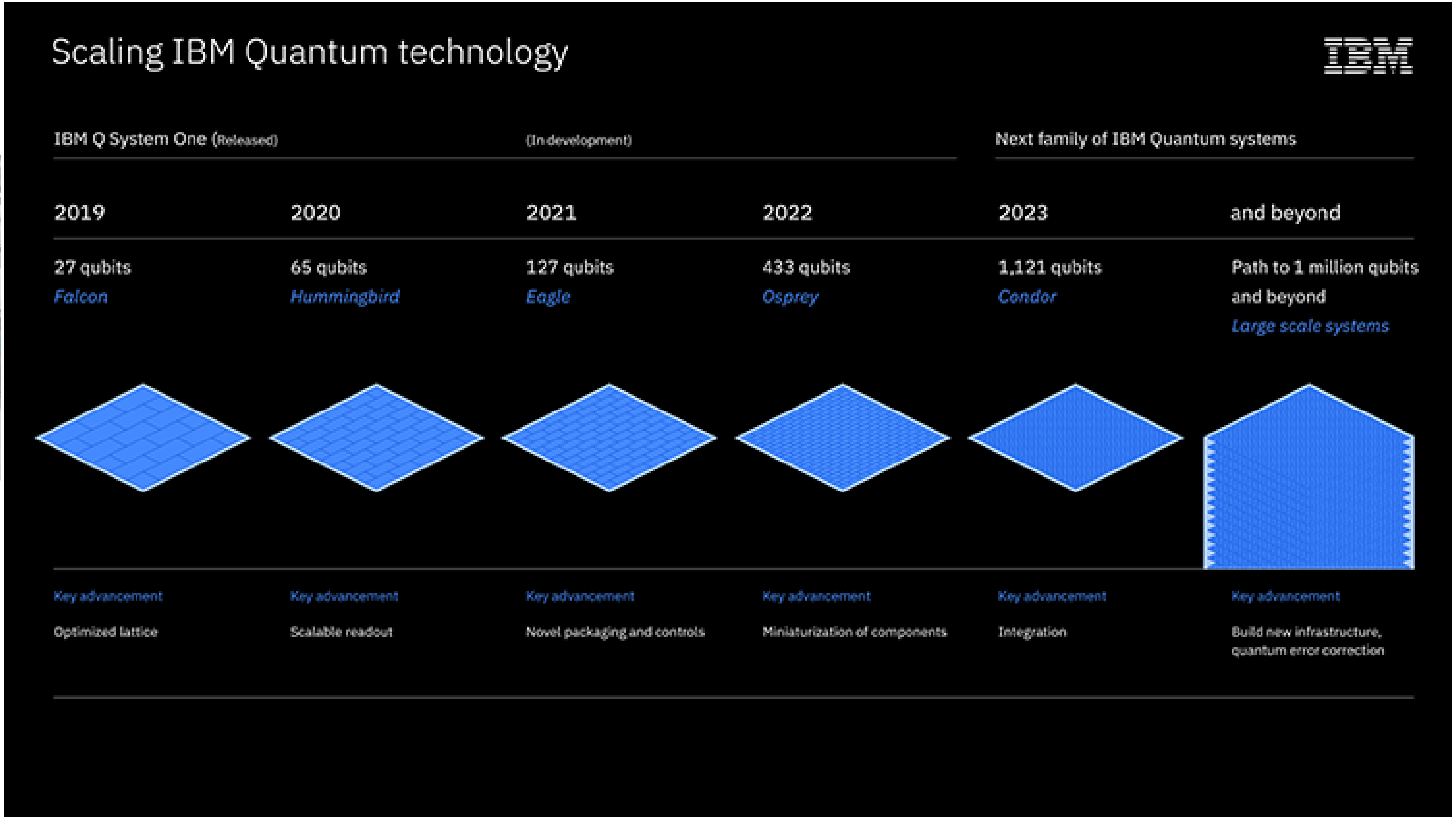
Quantum Computer – Current Trends

Quantum computer is coming true (1000+Qubit in 2023 by IBM)

IBM researchers have already installed the mounting hardware for a jumbo cryostat big enough to hold a quantum computer with 1 million qubits. CONNIE ZHOU/IBM

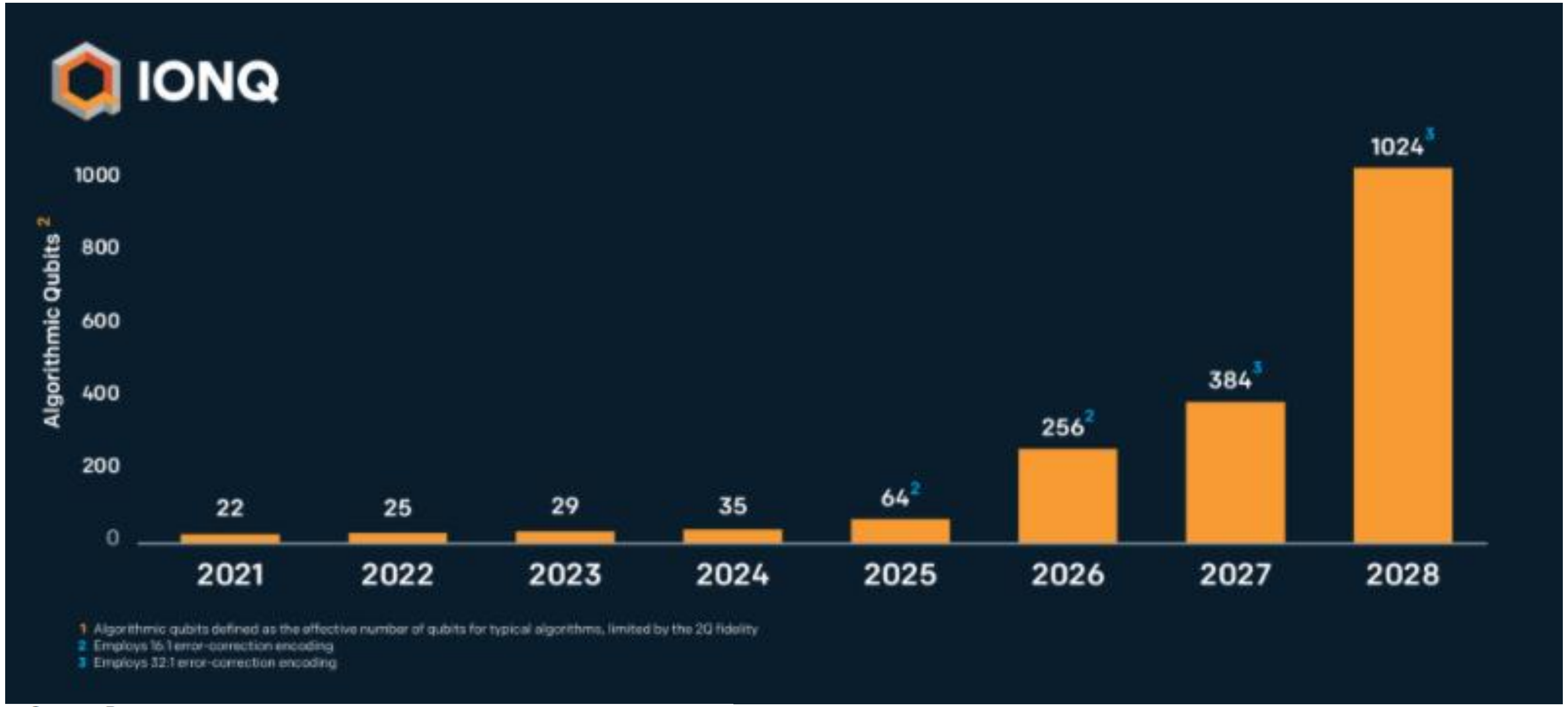
IBM promises 1000-qubit quantum computer—a milestone—by 2023

By Adrian Cho | Sep. 15, 2020 , 5:45 PM



Quantum Computer – Current Trends

Quantum computer is coming true (1000+Qubit in 2028 by IonQ)



System Parameters

Physical Qubits	Average 2Q Fidelity	Error Correction Overhead
100,000	99.99%	16:1

show advanced parameters   reset calculator

Quantum Compute Power

Algorithmic Qubits	Expected QV	Methodology
660	4.78E+198	Error Corrected Qubits

"**<u>IonQ</u>** plans to **<u>double the number of qubits every eight months</u>** for the next few years"
(Peter Chapman, IonQ CEO)

https://ionq.com/algorithmic-qubit-calculator

When will classical cryptography be fully attacked?

Theoretical quantum resources for solving ECDLP and factoring problems using Shor Algorithm

ECDLP in $E(\mathbb{F}_p)$ simulation results					Factoring of RSA modulus N interpolation from [21]		
$\lceil \log_2(p) \rceil$ bits	#Qubits	#Toffoli gates	Toffoli depth	Sim time sec	$\lceil \log_2(N) \rceil$ bits	#Qubits	#Toffoli gates
110	1014	$9.44 \cdot 10^9$	$8.66 \cdot 10^9$	273	512	1026	$6.41 \cdot 10^{10}$
160	1466	$2.97 \cdot 10^{10}$	$2.73 \cdot 10^9$	711	1024	2050	$5.81 \cdot 10^{11}$
192	1754	$5.30 \cdot 10^{10}$	$4.86 \cdot 10^{10}$	1 149	—	—	—
224	2042	$8.43 \cdot 10^{10}$	$7.73 \cdot 10^{10}$	1 881	2048	4098	$5.20 \cdot 10^{12}$
256	2330	$1.26 \cdot 10^{11}$	$1.16 \cdot 10^{11}$	3 848	3072	6146	$1.86 \cdot 10^{13}$
384	3484	$4.52 \cdot 10^{11}$	$4.15 \cdot 10^{11}$	17 003	7680	15362	$3.30 \cdot 10^{14}$
521	4719	$1.14 \cdot 10^{12}$	$1.05 \cdot 10^{12}$	42 888	15360	30722	$2.87 \cdot 10^{15}$

Table 2: Resource estimates of Shor's algorithm for computing elliptic curve discrete logarithms in $E(\mathbb{F}_p)$ versus Shor's algorithm for factoring an RSA modulus N.

- Martin Rötteler et al, "Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms", 2017

When will classical cryptography be fully attacked? – NISQ era

Theory and Reality (in Shor Algorithm)

$\lceil \log_2(N) \rceil$ bits	#Qubits	#Toffoli gates
512	1026	$6.41 \cdot 10^{10}$
1024	2050	$5.81 \cdot 10^{11}$
–	–	–
2048	4098	$5.20 \cdot 10^{12}$

Currently, there is no quantum computer with 4098 qubits

Minimum required qubits for attacking RSA-2048 : 4098 qubits
Google Bristlecone : 72 qubits

Then, will RSA-2048 be broken if a quantum computer with 4098 qubits is realized?

- Is it possible to implement arbitrary multi-qubit gates like Toffoli gate?
- Quantum gate has error rate.
- How long will it take to maintain quantum coherence in quantum chips?
- Is it possible to apply 2-qubit gate (e.g. CNOT) to any two qubits in any random position?

When will classical cryptography be fully attacked?

Theory and Reality (in Shor Algorithm)

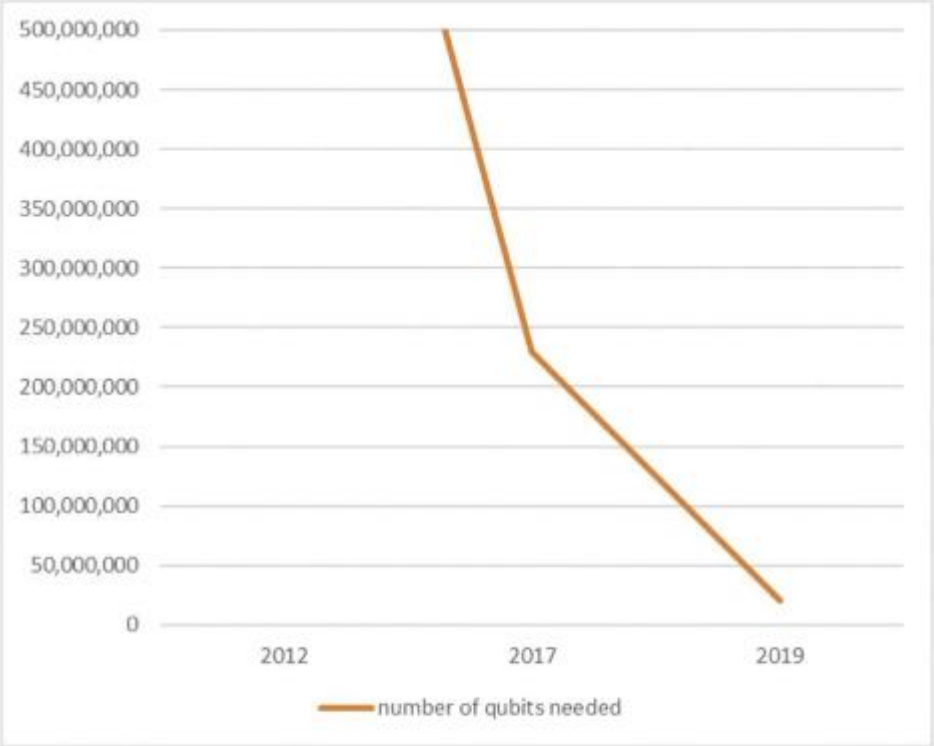
How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits

Craig Gidney[1,*] and Martin Ekerå[2]

[1] *Google Inc., Santa Barbara, California 93117, USA*
[2] *KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden*
Swedish NCSA, Swedish Armed Forces, SE-107 85 Stockholm, Sweden

(Dated: December 6, 2019)

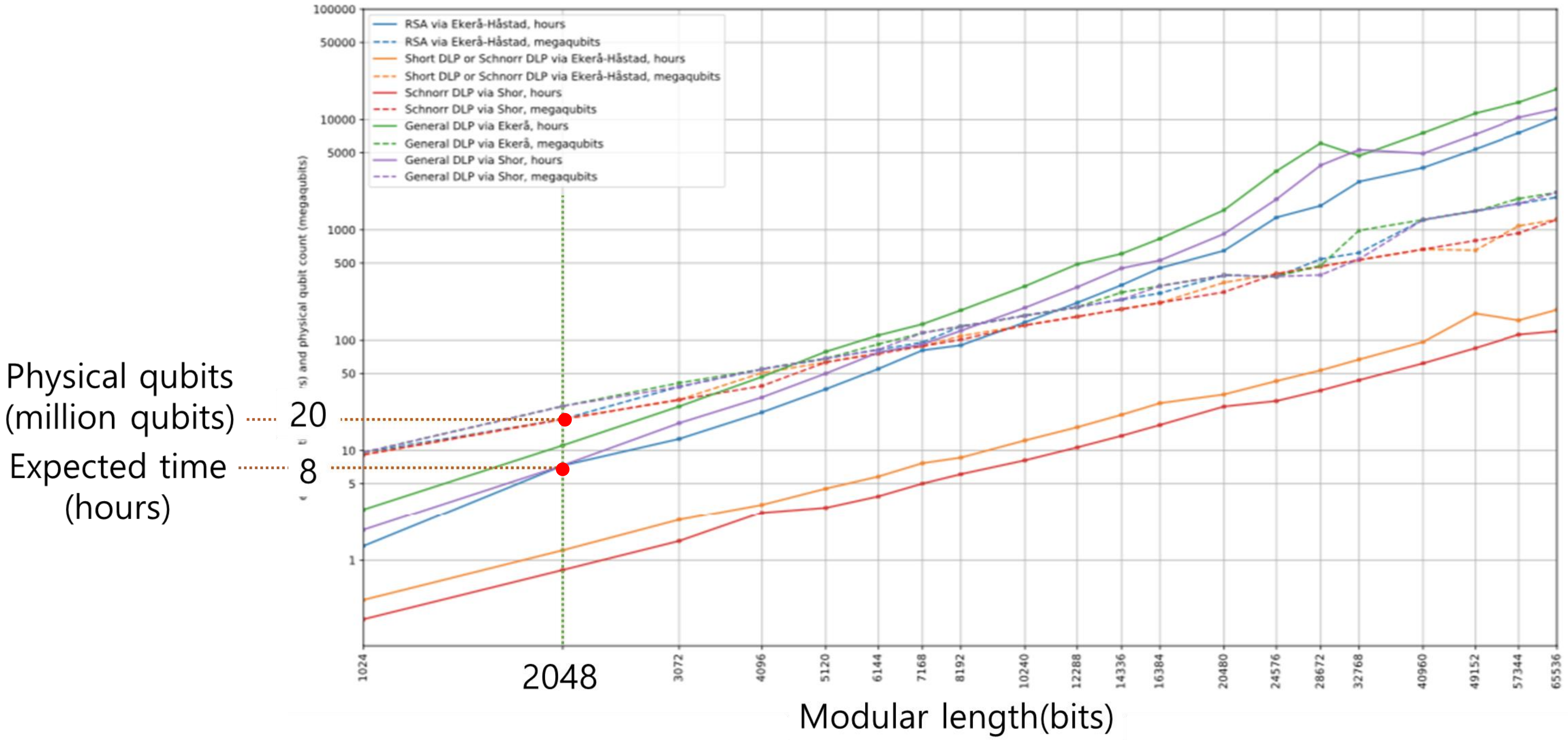
We significantly reduce the cost of factoring integers and computing discrete logarithms in finite fields on a quantum computer by combining techniques from Shor 1994, Griffiths-Niu 1996, Zalka 2006, Fowler 2012, Ekerå-Håstad 2017, Ekerå 2017, Ekerå 2018, Gidney-Fowler 2019, Gidney 2019. We estimate the approximate cost of our construction using plausible physical assumptions for large-scale superconducting qubit platforms: a planar grid of qubits with nearest-neighbor connectivity, a characteristic physical gate error rate of 10^{-3}, a surface code cycle time of 1 microsecond, and a reaction time of 10 microseconds. We account for factors that are normally ignored such as noise, the need to make repeated attempts, and the spacetime layout of the computation. When factoring 2048 bit RSA integers, our construction's spacetime volume is a hundredfold less than comparable estimates from earlier works (Fowler et al. 2012, Gheorghiu et al. 2019). In the abstract circuit model (which ignores overheads from distillation, routing, and error correction) our construction uses $3n + 0.002n \lg n$ logical qubits, $0.3n^3 + 0.0005n^3 \lg n$ Toffolis, and $500n^2 + n^2 \lg n$ measurement depth to factor n-bit RSA integers. We quantify the cryptographic implications of our work, both for RSA and for schemes based on the DLP in finite fields.



When will classical cryptography be fully attacked?

Theory and Reality (in Shor Algorithm)

- **Assumption in this paper**
  1. Large-scale superconducting qubit platform
  2. Error Correction Code: Surface Code
  3. Physical gate error: 10^{-3}
  4. Code distance : 27

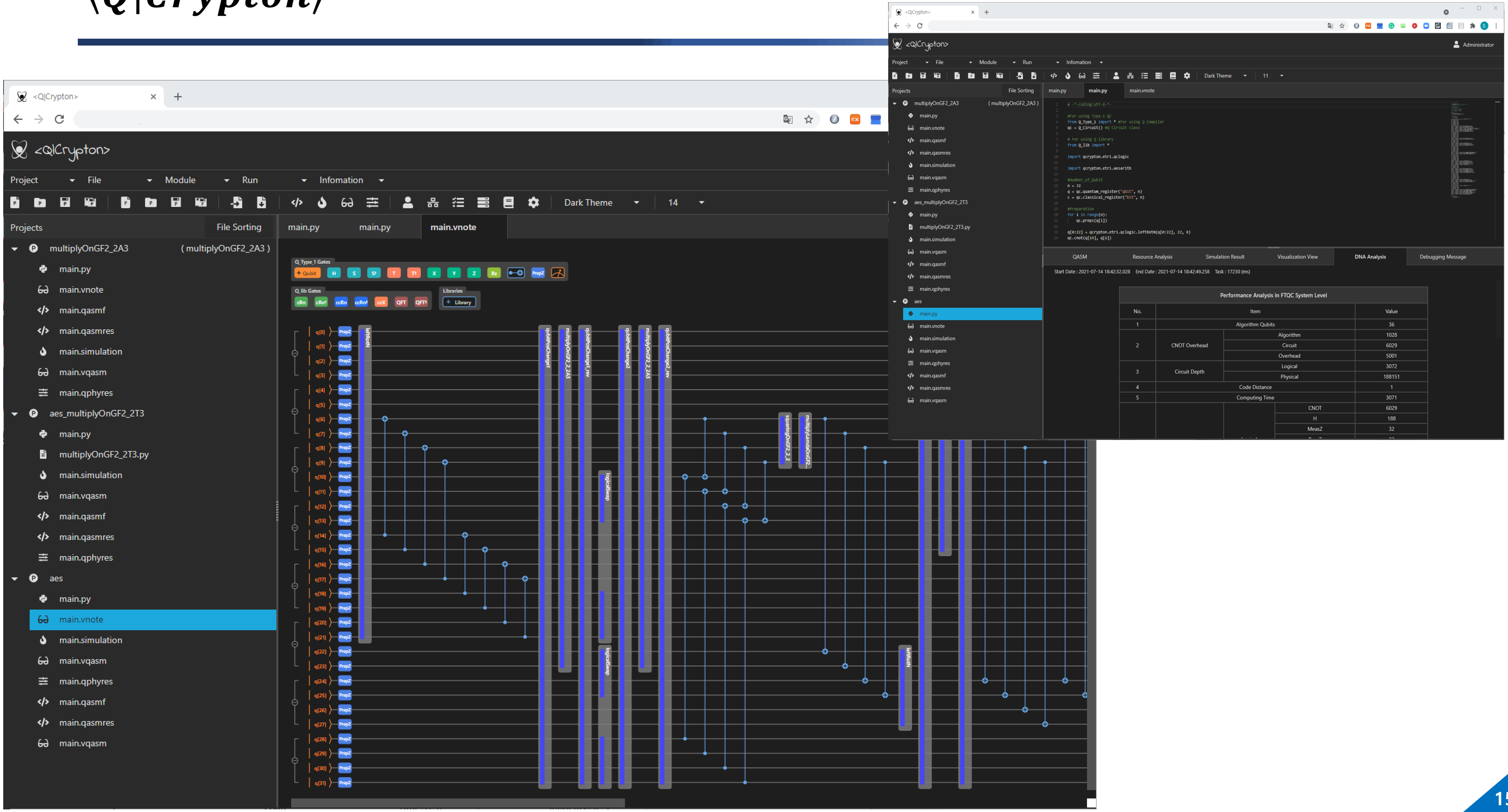


$\langle Q|Crypton\rangle$ - Motivation

We want to know the accurate Quantum Security of
not only RSA, but also AES, ECC, Hash, PQC, etc.

What if we have a platform that provides a fast and automated way to evaluate the exact quantum security of cryptographic algorithms?

$\langle Q|Crypton \rangle$

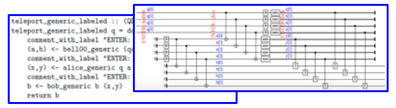


$\langle Q|Crypton \rangle$ - Concept

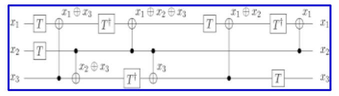
Quantum Resource Estimation-based Quantum Security Evaluation ?

Universal Quantum Computing Process

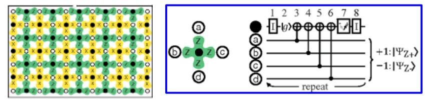
1 QP Quantum Program — Q-programming Code (reversible gate + quantum gate)



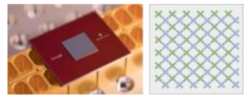
2 QC Quantum Compile — Elementary Gate (Clifford+T) Decomposition



3 FT Fault Tolerant Q-Comp. — Quantum Error Correction Code



4 HW HW Mapping — Dedicated Q-chip Architecture



Probabilistic Solution After many measurements

Correct Q-code

Quantum Algorithm Level
Quantum Queries
# of Qubits

+

Q-compiling Level
# of Qubits
Quantum Circuit Steps
# of Q-gates, T-depth

+

FTQC Level
# of Qubits
Quantum Circuit Steps
# of Q-gates, Comp. time

+

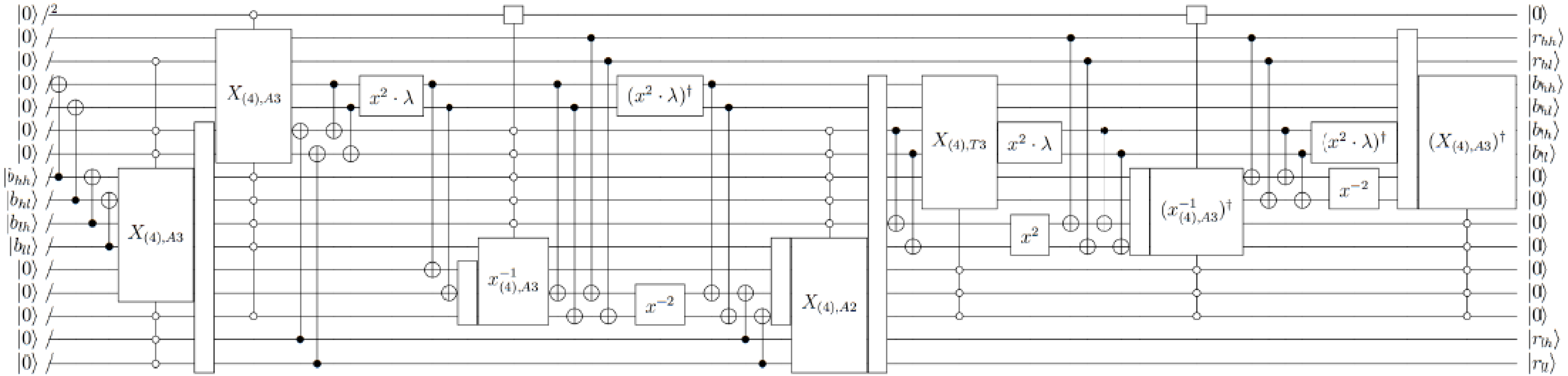
HW Level
# of Phy. Qubits
Quantum Circuit Steps
# of Q-gates, Comp. time

Total Quantum Resource Estimation for each Level (QASM/FTQC/Q-HW ➔ QASM/QHW-logical/QHW-FTQC)

→ Concrete Q-Security Estimation and Comparison

$\langle Q|Crypton\rangle$ - Concept

First Step: Implement quantum circuit in algorithm level



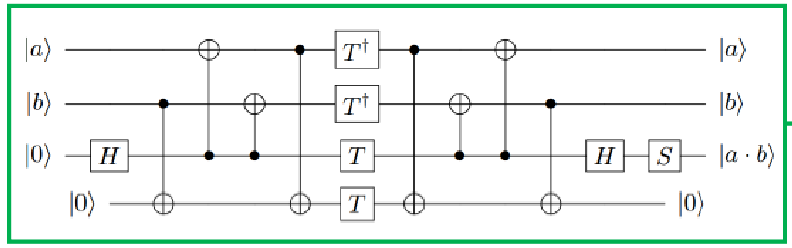
Quantum circuit for multiplicative inversion in GF(2^8), used for implementing AES s-box

* D. Chung et al., "Towards Optimizing Quantum Implementation of the AES S-box

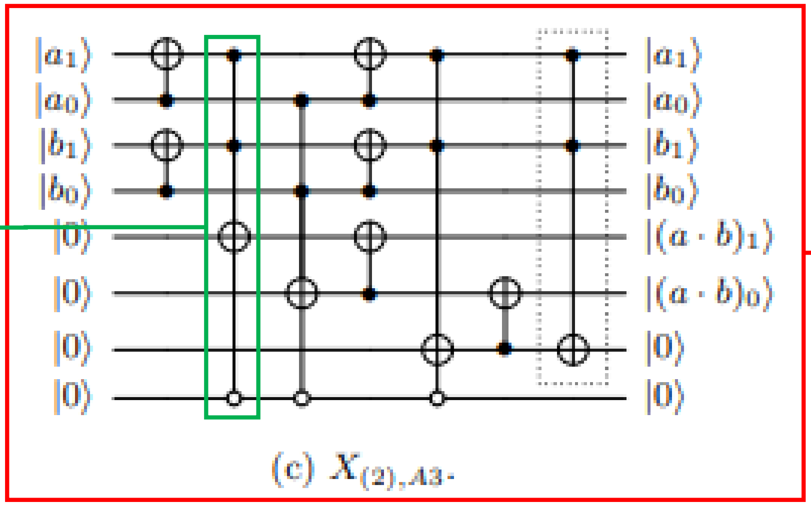
⟨Q|Crypton⟩ - Concept

First Step: Implement quantum circuit in algorithm level

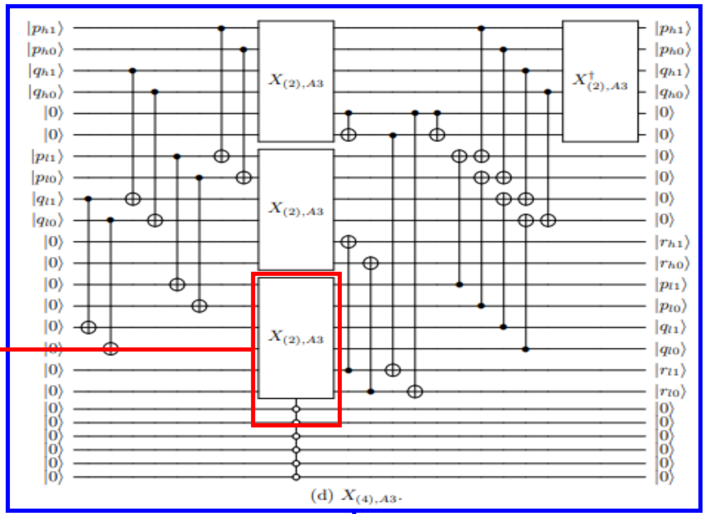
Quantum Arithmetic Circuit → Quantum Library Gate



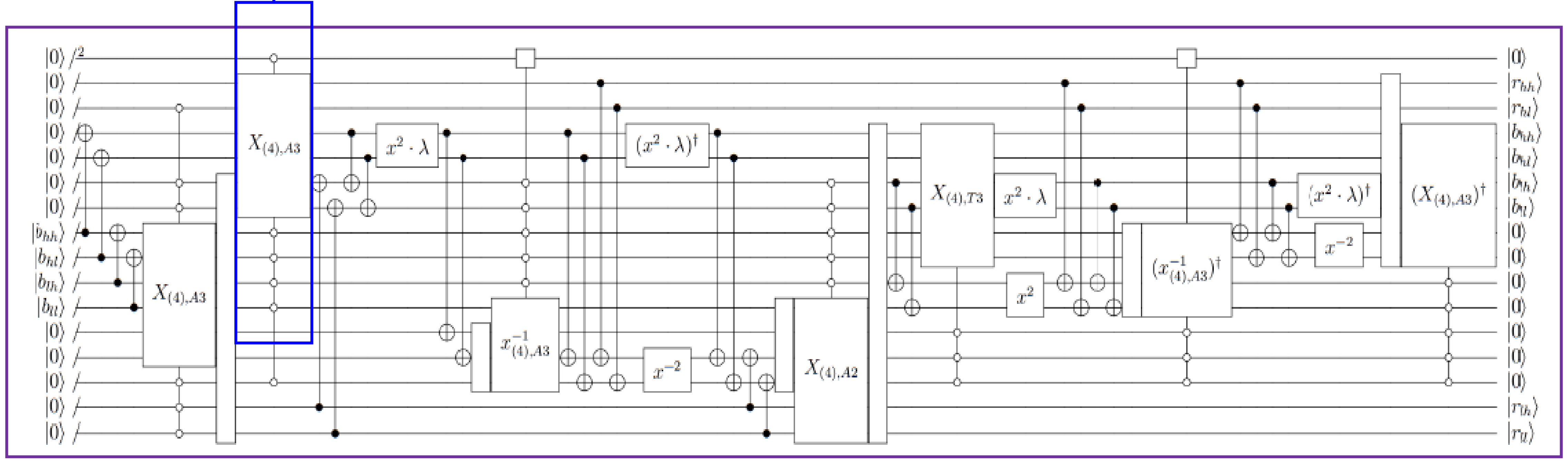
quantum circuits for AND



quantum circuits for multiplication in GF(2^2)



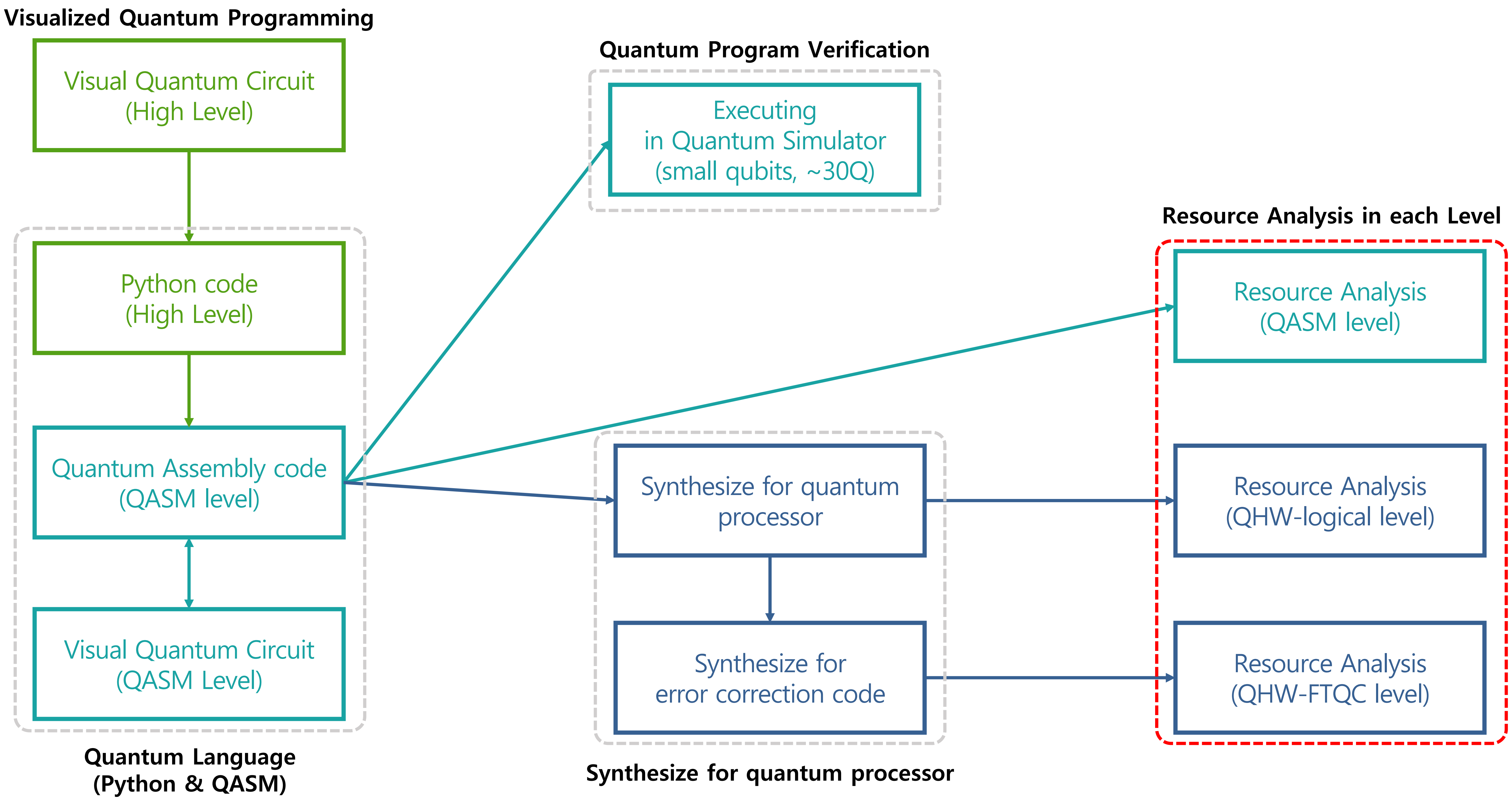
quantum circuit for multiplication in GF(2^4)



quantum circuit for multiplicative inversion in GF(2^8), used for implementing AES s-box

$\langle Q|Crypton\rangle$ - Concept

Next Step: Compile and synthesize to analyze quantum resource for the circuit



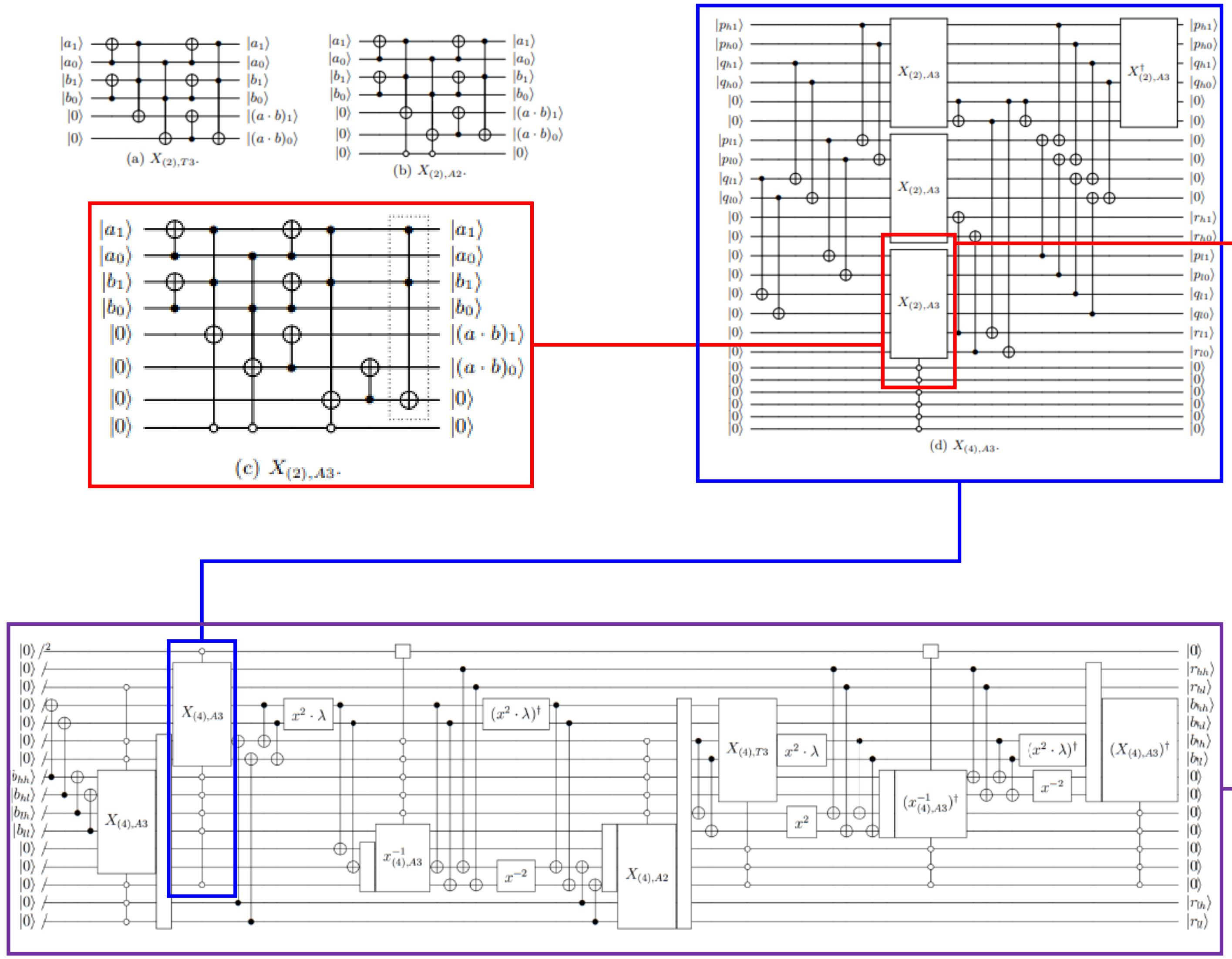
⟨Q|Crypton⟩ - Analysis Sequence

Implementation on ⟨Q|Crypton⟩ platform

Theoretical Algorithm (Multiplicative Inversion in AES)

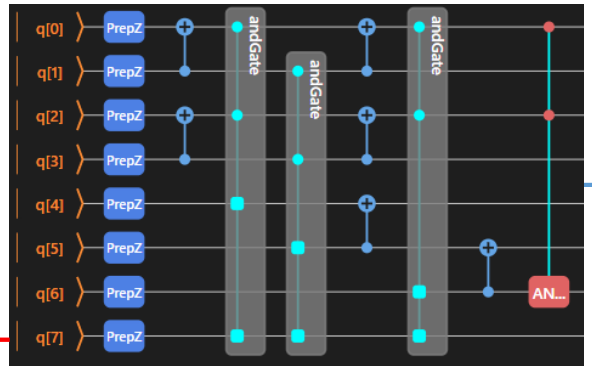
quantum circuits for multiplication in GF(2^2)

quantum circuit for multiplication in GF(2^4)

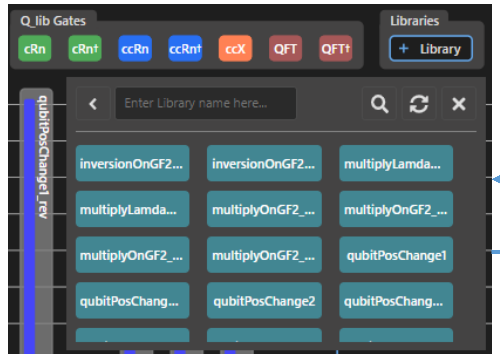


quantum circuit for multiplicative inversion in GF(2^8)

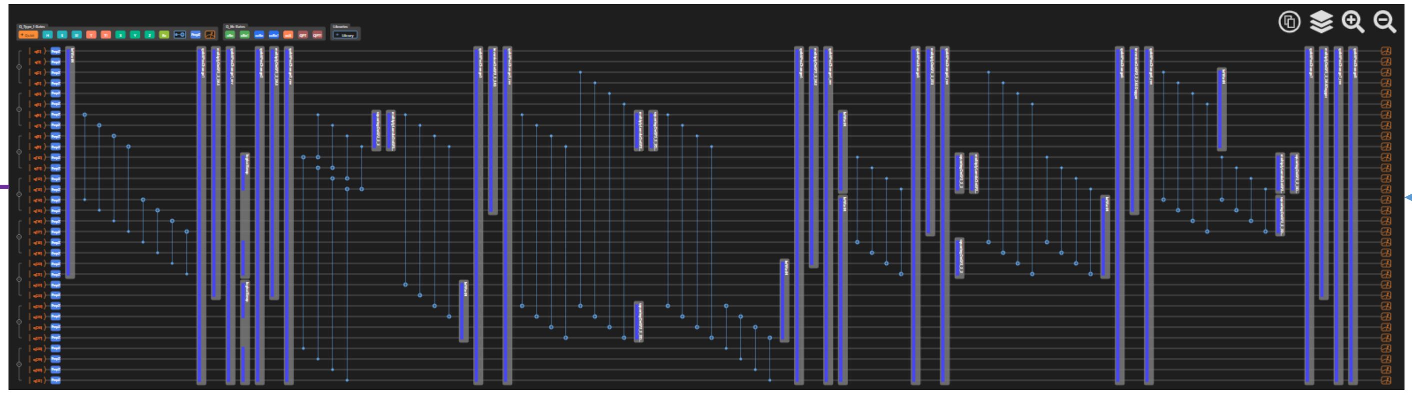
Implementation in ⟨Q|Crypton⟩



1. Draw quantum circuit and generate/share Q-library



2. Draw full circuit using Q-library



$\langle Q|Crypton\rangle$ - Analysis Sequence

QASM Level Process on $\langle Q|Crypton\rangle$ platform

Python-based Q-Program Code Generation

3. Auto-generate python-based Q-Program code by one-click

main.py | main.pyv | main_20210401_135057.dna | main.visualization

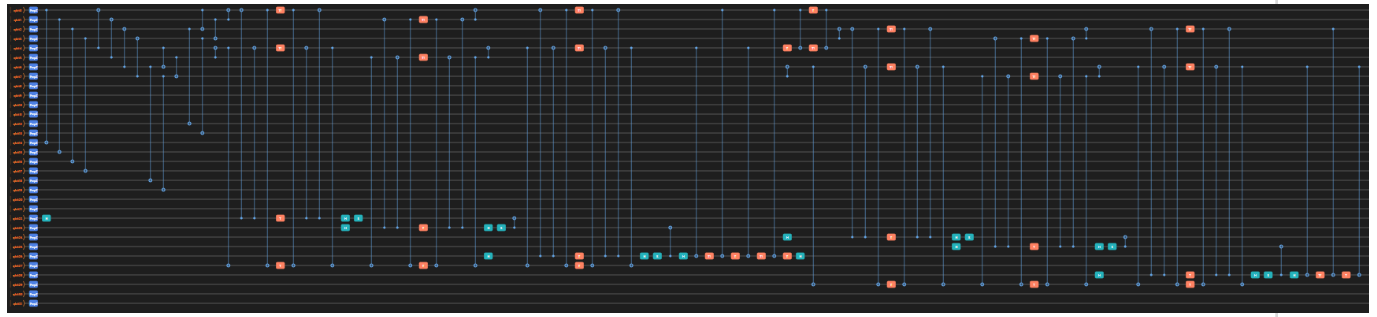
```python
# -*-coding:utf-8-*-

#For using Type-1 QC
from Q_Type_1 import * #For using Q Compiler
qc = Q_Circuit() #Q Circuit class

# For using Q library
from Q_lib import *

import qcrypton.etri.qclogic

import qcrypton.etri.aesarith

#Number_of_Qubit
n = 32
q = qc.quantum_register("qbit", n)
c = qc.classical_register("bit", n)

#Preparation
for i in range(n):
    qc.prepz(q[i])

q[0:22] = qcrypton.etri.qclogic.leftRotN(q[0:22], 22, 8)
qc.cnot(q[14], q[6])
qc.cnot(q[15], q[7])
qc.cnot(q[16], q[8])
qc.cnot(q[17], q[9])
qc.cnot(q[18], q[14])
qc.cnot(q[19], q[15])
qc.cnot(q[20], q[16])
qc.cnot(q[21], q[17])
q[0:32] = qcrypton.etri.aesarith.qubitPosChange1(q[0:32])
q[0:24] = qcrypton.etri.aesarith.multiplyOnGF2_2_2A3(q[0:24])
q[0:32] = qcrypton.etri.aesarith.qubitPosChange1_rev(q[0:32])
q[10:14], q[18:22] = qcrypton.etri.qclogic.logicalSwap(q[10:14], q[18:22], 4)
q[22:26], q[28:32] = qcrypton.etri.qclogic.logicalSwap(q[22:26], q[28:32], 4)
q[0:32] = qcrypton.etri.aesarith.qubitPosChange2(q[0:32])
q[0:24] = qcrypton.etri.aesarith.multiplyOnGF2_2_2A3(q[0:24])
q[0:32] = qcrypton.etri.aesarith.qubitPosChange2_rev(q[0:32])
qc.cnot(q[28], q[10])
qc.cnot(q[6], q[10])
qc.cnot(q[29], q[11])
qc.cnot(q[7], q[11])
qc.cnot(q[30], q[12])
qc.cnot(q[8], q[12])
qc.cnot(q[31], q[13])
qc.cnot(q[9], q[13])
q[6:10] = qcrypton.etri.aesarith.squaringOnGF2_2_2(q[6:10])
q[6:10] = qcrypton.etri.aesarith.multiplyLamdaOnGF2_2_2(q[6:10])
qc.cnot(q[6], q[22])
qc.cnot(q[7], q[23])
qc.cnot(q[8], q[24])
qc.cnot(q[9], q[25])
q[22:28] = qcrypton.etri.qclogic.leftRotN(q[22:28], 6, 4)
q[0:32] = qcrypton.etri.aesarith.qubitPosChange3(q[0:32])
q[0:16] = qcrypton.etri.aesarith.inversionOnGF2_2_2A3(q[0:16])
q[0:32] = qcrypton.etri.aesarith.qubitPosChange3_rev(q[0:32])
qc.cnot(q[6], q[24])
```

Q-Compile and Analysis

4. Compile to QASM

```
Qubit qbit0
Qubit qbit1
Qubit qbit2
Qubit qbit3
Qubit qbit4
Qubit qbit5
Qubit qbit6
Qubit qbit7
…
CNOT qbit5,qbit7
CNOT qbit2,qbit12
CNOT qbit3,qbit13
CNOT qbit0,qbit2
CNOT qbit1,qbit3
CNOT qbit1,qbit0
CNOT qbit5,qbit4
H qbit22
CNOT qbit4,qbit27
CNOT qbit22,qbit0
CNOT qbit22,qbit4
CNOT qbit0,qbit27
Tdag qbit0
Tdag qbit4
T qbit22
T qbit27
…
```

5. Auto-draw quantum circuit (QASM level)



6. Q-resource analysis (QASM level)

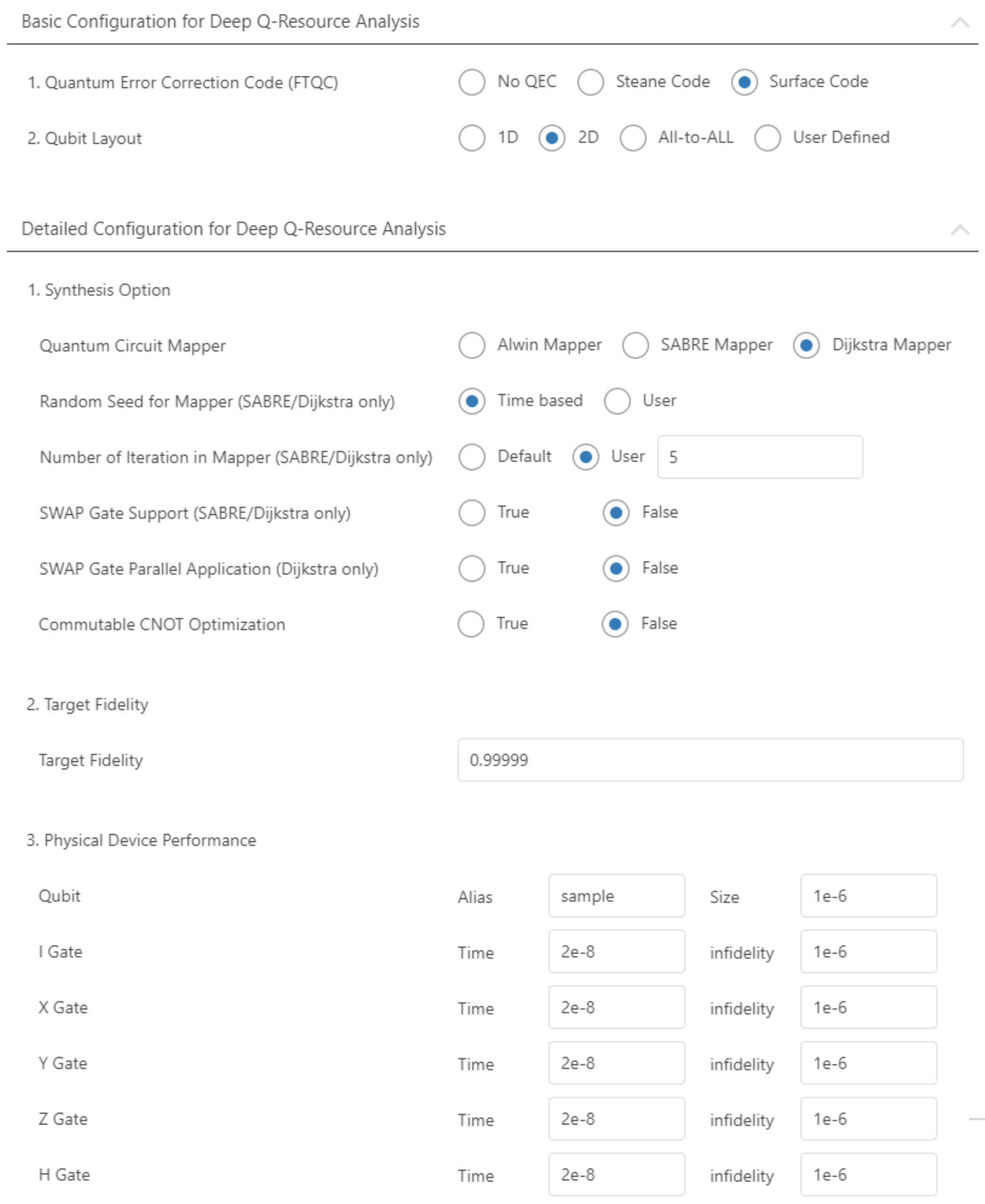
Resource Analysis in Compile Level		
No.	Item	Value
1	Qubit	32
2	Cbit	32
3	H	188
4	S	45
5	T	286
6	Tdag	237
7	CNOT	1028
8	Prepz	32
9	MeasZ	32
10	Total gates	1848
11	Depth	454
12	KQ	14528
13	T-Depth	103

$\langle Q|Crypton\rangle$ - Analysis Sequence

Quantum Resource Analysis on $\langle Q|Crypton\rangle$ platform

Detailed Q-Analysis Option in Quantum HW Level

7. Select options for detailed Q-resource analysis



- **Quantum Error Correction code**
  - None / Steane / Surface
- **Qubit Layout**
  - 1D / 2D / All-to-All / User-defined
- **Synthesis Option**
  - Circuit mapper, random seed, etc
- **Target Fidelity for the Circuit**
  - Real number (0~1)
- **Physical Device Performance**
  - Processing time and fidelity for each gate

⟨Q|$Crypton$⟩ - Analysis Sequence

Quantum Resource Analysis on ⟨Q|$Crypton$⟩ platform

Detailed Q-Analysis Result in Quantum HW Level

8-1. Detailed Q-resource analysis (Steane-1D-Fidelity:0.999)

Performance Analysis in FTQC System Level				
No.	Item			Value
1	Algorithm Qubits			32
2	CNOT Overhead	Algorithm		1028
		Circuit		1028
		Overhead		0
3	Circuit Depth	Logical		455
		Physical		32541
4	Computing Time			0.00437775
5	Concatenation Level			1
6	Function List	Logical	CNOT	1028
			H	188
			MeasZ	32
			PrepZ	32
			S	45
			T	286
			Tdag	237
		Physical	CNOT	2638786
			H	37092
			MeasZ	32461
			PrepZ	28800
			S	3976
			X	2002
			Z	1974
7	Gate Depth	CNOT		454
		T-Gate		103
8	Physical Qubits	Data		1536
		Magic		25104

8-2. Detailed Q-resource analysis (Surface-2D-Fidelity:0.999)

Performance Analysis in FTQC System Level				
No.	Item			Value
1	Algorithm Qubits			36
2	CNOT Overhead	Algorithm		1028
		Circuit		5918
		Overhead		4890
3	Circuit Depth	Logical		2935
		Physical		496620
4	Code Distance			3
5	Computing Time			0.0298112799999995
6	Function List	Logical	CNOT	5918
			H	188
			MeasZ	32
			PrepZ	32
			S	45
			T	286
			Tdag	237
		Physical	CNOT	2275707
			H	151389
			MeasX	1368960
			MeasZ	37054956
			PrepX	296754
			PrepZ	281760
			X	502080
			Z	753120
7	Gate Depth	CNOT		2934
		T-Gate		216
8	Physical Qubits	Data		5508
		Magic		1200285

$\langle Q|Crypton\rangle$ - Core Features

1. **Visualized High-Level Programming for Quantum Algorithm**

2. **Quantum Libraries of the Arithmetic for Cryptographic Algorithms**

3. **Quantum Resource Analysis in QASM, QHW-logical, and QHW-FTQC Levels**

$\langle Q|Crypton\rangle$ - Short-term plan

1 Manual Updates

- User Manual for $\langle Q|Crypton\rangle$ Platform
- Programming Manual for Quantum Cryptographic Library

2 Quantum Programming

- Visual Quantum Circuit Update (for supporting large-scale qubits/gates, etc)
- Various Quantum Cryptographic algorithms and its Arithmetic Libraries
- Optimized Implementation for Quantum Analysis Algorithm
- Improved Quantum Library Registration Method

3 Quantum Resource Analysis

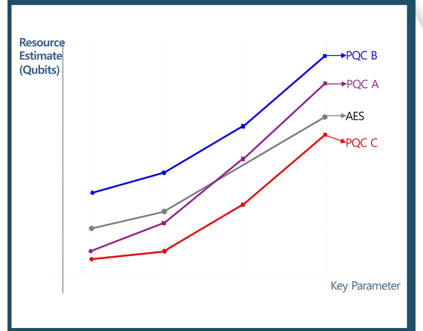
- Improved Accuracy for Quantum Resource Analysis
- Speed Up of Resource Analysis (for supporting large-scale qubits/gates, etc)

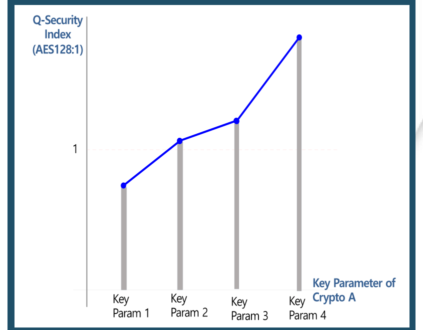
$\langle Q|Crypton \rangle$ - Long-term plan

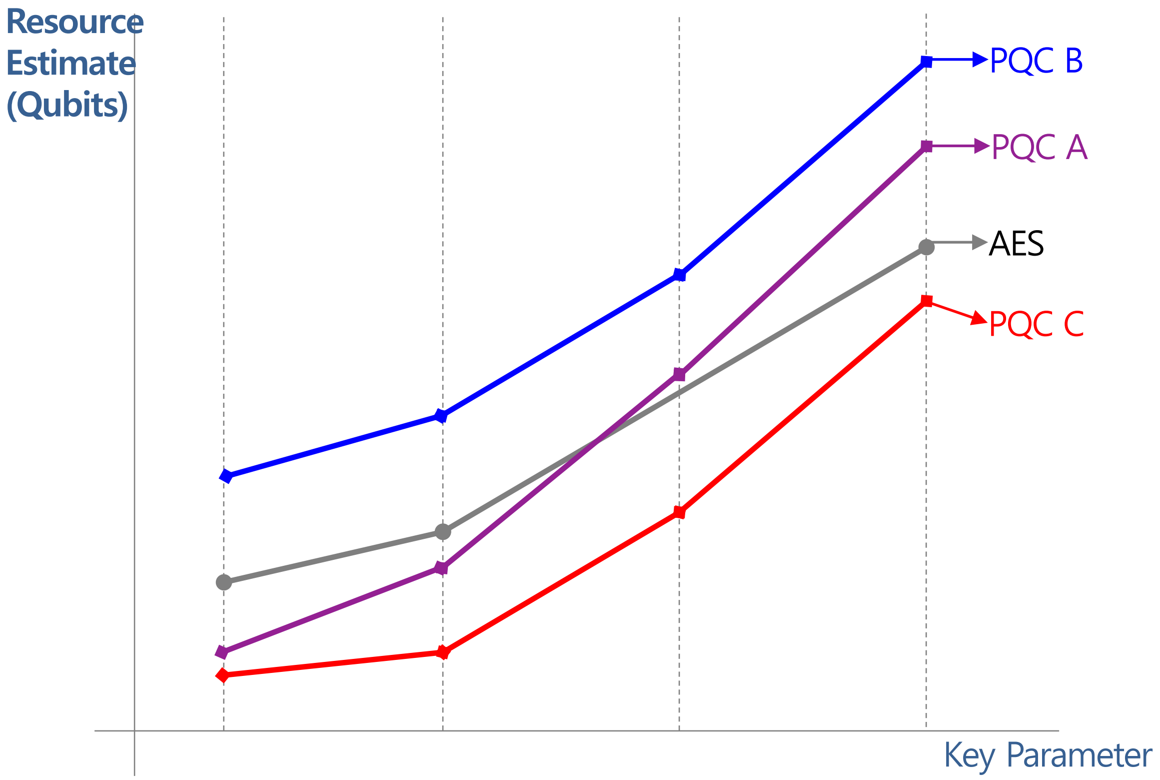
Graph for Users on Quantum Evaluation Results?

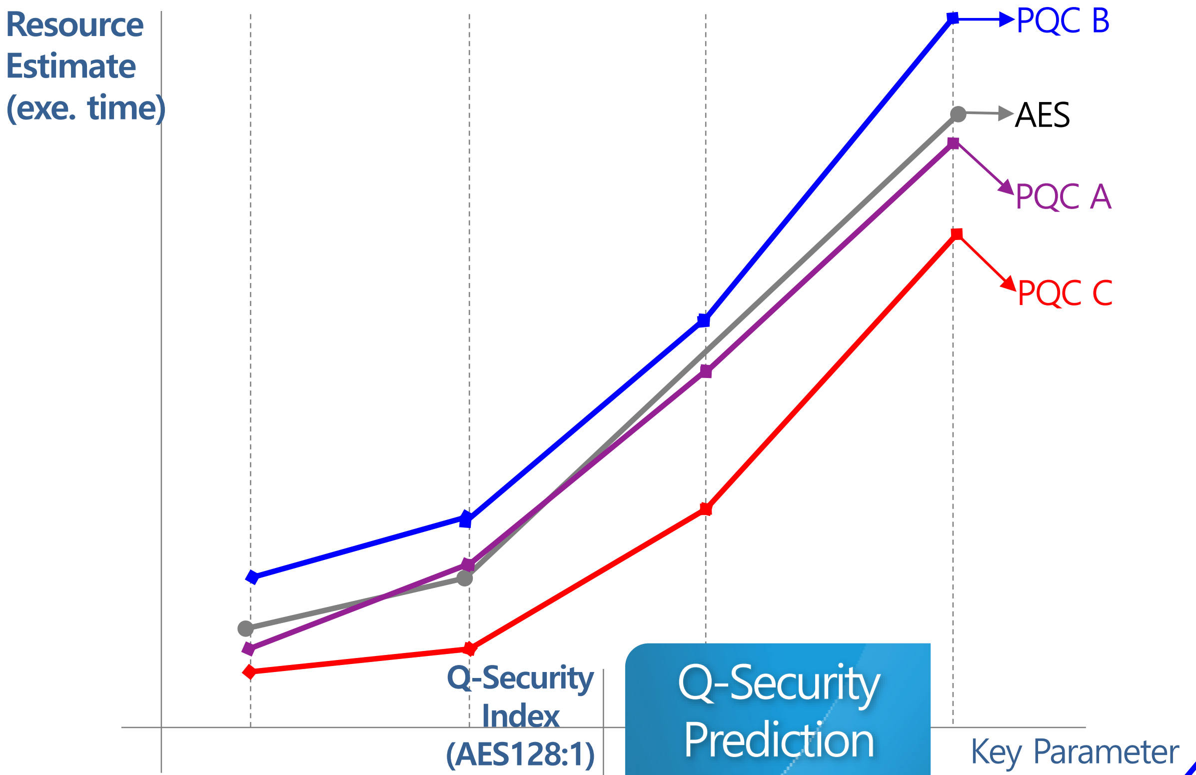
Q-eval. results

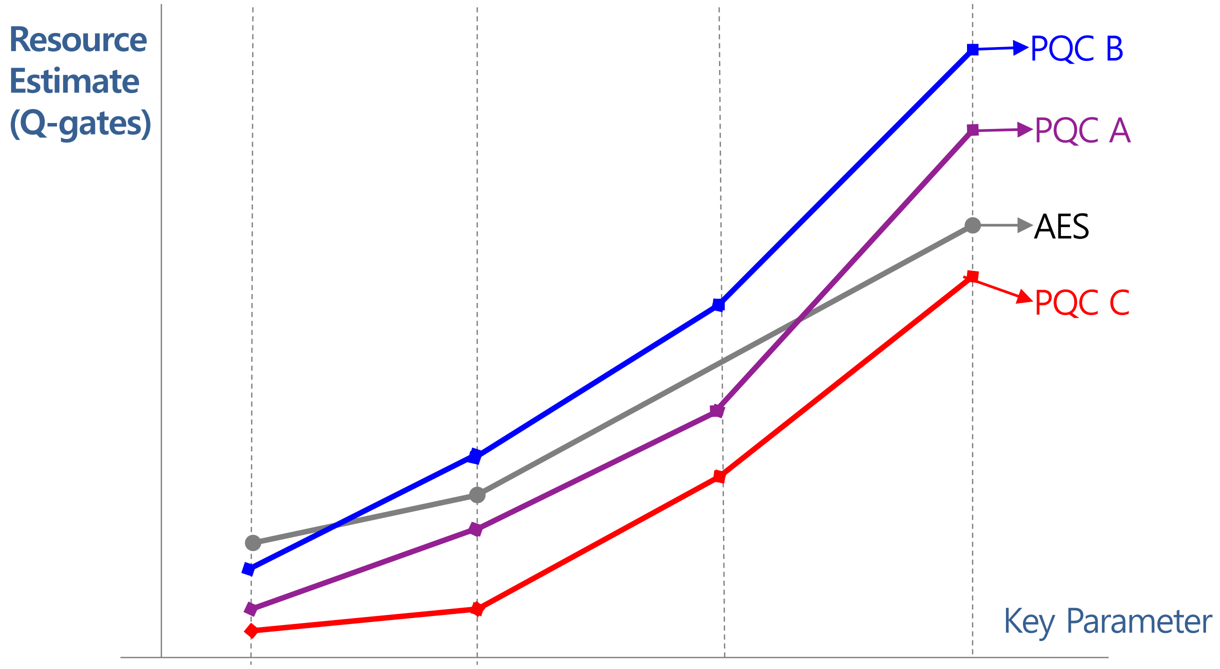
Q-security measurement/comparison to AES case

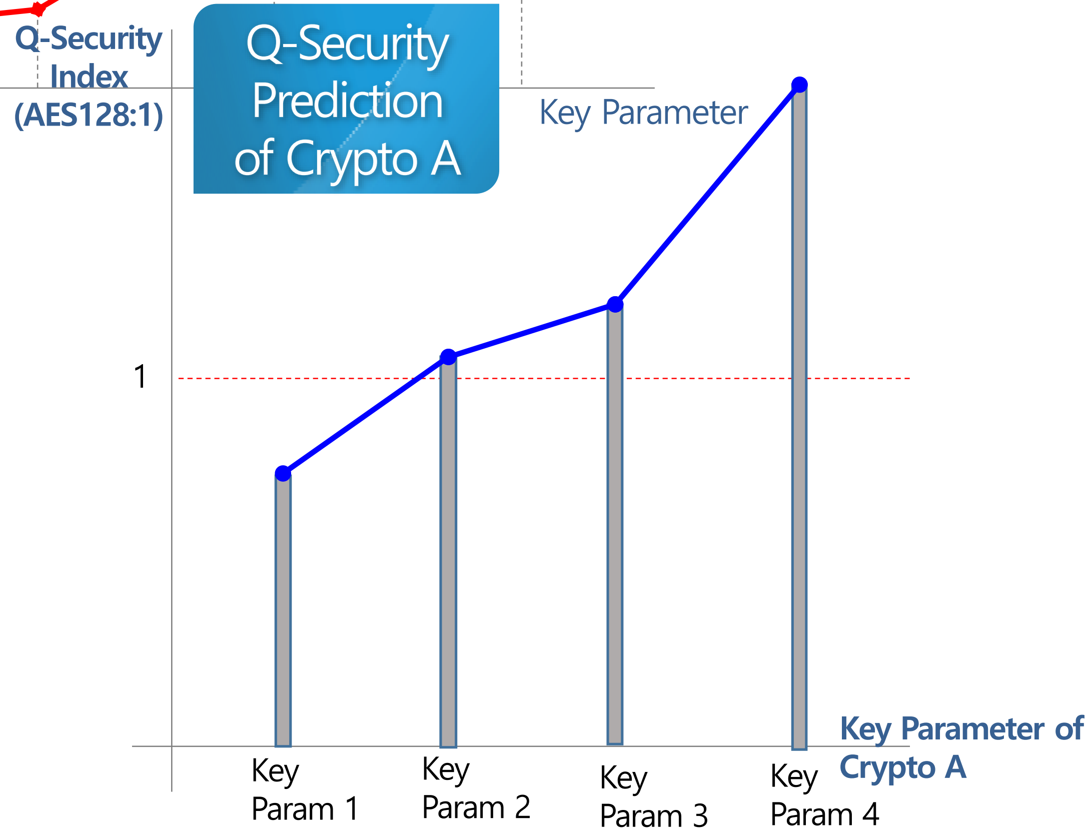












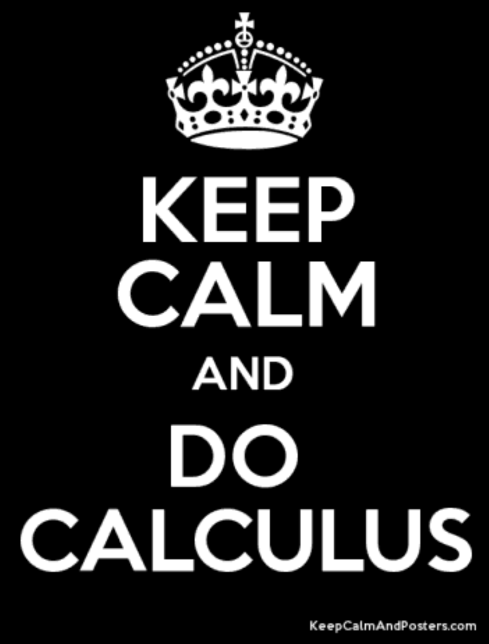
$\langle \boldsymbol{Q}|\boldsymbol{Crypton}\rangle$ - Open Plan

Open Plan

- We will open $\langle \boldsymbol{Q}|\boldsymbol{Crypton}\rangle$ platform **in Nov. 2021**

- Please, mail to junny@etri.re.kr if you want to use the platform
  - accessible only from permitted IP addresses

- We will welcome if you share your cryptographic or other quantum library for $\langle \boldsymbol{Q}|\boldsymbol{Crypton}\rangle$



$\langle Q|Crypton\rangle$